

# Automação Industrial baseada em PLCs

1ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

### **Objectivos:**

- Modelação de sistemas de eventos discretos.
- Análise de sistemas para automação industrial.
- Metodologias para a implementação de soluções em automação industrial.
- Linguagens de programação de PLCs.

## 1ª aula

### Cap. 1 - Introdução à Automação

- Introdução aos dispositivos utilizados em automação industrial.
- Lógica cablada e lógica programada.
- Introdução às metodologias de descrição de problemas em automação industrial.

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.
- Interligação entre PLCs.

## Programa da cadeira (cont.):

### Cap. 3 - Linguagens de Programação de PLCs

- Diagramas de contactos (ladder diagram).
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.

### Cap. 5 - Sistemas de Eventos Discretos

- Modelação de sistemas de eventos discretos.
- Autómatos.
- Redes de Petri.



## Programa da cadeira (cont.):

### Cap. 6 - Análise de Sistemas de Eventos Discretos

- Propriedades de SEDs.
- Metodologias para análise de SEDs.

### Cap. 7 - SEDs e Automação Industrial

- Relação GRAFCET/Redes de Petri.
- Análise das soluções de automação industrial por analogia com os sistemas de eventos discretos.

## Dispositivos utilizados em automação industrial

Para existir produção industrial as matérias primas e/ou os componentes constituintes dos produtos que se pretendem produzir têm de ser manipulados da forma apropriada.

Com a revolução industrial aparecem as máquinas que transformam os produtos. Com o aparecimento do computador passa-se a ter uma forma de comandar automaticamente os automatismos existentes.

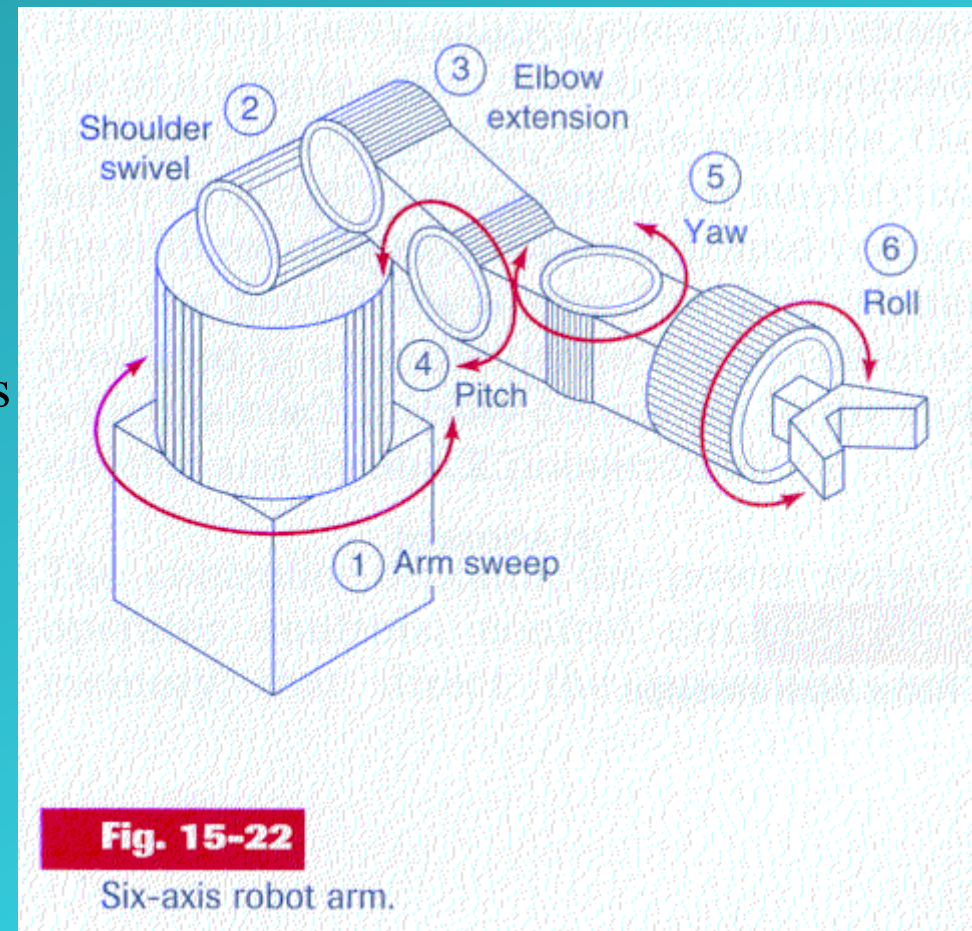


# Dispositivos utilizados em automação industrial:

## Manipuladores Robóticos

### Características relevantes:

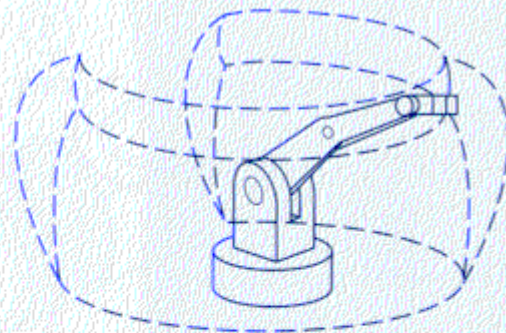
- Número de graus de liberdade
- Juntas de revolução ou prismáticas
- Programação (linguagem de alto nível, teach pendent, ...)
- Espaço de trabalho
- Repetibilidade, precisão
- Capacidade de carga e robustez.



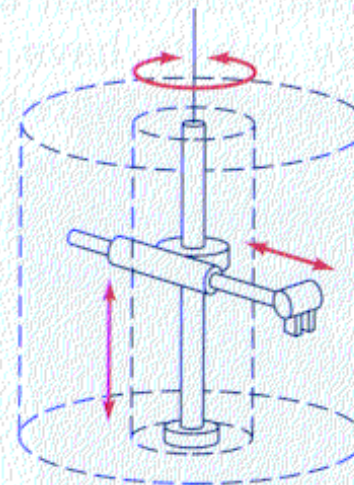


# Dispositivos utilizados em automação industrial:

## Manipuladores Robóticos (cont.)



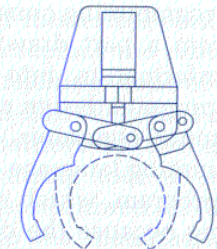
(b) Articulated



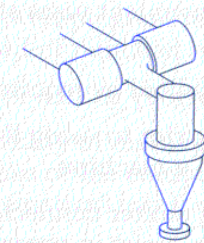
(a) Cylindrical

**Fig. 15-23**

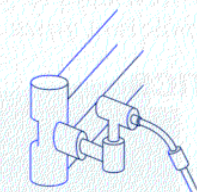
Robot work envelope.



(a) Gripper



(b) Grinder



(c) Gas welding torch

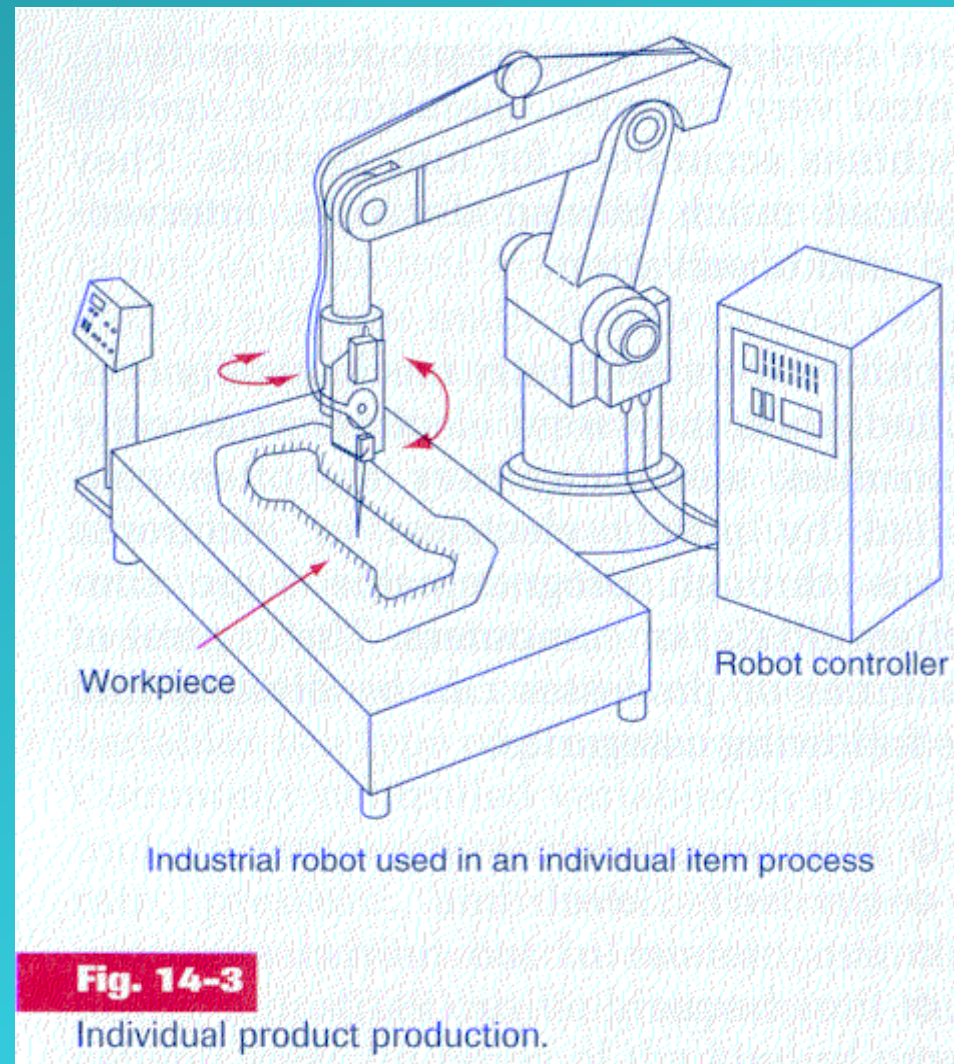
**Fig. 15-24**

End-of-arm tooling devices.

# Dispositivos utilizados em automação industrial:

## Manipuladores Robóticos (cont.)

Incorporação em células de trabalho implica dotar os manipuladores de interfaces para sincronismo, e comando externos.





# Dispositivos utilizados em automação industrial:

## Máquinas de Comando Numérico

### Características relevantes:

- Número de graus de liberdade
- Métodos de interpolação
- Automatização na carga/descarga e na mudança de ferramentas
- Programação (linguagem de alto nível, teach pendent, ...)
- Espaço de trabalho
- Repetibilidade, precisão.
- Capacidade de carga e robustez.
- Interface com o exterior
- Sincronismo com exterior

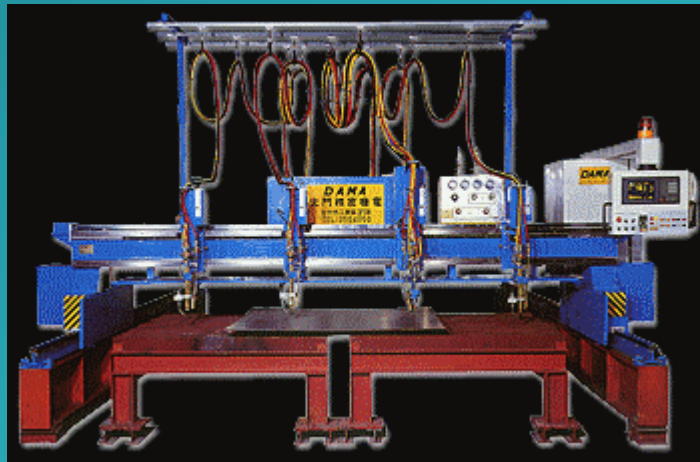
### Exemplos:

Tornos, Frezadoras, Quinadeiras, ...



# Dispositivos utilizados em automação industrial:

## Máquinas de Comando Numérico (cont.)



# Dispositivos utilizados em automação industrial:

## Tapetes de Transporte

### Características relevantes:

- Automatização na carga/descarga
- Comando
- Repetibilidade, precisão.
- Capacidade de carga e robustez.
- Interface com o exterior
- Sincronismo com exterior



# Dispositivos utilizados em automação industrial:

## AGVs (Automatic Guided Vehicles)

### Características relevantes:

- Automatização na carga/descarga
- Comando
- Repetibilidade, precisão.
- Capacidade de carga e robustez.
- Interface com o exterior
- Sincronismo com exterior





# Dispositivos utilizados em automação industrial:

## AGVs (cont.)

- Frota em operação em fábrica.





# Dispositivos utilizados em automação industrial:

## Motores DC, AC e passo-a-passo

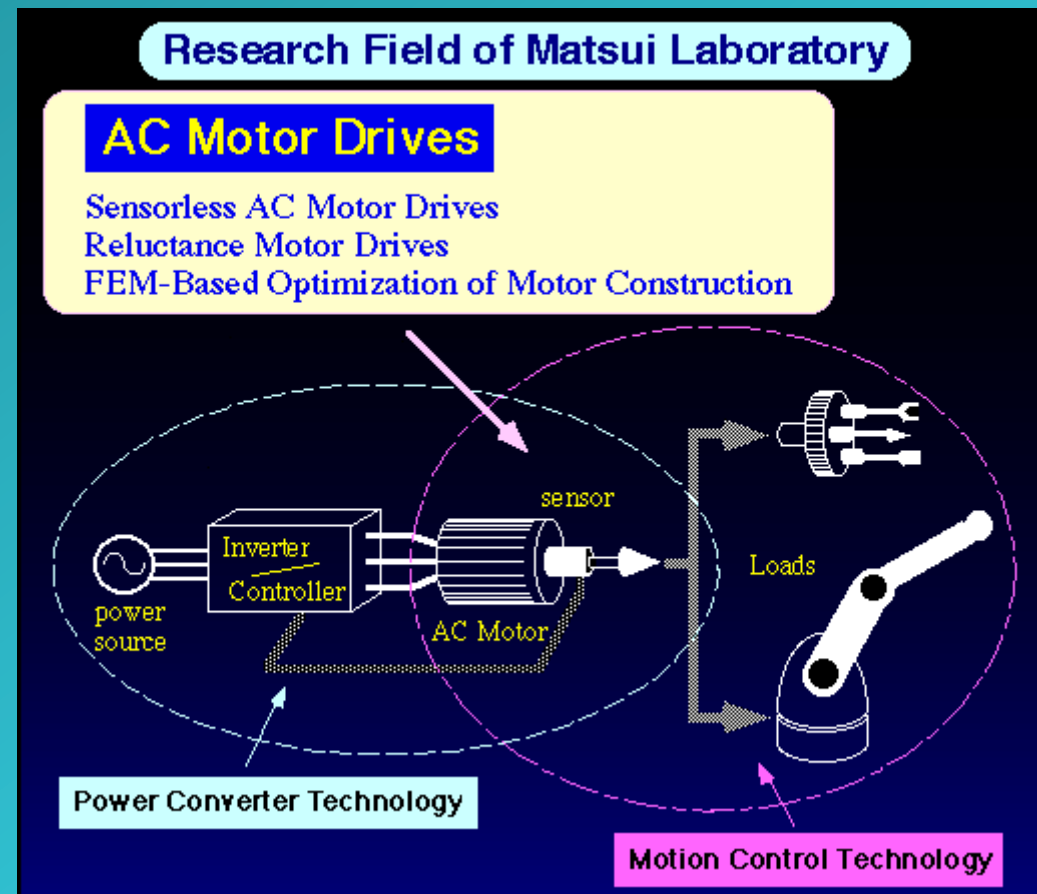
### Características relevantes:

- Tipo de arranque
- Tipo de controlo
- Repetibilidade, precisão.
- Capacidade de carga e robustez.
- Interface com o exterior
- Sincronismo com exterior



# Dispositivos utilizados em automação industrial:

## Exemplo de motor AC com driver



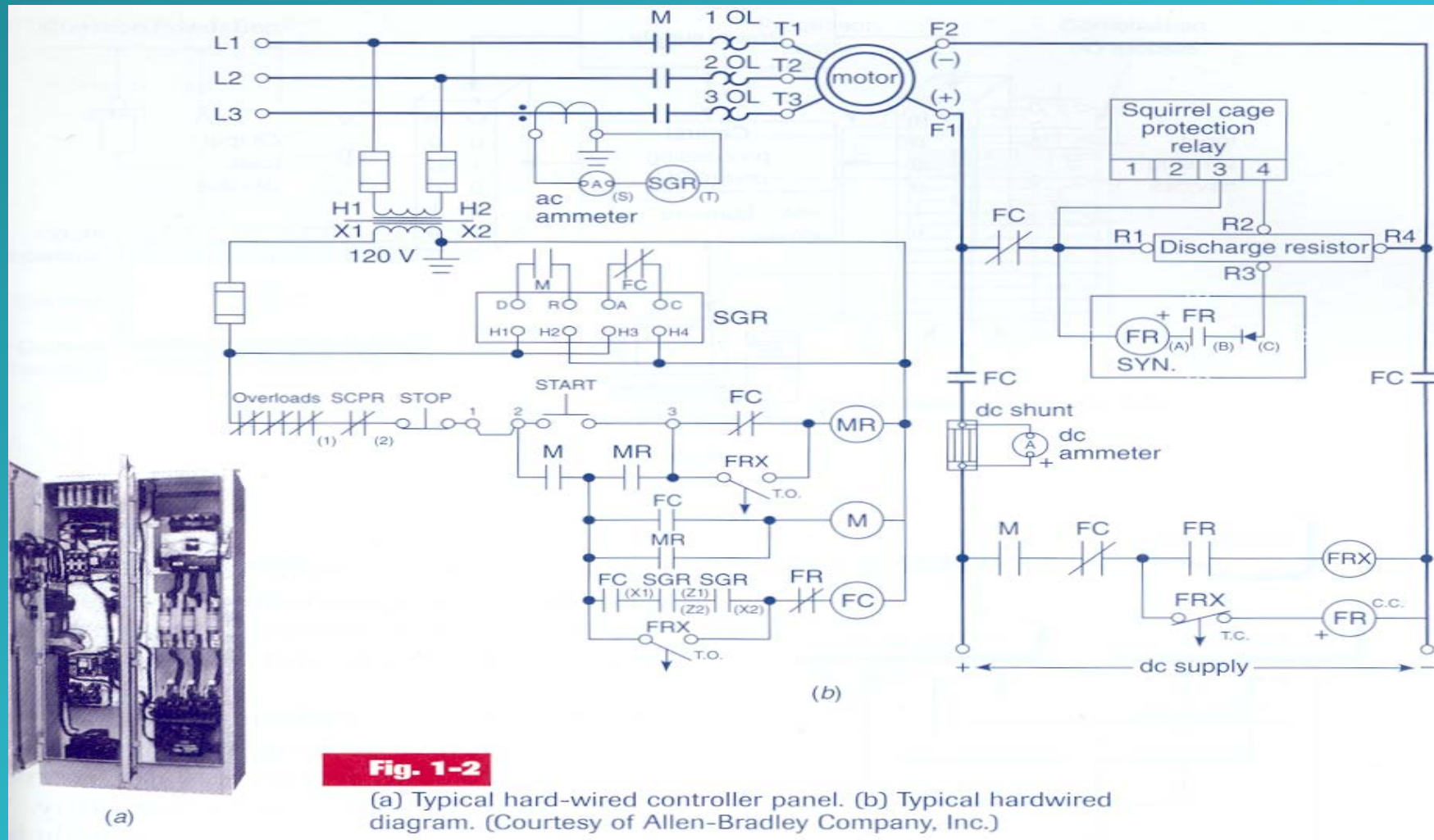
# Dispositivos utilizados em automação industrial:

## Dispositivos específicos

### Exemplo: produção de embalagens de alumínio

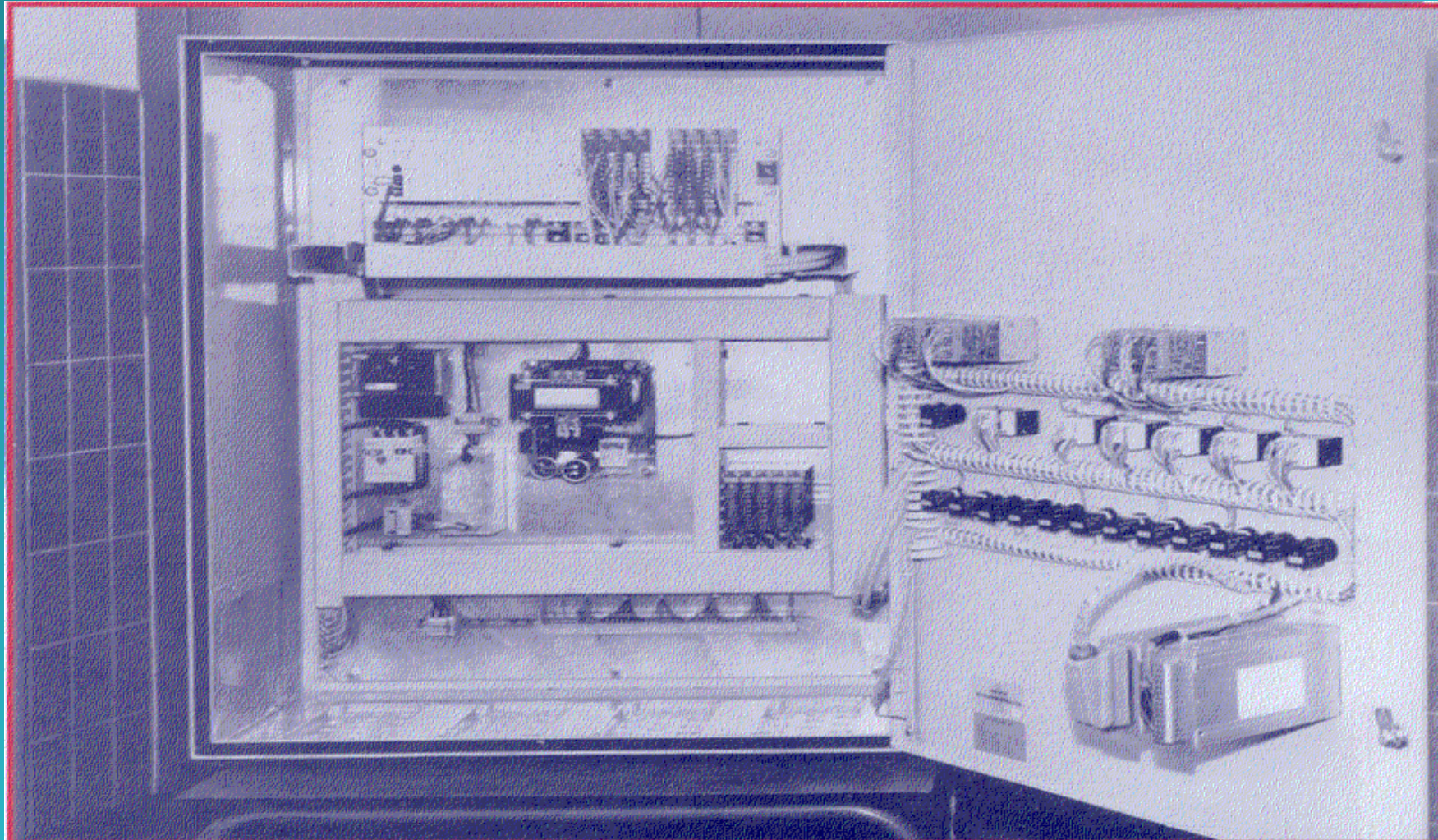


# Lógica cablada vs





# vs Lógica programada





# Automação Industrial baseada em PLCs

## 2ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 2<sup>a</sup> Aula

### Cap. 1 - Introdução à Automação

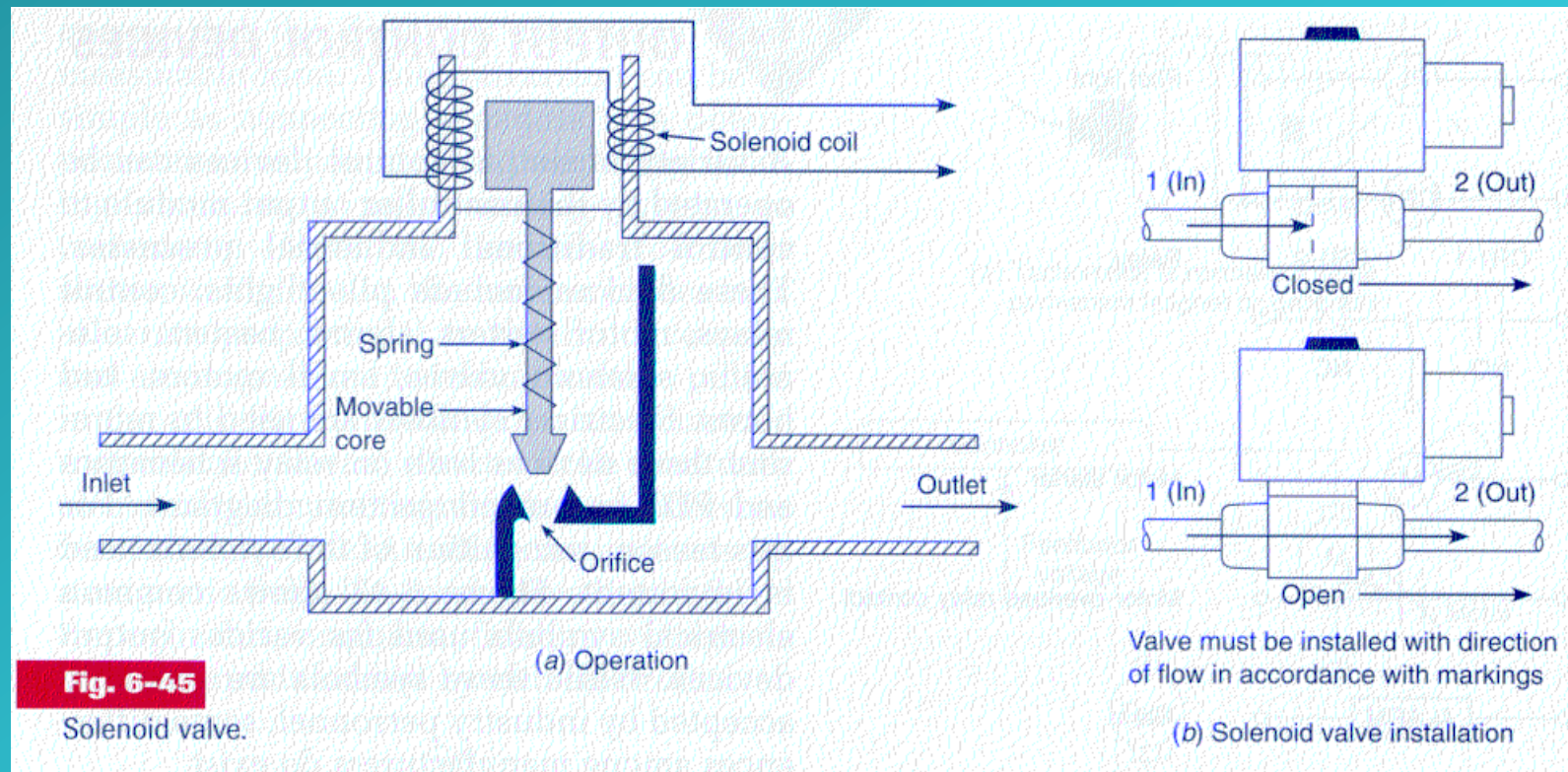
- Introdução aos dispositivos utilizados em automação industrial.
- Lógica cablada e lógica programada.
- **Introdução às metodologias de descrição de problemas em automação industrial.**

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.
- Interligação entre PLCs.

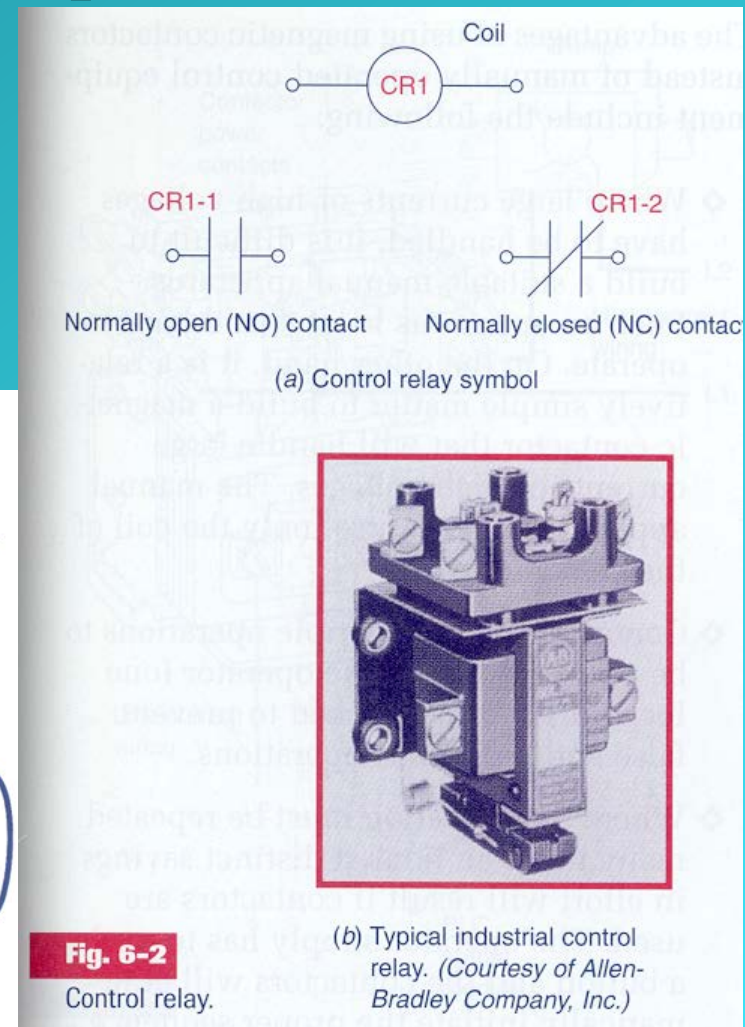
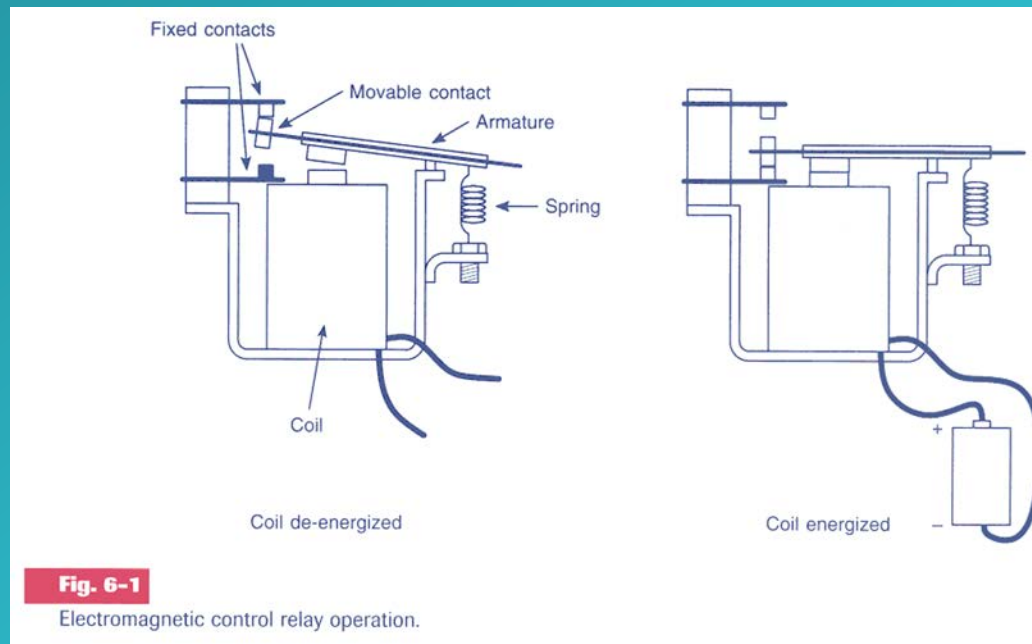
# Metodologias de descrição de problemas em Aut.

## Válvula de solenoide



# Metodologias de descrição de problemas em Aut.

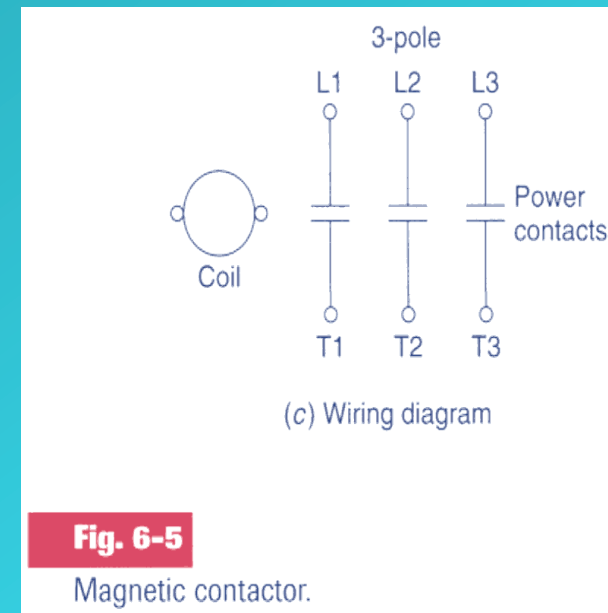
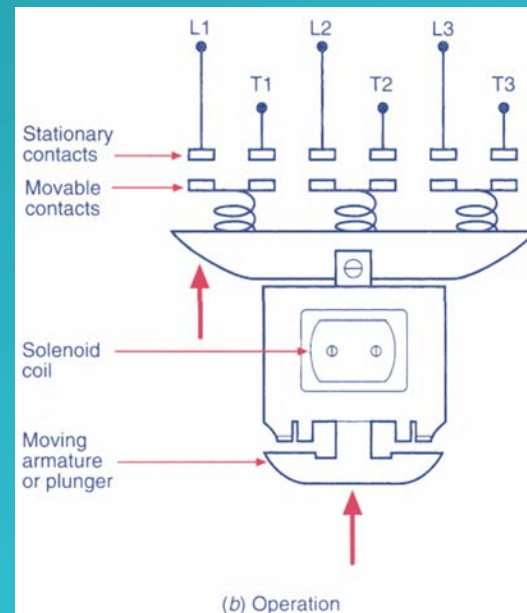
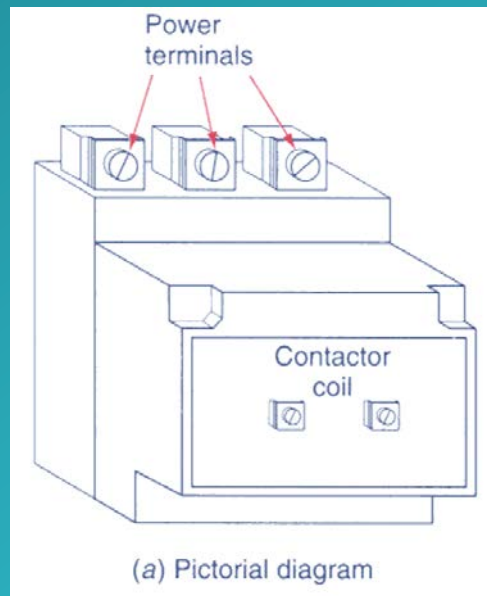
## Relés de comando





# Metodologias de descrição de problemas em Aut.

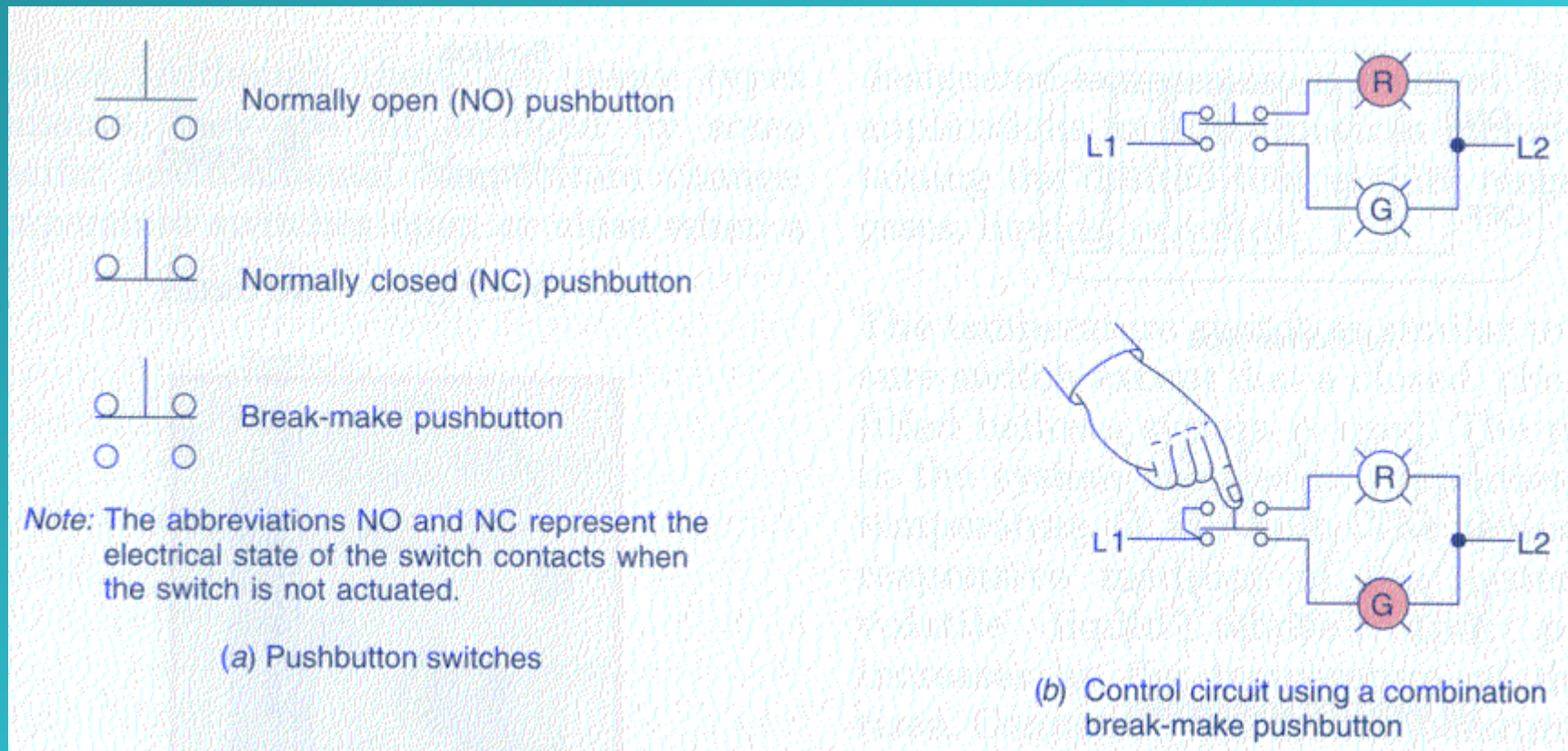
## Contactor





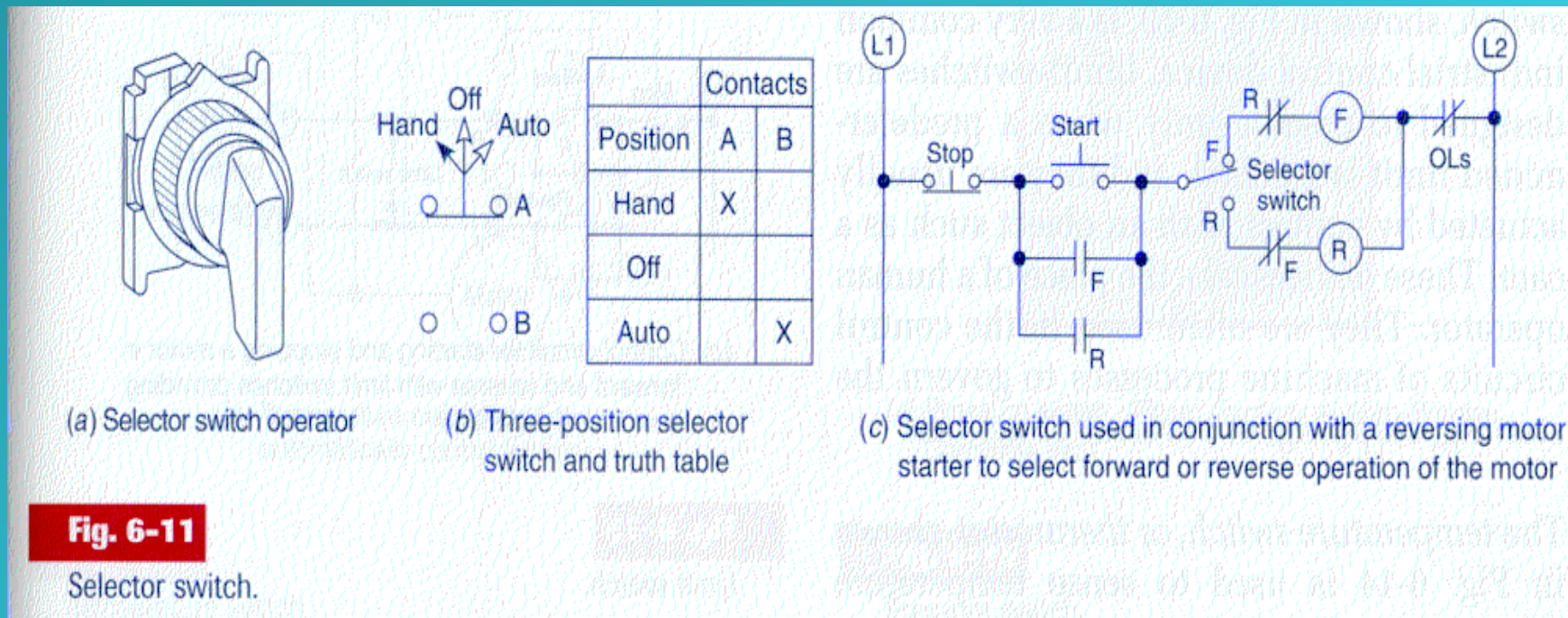
# Metodologias de descrição de problemas em Aut.

