Pontifícia Universidade Católica do Paraná
Produtrônica

Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas

Implementação de estrutura de controle de sistema a eventos discretos em controlador lógico programável utilizando a teoria Controle Supervisório Modular Local

Agnelo Denis Vieira

Junho, 2003

Implementação de estrutura de controle de sistema a eventos discretos em controlador lógico programável utilizando a teoria Controle Supervisório Modular Local

Introdução:

Será apresentada neste trabalho uma metodologia para implementação do controle de sistemas a eventos discretos em controlador lógico programável (CLP) baseada nos resultados obtidos através da síntese de supervisores minimamente restritivos. Estes supervisores são determinados através de uma extensão da teoria de Controle Supervisório (Ramadge e Wonham, 1989) e que é denominada Controle Supervisório Modular Local (Queiroz e Cury 2000).

Nesta metodologia, Queiroz e Cury realizam a implementação do programa do CLP através de uma estrutura pré-definida, a qual é subdividida nos seguintes componentes: supervisores locais, estrutura de desabilitações, sistema produto e seqüências operacionais. A cada um destes componentes é estabelecido um arranjo em diagrama escada, o que confere à solução obtida características plenamente satisfatórias, tais como: geração do programa de CLP através de um procedimento pré-estabelecido, facilidade para interpretação do programa, facilidade de alteração do programa em função da adição ou remoção de módulos do sistema físico à controlar, ou mesmo, em função da substituição de componentes físicos que executam uma determinada tarefa por outros componentes com características operacionais distintas.

Além disto, esta metodologia não se restringe apenas à implementação de programas para CLP elaborados em diagrama escada, sendo facilmente realizada a conversão para outras linguagens aceitas por CLP's (texto estruturado, blocos funcionais, Grafcet), para linguagens de programação de alto nível (Pacal, C, ..) ou mesmo para implementação direta em *hardware* (circuitos eletrônicos microprocessados, sistemas pneumáticos).

Comparação entre a teoria de controle supervisório (Ramadge e Wonham 1989) e a teoria de controle supervisório modular local (Queiroz e Cury 2000):

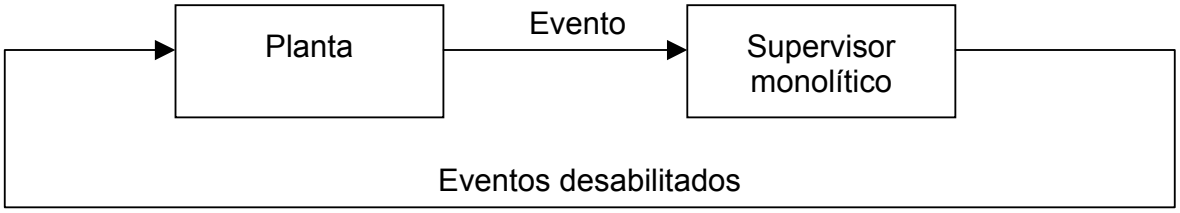
De forma a possibilitar a melhor compreensão da ação da teoria de controle supervisório modular local (TCSML) faz-se aqui uma comparação com a teoria de controle supervisório (TCS).

O controle de sistemas a eventos discretos realizado através da aplicação dos resultados obtidos com a TCS é um controle permissivo, ou seja, é obtido um supervisor que realiza o mapeamento da linguagem gerada pelo sistema sob supervisão e cuja função é a de desabilitar um conjunto de eventos controláveis, ou seja, que podem ser inibidos de acontecer pela ação deste supervisor. Se o

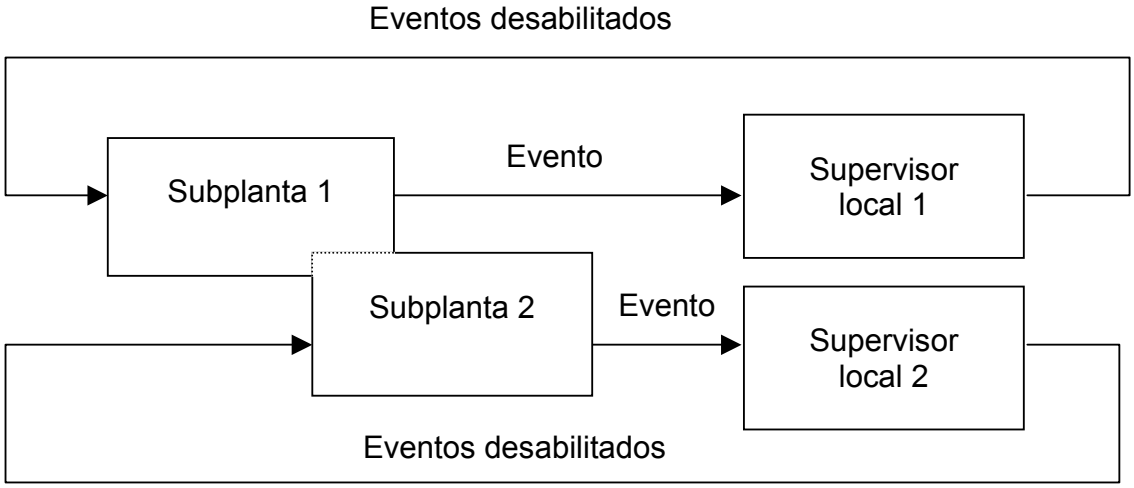
evento não está desabilitado pelo supervisor então ele é passível de ocorrer, sendo a evolução de eventos determinada pelo próprio sistema. Com esta teoria é obtido um único supervisor, denominado supervisor monolítico, que tem a função de atuar sob todo o conjunto de eventos controláveis, garantindo assim que a evolução do comportamento do sistema físico está em conformidade com todo o conjunto de restrições impostas ao mesmo.

Na TCSML são obtidos um conjunto de supervisores, denominados supervisores locais e que tem, cada um deles, a função de atuar sob um subconjunto de eventos controláveis fazendo com que o comportamento do sistema físico evolua de forma a respeitar uma dada restrição em particular. Neste caso o conjunto de eventos a serem desabilitados em função da evolução do comportamento do sistema é dado pela união dos eventos desabilitados por cada um dos supervisores locais.

Na figura abaixo são representadas a estrutura obtida com a teoria de controle supervisório e a teoria de controle supervisório modular local.



Estrutura de controle utilizando supervisor monolítico obtido pela TCS



Estrutura de controle utilizando supervisores locais obtidos pela TCSML

fig. 01 - Estruturas de controle segundo a TCS e a TCSML

Uma breve apresentação da aplicação da teoria de controle supervisório modular local para a síntese de supervisores locais minimamente restritivos:

A seguir são apresentados de forma sintética os passos necessários a serem seguidos para a obtenção do conjunto de supervisores locais que atendam, de forma minimamente restritiva, às especificações de funcionamento do sistema a controlar.

1 - desdobrar o sistema físico em módulos independentes;

2 - modelar da forma mais abstrata possível o funcionamento em malha aberta (sem controle) de cada módulo; na seção seguinte é apresentada uma sugestão de modelagem por geradores na qual o evento α_i corresponde ao início de operação do módulo "i" e o evento β_i corresponde ao final de operação deste módulo;

3 - calcular a mais refinada representação por sistema produto (RSP), ou seja, a representação do sistema constituída por diversas subplantas que não possuem eventos em comum; caso dois ou mais módulos representados no item anterior possuam eventos em comum deve ser realizada a composição síncrona dos mesmos e o resultado desta composição constitui a subplanta que representa os módulos que a originaram, os módulos que não possuem eventos em comum são representados por subplantas com estrutura idêntica à dos respectivos módulos;

4 - modelar através de geradores cada especificação isoladamente; considerar apenas os eventos relevantes;

5 - identificar o sub-alfabeto determinado por cada uma das especificações individuais, determinando assim quais subplantas são afetadas pelas mesmas;

6 - para cada uma das especificações, obter a linguagem da respectiva planta local através da composição síncrona das linguagens das subplantas afetadas;

7 - para cada uma das especificações, determinar a linguagem da respectiva especificação local através da composição síncrona entre a linguagem da especificação obtida no item 4 e a linguagem da planta local, obtida no item 6;

8 - para cada uma das especificações, calcular a máxima linguagem controlável contida na respectiva especificação local; esta linguagem representa a linguagem do supervisor local;

9 - verificar a modularidade das linguagens que representam os supervisores locais;

10 - se as linguagens que representam os supervisores locais são modulares está concluída esta etapa do projeto, em caso contrário deve ser resolvido o problema da não modularidade;

11 - minimização dos supervisores locais através do algorítimo de Vaz e Wonham;

Se a composição síncrona entre a linguagem dos supervisores locais puder ser representada por um gerador “TRIM” (todos os estados acessíveis e co-acessíveis) então está assegurado que:

- as linguagens são modulares e não há conflito entre a ação dos supervisores locais;

- a solução é ótima, ou seja, o resultado obtido será idêntico ao obtido utilizando a teoria de controle supervisório originalmente proposta por Ramadge e Wonham com a determinação de um supervisor monolítico.

Queiroz e Cury (2000) apresentam uma seção específica sobre a resolução de conflitos, não sendo neste trabalho discutida esta problemática.

Informalmente o resultado da não modularidade das linguagens dos supervisores locais pode ser explicado da seguinte forma:

> Como existe a possibilidade de haver eventos comuns entre as diversas plantas locais, há a possibilidade de existir uma dada seqüência de eventos “su” que é permitida pela linguagem representada por um dos supervisores locais (L(Sl1)) porém não é completamente permitida pela linguagem representada por outro supervisor local (L(Sl2)), contudo, um prefixo “s” desta seqüência pode ser permitido por esta segunda linguagem (L(Sl2)).
>
> Caso isto aconteça o sistema será levado a uma situação de bloqueio, pois uma tarefa, representada pela seqüência de eventos “su”, será iniciada com a seqüência “s” mas não poderá ser concluída, pois o segundo supervisor (Sl2), irá desabilitar um dado evento controlável impedindo a evolução do sistema para completar a tarefa representada pela seqüência “su”.

