

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
ESCOLA POLITÉCNICA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

***IMPLEMENTAÇÃO EM LADDER DE  
SISTEMAS DE AUTOMAÇÃO  
DESCRITOS POR REDES DE PETRI  
INTERPRETADAS PARA CONTROLE***

**RAFAEL PENTAGNA SILVESTRE**

Rio de Janeiro, RJ - Brasil

Janeiro de 2010

***IMPLEMENTAÇÃO EM LADDER DE  
SISTEMAS DE AUTOMAÇÃO DESCRITOS  
POR REDES DE PETRI INTERPRETADAS  
PARA CONTROLE***

**RAFAEL PENTAGNA SILVESTRE**

*PROJETO SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE  
ENGENHARIA ELÉTRICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE  
FEDERAL DO RIO DE JANEIRO, COMO PARTE DOS REQUISITOS NECES-  
SÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRICISTA.*

Aprovado por:

---

MARCOS VICENTE DE BRITO MOREIRA, D. Sc.  
(Orientador)

---

JOÃO CARLOS DOS SANTOS BASILIO, Ph.D.

---

LILIAN KAWAKAMI CARVALHO, M.Sc.

Rio de Janeiro, RJ - Brasil

Janeiro de 2010

# RESUMO

Rafael Pentagna Silvestre  
UFRJ - DEE

Projeto de Graduação  
Janeiro 2010

## **Implementação em Ladder de Sistemas de Automação Descritos por Redes de Petri Interpretadas para Controle**

A utilização de um sistema de controle automático é atualmente uma condição necessária para tornar qualquer processo industrial mais eficiente. A crescente complexidade desses processos gera a necessidade de novos meios de modelá-los e controlá-los. Nas últimas décadas foram desenvolvidas ferramentas matemáticas capazes de lidar de maneira mais estruturada com a lógica de controle de sistemas a eventos discretos (SEDs), como autômatos, redes de Petri e Grafcet. Neste projeto a estrutura utilizada é a rede de Petri, pois possui uma característica visual que facilita o estudo do comportamento dinâmico dos SEDs, tornando mais simples a detecção e solução de conflitos nos mesmos, e, por conseguinte, o controle dos SEDs. Entretanto, a implementação de sistemas de automação é em geral realizada utilizando-se controladores lógico programáveis (CLPs), cujas linguagens de programação normatizadas são: diagrama de blocos funcionais; lista de instruções; texto estruturado; diagrama Ladder; diagrama funcional sequencial, sendo o diagrama Ladder a mais utilizada. A proposta deste projeto é apresentar um método de conversão onde seja possível modelar a automação de um SED através de uma rede de Petri mais sofisticada, denominada rede de Petri interpretada para controle (RPIC), e em seguida, através de suas equações matemáticas e matrizes de incidência, realizar uma conversão para o próprio diagrama Ladder que será implementado no controlador lógico programável CLP, permitindo assim um considerável ganho de tempo, além de maior simplicidade na realização do controle do SED.

# Sumário

<b>RESUMO</b>	<b>ii</b>
<b>LISTA DE FIGURAS</b>	<b>vii</b>
<b>LISTA DE TABELAS</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Redes de Petri</b>	<b>5</b>
2.1 Conceitos Básicos . . . . .	5
2.1.1 Dinâmica da Rede de Petri . . . . .	6
2.1.1.1 Marcação . . . . .	6
2.1.1.2 Habilitação . . . . .	7
2.1.1.3 Disparo . . . . .	8
2.1.1.4 Equação de Estado e Matriz de Incidência . . . . .	10
2.2 Conflitos . . . . .	11
2.2.0.1 Conflito Estrutural . . . . .	11
2.2.0.2 Conflito Efetivo . . . . .	11
2.3 Estruturas Especiais . . . . .	12
2.3.1 Ordinária . . . . .	12
2.3.2 Segura . . . . .	12
2.3.3 Pura . . . . .	12
2.3.4 Generalizada . . . . .	12
2.4 Tipos de Arcos . . . . .	13
2.4.1 Arco Simples . . . . .	13

2.4.2	Arco Ponderado . . . . .	13
2.4.3	Arco Habilitador . . . . .	13
2.4.4	Arco Inibidor . . . . .	14
2.5	Resumo . . . . .	14
<b>3</b>	<b>Redes de Petri Não-Autônomas e Interpretadas para Controle</b>	<b>16</b>
3.1	Redes de Petri Não-Autônomas . . . . .	16
3.1.1	Redes de Petri Sincronizadas . . . . .	16
3.1.1.1	Marcação e Disparo . . . . .	18
3.1.1.2	Conflito Real . . . . .	19
3.1.2	Redes de Petri Temporizadas . . . . .	20
3.1.2.1	Redes de Petri P-Temporizadas . . . . .	21
3.1.2.2	Redes de Petri T-Temporizadas . . . . .	23
3.1.3	Redes de Petri Interpretadas . . . . .	24
3.1.3.1	Rede de Petri Interpretada para Controle de David e Alla . . . . .	25
3.2	Rede de Petri Interpretada para Controle . . . . .	27
3.3	Resumo . . . . .	30
<b>4</b>	<b>Controlador Lógico Programável e Diagrama Ladder</b>	<b>31</b>
4.1	Controlador Lógico Programável . . . . .	31
4.1.1	Operação . . . . .	31
4.2	Diagrama Ladder . . . . .	32
4.2.1	Instruções Básicas . . . . .	32
4.2.1.1	Corrente Lógica Fictícia . . . . .	33
4.2.2	Contato Normalmente Aberto . . . . .	33
4.2.3	Contato Normalmente Fechado . . . . .	34
4.2.4	Bobina . . . . .	35
4.2.5	SET (OTL - Output Latch) e RESET (OTU - Output Unlatch)	35
4.2.6	Disparo Único de Subida - OSR (One-Shot Rising) . . . . .	36
4.2.7	Instruções de Temporização . . . . .	36

4.2.8	Contadores . . . . .	37
4.2.8.1	Reset - RES . . . . .	37
4.3	Resumo . . . . .	37
<b>5</b>	<b>Método de Conversão de RPIC para Diagrama Ladder</b>	<b>39</b>
5.1	Método Usual Utilizado para a Conversão de Redes de Petri em Ladder	39
5.2	Problemas de Implementação . . . . .	40
5.2.1	Efeito Avalanche . . . . .	40
5.2.2	Resolução de Conflitos . . . . .	42
5.3	Método de Conversão . . . . .	44
5.3.1	Módulo de Inicialização . . . . .	44
5.3.2	Módulo de Eventos . . . . .	45
5.3.3	Módulo de Condições para o Disparo das Transições . . . . .	45
5.3.4	Módulo da Dinâmica da Rede de Petri . . . . .	47
5.3.5	Módulo das Ações . . . . .	47
5.3.6	Determinação do Tamanho do Código Ladder . . . . .	48
5.3.7	Eliminação do Efeito Avalanche . . . . .	49
5.4	Resumo . . . . .	49
<b>6</b>	<b>Exemplo</b>	<b>51</b>
6.1	Descrição do Sistema . . . . .	51
6.2	Rede de Petri Interpretada para Controle . . . . .	52
6.2.1	Solução através das Receptividades . . . . .	54
6.2.1.1	Conversão da RPIC em Diagrama Ladder . . . . .	62
6.2.2	Solução através da Priorização de Transições . . . . .	68
6.2.2.1	Conversão da RPIC em Diagrama Ladder . . . . .	72
6.2.3	Solução através do Uso de Arco Inibidor . . . . .	77
6.2.3.1	Conversão da RPIC em Diagrama Ladder . . . . .	77
6.3	Resumo . . . . .	82
<b>7</b>	<b>Conclusão</b>	<b>83</b>



# Lista de Figuras

2.1	Representação de lugar, transição e arco, onde o círculo representa o lugar, a barra a transição e a seta o arco. . . . .	6
2.2	Marcação inicial de uma rede de Petri. . . . .	7
2.3	Evolução da marcação. . . . .	9
2.4	Tipos de conflito. . . . .	12
2.5	Representação de arcos simples e ponderado. . . . .	13
2.6	Representação de arcos habilitador e inibidor. . . . .	14
3.1	Evolução da marcação de uma rede de Petri sincronizada. . . . .	19
3.2	Marcação de uma rede de Petri P-temporizada. . . . .	22
3.3	Marcação de uma rede de Petri T-temporizada. . . . .	23
3.4	Processamento de dados da Rede de Petri Interpretada para Controle. . . . .	26
3.5	Possíveis soluções para transições em conflito. . . . .	29
4.1	Sentido do fluxo de corrente fictícia. . . . .	33
4.2	Representação de contato NA. . . . .	34
4.3	Representação de contato NF. . . . .	35
4.4	Representação da instrução de bobina. . . . .	35
4.5	Representação de instruções SET (L) e RESET (U). . . . .	35
4.6	Instrução de disparo único de subida (OSR). . . . .	36
4.7	Temporizador de atraso ligado (TON). . . . .	36
4.8	Contador crescente (CTU) . . . . .	37
4.9	Instrução Reset (RES) de contadores e temporizadores. . . . .	37

5.1	Parte da estrutura de uma rede de Petri interpretada e segura, com seu respectivo diagrama Ladder. . . . .	40
5.2	Rearrumação dos rungs de $t_1$ e $t_2$ para solução do efeito avalanche. Com o rung de $t_2$ antes do rung de $t_1$ é possível eliminar o efeito avalanche. . . . .	41
5.3	Solução de um circuito em uma rede de Petri interpretada e segura. Além da rearrumação dos rungs é necessário o uso de uma variável responsável por quebrar o loop. . . . .	42
5.4	Solução de transições conflitantes através de priorização de seus disparos. . . . .	43
5.5	Módulo de eventos, cuja função é detectar mudanças na lógica de controle. . . . .	45
5.6	Solução de transições conflitantes através de priorização de seus disparos. . . . .	48
6.1	Ilustração do sistema de automação. . . . .	53
6.2	Rede de Petri com conflitos resolvidos através de receptividades mutuamente excludentes. . . . .	61
6.3	Módulo de inicialização da RPIC da figura 6.2. . . . .	62
6.4	Módulo de eventos da RPIC da figura 6.2. . . . .	64
6.5	Módulo de condições para disparo das transições da figura 6.2 dividido em duas partes. . . . .	65
6.6	Módulo da dinâmica da RPIC da figura 6.2 dividido em duas partes. . . . .	67
6.7	Módulo das ações da RPIC da figura 6.2. . . . .	68
6.8	Rede de Petri com conflitos resolvidos através de priorização do disparo de transições. . . . .	71
6.9	Módulo de inicialização da RPIC da figura 6.8. . . . .	73
6.10	Módulo de condições para disparo das transições da RPIC da figura 6.8 dividido em duas partes. . . . .	74
6.11	Módulo da dinâmica da RPIC da figura 6.8 dividido em duas partes. . . . .	76

6.12	Rede de Petri com conflitos resolvidos através do uso de arcos inibidores.	78
6.13	Módulo de condições para disparo das transições da RPIC da figura	
6.12.	. . . . .	80
6.14	Módulo da dinâmica da RPIC da figura 6.8 dividido em duas partes.	81

# Lista de Tabelas

4.1	Lógica de instrução do contato NA. . . . .	34
4.2	Lógica de instrução do contato NF. . . . .	34
6.1	Identificação dos Sensores e Sinais utilizados na RPIC. . . . .	54
6.2	Receptividades associadas às transições na RPIC. . . . .	60
6.3	Estados e ações associados aos lugares na RPIC. . . . .	60
6.4	Receptividades associadas às transições na RPIC. . . . .	70
6.5	Estados e ações associados aos lugares na RPIC. . . . .	72

# Capítulo 1

## Introdução

Redes de Petri são ferramentas matemáticas utilizadas para modelar SEDs (Peterson 1981, Murata 1989, David & Alla 2005, Cassandras & Lafortune 2008). Em uma rede de Petri, a dinâmica do sistema é representada pela evolução de fichas em determinados nós denominados lugares. Esta característica visual intrínseca à rede de Petri permite com facilidade a visualização de características importantes do sistema, como concorrência, sincronização, paralelismo e compartilhamento de recursos.

A rede de Petri é uma ferramenta intuitiva, capaz de simplificar a modelagem e controle de sistemas a eventos discretos (SEDs). A mesma possui ainda um formalismo matemático, que é representado por sua equação de estado que, por sua vez, também representa a dinâmica do SED modelado. A facilidade de visualização de características do sistema combinada com seu formalismo matemático tornam a rede de Petri uma importante ferramenta na modelagem e controle de SEDs.

Para a implementação de sistemas de automação em SED utiliza-se um tipo de computador mais robusto denominado Controlador Lógico Programável (CLP). Estes por sua vez não pode ser programado em redes de Petri, senão através de cinco linguagens definidas pela norma IEC 1131-3 (IEC 1995) e (Ohman et al. 1998): diagrama de blocos funcionais (Function Block Diagram - FBD); lista de instruções (instruction list - IL); texto estruturado (structured text - ST); diagrama Ladder (Ladder diagram - LD); diagrama funcional sequencial (sequential function chart - SFC). Dentre as cinco linguagens normatizadas, a mais antiga e ainda mais utilizada

é o diagrama Ladder e, por conseguinte, presente na grande maioria dos CLPs disponíveis no mercado.

O diagrama Ladder recebeu este nome devido à sua aparência semelhante à de uma escada, pois apresenta duas barras verticais paralelas entre si, conectadas por uma lógica de controle, formada por degraus denominados rungs (Georgini 2007). O diagrama Ladder é uma linguagem gráfica, e sua rápida assimilação e disseminação se deve aos símbolos encontrados em sua estrutura, que se assemelham aos encontrados nos esquemas elétricos, como bobinas e contatos, por exemplo.

A programação de um sistema de automação diretamente em diagrama Ladder não é indicada, pois apesar de sua característica visual, o diagrama Ladder não permite a análise da dinâmica ou de importantes características desses sistemas, como concorrência, sincronização, paralelismo e compartilhamento de recursos. Por estas razões, é preferível que se realize a modelagem e controle de um SED através de ferramentas mais bem estruturadas, como autômatos, Grafcet ou rede de Petri.

Neste projeto a estrutura escolhida para a modelagem da lógica de controle de um sistema a eventos discretos é um tipo especial de rede de Petri, considerado como uma rede de Petri estendida, denominada Rede de Petri Interpretada para Controle (RPIC). Na literatura, várias redes são propostas com o objetivo realizar o controle supervisorio de sistemas de automação, como em David & Alla (2005), Uzam et al. (1996), Uzam & Jones (1998), Uzam et al. (2000), Zhou & Venkatesh (1999), Jimenez et al. (2001), Satoh et al. (1992), Lee & Lee (2000), Lee et al. (2004), Richardsson & Fabian (2003), Moreira, Botelho & Basilio (2009) e Moreira, Botelho & Hazan (2009). Em Uzam et al. (1996) e Uzam & Jones (1998) são propostas, respectivamente, as chamadas Rede de Petri Controladora e Rede de Petri para Automação. Nesta última, são adicionados à estrutura da rede de Petri a leitura de sensores e atuadores, e são introduzidos os arcos inibidor e habilitador, sendo o método de conversão para Ladder obtido utilizando-se um sistema denominado de lógica de passagem de ficha, que se utiliza da evolução da marcação da rede para descrever o fluxo da lógica de controle. Em Jimenez et al. (2001) um outro método de conversão é proposto, sendo baseado em uma Rede de Petri Interpretada

Temporizada (RPIT). O método não faz uso da matriz de incidência ou equações de estado para a conversão da RPIT para Ladder. Em Satoh et al. (1992) é utilizado um mapeamento da rede de Petri para conversão em diagrama Ladder, mas a resolução de conflito não é bem especificada. Em Lee et al. (2004) uma rede de Petri para controle segura, determinística, temporizada e sincronizada é definida, e um método para conversão é apresentado. Contudo, o método leva a um diagrama Ladder em que uma pequena modificação na Rede de Petri requer a reconstrução de todo o código Ladder e, portanto, não leva a uma conversão simples e direta.

A proposta deste trabalho consiste em desenvolver um novo método de conversão entre o controlador do SED modelado através de uma RPIC, e o diagrama Ladder que é implementado no CLP. A vantagem desta técnica reside exatamente na capacidade de se poder realizar o controle supervisorio do SED através de uma ferramenta mais bem estruturada, no caso uma RPIC, e, em seguida, obter-se diretamente o diagrama Ladder do controlador modelado através da RPIC. A técnica proposta neste trabalho faz uso das equações de estado e matrizes de incidência para realizar uma conversão direta da lógica de controle descrita por uma RPIC para o diagrama Ladder, minimizando esforços na obtenção do diagrama Ladder do controlador do SED modelado através da RPIC. O método proposto neste trabalho permite ainda a obtenção da RPIC a partir do diagrama Ladder, o que pode ser útil caso seja necessário realizar alterações sistema de automação implementado no CLP, sem a necessidade de alteração de toda a sua lógica de programação. Neste projeto é utilizado um sistema de manufatura para ilustrar a técnica e suas vantagens.

A apresentação do conteúdo deste projeto é definida da seguinte forma: no capítulo 2 são descritos os conceitos básicos para o entendimento de redes de Petri, a distinção entre as redes de Petri autônomas e não-autônomas, e os tipos de conflito existentes; no capítulo 3 são abordadas redes de Petri não-autônomas com maior profundidade, como as redes sincronizadas, temporizadas, além das redes de Petri interpretadas para controle; ao final do capítulo 3 é apresentada a rede de Petri interpretada para controle no qual o projeto se baseia; no capítulo 4 é discutido o controlador lógico programável (CLP) e o diagrama Ladder, utilizado para

programar CLPs. No Capítulo 5 o método de conversão da RPIC para Ladder é apresentado. No capítulo 6 um exemplo de um sistema de manufatura é utilizado para ilustrar a modelagem de um SED utilizando a RPIC definida neste trabalho e o método de conversão proposto. Finalmente, no Capítulo 7 as conclusões são apresentadas.

# Capítulo 2

## Redes de Petri

Neste capítulo serão introduzidos conceitos, definições e características das Redes de Petri. Na Seção 2.1 são apresentados os conceitos básicos para a compreensão do funcionamento de uma rede de Petri. Na seção 2.1.1 é apresentado o tipo mais simples de rede de Petri, a rede de Petri autônoma. Além disso, é introduzido o conceito de conflito, os diferentes tipos de conflitos existentes, suas características e definições, e são também apresentados alguns tipos especiais de redes de Petri.

### 2.1 Conceitos Básicos

Redes de Petri são compostas por dois tipos de nós, denominados transições e lugares. Os lugares são representados por círculos e estão associados a condições que devem ser satisfeitas para que as transições possam ocorrer, enquanto que as transições têm uma barra como representação e estão normalmente associadas a eventos. A conexão desses dois nós é feita através de arcos, que podem partir de um lugar para uma transição ou de uma transição para um lugar. A figura 2.1 apresenta os dois tipos de nós encontrados em uma rede de Petri e os arcos que os conectam.

**Definição 1.** *(Cassandras & Lafortune 2008) A estrutura de Petri, ou grafo de Petri, é definida como um grafo bipartido ponderado,  $(P, T, A, \omega)$  em que  $P$  é o conjunto finito de lugares,  $T$  é o conjunto finito de transições,  $A \subseteq (P \times T) \cup (T \times P)$  é o conjunto de arcos que conecta lugares a transições e transições a lugares, e  $\omega : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$  é a função de ponderação, em que  $\omega(p_i, t_j) = 0$  e  $\omega(t_j, p_i) = 0$  se e somente se  $(p_i, t_j) \notin A$  ou  $(t_j, p_i) \notin A$ , respectivamente.  $\square$*

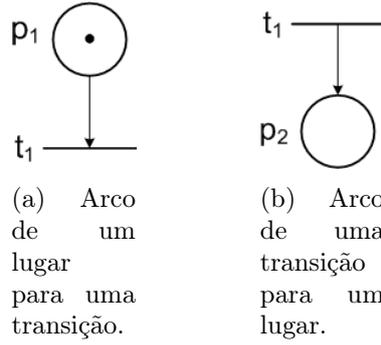


Figura 2.1: Representação de lugar, transição e arco, onde o círculo representa o lugar, a barra a transição e a seta o arco.

Em um grafo de Petri, múltiplos arcos podem conectar dois nós, ou, de forma equivalente, pode-se assinalar um peso para cada arco, representando assim o número total de arcos associado ao arco ponderado. Neste caso, a estrutura recebe o nome de multigrafo. Ainda, supõe-se que o grafo de Petri definido por  $(P, T, A, \omega)$  não tem lugares ou transições isolados. Representa-se o conjunto de lugares por  $P = \{p_1, p_2, \dots, p_n\}$ , e o conjunto de transições por  $T = \{t_1, t_2, \dots, t_m\}$ , onde a cardinalidade de  $P$  e  $T$  é dada por  $|P| = n$  e  $|T| = m$ , respectivamente. Define-se o conjunto de lugares de entrada de uma transição  $t_j$  como  $I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}$ , e o conjunto de lugares de saída de uma transição  $t_j$  como  $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$ .

## 2.1.1 Dinâmica da Rede de Petri

### 2.1.1.1 Marcação

Define-se o número de fichas contidos em um lugar  $p_i$  como  $x(p_i)$ , sendo  $x$  uma função  $x : P \rightarrow \mathbb{N}$ . A representação do número de fichas contidas em cada um dos lugares da rede de Petri é representado por um vetor  $\underline{x}$ , que define a marcação de uma rede de Petri (RP). A marcação inicial de uma RP é representada por  $\underline{x}_0$ . A RP representada na figura 2.2 possui seis lugares ( $|P| = 6$ ) e apenas uma ficha em  $p_1$ , sua marcação inicial é dada por  $x_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ , mas para simplicidade de notação será utilizada a forma  $x_0 = (1, 0, 0, 0, 0, 0)$ .

**Definição 2.** Uma rede de Petri marcada é definida como uma quintupla  $N = (P, T, A, \omega, \underline{x}_0)$ , em que  $(P, T, A, \omega)$  é um grafo de Petri, e  $\underline{x}_0$  é a marcação inicial do conjunto de lugares, definido como  $\underline{x} = [x(p_1) \ x(p_2) \ \dots \ x(p_n)]$ , sendo  $n$  o número de lugares da rede de Petri. □

### 2.1.1.2 Habilitação

Para que uma transição  $t_j$  seja habilitada é necessário que as condições associadas aos seus lugares de entrada,  $I(t_j)$ , sejam atendidas. Essas condições dependem do número de fichas presentes em cada um dos lugares de entrada e do peso dos arcos que conectam esses lugares de entrada à transição  $t_j$ . Isto significa que para uma transição  $t_j$  estar habilitada é necessário que o número de fichas presente em cada um dos lugares de entrada desta transição seja igual ou maior que o peso dos arcos que conectam esses lugares à transição  $t_j$ , ou seja, uma transição  $t_j$  está habilitada se e somente se:

$$x(p_i) \geq \omega(p_i, t_j), \forall p_i \in I(t_j). \quad (2.1)$$

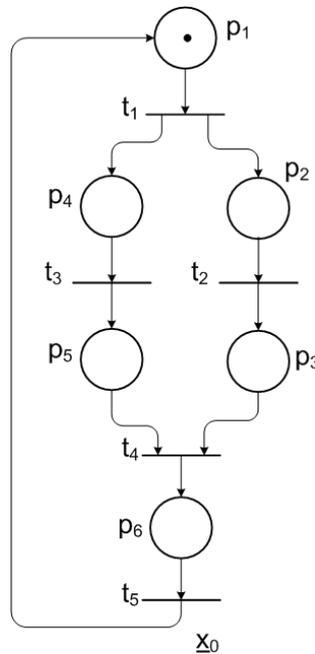


Figura 2.2: Marcação inicial de uma rede de Petri.

### 2.1.1.3 Disparo

Uma transição pode ser disparada se e somente se a mesma estiver habilitada. Entretanto, uma transição habilitada não significa necessariamente que será disparada imediatamente, apenas que esta pode ser disparada. Quando se dispara uma transição  $t_j$ , retira-se um número  $\omega(p_i, t_j)$  de fichas de cada um dos lugares de entrada desta transição ( $I(t_j)$ ), e adiciona-se um número  $\omega(t_j, p_i)$  de fichas, aos lugares de saída de  $t_j$  ( $O(t_j)$ ). O disparo de uma transição habilitada resulta em uma nova marcação. Para o caso da RP apresentada anteriormente na figura 2.2, o disparo da transição  $t_1$  resulta na nova marcação  $\underline{x}_0 \xrightarrow{t_1} \underline{x}_1 = (0, 1, 0, 1, 0, 0)$ . Para a marcação  $\underline{x}_1$ , duas transições estão habilitadas, e portanto, supondo que as duas não possam disparar simultaneamente, duas novas marcações podem ser alcançadas através de  $\underline{x}_1$ ,  $\underline{x}_1 \xrightarrow{t_2} \underline{x}_2 = (0, 0, 1, 1, 0, 0)$  e  $\underline{x}_1 \xrightarrow{t_3} \underline{x}_3 = (0, 1, 0, 0, 1, 0)$ . Para a marcação  $\underline{x}_2$  apenas a transição  $t_3$  está habilitada e, portanto, pode disparar, alcançando a nova marcação  $\underline{x}_2 \xrightarrow{t_3} \underline{x}_4 = (0, 0, 1, 0, 1, 0)$ . Novamente, em  $\underline{x}_4$  apenas a transição  $t_4$  pode disparar obtendo-se assim a marcação  $\underline{x}_4 \xrightarrow{t_4} \underline{x}_5 = (0, 0, 0, 0, 0, 1)$ , desta marcação,  $\underline{x}_5$ , apenas a transição  $t_5$  pode disparar, resultando na marcação inicial  $\underline{x}_6 \xrightarrow{t_5} \underline{x}_0 = (1, 0, 0, 0, 0, 0)$ . Todas as marcações alcançáveis nesta RP estão ilustradas na figura 2.3.