## Botões de pressão



# Metodologias de descrição de problemas em Aut.

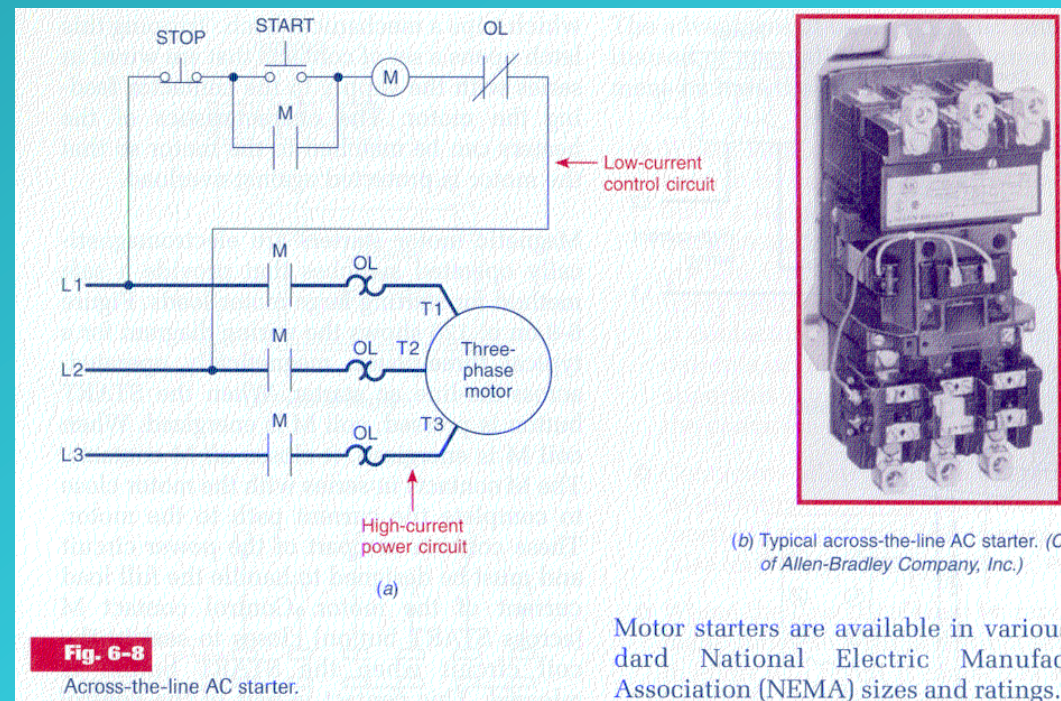
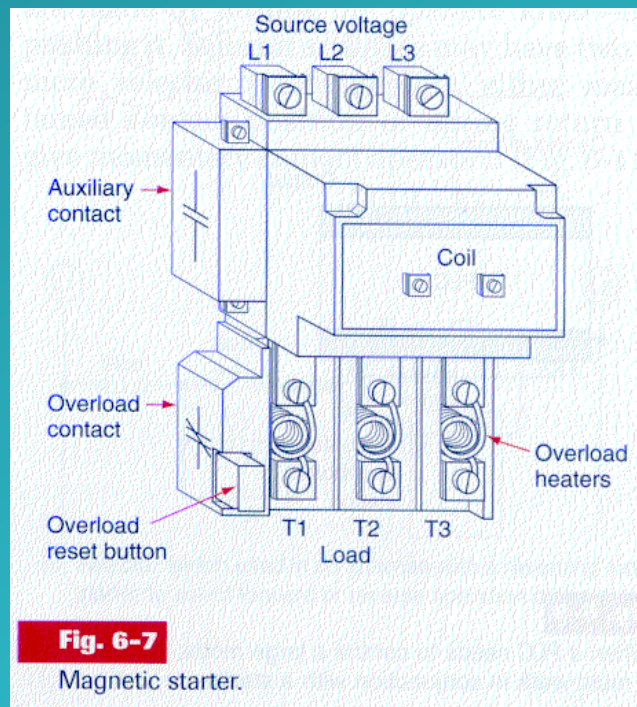
## Interruptor de selecção





# Metodologias de descrição de problemas em Aut.

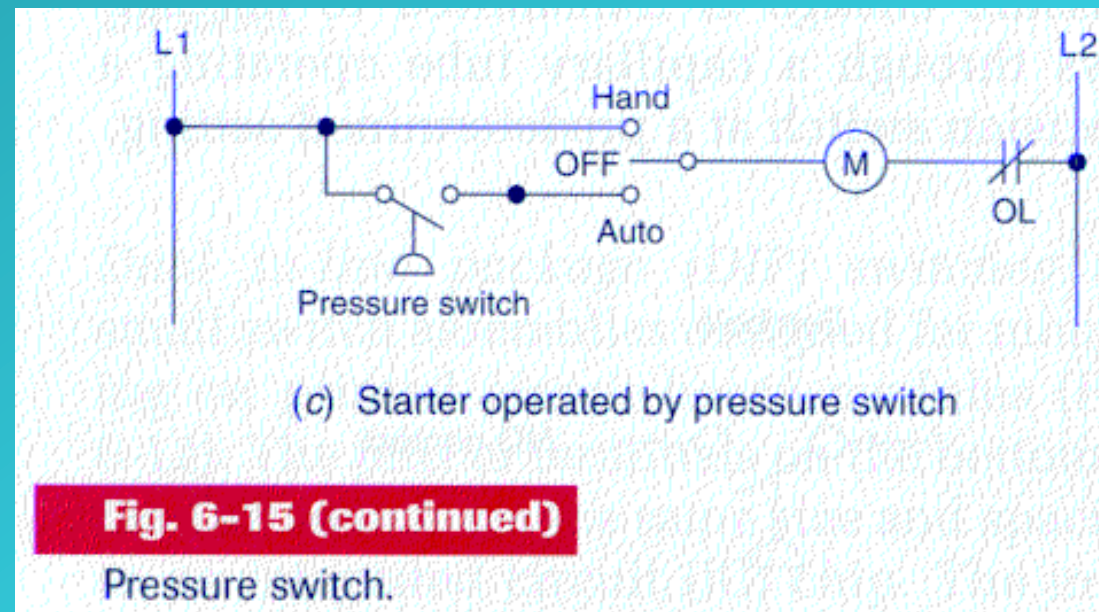
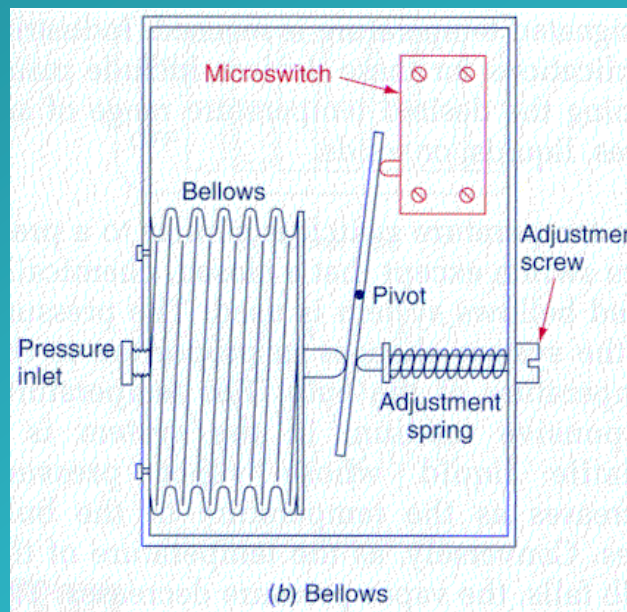
## Arrancador magnético









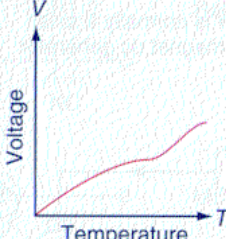
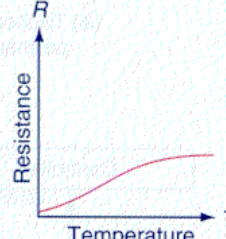
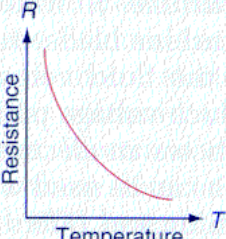
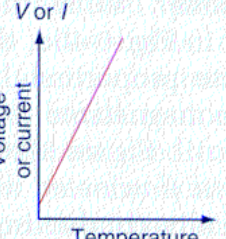
# Metodologias de descrição de problemas em Aut.

## Sensor de pressão



# Metodologias de descrição de problemas em Aut.

## Sensores de temperatura

	Thermocouple	RTD	Thermistor	IC Sensor
				
				
Advantages	<ul style="list-style-type: none"> <li>• Self-powered</li> <li>• Simple</li> <li>• Rugged</li> <li>• Inexpensive</li> <li>• Wide variety</li> <li>• Wide temperature range</li> </ul>	<ul style="list-style-type: none"> <li>• Most stable</li> <li>• Most accurate</li> <li>• More linear than thermocouple</li> </ul>	<ul style="list-style-type: none"> <li>• High output</li> <li>• Fast</li> <li>• Two-wire ohms measurement</li> </ul>	<ul style="list-style-type: none"> <li>• Most linear</li> <li>• Highest output</li> <li>• Inexpensive</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>• Nonlinear</li> <li>• Low voltage</li> <li>• Reference required</li> <li>• Least stable</li> <li>• Least sensitive</li> </ul>	<ul style="list-style-type: none"> <li>• Expensive</li> <li>• Power supply required</li> <li>• Small <math>\Delta R</math></li> <li>• Low absolute resistance</li> <li>• Self-heating</li> </ul>	<ul style="list-style-type: none"> <li>• Nonlinear</li> <li>• Limited temperature range</li> <li>• Fragile</li> <li>• Power supply required</li> <li>• Self-heating</li> </ul>	<ul style="list-style-type: none"> <li>• <math>T &lt; 200^{\circ}\text{C}</math></li> <li>• Power supply required</li> <li>• Slow</li> <li>• Self-heating</li> <li>• Limited configurations</li> </ul>

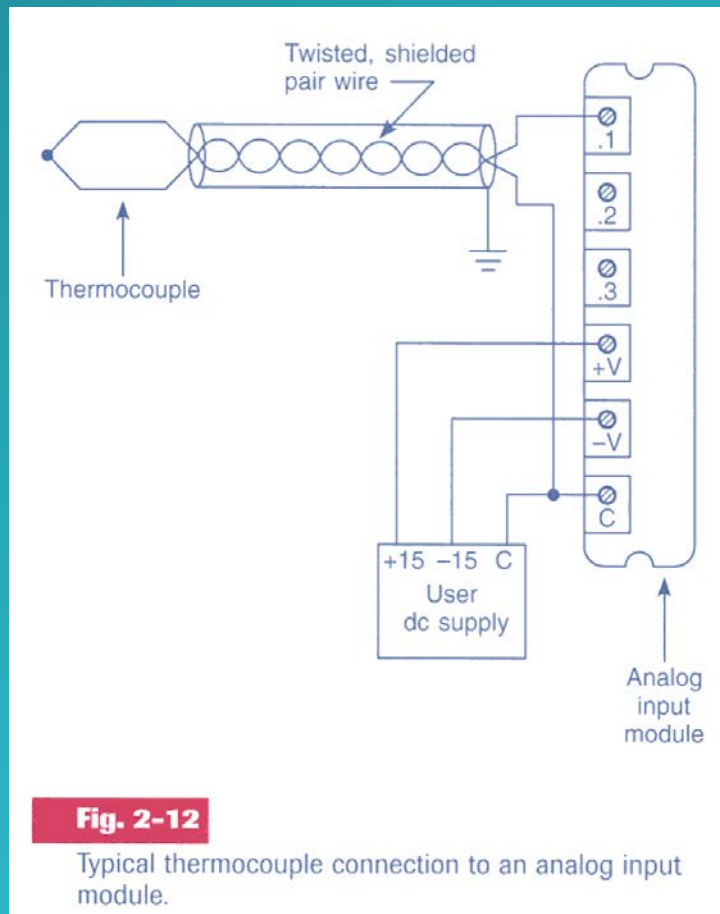
**Fig. 6-38**

Common temperature sensors.

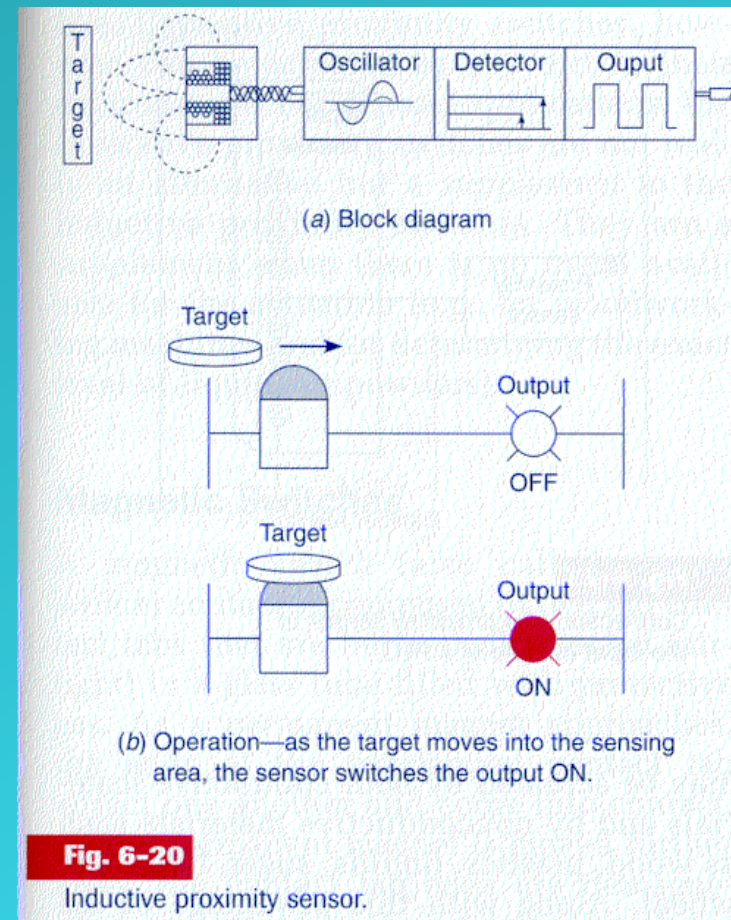


# Metodologias de descrição de problemas em Aut.

## Termopar



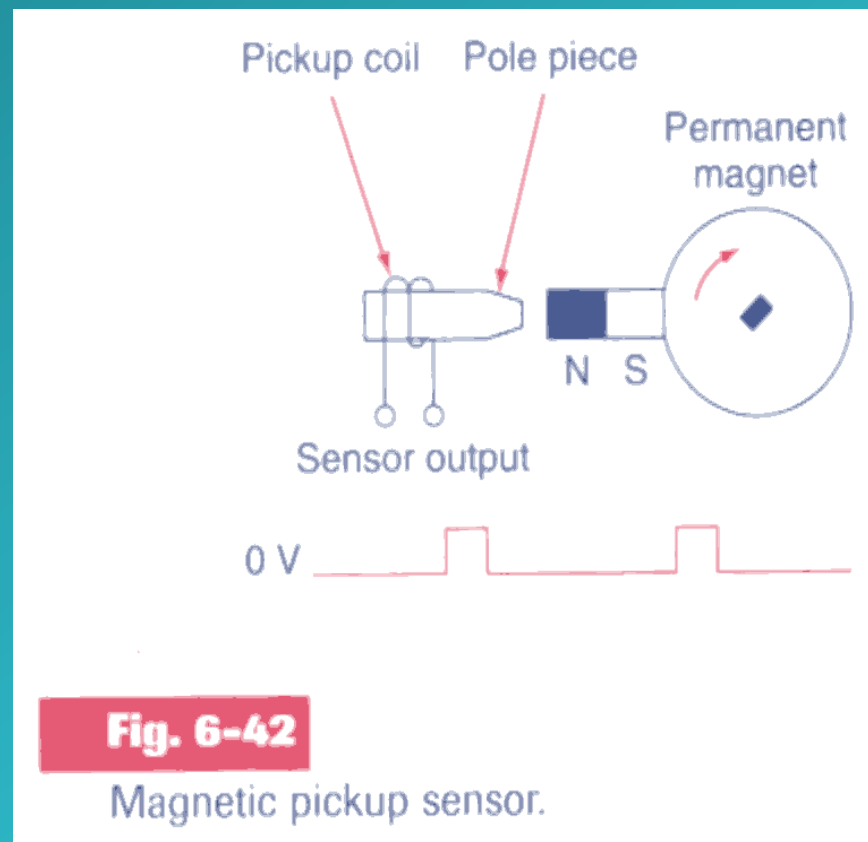
## Detector de proximidade



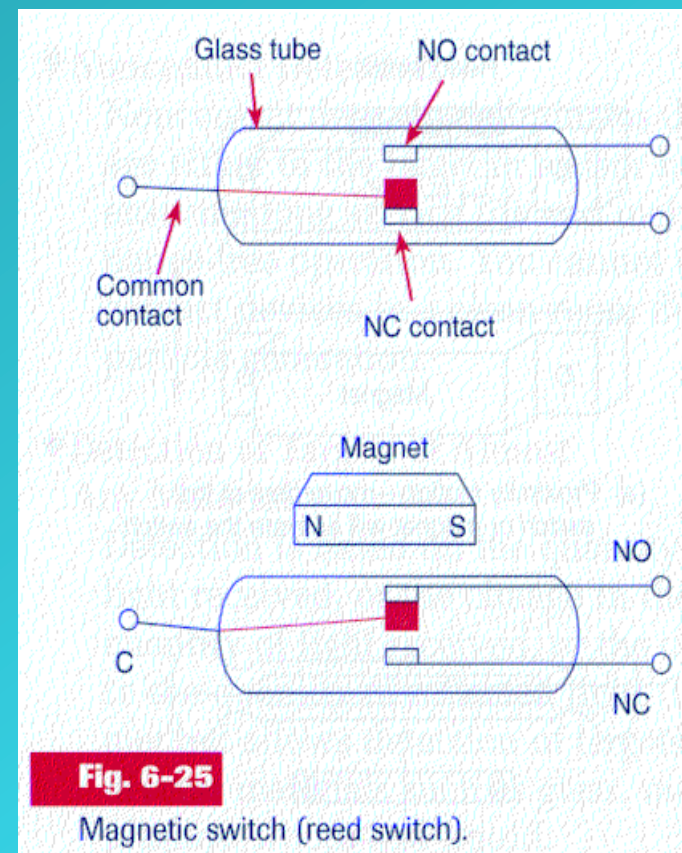


# Metodologias de descrição de problemas em Aut.

## Detector magnético

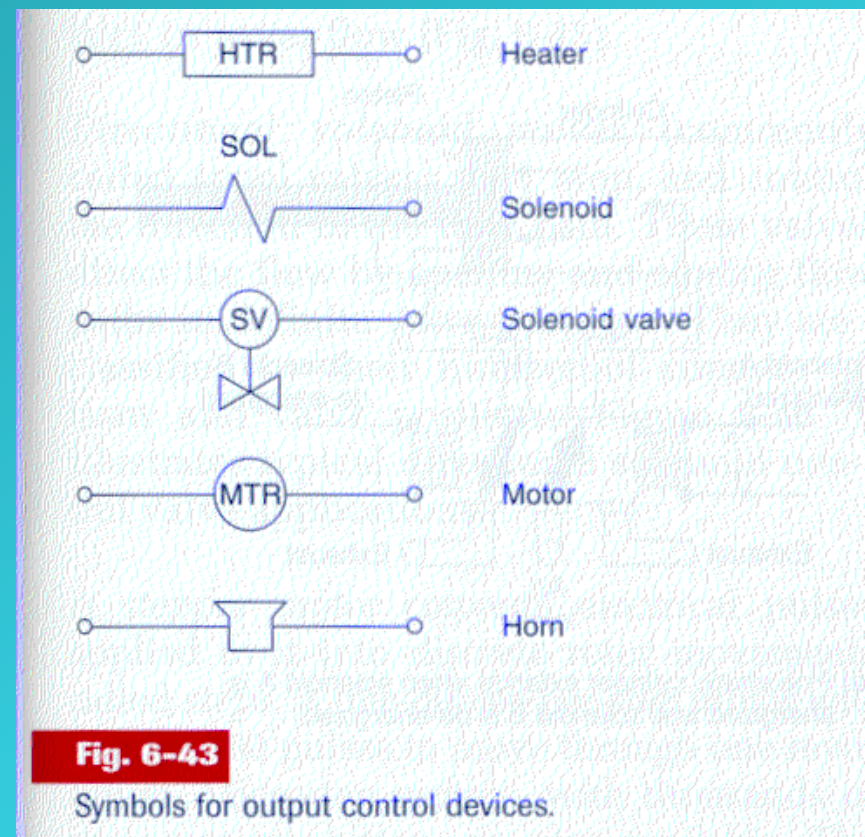
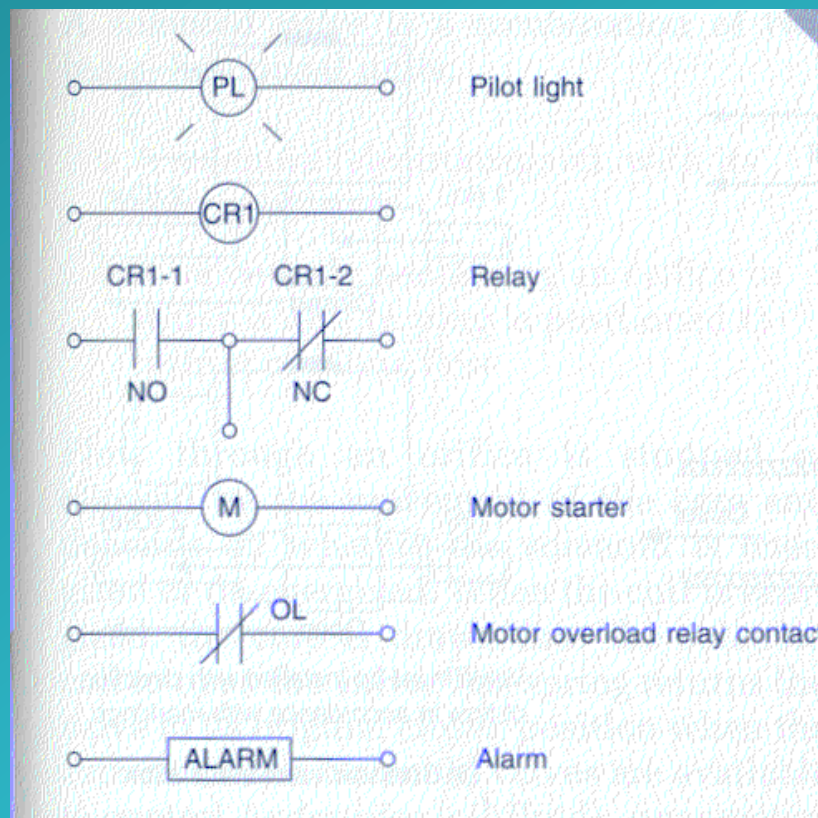


## Interruptor magnético



# Metodologias de descrição de problemas em Aut.

## Simbolos associados a vários actuadores



# Metodologias de descrição de problemas em Aut.

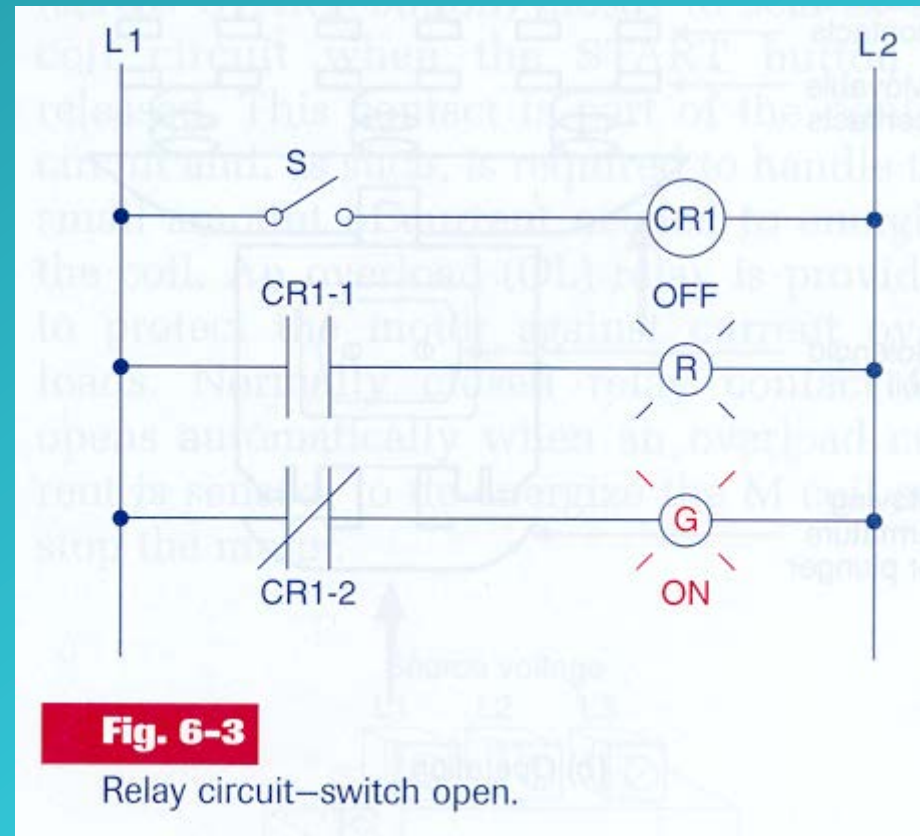
*Ladder Diagram*

ou

**Diagramas em Escada**

ou

**Diagramas de Contactos**

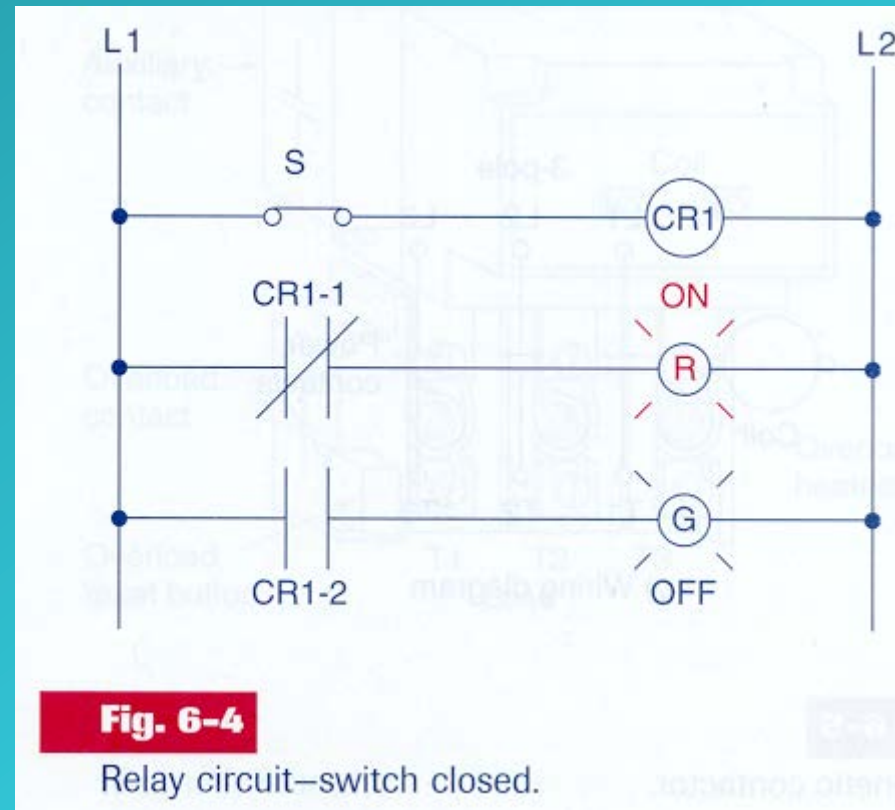




# Metodologias de descrição de problemas em Aut.

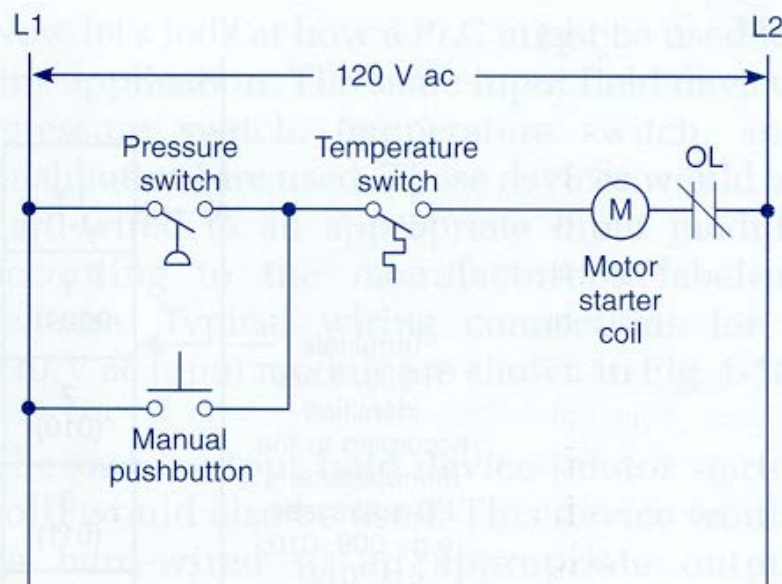
## *Ladder Diagram (cont.)*

Exemplo de funcionamento



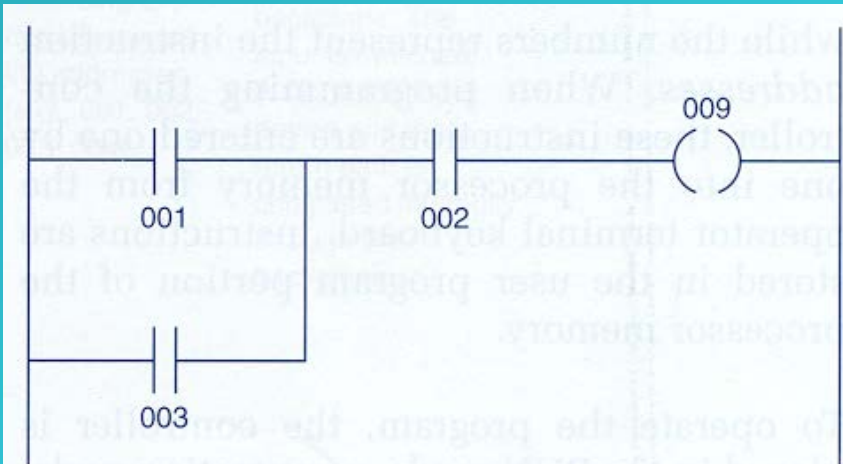
# Metodologias de descrição de problemas em Aut.

## *Exemplo*



**Fig. 1-13**

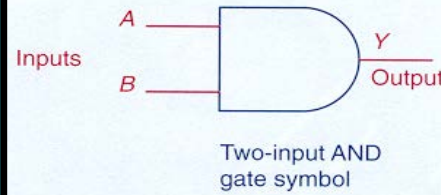
Relay ladder diagram for modified process.



**Fig. 1-14**

PLC ladder logic diagram for modified process.

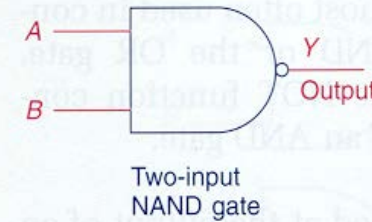
# Funções Lógicas



Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND truth table

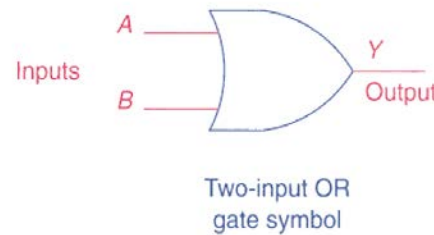
**Fig. 4-3**  
AND gate.



Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND truth table

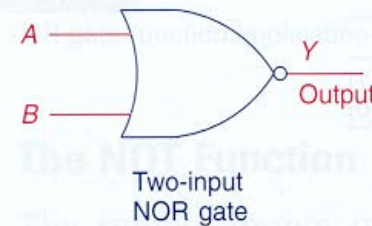
**Fig. 4-12**  
NAND gate symbol and truth table.



Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR truth table

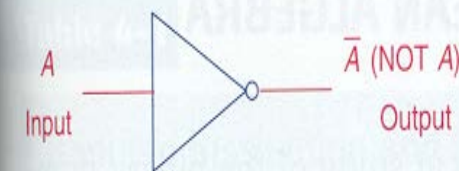
**Fig. 4-6**  
OR gate.



Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NOR truth table

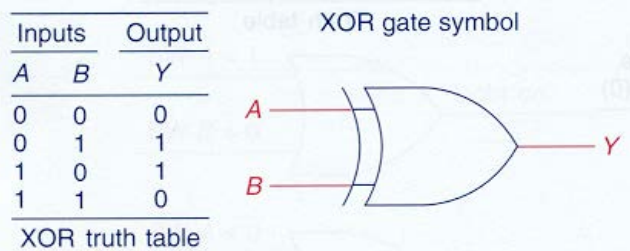
**Fig. 4-13**  
NOR gate symbol and truth table.