A minimização do supervisor referida na etapa 11 é fundamental no procedimento pois o número de estados do supervisor local obtido é determinante na viabilidade da implementação do programa do CLP, tendo em vista que o número total de estados dos supervisores determina o número de variáveis alocadas para a implementação dos supervisores.

Exemplo de aplicação da teoria de controle supervisório modular local:

De forma a permitir a compreensão do procedimento descrito na seção anterior é apresentada a seguir a resolução de um problema prático.

Na figura 02 é representado um sistema de manipulação de peças o qual realiza seqüêncialmente a remoção de peça bruta de um depósito inicial, a furação e o teste da peça e a armazenagem em um depósito final.

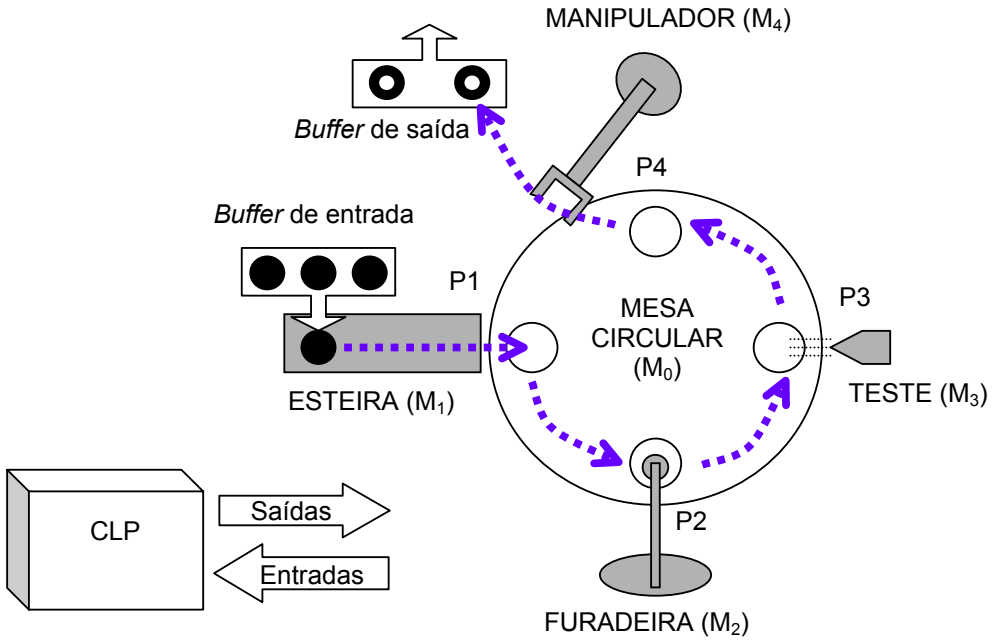


fig. 02 - Sistema de manipulação de peças

De forma a maximizar a análise da resolução do problema, a mesma será realizada utilizando o procedimento anteriormente descrito considerando o sistema físico constituído pelos seis módulos (M0 a M5) e constituído por cinco módulos (M0 a M4), verificando assim a influência sobre o programa para o CLP quando da inclusão de um módulo adicional. Será realizada também sua resolução através da aplicação da teoria de controle supervisório na sua forma original, verificando assim a influência sobre o programa para o CLP do número de estados do supervisor.

1 - desdobramento do sistema físico em módulos independentes;

é identificada a possibilidade de representar o sistema de manipulação de peças como sendo constituído de 6 módulos independentes:

M5 - distribuição (at. A, at. B, ger. vác. 1) (não representado no desenho)

M1 - transporte e classificação (at. C, at. D, esteira)

M0 - posicionamento rotativo

M2 - furação (at. E, at. G, furadeira)

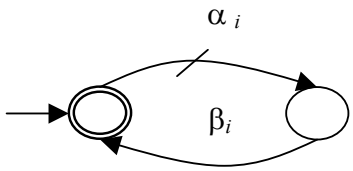
M3 - teste (at. T)

M4 - armazenagem (at. H, ger. vác. 2, pos. XY)

2 - representação dos módulos por geradores:

cada um dos módulos (M0 a M5) é representado por geradores conforme a estrutura abaixo:

M_i: i = 0 .. 5



onde:

α_i - corresponde ao início de operação do módulo "i"

β_i - corresponde ao final de operação deste módulo;

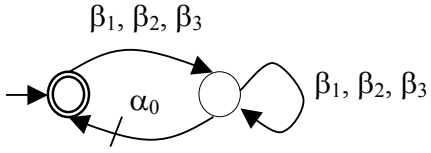
3 - representação por sistema produto RSP:

verifica-se que os eventos α_i e β_i utilizados em cada um dos geradores é restrito ao respectivo gerador, ou seja, não há módulos que contenham eventos em comum, desta forma o resultado obtido no item 2 já corresponde a RSP, assim cada subplanta (G0 a G5) possui estrutura idêntica ao respectivo módulo (M0 a M5)

4 - modelagem das especificações:

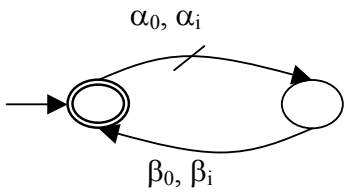
Ea:

evitar a operação do mód. de pos. rot. (G0) à toa, isto é, sem peça bruta em P1, sem peça furada em P2 ou sem peça testada em P3;



Eb_i: i = 1 .. 4

evitar a operação do módulo de transp. e clas. (G1), de fur. (G2), de teste (G3), de armaz. (G4) enquanto o mód. de pos. rot. (G0) estiver operando;

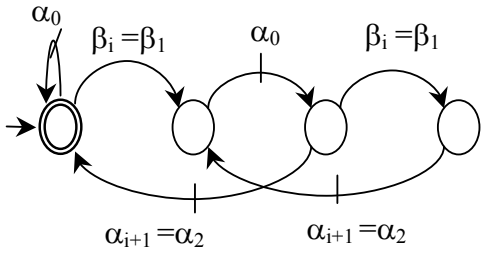


Ec1

evitar a sobreposição de peças na posição P1 da mesa;

evitar a furação sem peça bruta na posição P2 da mesa;

evitar girar a mesa com peça bruta na posição P2 da mesa;

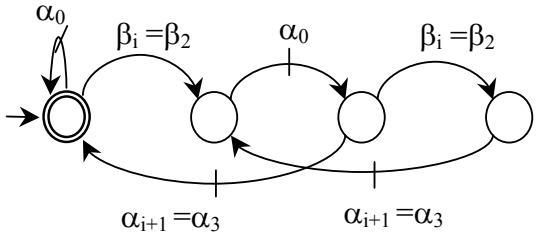


Ec2

evitar a furação de uma peça 2 vezes;

evitar o teste sem peça furada na posição P3 da mesa;

evitar girar a mesa com peça furada mas não testada na posição P3 da mesa;

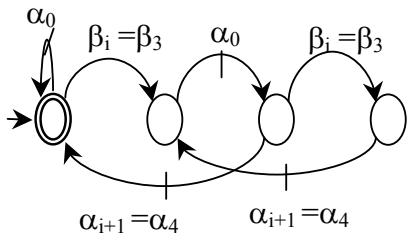


Ec3

evitar o teste de uma peça duas vezes;

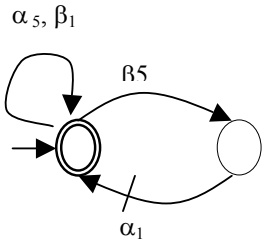
evitar o acionamento do mód. de armaz. sem peça testada na posição P4 da mesa;

evitar girar a mesa com peça na posição P4 da mesa;



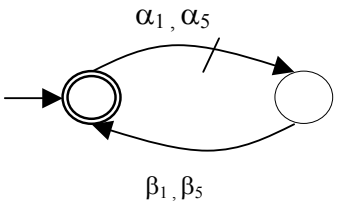
Ed1

evitar *under-flow* e *over-flow* do armazém intermediário (B1) entre os módulos de distribuição (G5) e transporte (G1) ;



Ed2

alternar a operação completa dos mód. de distribuição (G5) e transporte (G1), iniciando pelo mód. de distribuição (obtido através da especificação anterior);



5 - identificação dos sub-alfabetos definidos pelas especificações locais:

Especificação	Sub-alfabeto	Eventos relacionados											
		$\alpha 0$	$\beta 0$	$\alpha 1$	$\beta 1$	$\alpha 2$	$\beta 2$	$\alpha 3$	$\beta 3$	$\alpha 4$	$\beta 4$	$\alpha 5$	$\beta 5$
Ea	Σa	√			√		√		√				
Eb1	$\Sigma b1$	√	√	√	√								
Eb2	$\Sigma b2$	√	√			√	√						
Eb3	$\Sigma b3$	√	√					√	√				
Eb4	$\Sigma b4$	√	√							√	√		
Ec1	$\Sigma c1$	√			√	√							
Ec2	$\Sigma c2$	√					√	√					
Ec3	$\Sigma c3$	√							√	√			
Ed1	Σd			√	√							√	√
Ed2	Σd			√	√							√	√

6 - obtenção da linguagem das plantas locais:

Especificação	Sub-alfabeto	Planta local	
Ea	$\Sigma a = \{\alpha 0,\beta 1,\beta 2,\beta 3\}$	Gla = G0 || G1 || G2 || G3	16 estados 64 transições
Eb1	$\Sigma b1 = \{\alpha 0,\beta 0,\alpha 1,\beta 1\}$	Glb1 = G0 || G1	4 estados 8 transições
Eb2	$\Sigma b2 = \{\alpha 0,\beta 0,\alpha 2,\beta 2\}$	Glb2 = G0 || G2	4 estados 8 transições
Eb3	$\Sigma b3 = \{\alpha 0,\beta 0,\alpha 3,\beta 3\}$	Glb3 = G0 || G3	4 estados 8 transições
Eb4	$\Sigma b4 = \{\alpha 0,\beta 0,\alpha 4,\beta 4\}$	Glb4 = G0 || G4	4 estados 8 transições
Ec1	$\Sigma c1 = \{\alpha 0,\beta 1,\alpha 2\}$	Glc1 = G0 || G1 || G2	8 estados 24 transições
Ec2	$\Sigma c2 = \{\alpha 0,\beta 2,\alpha 3\}$	Glc2 = G0 || G2 || G3	8 estados 24 transições
Ec3	$\Sigma c3 = \{\alpha 0,\beta 3,\alpha 4\}$	Glc3 = G0 || G3 || G4	8 estados 24 transições
Ed	$\Sigma d = \{\alpha 1,\beta 1\alpha 5,\beta 5\}$	Gld = G1 || G5	4 estados 8 transições
obs:		Ed =Ed1 || Ed2	3 estados 4 transições