Como pode ser observado na figura 2.3, cada nova marcação é o resultado do disparo de uma única transição. Esta suposição é feita apenas para efeito de ilustração, pois, de fato, disparos simultâneos podem ocorrer. A sequência que leva uma determinada marcação  $\underline{x}$  a outra marcação  $\tilde{\underline{x}}$  qualquer é definida como sequência de disparo, e representada por  $S$ . Por exemplo, na figura 2.3, a sequência de disparo  $S$  que leva a marcação  $\underline{x}_0$  à marcação  $\underline{x}_2$  é definida como  $\underline{x}_0 \xrightarrow{S} \underline{x}_2$ , em que  $S = t_1 t_2$ . Já a marcação  $\underline{x}_0 \xrightarrow{S} \underline{x}_5$ , em que  $S = t_1 t_2 t_3 t_4$ , ou  $S = t_1 t_3 t_2 t_4$ , é responsável por levar a RP da marcação  $\underline{x}_0$  até  $\underline{x}_5$ , e neste caso não é única. Para transições com disparos simultâneos, usa-se colchetes para representar esses disparos simultâneos. Supondo, desta vez, que a RP apresentada na figura 2.3 dispare simultaneamente as transições  $t_2$  e  $t_3$ , a sequência a representar este disparo

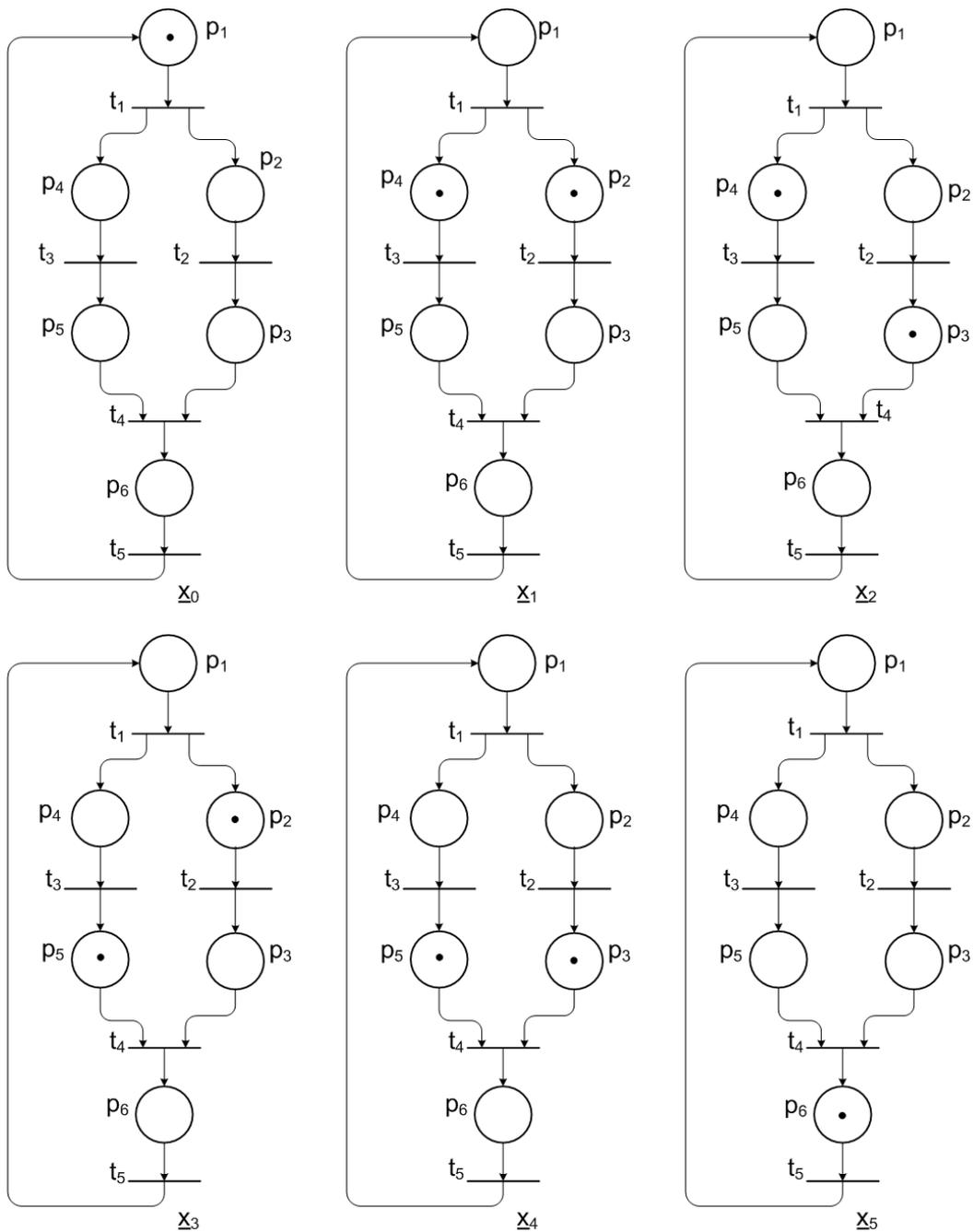


Figura 2.3: Evolução da marcação.

simultâneo seria dada por  $S = [t_2 t_3]$ , e permitiria que a marcação  $\underline{x}_4$  fosse alcançada diretamente a partir de  $\underline{x}_1$ .

#### 2.1.1.4 Equação de Estado e Matriz de Incidência

Como foi comentado anteriormente, o vetor  $\underline{x}$  define a marcação de uma rede de Petri. A evolução desta marcação corresponde ao comportamento dinâmico do SED modelado, e pode ser descrito pela equação de estado de uma rede de Petri. Por exemplo, suponha que as condições associadas a uma transição  $t_j$  sejam verdadeiras para uma dada marcação  $\underline{x}$ , e que esta transição dispare, resultando em uma nova marcação  $\tilde{\underline{x}}$ . A evolução desta marcação é obtida através da seguinte equação:

$$\tilde{x}(p_i) = x(p_i) - \omega(p_i, t_j) + \omega(t_j, p_i), i = 1, \dots, n. \quad (2.2)$$

Pode-se, então, descrever a evolução do vetor de estado, resultante do disparo da transição  $t_j$ , pela seguinte equação de estado:

$$\tilde{\underline{x}} = \underline{x} + W\underline{u}, \quad (2.3)$$

em que  $\underline{u}$  é um vetor coluna formado por zeros, com exceção do  $j$ -ésimo elemento, cujo valor é igual a 1, e que representa o disparo da transição  $t_j$ .  $W$ , por sua vez, representa a matriz de incidência, cujo formato é  $n \times m$ , em que  $n$  representa o número de lugares e  $m$  o número de transições na rede de Petri. A equação que define a matriz de incidência é dada por:

$$W = W_{out} - W_{in}, \quad (2.4)$$

em que  $W_{in} = [\omega_{ij}^{in}]$ , com  $\omega_{ij}^{in} = \omega(p_i, t_j)$  é definida como a matriz de incidência de entrada, e  $W_{out} = [\omega_{ij}^{out}]$ , com  $\omega_{ij}^{out} = \omega(t_j, p_i)$  é definida como a matriz de incidência de saída.

Utilizando-se a equação (2.3) podem-se obter todos os estados de uma rede de Petri alcançáveis a partir de  $\underline{x}$ , se  $\underline{u}$  for um vetor de disparo viável, ou seja, se a transição associada ao vetor  $\underline{u}$  estiver habilitada. Pode-se, ainda, perceber que uma transição  $t_j$  está habilitada para uma marcação  $\underline{x}$  se e somente se todos os elementos do vetor  $\underline{x} - W_{in}\underline{u}$  forem não-negativos, em que  $\underline{u}$  é o vetor de disparo associado a  $t_j$ .

## 2.2 Conflitos

No estudo de redes de Petri é importante a abordagem do conceito de conflito, os diferentes tipos e quando ocorrem. Os tipos de conflito abordados em seguida são: conflito estrutural e conflito efetivo, ambos representados na figura 2.4. Em David & Alla (2005), estes e outros conflitos podem ser encontrados.

### 2.2.0.1 Conflito Estrutural

Na figura 2.4(a) é ilustrado um conflito estrutural. Este tipo de conflito ocorre quando para um mesmo lugar  $p_i$  há duas ou mais transições de saída. Isto é, se  $T_K$  denota o conjunto de transições de saída de um lugar  $p_i \in P$ , então, um conflito representado por  $K = \langle p_i, T_K \rangle$  ocorre quando a cardinalidade do conjunto  $T_k$  é maior que 1. Por exemplo, uma rede de Petri  $(P, T, A, \omega, \underline{x}_0)$  possui um conflito estrutural, se e somente se  $\exists k, l \in \mathbb{N}^*$ , tais que  $\omega_{ik}^{in} > 0$  e  $\omega_{il}^{in} > 0$  para algum  $p_i \in P$ . O conflito estrutural não depende da marcação  $\underline{x}$  da rede de Petri, apenas de sua estrutura, por isso o nome conflito estrutural. Dessa forma, é possível verificar a existência de um conflito estrutural através da análise da matriz  $W_{in}$ . A figura 2.4(a) ilustra a existência de um conflito estrutural  $K = \langle p_2, \{t_1, t_2\} \rangle$ .

### 2.2.0.2 Conflito Efetivo

Um conflito efetivo depende não somente da estrutura da rede de Petri mas também de sua marcação, como pode ser percebido na figura 2.4(b). Um conflito efetivo é representado por  $K^E = \langle p_i, T_K, \underline{x} \rangle$ , em que as transições do conjunto  $T_K$  estão habilitadas e o número de fichas em  $p_i$  é menor que a soma dos pesos de todos os arcos que saem de  $p_i$  e entram nas transições do conjunto  $T_K$ , isto é  $x(p_i) \leq \omega(p_i, t_j) \forall t_j \in T_K$ . Este tipo de conflito depende não somente da estrutura da rede de Petri, mas também de sua marcação e, por conseguinte, não pode ser verificado através da análise da matriz de incidência. A figura 2.4(b) ilustra a existência de um conflito efetivo  $K = \langle p_2, \{t_1, t_2\}, (1, 1, 1) \rangle$ .

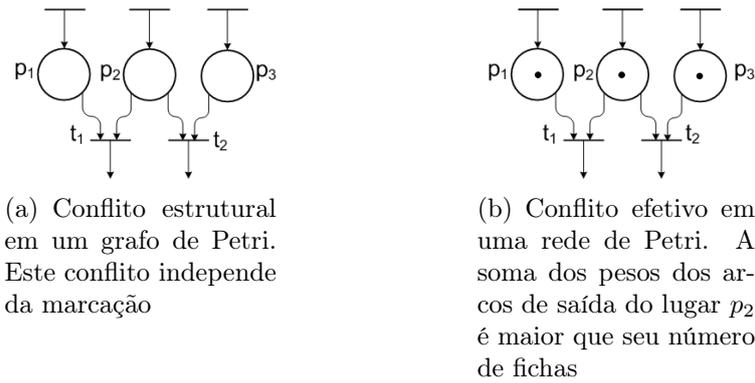


Figura 2.4: Tipos de conflito.

## 2.3 Estruturas Especiais

A seguir alguns tipos especiais de redes de Petri são apresentados.

### 2.3.1 Ordinária

Uma rede de Petri é denominada ordinária quando todos os seus arcos são simples, ou seja, possuem peso unitário, não sendo, portanto, o peso explicitado.

### 2.3.2 Segura

Uma rede de Petri cujo número máximo de fichas em um lugar é 1 para todos os estados alcançáveis é denominada rede de Petri segura.

### 2.3.3 Pura

Um par de nós formado por um lugar  $p_i$  e uma transição  $t_j$  é chamado de laço-próprio, se este lugar  $p_i$ , a que a transição  $t_j$  está associada, operar tanto como informação de entrada, quanto de saída. Uma rede de Petri que não tenha laço-próprio é denominada Rede de Petri pura.

### 2.3.4 Generalizada

Uma rede de Petri é denominada generalizada quando é atribuído peso maior que 1 a pelo menos um arco da rede. Todos os demais arcos cujos pesos não forem especificados são considerados arcos de peso unitário.

## 2.4 Tipos de Arcos

Existem basicamente quatro tipos de arcos utilizados em redes de Petri. Estes são arcos simples, arcos ponderados, arcos habilitadores e arcos inibidores.

### 2.4.1 Arco Simples

Arcos simples ou unitários são os mais comuns e são representados por um seta "cheia", como pode ser visto na figura 2.5(a). Podem partir tanto de um lugar para uma transição quanto de uma transição para um lugar.

### 2.4.2 Arco Ponderado

Arcos ponderados estão presentes em redes de Petri generalizadas para indicar que o arco representa dois ou mais arcos, sendo o seu peso, o indicativo de quantos arcos estão representados pelo arco ponderado. Assim como os arcos simples, podem conectar uma transição a um lugar ou vice-versa. A figura 2.5(b) ilustra um arco ponderado de peso 2.

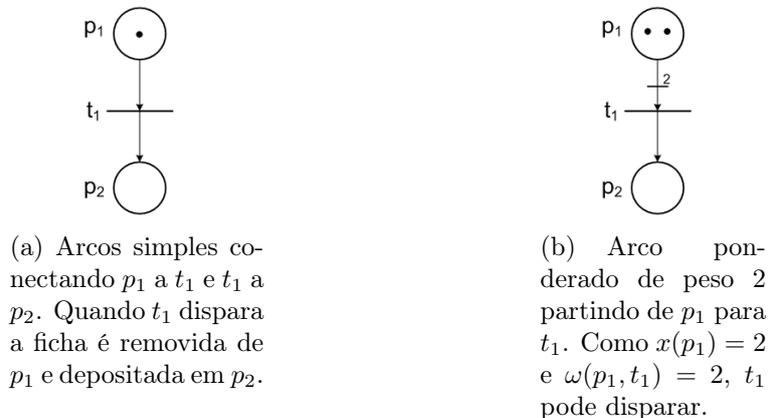


Figura 2.5: Representação de arcos simples e ponderado.

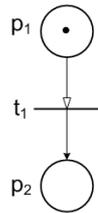
### 2.4.3 Arco Habilitador

Arcos habilitadores são representados por arcos cujo fim possui um seta vazia. Esse tipo de arco parte sempre de um lugar para uma transição. O arco habilitador

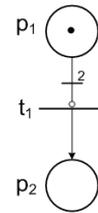
funciona como um laço-próprio, ou seja, quando um lugar de entrada de uma transição  $t_j$  é conectado através de um arco-habilitador, o disparo da transição  $t_j$  não ocasiona a perda da ficha do lugar de entrada, enquanto a transição  $t_j$  ganha uma ficha (Rede de Petri Segura). A figura 2.6(a) ilustra um arco habilitador.

#### 2.4.4 Arco Inibidor

Arcos inibidores são representados por arcos cujo fim possui um círculo vazio. Assim como o arco habilitador, o arco inibidor sempre parte de um lugar para uma transição. Um arco inibidor que conecta um lugar  $p_i$  a uma transição  $t_j$  só permite o disparo desta transição  $t_j$  se o número de fichas contido em  $p_i$  for menor que o peso do arco inibidor que os conecta. Assim como o arco habilitador, o lugar de entrada  $p_i$  não perde fichas no disparo da transição  $t_j$ , enquanto a transição  $t_j$  ganha uma ficha (Rede de Petri Segura). A figura 2.6(b) ilustra um arco inibidor.



(a) Arco habilitador conectando  $p_1$  a  $t_1$ . Quando a transição  $t_1$  dispara a ficha não é removida de  $p_1$  e uma ficha é depositada em  $p_2$ .



(b) Arco inibidor conectando  $p_1$  a  $t_1$ . Neste caso, uma vez que  $x(p_1) = 1$  e  $\omega(p_1, t_1) = 2$ , a transição  $t_1$  pode disparar. O disparo não ocasiona perda de ficha em  $p_1$  e uma ficha é depositada em  $p_2$ .

Figura 2.6: Representação de arcos habilitador e inibidor.

## 2.5 Resumo

Neste capítulo foram discutidos os conceitos básicos para compreensão da estrutura e funcionamento de uma rede de Petri. Conceitos importantes relacionados

à dinâmica das redes de Petri, como equações de estado e matriz de incidência foram apresentados, bem como o conceito de conflito, os diferentes tipos existentes e suas respectivas definições. Além disso, alguns tipos especiais de redes de Petri foram discutidos.

As propriedades encontradas nas redes de Petri não serão discutidas neste projeto, pois seu conhecimento e entendimento, apesar de importante para o estudo das redes de Petri, não são necessários para a compreensão deste trabalho. Informações sobre as diversas propriedades das redes de Petri, bem como suas descrições podem ser encontradas em Peterson (1981), Murata (1989) e David & Alla (2005).

No capítulo seguinte será introduzido o conceito de rede de Petri não-autônoma, um tipo mais complexo que permite temporização e sincronização com eventos externos. Serão apresentados tipos de redes de Petri não-autônomas mais sofisticados, como as redes de Petri interpretadas propostas por diversos autores, juntamente com suas definições e particularidades, e cujo propósito é o controle de SEDs. Ainda no fim deste capítulo será apresentada a rede de Petri adotada neste projeto, e com a qual, posteriormente, no Capítulo 5, será proposto o método de conversão.

# Capítulo 3

## Redes de Petri Não-Autônomas e Interpretadas para Controle

No capítulo anterior foram apresentadas as redes de Petri autônomas que são úteis para modelagem e análise de SEDs. Contudo, esse tipo de rede de Petri não tem funcionalidade quando o que se pretende realizar o controle de um SED. Isto acontece pois, apesar de fornecer informações sobre o comportamento dinâmico do sistema, este tipo de rede de Petri não é capaz de fornecer a informação de quando, de fato, a evolução deste comportamento dinâmico ocorre na rede de Petri. Por isso, um novo modelo de rede de Petri, que permite sincronização com eventos externos e/ou temporização, denominado rede de Petri não-autônoma, é apresentado a seguir. Este capítulo está estruturado da seguinte forma, na seção 3.1 são apresentadas as redes de Petri sincronizadas, temporizadas e interpretadas. Finalmente, na seção 3.2 é apresentada a rede de Petri Interpretada para Controle (RPIC) utilizada para a automação de SEDs neste projeto, e na qual será baseado o método de conversão para o diagrama Ladder.

### 3.1 Redes de Petri Não-Autônomas

#### 3.1.1 Redes de Petri Sincronizadas

Em redes de Petri autônomas, uma transição habilitada é também considerada uma transição que pode ser disparada, ou seja, não há diferenciação entre habilitação e disparo. Isto significa que, uma vez habilitada, uma transição  $t_j$  pode disparar, apesar de não se poder precisar quando esta irá disparar. Isto deve-se ao fato de

não haver uma sincronização com eventos externos e/ou temporização. No caso de uma rede de Petri não-autônoma com sincronização, o disparo de uma transição  $t_j$  depende não somente de sua habilitação, mas também da ocorrência do evento externo associado à mesma.

**Definição 3.** *Uma rede de Petri sincronizada (RPS) é uma tripla  $(N, E, Sync)$ , onde  $N$  é uma rede de Petri marcada,  $E$  é o conjunto de eventos externos, e  $Sync$  é uma função do conjunto de transições  $T : N \rightarrow E \cup \{e\}$ , sendo  $e$  o evento sempre ocorrente.* □

O conjunto de eventos externos  $E$  é da forma  $E = \{E^1, E^2, \dots\}$ . A notação usada para os eventos externos é  $E^i$  ( $E$  com índice superior  $i$ ), contudo para facilitar uma correspondência direta entre o evento externo e a transição  $t_j$  à qual o evento está associado, a notação adotada é  $E_j$  ( $E$  com subíndice  $j$ ). Ambas as notações são as mesmas adotadas em David & Alla (2005). Por exemplo, um evento  $E^4 \in E$  que esteja associado a uma transição  $t_5 \in T$  pode ser representado por  $E_5$ , para indicar sua posição correspondente no grafo de Petri. No caso de um mesmo evento estar associado a duas ou mais transições, o mesmo evento recebe um índice inferior diferente para indicar cada transição a que esteja associado e, por conseguinte, sua posição na rede de Petri.

Em David & Alla (2005) é considerada a existência de dois tipos de marcação: estável e instável. Segundo os autores, em uma rede de Petri não-autônoma, uma transição imediata é uma transição que é disparada assim que é habilitada, exceto quando há um conflito entre duas ou mais transições imediatas. Ainda segundo os autores, se uma transição  $t_j$  é imediata, qualquer marcação que habilite  $t_j$  é dita instável. Uma transição imediata em uma RPS é representada por uma transição sincronizada com o evento que sempre ocorre  $e$ .

Logo, uma rede de Petri sincronizada possui uma marcação instável  $x(p_i)$  se e somente se possui ao menos um lugar  $p_i \in P$  e uma transição  $t_j \in T$ , tal que  $x(p_i) \geq \omega(p_i, t_j), \forall p_i \in I(t_j)$ , equação 2.1, e a transição  $t_j$  é receptiva ao evento  $e$ . Esta marcação quando obtida é imediatamente modificada pelo disparo da transição

$t_j$ , por isso chamada por David & Alla (2005) de marcação instável.

**Observação:** Diferentemente de David & Alla (2005), é considerado neste projeto que cada transição  $t_j$  só pode disparar uma única vez quando habilitada, mesmo que esta transição  $t_j$  esteja associada ao evento que sempre ocorre  $e$ .

### 3.1.1.1 Marcação e Disparo

Em uma rede de Petri, considera-se que eventos externos ocorrem instantaneamente, ou seja, seu tempo de duração é zero e, portanto, pode ser desconsiderado. Assim como o tempo de duração dos eventos externos, o disparo das transições em um rede de Petri sincronizada é instantâneo. Na ocorrência de um evento externo ou conjunto de eventos externos simultâneos, sendo este um subconjunto de  $E \cup \{e\}$ , várias transições podem disparar simultaneamente, desde que estejam habilitadas. Sejam duas transições  $t_j$  e  $t_k$  habilitadas ao mesmo tempo e receptivas a um mesmo evento. O disparo simultâneo das transições  $t_j$  e  $t_k$  é denominado sequência elementar de disparo. Um exemplo de sequência elementar de disparo é ilustrado na figura 3.1. Num primeiro momento a RPS da figura 3.1 encontra-se em sua marcação inicial  $\underline{x}_0 = (1, 0, 0, 0, 0, 0, 0)$ , a transição  $t_1$  está habilitada e, portanto, receptiva a  $e$ , assim disparando e alcançando a marcação  $\underline{x}_1$ ,  $\underline{x}_0 \xrightarrow{t_1} \underline{x}_1 = (0, 1, 1, 1, 0, 0, 0)$ . Nesta marcação as transições  $t_2$ ,  $t_3$  e  $t_4$  estão habilitadas e portanto receptivas aos eventos  $E^1$ ,  $E^2$  e  $E^3$ , respectivamente. Caso o evento  $E^1$  ocorra primeiro, a transição  $t_2$  irá disparar, será obtida a nova marcação  $\underline{x}_2$ ,  $\underline{x}_1 \xrightarrow{t_2} \underline{x}_2 = (0, 0, 1, 1, 1, 0, 0)$ . Em seguida, caso ocorra o evento  $E^2$  e a transição  $t_3$  dispare, e uma nova marcação é alcançada  $\underline{x}_2 \xrightarrow{t_3} \underline{x}_3 = (0, 0, 0, 1, 1, 1, 0)$ . Caso, neste momento, venha a ocorrer o evento  $E^3$ , tanto a transição  $t_4$  e  $t_5$  estarão receptivas a este evento, e portanto, dispararão simultaneamente, alcançando a marcação  $\underline{x}_3 \xrightarrow{[t_4 t_5]} \underline{x}_4 = (0, 1, 0, 0, 0, 1, 1)$ . Finalmente, se o evento  $E^1$  ocorrer novamente neste momento, as transições  $t_6$  e  $t_2$  estarão habilitadas e receptivas ao eventos  $E^1$ , disparando simultaneamente. A nova marcação alcançada é  $\underline{x}_4 \xrightarrow{[t_2 t_6]} \underline{x}_5 = (1, 0, 0, 0, 1, 0, 0)$ .