A	NOT A
0	1
1	0

NOT truth table

**Fig. 4-9**  
NOT function symbol.



Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

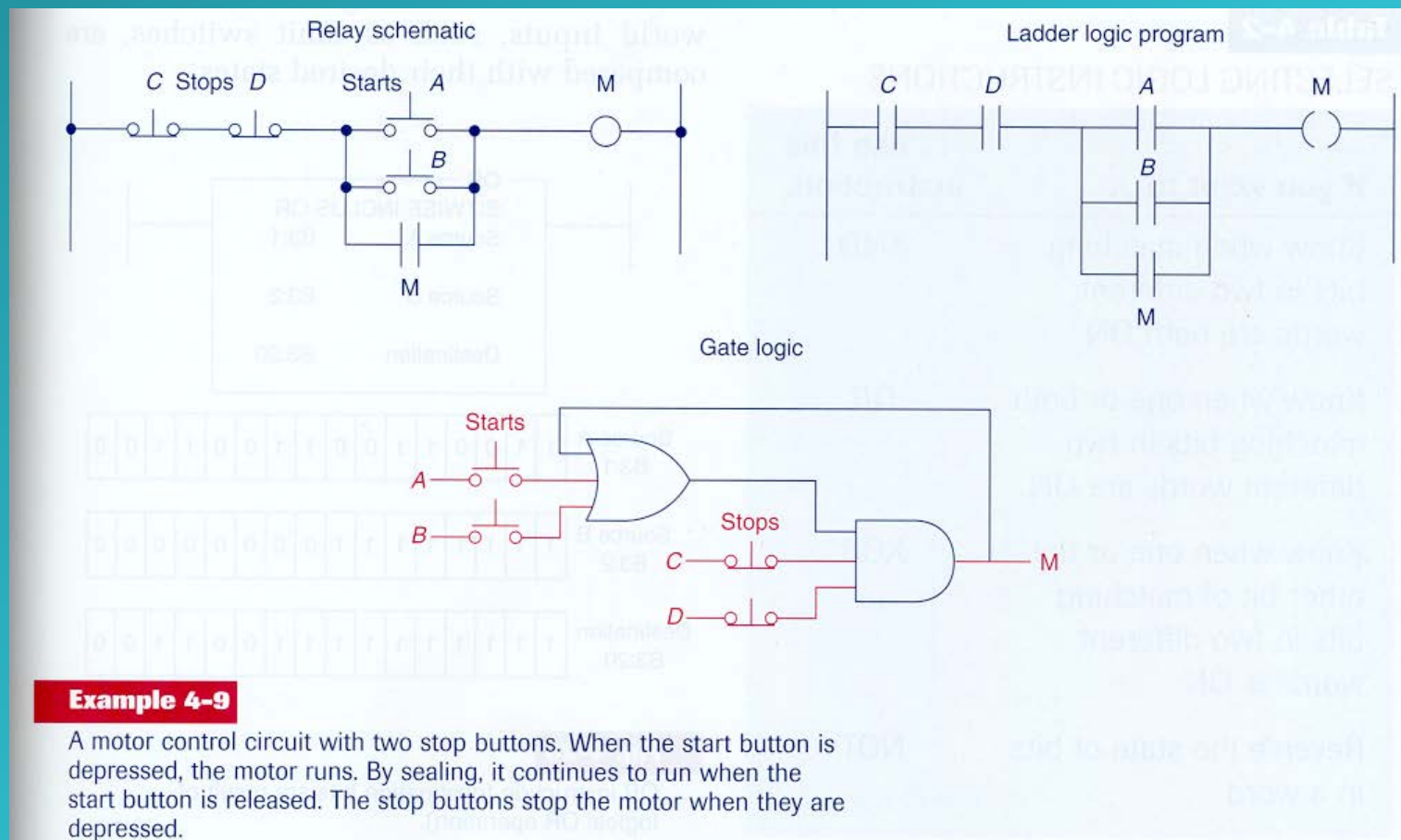
XOR truth table

**Fig. 4-14**  
The XOR (exclusive-OR) gate symbol and truth table.



# Metodologias de descrição de problemas em Aut.

## Exemplo



**Example 4-9**

A motor control circuit with two stop buttons. When the start button is depressed, the motor runs. By sealing, it continues to run when the start button is released. The stop buttons stop the motor when they are depressed.

# Automação Industrial baseada em PLCs

## 3ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 3<sup>a</sup> Aula

### Cap. 1 - Introdução à Automação

- Introdução aos dispositivos utilizados em automação industrial.
- Lógica cablada e lógica programada.
- Introdução às metodologias de descrição de problemas em automação industrial.

### Cap. 2 - Introdução aos PLCs

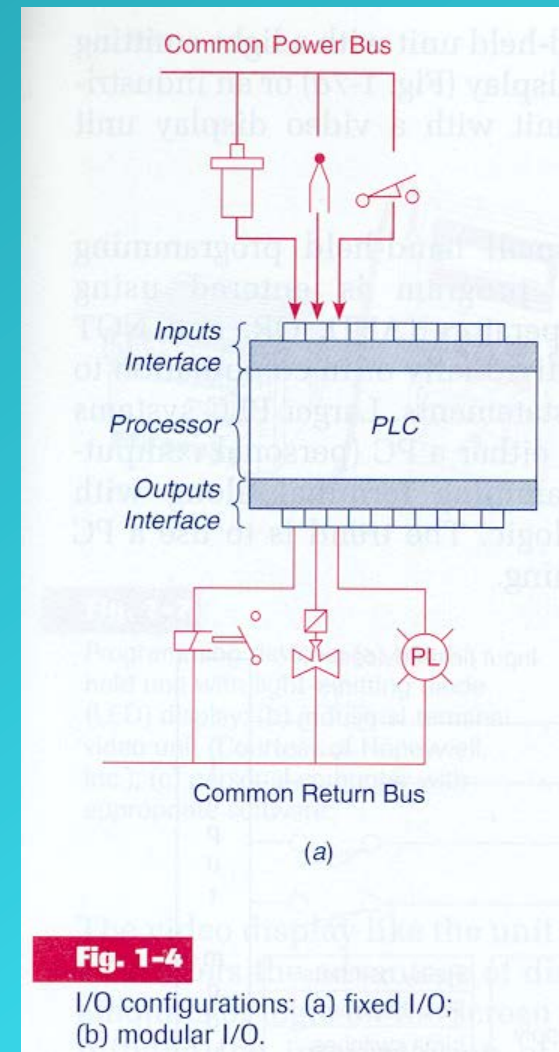
- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.
- Interligação entre PLCs.



## Componentes constituintes dos PLCs

É necessária a ligação de dispositivos de entrada (de comando ou sensores) e de saída (para aviso ou de actuação) aos PLCs.

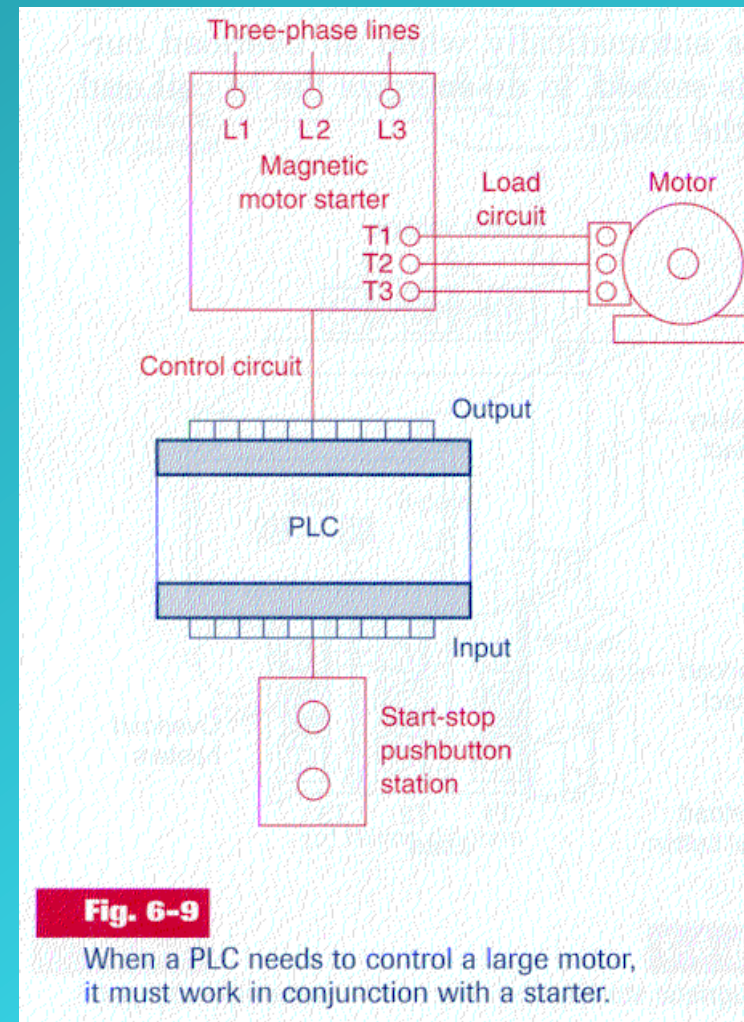
Será no PLC onde é implementado o programa que implementa a solução de automação especificada pelo utilizador.



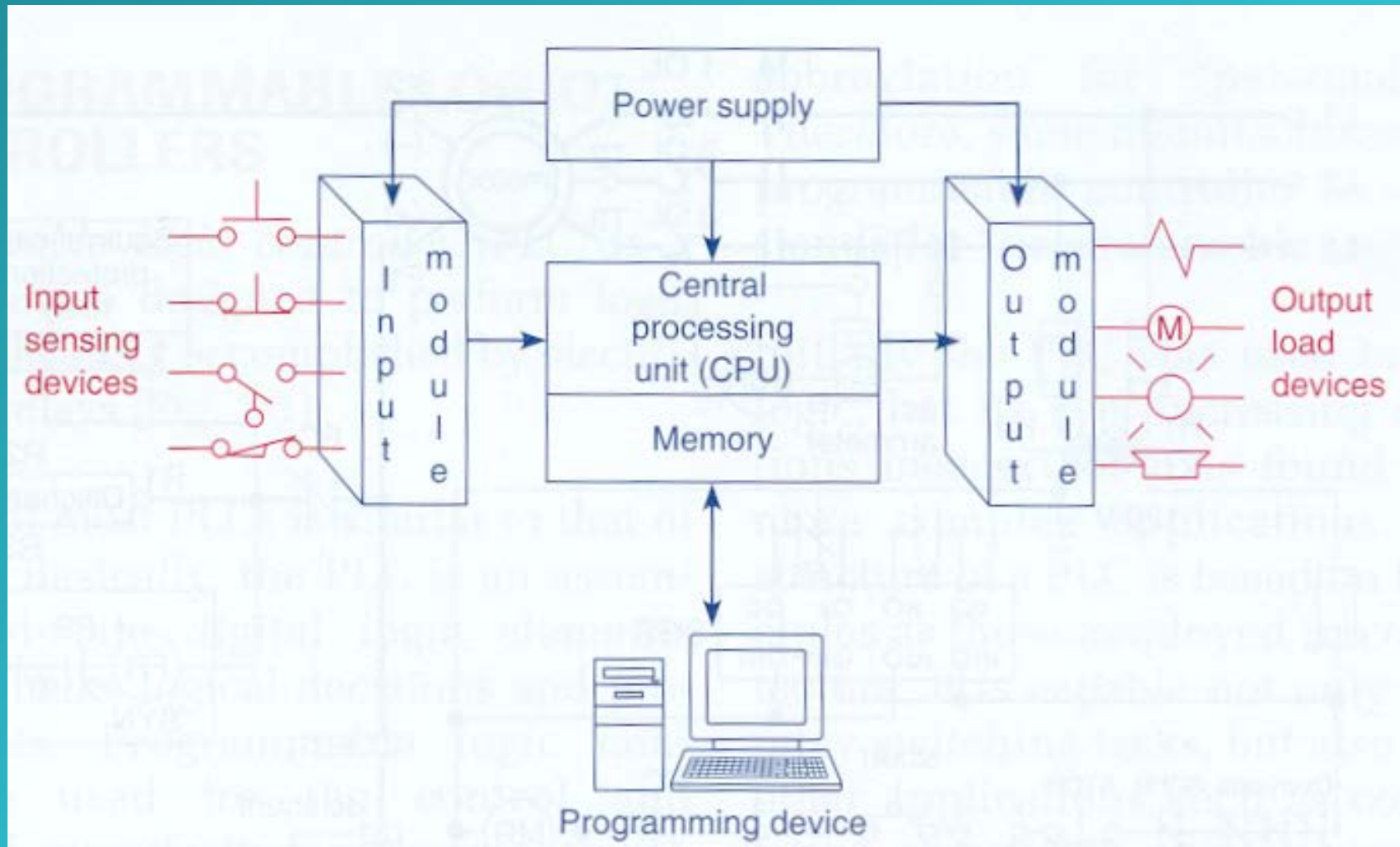
# Componentes constituintes dos PLCs

Exemplo:

ligação de um motor a partir de uma botoneira, com botões de arranque e de paragem.

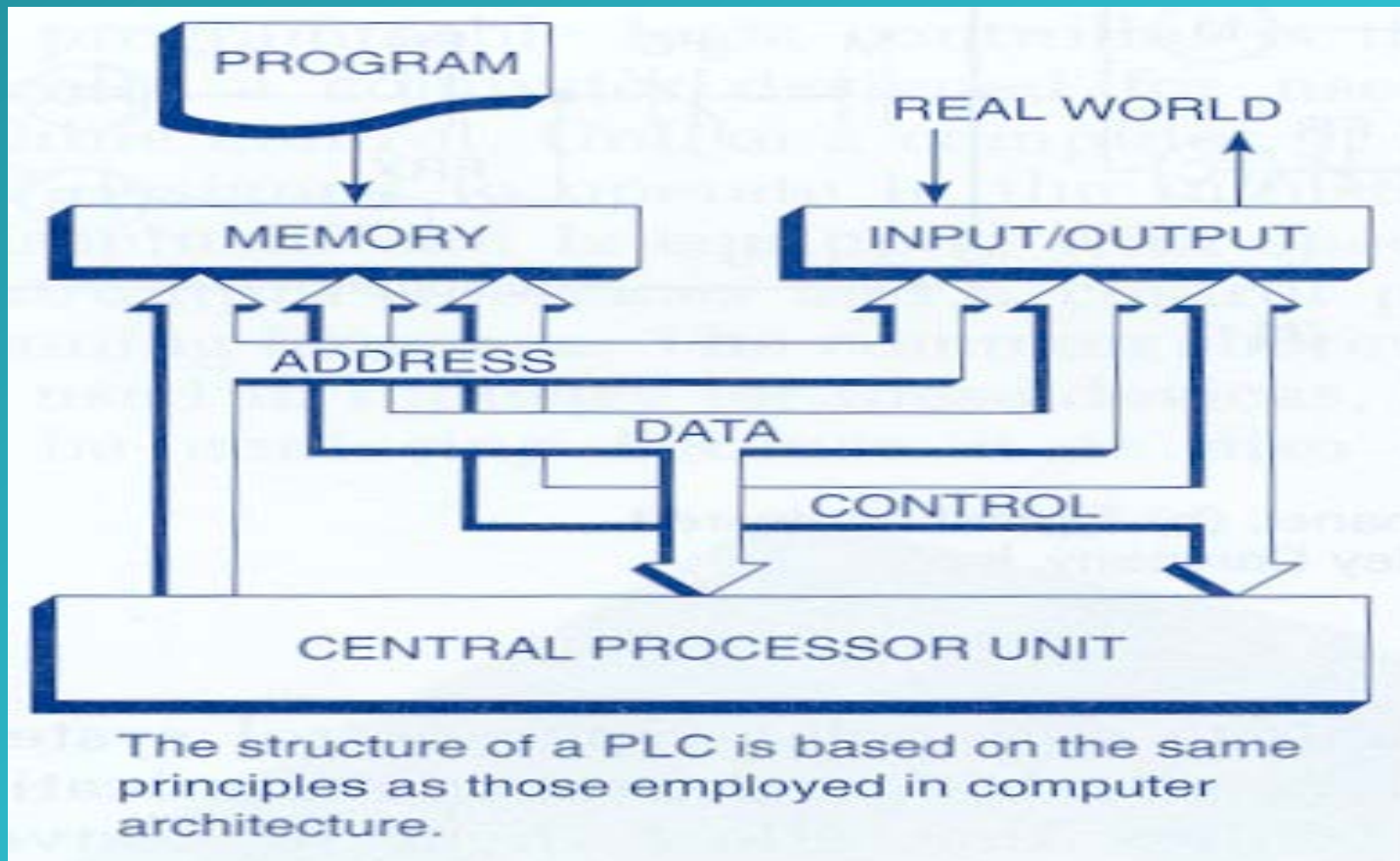


# Componentes constituintes dos PLCs



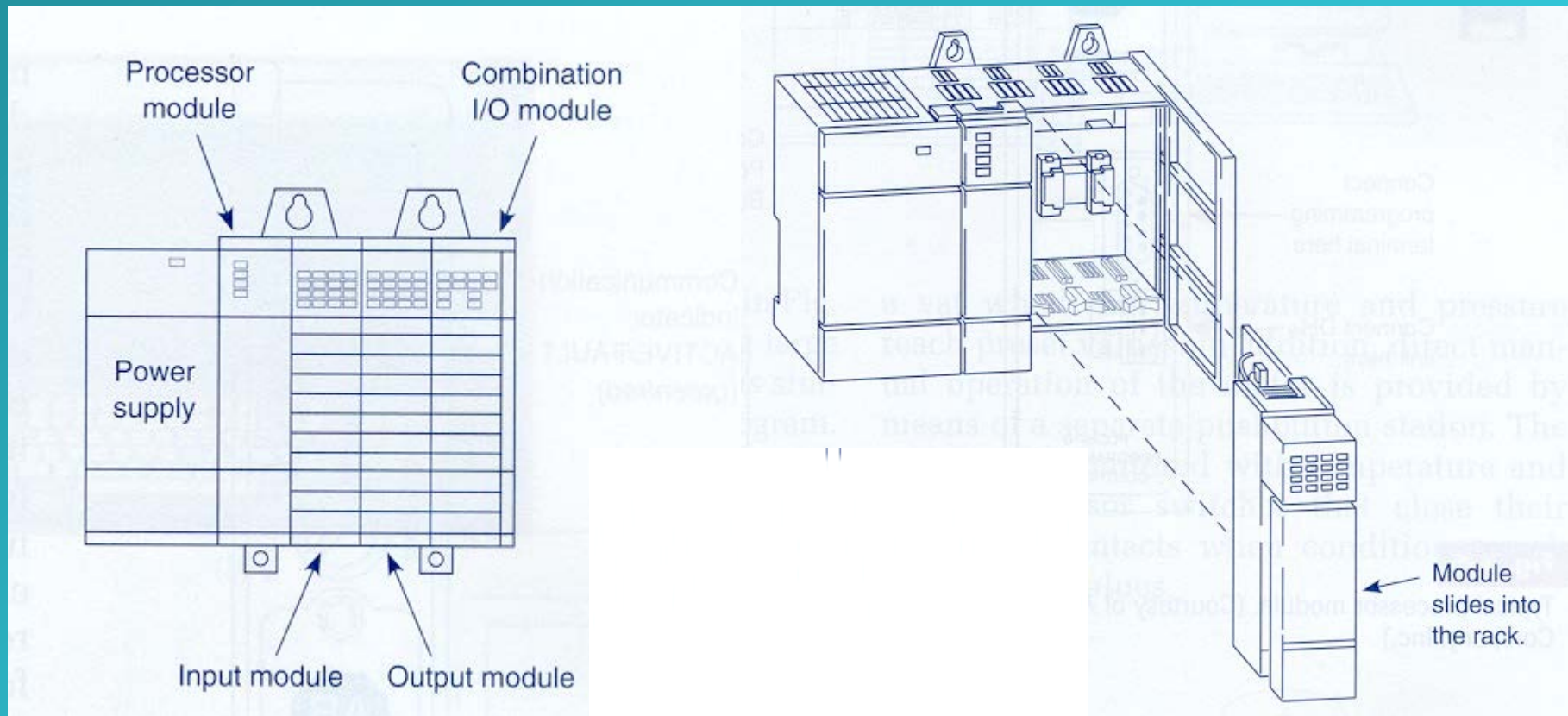


## Componentes constituintes dos PLCs

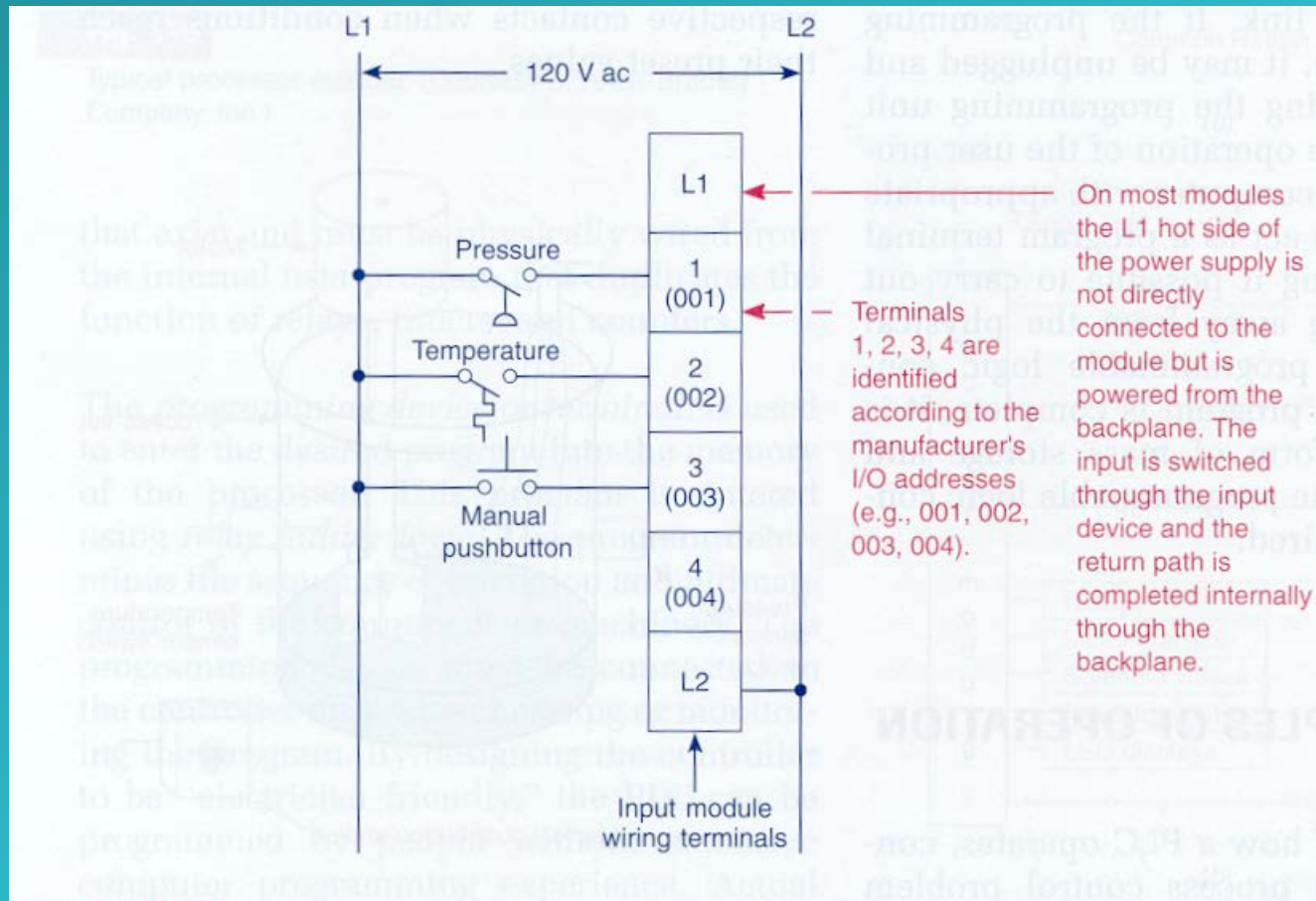


# Componentes constituintes dos PLCs

Sistemas modulares e sistemas fechados.



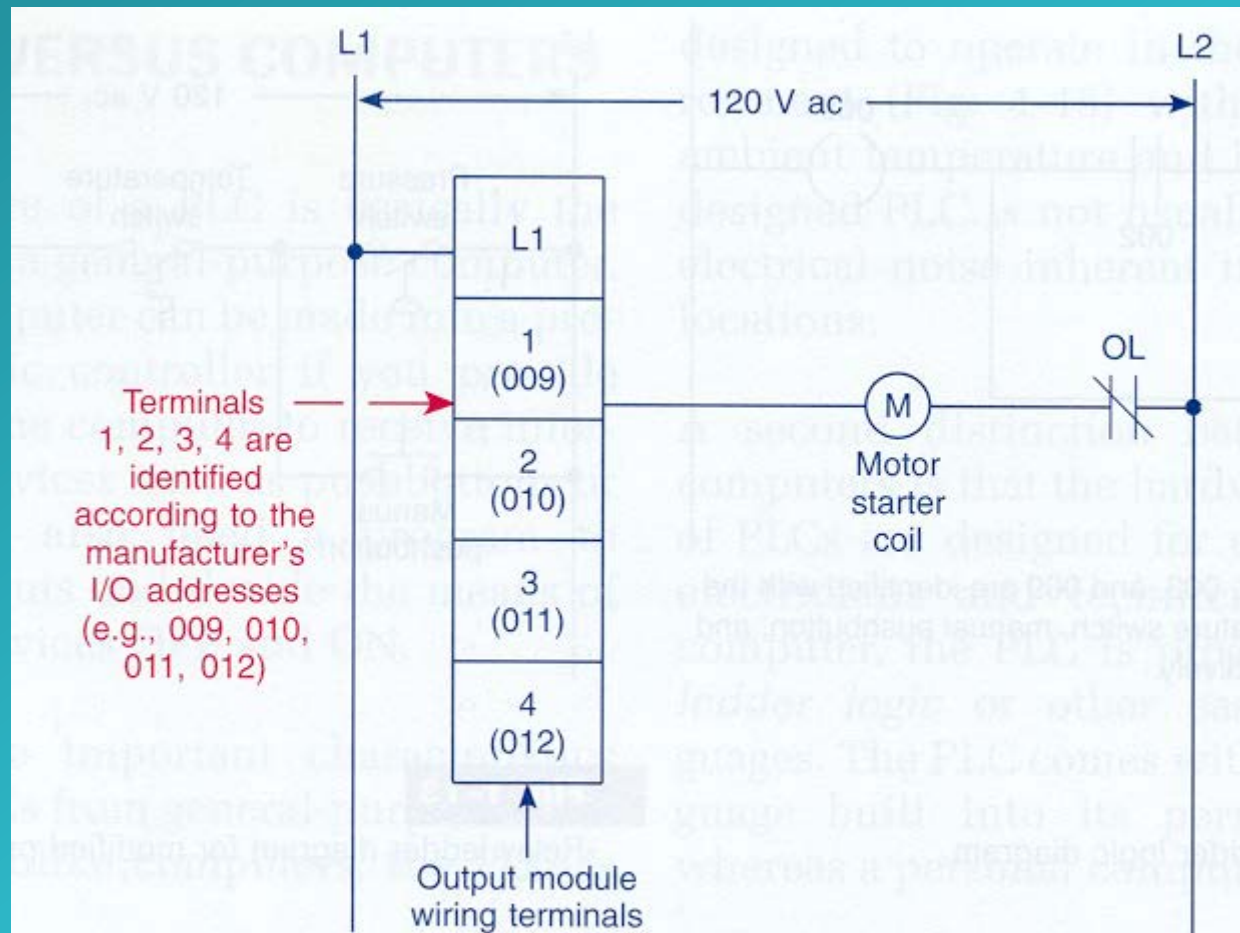
# Componentes constituintes dos PLCs



**Fig. 1-10**  
Typical input module wiring connections.



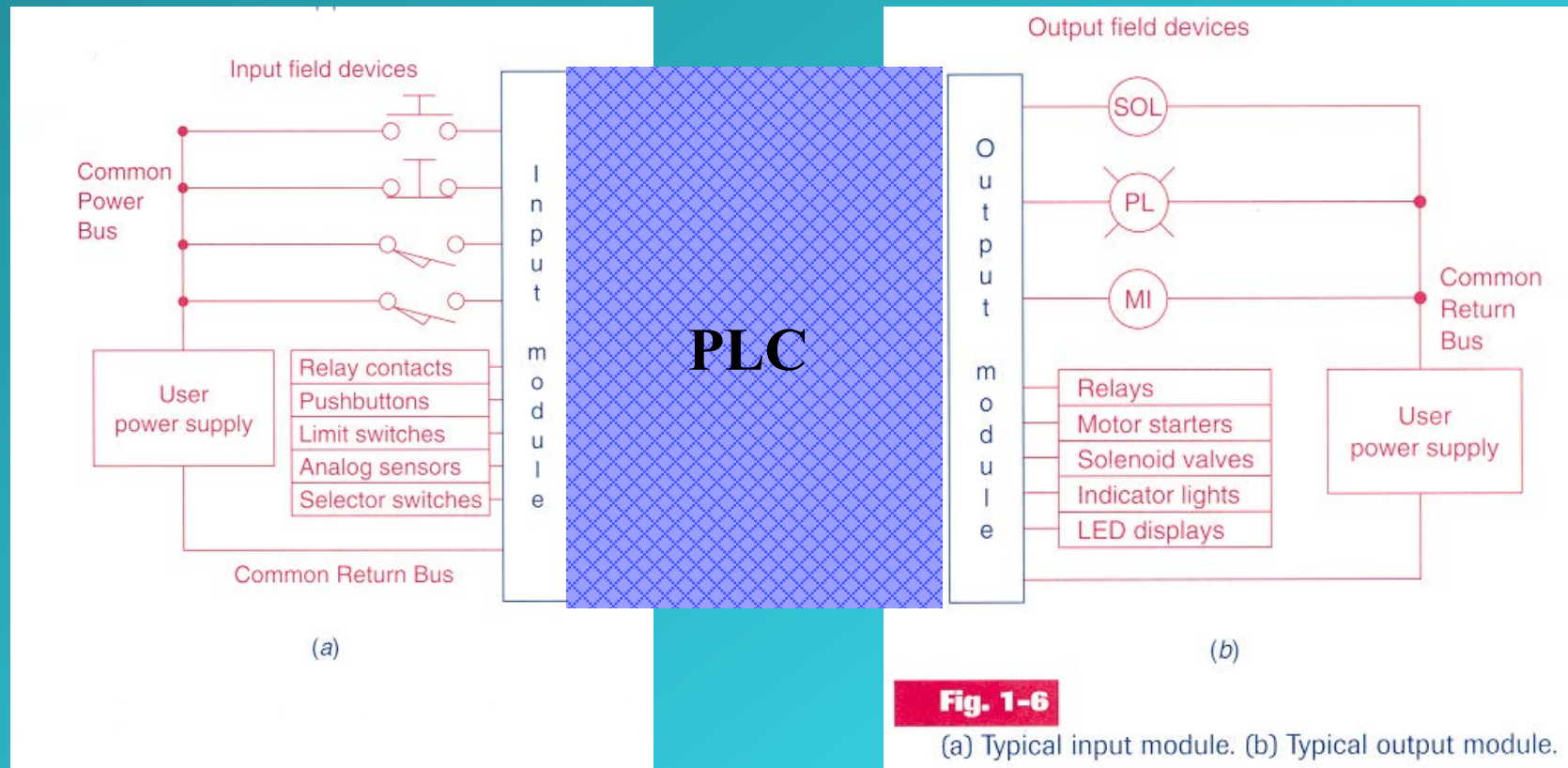
# Componentes constituintes dos PLCs



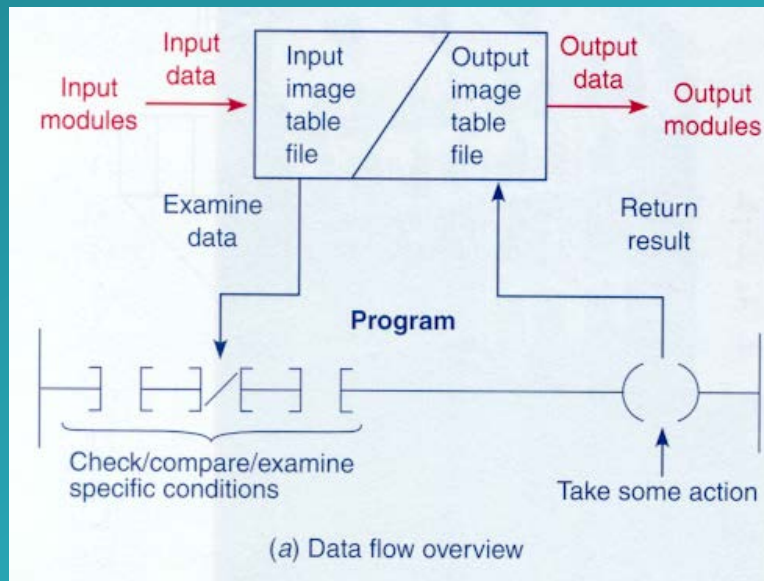
**Fig. 1-11**

Typical output module wiring connections.

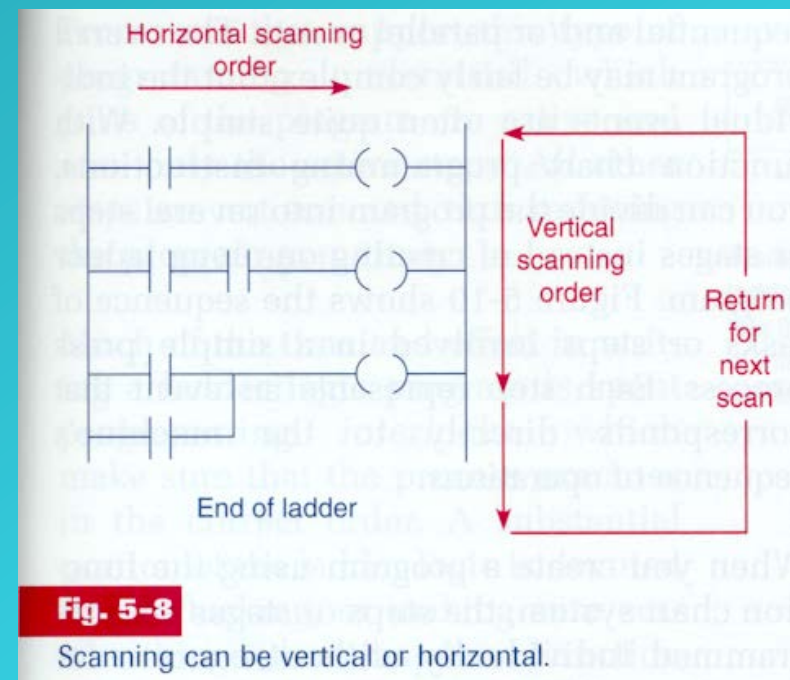
# Componentes constituintes dos PLCs



# Estrutura interna e funcionamento.



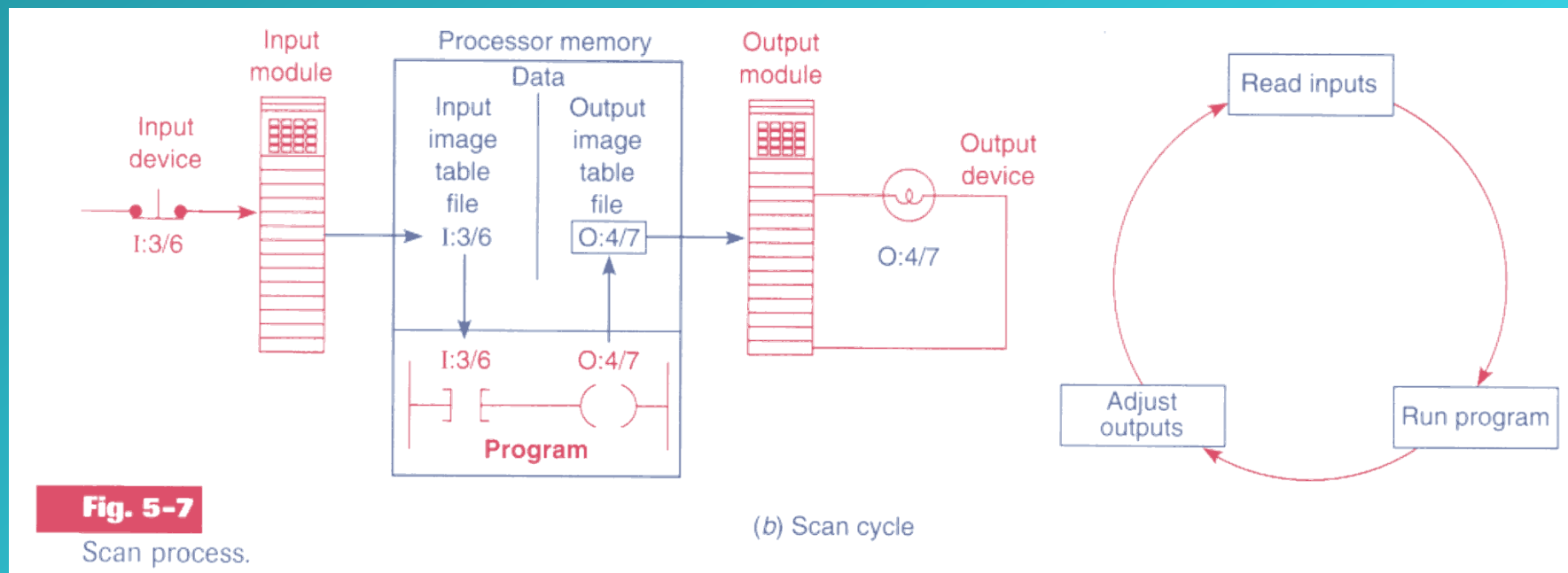
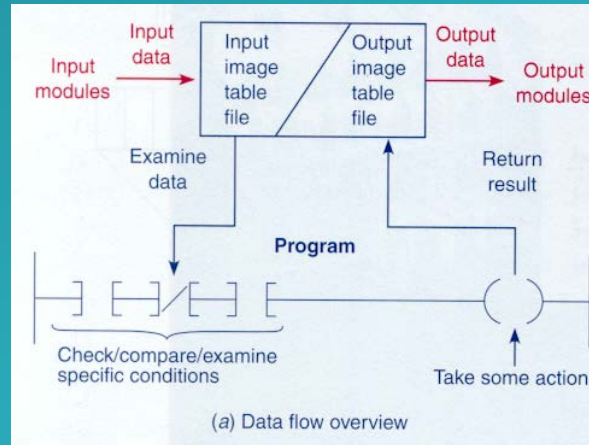
*Interface para as entradas e saídas*



*Scanning de escadas*



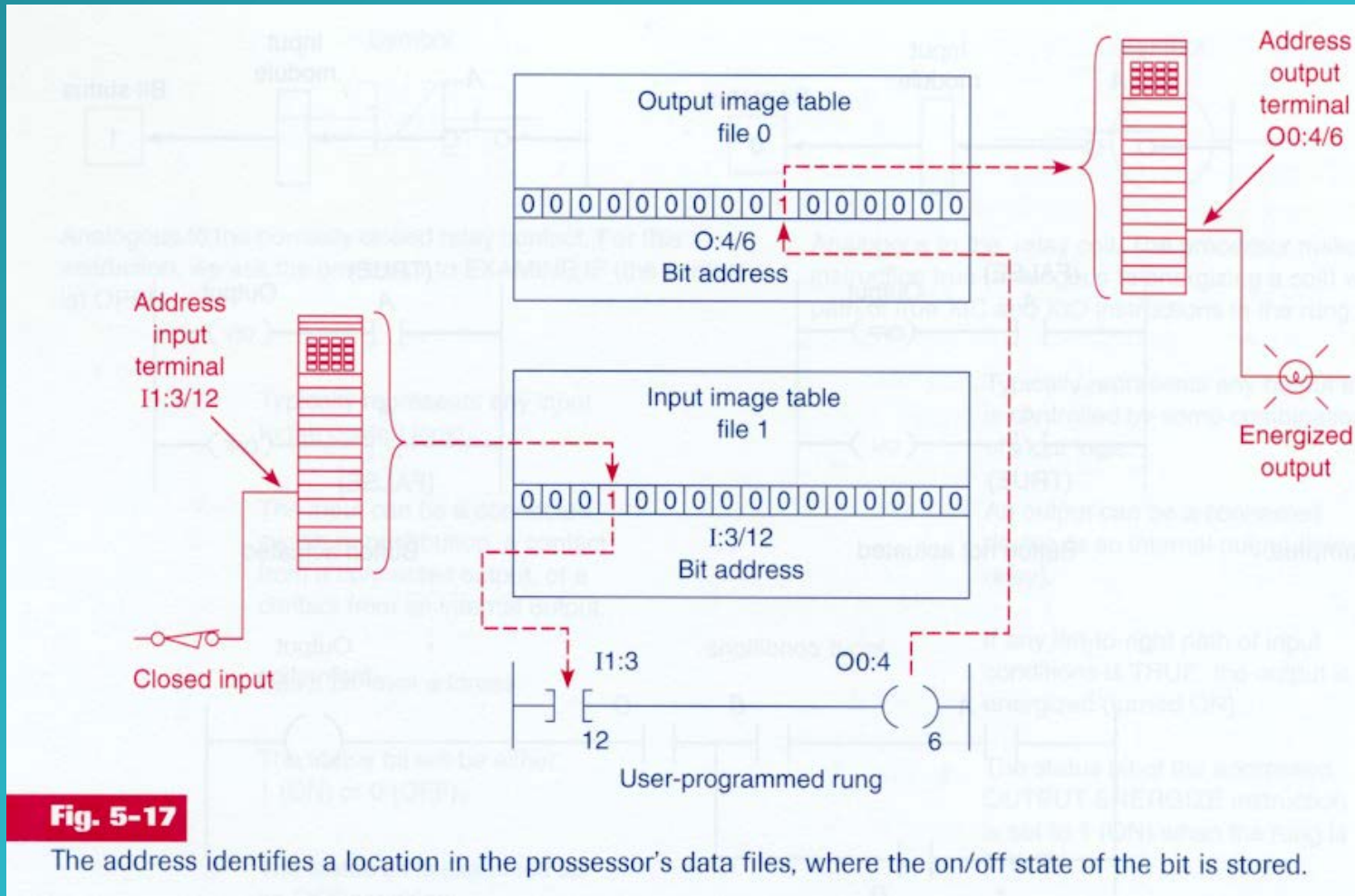
# Estrutura interna e funcionamento.