7 - determinação da linguagem das especificações locais :

Especifi-cação	Sub-alfabeto	Especificação local	
Ea	$\Sigma a = \{\alpha 0,\beta 1,\beta 2,\beta 3\}$	Ela = Gla || Ea	32 estados 120 transições
Eb1	$\Sigma b1 = \{\alpha 0,\beta 0,\alpha 1,\beta 1\}$	Elb1 = Glb1 || Eb1	3 estados 4 transições
Eb2	$\Sigma b2 = \{\alpha 0,\beta 0,\alpha 2,\beta 2\}$	Elb2 = Glb2 || Eb2	3 estados 4 transições
Eb3	$\Sigma b3 = \{\alpha 0,\beta 0,\alpha 3,\beta 3\}$	Elb3 = Glb3 || Eb3	3 estados 4 transições
Eb4	$\Sigma b4 = \{\alpha 0,\beta 0,\alpha 4,\beta 4\}$	Elb4 = Glb4 || Eb4	3 estados 4 transições
Ec1	$\Sigma c1 = \{\alpha 0,\beta 1,\alpha 2\}$	Elc1 = Glc1 || Ec1	32 estados 72 transições
Ec2	$\Sigma c2 = \{\alpha 0,\beta 2,\alpha 3\}$	Elc2 = Glc2 || Ec2	32 estados 72 transições
Ec3	$\Sigma c3 = \{\alpha 0,\beta 3,\alpha 4\}$	Elc3 = Glc3 || Ec3	32 estados 72 transições
Ed	$\Sigma d = \{\alpha 1,\beta 1\alpha 5,\beta 5\}$	Eld = Gld || Ed	4 estados 4 transições

8 - calculo da máxima linguagem controlável:

Especifi-cação	Sub-alfabeto	Supervisor local	
Ea	$\Sigma a = \{\alpha 0,\beta 1,\beta 2,\beta 3\}$	Sla = supC (Gla , Ela)	* → componente trim e minimizada do supervisor 32 estados 120 transições
Eb1	$\Sigma b1 = \{\alpha 0,\beta 0,\alpha 1,\beta 1\}$	Slb1 = supC (Glb1, Elb1)	3 estados 4 transições
Eb2	$\Sigma b2 = \{\alpha 0,\beta 0,\alpha 2,\beta 2\}$	Slb2 = supC (Glb2, Elb2)	3 estados 4 transições
Eb3	$\Sigma b3 = \{\alpha 0,\beta 0,\alpha 3,\beta 3\}$	Slb3 = supC (Glb3, Elb3)	3 estados 4 transições
Eb4	$\Sigma b4 = \{\alpha 0,\beta 0,\alpha 4,\beta 4\}$	Slb4 = supC (Glb4, Elb4)	3 estados 4 transições
Ec1	$\Sigma c1 = \{\alpha 0,\beta 1,\alpha 2\}$	Slc1 = supC (Glc1, Elc1)	24 estados 52 transições *
Ec2	$\Sigma c2 = \{\alpha 0,\beta 2,\alpha 3\}$	Slc2 = supC (Glc2, Elc2)	24 estados 52 transições *
Ec3	$\Sigma c3 = \{\alpha 0,\beta 3,\alpha 4\}$	Slc3 = supC (Glc3, Elc3)	24 estados 52 transições *
Ed	$\Sigma d = \{\alpha 1,\beta 1\alpha 5,\beta 5\}$	Sld = supC (Gld, Eld)	4 estados 4 transições
	somatório de estados para	G0 a G4	116 estados 288 transições
		G0 a G5	120 estados 292 transições

9 - verificação da modularidade das linguagens dos supervisores:

(Sla || Slb1 || Slb2 || Slb3 || Slb4 || Slc1 || Slc2 || Slc3 || Sld) =

= Trim (Sla || Slb1 || Slb2 || Slb3 || Slb4 || Slc1 || Slc2 || Slc3 || Sld)

a composição síncrona dos supervisores locais é trim, desta forma fica assegurada a condição de modularidade local, além disto pode-se afirmar que a solução obtida é ótima.

10 – resolução do problema da não modularidade das linguagens dos supervisores;

não se aplica ao caso em questão.

11 - minimização dos supervisores locais através do algorítimo de Vaz e Wonham

Supervisor local		
	Obtido através do CTCT * $\rightarrow$ componente trim e minimizada do supervisor	Minimizado por alg. Vaz e Wonham já eliminados os "self-loop"
Sla	32 estados 120 transições	2 estados 4 transições
Slb1	3 estados 4 transições	2 estados 4 transições
Slb2	3 estados 4 transições	2 estados 4 transições
Slb3	3 estados 4 transições	2 estados 4 transições
Slb4	3 estados 4 transições	2 estados 4 transições
Slc1	24 estados 52 transições *	4 estados 5 transições
Slc2	24 estados 52 transições *	4 estados 5 transições
Slc3	24 estados 52 transições *	4 estados 5 transições
Sld	4 estados 4 transições	2 estados 2 transições
M0 a M4	116 estados 288 transições	22 estados 35 transições
M0 a M5	120 estados 292 transições	24 estados 37 transições

Procedimento de obtenção do supervisor monolítico utilizando a teoria de controle supervisório original

obs:

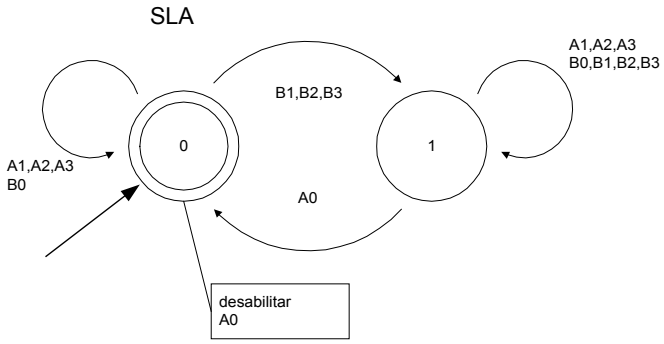
nas duas colunas da esquerda é apresentado o desenvolvimento do problema considerando o sistema constituído dos módulos M0 a M4 e nas duas colunas da direita o desenvolvimento do problema para o sistema constituído dos módulos M0 a M5

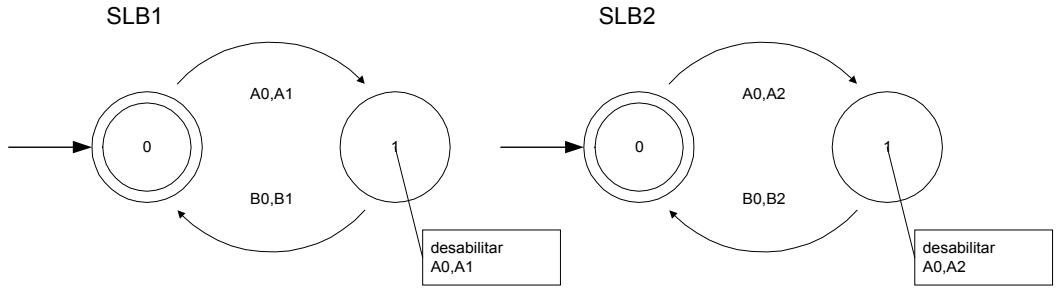
Planta global			
$G_g = \|\|_{i=0,...,4} G_i$	32 estados e 160 transições	$G_g = \|\|_{i=0,...,5} G_i$	64 estados e 384 transições
Especificação global (comp. trim)			
$E_g = \|\|_{x \in \{a, b1, b2, b3, b4, c1, c2, c3\}} E_x$	296 estados e 745 transições	$E_g = \|\|_{x \in \{a, b1, b2, b3, b4, c1, c2, c3,d\}} E_x$	755 estados e 2052 transições
Especificação global (comp. trim)			
$K_g = E_g \,\|\|\, L_m(G_g)$	151 estados e 350 transições	$K_g = E_g \,\|\|\, L_m(G_g)$	357 estados e 904transições
Supervisor			
S_{mon} =supC (K_g,$L_m(G_g)$)	151 estados e 350 transições	S_{mon} = supC (K_g,$L_m(G_g)$)	357 estados e 904 transições

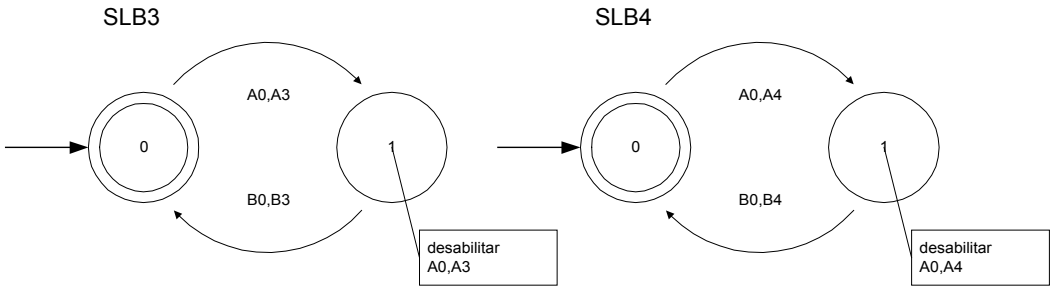
Pelo procedimento demonstrado nesta última etapa (obtenção do supervisor monolítico) verifica-se que a inclusão de mais um módulo com eventos assíncronos em relação aos anteriores duplica o tamanho da planta global, tendo influência similar sobre o número de estados do supervisor, fato que não é observado na obtenção dos supervisores locais, onde ocorreu um pequeno aumento do número total de estados.