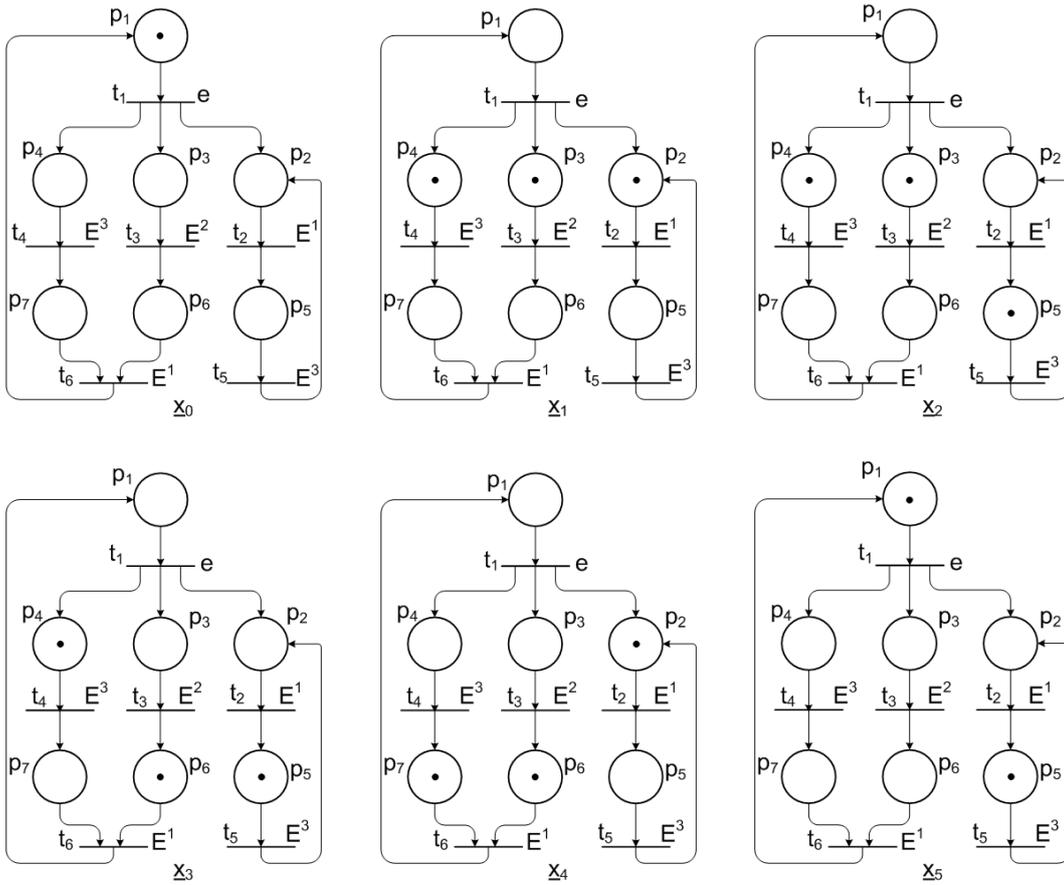


Figura 3.1: Evolução da marcação de uma rede de Petri sincronizada.

### 3.1.1.2 Conflito Real

Em uma rede de Petri autônoma os conflitos não podem ser resolvidos devido à falta de sincronização e/ou temporização. Já em uma rede não-autônoma os conflitos podem ser, em alguns casos, resolvidos com a atribuição de eventos e/ou temporização às transições da rede de Petri. Por exemplo, considere duas transições  $t_j$  e  $t_k$  em conflito efetivo, mas com eventos independentes  $E_j$  e  $E_k$  respectivamente associados. Ainda que as duas transições estejam habilitadas simultaneamente, apenas a transição cujo evento associado ocorrer será disparada, eliminando, desse modo, o conflito. Entretanto, nos casos em que ocorra um conjunto de eventos externos dependentes associados a duas ou mais transições em conflito efetivo, ou o mesmo evento esteja associado a essas duas transições, o conflito não é resolvido. Retornemos ao exemplo das duas transições  $t_j$  e  $t_k$  em conflito efetivo e com os eventos

$E_j$  e  $E_k$  respectivamente associados. Na prática é possível que os eventos  $E_j$  e  $E_k$  ocorram num mesmo ciclo de varredura do CLP, onde é implementado o sistema de automação. Neste caso o conflito não é resolvido, mesmo com a sincronização por eventos externos. Na realidade este tipo de conflito, onde transições conflitantes estão habilitadas e seus respectivos eventos, ou conjunto de eventos pertencentes a  $E$ , ocorrem é denominado **conflito real**. A resolução de um conflito real pode ser obtida através de atribuição de prioridades entre as transições envolvidas. Em caso de haver um conflito efetivo entre uma transição imediata  $t_j$  e uma transição não-imediata  $t_k$ , a transição imediata  $t_j$  tem sempre prioridade sobre a transição não-imediata  $t_k$ , pois a transição  $t_j$  está associada ao evento que sempre ocorre  $e$ .

### 3.1.2 Redes de Petri Temporizadas

Redes de Petri temporizadas são aquelas que possuem, como o próprio nome já diz, uma temporização associada à sua estrutura. Uma rede de Petri pode ter uma temporização associada aos seus lugares, denominada rede de Petri P-temporizada (David & Alla 2005), ou pode ter uma temporização associada às suas transições, denominada, nesse caso, rede de Petri T-temporizada. Ainda há um terceiro tipo onde é considerada a possibilidade de haver fichas reservadas em uma rede de Petri T-temporizada. Este tipo é denominado rede de Petri T-temporizada com ficha reservada (David & Alla 2005).

Em redes de Petri temporizadas o atraso é representado por  $d$ , que equivale ao tempo de espera de uma condição (rede de Petri P-temporizada) ou de um disparo (rede de Petri T-temporizada). Para os casos em que não seja especificado o tempo relacionado ao lugar ou transição, considera-se que este tempo é nulo. Isto significa dizer que no caso de uma rede de Petri P-temporizada cujo lugar  $p_i$  não possui temporização a ficha fica disponível imediatamente e, analogamente, no caso de uma rede de Petri T-temporizada cuja transição  $t_j$  não possui temporização, o disparo é feito assim que esta se torna habilitada.

### 3.1.2.1 Redes de Petri P-Temporizadas

Nesse tipo de rede de Petri, um tempo  $d_i$ ,  $i \in \{1, 2, \dots, n\}$  está associado a cada lugar, e indica o lugar ao qual o atraso se associa.

**Definição 4.** *Uma rede de Petri P-temporizada é um par  $(N, \text{Tempo})$ , onde  $N$  é uma rede de Petri marcada, e  $\text{Tempo}$  é uma função  $\text{Tempo} : P \rightarrow \mathbb{R}$ ,  $p_i \mapsto \text{Tempo}(p_i) = d_i$  equivale ao tempo associado a cada lugar  $p_i$ .  $\square$*

A rede de Petri temporizada aqui descrita é definida com temporização por números reais. Nesta estrutura de rede de Petri P-temporizada, quando uma ficha é depositada em um lugar  $p_i$ , e este possui uma temporização  $d_i$  não nula a ele associada, a mesma deverá permanecer por este tempo mínimo  $d_i$  naquele lugar. Durante este período de tempo  $d_i$ , a ficha encontra-se indisponível e, somente após passado esse tempo  $d_i$  é que a ficha torna-se, então, disponível.

Em uma rede de Petri P-temporizada, qualquer marcação  $\underline{x}$  alcançada é a soma de outras duas marcações  $\underline{x}^d + \underline{x}^i$ , sendo  $\underline{x}^d$  a marcação correspondente às fichas disponíveis, e  $\underline{x}^i$  é a marcação correspondente às fichas indisponíveis. Uma transição  $t_j$  será considerada habilitada para a marcação  $\underline{x} = \underline{x}^d + \underline{x}^i$ , se esta transição  $t_j$  estiver habilitada para a marcação  $\underline{x}^d$ .

A figura 3.2 ilustra a marcação de uma rede de Petri P-temporizada, onde as fichas claras, representam as fichas indisponíveis,  $\underline{x}^i$ . A marcação inicial da rede de Petri P-temporizada é representada por  $\underline{x}_0 = (1, 0, 1, 0)$ , neste instante a ficha encontra-se disponível em  $p_1$  e  $p_3$ , como pode ser observado na figura 3.2(a), e, portanto, a transição  $t_1$  está habilitada e dispara, sendo a marcação  $\underline{x}_1$  obtida. Neste ponto deve-se ressaltar que a marcação  $\underline{x}_1 = \underline{x}_1^d + \underline{x}_1^i$ , em que  $\underline{x}_1^d = (0, 0, 0, 0)$  e  $\underline{x}_1^i = (0, 1, 0, 0)$ . Entretanto, como o que indica, de fato, a marcação de uma rede de Petri P-temporizada é a marcação disponível, então  $\underline{x}_1^d = (0, 0, 0, 0)$ , indicando que a transição  $t_2$  não está habilitada e, portanto, não pode ser disparada, figura 3.2(b). Apenas após transcorrido um tempo  $d_2 = 3$ , a ficha fica disponível em  $p_2$ , e a transição  $t_2$  habilitada dispara. A figura 3.2(c) é meramente ilustrativa, pois no momento em que a ficha fica disponível, a transição  $t_2$  dispara imediatamente. A

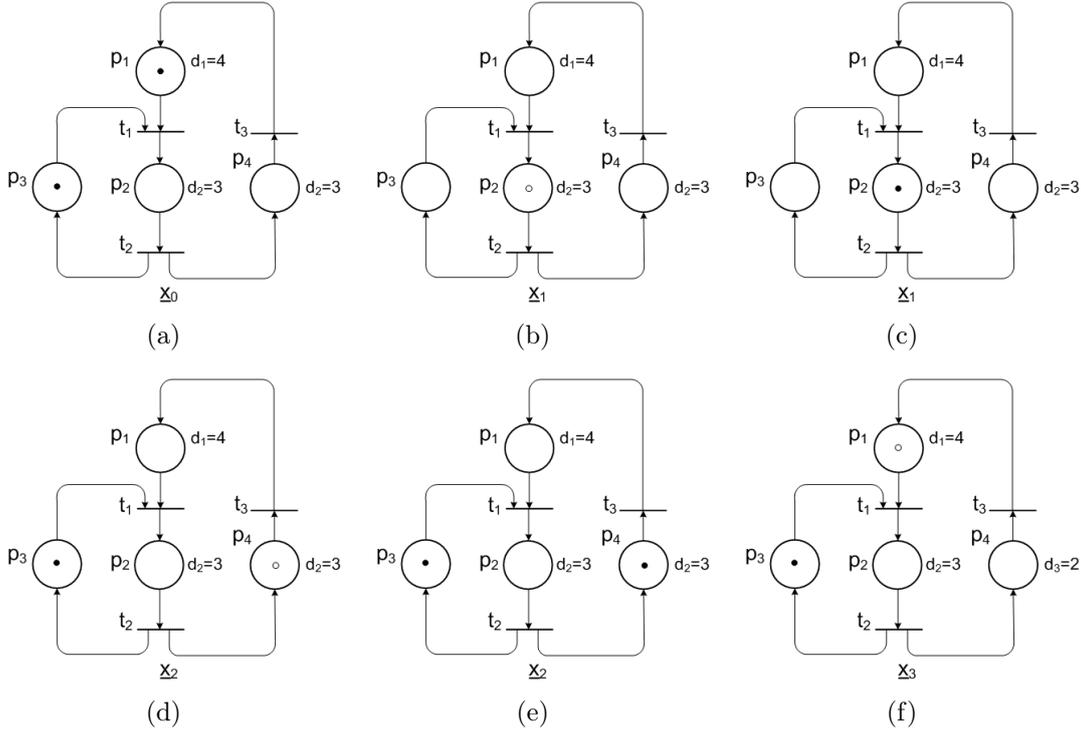


Figura 3.2: Marcação de uma rede de Petri P-temporizada.

marcação  $\underline{x}_2 = \underline{x}_2^d + \underline{x}_2^i$  é então alcançada, como pode ser observado através da figura 3.2(d), sendo  $\underline{x}_2^d = (0, 0, 1, 0)$  e  $\underline{x}_2^i = (0, 0, 0, 1)$ . Após transcorrido um tempo  $d_3 = 2$  a ficha fica disponível em  $p_4$ , e a transição  $t_3$  dispara assim que é habilitada, neste instante o tempo total transcorrido é  $d_{total} = d_2 + d_3 = 5$ . Novamente a figura 3.2(e) ilustra apenas o momento em que a ficha fica disponível em  $p_4$ . A marcação alcançada agora é  $\underline{x}_3 = \underline{x}_3^d + \underline{x}_3^i$ , figura 3.2(f), sendo  $\underline{x}_3^d = (0, 0, 1, 0)$  e  $\underline{x}_3^i = (1, 0, 0, 0)$ . Pode-se perceber nesta marcação que há uma ficha disponível em  $p_3$  e uma indisponível em  $p_1$ , não habilitando a transição  $t_1$ , que só ficará habilitada e, portanto, disparar quando um tempo  $d_1 = 3$  for transcorrido.

**Observação:** O disparo de uma rede de Petri P-temporizada é equivalente ao disparo de uma rede não temporizada. Isto significa que uma vez que uma transição  $t_j$  está habilitada, então esta transição  $t_j$  é disparada imediatamente, o que consiste em retirar as fichas de  $I(t_j)$  e colocar em  $O(t_j)$ . Ao ser depositada em um lugar  $p_i$ , após um disparo, em um dado instante de tempo  $t_1$ , essa ficha fica indisponível por um intervalo de tempo  $[t_1, t_1 + d_i)$ .

### 3.1.2.2 Redes de Petri T-Temporizadas

Este tipo de rede de Petri possui uma temporização  $d_j$ , desta vez associada às suas transições, sendo  $j \in \{1, 2, \dots, m\}$ , e indica a transição à qual o atraso se associa. Analogamente à rede de Petri P-temporizada, quando o atraso  $d_j$  não for especificado este será considerado nulo.

**Definição 5.** Uma rede de Petri T-temporizada é um par  $(N, \text{Tempo})$ , em que  $N$  é uma rede de Petri marcada, e  $\text{Tempo}$ , por sua vez, é uma função  $\text{Tempo} : T \rightarrow \mathbb{R}$ ,  $t_j \mapsto \text{Tempo}(t_j) = d_j$  equivale ao atraso de disparo associado a cada transição  $t_j$ .  $\square$

Assim como a rede de Petri P-temporizada descrita anteriormente, a rede de Petri T-temporizada é definida com temporização por números reais. Em uma rede de Petri T-temporizada uma transição  $t_j$  só é disparada após passado um tempo  $d_j$ , exceto nos casos onde ocorre um conflito entre duas ou mais transições habilitadas e que possuam a mesma temporização  $d_j$ .

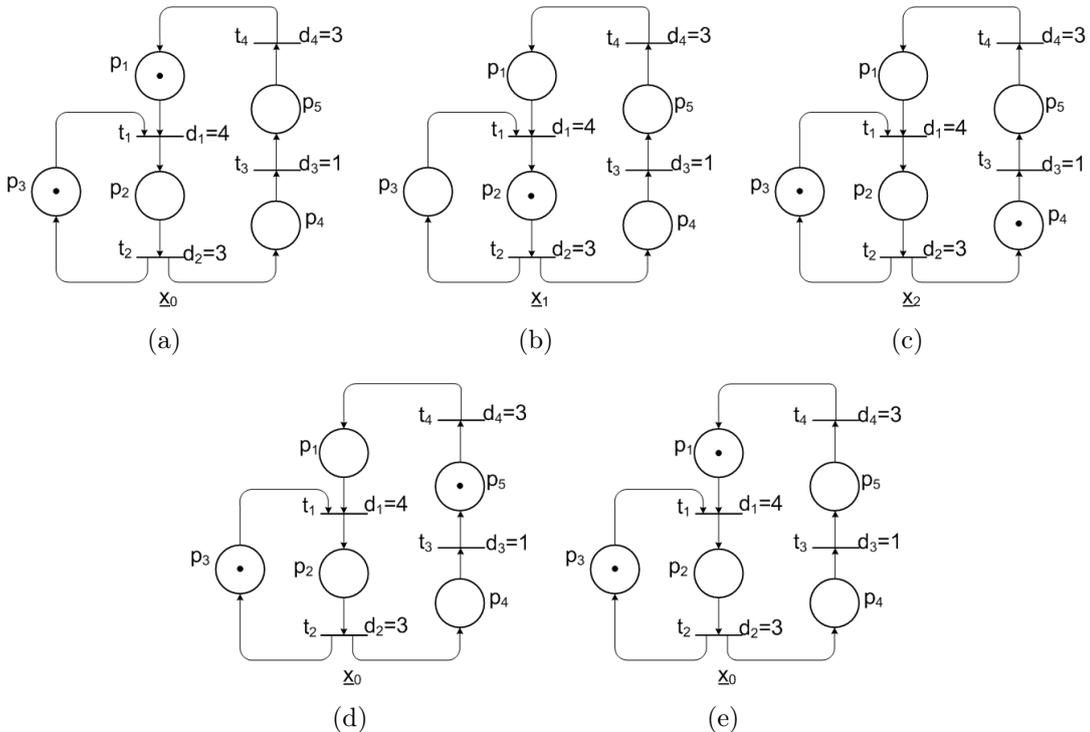


Figura 3.3: Marcação de uma rede de Petri T-temporizada.

Considere o exemplo mostrado na figura 3.3(a). A marcação inicial é  $\underline{x}_0 =$

$(1, 0, 1, 0, 0)$ , e a transição  $t_1$  encontra-se habilitada. Após transcorrido o tempo de disparo  $d_1 = 4$ , associado à transição  $t_1$ , esta dispara, sendo a marcação  $\underline{x}_1 = (0, 1, 0, 0, 0)$  alcançada, como pode ser observado na figura 3.3(b). Nesta marcação a transição  $t_2$  está habilitada, mas só após transcorrido um tempo  $d_2 = 3$ , essa transição irá disparar, alcançando a marcação  $\underline{x}_2 = (0, 0, 1, 1, 0)$ , como pode ser observado na figura 3.3(c). Para a marcação  $\underline{x}_2$ , a transição  $t_3$  está habilitada, mas dispara somente quando seu tempo  $d_3 = 1$  for transcorrido, obtendo-se a marcação  $\underline{x}_3 = (0, 0, 1, 0, 1)$ , como pode ser visto na figura 3.3(d). Finalmente, após transcorrido um tempo  $d_4 = 3$ , a transição  $t_4$ , que fora habilitada pela marcação  $\underline{x}_3$ , dispara, sendo a marcação inicial novamente obtida,  $\underline{x}_0 = (1, 0, 1, 0, 0)$  ilustrada na figura 3.3(e).

**Observação:** No caso de existir um conflito efetivo entre duas ou mais transições, aquela transição que tiver a menor temporização é disparada primeiro, evitando assim o conflito real, com exceção dos casos em que duas ou mais dessas transições envolvidas em conflito tiverem a mesma temporização, ou seja, o mesmo  $d_j$ .

### 3.1.3 Redes de Petri Interpretadas

Redes de Petri interpretadas são redes de Petri não-autônomas que possuem sincronização com eventos externos e/ou temporização. Essas redes possuem essa denominação devido às diversas aplicações, ou interpretações, que podem ter, dependendo da área em que se pretende aplicá-las e sua adaptação às mesmas. O uso da rede interpretada neste projeto está voltada para o controle de processos de automação e, por conseguinte, a mesma recebe o nome de Rede de Petri Interpretada para Controle. Ainda assim é importante ressaltar a existência de outras redes interpretadas com o mesmo objetivo, e propostas por outros autores como em: David & Alla (2005), Uzam et al. (1996), Uzam & Jones (1998), Uzam et al. (2000), Zhou & Venkatesh (1999) e Jimenez et al. (2001).

### 3.1.3.1 Rede de Petri Interpretada para Controle de David e Alla

A Rede de Petri Interpretada para Controle proposta por David & Alla (2005) recebe informações do meio externo, ou do sistema controlado, sendo estas informações formadas por variáveis Booleanas  $C_j^e$ , que representam condições do meio externo, e  $E_j$ , eventos associados a mudanças no nível lógico dos sinais enviados pelos sensores do meio externo. Esta rede também envia informações para o meio, que são as ações responsáveis pela execução de tarefas. Estas informações de saída podem ser de nível ( $A_i$ ), saídas Booleanas dependendo da marcação, de impulso, eventos que dependem de uma mudança na marcação ( $B_i$ ), e variáveis que resultam de um cálculo  $V_k$ , podendo ser numérica ou Booleana. A figura 3.4 representa esta parte de processamento de dados e as variáveis de entrada e saída comentadas anteriormente.

Esta rede interpretada para controle possui ainda as seguintes características: é sincronizada por eventos externos, é segura, é determinística, e possui uma parte responsável pelo processamento de dados, cujo estado é definido por um conjunto de variáveis  $V = \{V_1, V_2, \dots\}$ . O estado é modificado pelo operador  $O_i$  que é associado aos lugares da rede de Petri interpretada para controle, e determina o valor de  $C_j^o$ ; a parte responsável pelo controle recebe informação Booleana  $C_j^e$  do ambiente externo, e envia ações de nível  $A_i$  e de impulso  $B_i$ , associadas aos lugares, para o meio externo.

Em uma Rede de Petri Interpretada para Controle, a parte responsável pelo controle envia ordens de informação  $O_i$  para a parte de processamento de dados, e recebe informações Booleanas do dado processado  $C_j^o$  no meio externo. Neste modelo supõe-se que o cálculo não tem duração, ou seja, o resultado do cálculo é obtido imediatamente.

#### Regras para Disparo

Uma transição  $t_j$  dispara se esta transição  $t_j$  estiver habilitada, isto é, se a condição  $C_j$  associada à transição  $t_j$  for verdadeira, e quando o evento  $E_j$  associado

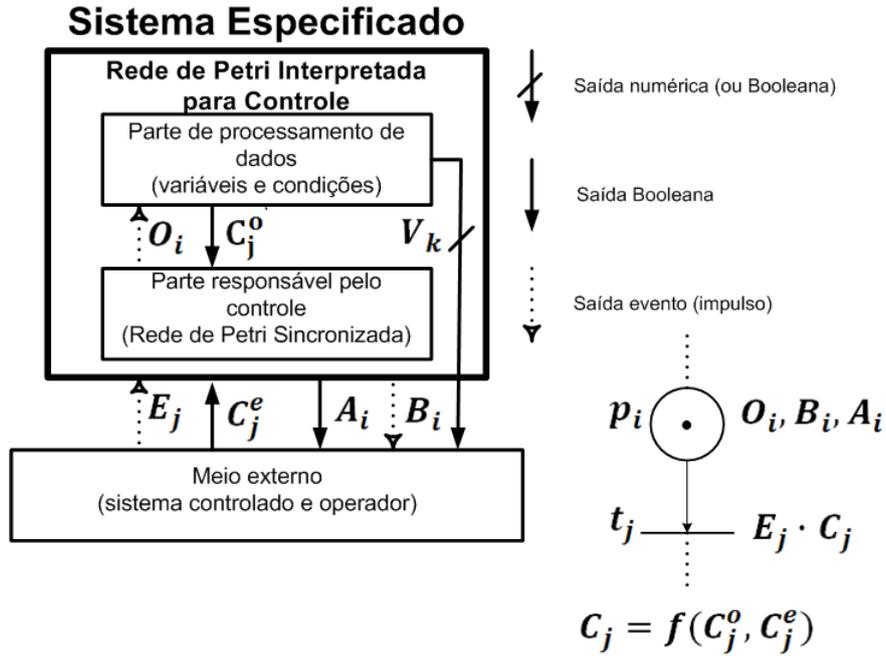


Figura 3.4: Processamento de dados da Rede de Petri Interpretada para Controle.

à transição  $t_j$  ocorrer. O produto entre  $E_j$  e  $C_j$  é chamado de receptividade de  $t_j$ , e é da forma  $R_j = E_j \cdot C_j$ . Além disso, se  $E_j$  não for especificado, então  $E_j = e$ ; se  $C_j$  não for especificado,  $C_j = 1$ , ou seja, condição verdadeira; se  $O_i$  não for especificado,  $O_i$  é o operador identidade, ou seja, não haverá modificação no estado das variáveis na parte de processamento de dados; se  $B_i$  não for especificado, não há ação impulsional; se  $A_i$  não for especificado, todas as saídas Booleanas associadas às ações de nível possuem valor 0.

Abaixo segue o algoritmo apresentado por (David & Alla 2005) que mostra como, e em que ordem, são realizadas operações e ações, bem como a evolução da marcação da Rede de Petri Interpretada para Controle.

#### Algoritmo 1.

1. *Inicialização da marcação; inicialização de todas as ações de nível com valor 0; execução das operações e ações de impulso associadas aos lugares marcados. Ir para o passo 5.*
2. *Esperar pelo próximo evento externo. Quando um evento externo ocorrer (ou*

- conjunto de eventos), determinar o conjunto de transições  $\mathcal{T}$  receptivas a esse evento (ou conjunto de eventos). Se  $\mathcal{T}$  for vazio ir para o passo 6.
3. Realizar a sequência de disparo elementar para todas as transições em  $\mathcal{T}$ .
  4. Realizar todas as operações e ações de impulso associadas aos lugares que ficaram marcados em 3.
  5. Determinar o conjunto de transições  $\mathcal{T}$  receptivo ao evento que sempre ocorre e, se  $\mathcal{T} \neq \emptyset$ , ir para o passo 3. Caso contrário ir para o passo 6.
  6. A marcação é estável.
    - (a) Determinar o conjunto  $\mathfrak{A}_0$  de todas as ações de nível que devem ser desativadas.
    - (b) Determinar o conjunto  $\mathfrak{A}_1$  de todas as ações de nível que devem ser ativadas
    - (c) Colocar em 0 todas as ações que pertencerem a  $\mathfrak{A}_0$  e não pertencerem a  $\mathfrak{A}_1$ . Colocar em 1 todas as ações que pertencerem a  $\mathfrak{A}_1$ . Ir para o passo 2.