**Fig. 5-7**

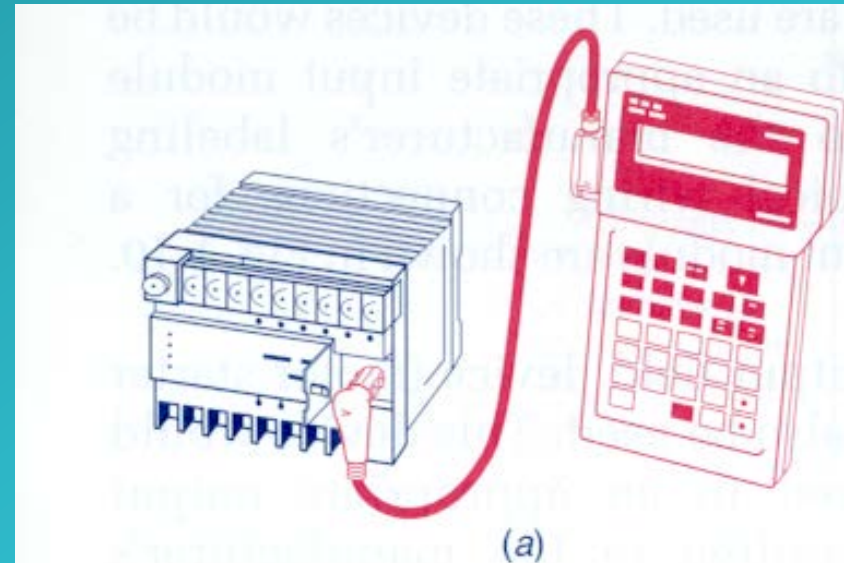
Scan process.

# Componentes constituintes dos PLCs



## Componentes constituintes dos PLCs

Programação utilizando dispositivos específicos



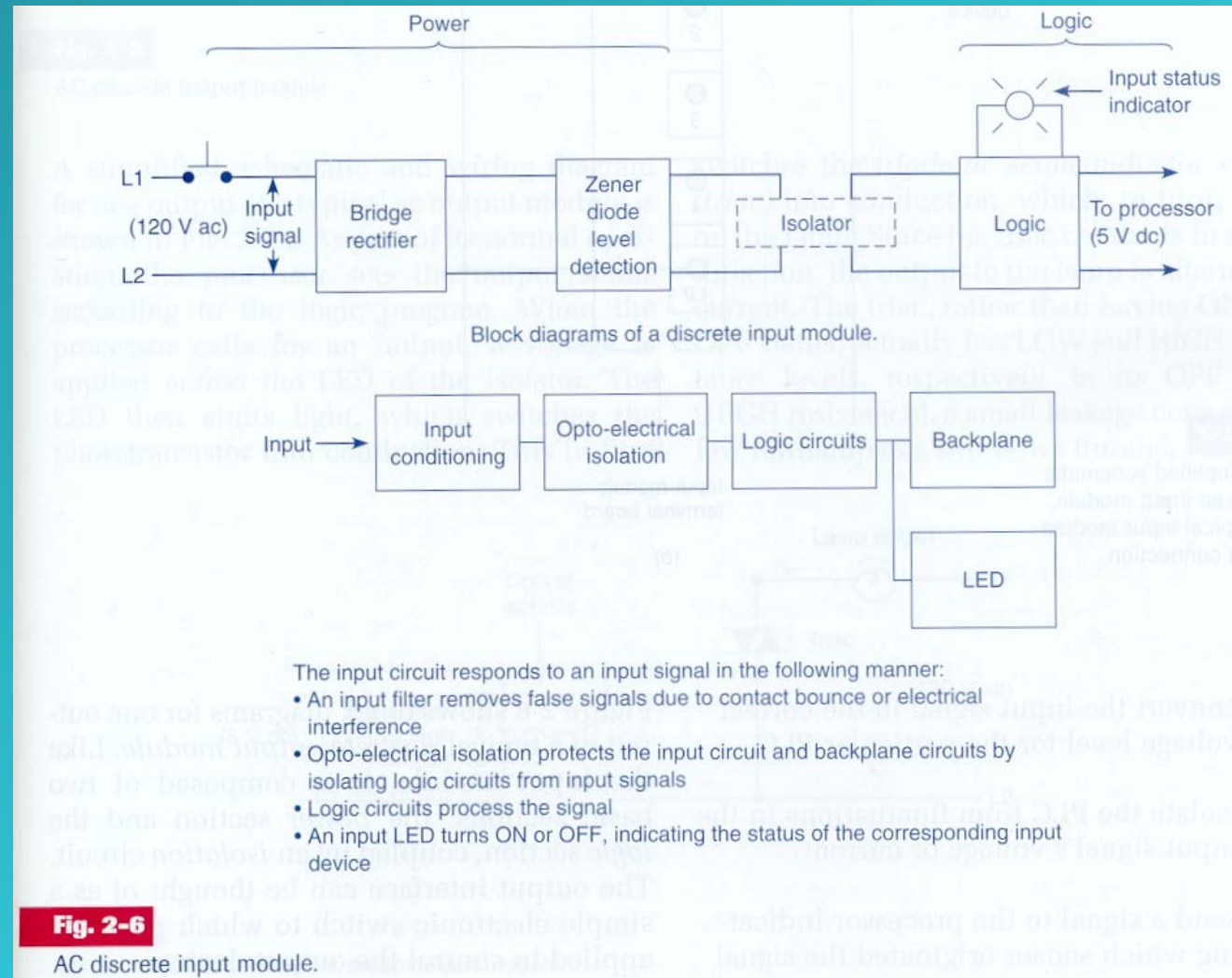
**Fig. 1-7**

Programming devices: (a) hand-held unit with light-emitting diode (LED) display; (b) industrial terminal video unit (Courtesy of Honeywell, Inc.); (c) personal computer with appropriate software.



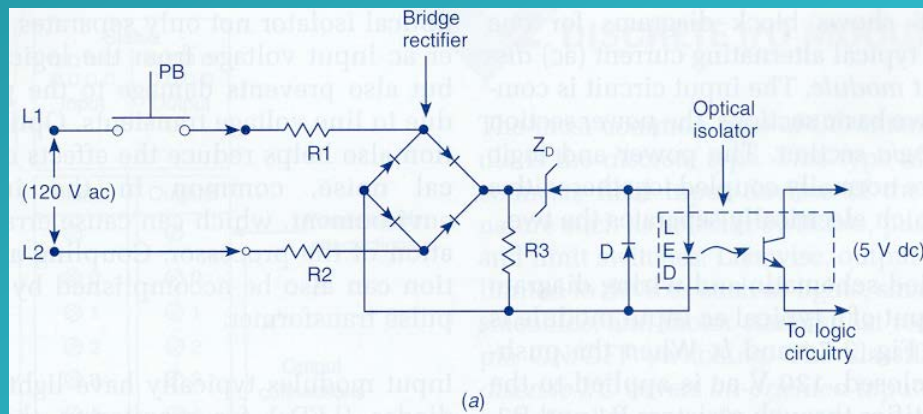
# Interfaces de entrada e de saída

## Módulo de entrada AC discreto

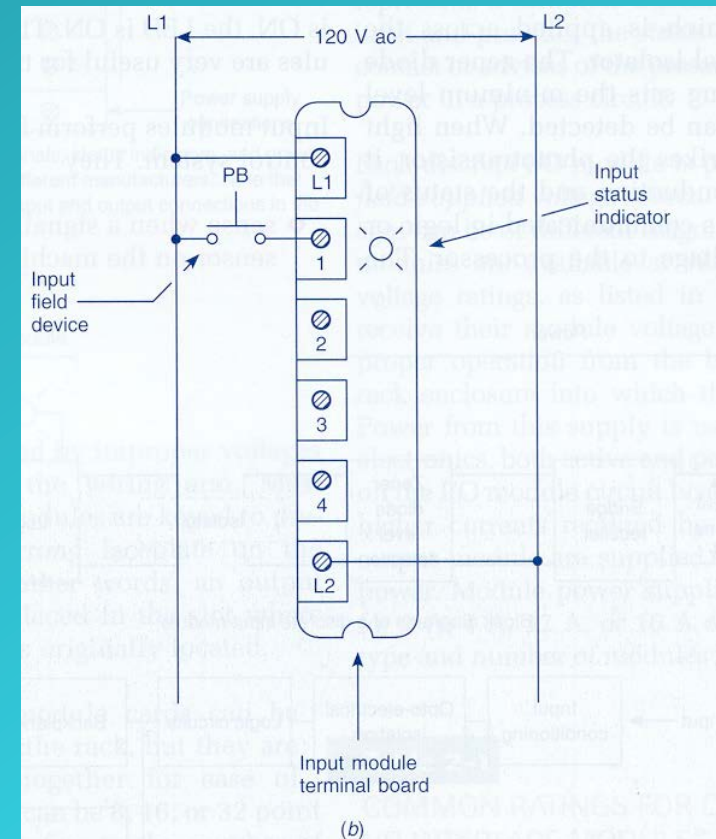


# Interfaces de entrada e de saída

Módulo de entrada  
AC discreto:  
implementação simplificada



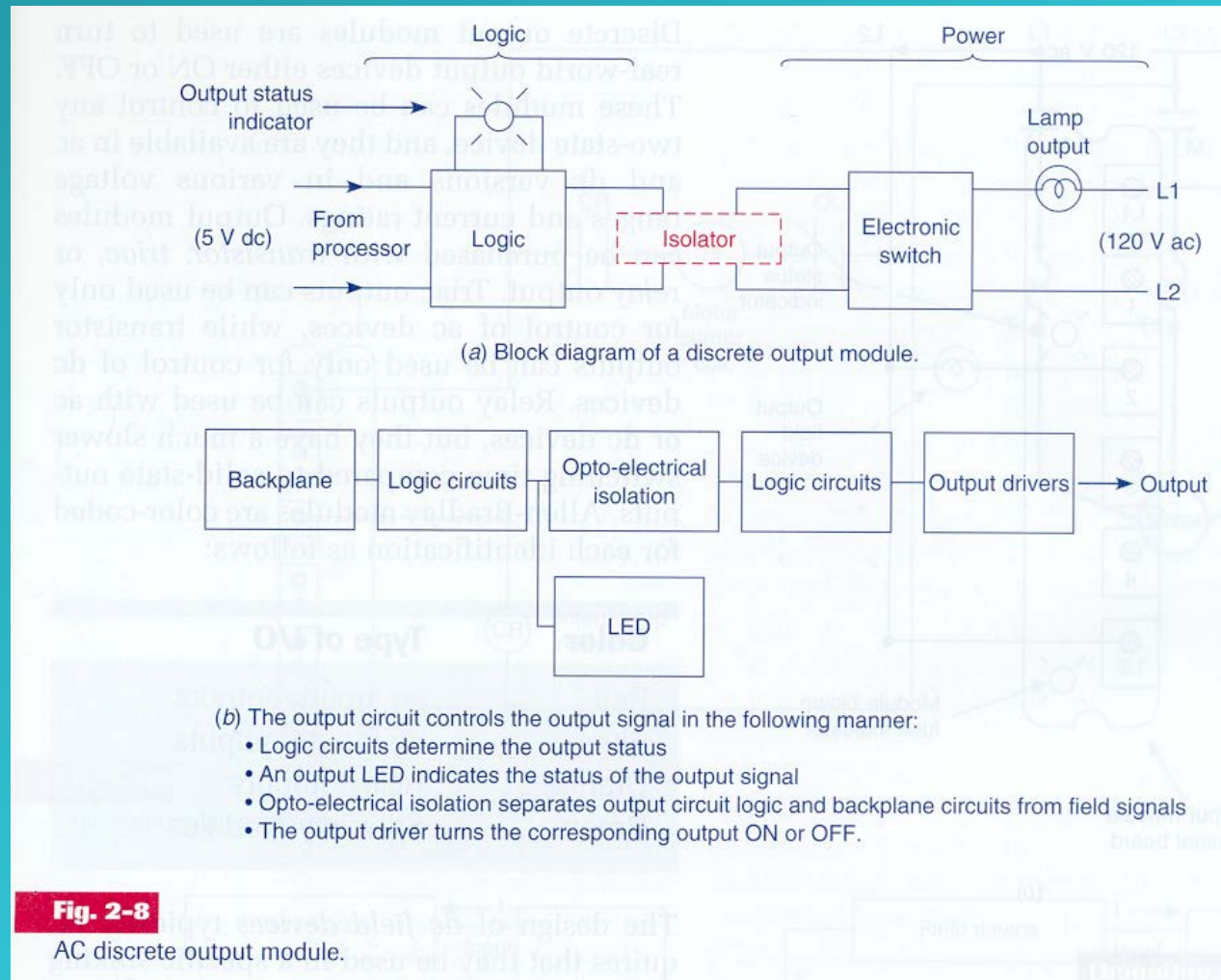
Circuito electrónico



Ligações ao terminais de um PLC

# Interfaces de entrada e de saída

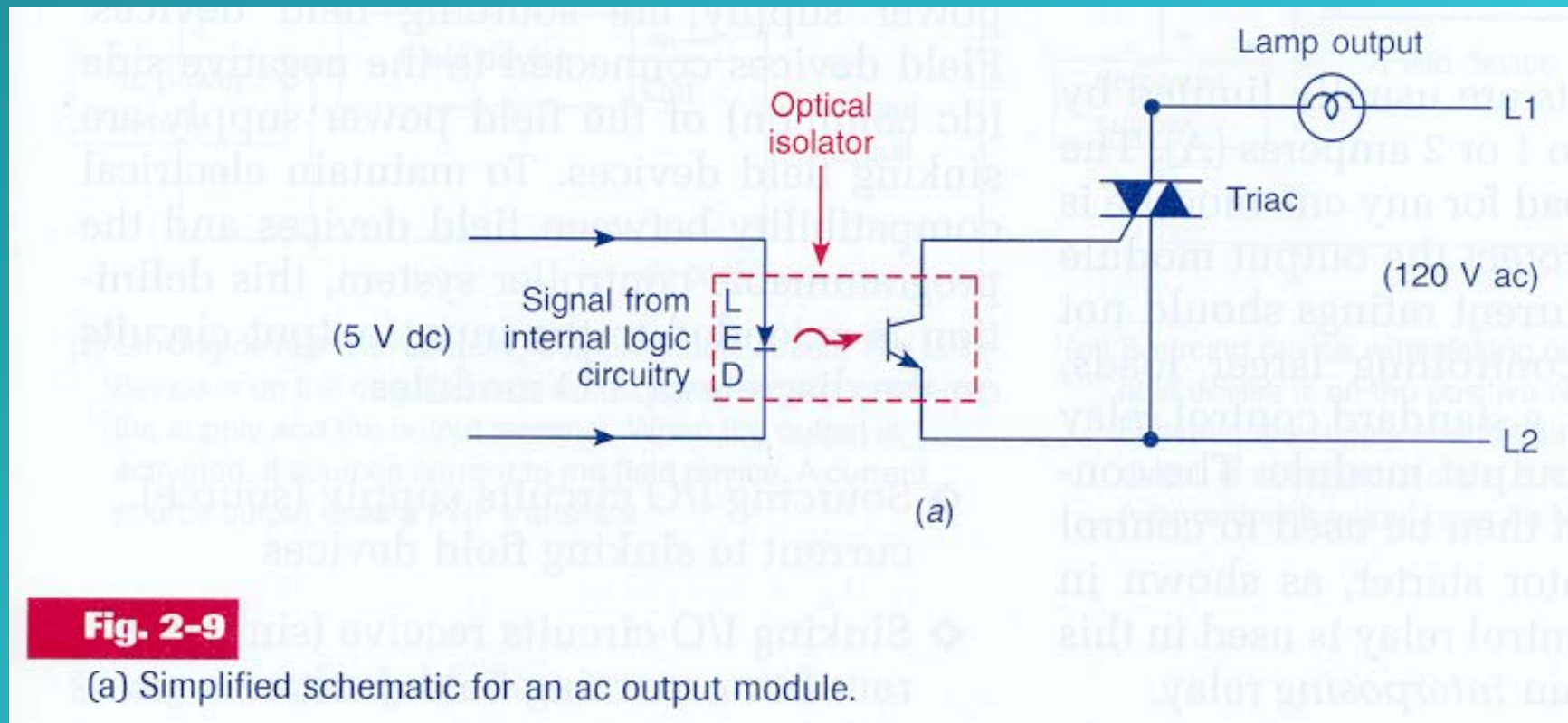
## Módulo de saída AC discreto





# Interfaces de entrada e de saída

## Módulo de saída AC discreto



Circuito electrónico

# Automação Industrial baseada em PLCs

## 4ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 4ª Aula

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.

### Cap. 3 - Linguagens de Programação de PLCs

- Diagramas de contactos (ladder diagram).
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.



# Diagramas de contactos (ladder diagram)

*Ladder Diagram*

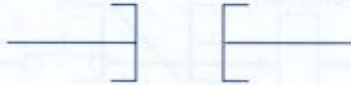
ou

**Diagramas em Escada**

ou


**Diagramas de Contactos**

Symbol




Analogous to the normally open relay contact. For this instruction, we ask the processor to EXAMINE IF (the contact is) CLOSED.

Symbol



Analogous to the normally closed relay contact. For this instruction, we ask the processor to EXAMINE IF (the contact is) OPEN.

Symbol

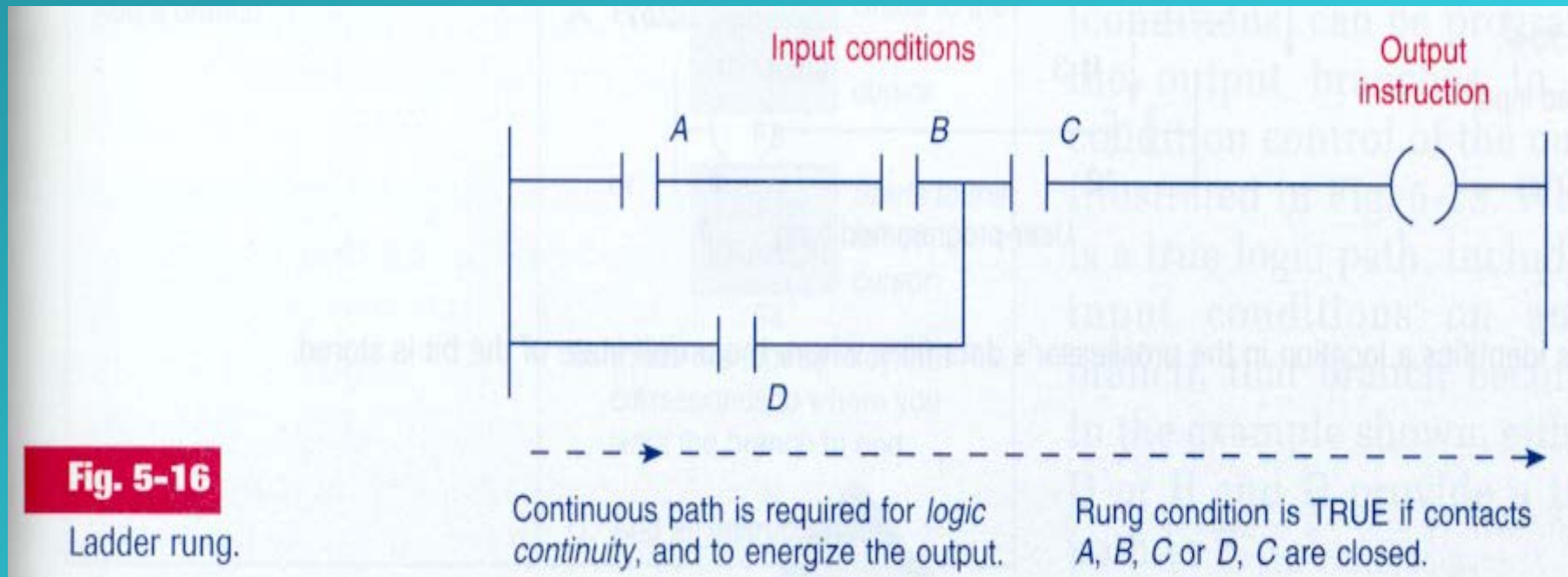


Analogous to the relay coil. The processor makes this instruction true (analogous to energizing a coil) when there is a path of true XIC and XIO instructions in the rung.

## Diagramas de contactos (cont.)

### *Instruções tipo relé*

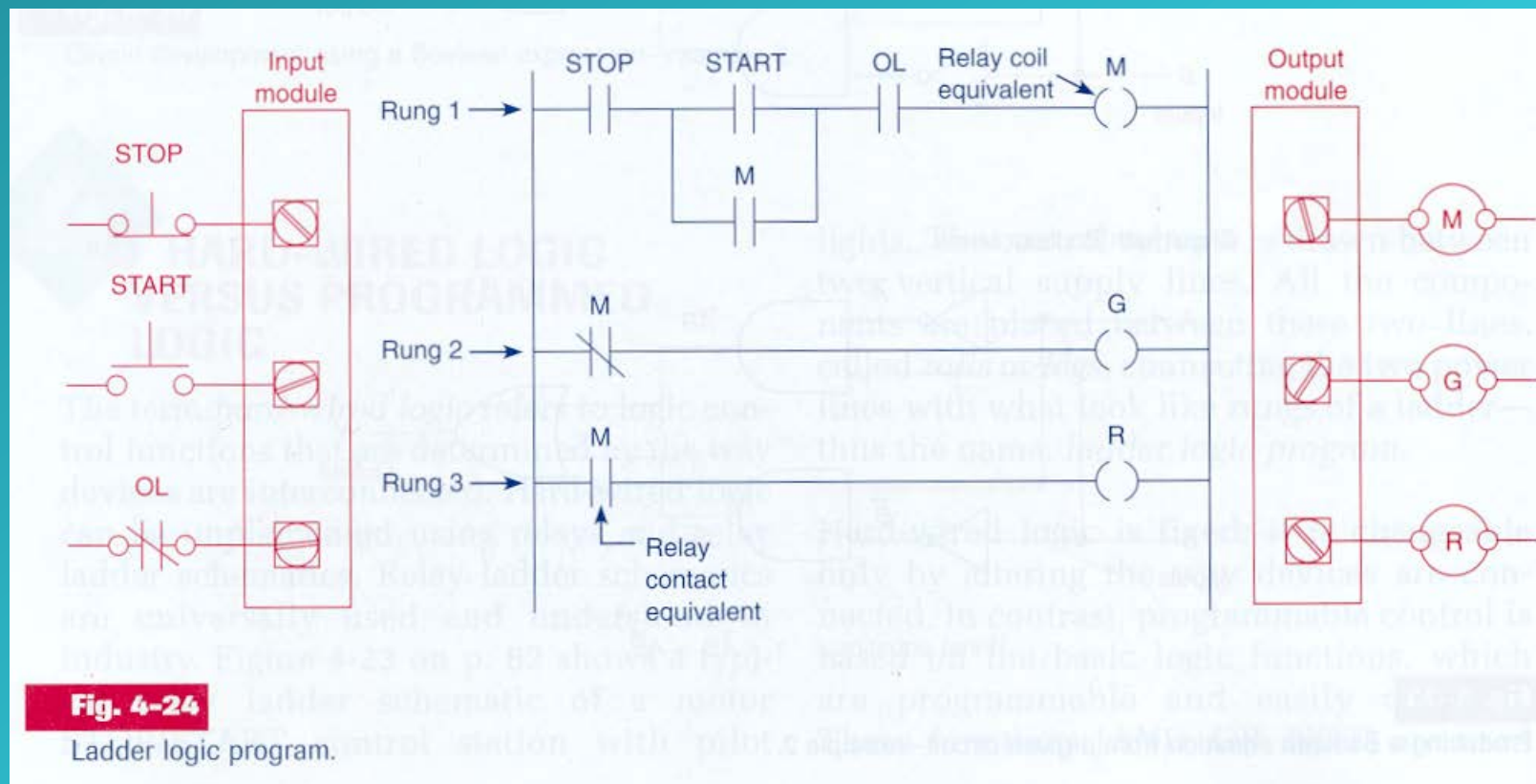
### *Exemplo:*



**Fig. 5-16**

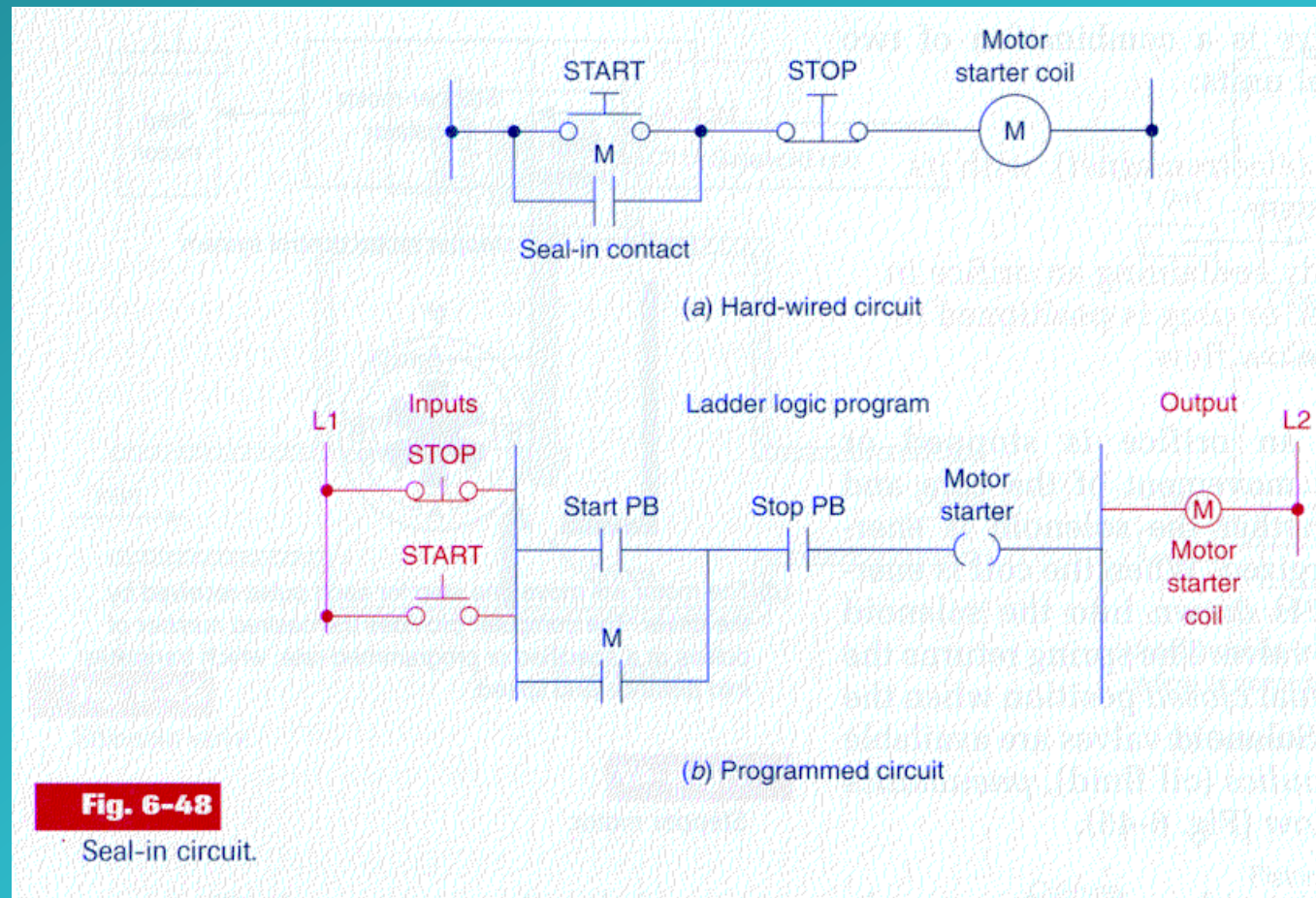
Ladder rung.

# Diagramas de contactos (cont.)

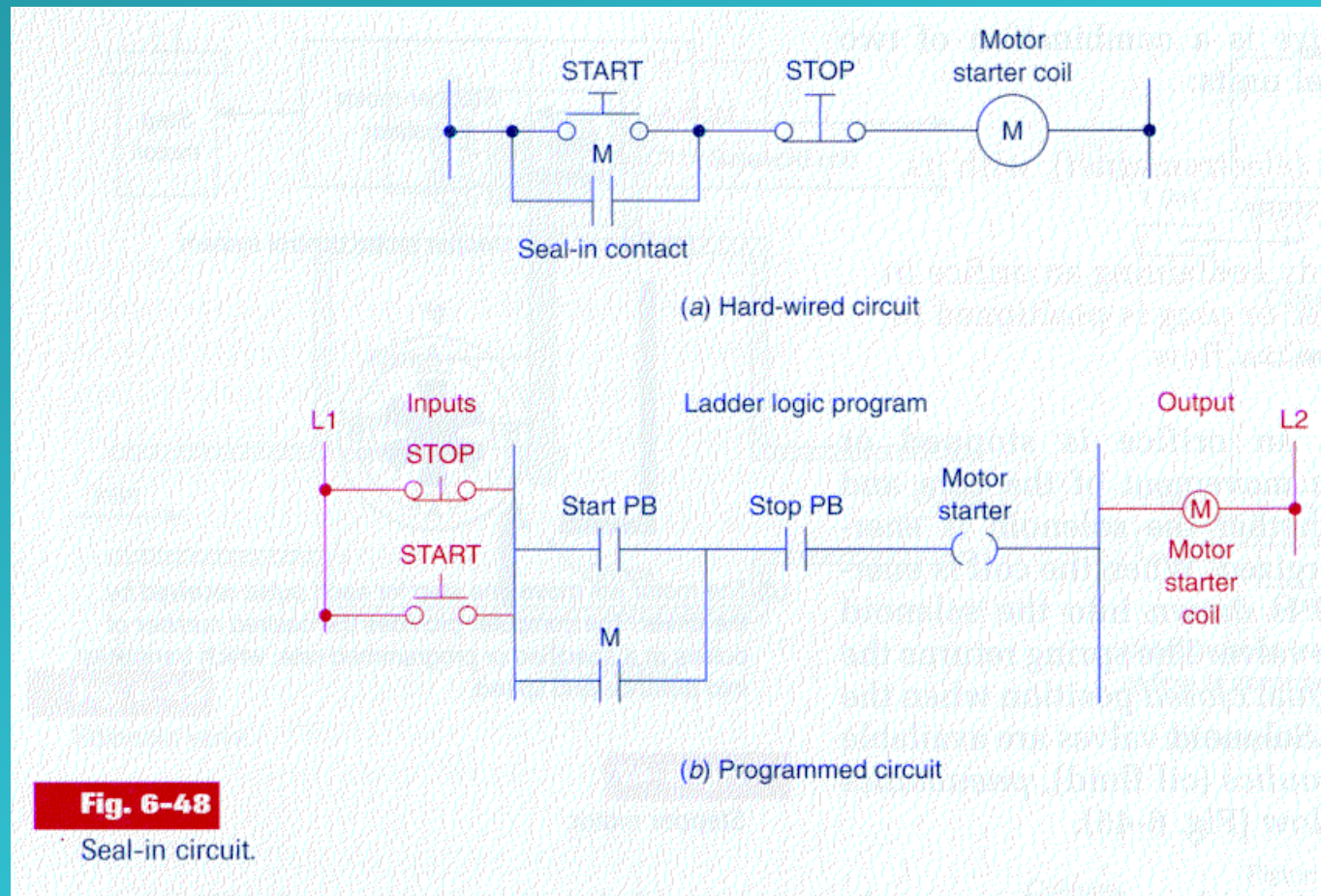




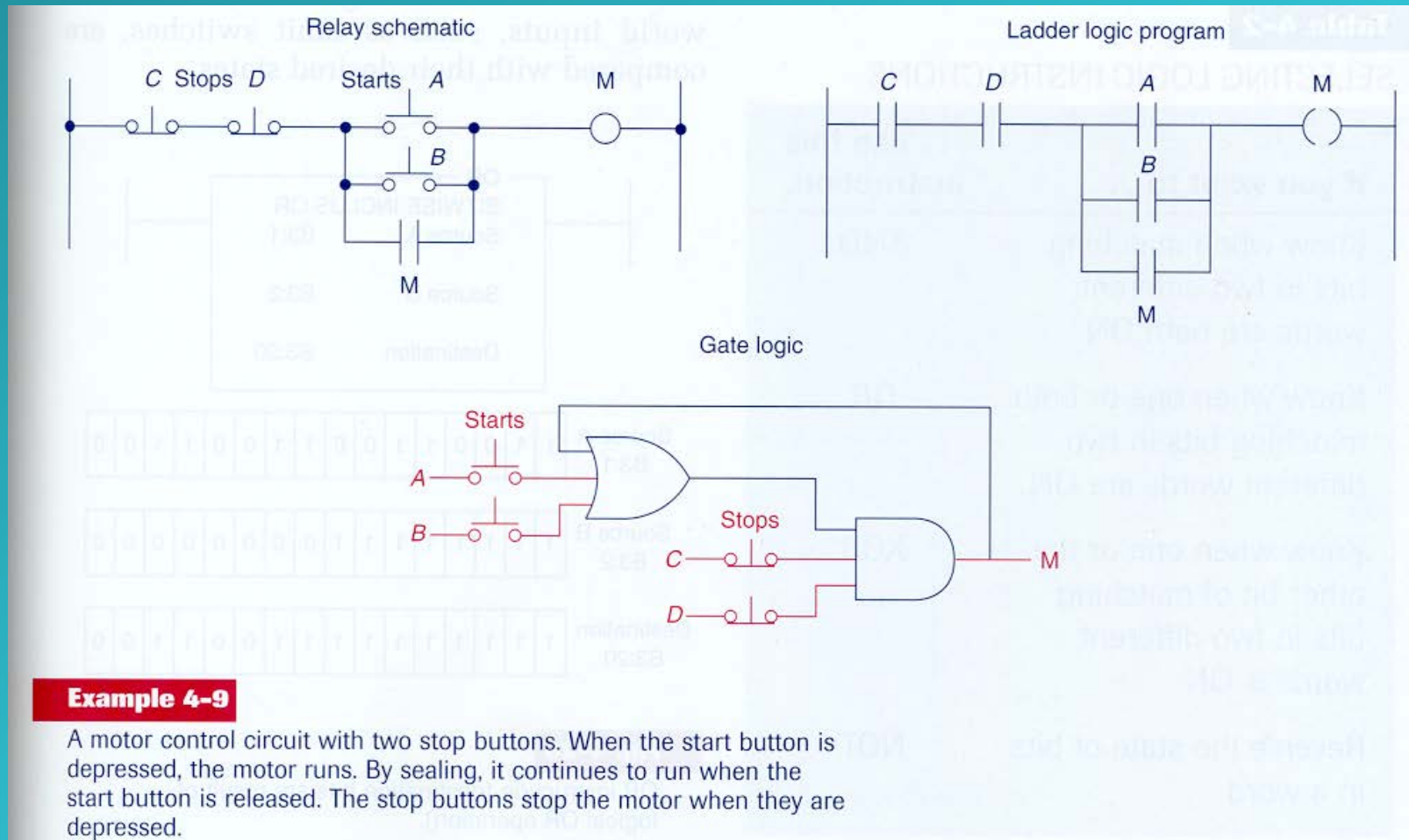
# Diagramas de contactos (cont.)