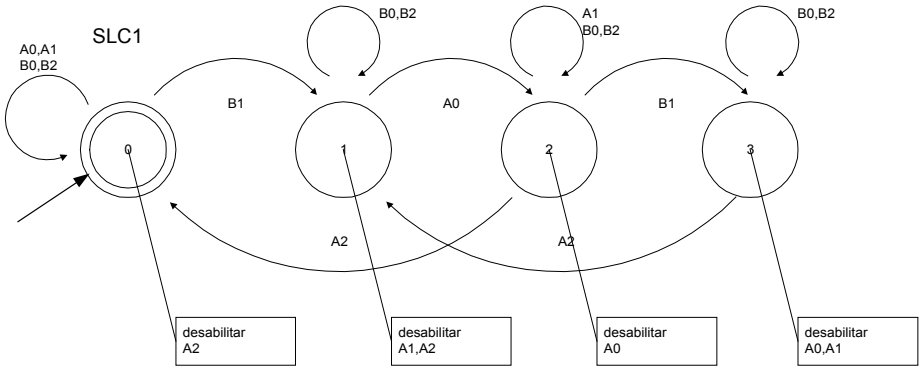
Além disto, em relação à implementação do programa em CLP a inclusão do módulo M5 não irá afetar de forma significativa o programa que tenha sido implementado para o controle do sistema constituído dos módulos M0 a M4. Caso ocorra a inclusão do módulo M5 o programa original será todo reutilizado com pequenas atualizações, fato que não seria observado com o supervisor monolítico, onde, em função do elevado número de estados do supervisor seria mais prático descartar todo o programa original.

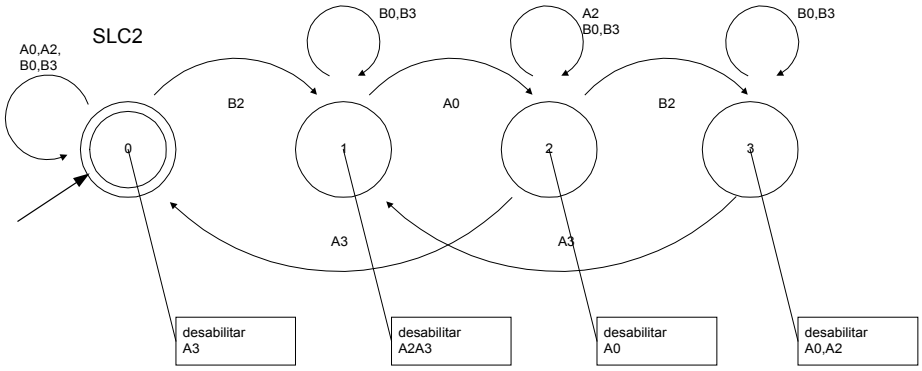
A seguir é apresentada a estrutura dos supervisores locais obtidos através do procedimento descrito anteriormente:

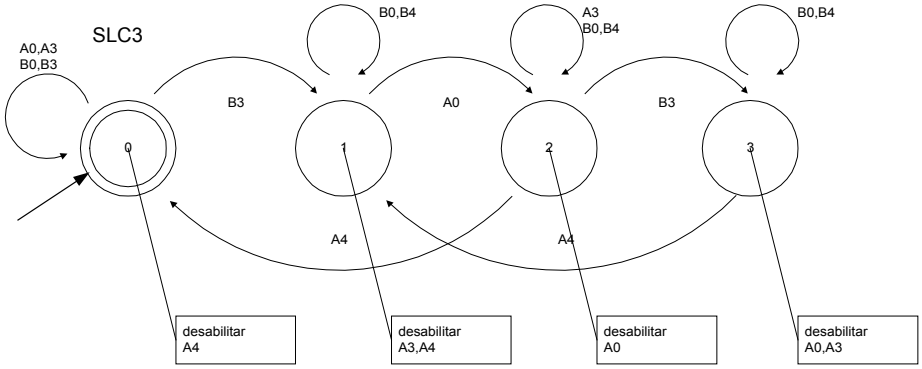


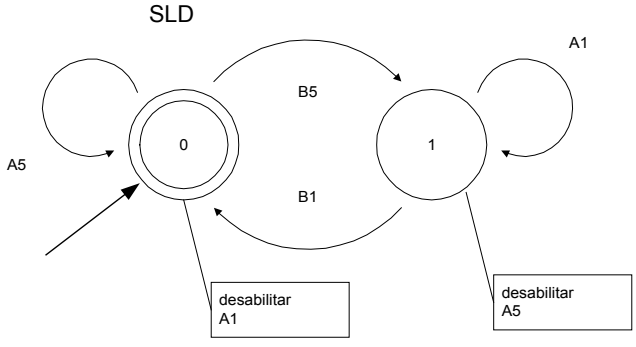





SLC1
A0,A1
B0,B2
B0,B2
A1
B0,B2
B0,B2
B1
A0
B1
0
1
2
3
A2
A2
desabilitar
A2
desabilitar
A1,A2
desabilitar
A0
desabilitar
A0,A1


SLC2
A0,A2,
B0,B3
B0,B3
A2
B0,B3
B0,B3
B2
A0
B2
0
1
2
3
A3
A3
desabilitar
A3
desabilitar
A2A3
desabilitar
A0
desabilitar
A0,A2


SLC3
A0,A3
B0,B3
B0,B4
A3
B0,B4
B0,B4
B3
A0
B3
0
1
2
3
A4
A4
desabilitar
A4
desabilitar
A3,A4
desabilitar
A0
desabilitar
A0,A3


SLD
A1
B5
A5
0
1
B1
desabilitar
A1
desabilitar
A5

Estrutura de implementação:

A estrutura de controle do sistema real; constituída dos supervisores locais, do sistema produto e das seqüências operacionais é implementada em uma estrutura hierárquica de três níveis, conforme apresentada na figura a seguir.

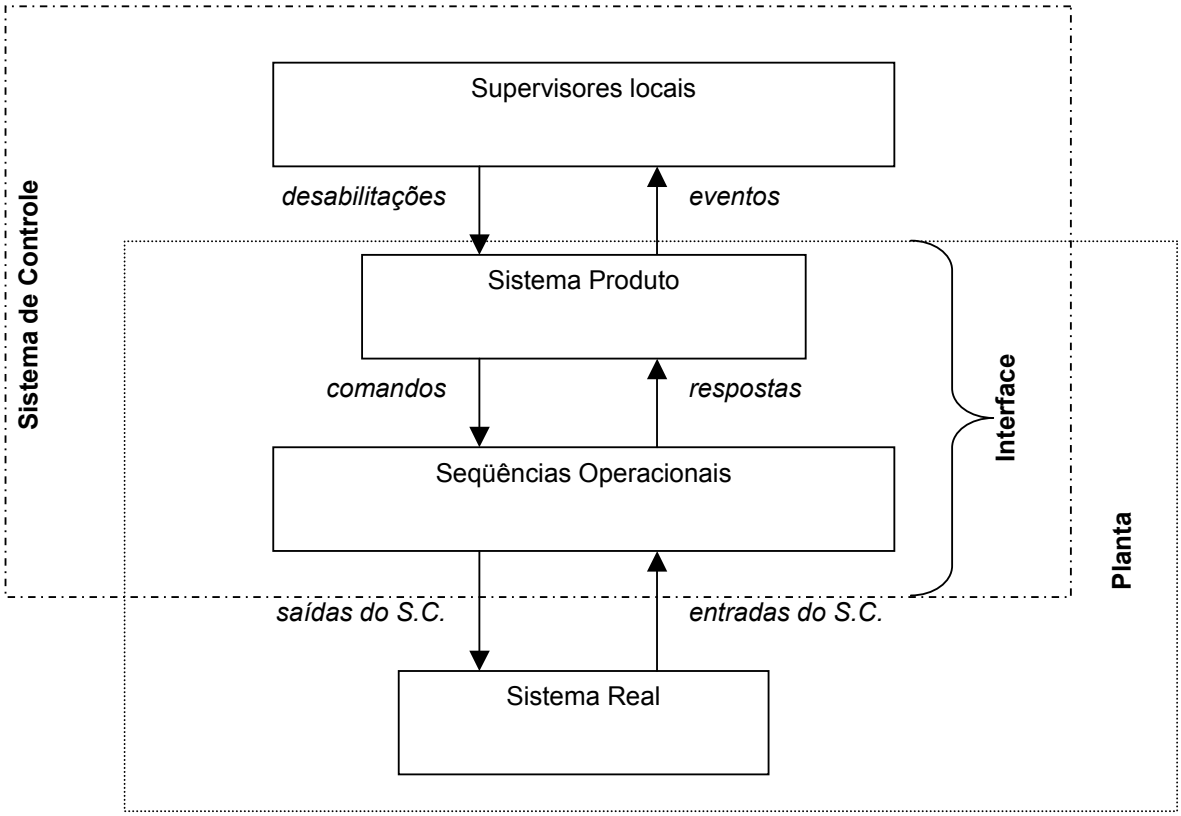


fig. 03 - Estrutura de controle do sistema real

O conjunto de supervisores locais (reduzidos) é implementado no nível mais alto da estrutura de controle. A evolução de estados dos supervisores locais é realizada através de eventos gerados nas subplantas do sistema produto (α_i – início de operação e β_i – final de operação da subplanta). Um mapa de retroalimentação associa os estados ativos destes supervisores a um conjunto de sinais de desabilitação dos eventos controláveis (αD_i) que controlam a evolução das subplantas do sistema produto.

Na teoria de controle supervisório é assumido que o sistema físico apresenta um comportamento expontâneo, gerando por si próprio eventos que determinam sua evolução. Como isto não é observado nos sistemas de automação industrial, é necessário realizar a implementação do sistema produto

no programa, o qual representa o sistema físico a controlar na sua forma mais abstrata.