## 3.2 Rede de Petri Interpretada para Controle

**Definição 6.** Uma rede de Petri Interpretada para Controle (RPIC) é uma n-úpla  $(N, C, E, O, O_I, En, In, Pr, D)$ , em que  $N = (P, T, A, \omega, \underline{x}_0)$  é uma rede de Petri marcada. O conjunto de transições é  $T = T_0 \cup T_D$ , onde  $T_0$  é o conjunto de transições sem atraso associado, e  $T_D$  é o conjunto de transições com atraso associado;  $C = \{C_1, \dots, C_m\}$  é o conjunto de condições associadas ao conjunto de transições  $T$ ;  $E = \{E_1, \dots, E_m\}$  é o conjunto de eventos também associados ao conjunto de transições  $T$ ;  $O$  e  $O_I$  são, respectivamente, o conjunto de ações de nível e impulsivas associadas ao conjunto de lugares  $P$ ;  $En : (P \times T) \rightarrow \mathbb{N}$  é uma função de habilitação que define os arcos habilitadores de um lugar para uma transição;  $In : (P \times T) \rightarrow \mathbb{N}$

é uma função de inibição que define os arcos inibidores de um lugar para uma transição;  $Pr = \{(T_{K_1}, O_{r_1}), (T_{K_2}, O_{r_2}), \dots, (T_{K_l}, O_{r_l})\}$  em que  $T_{K_i}$ , para  $i = 1, \dots, l$  é um conjunto de transições envolvidas em um conflito efetivo e que se deseja atribuir uma ordem de prioridade para o disparo, e  $O_{r_i}$  é um sequência formada pelos índices das transições em conflito, e representando a ordem de prioridade no disparo das transições; finalmente,  $D = \{d_j : t_j \in T_D\}$  é o conjunto de atrasos de disparo associados às transições temporizadas de  $T_D$ .  $\square$

A rede de Petri proposta neste projeto é baseada na rede de Petri de David & Alla (2005). Contudo, diferentemente da definição utilizada em redes de Petri T-temporizadas, que admitem transições com temporização nula, a RPIC proposta neste trabalho possui dois conjuntos de transições, com temporização, representado por  $T_D$ , e sem temporização, representado por  $T_0$ .

A rede de Petri proposta é segura, isto é, cada lugar  $p_i$  possui no máximo uma ficha, e é determinística, ou seja, para toda e qualquer marcação alcançada o disparo da rede é sempre único. Isto significa que todos os tipos de conflito podem ser resolvidos, inclusive o conflito real. A RPIC definida neste trabalho permite sincronização com eventos externos e/ou temporização, diferentemente de David & Alla (2005), que também considera a temporização como um evento externo, além de permitir priorização entre transições como forma de resolução de conflitos. A RPIC possui também uma parte de processamento de dados, assim como a rede definida por David & Alla (2005). A rede permite três tipos de arcos: arcos simples, arcos inibidores e arcos habilitadores. Nesta rede, o arco habilitador funciona como um laço-próprio, não sendo capaz de resolver conflitos.

Assim como na rede definida por David & Alla (2005), nesta RPIC uma transição  $t_j$  não temporizada dispara se  $t_j$  está habilitada, se a condição  $C_j$  associada à transição  $t_j$  é verdadeira, e quando o evento  $E_j$  associado à transição  $t_j$  ocorre. O produto entre  $E_j$  e  $C_j$  é chamado de receptividade de  $t_j$ , e é da forma  $R_j = E_j \cdot C_j$ . Além disso, se  $E_j$  não for especificado, então  $E_j = e$ ; se  $C_j$  não for especificado, então  $C_j = 1$ . Desta forma, cada uma das  $m$  transições pertencentes a  $T$  possui uma expressão Booleana de receptividade associada.

No caso de haver conflito efetivo entre duas ou mais transições na RPIC, o mesmo pode ser resolvido de três formas: através de receptividades mutuamente excludentes entre as transições em conflito; através de priorização de transições; e através da inclusão de arcos inibidores às transições em conflito. Há ainda a possibilidade da combinação das soluções citadas acima. A escolha da solução depende da configuração e da complexidade do sistema, e dos recursos disponíveis (disponibilidade de mais ou menos sensores).

A figura 3.5 ilustra um conflito entre as transições  $t_1$  e  $t_2$ , com receptividades  $R_1 = E_1C_1$  e  $R_2 = E_2C_2$ , respectivamente. Contudo, supondo-se que seja possível as duas transições  $t_1$  e  $t_2$  tornarem-se disparáveis ao mesmo tempo, ainda que tenham receptividades associadas, é necessário escolher uma ordem de prioridade para o disparo das transições. Neste exemplo escolheu-se a prioridade no disparo de  $t_1$  sobre  $t_2$ , ou seja, a transição  $t_2$  só pode disparar se a transição  $t_1$  não for disparável. Isso significa que  $t_2$  só pode disparar se ocorrer  $E_2C_2(\overline{p_1} + \overline{C_1} + \overline{E_1})$ . Com a solução por arco inibidor não é necessário negar o disparo de  $t_1$ , apenas a habilitação de  $t_1$ , ou seja,  $t_2$  irá disparar se  $t_1$  não estiver habilitada. Neste caso  $t_2$  pode disparar se  $E_2C_2\overline{p_1}$ . A solução por arco habilitador é demonstrada na figura 3.5(b).

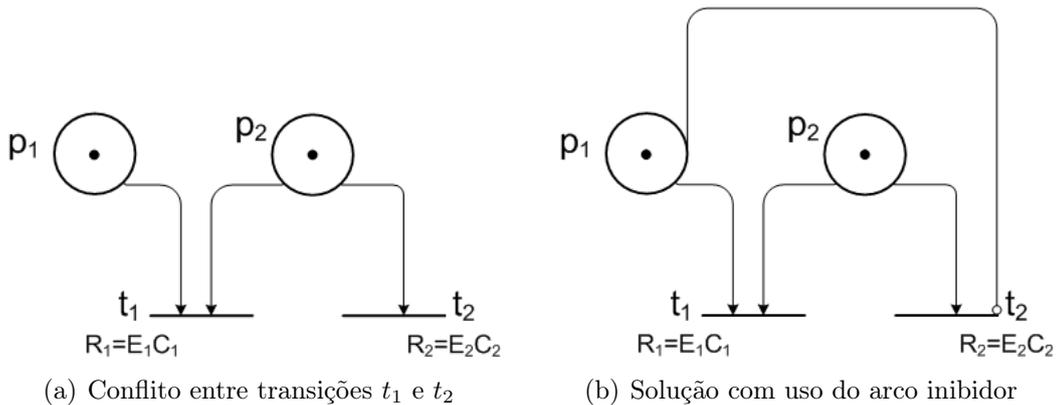


Figura 3.5: Possíveis soluções para transições em conflito.

### 3.3 Resumo

Neste capítulo foram apresentadas algumas redes de Petri não-autônomas, como a rede de Petri sincronizada, a rede de Petri P-temporizada, e a rede de Petri T-temporizada. Em seguida foi abordada a Rede de Petri Interpretada, cujo modelo representa uma sofisticação das primeiras redes de Petri não-autônomas, e que possuem a capacidade de representar a lógica de controle de sistemas a eventos discretos. Finalmente foi apresentado o modelo de Rede de Petri Interpretada para Controle proposto para a realização do método de conversão.

A RPIC proposta neste capítulo possui características importantes, como a presença de arcos habilitadores e inibidores em sua estrutura, e a possibilidade de utilização de priorização entre transições para lidar com conflitos reais. Por ser capaz de resolver conflitos reais através da utilização de receptividades mutuamente excludentes para transições em conflito, e ainda através de priorização de transições em conflito, toda sequência elementar de disparo da RPIC é única, o que a torna determinística.

No capítulo seguinte são abordados o funcionamento e programação de um controlador lógico programável e os conceitos da lógica Ladder necessários para a compreensão do diagrama Ladder para implementação das informações no CLP.

# Capítulo 4

## Controlador Lógico Programável e Diagrama Ladder

Neste capítulo é apresentado, na seção 4.1 o funcionamento de um CLP, e na seção 4.2 a linguagem Ladder, que é a linguagem utilizada para programação do CLP.

### 4.1 Controlador Lógico Programável

O controlador lógico programável foi desenvolvido no final da década de sessenta como uma alternativa ao controle de dispositivos eletromecânicos que eram feitos através de relés. A proposta era melhorar a qualidade e a eficiência dos processos, reduzindo custos com reposição dos relés e aumentando a confiabilidade do sistema de controle. Nessa conjuntura, o CLP alavancou a automação industrial, tornando-se indispensável seu uso na indústria nas décadas que se seguiram (Georgini 2007). Por ser mais robusto que um computador pessoal, o CLP é capaz de suportar temperaturas mais altas, além de ser resistente a perturbações como ruído elétrico e vibrações, entre outras características. Quanto ao funcionamento, o CLP é capaz de funcionar com múltiplas combinações de entrada e saída.

#### 4.1.1 Operação

A operação de um CLP consiste em executar seu ciclo de varredura ou scan. A execução do ciclo de varredura corresponde à realização completa das seguintes atividades: atualização das entradas, execução do código do programa e atualização

das saídas. Na etapa de atualização de entradas o CLP realiza a leitura de todas as informações de entrada, associando essas informações a bits internos do CLP. Na etapa de execução do código do programa, o CLP executa as informações do código do programa. Estas informações são responsáveis por definir a relação entre a condição das informações de entrada e a atuação das informações de saída, que representam a lógica de controle que deve ser realizada pelo CLP.

## 4.2 Diagrama Ladder

A linguagem Ladder foi a primeira desenvolvida para programação de CLPs. Por ser uma linguagem gráfica, cujos símbolos se assemelham aos símbolos usados para representar esquemas elétricos, foi rapidamente assimilada e difundida. Apesar de ter sido a primeira linguagem desenvolvida para programação de CLPs, o diagrama Ladder é ainda hoje uma das linguagens mais utilizadas por engenheiros e técnicos da área de automação.

Segundo a norma IEC 1131-3 (IEC 1995), além da linguagem Ladder (Ladder Diagram - LD), outras quatro linguagens são definidas para programação de CLPs, sendo estas: diagrama de blocos funcionais (Function Block Diagram - FBD), lista de instrução (instruction list - IL), texto estruturado (structured text - ST) e diagrama funcional sequencial (sequential function chart - SFC).

O nome diagrama Ladder deve-se à sua representação gráfica semelhante à de uma escada, constituída por segmentos de reta paralelos entre si e conectados através de linhas horizontais, denominadas rungs, que representam a lógica de controle do diagrama Ladder. A linguagem Ladder foi sofisticada com o tempo, e hoje permite operações mais refinadas realizadas por comparadores, contadores e temporizadores, apenas para citar algumas.

### 4.2.1 Instruções Básicas

Para a compreensão do funcionamento da linguagem Ladder e, por conseguinte, para o entendimento do método de conversão a ser apresentado no capítulo seguinte, é necessário apresentar o conceito de corrente lógica fictícia, instruções Booleanas

de comparação, e temporização. As instruções apresentadas a seguir baseiam-se no CLP SLC 5/02 da Rockwell, utilizado no laboratório de Controle e Automação da Universidade Federal do Rio de Janeiro. Maiores informações sobre o CLP e as instruções não contidas neste projeto podem ser obtidas em Rockwell (2002).

#### 4.2.1.1 Corrente Lógica Fictícia

Em um diagrama Ladder a leitura e, por conseguinte, acionamento dos elementos é feita sempre em um único sentido, da esquerda para a direita. Na realidade supõe-se um modelo físico, onde a barra situada à esquerda representa o ponto de maior potencial, enquanto a barra à direita representa o ponto de menor potencial. Dessa maneira, assim como em um circuito elétrico, onde a corrente flui sempre do ponto de maior para o de menor potencial, a corrente lógica fictícia também obedece esse princípio e, portanto, flui sempre da barra da esquerda para a da direita, como pode ser observado na figura 4.1. Além disso, a ordem da leitura dos rungs no diagrama Ladder é sempre feita de cima para baixo.

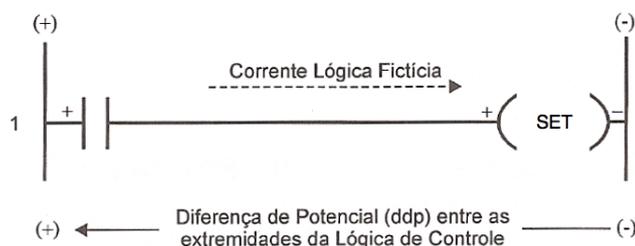


Figura 4.1: Sentido do fluxo de corrente fictícia.

#### 4.2.2 Contato Normalmente Aberto

A instrução de contato normalmente aberto (NA) é utilizada no Ladder para determinar se um bit está ligado, ou seja, se a variável binária associada à lógica do contato é 1. Se o bit endereçado estiver ligado (1) quando a instrução é executada, a lógica da instrução será avaliada como verdadeira. Caso o bit endereçado esteja desligado (0) quando a instrução for executada, a lógica da instrução será avaliada como falsa, como pode ser observado na tabela 4.1. A figura 4.2 ilustra um contato

NA.

Contato Normalmente Aberto - Estado do Bit	
Estado do Bit	Contato NA - Lógica da Instrução
0	Falsa
1	Verdadeira

Tabela 4.1: Lógica de instrução do contato NA.

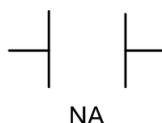


Figura 4.2: Representação de contato NA.

### 4.2.3 Contato Normalmente Fechado

A instrução de contato normalmente fechado (NF) é utilizada no Ladder para determinar se um bit está desligado, ou seja, se a variável binária associada à lógica do contato é 0. Se o bit endereçado estiver desligado (0) quando a instrução será executada, a lógica da instrução é então avaliada como verdadeira. Caso o bit endereçado esteja ligado (1) quando a instrução for executada, a lógica da instrução será avaliada como falsa. A tabela 4.2 ilustra a lógica de instrução de um bit associado a um contato NF, enquanto a figura 4.3 ilustra um contato NF.

Contato Normalmente Fechado - Estado do Bit	
Estado do Bit	Contato NF - Lógica da Instrução
0	Verdadeira
1	Falsa

Tabela 4.2: Lógica de instrução do contato NF.

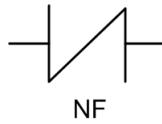


Figura 4.3: Representação de contato NF.

#### 4.2.4 Bobina

A instrução de bobina possui uma variável binária associada, que recebe valor 1 quando a lógica de seu rung é verdadeira. A bobina pode funcionar tanto como uma saída do controle como uma variável interna. A figura 4.4 ilustra a representação de uma bobina.

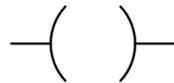


Figura 4.4: Representação da instrução de bobina.

#### 4.2.5 SET (OTL - Output Latch) e RESET (OTU - Output Unlatch)

As instruções *SET* (OTL) e *RST* (OTU) são instruções de retenção de informação de saída. A instrução *SET* é capaz de ligar um bit, enquanto a instrução *RST* é capaz de desligá-lo. Estas instruções são geralmente utilizadas em pares, com ambas endereçando o mesmo bit. A figura 4.5 ilustra as instruções de retenção *SET*, representada por *L*, e *RESET*, representada por *U*.

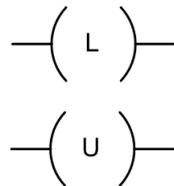


Figura 4.5: Representação de instruções SET (L) e RESET (U).

## 4.2.6 Disparo Único de Subida - OSR (One-Shot Rising)

A instrução de disparo único de subida (OSR) é utilizada quando se pretende identificar a mudança de falso para verdadeiro no nível lógico de um rung. Quando isto ocorre, a lógica da instrução de disparo único de subida (OSR) é verdadeira por um ciclo de varredura. Ao final deste ciclo de varredura a lógica da instrução OSR torna-se falsa, mesmo que a lógica do rung ao qual a instrução está associada permaneça verdadeira. Esta instrução OSR somente tornará a ser verdadeira caso a lógica do seu rung torne a mudar de falsa para verdadeira. A figura 4.6 ilustra a instrução OSR.

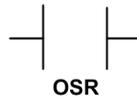


Figura 4.6: Instrução de disparo único de subida (OSR).

## 4.2.7 Instruções de Temporização

A instrução de temporização é utilizada quando se pretende associar um atraso ao disparo de uma transição. Um temporizador começa sua contagem quando a lógica do rung passa de falsa para verdadeira. Enquanto a lógica do rung permanecer verdadeira o valor acumulado da contagem do temporizador é acrescido até alcançar seu valor de *preset*. Uma vez que o valor de *preset* é alcançado, a bobina interna do temporizador é energizada. O temporizador utilizado neste trabalho é do tipo temporizador de atraso ligado (TON). Para o TON, caso a lógica do rung mude em algum momento de verdadeiro para falso, o valor acumulado da contagem é zerado. A figura 4.7 ilustra um temporizador de atraso ligado.

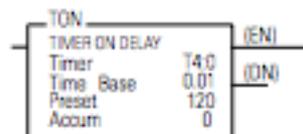


Figura 4.7: Temporizador de atraso ligado (TON).

## 4.2.8 Contadores

A instrução de contagem é utilizada para contar as mudanças de falso para verdadeiro na lógica dos rungs. Essas transições podem ser ocasionadas por eventos ocorrentes no programa, sejam eles da lógica interna ou de dispositivos do meio externo. Quando a lógica de um rung muda de falsa para verdadeira, o valor acumulado do contador é acrescido em uma unidade. Quando o valor de *preset* do contador é alcançado a bobina interna do contador é energizada. O contador utilizado neste trabalho é do tipo count-up (CTU). A figura 4.8 ilustra um contador do tipo CTU.

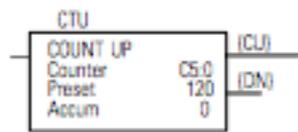


Figura 4.8: Contador crescente (CTU) .

### 4.2.8.1 Reset - RES

A instrução RES, não é equivalente à instrução RST (ou Unlatch), na verdade esta instrução é utilizada apenas para zerar o valor acumulado de algum contador ou temporizador, desde que estes tenham seu bit interno associado (endereçado) à instrução RES.

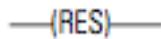


Figura 4.9: Instrução Reset (RES) de contadores e temporizadores.

## 4.3 Resumo

Neste capítulo foram apresentados o funcionamento e características dos CLPs e do diagrama Ladder. Foi discutido o funcionamento de um CLP e seu ciclo de varredura. Quanto ao diagrama Ladder foram introduzidos conceitos básicos para o entendimento de sua lógica de funcionamento, e os seus principais elementos. Como

foi comentado neste capítulo, as exposições sobre o diagrama Ladder foram baseadas no CLP SLC 5/02 da Rockwell. Informações mais extensas sobre os conceitos deste CLP, bem como informações gerais sobre CLP's e o diagrama Ladder podem ser encontradas em Rockwell (2002) e Georgini (2007), respectivamente.

No capítulo 5 serão apresentados os métodos de conversão propostos na literatura e seus problemas de implementação. Ainda neste capítulo será apresentado o método de conversão proposto neste trabalho, que garante uma conversão direta e simples.

# Capítulo 5

## Método de Conversão de RPIC para Diagrama Ladder

Neste capítulo é apresentado um método de conversão de Redes de Petri Interpretadas para Controle (RPIC), definidas no Capítulo 3, para o diagrama Ladder. O conteúdo deste capítulo é apresentado da seguinte forma: na seção 5.2 são apresentados os problemas relacionados a alguns dos métodos de conversão propostos na literatura. Na seção 5.3 é apresentado e discutido o método de conversão proposto neste projeto.

### 5.1 Método Usual Utilizado para a Conversão de Redes de Petri em Ladder

O métodos atualmente utilizados para a conversão de redes de Petri interpretadas em Ladder não se utilizam diretamente da equação de estado de uma rede de Petri ou de sua matriz de incidência. Nesses métodos cada rung representa uma transição  $t_j$ , que possui como condições para disparo seus lugares de entrada ( $I(t_j)$ ) e seu evento associado. No fim de cada rung são utilizadas as bobinas *RESET* e *SET* (considerando-se neste caso uma rede de Petri interpretada e segura), para remover a ficha de  $I(t_j)$  e adicionar uma ficha a  $O(t_j)$ , respectivamente. A figura 5.1(a) representa parte da estrutura de uma rede de Petri interpretada e segura, enquanto a figura 5.1(b) ilustra o diagrama Ladder da rede de Petri em que cada rung corresponde a uma transição da rede de Petri.

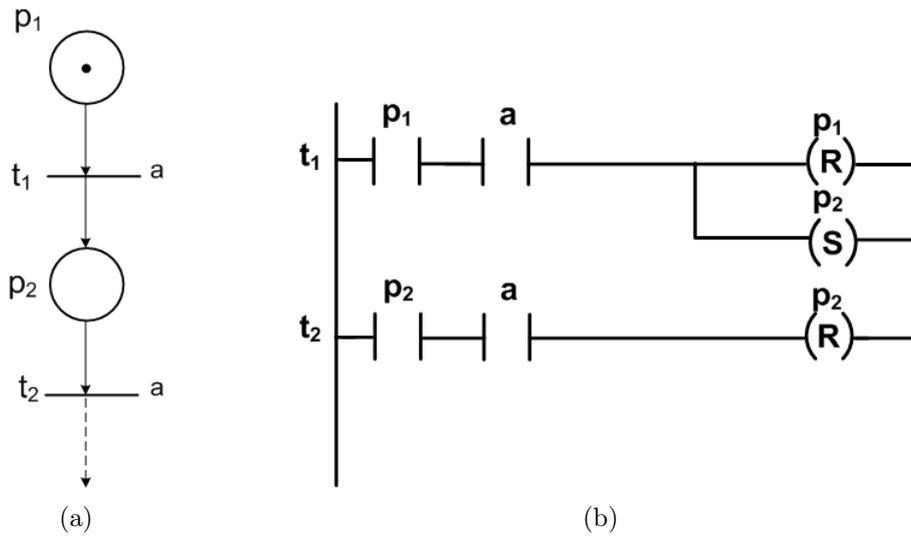


Figura 5.1: Parte da estrutura de uma rede de Petri interpretada e segura, com seu respectivo diagrama Ladder.

## 5.2 Problemas de Implementação

A implementação de sistemas de automação em Ladder pode apresentar alguns problemas, sendo os mais comuns o efeito avalanche e a necessidade de escolha quando há transições conflitantes. A implementação desses métodos torna-se ainda mais complicada para sistemas com muitos estados, pois o método não é bem estruturado.

### 5.2.1 Efeito Avalanche

O efeito avalanche é o resultado da transposição de duas ou mais transições, fazendo com que estados que deveriam ser alcançados na rede de Petri sejam ignorados no diagrama Ladder. Essa perda de informação deve-se ao fato de duas ou mais transições possuírem receptividades idênticas e estarem dispostas de maneira sequencial na rede de Petri. Em Fabian & Hellgren (1998) é proposta uma solução para este problema que consiste na reorganização dos rungs do diagrama Ladder na ordem reversa em que seriam normalmente colocados. O exemplo utilizado na figura 5.1(a) ilustra a situação em que duas transições  $t_1$  e  $t_2$  estão em sequência direta e possuem o mesmo evento associado ao seu disparo. Neste caso, quando a

transição  $t_1$  estiver habilitada e seu evento associado ocorrer,  $t_1$  dispara e habilita  $t_2$ . A estrutura do diagrama Ladder ilustrada na figura 5.1(b) resulta no disparo das duas transições  $t_1$  e  $t_2$  de uma só vez, fazendo com que o estado representado por  $p_2$  não seja alcançado.

A solução para este problema está na rearrumação dos rungs associados às transições  $t_1$  e  $t_2$ . No diagrama Ladder o rung da última transição ( $t_2$ ) é posicionado antes do rung da primeira transição ( $t_1$ ), impedindo que a transição  $t_2$  fique habilitada no mesmo instante do disparo da transição  $t_1$ , e evitando assim, o efeito avalanche. A figura 5.2 ilustra a técnica de rearrumação de rungs utilizada para solução do efeito avalanche apresentado na figura 5.1.

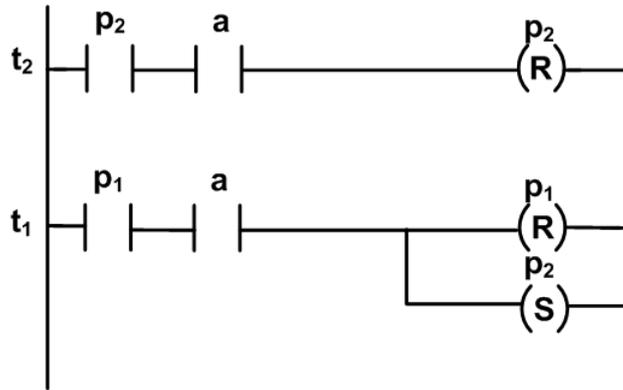


Figura 5.2: Rearrumação dos rungs de  $t_1$  e  $t_2$  para solução do efeito avalanche. Com o rung de  $t_2$  antes do rung de  $t_1$  é possível eliminar o efeito avalanche.

Contudo, esta técnica não é tão simples de ser implementada quando há a presença de um circuito entre as transições com mesmo evento associado ao seu disparo, como pode ser observado na figura 5.3(a). Neste caso é necessário a introdução da variável  $B1$ , cuja finalidade é quebrar o loop, impedindo o efeito avalanche, como pode ser observado na figura 5.3(b). O problema deste tipo de implementação está exatamente no posicionamento desta variável responsável pela quebra do loop, que em sistemas mais complexos torna-se consideravelmente mais difícil.