# Diagramas de contactos (cont.)



# Diagramas de contactos (cont.)



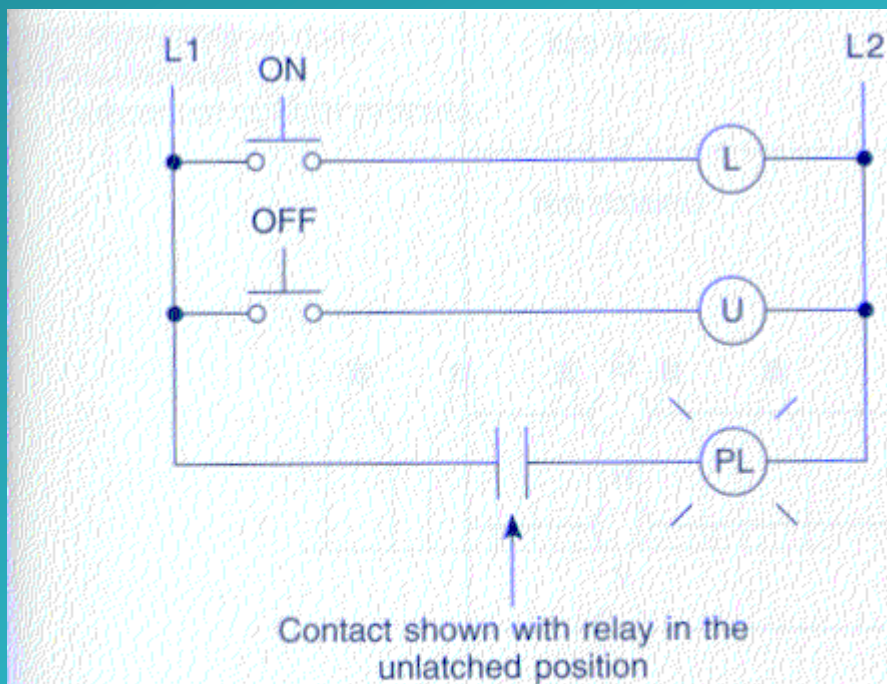
### Example 4-9

A motor control circuit with two stop buttons. When the start button is depressed, the motor runs. By sealing, it continues to run when the start button is released. The stop buttons stop the motor when they are depressed.

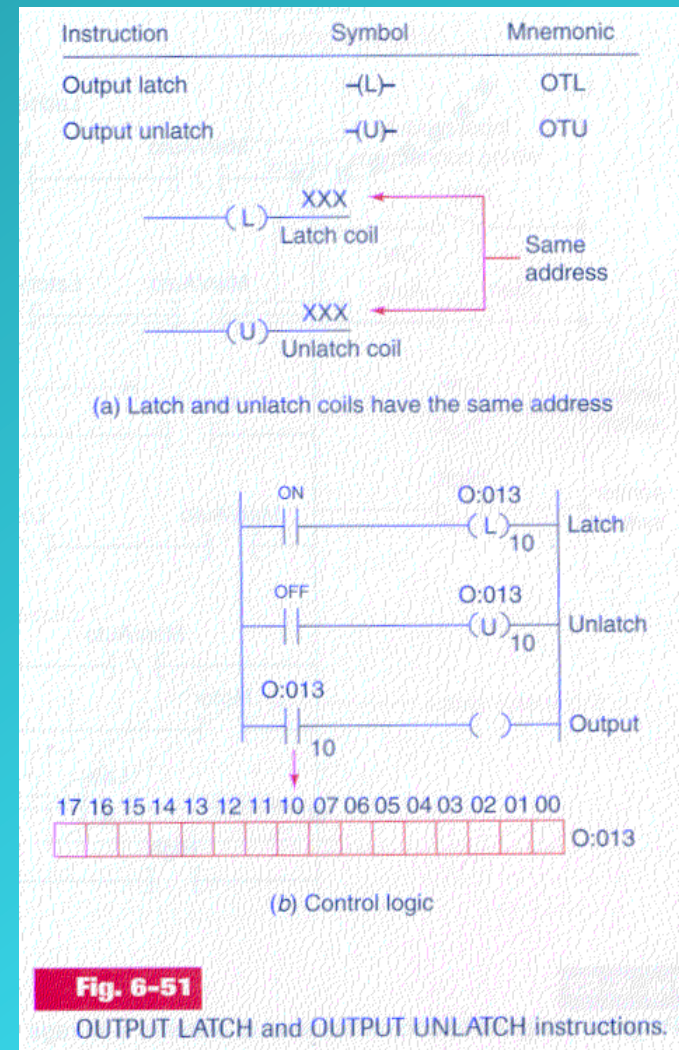


# Diagramas de contactos (cont.)

Relés com *latch* e *unlatch*



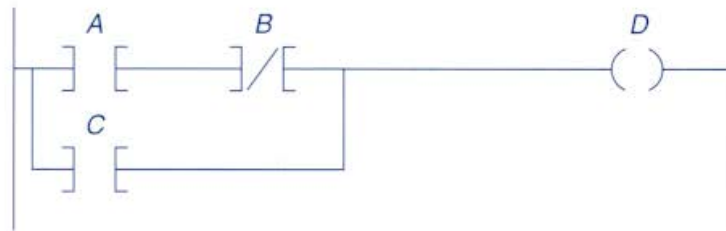
**Fig. 6-50**  
Schematic of electromagnetic latching relay.



**Fig. 6-51**  
OUTPUT LATCH and OUTPUT UNLATCH instructions.

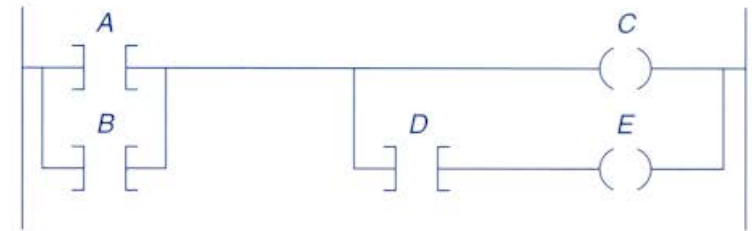
## Diagramas de contactos (cont.)

### Caso Geral Entradas e Saídas em paralelo, com derivações



**Fig. 5-21**

Parallel input branching.



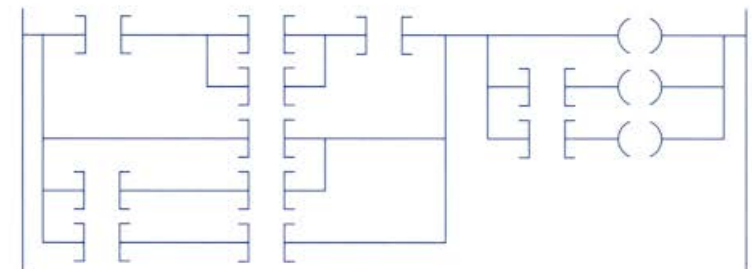
**Fig. 5-23**

Parallel output branching with conditions.



**Fig. 5-22**

Parallel output branching.

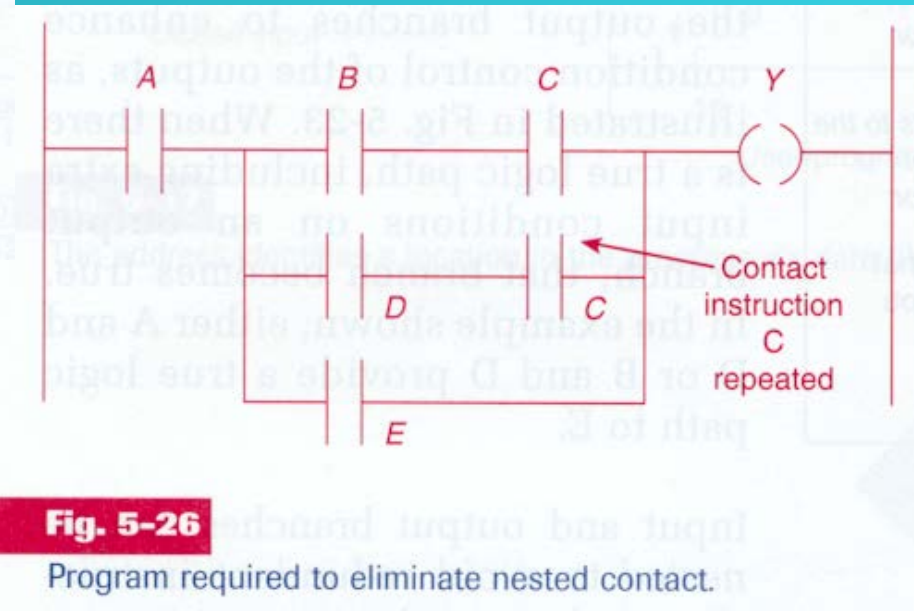
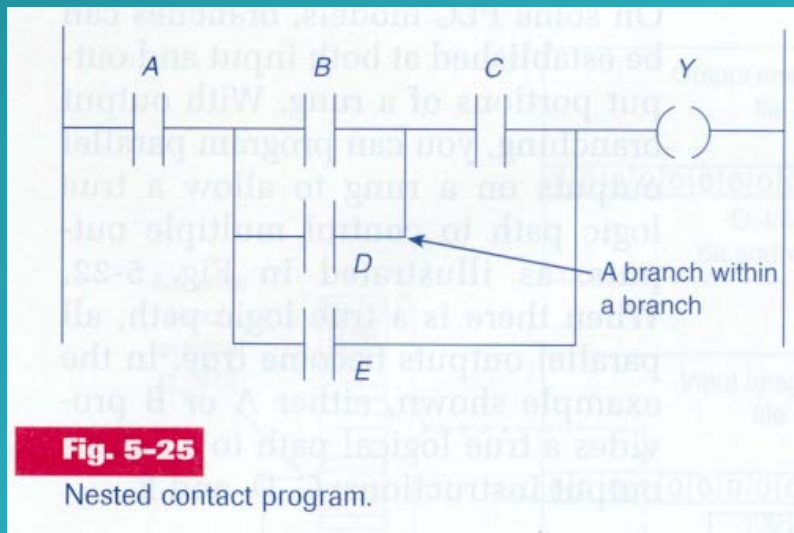


**Fig. 5-24**

Nested input and output branches.

## Diagramas de contactos (cont.)

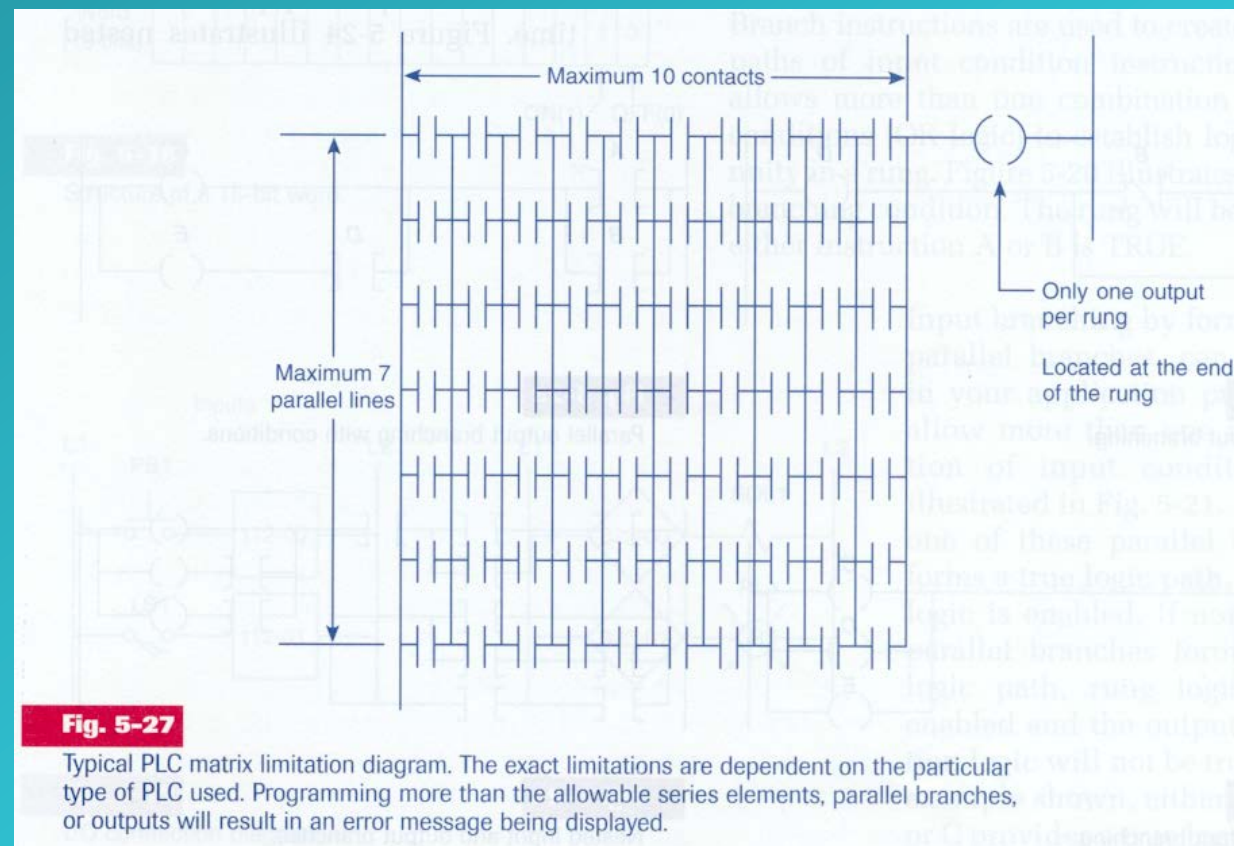
### Contactos imbrincados e solução alternativa





### Diagramas de contactos (cont.)

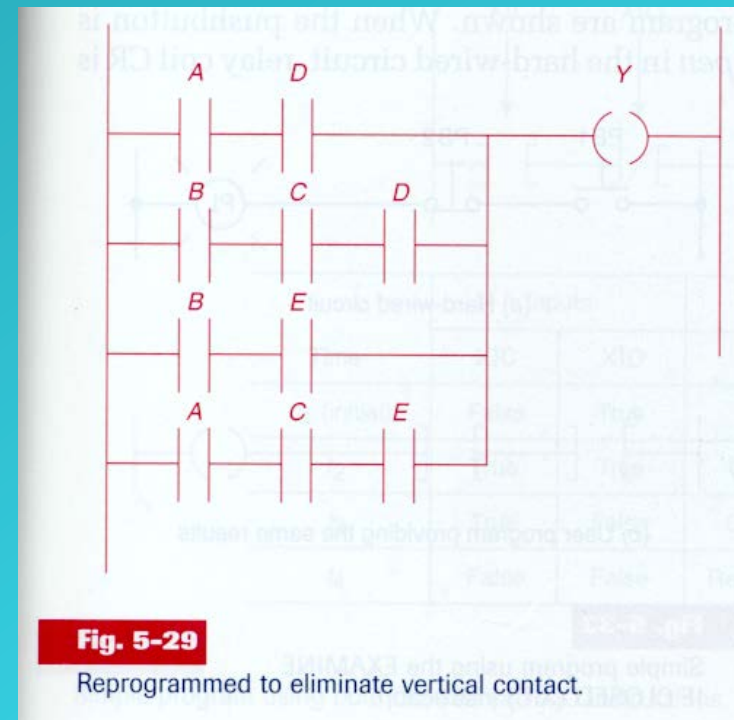
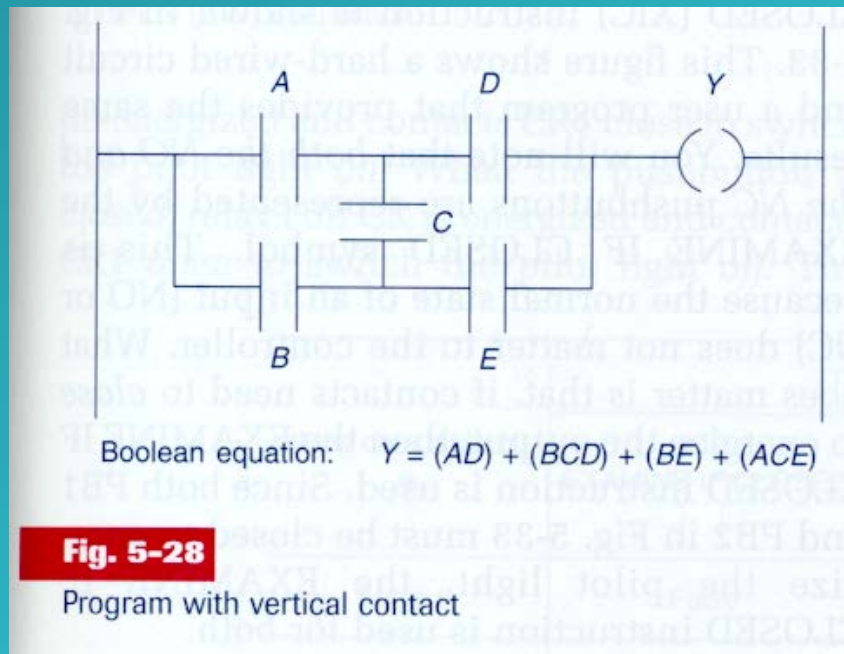
No caso prático de cada PLC existem limitações nas matrizes de ligações a especificar.



## Diagramas de contactos (cont.)

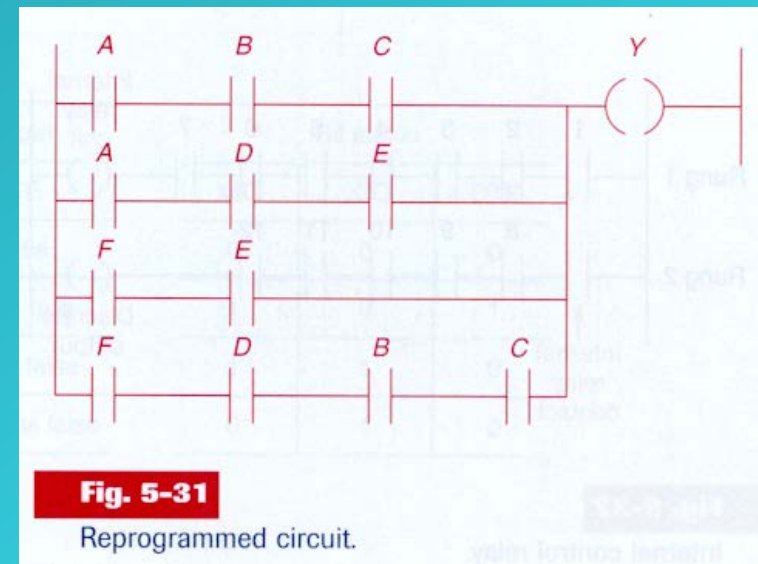
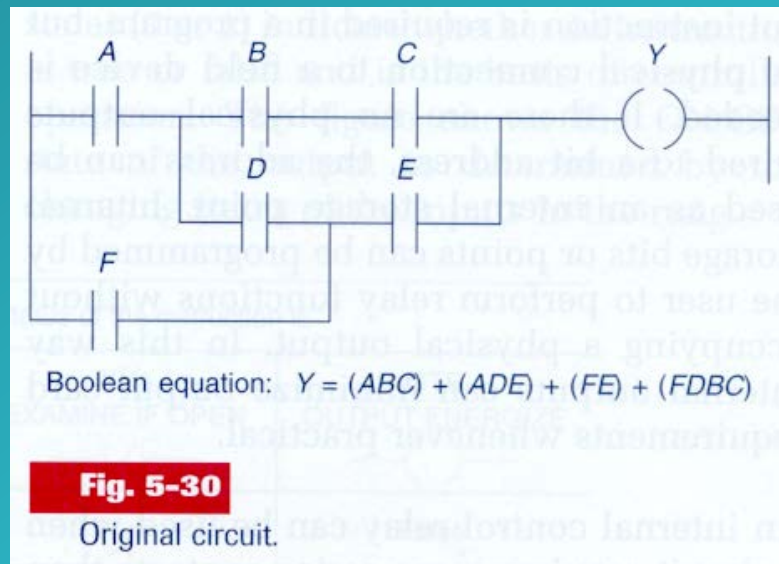
### Contactos verticais e solução alternativa

Exemplo:



## Diagramas de contactos (cont.)

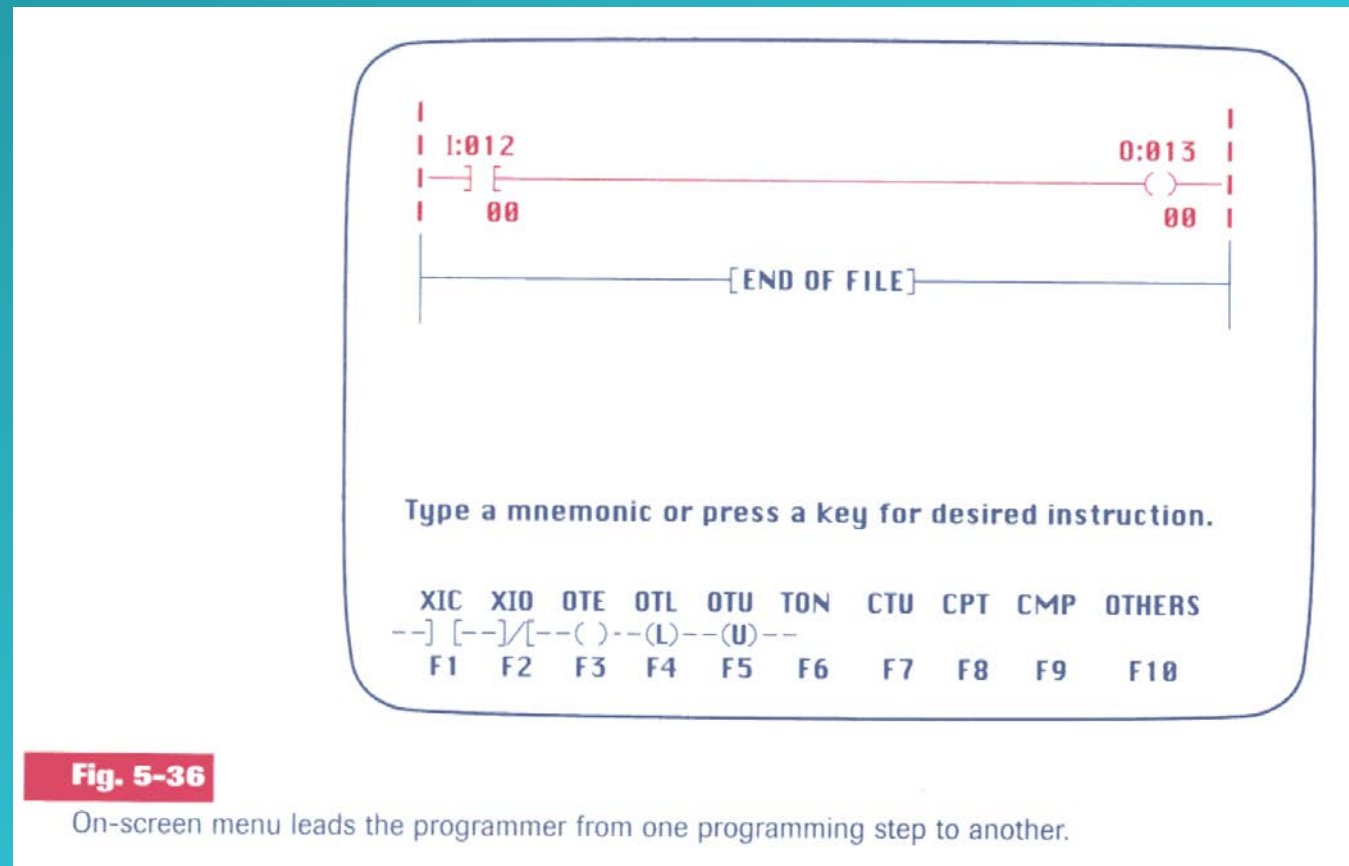
Contactos verticais e solução alternativa  
outro exemplo...





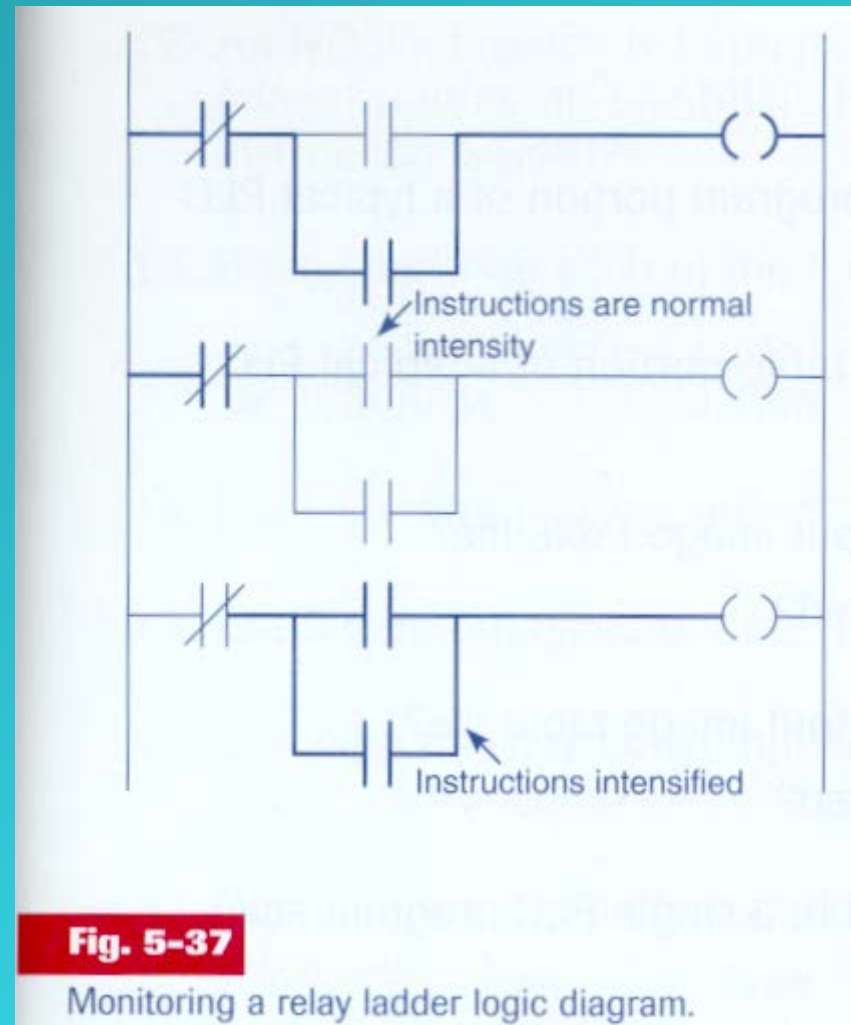
## Diagramas de contactos (cont.)

Ferramentas para escrita de programas e ...



## Diagramas de contactos (cont.)

Ferramentas para  
debug de programas



# Automação Industrial baseada em PLCs

5ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)



## 5ª Aula

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.

### Cap. 3 - Linguagens de Programação de PLCs

- Diagramas de contactos (ladder diagram).
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

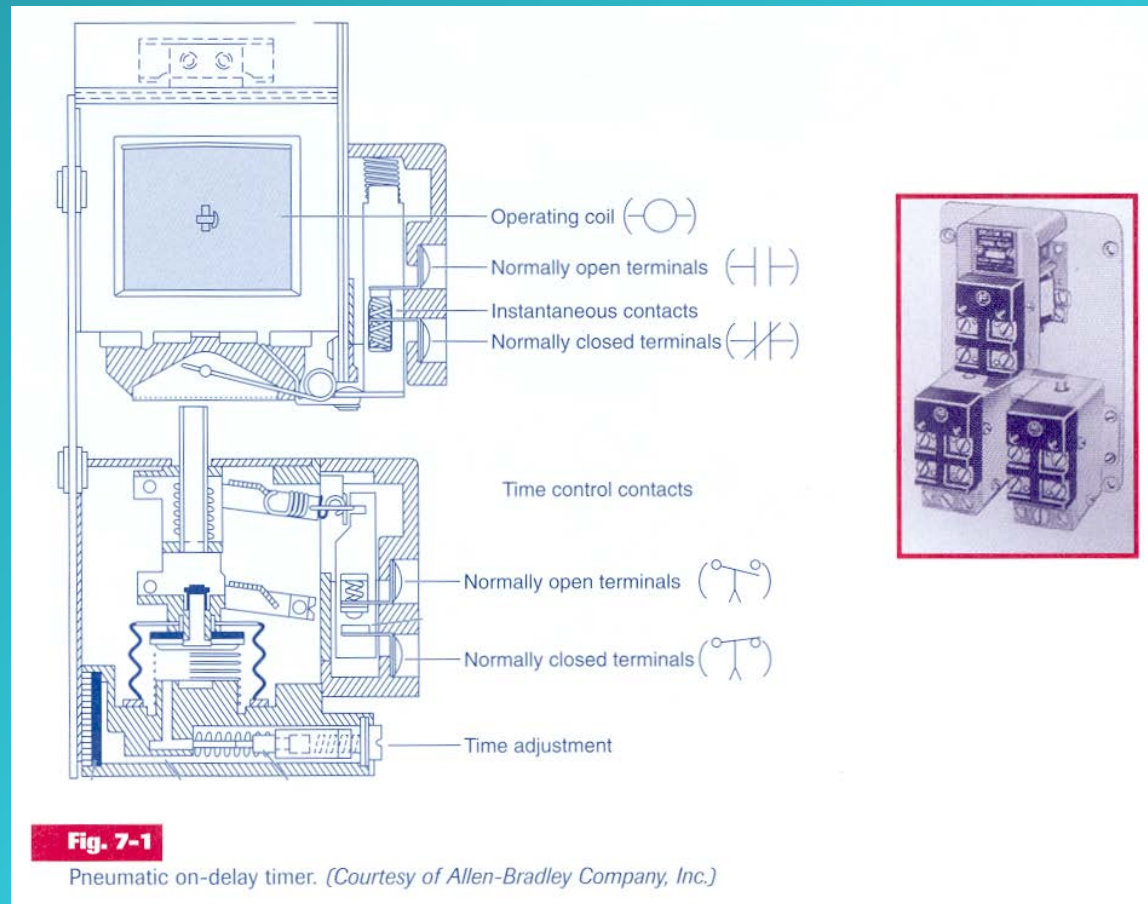
### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.

# Diagramas de contactos (cont.)

## *Timers*

## Relés temporizados

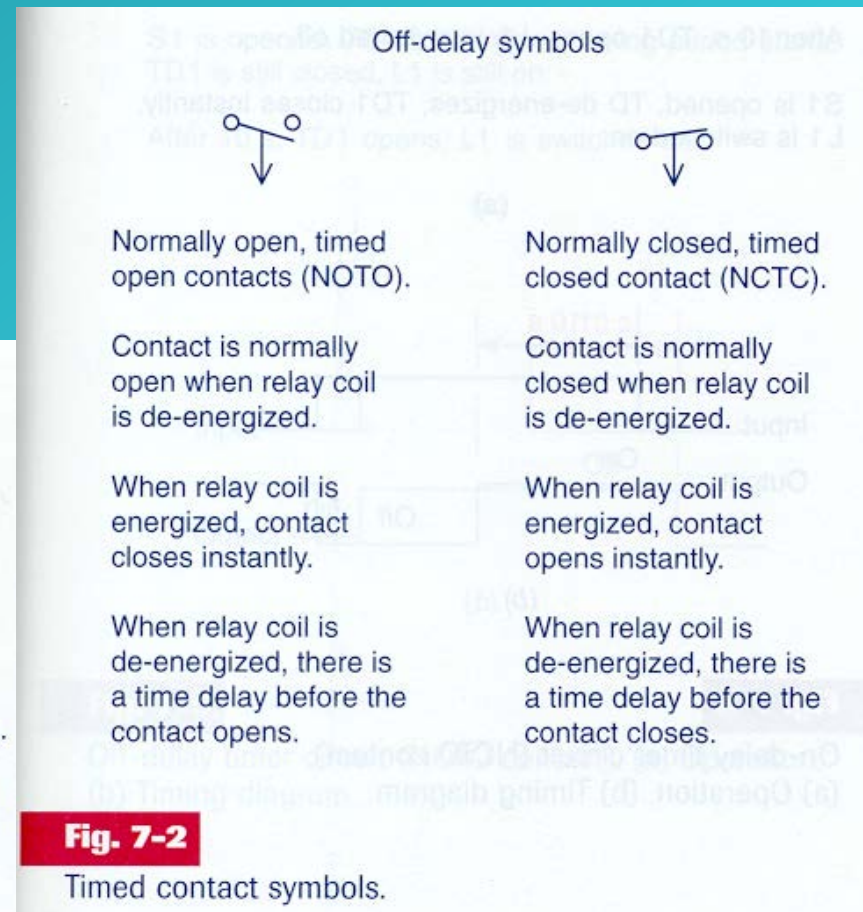
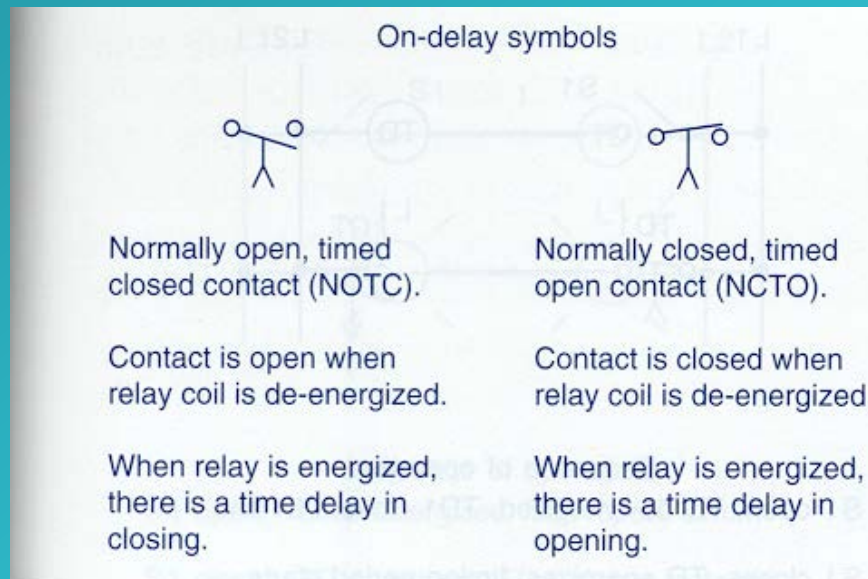


# Diagramas de contactos (cont.)

## Timers

### Relés temporizados

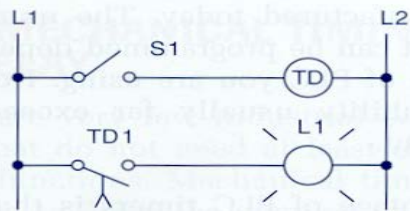
Símbolos utilizados





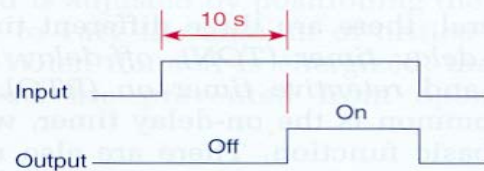
# Diagramas de contactos (cont.)

## Timers: exemplos



Sequence of operation:  
 S1 open, TD de-energized, TD1 open, L1 off.  
 S1 closes, TD energizes, timing period starts, TD1 is still open, L1 is still off.  
 After 10 s, TD1 closes, L1 is switched on.  
 S1 is opened, TD de-energizes, TD1 opens instantly, L1 is switched off.

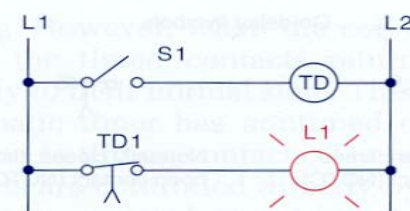
(a)



(b)

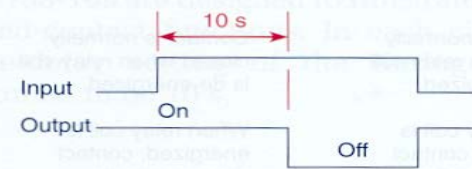
**Fig. 7-3**

On-delay timer circuit (NOTC contact). (a) Operation. (b) Timing diagram.



Sequence of operation:  
 S1 open, TD de-energized, TD1 closed, L1 on.  
 S1 closes, TD energizes, timing period starts, TD1 is still closed, L1 is still on.  
 After 10 s, TD1 opens, L1 is switched off.  
 S1 is opened, TD de-energizes, TD1 closes instantly, L1 is switched on.