A evolução de estado das subplantas do sistema produto é determinada pelo conjunto de sinais de desabilitação dos eventos controláveis (αD_i) e por sinais oriundos das seqüências operacionais (SOi_FIM – finalização das atividades realizadas pela subplanta e implementadas na seqüência operacional). Conforme mencionado em parágrafo anterior, nas subplantas do sistema produto serão gerados os eventos responsáveis pela evolução dos supervisores locais (α_i e β_i), bem como os sinais necessários para o controle das respectivas seqüências operacionais (SOi0 – ativação do passo inicial das seqüência operacional e SOi_último – desativação do último passo da seqüência operacional).

Tendo em vista que a teoria de controle supervisório não prevê a ocorrência simultânea de eventos, é fundamental que o estado dos supervisores locais esteja sempre atualizado antes que ocorra uma nova evolução no sistema produto, isto é, se ocorrer um evento α_i ou β_i (gerado durante a evolução de estado de uma subplanta do sistema produto) é necessário que ocorra a correspondente atualização dos estados dos supervisores locais e do subconjunto dos eventos controláveis desabilitados αD_i antes que seja avaliada a possibilidade de ocorrer a evolução de estado das demais subplantas. Caso isto não seja observado, há a possibilidade de ocorrer uma evolução de estado indevida de uma ou mais subplantas, resultando na violação da lei de controle sintetizada bem como de uma ou mais especificações de funcionamento. Isto é alcançado com a ativação, no instante em que ocorre a evolução de estado de uma subplanta, de uma variável que sinaliza a evolução da planta (ev_planta) impedindo assim a avaliação da possibilidade de evolução de estado das demais subplantas até que ocorra a atualização dos supervisores locais. Outra possibilidade, também avaliada com sucesso, é o desvio da execução do programa, no instante em que ocorre a evolução de estado de uma subplanta, para um ponto pré-estabelecido do mesmo. Contudo esta última possibilidade não será discutida com maiores detalhes nesta versão do procedimento.

No item 2 do procedimento para síntese dos supervisores locais foi proposta uma abstração estrutural e funcional dos módulos do sistema físico a controlar, abstração esta necessária para viabilizar o referido procedimento. Através da implementação das seqüências operacionais é realizada a compatibilização dos resultados obtidos acima com a solução tecnológica utilizada para realização do sistema físico. Em função da abstração realizada durante a modelagem do sistema e da proposta de implementação das seqüências operacionais, a substituição de um módulo do sistema por outro com características estruturais e funcionais distintas, não exige nova síntese dos supervisores locais, além disto, grande parte do programa será mantida inalterada, sendo necessário atualizar apenas a parte referente à seqüência operacional do referido módulo.

Procedimento para implementação em diagrama escada:

Nesta seção é apresentado o procedimento para implementação em diagrama escada da estrutura de controle conforme esquematizado na figura 03.

- supervisores locais:

Para cada estado dos supervisores locais deverá ser implementada uma estrutura que, na ocorrência dos eventos α_i ou β_i, ativa o estado do supervisor e desativa os estados anteriores do mesmo, esta estrutura é apresentada de forma genérica na figura 04.

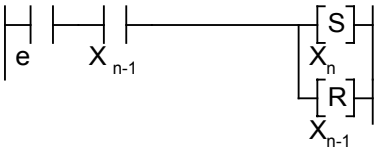


fig. 04 - Estrutura genérica para implementação dos supervisores locais

onde:

e - representa os eventos α_i ou β_i,

X_n - representa o estado do supervisor que está sendo implementado

X_{n-1} – representa o estado do supervisor anterior ao estado que está sendo implementado

Nesta estrutura cada uma das variáveis listadas deverá ser associada a uma variável interna do CLP. Justifica-se assim o comentário realizado quando da comparação dos resultados obtidos entre o controle modular local e o controle monolítico. Dentre os principais parâmetros que determinam a capacidade de um CLP estão o número de entradas e saídas de sinal e o número de variáveis internas, neste sentido o número de estados dos supervisores é determinante na seleção do modelo de CLP a ser utilizado para implementação do controle.

Na figura 05 é realizada a exemplificação da adaptação necessária à estrutura apresentada na figura anterior caso um determinado estado do supervisor possa ser alcançado a partir de mais de um estado preliminar.

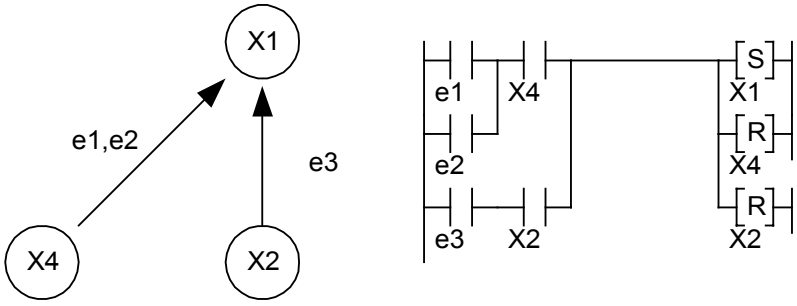


fig. 05 - exemplo de implementação de estado de supervisor

Destaca-se ainda que, em função da forma como é realizada a implementação dos supervisores, não é necessário realizar a inclusão no programa dos eventos que saem e entram no mesmo estado (auto-arcos), visto que, a ocorrência destes eventos não provoca a alteração de estado do supervisor.

- estrutura de desabilitações:

A função do conjunto de supervisores locais é a de desabilitar um subconjunto de eventos controláveis em função da evolução do sistema físico, neste sentido deve ser implementado uma estrutura que ativa uma variável que corresponde a desabilitação de um dado evento controlável caso os supervisores estejam em um determinado subconjunto de estados.

fig. 06 - estrutura genérica para desabilitação de eventos controláveis

onde:

Xj Xk Xm - representa um subconjunto genérico de estados dos supervisores locais nos quais o evento controlável α_i deve ser desabilitado

αDi - representa a desabilitação do evento α_i

Nesta estrutura as variáveis αDi deverão ser relacionadas a variáveis internas do CLP.

- sistema produto:

Na etapa 2 do procedimento para obtenção dos supervisores locais foi proposta a abstração das características estruturais e funcionais dos módulos do sistema físico a controlar, a figura 07 corresponde à representação através de um gerador de uma subplanta genérica do sistema produto obtido na etapa 3 do procedimento.

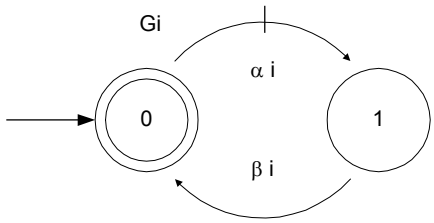


fig. 07 - estrutura de subplanta genérica "Gi" do sistema físico utilizada na fase de obtenção dos supervisores modulares

Para implementação do sistema produto no programa do CLP e para obter a sincronização deste com os supervisores locais e as seqüências operacionais cada uma das subplantas do sistema produto é representada através da estrutura apresentada na figura 08.

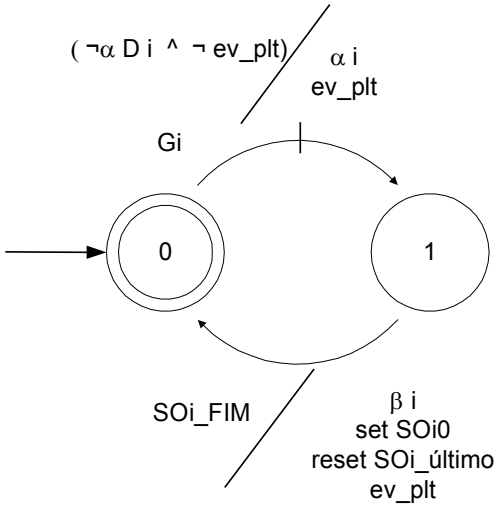


fig. 08 - estrutura da subplanta genérica "Gi" para implementação do sistema produto

onde:

α_i - representa o início de operação da subplanta "Gi"

β_i - representa o final de operação da subplanta "Gi"

αD_i - representa a desabilitação do evento α_i

$\neg\alpha D_i$ - negação da desabilitação do evento α_i

SO_i_FIM - representa a finalização das atividades realizadas pela subplanta "Gi" implementadas na seqüência operacional "SO_i"

set SO_i0 - representa a ativação do passo inicial da seqüência operacional "SO_i"

reset SO_i_último - representa a desativação do último passo da seqüência operacional "SO_i"

ev_plt – sinaliza a evolução da planta

¬ev_plt – negação da evolução da planta

Cada uma das variáveis acima deve ser relacionada a uma variável interna do CLP.

Para uma melhor compreensão da estrutura utilizada para implementação das subplantas "Gi" no sistema produto e da sua interação com os supervisores locais e as seqüências operacionais, é apresentada a seguir uma descrição da evolução do programa implementado no CLP.

Na condição inicial do programa todas as subplantas "Gi" implementadas no sistema produto estarão no estado "0". Como condição inicial de funcionamento da planta deverá haver pelo menos um evento α_i que não esteja desabilitado por nenhum dos supervisores locais, digamos que (para i = 3) α_3 não está desabilitado, desta forma a variável αD_3 estará em “nível lógico 0” (αD_3=nl0). Quando houver a avaliação da estrutura de transição do estado 0 para o estado 1 da subplanta G3 as condições de ativação deste estado estarão satisfeitas [G3 no estado 0 (G30=nl1) e evento α_3 não está desabilitado ($\neg\alpha D_3$=nl1) e não houve evolução da planta ($\neg$ev_plt=nl1)], resultando assim na evolução da subplanta G3 do estado 0 para o estado 1.