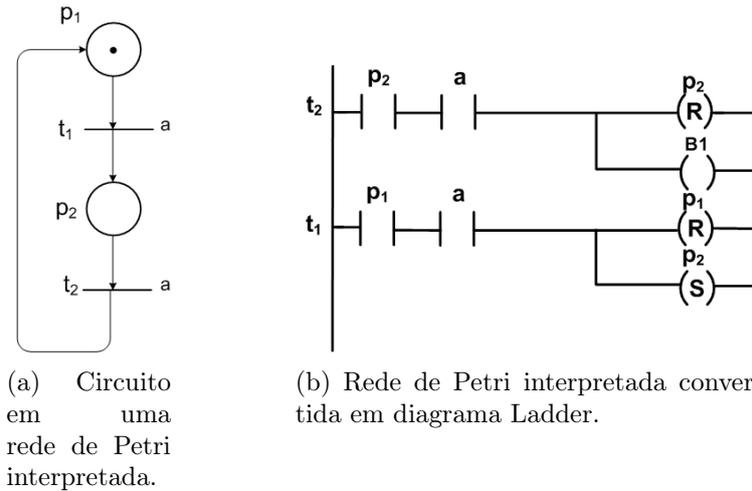


Figura 5.3: Solução de um circuito em uma rede de Petri interpretada e segura. Além da reorganização dos rungs é necessário o uso de uma variável responsável por quebrar o loop.

## 5.2.2 Resolução de Conflitos

Na modelagem de sistemas a eventos discretos, é comum a presença de conflitos. Em alguns trabalhos propostos na literatura a resolução do conflito é realizada através de priorização de transições. Isto ocorre porque, segundo a teoria de controle supervisorio, dois ou mais eventos não podem ocorrer ao mesmo tempo, enquanto que para um CLP isso é possível, podendo ocasionar conflitos no disparo de transições. Em Fabian & Hellgren (1998) é proposta uma solução para este problema, que consiste em ordenar os rungs das transições conflitantes segundo a prioridade de seus disparos. No diagrama Ladder, um rung com transição de maior prioridade é posicionado antes do rung de uma transição com menor prioridade. Com isso, o disparo das transições com maior prioridade desabilita o disparo das transições com menor prioridade, resolvendo o problema do conflito entre as transições na conversão da rede de Petri para o diagrama ladder. Essa escolha deve ser realizada a priori pelo programador, necessitando que todas as transições em conflito sejam reorganizadas antes da implementação no CLP.

Apesar deste método ser bem definido, uma maneira mais direta de incorporar as relações de prioridade entre as transições conflitantes no Ladder tornaria a con-

versão mais clara. Este é o propósito do método apresentado neste trabalho. Um exemplo é ilustrado na figura 5.4, que possui quatro transições  $t_1$ ,  $t_2$ ,  $t_3$  e  $t_4$  com seus respectivos eventos associados a seus disparos. Neste exemplo, pode-se perceber que há um conflito estrutural  $\langle p_1, \{t_1, t_2\} \rangle$  e  $\langle p_2, \{t_1, t_3, t_4\} \rangle$ . O conflito ocorre pois, apesar das transições possuírem eventos diferentes associados a seus disparos, com exceção de  $t_1$  e  $t_4$  que não estão em conflito entre si, existe a possibilidade de ocorrência de mais de um evento durante um ciclo de varredura do CLP. Com isso é necessário realizar uma priorização entre os disparos das transições conflitantes, para garantir o correto funcionamento do supervisor implementado no CLP. Neste caso a priorização foi feita da seguinte forma:  $t_2 < t_1$  e  $t_2 < t_4 < t_3$ . Desta forma o disparo de  $t_2$  independe do disparo das demais transições, sendo seu rung colocado na primeira linha do diagrama Ladder. Em seguida a transição  $t_1$  é colocada após o rung da transição  $t_2$ , pois tem prioridade sobre as transições  $t_3$  e  $t_4$ . A transição  $t_4$ , que tem prioridade sobre a transição  $t_3$ , é colocada na terceira linha e, por último, é colocada a transição  $t_3$ . A figura 5.4(b) ilustra o diagrama Ladder deste exemplo com a reorganização dos rungs. Para exemplificação do problema é suposto que os eventos  $E_1$ ,  $E_2$ ,  $E_3$  e  $E_4$  possam ocorrer em um mesmo ciclo de varredura.

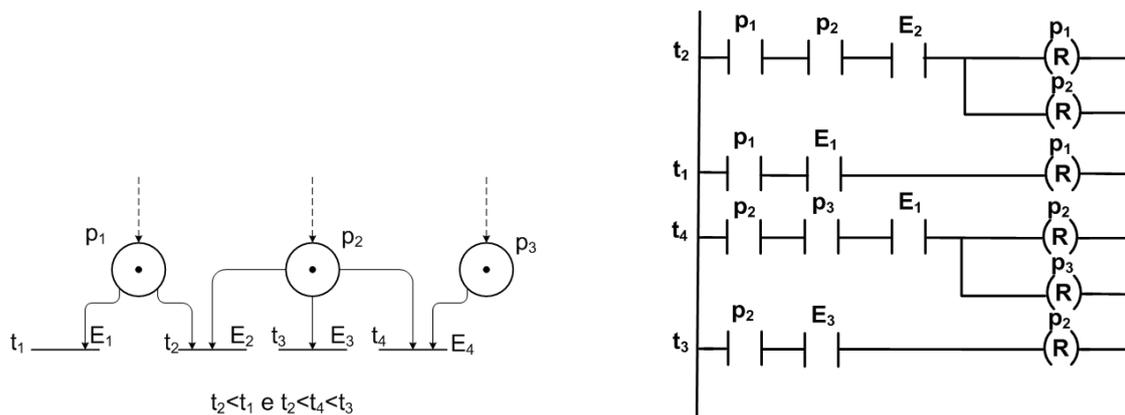


Figura 5.4: Solução de transições conflitantes através de priorização de seus disparos.

## 5.3 Método de Conversão

O método de conversão proposto neste projeto, e baseado em Moreira, Botelho & Basilio (2009), apresenta-se vantajoso sob dois aspectos principais, é consideravelmente simples, pois utiliza-se apenas da equação de estado e matrizes de incidência que provêm da estrutura da RPIC e evita a necessidade de reorganização dos rungs devido ao efeito avalanche ou conflitos entre transições, permitindo uma conversão direta entre uma RPIC e o diagrama Ladder. O método é ainda útil caso seja necessário realizar modificações no Ladder, pois ele também permite uma conversão direta do Ladder para uma RPIC equivalente. Logo, com o uso do método tem-se liberdade para modificar uma RPIC ou o diagrama Ladder sem a perda de informações na conversão entre a RPIC e o Ladder.

O método de conversão é dividido em cinco módulos que representam, cada um, uma parte da estrutura da RPIC utilizada para modelagem do supervisor.

1. O primeiro módulo é denominado módulo de inicialização.
2. O segundo módulo recebe o nome de módulo de eventos.
3. O terceiro módulo é o módulo de condições para disparo das transições.
4. O quarto módulo recebe o nome de módulo da dinâmica da RPIC.
5. O quinto e último módulo é denominado módulo das ações.

### 5.3.1 Módulo de Inicialização

O módulo de inicialização está associado à marcação inicial da RPIC. Sua função é a atribuição de fichas para os lugares inicialmente marcados. Este módulo é representado por um único rung que possui um contato normalmente fechado (NF) associado a um bit interno do CLP. No primeiro ciclo de varredura do CLP as bobinas associadas aos lugares inicialmente marcados na RPIC são energizadas, indicando que cada lugar possui uma ficha. Este rung é executado, como o próprio nome diz, apenas na inicialização do diagrama Ladder. O contato NF é associado a

um bit interno do CLP e será aberto após a execução do primeiro ciclo de varredura, impedindo assim que o módulo seja executado novamente no próximo ciclo. Em cada um dos lugares que deve receber uma ficha é utilizada a bobina *SET*.

### 5.3.2 Módulo de Eventos

O segundo módulo está associado à ocorrência de eventos. Eventos do meio externo são traduzidos por sensores em sinais de borda de subida ( $\uparrow$ ) ou descida ( $\downarrow$ ), e são responsáveis por mudanças na lógica de controle, pois funcionam como informação de entrada para o controlador. A figura 5.5 ilustra o funcionamento do módulo de eventos, onde há um contato NA associado ao sinal *E*, que é responsável por detectar os sinais de borda de subida ( $\uparrow E$ ), e um contato NF responsável por detectar os sinais de borda de descida ( $\downarrow E$ ) traduzidos pelos sensores. Neste caso é utilizada também a instrução OSR, que torna-se verdadeira ao identificar a mudança de falso para verdadeiro no nível lógico de um rung, e ao final do ciclo de varredura torna-se novamente falsa. No segundo rung deste módulo, a variável binária *B2* deve ser iniciada com valor 1, pois a instrução OSR deste rung está associada a um contato NF. Isto é necessário para evitar que no primeiro ciclo as instruções OSR associadas aos contatos NF tornem-se verdadeiras.

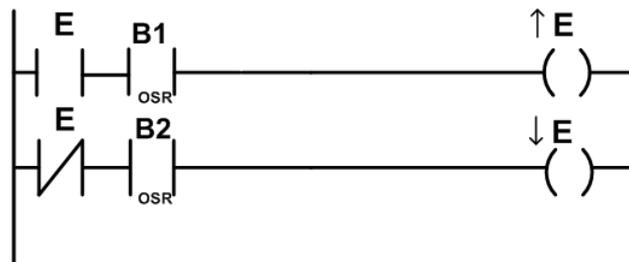


Figura 5.5: Módulo de eventos, cuja função é detectar mudanças na lógica de controle.

### 5.3.3 Módulo de Condições para o Disparo das Transições

O módulo de condições é responsável por descrever as condições necessárias para o disparo das transições. Estas condições são constituídas pelos lugares de

entrada das transições e suas respectivas receptividades. Os lugares de entrada das transições são determinados através da matriz de incidência  $W_{in}$ . Este módulo tem  $m$  rungs, que representa cada uma das condições das  $m$  transições presentes na RPIC.

Conforme foi comentado nos capítulos 2 e 3, uma transição  $t_j$  que não possui um lugar de entrada ligado por um arco inibidor, está habilitada se e somente se o número de fichas presente em cada um dos lugares de entrada desta transição  $t_j$  for igual ou maior que o peso dos arcos que conectam esses lugares à mesma. Caso exista um lugar de entrada conectado por um arco inibidor, esta transição  $t_j$  fica habilitada se e somente se o número de fichas presente neste lugar conectado por arco inibidor for menor que o peso do arco inibidor. Entretanto, para o disparo de uma transição  $t_j$  é necessário que a lógica de sua receptividade seja verdadeira, ou seja, a condição  $C_j$  associada à transição  $t_j$  deve ser verdadeira, e o evento  $E_j$  deve ocorrer.

As condições de habilitação e disparo são representadas no diagrama Ladder por uma série de contatos normalmente abertos (NA). No caso de uma transição  $t_j$  ter um lugar de entrada conectado à mesma através de um arco inibidor, utiliza-se um contato normalmente fechado associado a este lugar de entrada.

Todas as condições de uma transição  $t_j$  são dispostas em série em um único rung no diagrama Ladder, e para que a mesma possa disparar, a lógica do rung deve ser verdadeira. Ao final de cada rung é colocada uma bobina. Uma vez que a lógica do rung é verdadeira, a bobina é energizada, representando que a transição é disparável.

Para o caso de transições temporizadas, é adicionado um temporizador com valor de *preset* equivalente ao atraso associado à transição temporizada. O temporizador utilizado é o temporizador de atraso on delay. O temporizador possui uma bobina que é energizada quando o valor do acumulador atinge o valor de *preset*.

### 5.3.4 Módulo da Dinâmica da Rede de Petri

O módulo da dinâmica da rede de Petri está associado à atualização das fichas nos lugares de entrada e saída das transições. Uma nova marcação é obtida com o disparo das transições, e através da análise da matriz de incidência  $W$ , pode-se perceber que lugares recebem e/ou perdem fichas com o disparo dessas transições. Este módulo também possui  $m$  rungs, e cada rung está associado a uma transição que é responsável pela alteração na marcação dos lugares após seu disparo.

O módulo da dinâmica da rede de Petri é ainda responsável pela solução de transições conflitantes através de priorização de seus disparos. Esta solução é obtida através da inserção de contatos normalmente fechados aos rungs das transições conflitantes e com menor prioridade, como pode ser observado na figura 5.6. A figura 5.6 representa o diagrama Ladder da figura 5.4(a), mas desta vez sem a necessidade de reorganização dos rungs das transições conflitantes, garantindo uma conversão direta. Como a transição  $t_1$  só pode disparar caso a transição  $t_2$  não possa disparar ( $t_2 < t_1$ ), é inserido um contato NF associado a  $t_2$  no rung de  $t_1$ , garantindo a prioridade do disparo de  $t_2$  sobre  $t_1$ . O disparo da transição  $t_2$  independe dos disparos das demais transições e, portanto, não é inserido um contato NF em seu rung. A transição  $t_3$  (em conflito com  $t_2$  e  $t_4$ ) só pode disparar caso as transições  $t_2$  e  $t_4$  não possam disparar. Neste caso, são inseridos dois contatos NF, associados a  $t_2$  e  $t_4$ , garantindo a ordem do disparo da transição  $t_3$  que é dada por  $t_2 < t_4 < t_3$ . A transição  $t_4$  também está em conflito com  $t_2$  e  $t_3$ , mas com prioridade sobre  $t_3$ . Assim sendo é inserido apenas um contato NF, associado à transição  $t_2$ , para garantir que  $t_2$  tenha prioridade sobre  $t_4$ .

### 5.3.5 Módulo das Ações

O último módulo representa a execução das ações associadas a seus respectivos lugares, e por isso é chamado de módulo das ações. Este módulo associa uma bobina de saída aos lugares que tem uma ação de nível ou de impulso. Neste módulo também pode-se utilizar a instrução de disparo único de subida para os lugares que têm de realizar uma ação impulsional, ou seja, que acontecem somente uma vez por ciclo

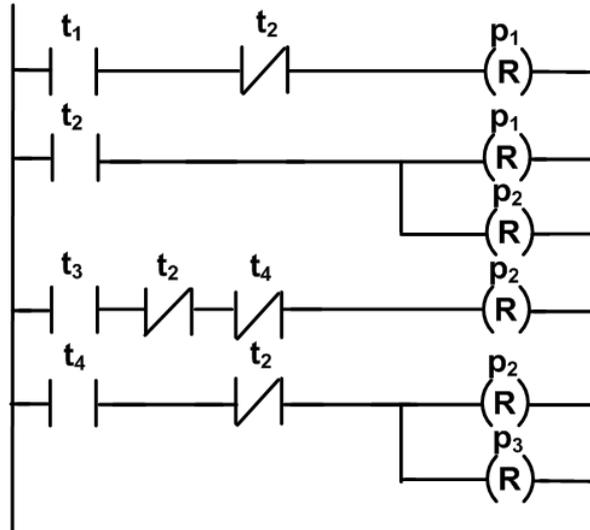


Figura 5.6: Solução de transições conflitantes através de priorização de seus disparos.

de varredura.

Cada lugar pode ser responsável por executar dois tipos de ações, de nível e de impulso. Se uma das ações associada a um lugar for impulsional, a mesma deve ser executada em apenas um ciclo de varredura. Neste projeto, assim como em Moreira, Botelho & Basilio (2009), uma ação de impulso é habilitada por apenas um ciclo de varredura, ainda que a ficha permaneça no mesmo lugar. Uma nova ocorrência da ação acontecerá apenas quando o número de fichas passar de zero para um. Para tal pode-se utilizar a instrução de disparo único de subida, pois a lógica desta instrução será verdadeira apenas por um ciclo de varredura quando a lógica de seu rung associado for verdadeira.

### 5.3.6 Determinação do Tamanho do Código Ladder

Um das vantagens do método utilizado reside na capacidade de determinação do tamanho do código Ladder através da estrutura da RPIC, antes mesmo da conversão da mesma em Ladder ou da implementação do código no CLP. O código pode ser obtido diretamente através da RPIC com a seguinte fórmula:  $1 + l + 2n + m_*$ . Nesta fórmula, 1 corresponde ao número de rungs do módulo de inicialização, que é sempre unitário,  $l$  corresponde ao número de rungs do módulo de eventos, em que

$l$  é o número de eventos pertencentes a  $E$  e associados às transições pertencentes a  $T$ . O número de rungs do módulo de condições para o disparo das transições é  $n$ , assim como o número de rungs do módulo da dinâmica da rede de Petri, por esta razão, o número total de rungs associados às transições da RPIC é  $2n$ . Finalmente, o número total de rungs associado ao módulo das ações é  $m_*$ , em que  $m$  é o número total de lugares da RPIC. Entretanto, como em uma RPIC nem todos os lugares possuem uma ação associada, é representado o número de rungs do módulo de ações como  $m_*$ , em que  $m_*$  é menor ou igual a  $m$ .

### 5.3.7 Eliminação do Efeito Avalanche

Como foi comentado anteriormente, este método é capaz de evitar o efeito avalanche. Isto ocorre pois os módulos são lidos sempre de maneira sequencial, com exceção do módulo de inicialização, que é lido apenas uma vez, com o módulo de condições para o disparo das transições precedendo o módulo da dinâmica da rede de Petri. Como a leitura dos módulos e a atualização das saídas no CLP dura sempre um ciclo de varredura, ainda que o módulo da dinâmica da rede de Petri dispare uma transição, removendo uma ficha de um lugar e acrescentando uma ficha a outro lugar, o que poderia habilitar uma transição subsequente a uma transição que acabou de disparar, esta transição subsequente só será reconhecida como uma transição habilitada para o disparo durante o próximo ciclo de varredura, quando o rung desta transição for lido pelo CLP no módulo de condições para o disparo das transições.

## 5.4 Resumo

Neste capítulo foram apresentados alguns dos problemas associados aos métodos propostos na literatura, como o efeito avalanche e o conflito entre transições. Apesar de haver soluções para esses problemas, sua implementação não é sempre simples e direta, exigindo ainda uma manipulação do diagrama Ladder. Em seguida foi apresentado o método de conversão deste trabalho e apresentadas as suas vantagens. Como o método se baseia na equação de estados e matrizes de incidência

de uma RPIC, o método de conversão é estruturado, funcionando para qualquer sistema a evento discreto, seja ele mais simples ou complexo. Ainda, com o uso deste método é possível determinar o tamanho do código Ladder, pois este depende somente do tamanho da RPIC. Deve-se ressaltar ainda que o método garante uma conversão direta entre uma RPIC e seu Ladder, e vice-versa.

No capítulo 6 será descrito um sistema de manufatura, sendo este modelado por três tipos diferentes de RPIC. Em seguida, o método de conversão será ilustrado através da obtenção do diagrama Ladder das três RPICs.

# Capítulo 6

## Exemplo

Neste capítulo uma célula de manufatura será utilizada para ilustrar o método de conversão proposto neste trabalho. Na seção 6.1 é realizada a descrição do sistema de manufatura e seu funcionamento para a realização da conversão da Rede de Petri Interpretada para Controle em diagrama Ladder. Serão utilizadas três formas de solução para o disparo de transições conflitantes: na seção 6.2.1 é discutida a solução através de receptividades mutuamente excludentes; na seção 6.2.2 é discutida a solução através de priorização no disparo das transições; finalmente na seção 6.2.3, o conflito entre transições é resolvido através do uso de arco inibidor.

### 6.1 Descrição do Sistema

O sistema utilizado é apresentado na figura 6.1, sendo composto por um braço robótico, duas esteiras mecânicas, uma região de espera e uma máquina. O sistema possui uma esteira em sua entrada, através da qual as peças chegam ao sistema. As peças podem ser de dois tipos: tipo 1 (com prioridade de serviço) ou tipo 2 (sem prioridade de serviço). As peças que chegam à esteira 1 movem-se sempre do ponto *A* para o ponto *B*. Não é permitida mais de uma peça por vez na esteira 1. Quando as peças chegam ao ponto *B* da esteira 1, ela é desligada, e toda vez que uma peça é removida da esteira 1, ela é religada. O braço robótico é responsável pelo transporte das peças através do sistema, pegando as peças no ponto *B* da esteira 1 e colocando-as no recipiente localizado no ponto *A* da esteira 2.

A esteira 2 também só permite uma peça por vez, e encontra-se desligada

quando uma peça chega ao recipiente localizado no ponto  $A$ . A chegada da peça neste recipiente, aciona a máquina para atuar na peça. Ao terminar sua operação a máquina é desligada, fazendo com que a esteira 2 seja acionada do ponto  $A$  para o ponto  $B$ . O recipiente contendo a peça é então movida em direção ao ponto  $B$ , sendo descarregada no fim da esteira 2. Ao final deste processo a esteira é desligada, sendo em seguida religada no sentido de  $B$  para  $A$ , fazendo com que seu recipiente volte para a posição  $A$ . Quando o recipiente retorna à posição  $A$ , a esteira é desligada novamente e fica à espera de uma nova peça.

Caso chegue uma peça do tipo 1 à esteira 1 e haja uma peça do tipo 2 sendo processada na esteira 2, a máquina para sua operação, o braço robótico move então a peça em processamento para a região de espera e, em seguida, coloca a peça do tipo 1 na esteira 2 para que seja processada pela máquina. Uma peça na região de espera tem prioridade sobre uma peça do tipo 2 que chegue à esteira 1. Uma peça do tipo 1 nunca é colocada na região de espera, e uma peça do tipo 1 na esteira 2 nunca tem seu serviço interrompido por uma peça, também do tipo 1, que eventualmente chegue à esteira 1.

## 6.2 Rede de Petri Interpretada para Controle

Neste projeto são propostas três formas de resolução de conflitos em uma RPIC: receptividade, priorização das transições, e através do uso de arcos inibidores. A utilização de arcos habilitadores não resolve o conflito, pois estes funcionam como os laços próprios. A utilização deste tipo de arco visa unicamente reduzir o número de arcos que teriam de ser representados caso fossem utilizados arcos-próprios, facilitando a visualização da dinâmica do sistema de automação.

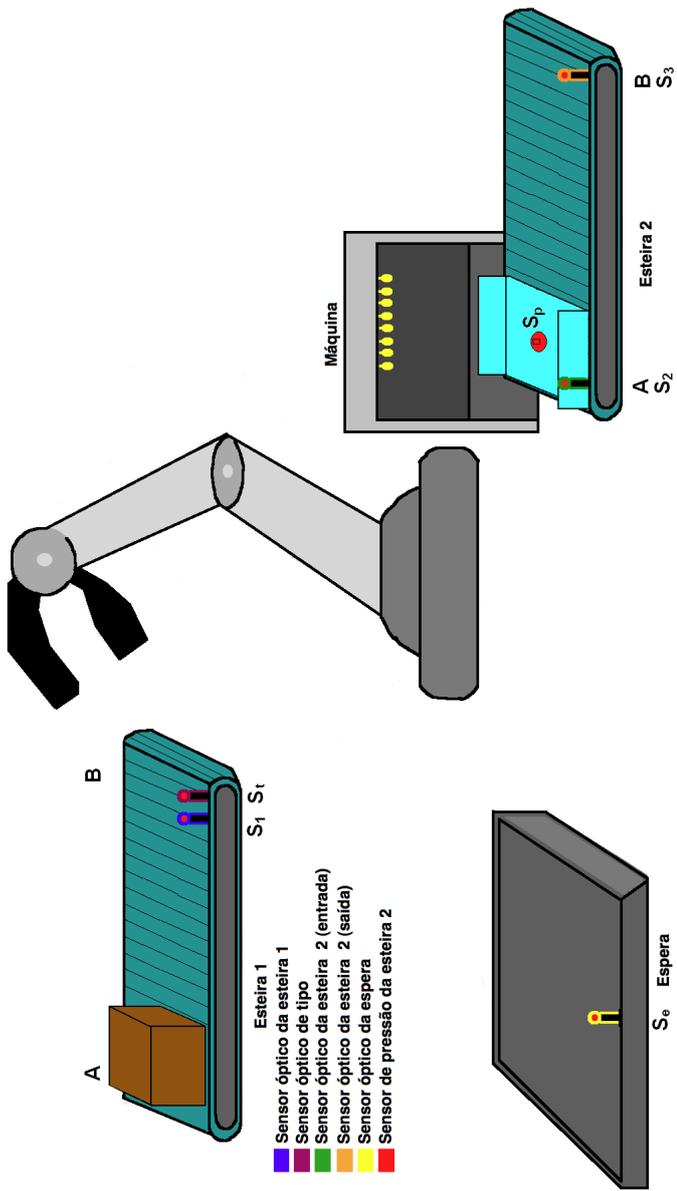


Figura 6.1: Ilustração do sistema de automação.

### 6.2.1 Solução através das Receptividades

A primeira modelagem do sistema de automação apresentada para resolver os conflitos é através de receptividades mutuamente excludentes associadas ao disparo das transições. A primeira RPIC fará uso dos 6 sensores colocados no sistema e ilustrados na figura 6.1. A tabela 6.1 fornece a identificação da nomenclatura utilizada pra identificar os sinais e sensores utilizados nos sistemas de automação.