(a)



(b)

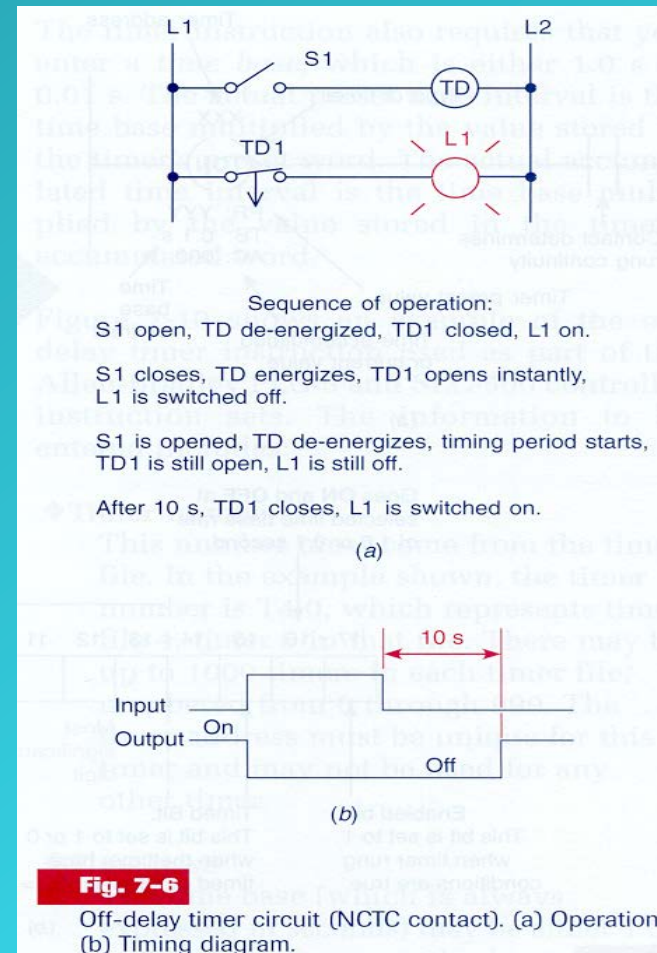
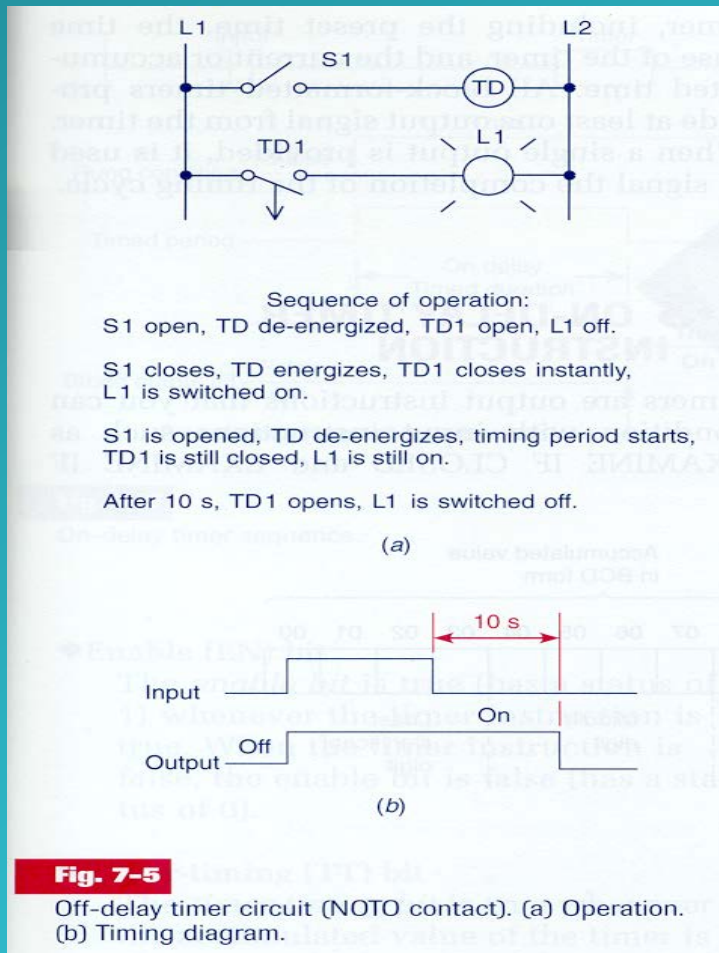
**Fig. 7-4**

On-delay timer circuit (NCTO contact). (a) Operation. (b) Timing diagram.



# Diagramas de contactos (cont.)

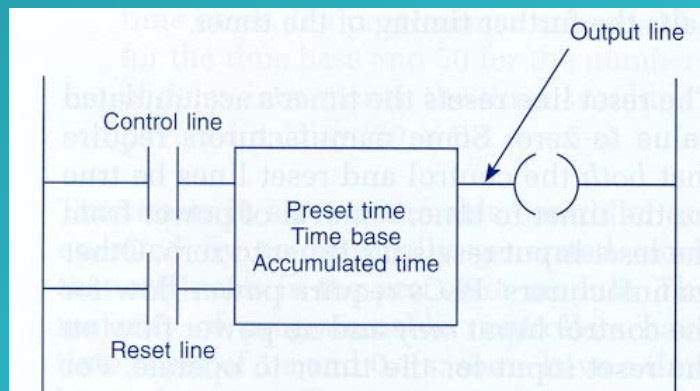
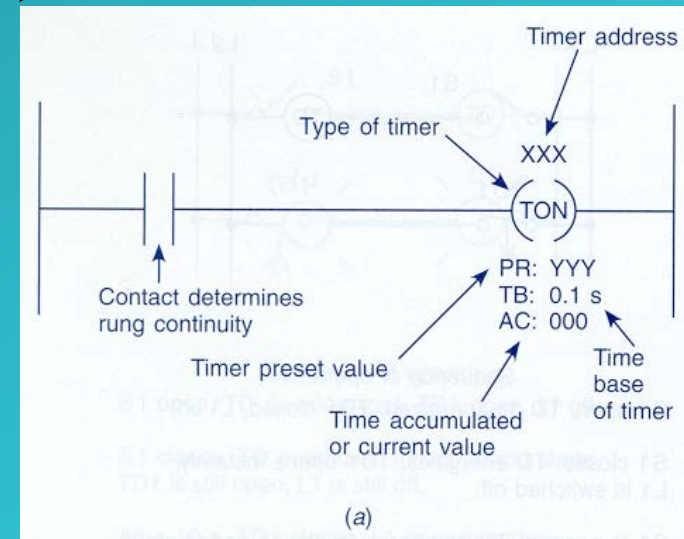
## Timers: exemplos



# Diagramas de contactos (cont.)

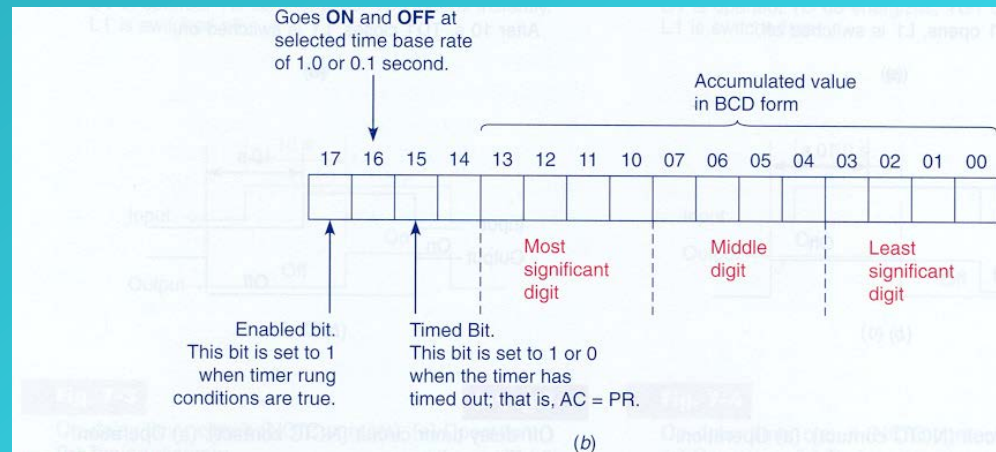
Implementação de *timers*  
no PLC-5 da *Allen-Bradley*:

dois métodos de representação



**Fig. 7-8**

Block-formatted timer instruction.

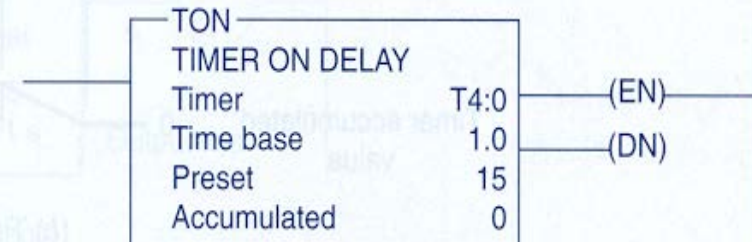
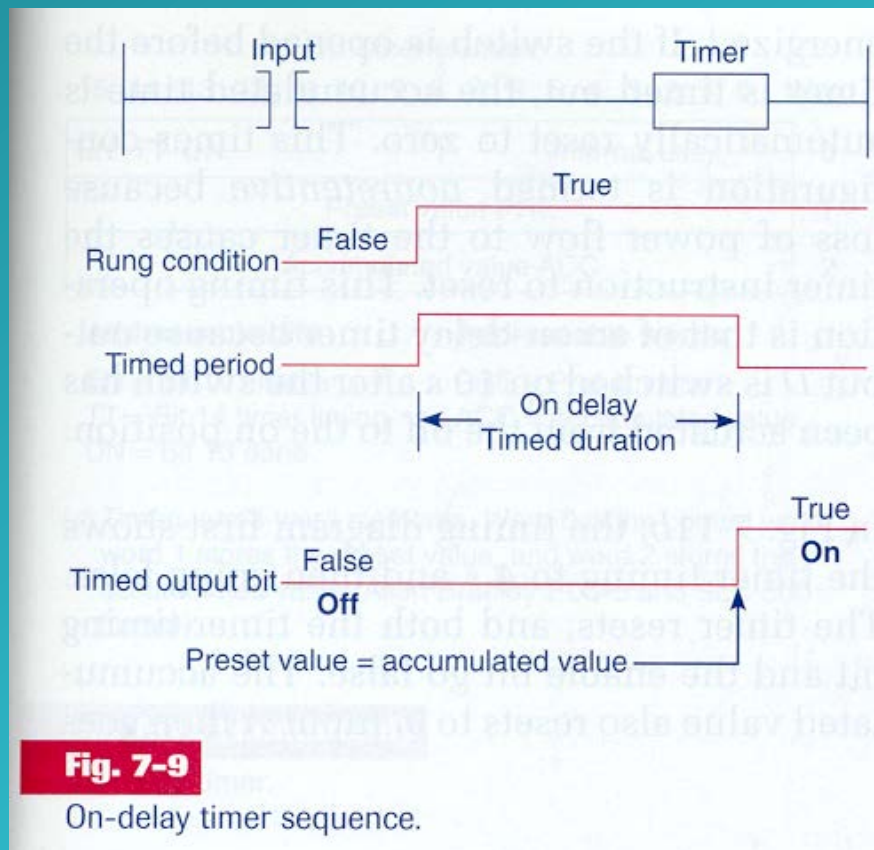


**Fig. 7-7**

Coil-formatted timer instruction. (a) Generic instruction. (b) Allen-Bradley PLC-2 timer accumulated value word.

## Diagramas de contactos (cont.)

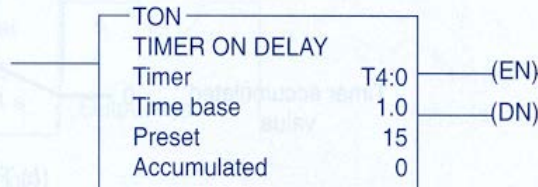
### Funcionamento de *timers* no PLC-5 da *Allen-Bradley*





# Diagramas de contactos (cont.)

## Exemplo de *timer on-delay*



**Fig. 7-10**

On-delay timer instruction.

Timer element																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
EN	TT	DN													Internal use	0
Preset value PRE															1	
Accumulated value ACC															2	

**Addressable bits**

EN = Bit 15 enable  
 TT = Bit 14 timer timing  
 DN = Bit 13 done

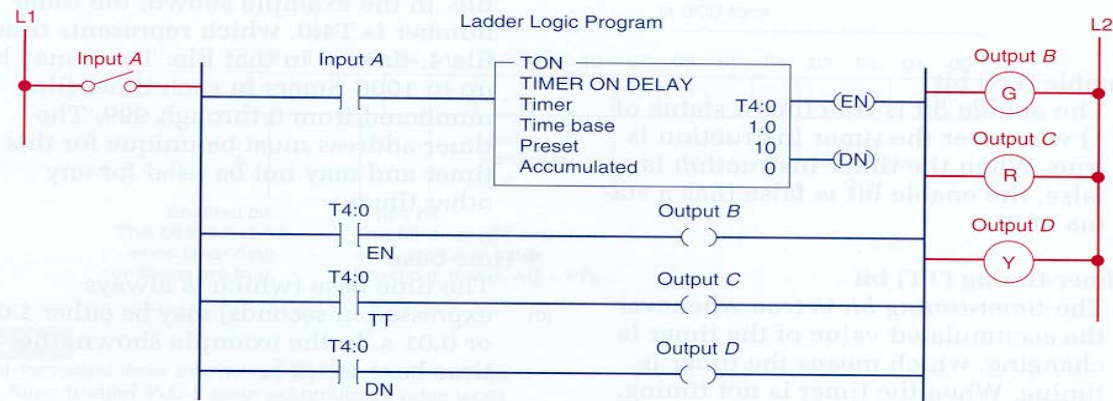
**Addressable words**

PRE = Preset value  
 ACC = Accumulated value

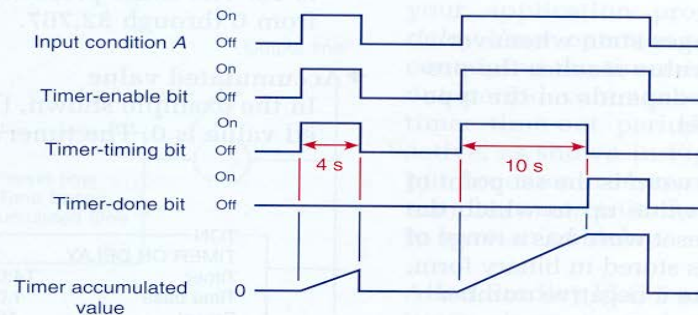
(c) Timers are 3-word elements. Word 0 is the control word, word 1 stores the preset value, and word 2 stores the accumulated value (Allen-Bradley PLC-5 and SLC-500 format).

**Fig. 7-11 (continued)**

On-delay timer.



(a) Ladder diagram



(b) Timing diagram

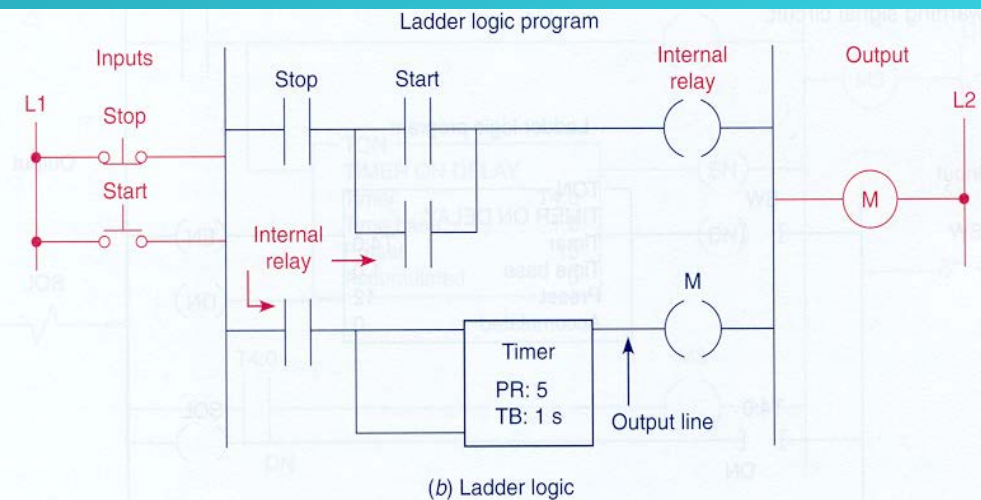
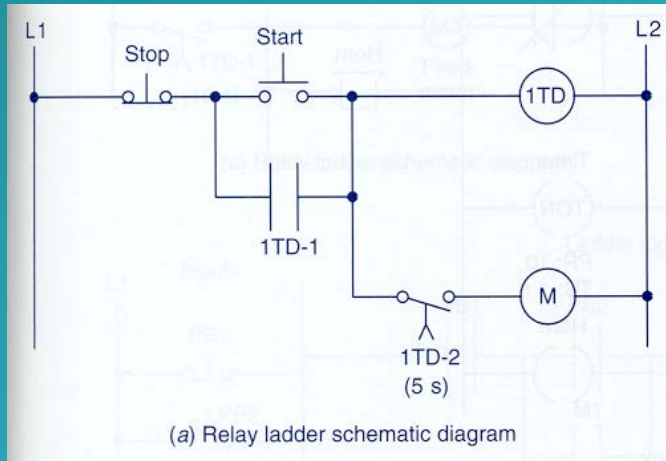
**Fig. 7-11**

On-delay timer.



## Diagramas de contactos (cont.)

### Exemplo de *timer on-delay com ligação à saída*

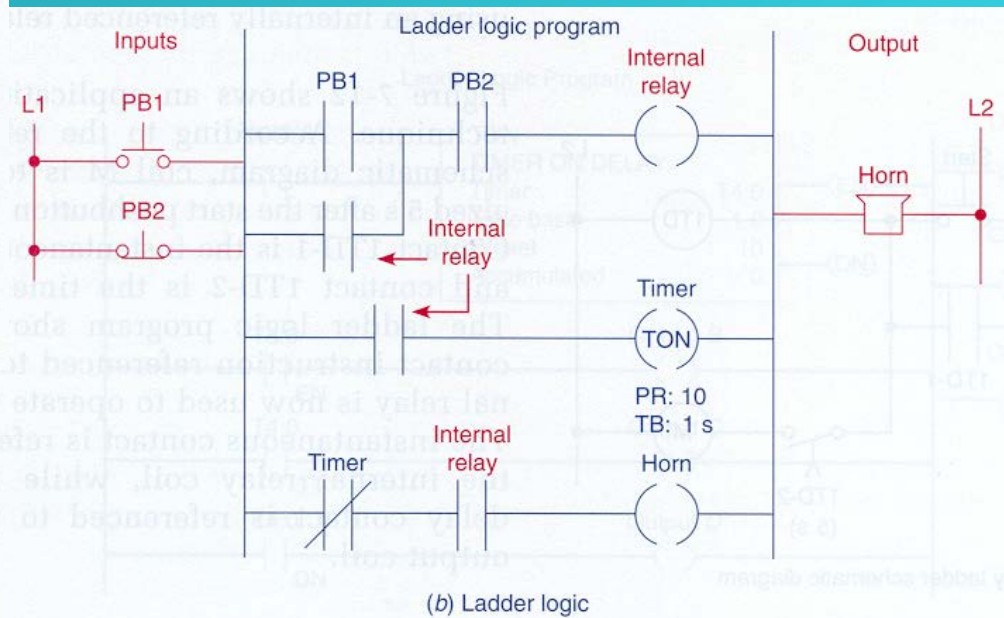
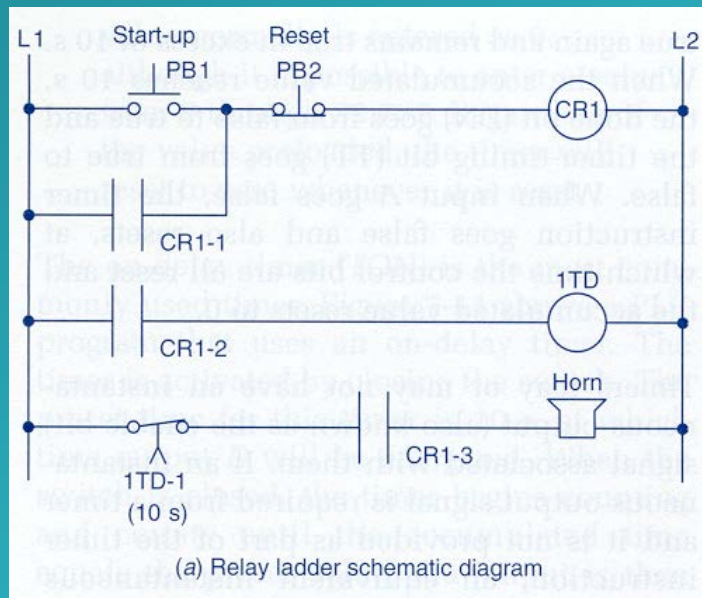


**Fig. 7-12**

On-delay timer with instantaneous output programming.

# Diagramas de contactos (cont.)

## Exemplo de *timer on-delay*



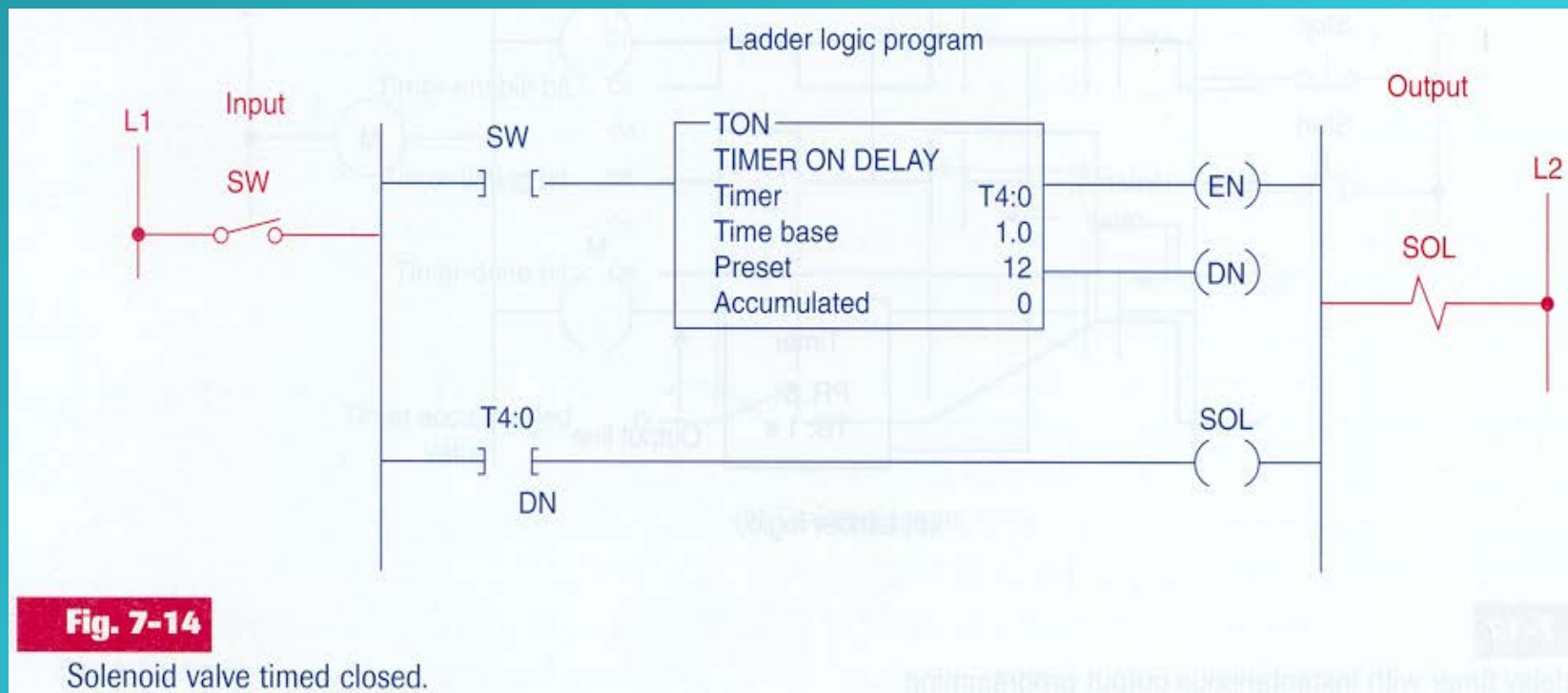
**Fig. 7-13**

Starting-up warning signal circuit.

## Diagramas de contactos (cont.)

### Exemplo de *timer on-delay*

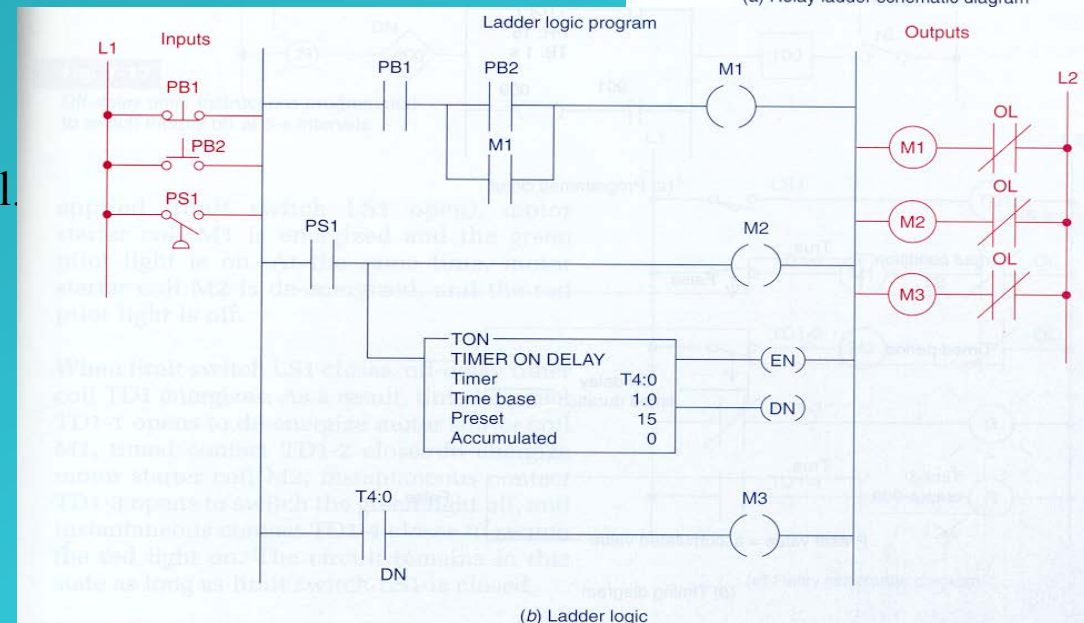
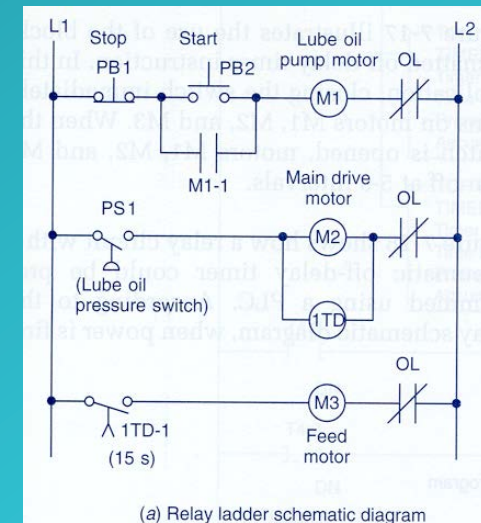
Solenóide é energizado se o interruptor estiver fechado 12 segundos



# Diagramas de contactos (cont.)

## Exemplo de *timer on-delay*

- Se PB2 for actuado liga o motor de bombagem de óleo.
- Quando a pressão subir PS1 vai detectar a subida da pressão e ligar motor principal.
- 15 segundos depois arranca o motor de alimentação principal.



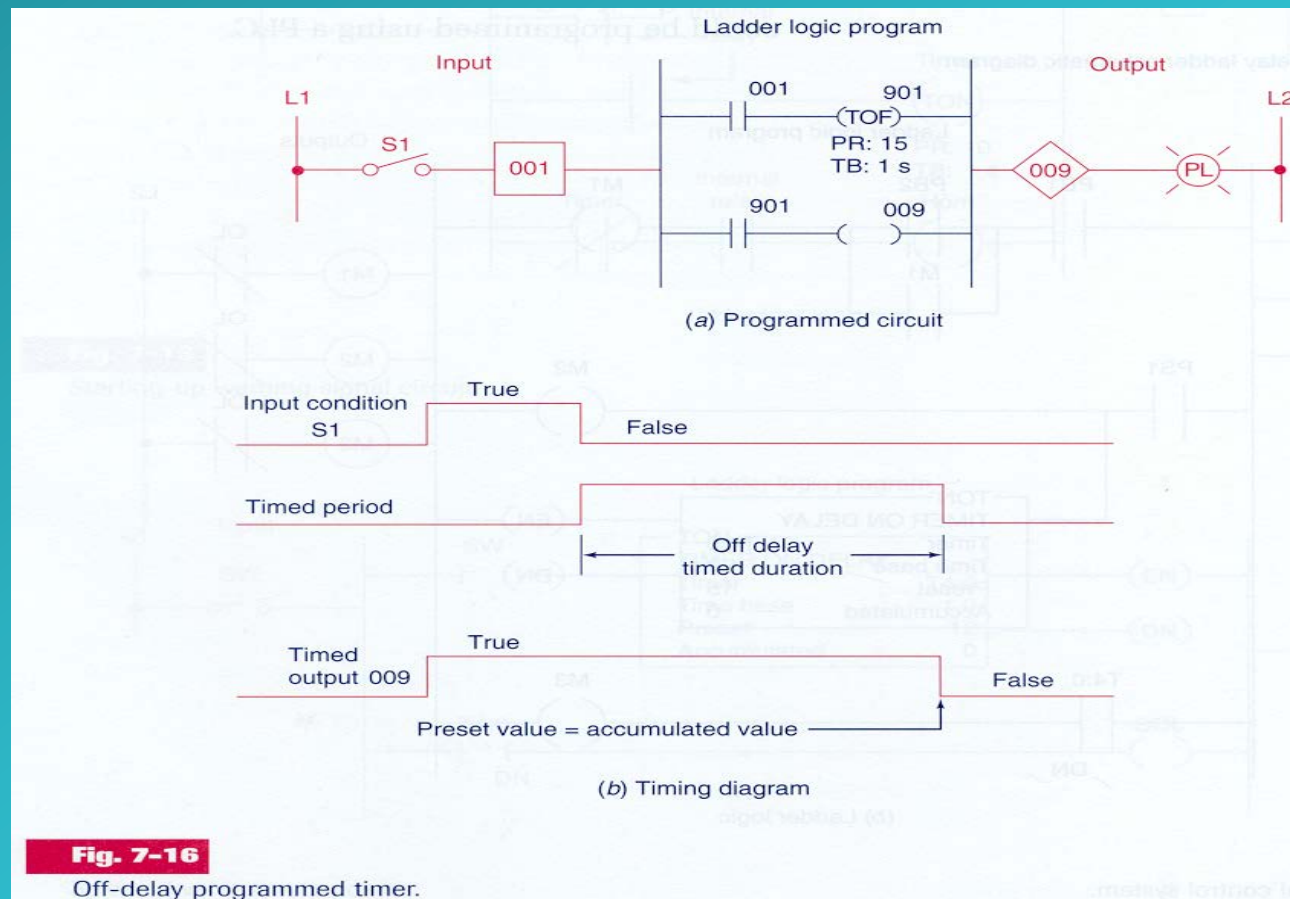
**Fig. 7-15**

Automatic sequential control system.



# Diagramas de contactos (cont.)

## Exemplo de *timer* programado *off-delay*

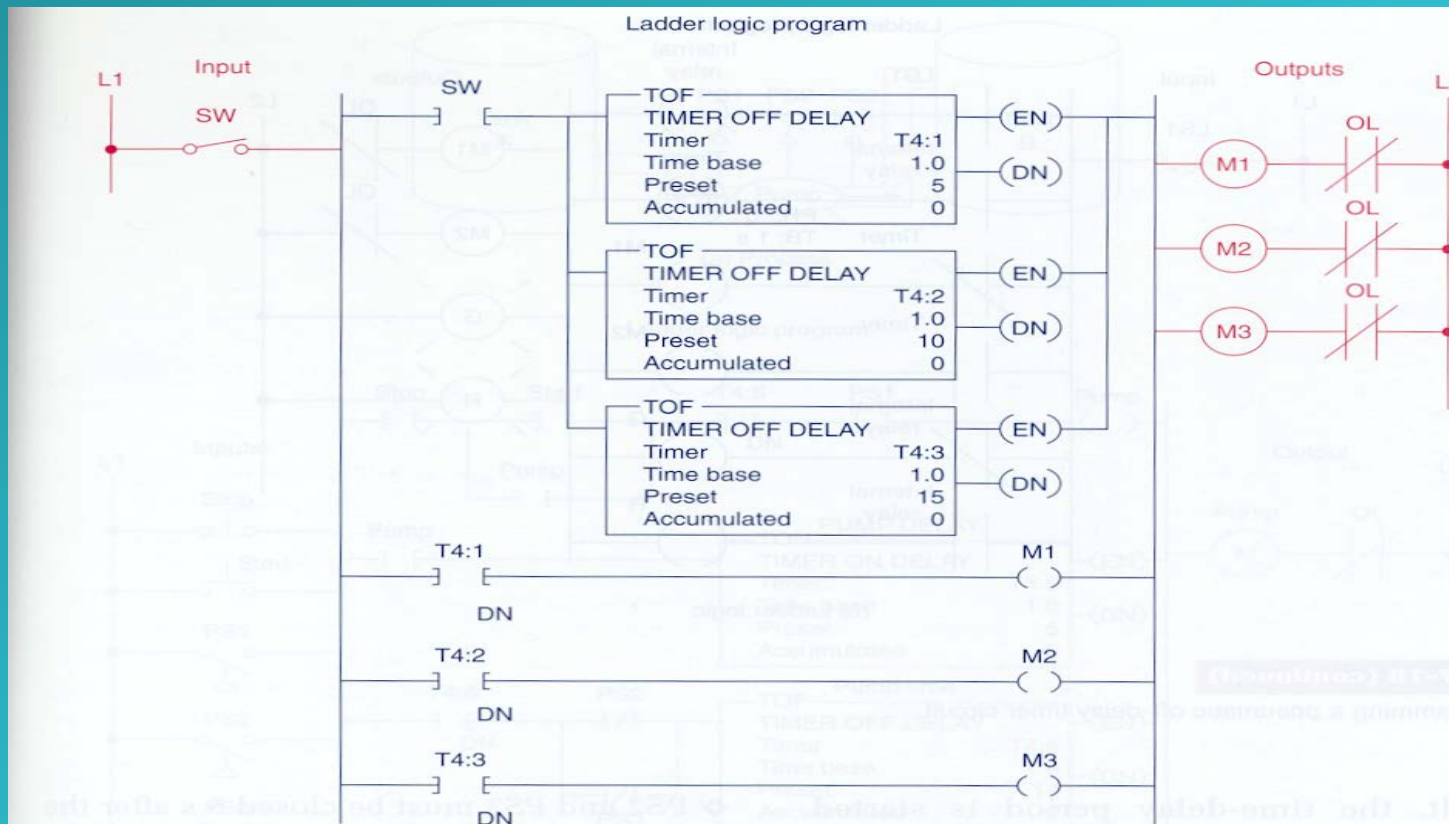


**Fig. 7-16**

Off-delay programmed timer.

# Diagramas de contactos (cont.)

## Exemplo de *timer* programado *off-delay* (2)

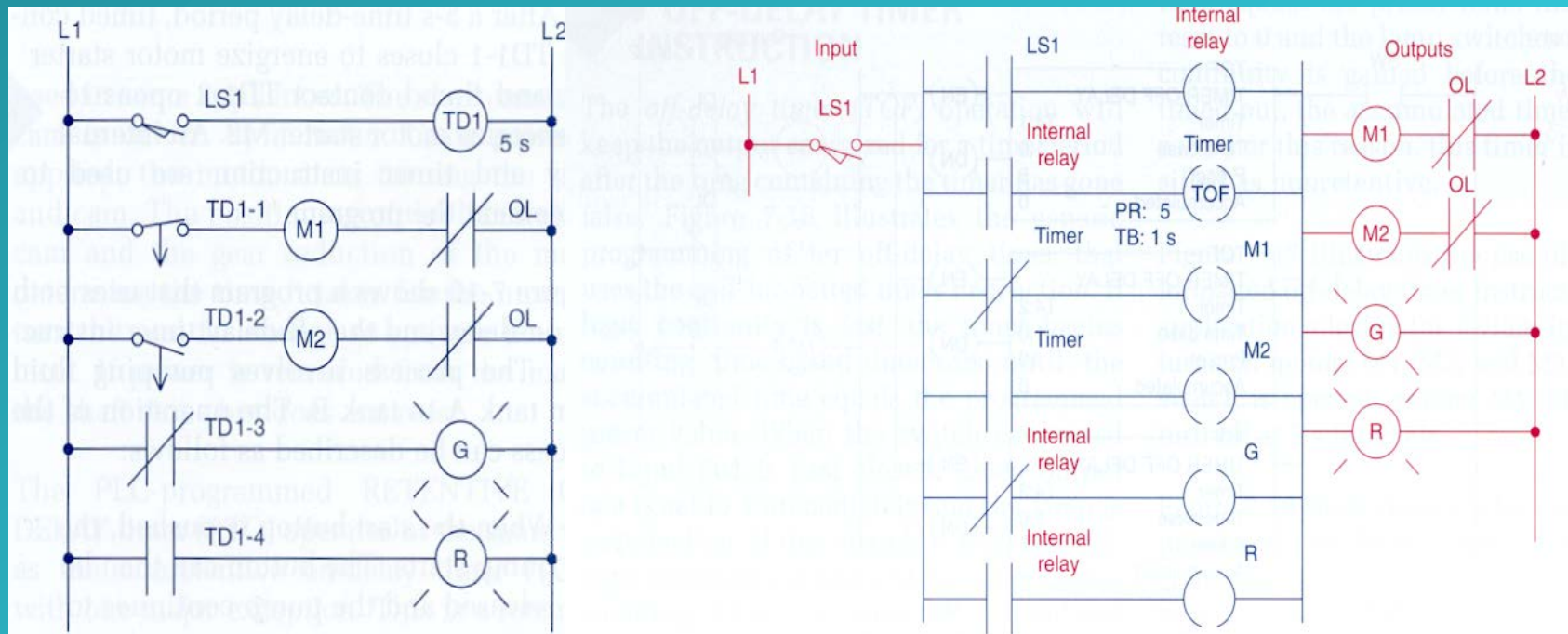


**Fig. 7-17**

Off-delay timer instructions programmed to switch motors off at 5-s intervals.