Durante esta evolução de estado as variáveis α_3 e ev_plt serão ativadas (colocadas em nível lógico 1) além disto a variável G31 será ativada com retenção e a variável G30 será desativada com retenção. Devido à ativação da variável ev_plt, as estruturas de transição de estado implementadas após a estrutura de transição do estado 0 para o estado 1 da supblanta G3 não terão suas condições de ativação satisfeitas, pois em todas elas será observado que houve evolução da planta ($\neg$ev_plt=nl0). Desta forma mesmo que as demais condições para transição de estado das subplantas sejam satisfeitas, estas transições não ocorrerão durante este ciclo de varredura do programa.

A utilização desta solução produz efeito similar ao da utilização de comandos apropriados para execução de “salto na execução do programa”.

Ainda durante este ciclo de varredura do programa (aqui denominado ciclo c1) serão processados os comandos implementados na estrutura referentes às

seqüências operacionais. Como houve a ativação com retenção da variável G31 terá início a seqüência operacional referente à subplanta G3. Ao final deste ciclo de varredura ocorre portanto o envio de comandos do CLP para o sistema físico dando início às atividades realizadas pelo módulo 3.

No próximo ciclo de varredura do programa (ciclo c2) será realizada a atualização dos estados dos supervisores causada pela variação ocorrida com a variável α_3, e estabelecido um novo subconjunto de eventos desabilitados, representado pela ativação das variáveis αD_i correspondentes.

Como no ciclo de varredura anterior (ciclo c1) houve a ativação da variável ev_plt, (ev_plt=nl1) esta ainda permanecerá ativa durante o ciclo c2, fazendo com que neste ciclo não ocorra a transição de estado de nenhuma subplanta, e portanto não ocorra nova ativação desta variável, desta forma no ciclo de varredura seguinte (ciclo c3) esta variável estará desativada (ev_plt=nl0). Somente no ciclo c3 haverá a possibilidade de ocorrer nova transição de estado de alguma subplanta.

Verifica-se que, com esta implementação não há a possibilidade de ocorrer no mesmo ciclo de varredura do programa a ativação de duas ou mais variáveis do tipo α_i ou β_i, ou seja, não há a ocorrência de eventos simultâneos no sistema. Há porém a existência de um ciclo de varredura do programa em que não pode ocorrer a evolução da planta. Para eliminar este problema pode ser implementada, após todos os comandos referentes ao sistema produto, uma linha de programa que desativa a variável ev_plt.

Após a conclusão da última atividade física realizada pela seqüência operacional será ativada a variável SOi_FIM, sinalizando a finalização das atividades realizadas pela respectiva subplanta Gi, permanecendo ativa até que a variável do passo correspondente da seqüência operacional (SOi_último) seja desativada. A ativação desta variável irá permitir que a subplanta Gi evolua do estado 1 para o estado 0. Durante esta evolução a variável relacionada ao evento β_i será ativada, além disto, a variável relacionada ao passo inicial da seqüência operacional (SOi0) será ativada com retenção e a variável relacionada ao último passo da seqüência operacional (SOi_último) será desativada com retenção (condições obtidas na inicialização do programa). Será ativada também a variável ev_plt permitindo o controle adequado para que não ocorra a transição de estado de mais de uma subplanta no mesmo ciclo de varredura, evitando assim a ocorrência de eventos simultâneos conforme descrito anteriormente.

Vamos supor agora que em um determinado ciclo de varredura do programa (aqui denominado ciclo 11) tenha havido a finalização de mais de uma seqüência operacional, ativando assim as variáveis SO4_FIM e SO5_FIM, e além disto, que a seqüência operacional referente ao módulo 5 tenha sido concluída antes que a seqüência operacional referente ao módulo 4. É possível ainda, que no ciclo de varredura anterior tenha ocorrido a evolução da planta fazendo com que o evento α_2 não esteja mais desabilitado. Têm-se assim, simultaneamente, as condições para que a subplanta G2 evolua do estado 0 para o estado 1, que a

subplanta G4 evolua do estado 1 para o estado 0 e que a subplanta G5 evolua do estado 1 para o estado 0.

Analisaremos a seguir como deverá ser realizada a evolução da planta.

Ao ser realizada a varredura do programa no ciclo c12 a primeira estrutura de transição de estado que tiver suas condições satisfeitas será realizada, ativando assim as variáveis correspondentes, bem como a variável ev_plt, fazendo com que neste ciclo não ocorra a transição de estado de nenhuma outra subplanta, serão ainda atualizados adequadamente os comandos referentes às seqüências operacionais. Supondo que ao final deste ciclo tenha sido realizada a desativação da variável ev_plt, no ciclo seguinte (c13) após realizar a adequada atualização de estado dos supervisores locais e da estrutura de desabilitações serão avaliadas as condições para transição de estado das subplantas, novamente, a primeira estrutura de transição de estado que tiver suas condições satisfeitas será realizada, ativando a variável ev_plt e impedindo as demais transições de estado. Este processo se repete continuamente.

Pelo exposto no parágrafo anterior, verifica-se que é estabelecida uma prioridade na seqüência em que ocorrem as transições de estado das subplantas em função da ordem em que estas estruturas foram implementadas no programa. Assim, é conveniente implementar inicialmente as estruturas de transição de estado devidas a ocorrência de eventos não controláveis, priorizando a atualização da planta pela ocorrência de eventos não controláveis em detrimento dos eventos controláveis.

Retornando ao caso proposto anteriormente e supondo que as estruturas de transição de estado das subplantas tenham sido implementadas no programa na seguinte seqüência: inicialmente as estruturas de transição de estado devidas a eventos não controláveis e após estas as estruturas de transição de estado devidas a eventos controláveis, ambas implementadas em ordem crescente de numeração da subplanta,será observada a seguinte evolução da planta.

Como a estrutura de transição do estado 1 para o estado 0 da subplanta G4 com a variável SO4_FIM está implementada antes que as estruturas de transição de estado da subplanta G5 (estado 1 para estado 0) com a variável SO5_FIM e da subplanta G2 (estado 0 para estado 1) com a variável αD_2, esta será a transição que ocorrerá no ciclo c12, havendo assim a ativação, dentre outras, das variáveis β_4 e ev_plt. Como explicado anteriormente, as demais estruturas de transição de estado das subplantas não serão efetuadas no ciclo de varredura do programa c12. Ao final deste ciclo será desativada a variável ev_plt.

No ciclo c13 ocorrerá a devida atualização de estado dos supervisores locais, devido à ocorrência do evento β_4, e a atualização da estrutura de desabilitação de eventos controláveis. Ainda neste ciclo ocorrerá a transição do estado 1 para o estado 0 da subplanta G5 com a variável SO5_FIM, havendo agora a ativação das variáveis β_5 e ev_plt, não ocorrendo neste ciclo de varredura do programa nenhuma outra transição de estado das subplantas. Novamente, ao final deste ciclo será realizada a desativação da variável ev_plt.

No ciclo c14 ocorrerá a devida atualização de estado dos supervisores locais, devido à ocorrência do evento β_5, e a atualização da estrutura de desabilitação de eventos controláveis. É possível que neste ciclo, em função da evolução da planta ocorrida nos ciclos c13 e c14 a nova estrutura de desabilitações de eventos defina um subconjunto de eventos controláveis desabilitados distinto ao observado no ciclo c12, havendo inclusive a possibilidade de que o evento α_2, habilitado no ciclo c12, esteja agora desabilitado, não ocorrendo assim a transição do estado 0 para o estado 1 da subplanta G2 prevista no ciclo c12.

Verifica-se também que, apesar da seqüência operacional referente ao módulo 5 ter sido concluída antes que a seqüência operacional referente ao módulo 4, em função da prioridade de transição de estados das subplantas, definida pela seqüência em que estas foram implementadas no programa, houve a transição do estado 1 para o estado 0 da subplanta G4 antes que a transição do estado 1 para o estado 0 da subplanta G5.

Para implementação em diagrama escada do gerador apresentado na figura 08 é utilizada a estrutura apresentada na figura 09, e na figura 10 é apresentada a estrutura para desativação da variável ev_plt a qual, conforme mencionado anteriormente, deve ser implementada após a última estrutura de ativação de estado das subplantas do sistema produto

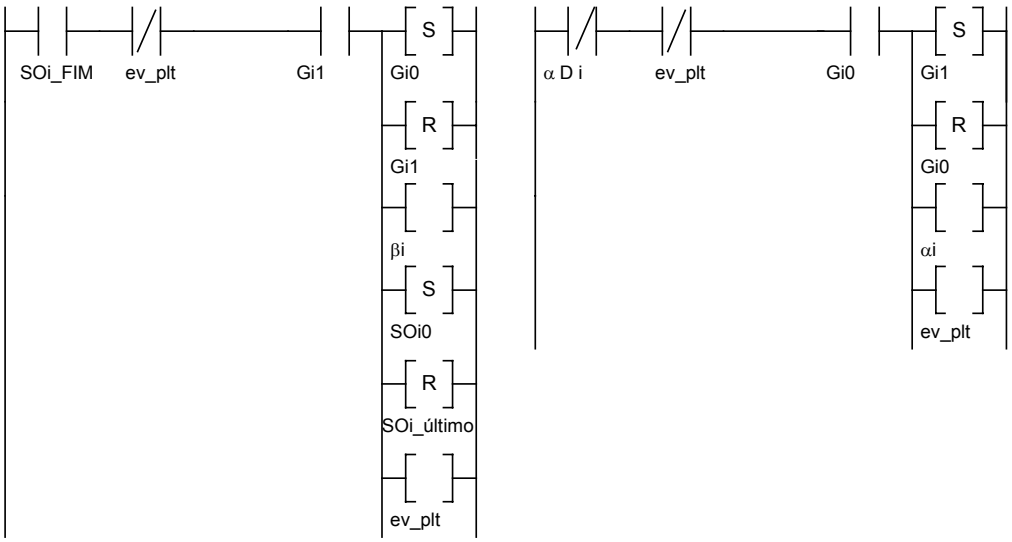


fig. 09 - estrutura genérica para implementação das subplantas “Gi” do sistema produto

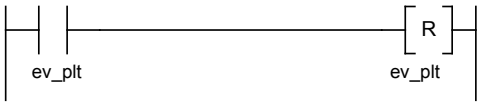


fig.10 – estrutura para desativação da variável ev_plt

- seqüências operacionais:

O conjunto de atividades a serem executadas pelas subplantas "Gi" e até o momento abstraídas da implementação do controle do sistema a eventos discretos é representado através de um diagrama Grafcet, cuja estrutura genérica é apresentada na figura 11.