Identificação dos Sensores e Sinais	
Sensores/Sinais	Identificação
$S_1$	Sensor óptico localizado no ponto B da esteira 1
$S_t$	Sensor de tipo localizado no ponto B da esteira 1
$S_e$	Sensor óptico localizado na região de espera
$E^1$	Sinal emitido pelo braço robótico ao fim de sua operação
$E^2$	Sinal emitido pela máquina ao fim de sua operação
$S_2$	Sensor óptico localizado no ponto A da esteira 2
$S_3$	Sensor óptico localizado no ponto B da esteira 2
$S_p$	Sensor de pressão localizado no recipiente situado em cima da esteira 2

Tabela 6.1: Identificação dos Sensores e Sinais utilizados na RPIC.

#### Esteira 1

A esteira 1 é responsável pela entrada das peças no sistema de automação. No final desta esteira, ponto  $B$ , são posicionados dois sensores, sendo o sensor  $S_1$  responsável pela detecção da chegada ou saída da peça do ponto  $B$ , enquanto o sensor  $S_t$  é responsável por detectar o tipo da peça que chega à esteira 1. Este último sensor atribui a uma peça prioritária valor 1, enquanto que a uma peça não prioritária é atribuído valor 0. Os sinais detectados por esses sensores são traduzidos para o sistema de automação, e atuam como informação de entrada para o CLP.

O lugar  $p_1$  representa o estado esteira ligada, sendo uma ação de nível associada a ele. Esta ação é responsável por ligar e manter ligada a esteira 1 enquanto houver uma ficha em  $p_1$ . No ponto  $B$  o sensor  $S_1$  é responsável pela detecção da peça, havendo uma mudança na interpretação do nível lógico do sensor pelo CLP, de 0 para 1. Esta variação é interpretada como a ocorrência do evento chegada da peça

ao ponto  $B$  da esteira 1 e funciona como um evento de borda de subida. Este evento está associado ao disparo da transição  $t_1$ . Logo, na ocorrência de  $\uparrow S_1$ , a transição  $t_1$  dispara, o lugar  $p_1$  perde sua ficha e o lugar  $p_2$  recebe uma ficha, sendo a esteira 1 desligada. Ainda no ponto  $B$  da esteira 1, o sensor  $S_t$  é responsável por detectar se a peça que chega é do tipo 1 ou do tipo 2 (este sensor funciona como uma condição para o sistema). Quando uma peça é retirada do ponto  $B$  da esteira 1 pelo braço robótico, há novamente uma mudança na interpretação do nível lógico do sensor  $S_1$  pelo CLP, desta vez de 1 para 0. Esta variação é interpretada como a ocorrência do evento saída da peça do ponto  $B$  da esteira 1 e funciona como um evento de borda de descida. Desta forma, na ocorrência de  $\downarrow S_1$ , transição  $t_2$  dispara, fazendo com que  $p_2$  perca sua ficha,  $p_1$  ganhe uma ficha e ligando novamente a esteira 1.

### **Verificação de Prioridade e Região de Espera**

Quando uma peça chegar à esteira 1, ela poderá ser do tipo 1 ou 2, sendo esta verificação de prioridade realizada pelo sensor  $S_t$  situado no ponto  $B$  dessa esteira. Como uma peça do tipo 1 tem prioridade sobre uma peça do tipo 2, seria necessário realizar a verificação de prioridade também na esteira 2. Entretanto, como não há um sensor para fazer esta verificação na esteira 2, são utilizados dois lugares a mais no sistema para indicar se há ou não uma peça com prioridade sendo processada pelo sistema. Os lugares utilizados são  $p_{13}$  e  $p_{14}$ , que correspondem, respectivamente, aos estados: sistema sem peça com prioridade e sistema com peça com prioridade. A condição de disparo da transição  $t_{17}$  é o sensor  $S_t = 1$  e seu evento associado é  $\downarrow S_1$ , ou seja, toda vez que o sensor  $S_t$  indicar que uma peça do tipo 1 estiver na esteira 1 e, em seguida, o braço robótico retirá-la, a transição  $t_{17}$  disparará. Esse disparo fará com que o lugar  $p_{13}$  perca sua ficha e  $p_{14}$  ganhe uma ficha, o que indicará que uma peça com prioridade está sendo processado pelo sistema. O evento associado ao disparo da transição  $t_{18}$  é  $\uparrow E^2$ , que é o sinal emitido pela máquina após processar a peça na esteira 2.

A região de espera, por sua vez, possui um sensor óptico que realiza a verifica-

ção da chegada ( $S_e = 1$ ) ou saída ( $S_e = 0$ ) de uma peça da região. A utilização do sensor  $S_e$  evita que sejam criados outros dois lugares para representar os estados: sistema sem peça na região de espera e sistema com peça na região de espera.

## **Esteira 2**

A esteira 2 possui três sensores, um no ponto  $A$  ( $S_2$ ), que detecta se o recipiente que fica na esteira 2 está na posição inicial, um no ponto  $B$  ( $S_3$ ), que detecta se o recipiente está na posição final e um sensor de pressão ( $S_p$ ), que fica no recipiente da esteira 2, sendo responsável por detectar o carregamento e descarregamento do recipiente.

O lugar  $p_7$  representa o estado esteira 2 desligada e o recipiente no ponto  $A$  da mesma. O disparo da transição  $t_{10}$  está associado ao evento  $\uparrow E^2$ , que indica que a máquina terminou de processar a peça. Na ocorrência deste evento, a transição  $t_{10}$  disparará, o lugar  $p_7$  perderá sua ficha e o lugar  $p_8$  ganhará uma ficha. O lugar  $p_8$  representa o estado esteira 2 ligada no sentido de  $A$  para  $B$ . Por isso, a este lugar é associado uma ação de nível que mantém a esteira ligada. A esteira será desligada quando a transição  $t_{11}$  disparar. Neste caso, uma ficha será removida de  $p_8$  e uma ficha será adicionada a  $p_9$ . O disparo da transição  $t_{11}$  está associada ao evento  $\downarrow S_p$ , que indica que a peça foi removida do recipiente da esteira 2. A transição  $t_{12}$  só disparará após transcorrido cinco segundos. Com o disparo de  $t_{12}$ , o lugar  $p_9$  perderá sua ficha e o lugar  $p_{10}$  ganhará uma ficha. O lugar  $p_{10}$  também possui uma ação de nível associada, que faz com que a esteira 2 seja ligada, desta vez, no sentido de  $B$  para  $A$ . Uma vez que o sensor óptico  $S_2$  detectar a chegada do recipiente novamente no ponto  $A$  desta esteira a transição  $t_{13}$  disparará. O disparo desta transição removerá uma ficha de  $p_{10}$  e desligará a esteira, o que acrescentará uma ficha novamente a  $p_7$ .

## Máquina

O lugar  $p_{11}$  representa o estado máquina desligada no sistema de automação. A máquina só será ligada quando a transição  $t_{14}$  disparar. Neste caso, o lugar  $p_{11}$  perderá sua ficha e o lugar  $p_{12}$  ganhará uma ficha. O lugar  $p_{12}$  possui uma ação de nível associada, que é responsável por ligar a máquina. O disparo da transição  $t_{14}$  está associado ao evento  $\uparrow S_p$ . A transição  $t_{15}$  só disparará quando a máquina emitir o sinal  $\uparrow E^2$ , que representa o fim de sua operação. Este sinal é sempre emitido ao final do processamento de uma peça pela máquina. Com o disparo da transição  $t_{15}$ , o lugar  $p_{12}$  perderá sua ficha e o lugar  $p_{11}$  ganhará novamente uma ficha. A transição  $t_{16}$  só ocorrerá se houver uma peça do tipo 1 ( $S_t = 1$ ) na esteira 1 e a peça que estiver sendo processada pela máquina na esteira 2 for do tipo 2, ou seja, o lugar  $p_{13}$  deverá estar marcado, indicando que o sistema não possui uma peça com prioridade sendo processada. Neste caso a máquina não terminará de processar a peça e, portanto, não emitirá o sinal  $\uparrow E^2$ . Isso significa que operação da máquina será interrompida para que a peça do tipo 2 seja retirada pelo braço e uma peça do tipo 1 seja colocada em seu lugar.

## Braço Robótico

As transições  $t_3$ ,  $t_5$ ,  $t_6$  e  $t_8$  estão inicialmente em conflito na estrutura da RPIC, como pode ser observado através da figura 6.2. Entretanto, este problema é solucionado através de suas receptividades, que são mutuamente excludentes, e estão representadas na tabela 6.2. As condições de habilitação da transição  $t_3$  são os lugares  $p_2$ ,  $p_3$  e  $p_7$ . O lugar  $p_2$  indica que é necessário a esteira 1 estar desligada, o lugar  $p_3$  indica que é necessário o braço robótico estar disponível e o lugar  $p_7$  indica a necessidade da esteira 2 estar desocupada e seu recipiente no ponto  $A$ . A transição  $t_5$  também tem como condições de habilitação os lugares  $p_2$ ,  $p_3$  e  $p_7$ . A transição  $t_6$  tem como condições de habilitação os lugares  $p_2$ ,  $p_3$ ,  $p_7$ ,  $p_{11}$  e  $p_{13}$ . O lugar  $p_{11}$  indica que a transição só estará habilitada se a máquina estiver desligada, enquanto  $p_{13}$  indica que a transição  $t_6$  só estará habilitada se não houver um objeto do tipo 1

em serviço. A transição  $t_8$  tem como condições de habilitação os lugares  $p_3$  e  $p_7$ .

O disparo das transições depende de suas respectivas receptividades, que no caso de  $t_3$  é dada por  $R_3 = \overline{S_e} \cdot \overline{S_p} \cdot \overline{S_t}$ , em que  $\overline{S_e}$  indica que não há objeto na região de espera (lógica Booleana 0),  $\overline{S_p}$  indica que não há objeto no recipiente da esteira 2 e  $\overline{S_t}$  indica que o objeto na esteira 1 não é do tipo 1. A receptividade de  $t_5$  é dada por  $R_5 = \overline{S_p} \cdot S_t$  (lógica 0 para  $S_p$  e 1 para  $S_t$ ). A receptividade de  $t_6$  é dada por  $R_6 = S_p \cdot S_t$  (lógica 1 para  $S_p$  e  $S_t$ ). A receptividade de  $t_8$  é dada por  $R_8 = S_e \cdot \overline{S_p} \cdot \overline{S_t}$  (lógica 1 para  $S_e$  e 0 para  $S_p$  e  $S_t$ ).

Como pode ser observado, as expressões das receptividades apresentadas são mutuamente excludentes, evitando o conflito real entre elas. O disparo de qualquer uma das transições  $t_3$ ,  $t_5$  ou  $t_6$  significa que o objeto é removido da esteira 1 ( $p_3$  perde a ficha e os demais lugares de entrada dessas transições, por terem arco habilitador não perdem fichas), e no sensor  $S_1$  ocorre uma mudança em seu nível lógico, de 1 para 0. Esta variação ( $\downarrow S_1$ ) é interpretada pelo CLP como a ocorrência do evento saída do objeto no ponto  $B$  da esteira 1, sendo responsável por disparar a transição  $t_2$  ( $p_2$  perde sua ficha e  $p_1$  ganha novamente uma ficha). O lugar  $p_1$  volta a ficar marcado, e a esteira 1 é novamente ligada pelo atuador associado a esse lugar. Todas as ações associadas aos lugares da RPIC estão representadas na tabela 6.3. O disparo das transições  $t_3$  ou  $t_5$  remove a ficha em  $p_3$  e marca o lugar  $p_4$ . O lugar  $p_4$  possui uma ação 2 associada (comando 1) que é responsável por mover o braço da posição original até a esteira 1, pegar o objeto parado na esteira 1, mover o braço até a esteira 2, deixar o objeto no recipiente da esteira 2 e voltar o braço para a posição original. Sempre que o braço fica novamente disponível um sinal de tarefa executada ( $E^1$ ) é enviado ao sistema de automação, indicando a disponibilidade do braço robótico.

O disparo das transições  $t_4$  e  $t_9$  depende do envio do sinal de tarefa concluída do braço robótico ( $\uparrow E^1$ ), e é responsável por marcar novamente o lugar  $p_3$ , sendo a ficha retirada de  $p_4$  (disparo de  $t_4$ ) ou  $p_6$  (disparo de  $t_9$ ). O disparo da transição  $t_6$  marca o lugar  $p_5$  e remove a ficha de  $p_3$ . O lugar  $p_5$  possui uma ação 3 associada (comando 2) que é responsável por mover o braço da posição original até a esteira 2, pegar o objeto do tipo 2 na esteira 2, mover o braço até a região de espera, colocar

o objeto do tipo 2 na região de espera e voltar o braço para a posição original. No final deste comando, quando o sinal  $\uparrow E^1$  é emitido para o sistema de automação, a transição  $t_7$  dispara, o lugar  $p_4$  é marcado e a ficha é retirada de  $p_5$ . O disparo da transição  $t_8$  remove a ficha de  $p_3$  e marca o lugar  $p_6$ . O lugar  $p_6$  possui uma ação 4 associada (comando 3) que é responsável por mover o braço da posição original até o recipiente de espera, pegar o objeto no recipiente de espera, mover o braço até a esteira 2, colocar o objeto no recipiente da esteira 2 e voltar o braço para a posição original.

As ações de nível são executadas durante todo o tempo de duração de um comando, como ligar uma esteira ou uma máquina. Nesses casos a ação deve ser mantida enquanto o comando estiver sendo executado. Diferentemente das ações de nível, as ações impulsivas são executadas apenas uma vez, não necessitando serem mantidas durante a execução do comando. Apenas para efeito de compreensão, as ações de nível seriam semelhantes a uma função degrau, enquanto as funções impulsivas seriam semelhantes a uma função impulsional.

A RPIC que representa o controle deste sistema é apresentada na figura 6.2. As receptividades, ações e estados são descritos nas tabelas 6.2 e 6.3, respectivamente.

Transições		
	Receptividade	Atraso
$t_1$	$R_1 = \uparrow S_1$	
$t_2$	$R_2 = \downarrow S_1$	
$t_3$	$R_3 = \overline{S_e} \cdot \overline{S_p} \cdot \overline{S_t}$	
$t_4$	$R_4 = \uparrow E^1$	
$t_5$	$R_5 = \overline{S_p} \cdot S_t$	
$t_6$	$R_6 = S_p \cdot S_t$	
$t_7$	$R_7 = \uparrow E^1$	
$t_8$	$R_8 = S_e \cdot \overline{S_p} \cdot \overline{S_t}$	
$t_9$	$R_9 = \uparrow E^1$	
$t_{10}$	$R_{10} = \uparrow E^2$	
$t_{11}$	$R_{11} = \downarrow S_p$	
$t_{12}$	$R_{12} = e$	$d_{12} = 5$
$t_{13}$	$R_{13} = \uparrow S_2$	
$t_{14}$	$R_{14} = \uparrow S_p$	
$t_{15}$	$R_{15} = \uparrow E^2$	
$t_{16}$	$R_{16} = S_p \cdot S_t$	
$t_{17}$	$R_{17} = \downarrow S_1 \cdot S_t$	
$t_{18}$	$R_{18} = \uparrow E^2$	

Tabela 6.2: Receptividades associadas às transições na RPIC.

Lugares		
	Estado	Ação
$p_1$		Ligar esteira 1 (Ação 1)
$p_2$	Esteira 1 desligada	
$p_3$	Braço disponível	
$p_4$		Realizar comando 1 (Ação 2)
$p_5$		Realizar comando 2 (Ação 3)
$p_6$		Realizar comando 3 (Ação 4)
$p_7$	Esteira 2 desligada	
$p_8$		Ligar esteira 2 de $A$ para $B$ (Ação 5)
$p_9$	Esteira 2 parada	
$p_{10}$		Ligar esteira 2 de $B$ para $A$ (Ação 6)
$p_{11}$	Máquina desligada	
$p_{12}$		Ligar máquina (Ação 7)
$p_{13}$	Ausência de objeto do tipo 1 em serviço	
$p_{14}$	Objeto do tipo 1 em serviço	

Tabela 6.3: Estados e ações associados aos lugares na RPIC.

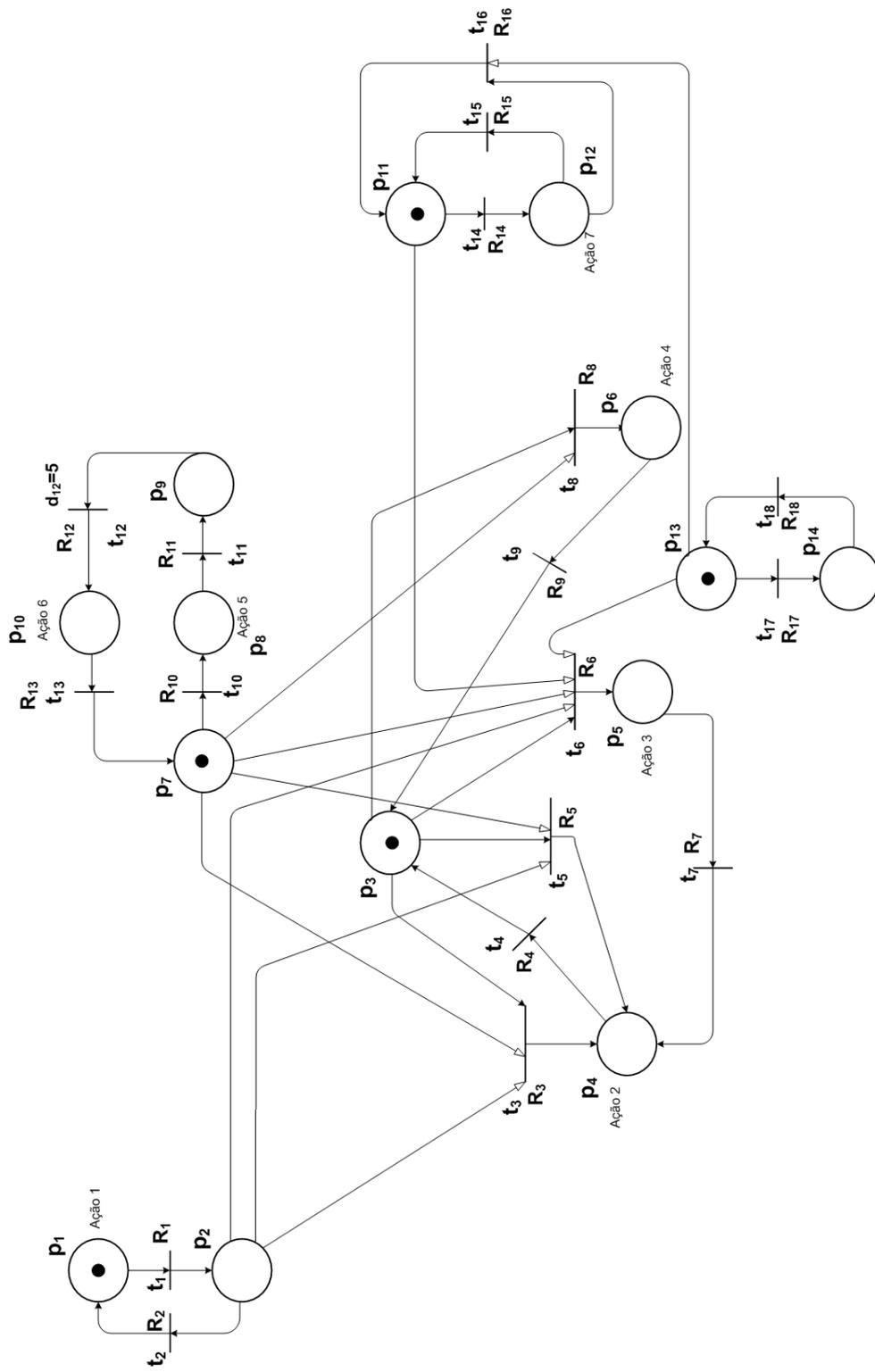


Figura 6.2: Rede de Petri com conflitos resolvidos através de receptividades mutuamente excludentes.

### 6.2.1.1 Conversão da RPIC em Diagrama Ladder

A primeira modelagem do sistema de automação foi através da RPIC apresentada na figura 6.2, que precisa, em seguida, ser implementada no CLP. Para que este sistema de automação seja implementado é necessário fazer a conversão da RPIC em diagrama Ladder.

#### Módulo de Inicialização

Na figura 6.2 a marcação inicial da Rede de Petri Interpretada para Controle é dada por  $\underline{x} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0]^T$ . Neste vetor os lugares inicialmente marcados são  $p_1, p_3, p_7, p_{11}$  e  $p_{13}$ . O primeiro rung do diagrama Ladder é formado pelo módulo de inicialização, sendo as bobinas associadas aos lugares inicialmente marcados energizadas. O módulo de inicialização é executado apenas uma vez, pois o bit interno associado ao contato NF no início do rung recebe valor 1, abrindo o contato e não permitindo que este módulo seja novamente executado no próximo ciclo de varredura, como pode ser visto na figura 6.3

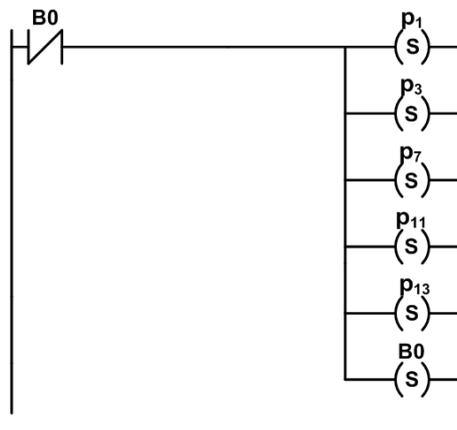


Figura 6.3: Módulo de inicialização da RPIC da figura 6.2.

## Módulo de Eventos

O conjunto de eventos externos é traduzido para o sistema por sinais captados por sensores e enviados como sinais de entrada à RPIC. Quando há uma mudança no nível lógico de um sensor, a variação é interpretada pelo CLP como a ocorrência de um evento, que pode ser de borda de subida ou descida, e tem a duração de apenas um ciclo de varredura. Assim sendo, a lógica dos bits internos associados aos rungs do módulo de eventos deve ser verdadeira apenas por um ciclo de varredura. Portanto, é utilizada a instrução OSR cujo bit interno do CLP associado é verdadeiro apenas quando a lógica do rung muda de falso para verdadeiro, tornando-se falsa ao final do ciclo. Para detectar a borda de subida de um sinal é utilizado um contato normalmente aberto antes do OSR, para que quando haja uma mudança no valor do sinal de 0 para 1, o bit associado ao OSR receba o valor 1. Para detectar a borda de descida de um sinal é utilizado um contato normalmente fechado antes do OSR, para que quando haja uma mudança no valor do sinal de 1 para 0, o bit associado ao OSR receba o valor 1 e energize a bobina ao final do rung. Além disso, nos casos de contatos normalmente fechados é necessário atribuir o valor 1 ao bit da instrução OSR, pois o CLP SLC 5/02 realiza uma mudança no contato NF de 1 para 0 em seu primeiro ciclo de varredura. A atribuição de valor 1 ao bit associado à instrução OSR evita que a bobina ao final do rung seja energizada logo no primeiro ciclo de varredura, mesmo que não tenha ocorrido mudança na lógica do sinal de 1 para 0.

O primeiro rung da figura 6.4 ilustra um contato normalmente aberto associado ao sensor  $S_1$ . Quando a lógica do contato muda de falsa (0) para verdadeira (1), ou seja, o contato muda de aberto para fechado, a instrução OSR torna-se verdadeira, energizando a bobina associada ao sinal  $\uparrow S_1$ . Esta bobina permanecerá energizada por um ciclo de varredura e indica a passagem de 0 para 1 no nível lógico de  $S_1$ . O segundo rung da figura 6.4 ilustra um contato normalmente fechado associado ao sensor  $S_1$ . Quando a lógica do contato muda de falsa (1) para verdadeira (0), ou seja, o contato muda de aberto para fechado, a instrução OSR torna-se verdadeira energizando a bobina associada ao sinal  $\downarrow S_1$ .

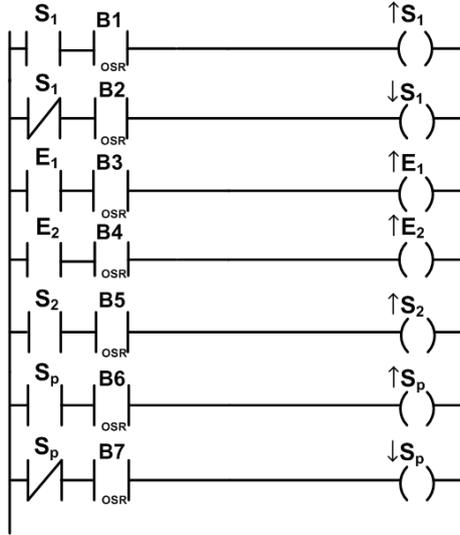


Figura 6.4: Módulo de eventos da RPIC da figura 6.2.