# Diagramas de contactos (cont.)

## Exemplo de *timer* programado *off-delay* (3)



(a) Relay schematic diagram

(b) Ladder logic

**Fig. 7-18**

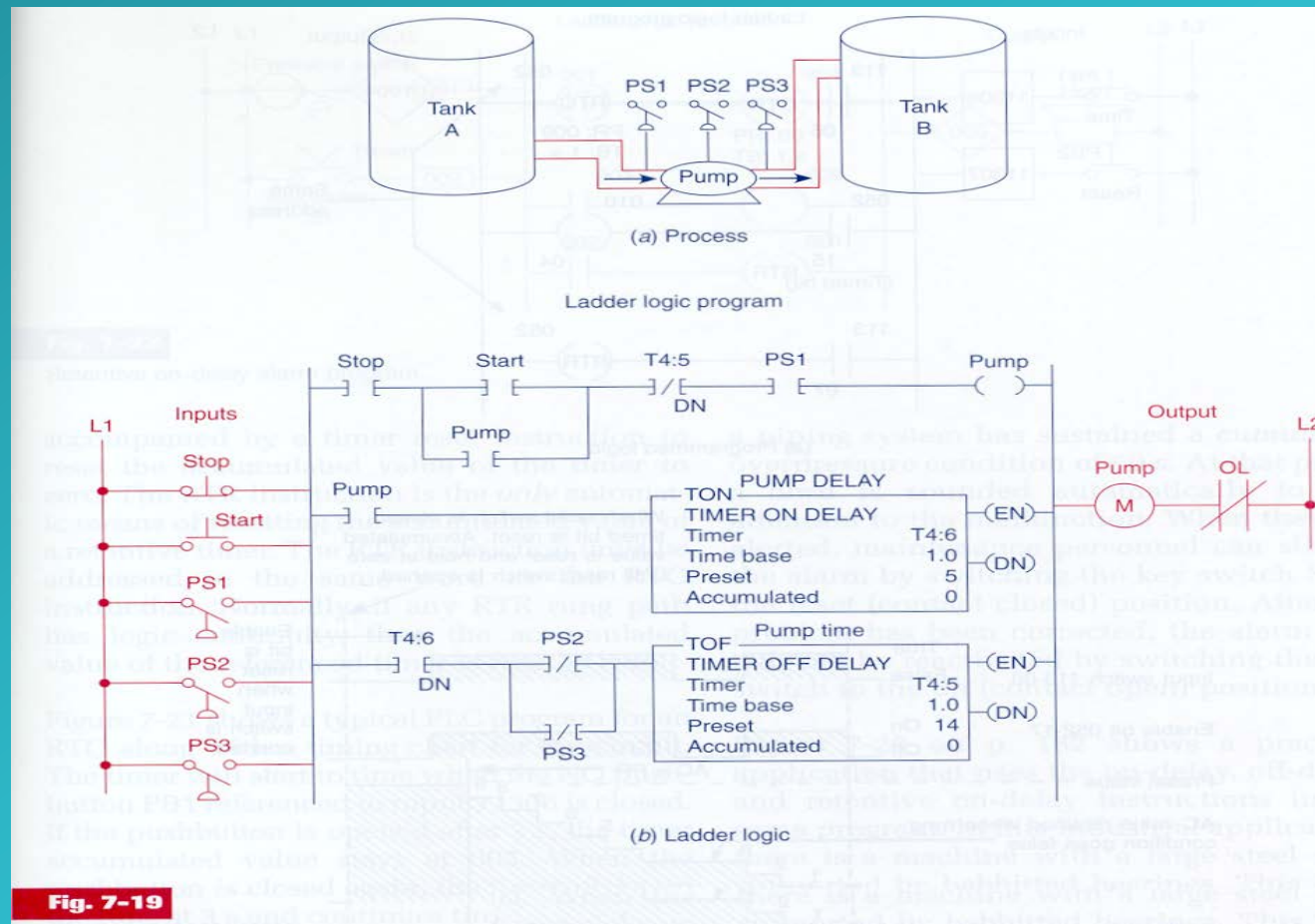
Programming a pneumatic off-delay timer circuit.

**Fig. 7-18 (continued)**

Programming a pneumatic off-delay timer circuit.

# Diagramas de contactos (cont.)

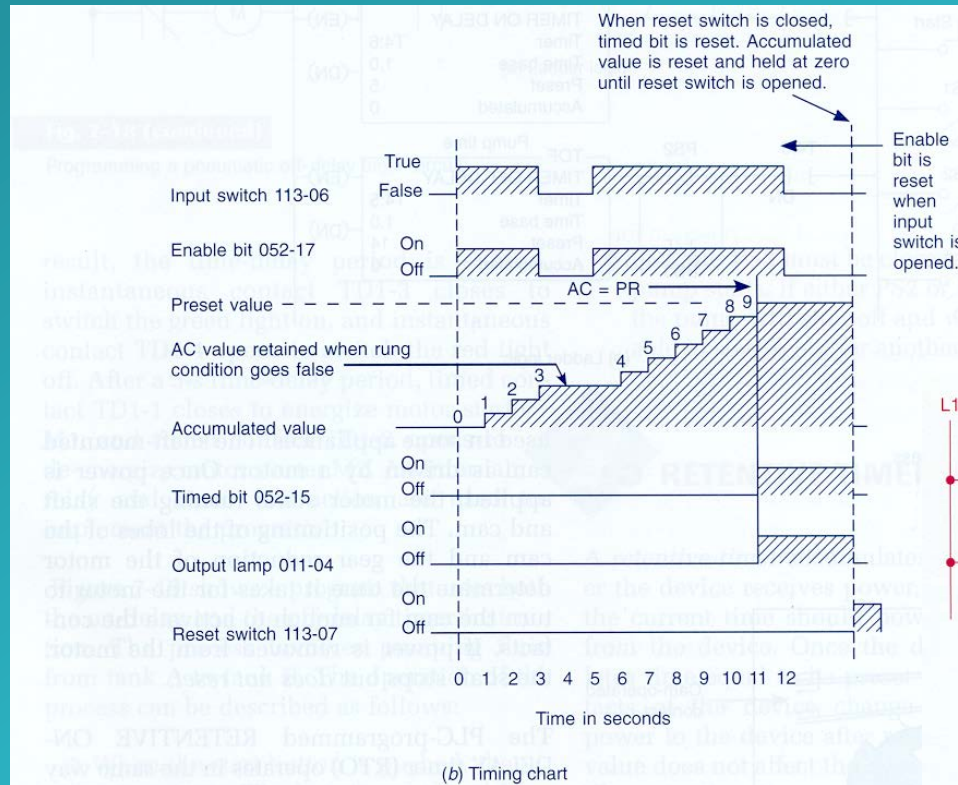
## Exemplo de *timers* programados *on-delay* e *off-delay*





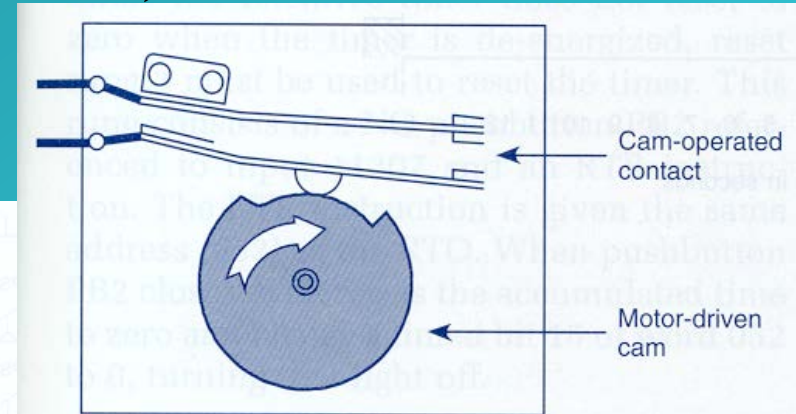
# Diagramas de contactos (cont.)

## Timers com retenção



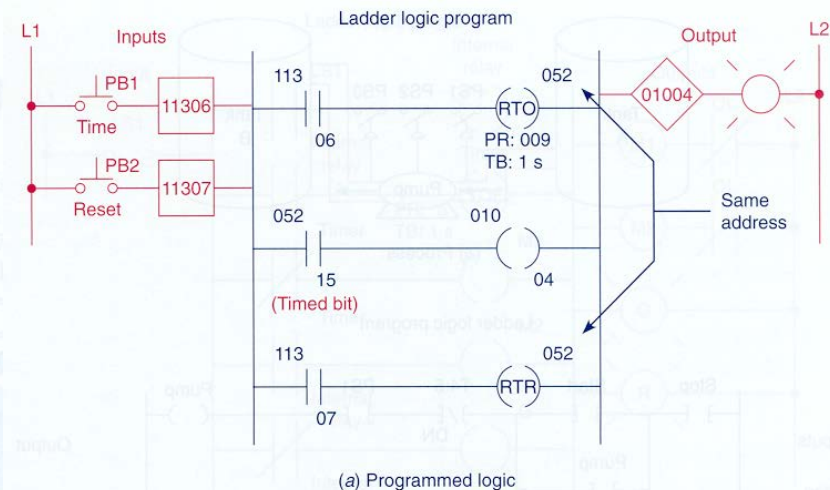
**Fig. 7-21**

Retentive on-delay timer program and timing chart.



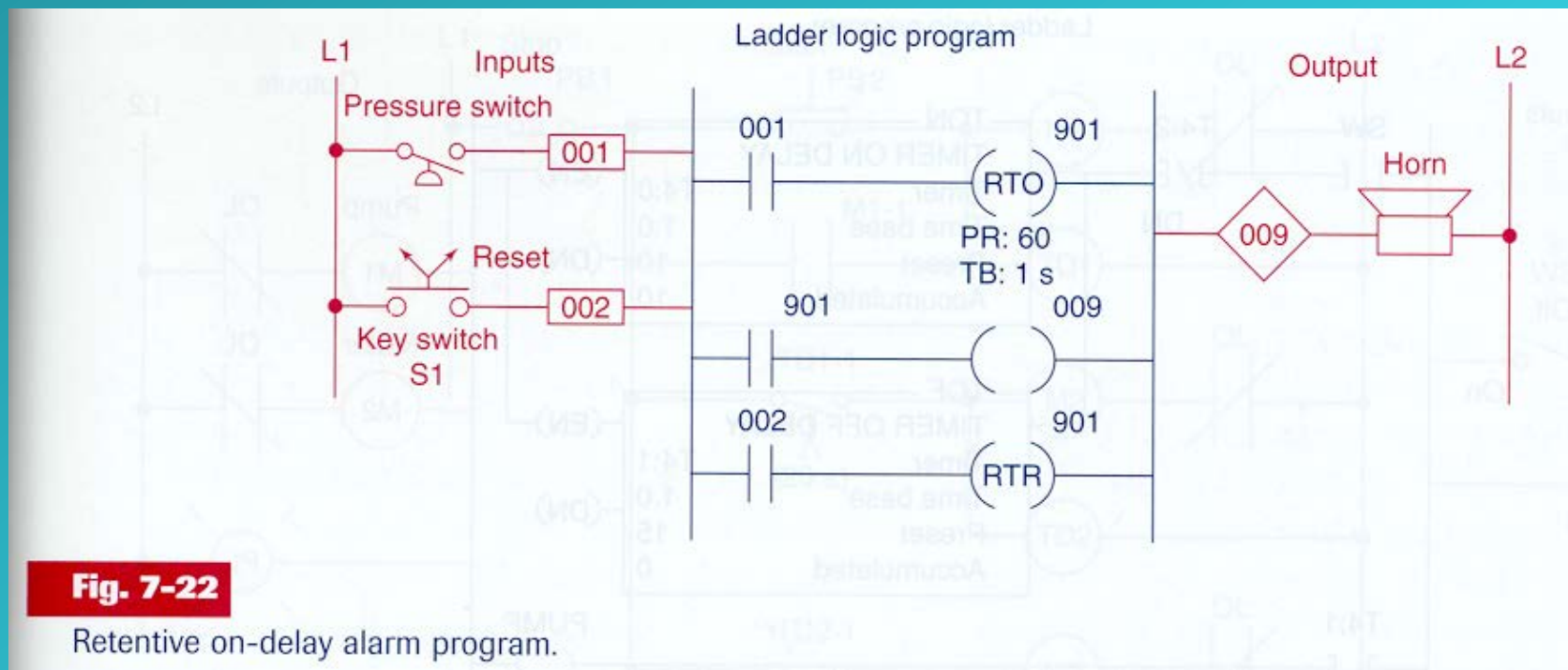
**Fig. 7-20**

Electromechanical retentive timer.



## Diagramas de contactos (cont.)

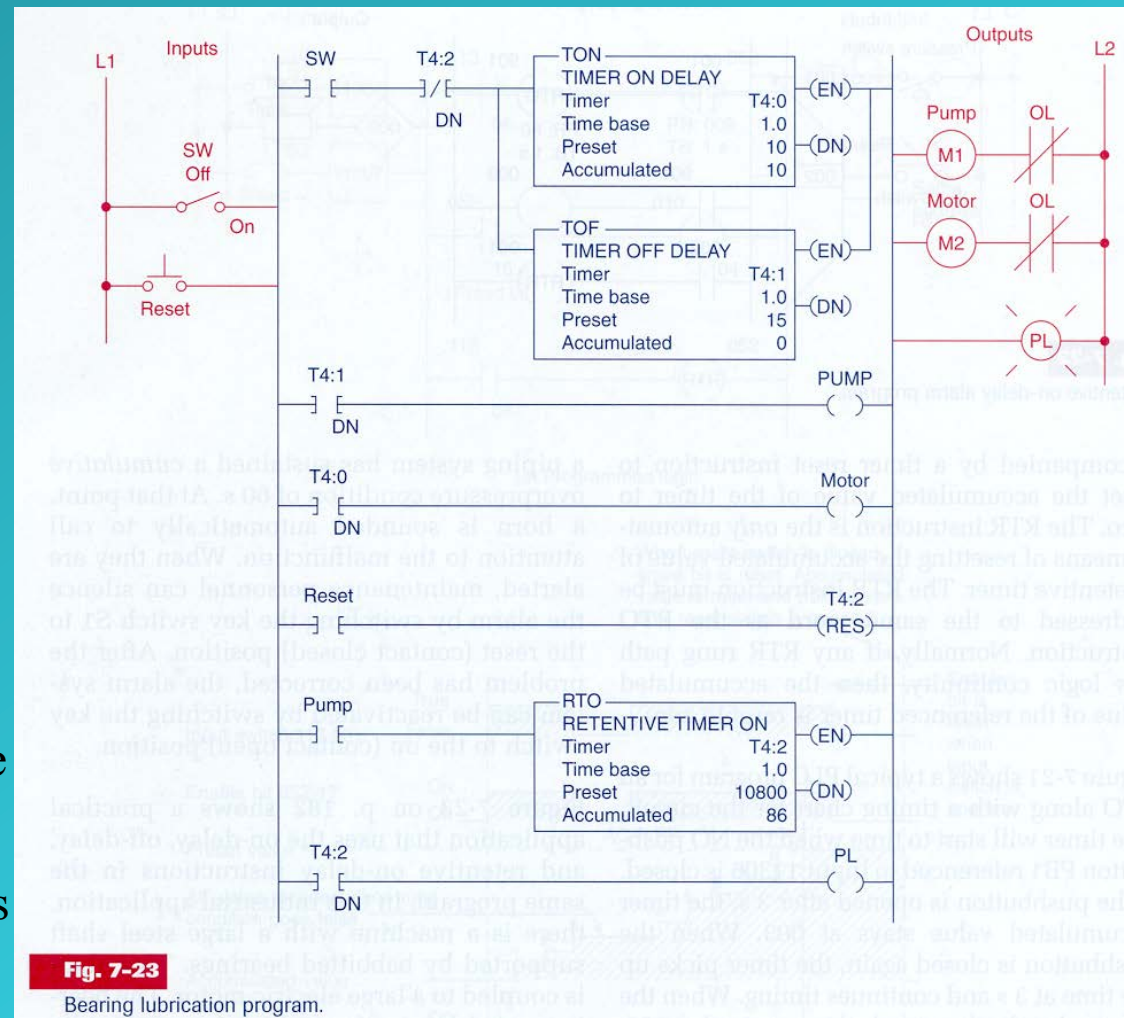
### Exemplo com *timers* com retenção



# Diagramas de contactos (cont.)

## Exemplo

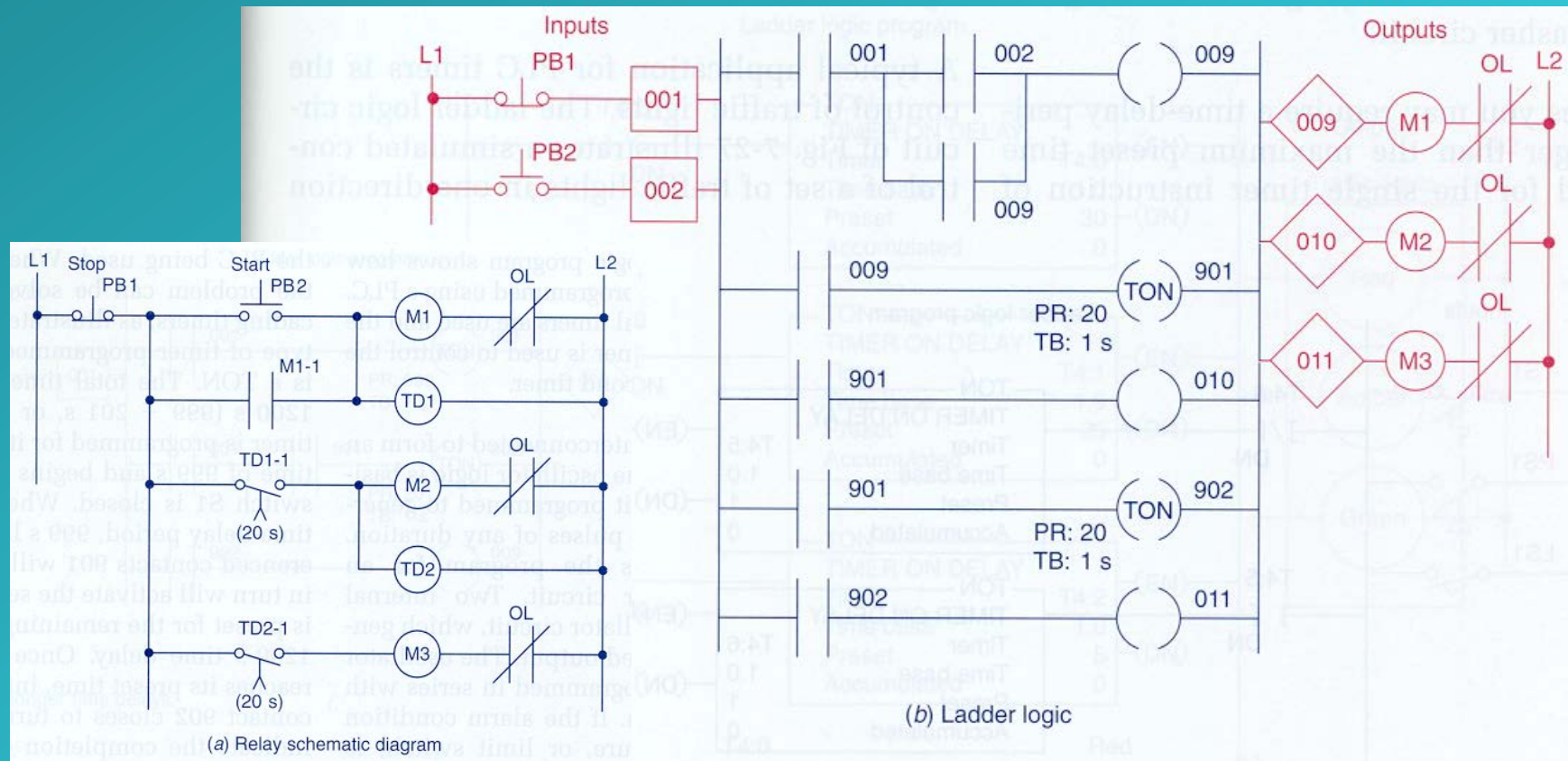
- SW On para iniciar
- Antes do motor arrancar, lubrificar 10s com óleo
- SW off para parar.
- Continuar a lubrificar mais 15 segundos.
- Quando a bomba estiver a funcionar 3 horas, desligar o motor e acender luz piloto para avisar da necessidade de trocar o filtro.
- Reset disponível para depois de trocar o filtro.





# Diagramas de contactos (cont.)

## Timers em cascata

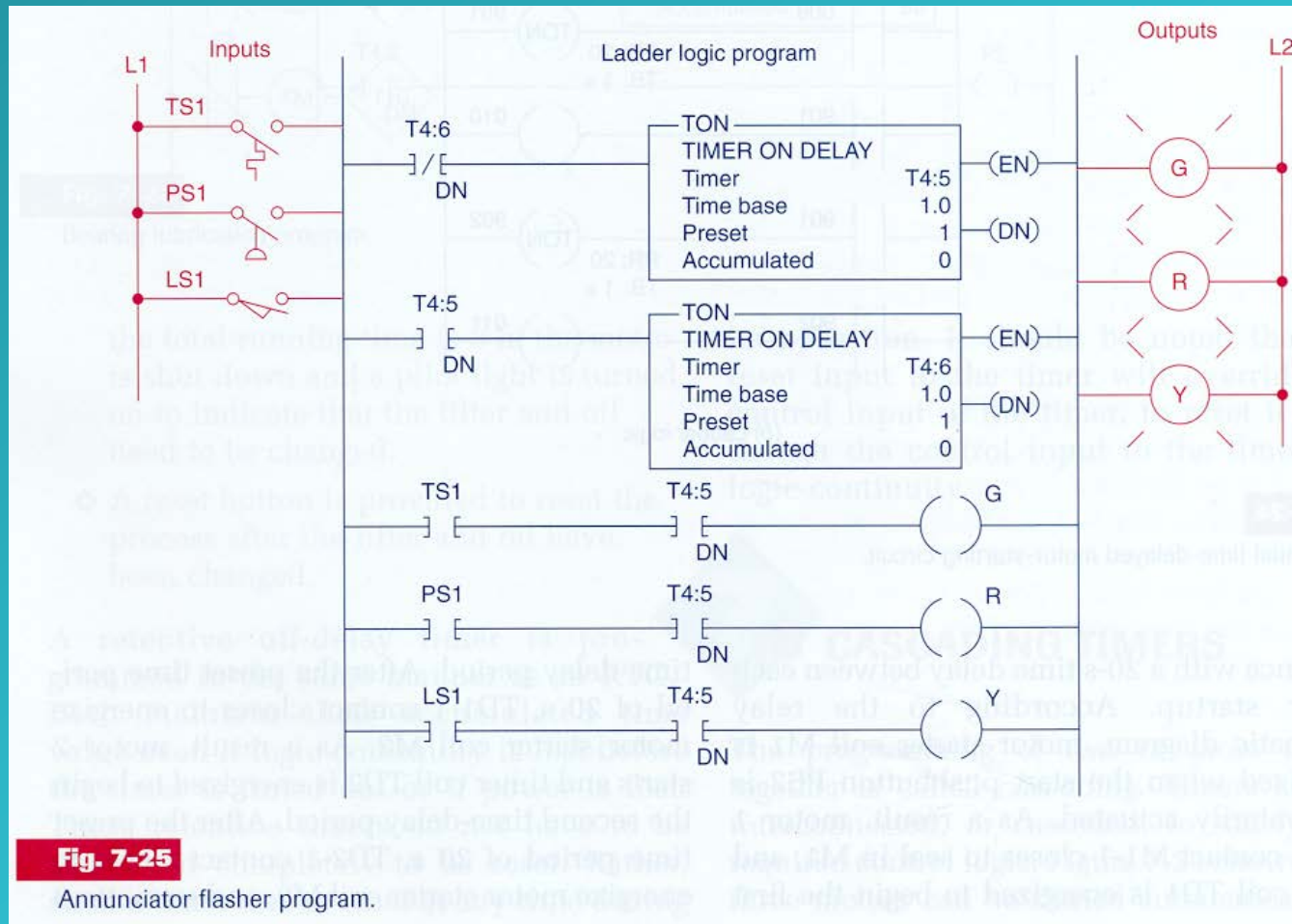


**Fig. 7-24**  
Sequential time-delayed motor-starting circuit.



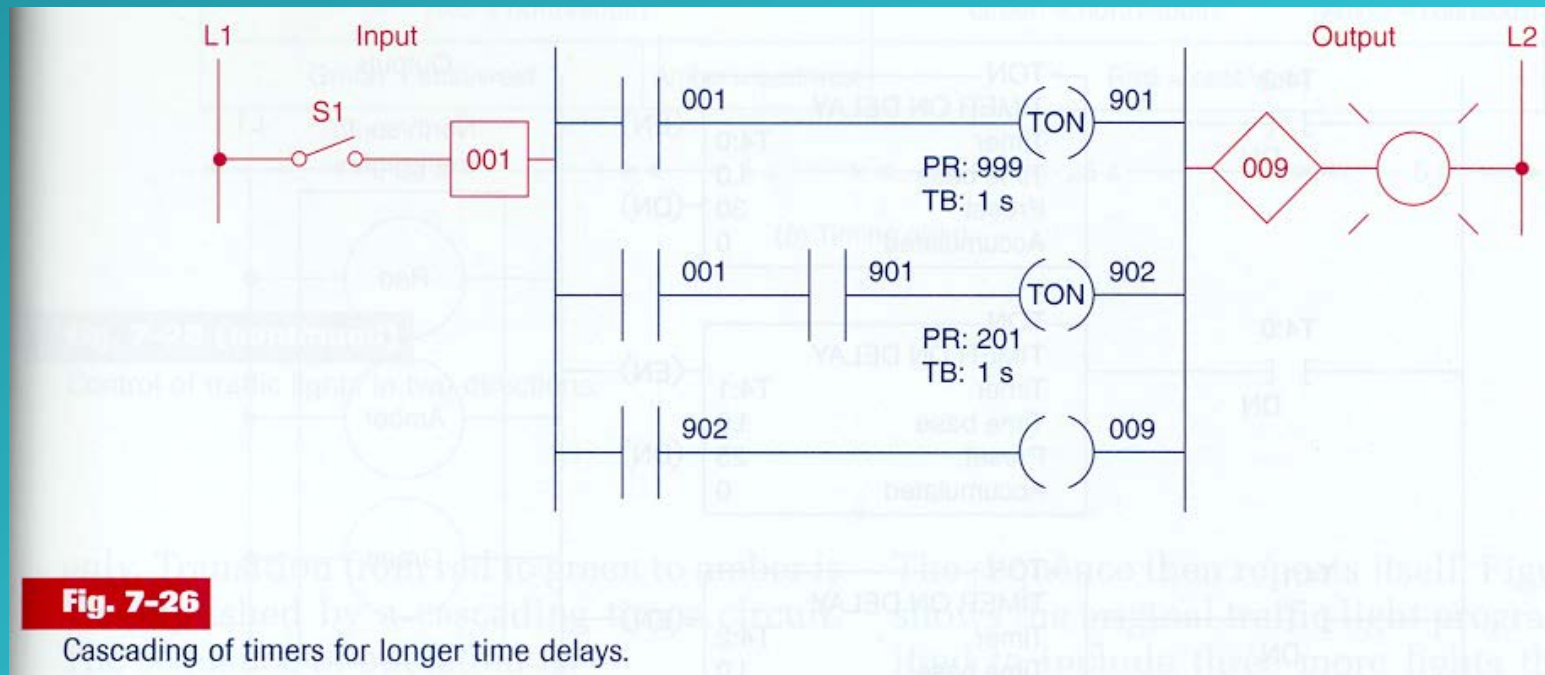
# Diagramas de contactos (cont.)

## Timers em cascata (circuito oscilatório)



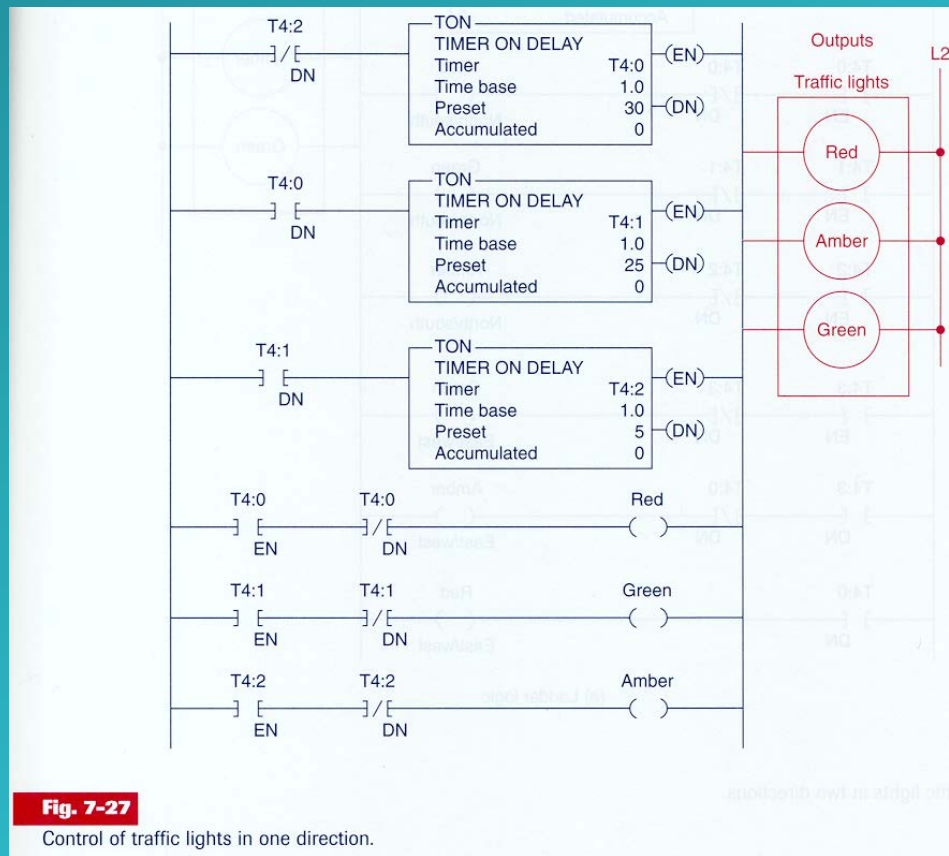
## Diagramas de contactos (cont.)

### Timers para tempos muito longos



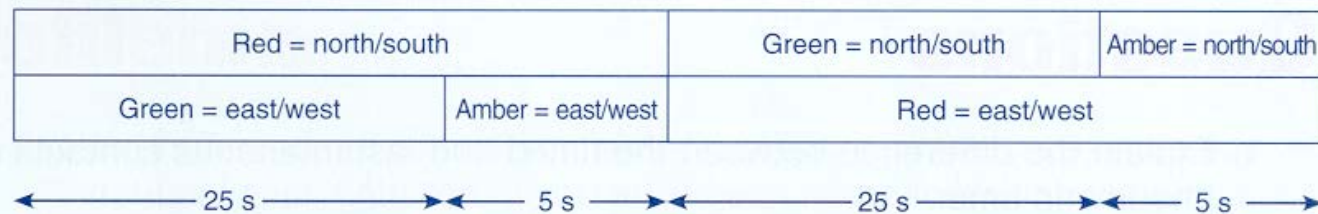
# Diagramas de contactos (cont.)

## Exemplo de um semáforo



### Exemplo de semáforos em duas direcções

Red	30 s on
Green	25 s on
Amber	5 s on



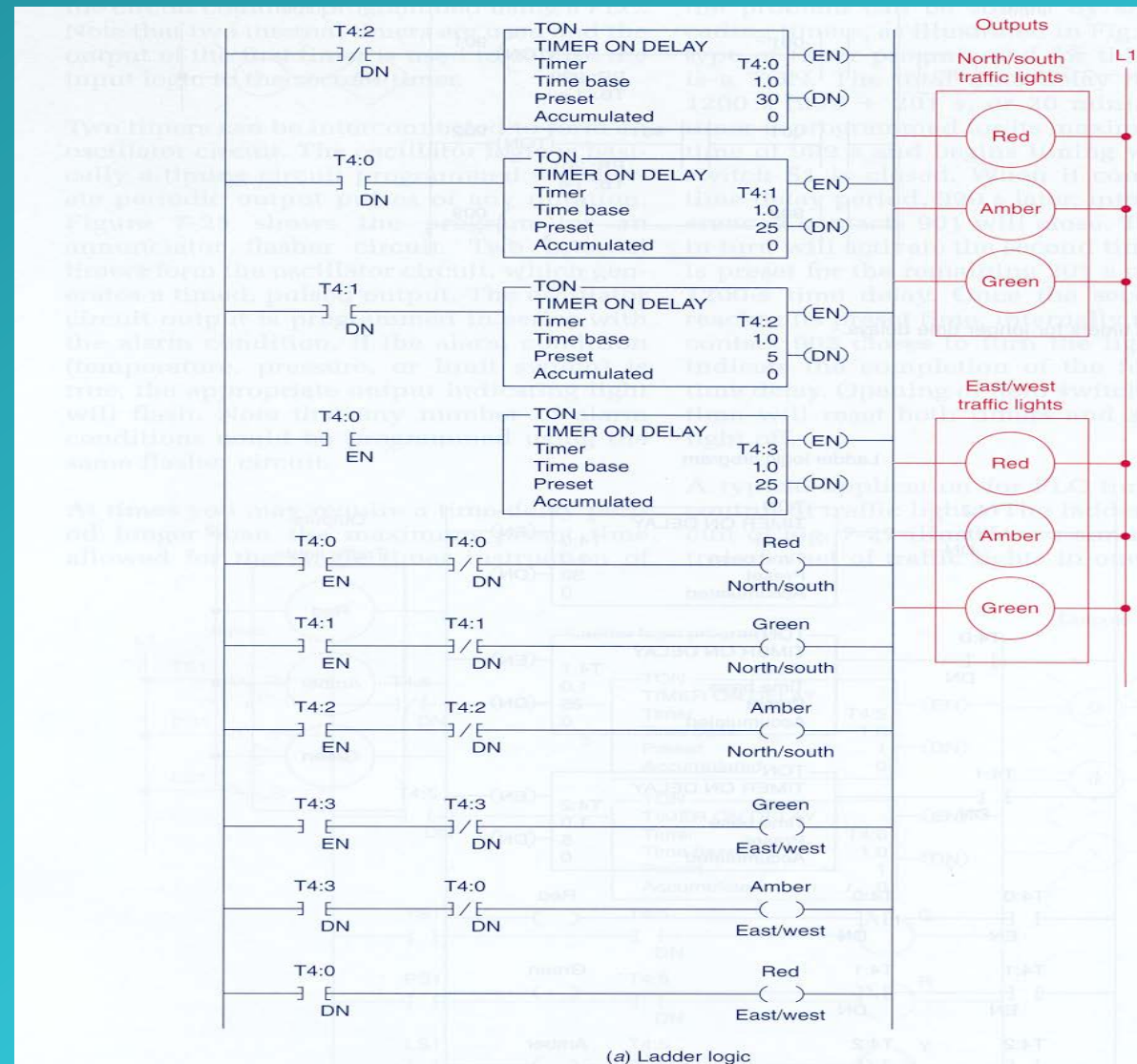
(b) Timing chart

**Fig. 7-28 (continued)**

Control of traffic lights in two directions.



## Exemplo de semáforos em duas direcções



**Fig. 7-28**

Control of traffic lights in two directions.

# Automação Industrial baseada em PLCs

## 6ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 6ª Aula

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.