Conforme apresentado na descrição da interação entre o sistema produto e os supervisores locais e seqüências operacionais, a condição de prosseguimento que ativa o passo SOi1 é a ativação da variável relacionada ao estado 1 do gerador utilizado para representar a subplanta "Gi" no sistema produto (Gi1). Observa-se também que a ação do último passo da seqüência é a ativação da variável que indica a finalização das atividades realizadas pela subplanta "Gi".

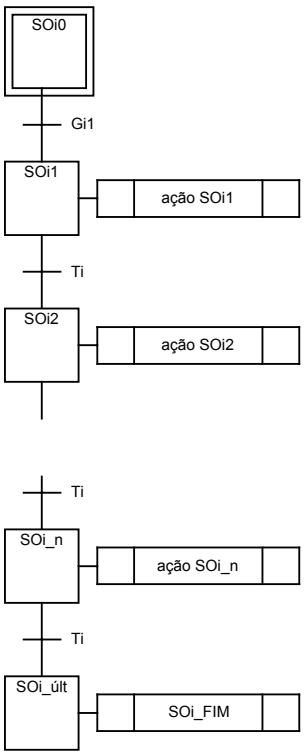


fig. 11 - estrutura genérica de diagrama Grafcet

para implementação da seqüencia operacional

Para sincronização de operação da seqüência operacional, representada através do diagrama Grafcet (fig. 11) , e a subplanta "Gi" (fig. 08 e 09) do sistema produto, é necessário que a ativação da variável relacionada ao primeiro passo da seqüência (SOi0) seja ativada durante a evolução do estado 1 para o estado 0 do gerador utilizado na representação da subplanta "Gi", ainda durante esta evolução

de estado deve ser desativada a variável relacionada com o último passo da seqüência (SOi_último).

Para tanto deve ser implementada em diagrama escada a estrutura apresentada na figura 12, na qual verifica-se que:

- não é necessário realizar a implementação do passo 0, pois a mesma é realizada junto a implementação do sistema produto;

- na estrutura relacionada ao último passo não é implementada a desativação da variável SOi_último, pois a mesma é realizada junto a implementação do sistema produto;

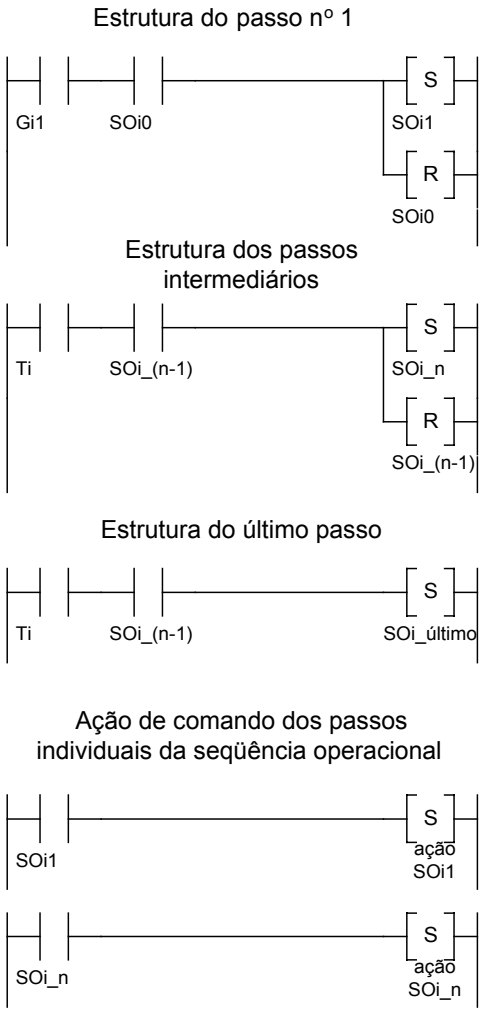
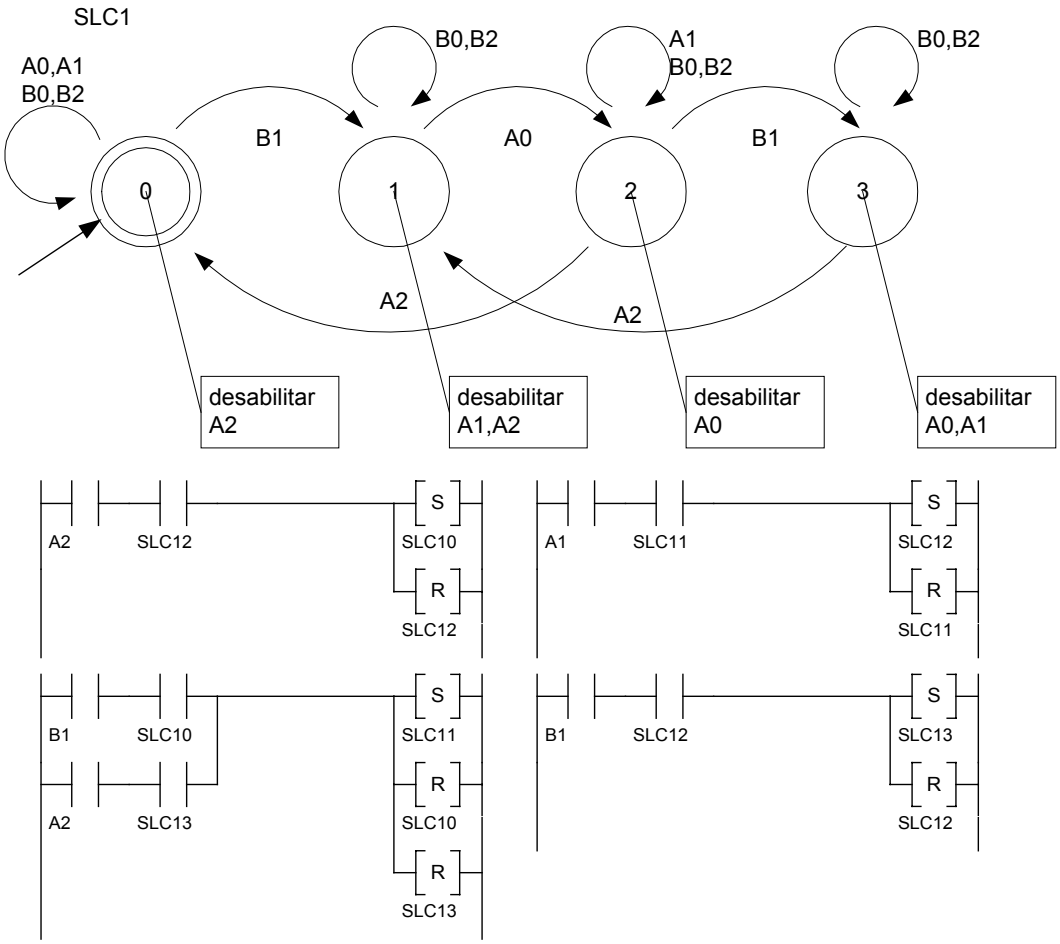


fig. 12 - estrutura em diagrama escada para implementação da seqüência operacional

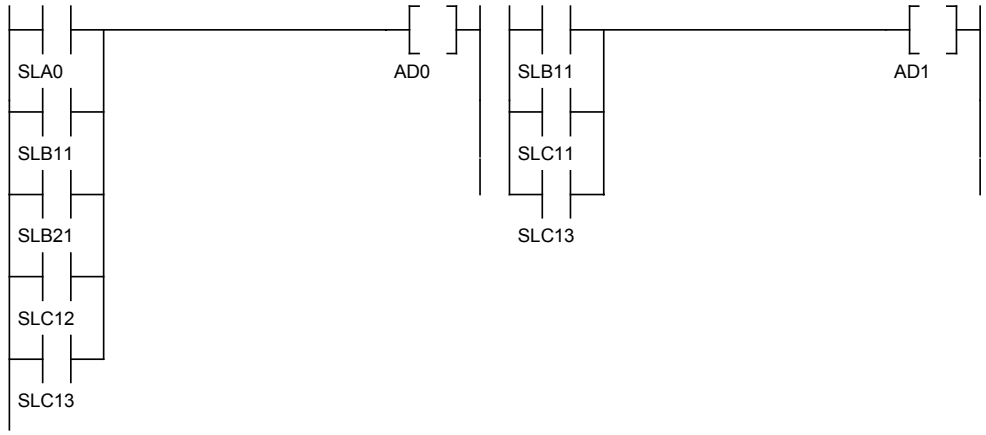
Exemplo parcial de aplicação da implementação do programa em diagrama escada:

-implementação do supervisor modular local SLC1

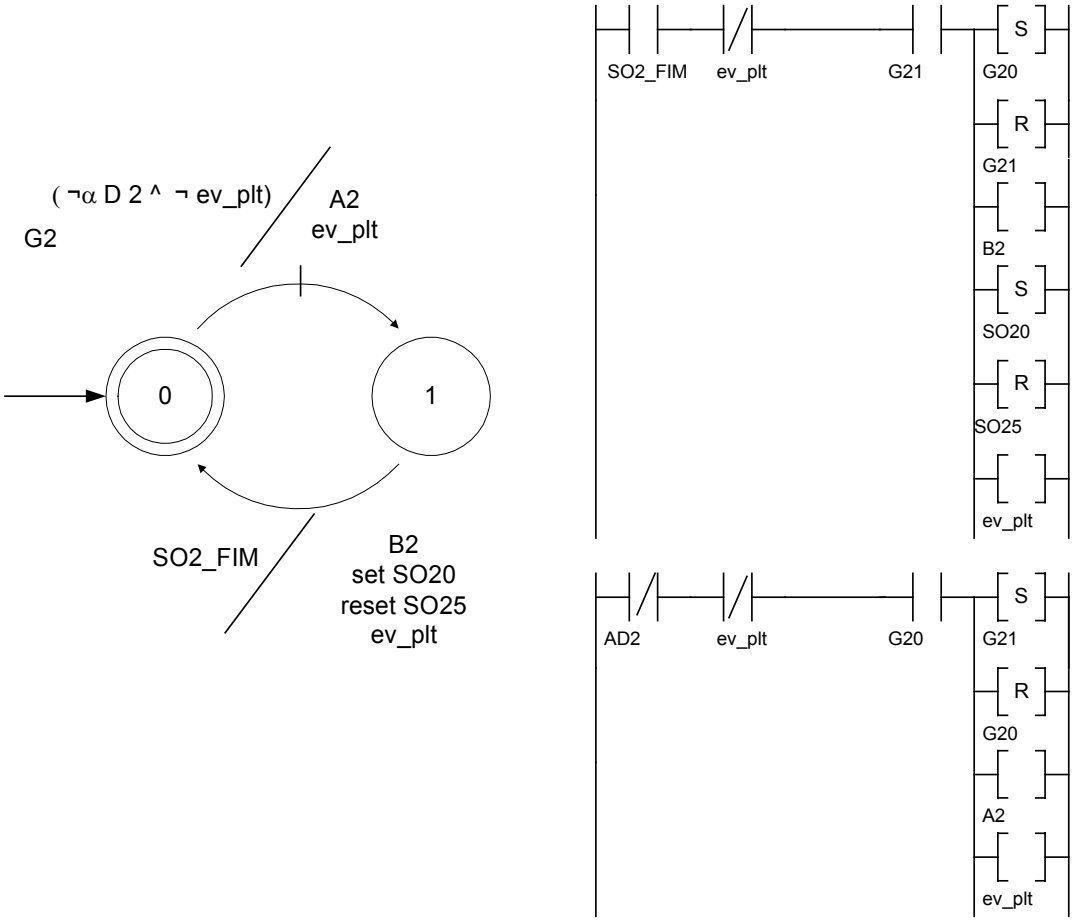


como pode ser observado no diagrama escada acima, não são implementadas as transições que saem e entram no mesmo estado (auto-arcos).

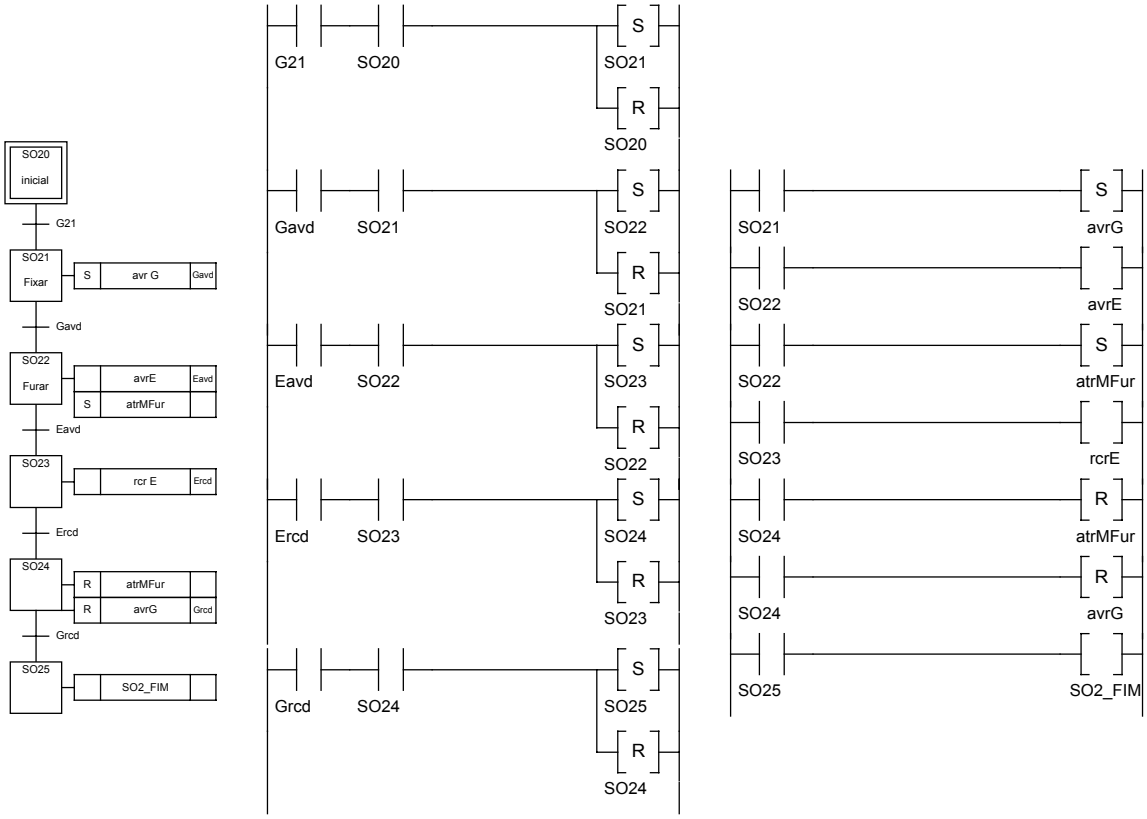
- implementação parcial da estrutura de desabilitações dos eventos $\alpha 0$ e $\alpha 1$:



- implementação da subplanta G2 no sistema produto



- implementação da seqüência operacional SO2:



- inicialização de variáveis do programa:

É necessário ainda realizar a inicialização de variáveis internas e das saídas de sinal, colocando o programa na condição inicial, ou seja:

- ativar as variáveis utilizadas para representar o estado 0 dos geradores que representam os supervisores locais e desativar as variáveis relacionadas aos demais estados;

- ativar as variáveis utilizadas para representar o estado 0 dos geradores que representam as subplantas "Gi" no sistema produto e desativar as demais variáveis;

- ativar as variáveis utilizadas para representar o passo 0 dos Grafcet que representam as seqüências operacionais;

- desativar todas as saídas de sinal;

Esta atividade pode ser realizada utilizando o comando de atribuição de valor a variável do tipo multi-bit.

Na tabela abaixo é apresentado o estado inicial em que algumas variáveis do programa implementado para controle do sistema de manipulação de peças descrito devem estar, bem como o valor decimal correspondente à palavra de 8 ou16 bits formada pela associação das variáveis individuais.

	Fi.0	Fi.1	Fi.2	Fi.3	Fi.4	Fi.5	Fi.6	Fi.7	Valor
FW0	SLA0	SLA1	SLB10	SLB11	SLB20	SLB21	SLB30	SLB31	17749
	1	0	1	0	1	0	1	0	
	SLB40	SLB41	SLC10	SLC11	SLC12	SLC13	SLC20	SLC21	
	1	0	1	0	0	0	1	0	
FW1	SLC22	SLC23	SLC30	SLC31	SLC32	SLC33	SLD0	SLD1	68
	0	0	1	0	0	0	1	0	
	0	0	0	0	0	0	0	0	
FW3	AD0	AD1	AD2	AD3	AD4	AD5			0
	0	0	0	0	0	0			
FW4	AA0	AA1	AA2	AA3	AA4	AA5			0
	0	0	0	0	0	0			
FW5	B0	B1	B2	B3	B4	B5			0
	0	0	0	0	0	0			
FW6	G00	G01	G10	G11	G20	G21	G30	G31	1365
	1	0	1	0	1	0	1	0	
	G40	G41	G50	G51					
	1	0	1	0					
FW8	SO20	SO21	SO22	SO23	SO24	SO25			1
	1	0	0	0	0	0			
								SO2_FIM	
								0	

	Oi.0	Oi.1	Oi.2	Oi.3	Oi.4	Oi.5	Oi. 6	Oi.7	Valor
OW0	avrA	dtrGV1	rcrB	avrB	avrB	atrGV1	rcrD	avrD	0
OW1	avrC	atrME	avrE	rcrE	atrMFur	avrG	atrMM		0

Conclusão:

Conforme observado durante as etapas de obtenção dos supervisores locais e implementação do programa em CLP, a metodologia proposta por Queiroz e Cury é uma solução bastante viável e prática para a implementação do controle de sistemas a eventos discretos, permitindo aos engenheiros, que possuam um conhecimento mínimo sobre a utilização dos pacotes computacionais para resolução de problemas utilizando a teoria de controle supervisório (CTCT ou GRAIL) bem como programação básica de CLP, a possibilidade de resolução sistemática de problemas que tradicionalmente são abordados e solucionados informalmente, muitas vezes pelo método da tentativa e erro.

Como principais características desta metodologia destam-se:

- abordagem sistemática para elaboração e implementação do sistema de controle;
- facilidade para adaptação do programa quando da inclusão ou remoção de módulos no sistema físico, com reutilização de grande parte do programa original;
- facilidade para adaptação do programa quando da alteração das características estruturais ou operacionais dos módulos do sistema físico, com reutilização de grande parte do programa original;
- implementação da solução minimamente restritiva;
- viabilidade de implementação levando em consideração os recursos tecnológicos disponíveis (capacidade de memória e processamento dos computadores pessoais utilizados no cálculo dos supervisores locais e capacidade de variáveis internas dos CLP's);
- facilidade para implementação em outros dispositivos de controle (microcomputador dotado de interface para entrada e saída de dados, circuitos eletrônicos, circuitos pneumáticos, ...).

Bibliografia:

Queiroz, M.H. (2000). Controle supervisório modular de sistemas de grande porte. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica, Florianópolis.

Queiroz, M.H. de, E.A.P. Santos e J.E.R. Cury (2001). Síntese modular do controle supervisório em diagrama escada para uma célula de manufatura. *Anais do V Simpósio Brasileiro de Automação Inteligente*, Gramado RS.

Queiroz, M.H. de (2001). Contribuições Teóricas e Práticas ao Controle Supervisório de Sistemas Compostos. Proposta de Tese, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica, Florianópolis.