### Módulo de Condições para o Disparo das Transições

O módulo de condições representa o disparo das transições e pode ser obtido a partir da matriz de incidência de entrada  $W_{in}$ . As receptividades das transições da figura 6.2 estão representadas na tabela 6.2, e os lugares de entrada podem ser obtidos através da matriz de incidência  $W_{in}$  que é dada por:

$$W_{in}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A figura 6.5 representa o diagrama Ladder do módulo de condições para o disparo das transições, sendo composto pelas condições de habilitação das transições

da RPIC, e suas respectivas receptividades. Essas condições são responsáveis pelo disparo das transições, sendo as condições de habilitação representadas por contatos NA, e as condições de disparo são representadas por contatos NA ou NF, dependendo da lógica das condições de disparo associadas. Uma vez que estes contatos são acionados, a lógica do rung torna-se verdadeira e a bobina associada é energizada, tornando a transição disparável. Nas transições com atraso é utilizado um temporizador. Quando o valor o valor de *preset* do temporizador for alcançado, o temporizador será energizado, e a transição torna-se á disparável. O temporizador ilustrado na figura 6.5 não possui a mesma representação da instrução de temporização descrita na seção 4.2.7 e ilustrado na figura 4.7, pois a intenção deste diagrama ladder é apenas demonstrar o funcionamento do método. Como o bloco do temporizador depende do tipo de CLP, foi utilizada uma representação genérica.

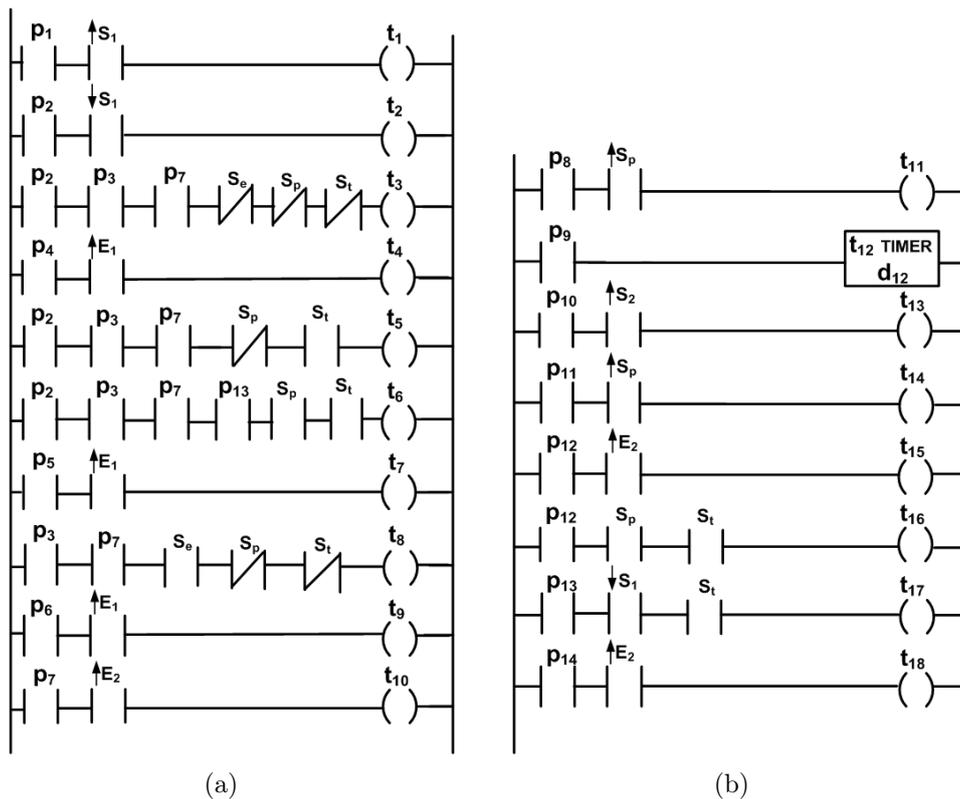


Figura 6.5: Módulo de condições para disparo das transições da figura 6.2 dividido em duas partes.

## Módulo da Dinâmica da Rede de Petri

O diagrama Ladder do módulo da dinâmica da rede de Petri é obtido através da matriz de incidência  $W$  que é dada por:

$$W^T = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

A figura 6.6 representa o diagrama Ladder do quarto módulo, onde estão contidas as condições de disparo das transições. Quando uma transição  $t_j$  disparar, os seus lugares de entrada  $I(t_j)$  perderão suas fichas e ao seu lugar de saída adicionar-se-á uma ficha. Como pode ser observado na figura 6.6, ao final de cada rung é adicionada uma instrução de *SET* (S) e *RESET* (R). A função destas instruções é exatamente remover a ficha do lugar de entrada (*RESET*) e adicionar a ficha ao lugar de saída (*SET*). Desta forma a dinâmica do sistema de automação modelado pela RPIC é representada no diagrama Ladder. Como pode ser observado na figura 6.6(b), a transição  $t_{12}$  tem um atraso associado ao seu disparo. Neste caso a transição só dispara quando o valor de *preset* do temporizador é alcançado.

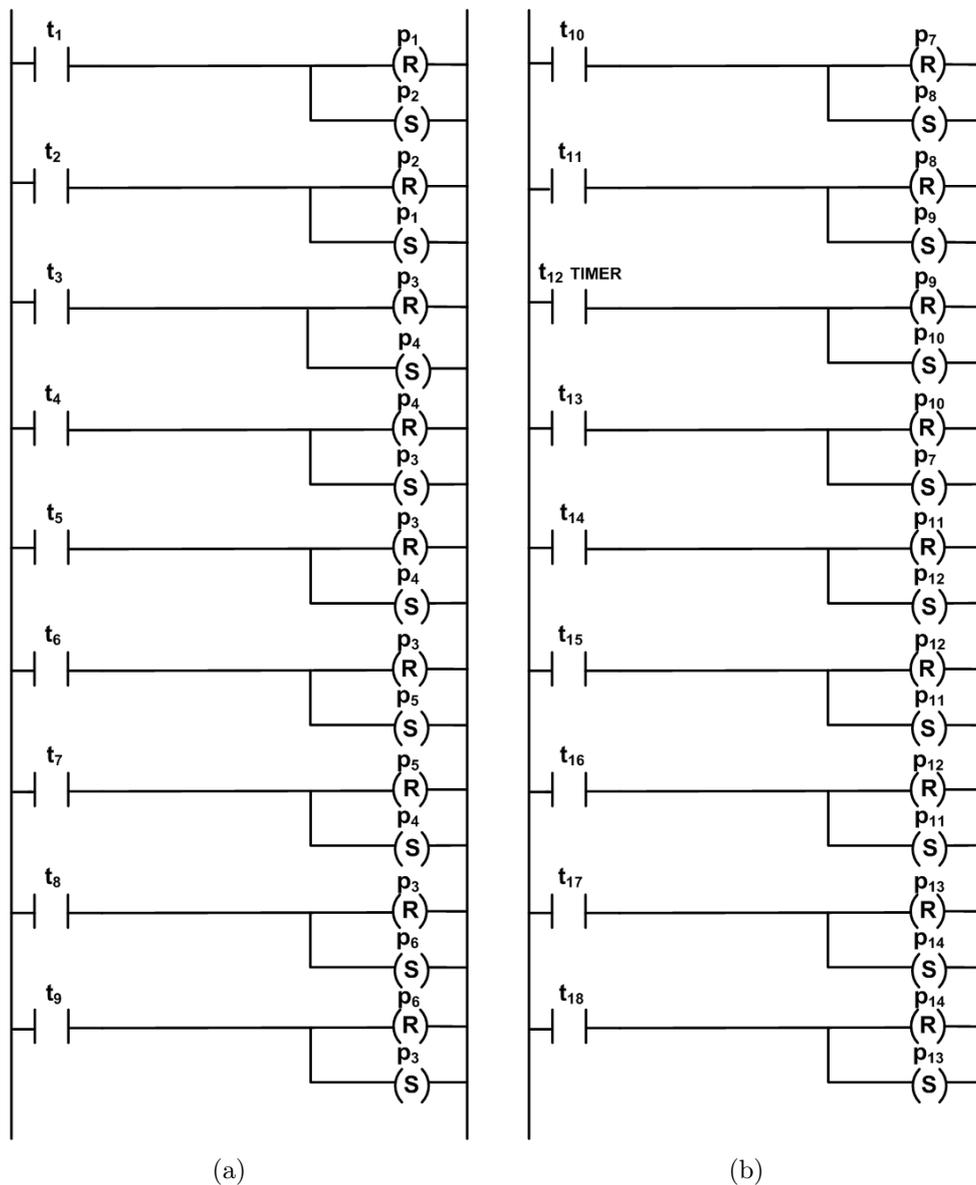


Figura 6.6: Módulo da dinâmica da RPIC da figura 6.2 dividido em duas partes.

### Módulo das Ações

O quinto e último módulo da conversão da RPIC em diagrama Ladder é o módulo das ações. Neste módulo estão contidas as informações para a realização das ações associadas aos lugares da RPIC. Na figura 6.7 é representado o módulo das ações para a RPIC da figura 6.2, cujas ações apresentadas estão descritas na tabela 6.3. As ações ligar a esteira 1, ligar a esteira 2 de *A* para *B*, ligar a esteira 2 de *B* para *A*, e ligar a máquina são todas ações de nível, pois uma vez acionadas

devem continuar em operação até que outra informação de entrada faça-as parar. Todas as ações de movimento do braço robótico são de impulso, pois o braço tem sua operação programada em seu sistema. Quando um lugar fica marcado na RPIC, seu contato passa de aberto para fechado, energizando a bobina associada à ação. No caso dos lugares associados a ações de impulso, é utilizado a instrução denominada OSR. Dessa forma, quando um lugar associado a um ação de impulso fica marcado, a instrução OSR torna-se verdadeira por um ciclo, energizando a bobina associada à ação de impulso.

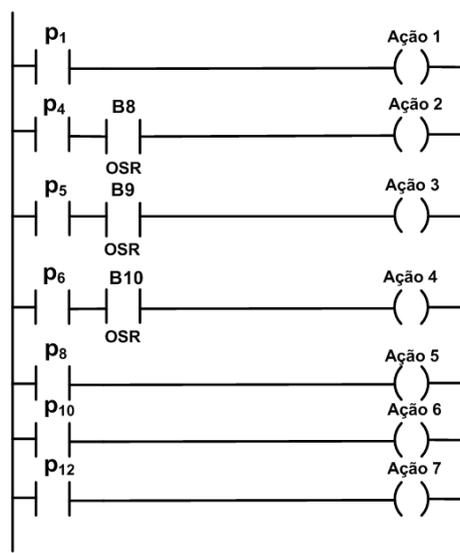


Figura 6.7: Módulo das ações da RPIC da figura 6.2.

## 6.2.2 Solução através da Priorização de Transições

Em alguns casos o conflito entre transições não pode ser resolvido escolhendo-se receptividades mutuamente excludentes. Nestes casos é necessário definir uma ordem de prioridade para o disparo das transições conflitantes. Ainda, dependendo do caso é possível resolver esses conflitos através do uso de arcos inibidores. Nesta seção, a solução para o conflito de transições é através da priorização do disparo das transições em conflito.

Nesta segunda modelagem para o sistema apresentado na figura 6.1, o sensor da região de espera é eliminado. Em muitos casos é possível eliminar sensores

sem perda de controlabilidade do sistema de automação. Entretanto, nesses casos é necessário aumentar o número de lugares da RPIC para garantir que o sistema armazene a informação da dinâmica do sistema, como pode ser observado na figura 6.8. Como não há o sensor de espera, o sistema tem de armazenar a informação do estado da região de espera.

Nesta segunda RPIC o funcionamento das esteira 1 e 2 é análogo ao funcionamento da RPIC utilizada na primeira modelagem, assim como o funcionamento da máquina. Apenas as transições  $t_3$ ,  $t_8$  sofrem alterações em seu disparo devido à ausência do sensor na região de espera. Por esse motivo é incluído o lugar  $p_{15}$  que indica a presença de um objeto na região de espera.

As condições de habilitação da transição  $t_3$  são os lugares de entrada  $p_2$ ,  $p_3$  e  $p_7$ , e seu disparo depende de sua receptividade, que é dada agora por  $R_3 = \overline{S_p} \cdot \overline{S_t}$ . O lugar  $p_2$  indica que a esteira 1 deve estar desligada e o objeto no ponto  $B$ , o lugar  $p_3$  indica que braço robótico deve estar disponível, e o lugar o lugar  $p_7$  indica que a esteira 2 deve estar desligada e seu recipiente na posição  $A$ . Já as condições para habilitação da transição  $t_8$  são os lugares de entrada  $p_3$ ,  $p_7$  e  $p_{15}$ , e seu disparo depende agora de  $R_8 = \overline{S_p} \cdot \overline{S_t}$ .

A transição  $t_4$  ainda tem como receptividade  $R_4 = \uparrow E^1$ , assim como as transições  $t_7$  e  $t_9$ , que tem suas respectivas receptividades,  $R_7$  e  $R_9$ , iguais a  $R_4$ . As condições para habilitação da transição  $t_5$  são os lugares de entrada  $p_2$ ,  $p_3$  e  $p_7$ , e seu disparo depende de  $R_5 = \overline{S_p} \cdot S_t$ . As condições para habilitação de  $t_6$  são os lugares de entrada  $p_2$ ,  $p_3$ ,  $p_7$ ,  $p_{11}$  e  $p_{13}$ , e seu disparo depende de  $R_6 = S_p \cdot S_t$ .

Nesta RPIC o uso de receptividades mutuamente excludentes não resolve todos os conflitos entre as transições  $t_3$ ,  $t_5$ ,  $t_6$  e  $t_8$ , pois as transições  $t_3$  e  $t_8$  possuem a mesma receptividade,  $R_3 = R_8 = \overline{S_p} \cdot \overline{S_t}$ , uma vez que não há mais o sensor  $S_e$ . Com isso é necessário criar uma ordem de prioridade para o disparo dessas duas transições. Como um objeto na região de espera tem prioridade sobre um objeto do tipo 2 é feito a seguinte priorização:  $t_8 < t_3$ . Com isso, a transição  $t_3$  só dispara se a transição  $t_8$  não puder disparar.

A transição  $t_{19}$  possui o evento que sempre ocorre como sua receptividade

( $R_{19} = e$ ) e, portanto, não há conflito com a transição  $t_7$ , pois esta depende do evento  $\uparrow E^1$ . A transição  $t_{20}$  possui a mesma receptividade de  $t_{19}$ ,  $R_{20} = R_{19} = e$  e, por isso também não há conflito com a transição  $t_9$ , que também depende de  $\uparrow E^1$ .

Transições		
	Receptividade	Atraso
$t_1$	$R_1 = \uparrow S_1$	
$t_2$	$R_2 = \downarrow S_1$	
$t_3$	$R_3 = \cdot \bar{S}_p \cdot \bar{S}_t$	
$t_4$	$R_4 = \uparrow E^1$	
$t_5$	$R_5 = \bar{S}_p \cdot S_t$	
$t_6$	$R_6 = S_p \cdot S_t$	
$t_7$	$R_7 = \uparrow E^1$	
$t_8$	$R_8 = \bar{S}_p \cdot \bar{S}_t$	
$t_9$	$R_9 = \uparrow E^1$	
$t_{10}$	$R_{10} = \uparrow E^2$	
$t_{11}$	$R_{11} = \downarrow S_p$	
$t_{12}$	$R_{12} = e$	$d_{12} = 5$
$t_{13}$	$R_{13} = \uparrow S_2$	
$t_{14}$	$R_{14} = \uparrow S_p$	
$t_{15}$	$R_{15} = \uparrow E^2$	
$t_{16}$	$R_{16} = S_p \cdot S_t$	
$t_{17}$	$R_{17} = \downarrow S_1 \cdot S_t$	
$t_{18}$	$R_{18} = \uparrow E^2$	
$t_{19}$	$R_{19} = e$	
$t_{20}$	$R_{20} = e$	

Tabela 6.4: Receptividades associadas às transições na RPIC.

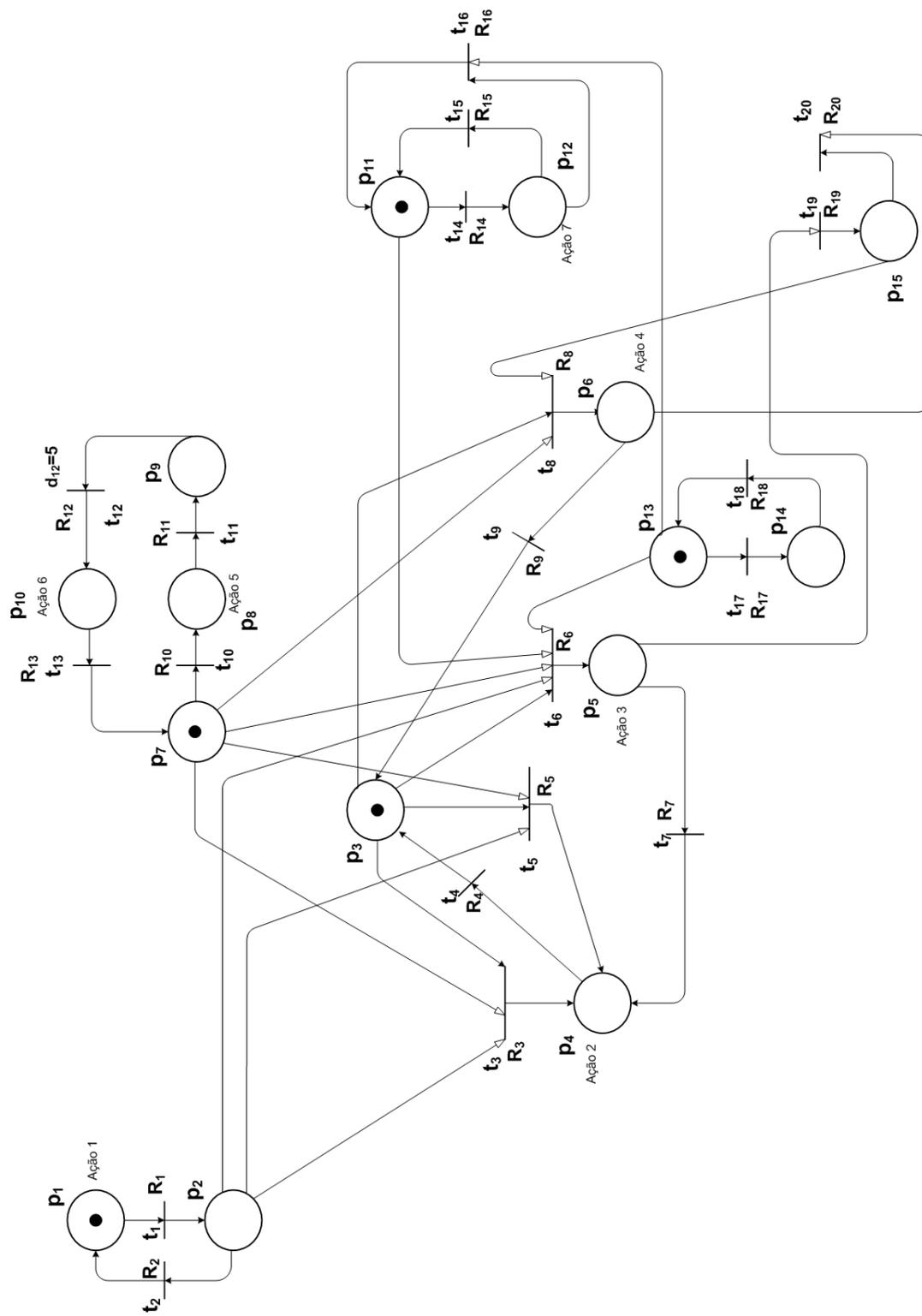


Figura 6.8: Rede de Petri com conflitos resolvidos através de priorização do disparo de transições.

Lugares		
	Estado	Ação
$p_1$		Ligar esteira 1 (Ação 1)
$p_2$	Esteira 1 desligada	
$p_3$	Braço disponível	
$p_4$		Realizar comando 1 (Ação 2)
$p_5$		Realizar comando 2 (Ação 3)
$p_6$		Realizar comando 3 (Ação 4)
$p_7$	Esteira 2 desligada	
$p_8$		Ligar esteira 2 de $A$ para $B$ (Ação 5)
$p_9$	Esteira 2 parada	
$p_{10}$		Ligar esteira 2 de $B$ para $A$ (Ação 6)
$p_{11}$	Máquina desligada	
$p_{12}$		Ligar máquina (Ação 7)
$p_{13}$	Ausência de objeto do tipo 1 em serviço	
$p_{14}$	Objeto do tipo 1 em serviço	
$p_{15}$	Sistema com objeto na região de espera	

Tabela 6.5: Estados e ações associados aos lugares na RPIC.

### 6.2.2.1 Conversão da RPIC em Diagrama Ladder

A segunda modelagem do sistema de automação foi através da RPIC, sendo utilizado uma ordem de priorização no disparo das transições, o que resulta em uma estrutura com um número maior de lugares, como pode ser visto na figura 6.8. Esse maior número de lugares também resulta em um diagrama Ladder com maior número de rungs.

#### Módulo de Inicialização

Na figura 6.8 a marcação inicial da Rede de Petri Interpretada para Controle é dada por  $\underline{x} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T$ . Neste vetor os lugares inicialmente marcados são  $p_1$ ,  $p_3$ ,  $p_7$ ,  $p_{11}$  e  $p_{13}$ . Esses lugares marcados tem suas bobinas associadas energizadas. A figura 6.9 ilustra esse diagrama Ladder.

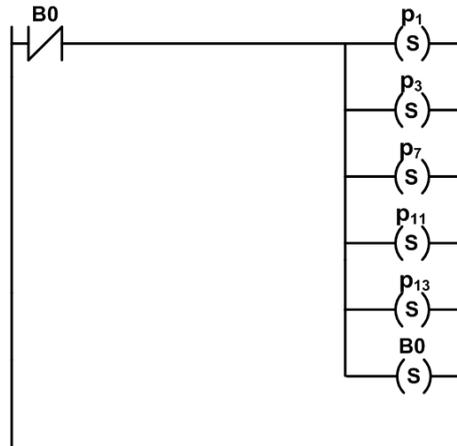


Figura 6.9: Módulo de inicialização da RPIC da figura 6.8.

### Módulo de Eventos

Os eventos relacionados a borda de subida ou descida de sinais podem ser encontrados na tabela 6.4, que corresponde à tabela de receptividades da RPIC da figura 6.8. Nesta tabela pode ser observado que não há mais nenhuma condição associada ao sensor  $S_e$ , pois a ausência deste sensor não altera a controlabilidade do sistema. Contudo, a representação do diagrama Ladder do módulo de eventos deste sistema de automação é ainda a mesma do anterior, pois o sensor  $S_e$  funcionava apenas como um condição, não sendo eliminado nenhum sinal de borda de subida ou descida do sistema de automação. Este módulo está ilustrado na figura 6.4.

### Módulo de Condições para o Disparo das Transições

As receptividades das transições da figura 6.8 podem ser obtidas através da tabela 6.4, e os lugares de entrada através da matriz de incidência  $W_{in}$ . A figura 6.10 representa o diagrama Ladder do terceiro módulo. Nesta figura estão representados os lugares de entrada e receptividades. Como pode ser observado na figura 6.13(b), os dois últimos rungs não possuem receptividades associadas. Isto ocorre pois as transições associadas a estes rungs tem a lógica de suas receptividades sempre verdadeira ( $R_j = 1$ ), ou seja, a condição  $C_j$  é sempre verdadeira e o evento  $E_j$  é o evento que sempre ocorre.

$$W_{in}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

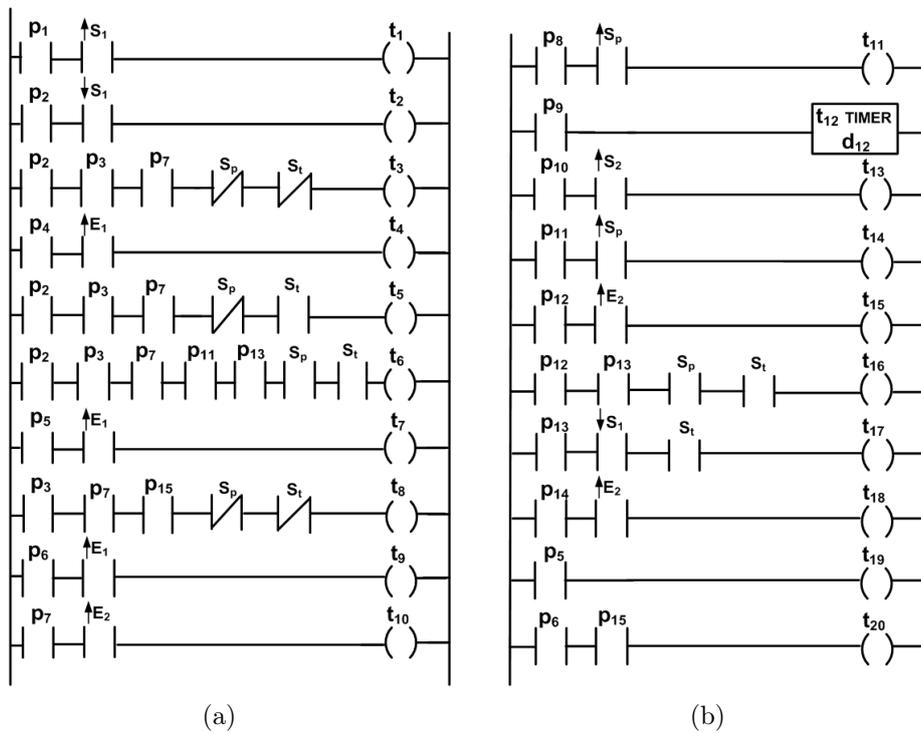


Figura 6.10: Módulo de condições para disparo das transições da RPIC da figura 6.8 dividido em duas partes.

## Módulo da Dinâmica da Rede de Petri

O diagrama Ladder do módulo da dinâmica da rede de Petri é obtido através da matriz de incidência  $W$  que é dada por:

$$W^T = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

A figura 6.11 representa o diagrama Ladder do quarto módulo. É exatamente no módulo da dinâmica da rede de Petri que é feita a ordem de priorização do disparo das transições  $t_3$  e  $t_8$ . Segundo o sistema descrito no início deste capítulo, um objeto na região de espera tem prioridade ( $t_8$ ) sobre um objeto do tipo 2 que chegue à esteira 1 ( $t_3$ ). Dessa forma é realizada a seguinte priorização:  $t_8 < t_3$ . Assim sendo, é adicionado um contato NF associado a  $t_8$  ao rung da transição  $t_3$ , garantindo que a transição  $t_3$  só possa disparar caso a transição  $t_8$  não seja disparável.

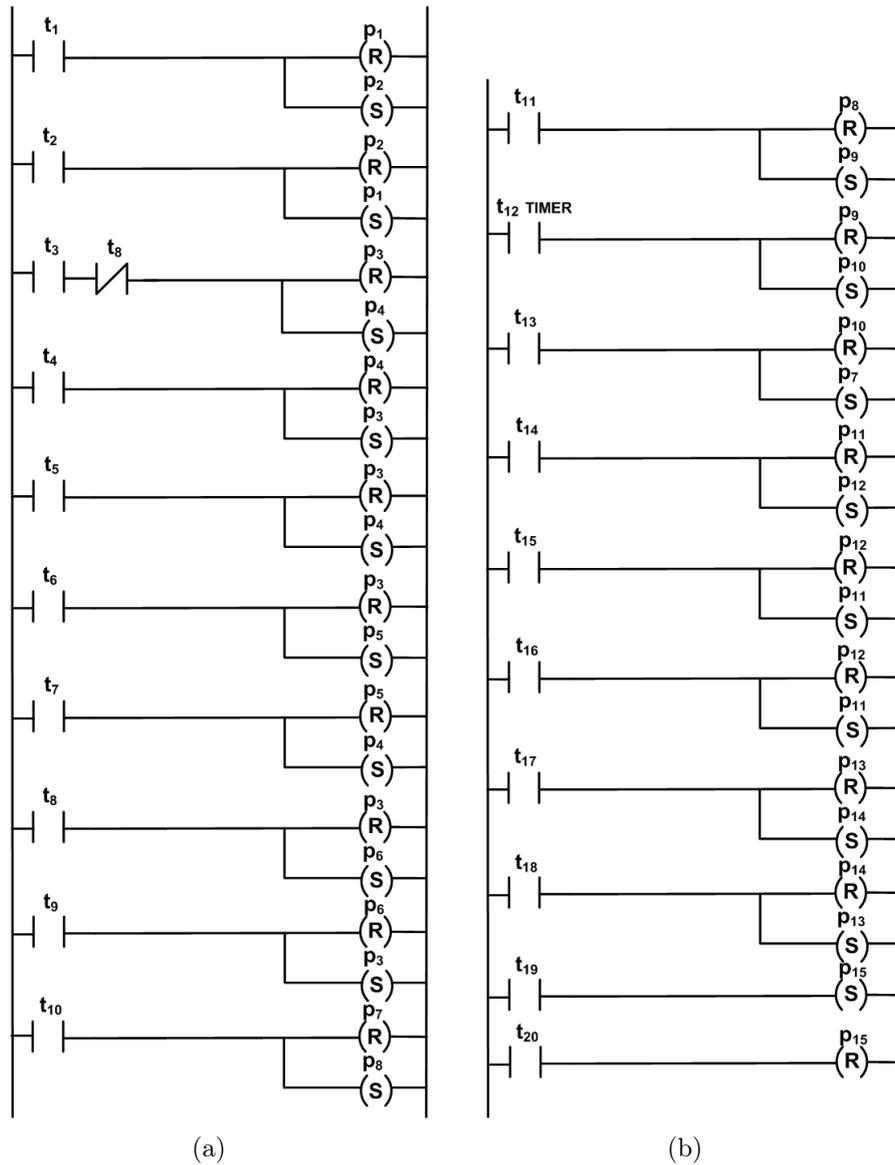


Figura 6.11: Módulo da dinâmica da RPIC da figura 6.8 dividido em duas partes.

### Módulo das Ações

O diagrama Ladder do módulo das ações desta segunda RPIC é semelhante ao módulo das ações da primeira RPIC, pois nenhuma nova ação é adicionada ou tem seu funcionamento alterado. Este módulo é representado na figura 6.7.

### 6.2.3 Solução através do Uso de Arco Inibidor

Nesta última modelagem a solução dos conflitos é feita através do uso do arco inibidor. Na modelagem com solução de conflitos através do uso do arco inibidor também é necessário utilizar um maior número de lugares para que o CLP tenha a informação mais precisa do que acontece no sistema, como pode ser visto na figura 6.12. Nesta modelagem, assim como na anterior, não é utilizado o sensor da região de espera.

A única alteração nesta modelagem do sistema de automação se dá nas condições para habilitação das transições  $t_3$  e  $t_8$  que se encontram em conflito. Neste caso o arco inibidor funciona como uma condição de habilitação para  $t_3$ , garantindo que a transição  $t_3$  não fique habilitada caso a transição  $t_{18}$  esteja habilitada. As tabelas de estados e ações, das receptividades associadas às transições são idênticas às tabelas da segunda modelagem do sistema de automação, tabelas 6.5 e 6.4, respectivamente.

#### 6.2.3.1 Conversão da RPIC em Diagrama Ladder

No capítulo 3 foi discutida a diferença entre a solução através do uso de arcos inibidores e a priorização do disparo de transições conflitantes em uma RPIC, e ilustrada na figura 3.5. Já nesta parte é importante ressaltar a diferença entre as duas soluções com relação à obtenção do diagrama Ladder. O arco inibidor atua no módulo de condições para disparo das transições, através do uso de contatos NF para indicar os lugares de entrada de uma transição conectados por arcos inibidores. Já a priorização de transições atua no módulo da dinâmica da rede de Petri, que é o módulo responsável pelo disparo das transições e evolução da dinâmica do sistema. Neste caso são utilizados contatos NF associados às transições de maior prioridade nos rungs das transições de menor prioridade, impedindo que uma transição de menor prioridade possa disparar caso uma transição de maior prioridade seja disparável.

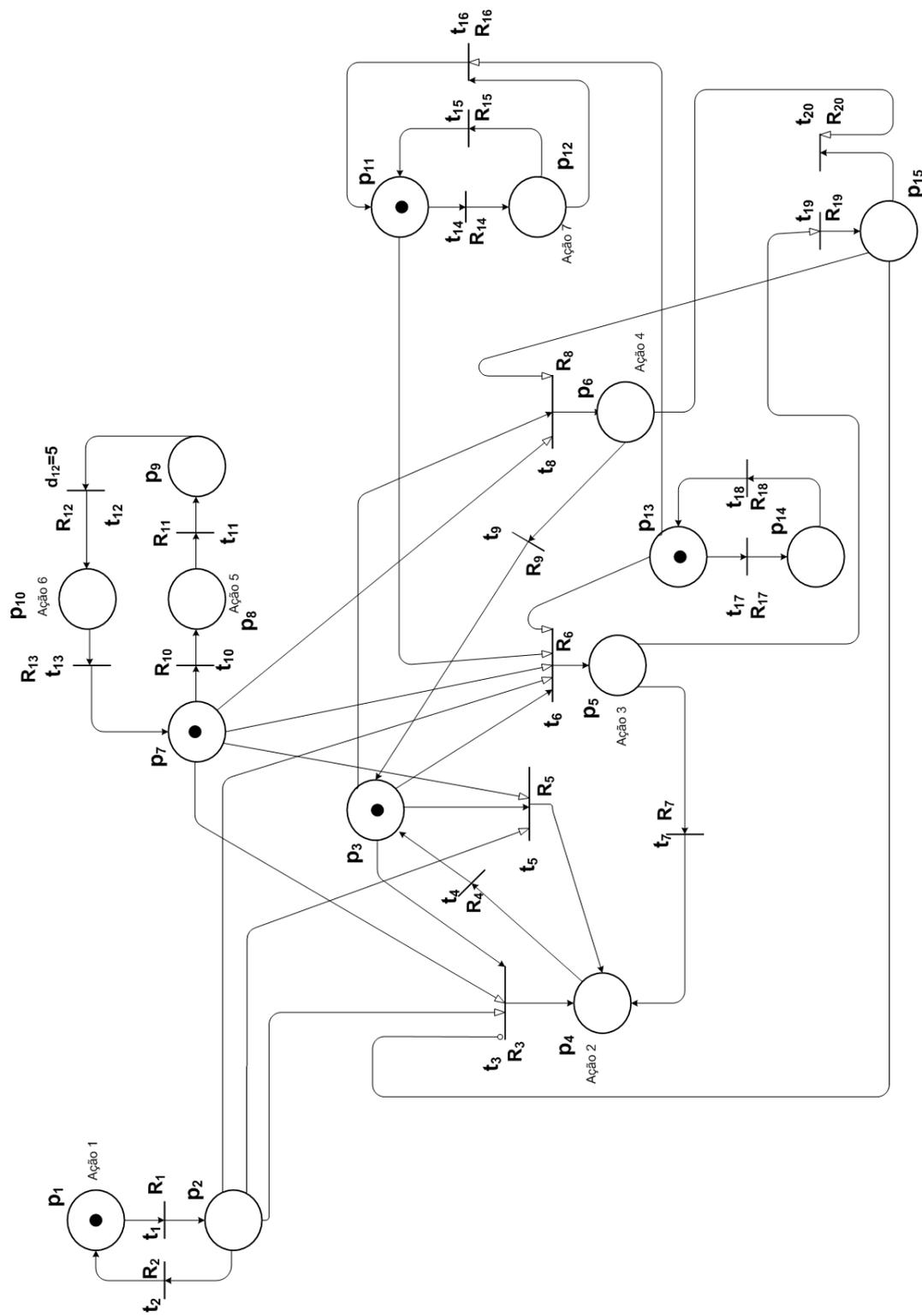


Figura 6.12: Rede de Petri com conflitos resolvidos através do uso de arcos inibidores.

### Módulo de Inicialização

A RPIC representada na figura 6.12 possui a mesma marcação inicial da segunda RPIC, representada anteriormente na figura 6.8, e, portanto, têm o mesmo módulo de inicialização.

### Módulo de Eventos

Na última modelagem também não há mais nenhum evento associado ao sensor  $S_e$ , assim como na segunda modelagem do sistema de automação. O diagrama Ladder do módulo de eventos da terceira modelagem é idêntico aos demais módulos, e ilustrado na figura 6.4.

### Módulo de Condições para o Disparo das Transições

Todas as receptividades das transições da RPIC da figura 6.12 são as mesmas da segunda modelagem, e estão representadas na tabela 6.4. Os lugares de entrada são obtidos a partir da matriz de incidência  $W_{in}$ :

$$W_{in}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{1} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Os arcos inibidores são representados por na matriz de incidência por  $\bar{1}$ , e funcionam apenas como uma condição para a habilitação das transições, não removendo fichas dos lugares que conectam. Neste caso utilizou-se o arco inibidor somente na transição  $t_3$ , pois o único conflito real é entre as transições  $t_3$  e  $t_8$ , sendo que  $t_8$  tem prioridade sobre  $t_3$ . Como o lugar  $p_{15}$  serve como condição para habilitação de  $t_8$  através de um arco habilitador, o uso do arco inibidor conectando  $p_{15}$  a  $t_3$  garante que as transições  $t_3$  e  $t_8$  não ficarão habilitadas ao mesmo tempo. Este arco inibidor pode ser observado no elemento  $w_{3,15}$  de  $W_{in}^T$ .

A figura 6.13 representa o diagrama Ladder do módulo de condições para o disparo das transições. Neste módulo pode-se perceber a presença de contatos NF nos rungs das transições que possuem um lugar de entrada conectado através de arco inibidor.

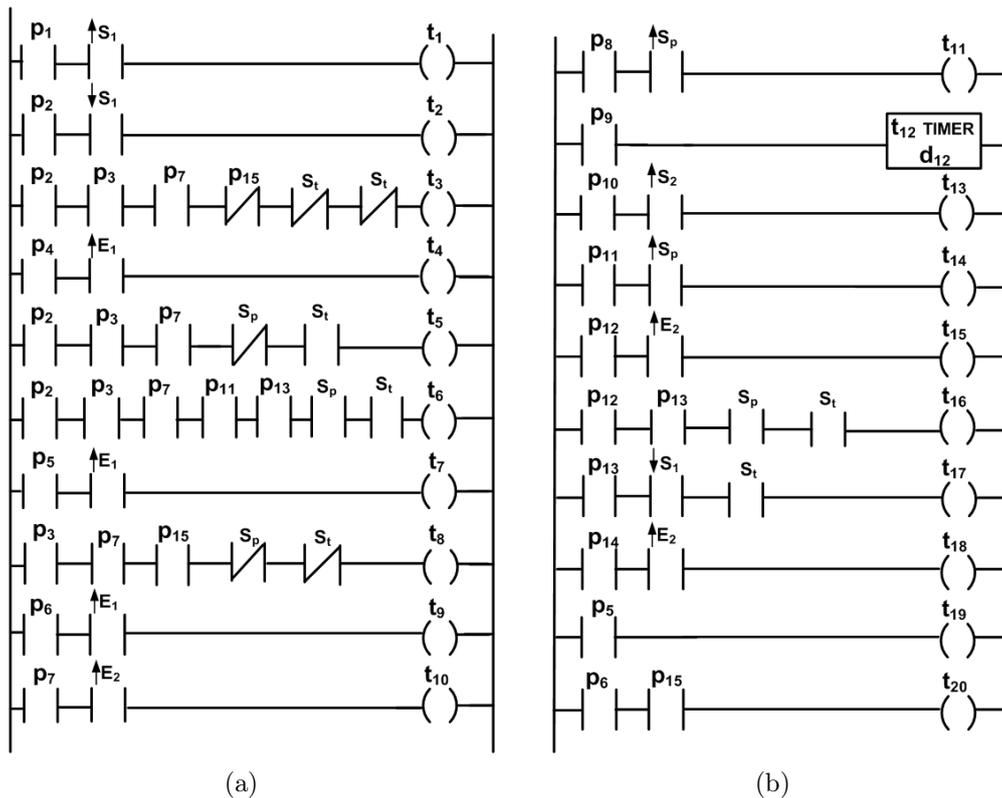


Figura 6.13: Módulo de condições para disparo das transições da RPIC da figura 6.12.

## Módulo da Dinâmica da Rede de Petri

O Ladder do módulo da dinâmica da rede de Petri é obtido através da matriz de incidência  $W$ , que é a mesma matriz de incidência da segunda modelagem do sistema de automação. Contudo, como a solução do conflito foi feita no módulo das condições para o disparo das transições, através do uso de arco inibidor, não é acrescido um contato NF ao rung da transição  $t_3$  no módulo da dinâmica da rede de Petri. Este módulo é ilustrado na figura 6.14.

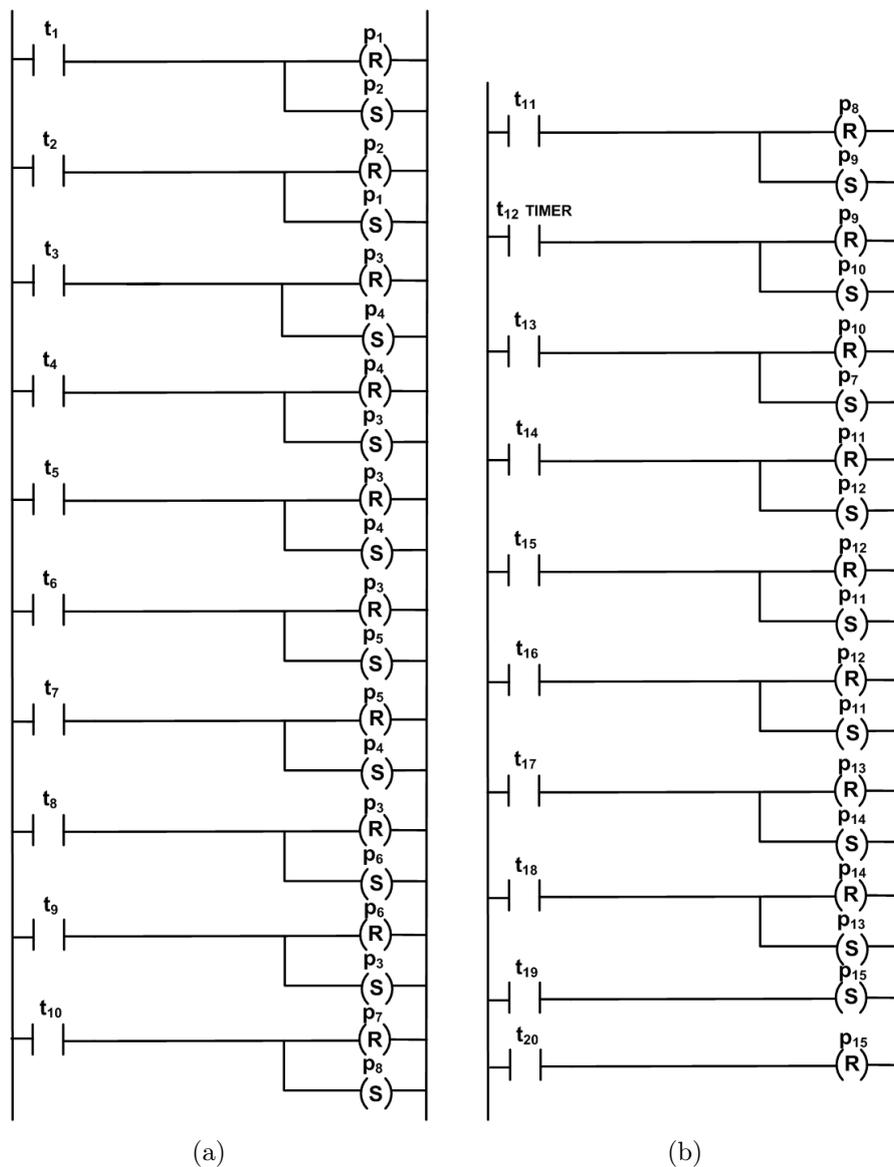


Figura 6.14: Módulo da dinâmica da RPIC da figura 6.8 dividido em duas partes.

### **Módulo das Ações**

O diagrama Ladder deste último módulo é idêntico ao diagrama Ladder das ações das modelagens anteriores, pois não há mudança nas ações.

## **6.3 Resumo**

Neste capítulo foi apresentado um sistema de manufatura, e descrito seu funcionamento com o intuito de ilustrar o uso da RPIC definida no trabalho e apresentar o método de conversão. Em seguida, o sistema de automação foi modelado por três RPIC's com estruturas distintas, demonstrando como o método de conversão pode ser aplicado para resolver cada um dos casos: através de receptividades mutuamente excludentes; priorização no disparo transições; e através do uso de arcos inibidores.

# Capítulo 7

## Conclusão

Este projeto apresentou uma novo tipo de rede de Petri Interpretada para Controle (RPIC). Essa rede de petri segura e permite o uso de arcos habilitadores e inibidores em sua estrutura., sendo ainda determinística, pois é capaz de resolver conflitos reais. Isto é possível, pois além de permitir o uso de receptividades, temporizações nas transições, e arcos inibidores, este tipo de rede ainda permite a ordenação do disparo de transições em conflito.

Como exemplo foi utilizado um sistema de manufatura, cujo sistema de automação foi modelado utilizando-se uma RPIC. Foram utilizadas três RPICs diferentes para resolver os conflitos no disparo das transições. A primeira RPIC foi utilizada para resolver os conflitos do sistema de manufatura com receptividades mutuamente excludentes entre as transições em conflito. A segunda RPIC utilizou-se de uma ordem de priorização entre as transições conflitantes, enquanto na última foram utilizados arcos inibidores para a solução do problema. Com a solução por receptividades mutuamente excludentes percebe-se que o sistema apresenta menos lugares, uma vez que o CLP precisa armazenar menos informações sobre o sistema devido à presença de sensores que detectam mudanças no ambiente externo e enviam sinais para o CLP. A RPIC torna-se menor e mais compacta, facilitando a realização do controlador, e posterior aplicação no CLP. Porém esta solução apresenta-se mais custosa, pois é necessário um maior número de sensores, e o sistema fica dependente dos sinais enviados pelos sensores. Com a utilização de menos sensores, tanto no caso de priorização de transições, como no uso de arcos inibidores, o CLP precisa

armazenar mais informações sobre o que ocorre no sistema, e o supervisor necessita de mais lugares e transições para representar cada alteração de estado do sistema.

Para a comprovação do método, foi implementado o código Ladder obtido da primeira RPIC no software RSLogix 500 da Rockwell. O código obtido permitiu que todos os estados da RPIC fossem alcançados sem haver descontinuidade. A simulação foi realizada apenas para a primeira RPIC uma vez que a estrutura das demais RPICs é consideravelmente semelhante às demais RPICs.

O método apresentado é vantajoso principalmente para sistemas muito complexos, pois nestes casos a construção em Ladder do sistema de automação de um sistema a evento discreto diretamente em um CLP torna-se impraticável, enquanto que o método de conversão apresentado permite uma conversão direta entre uma RPIC e seu diagrama Ladder. Ainda, como o método se baseia nas matrizes de incidência e incidência de entrada de uma RPIC, é possível determinar o tamanho do diagrama Ladder da RPIC utilizada como sistema de automação. Foi mostrado que o método proposto é capaz de resolver problemas de implementação como o efeito avalanche. Por todas essas razões, e ainda por se tratar de um método de conversão bem estruturado e baseado na matrizes de incidência, a obtenção do Ladder de um SED modelado por uma RPIC é mais simples e rápida que em outros métodos encontrados na literatura.

# Referências Bibliográficas

- Cassandras, C. & Lafortune, S. (2008). *Introduction to Discrete Event Dynamic Systems*, Springer, New York, N.Y., USA.
- David, R. & Alla, H. (2005). *Discrete, Continuous, and Hybrid Petri Nets*, Springer, Berlin-Heidelberg, Germany.
- Fabian, M. & Hellgren, A. (1998). PLC-based implementation of supervisory control for discrete event systems, *Proceedings of the 37th IEEE Conference on Decision and Control* .
- Georgini, M. (2007). *Automação Aplicada Descrição e Implementação de Sistemas Sequenciais com PLCs*, Érica Ltda, São Paulo, Brasil.
- IEC (1995). IEC 1131-3, *Technical report*, IEC.
- Jimenez, I., Lopez, E. & Ramirez, A. (2001). Synthesis of Ladder diagrams from petri nets controller models, *Proceedings of the 2001 IEEE International Symposium on Intelligent Control* **1**: 225–230.
- Lee, G. B. & Lee, J. S. (2000). The state equation of petri net for ld program, *IEEE International Conference On Systems and Man. and Cybernetics* **1**: 3051–3056.
- Lee, G. B., Zandong, H. & Lee, J. S. (2004). Automatic generation of Ladder diagram with control petri nets, *Journal of Intelligent Manufacturing* **15**: 245–252.
- Moreira, M. V., Botelho, D. S. & Basilio, J. C. (2009). Ladder diagram implementation of control interpreted petri nets: a state equation approach, *4th IFAC Workshop on Discrete-Event System Design* .

- Moreira, M. V., Botelho, D. S. & Hazan, S. S. (2009). Implementação de sistemas de automação descritos por redes de petri interpretadas para controle, *37 Congresso Brasileiro de Educação em Engenharia* .
- Murata, T. (1989). Petri nets: Properties, analysis and applications, *Proceedings of the IEEE* **77**: 541–580.
- Ohman, M., Johansson, S. & Arzen, K. (1998). Implementation aspects of the PLC standard IEC 1131-3, *Control Engineering Practice* **6**: 547–555.
- Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, N.J., USA.
- Richardsson, J. & Fabian, M. (2003). Automatic generation of PLC programs for control of flexible manufacturing cells, *Proceedings of the 2003 IEEE Emerging Technologies and Factory Automation* .
- Rockwell, A. (2002). *SLC 500 Instruction Set*, Rockwell Automation, Inc, Milwaukee, WI - USA.
- Satoh, T., Oshima, H., Nose, K. & Kumagai, S. (1992). Automatic generation system of ladder list program by Petri net, *IEEE International Workshop on Emerging Technologies and Factory Automation* .
- Uzam, M. & Jones, A. H. (1998). Discrete event control system design using automation petri nets and their ladder diagram implementation, *Int. J. Adv. Manuf. Technol.* **14**: 716–728.
- Uzam, M., Jones, A. H. & Ajlouni, N. (1996). Conversion of petri nets controllers for manufacturing systems into ladder logic diagrams, *IEEE Conference on Emerging Technologies and Factory Automation* .
- Uzam, M., Jones, A. H. & Yücel, I. (2000). Using a petri-net-based approach for the real-time supervisory control of an experimental manufacturing system, *Int. J. Adv. Manuf. Technol.* **16**: 498–515.

Zhou, M. & Venkatesh, K. (1999). *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*, World Scientific Publishing Company, New Jersey, N.J., USA.