### Cap. 3 - Linguagens de Programação de PLCs

- Diagramas de contactos (ladder diagram).
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

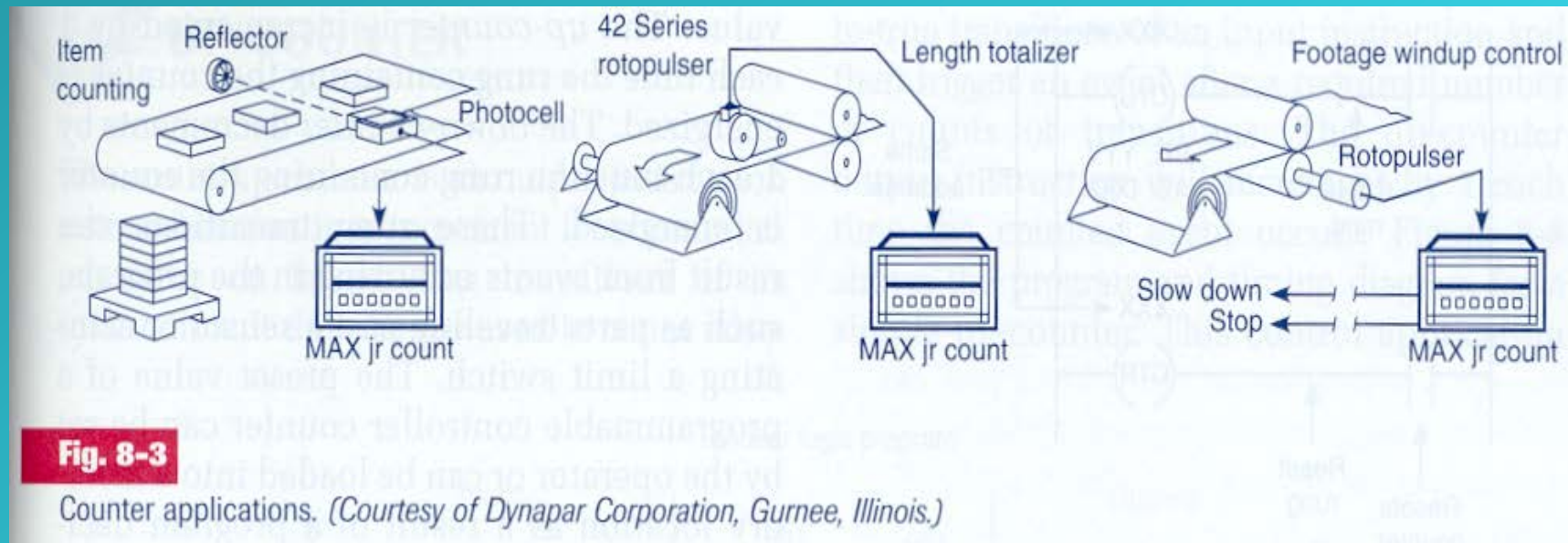
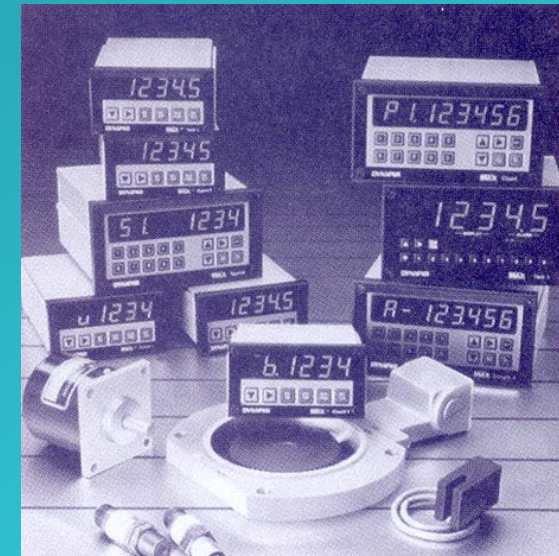
### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.

# Diagramas de contactos (cont.)

## Contadores

## Aplicações típicas

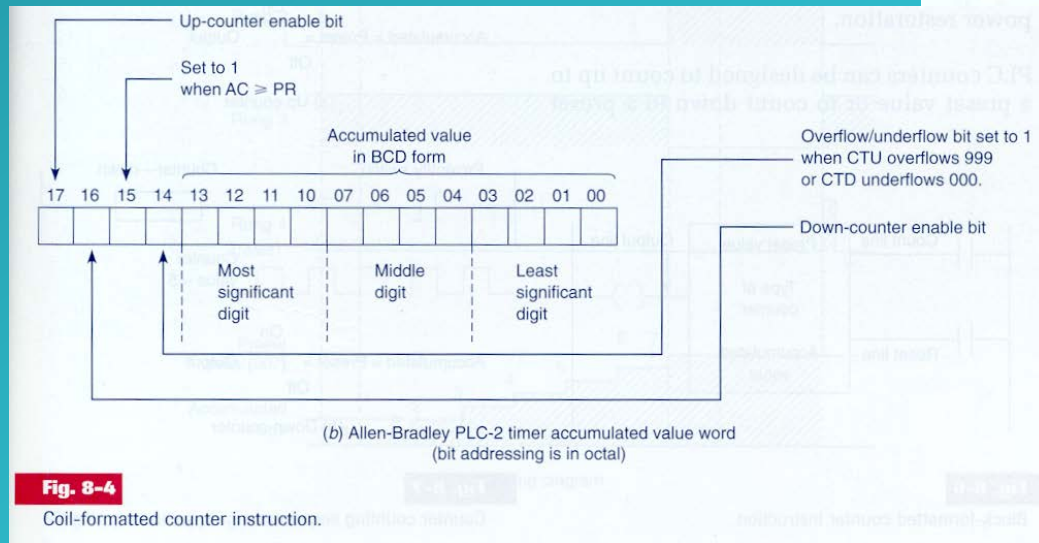
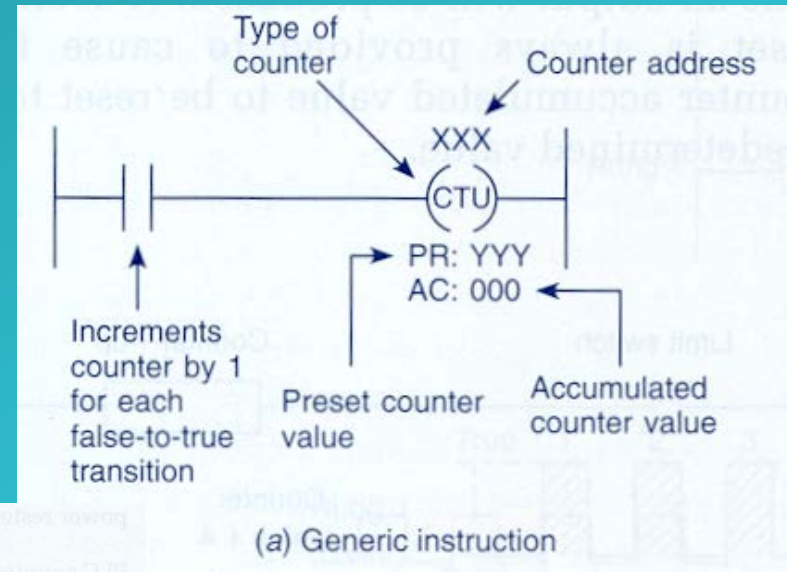




# Diagramas de contactos (cont.)

## Contadores

### Estrutura interna

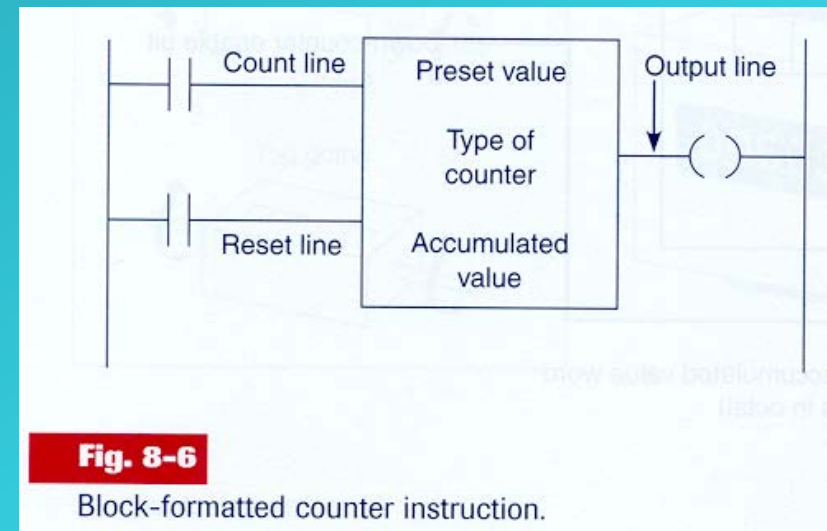
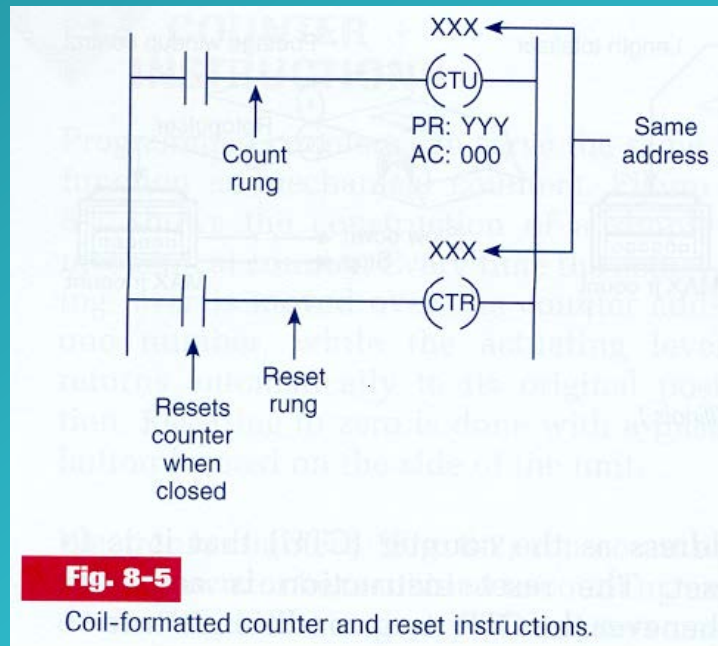


### Representação

# Diagramas de contactos (cont.)

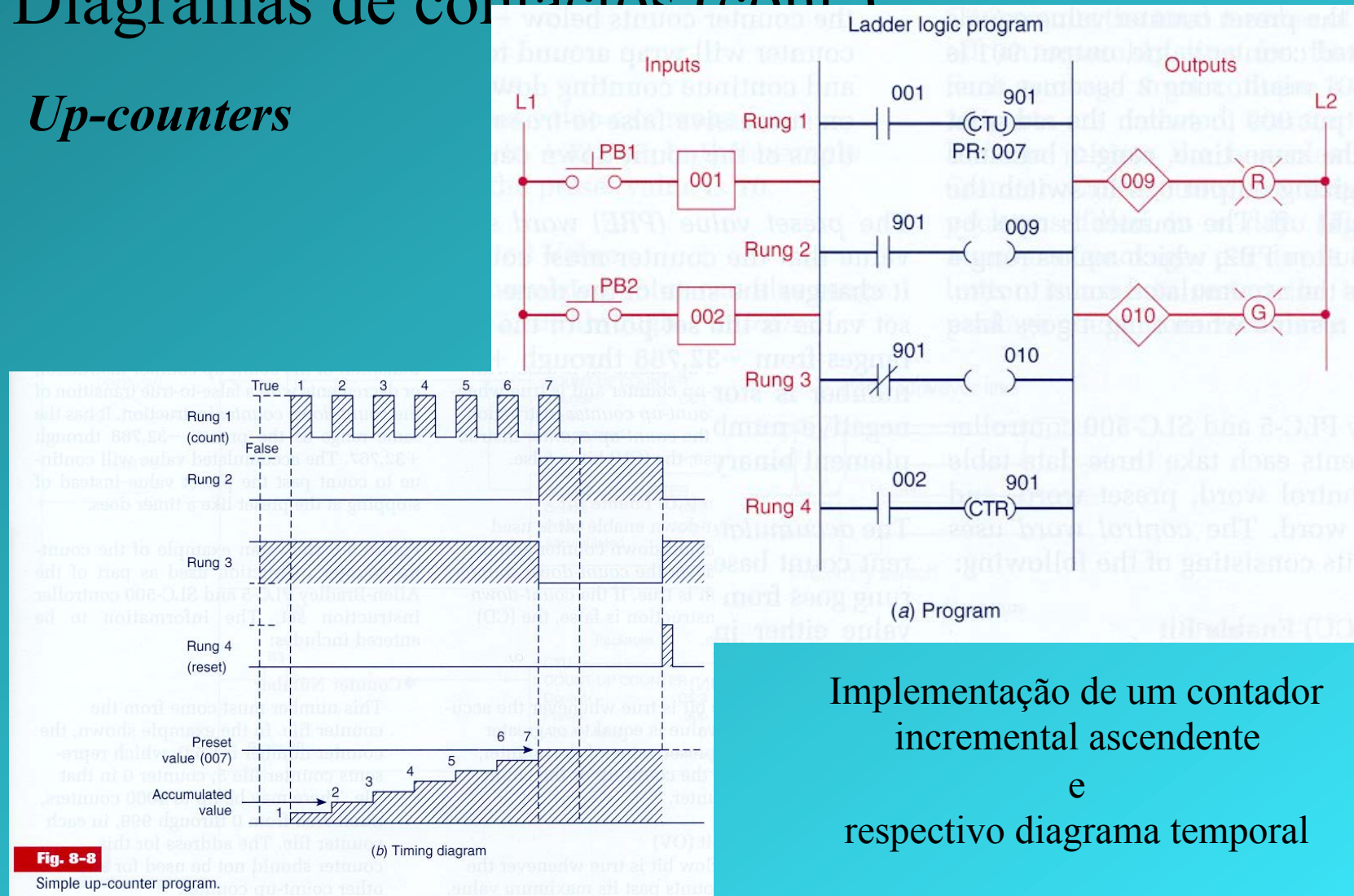
## Contadores

### Representações alternativas



# Diagramas de contactos (cont.)

## Up-counters



Implementação de um contador incremental ascendente e respectivo diagrama temporal

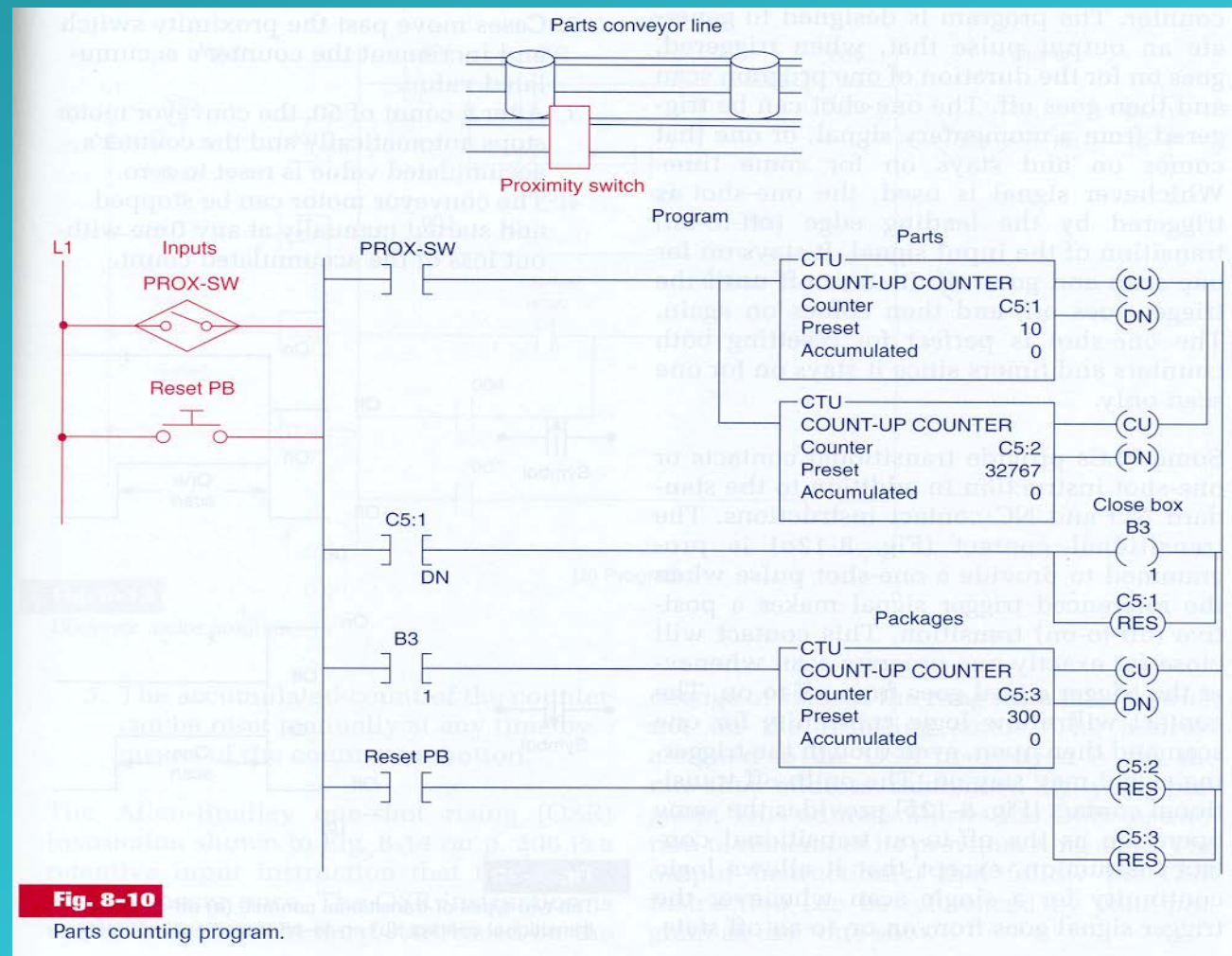


## Diagramas de contactos (cont.)

### Up-counters

Exemplo:

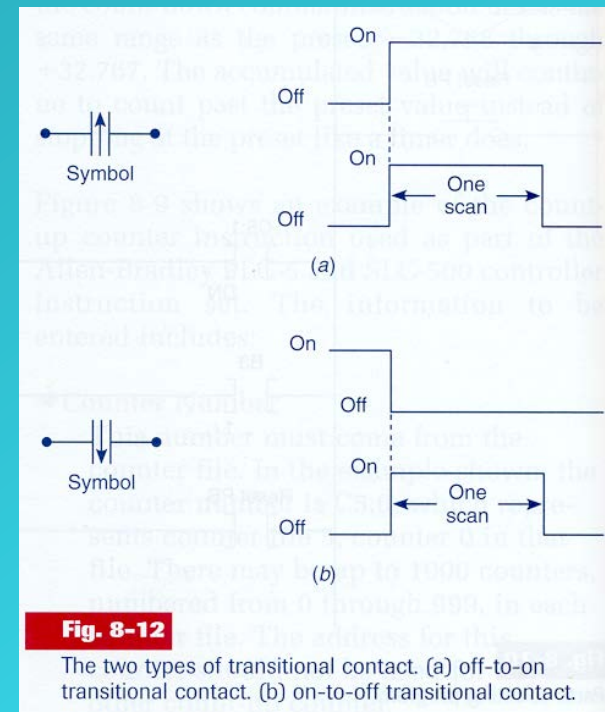
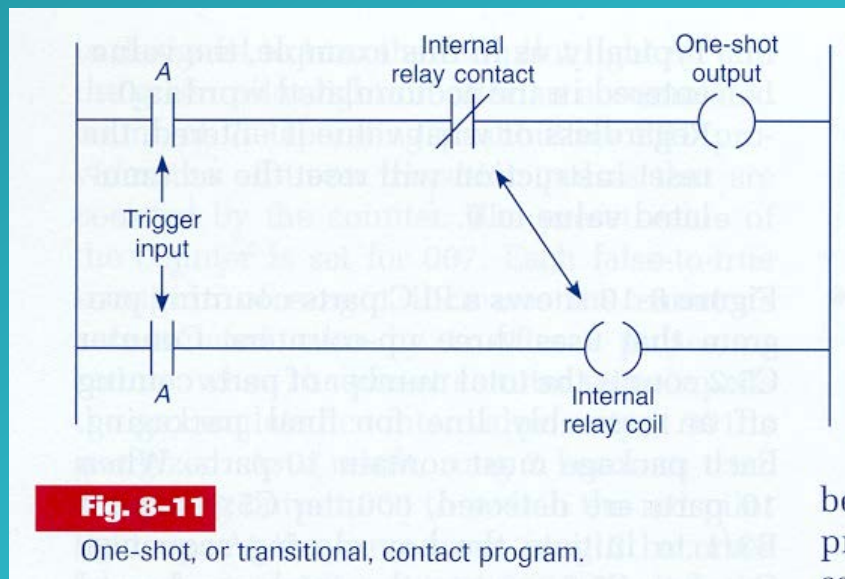
contagem de peças





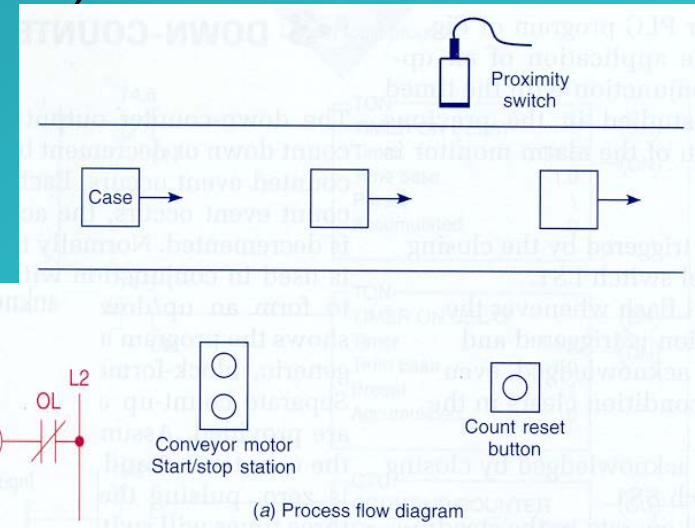
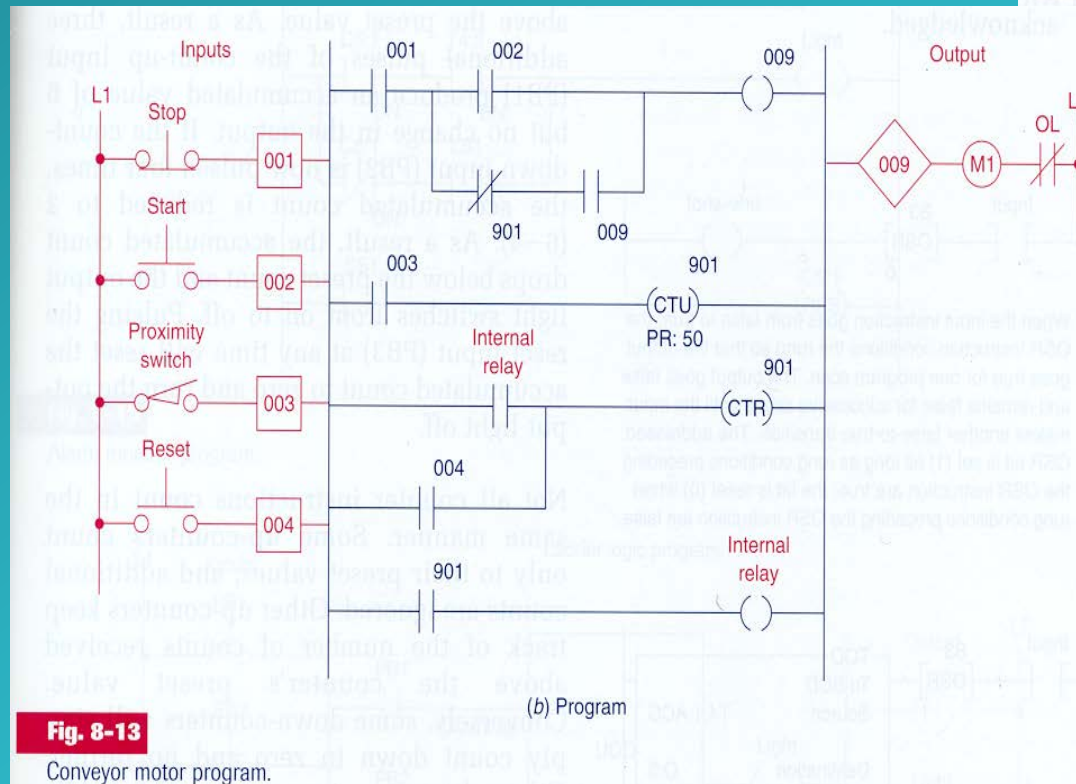
# Diagramas de contactos (cont.)

*One-shot* para fazer reset aos contadores



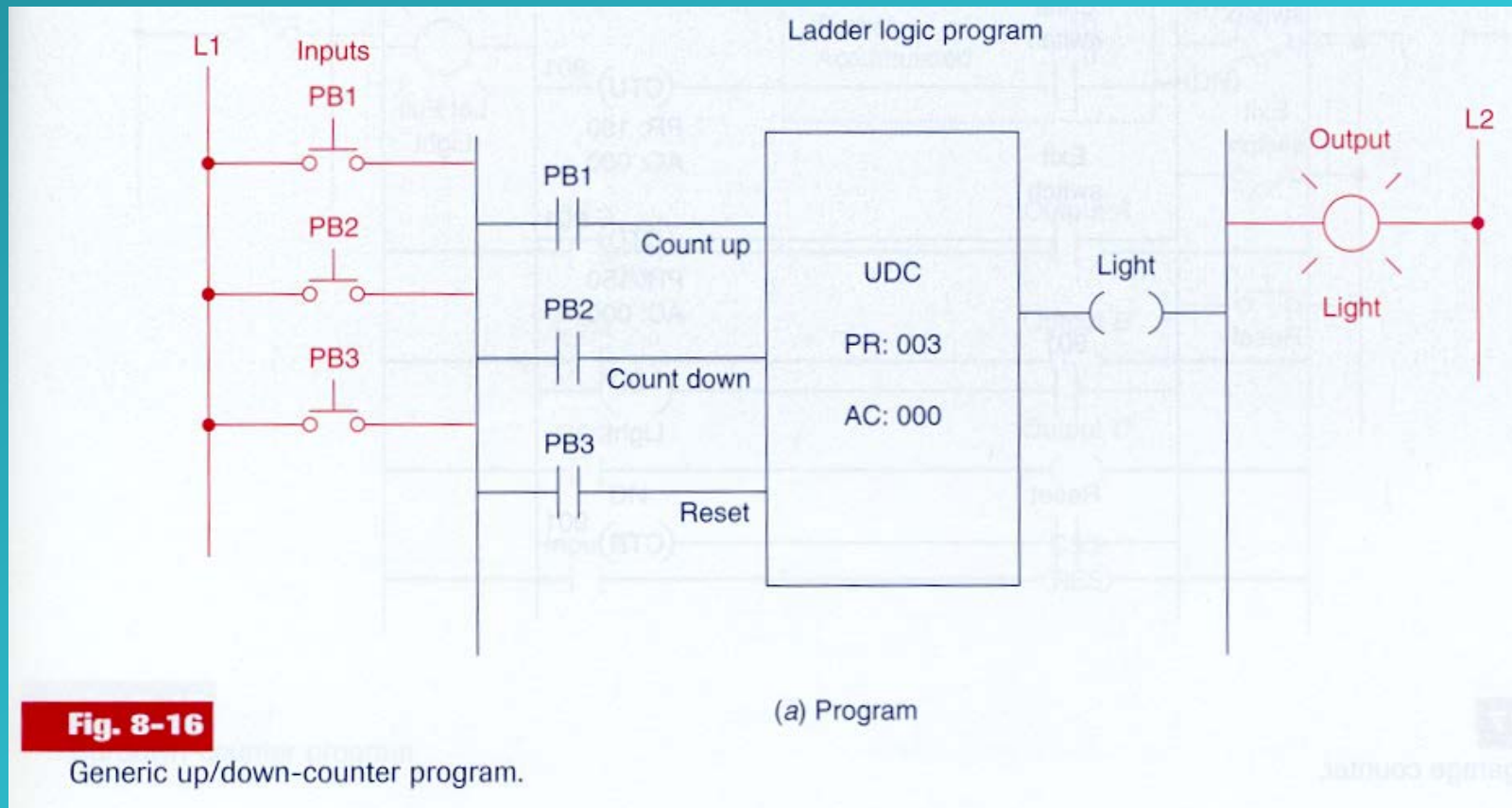
# Diagramas de contactos (cont.)

## Exemplo



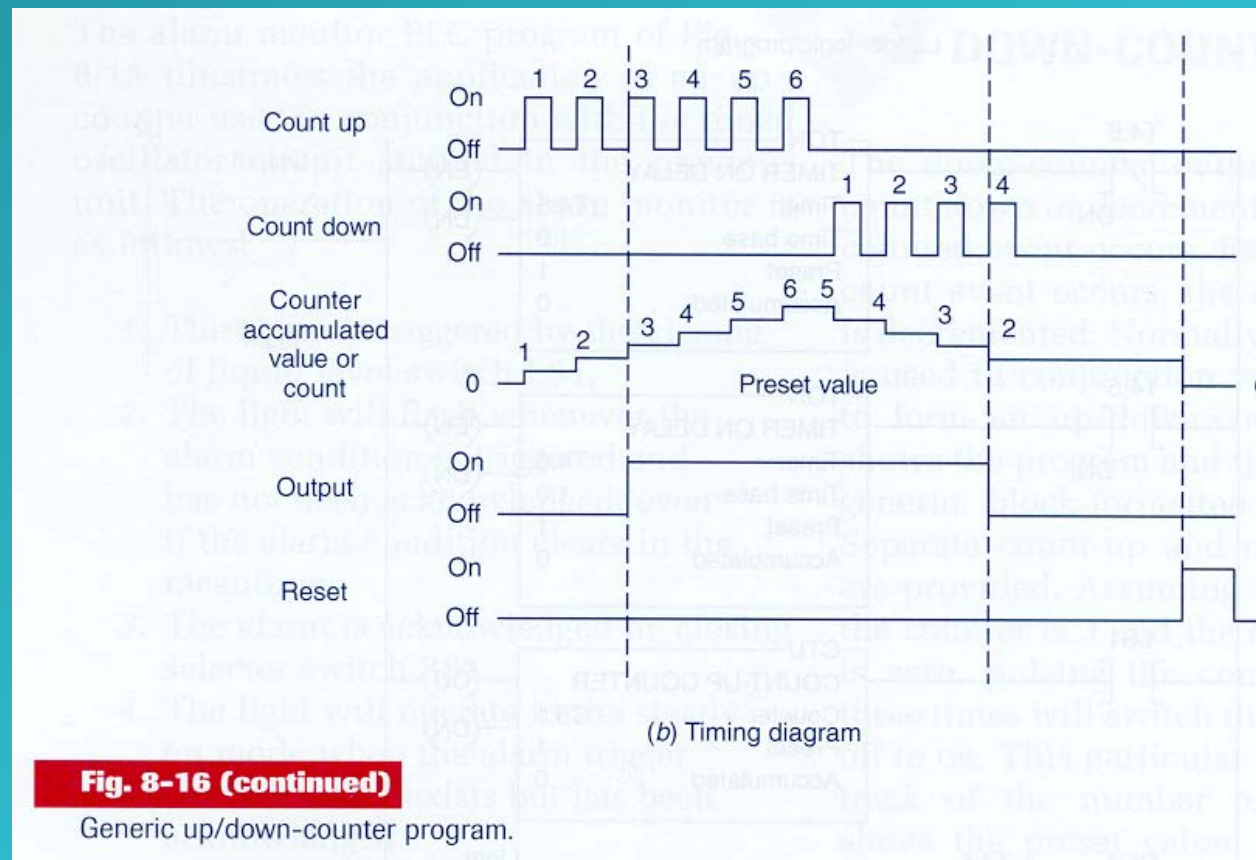
## Diagramas de contactos (cont.)

### *Up/down-counters*



# Diagramas de contactos (cont.)

## *Up/down-counters*



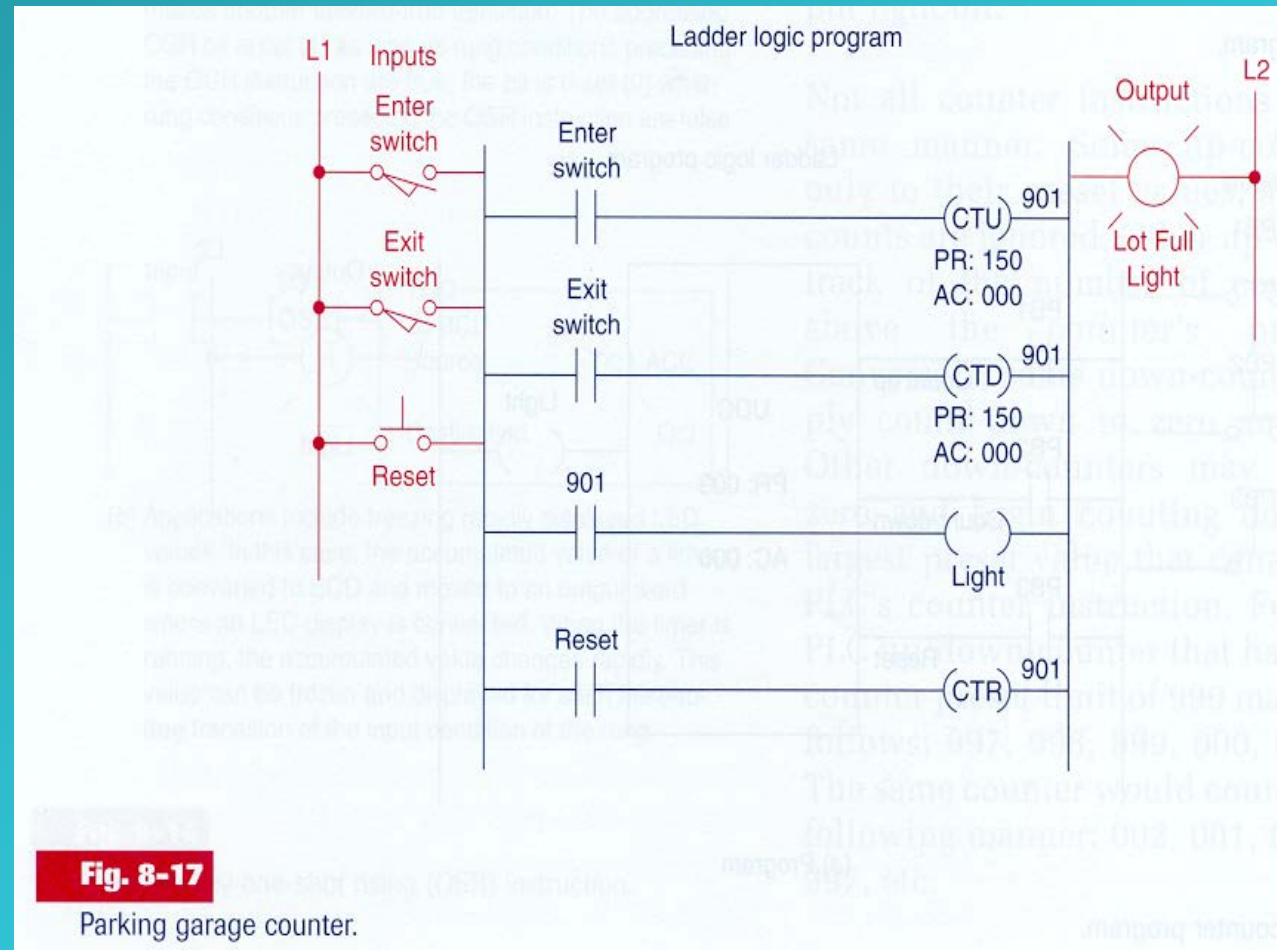


# Diagramas de contactos (cont.)

### *Up/down-counters*

Exemplo:

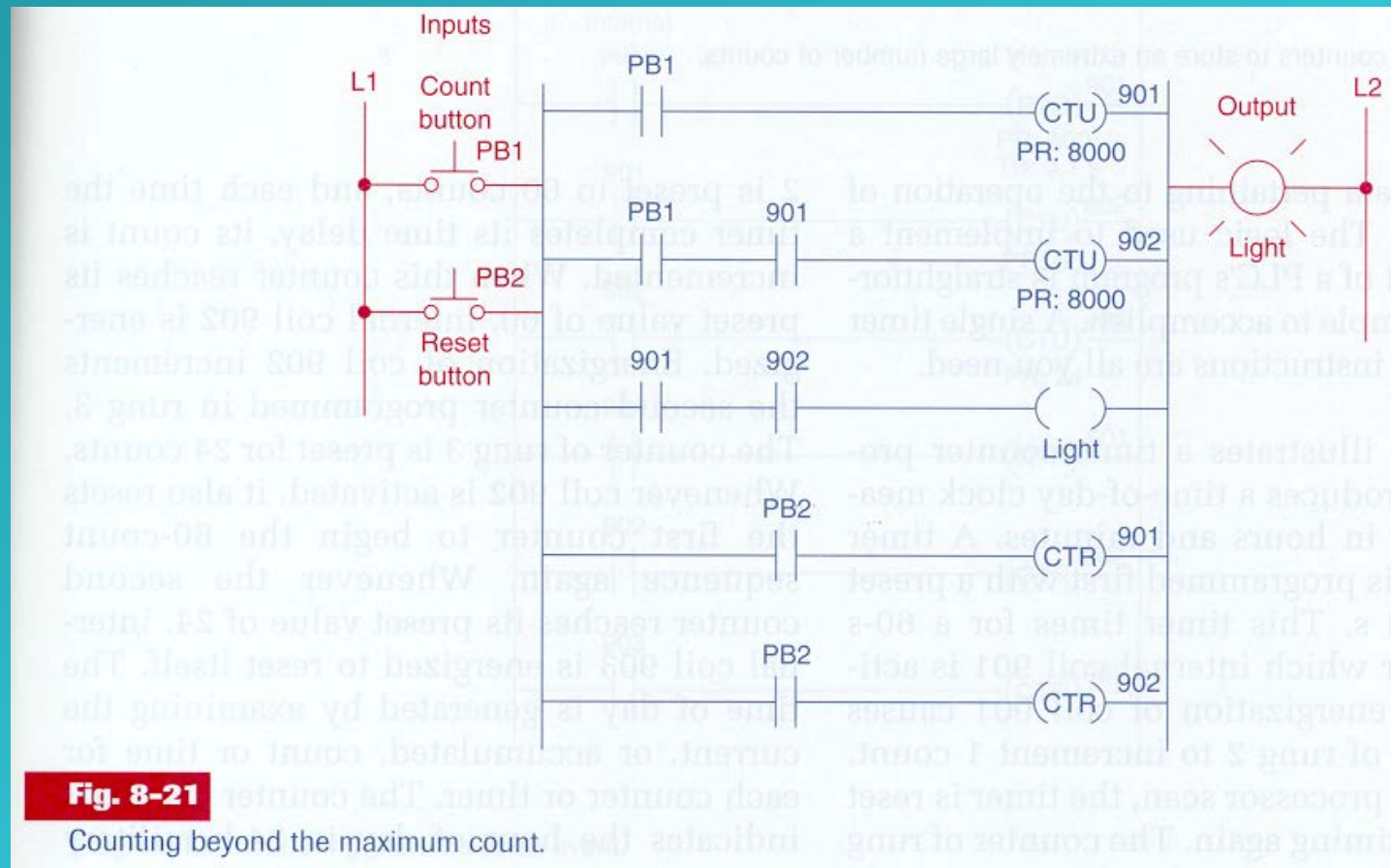
garagem com lotação



# Diagramas de contactos (cont.)

## Contadores em cascata

Exemplo:

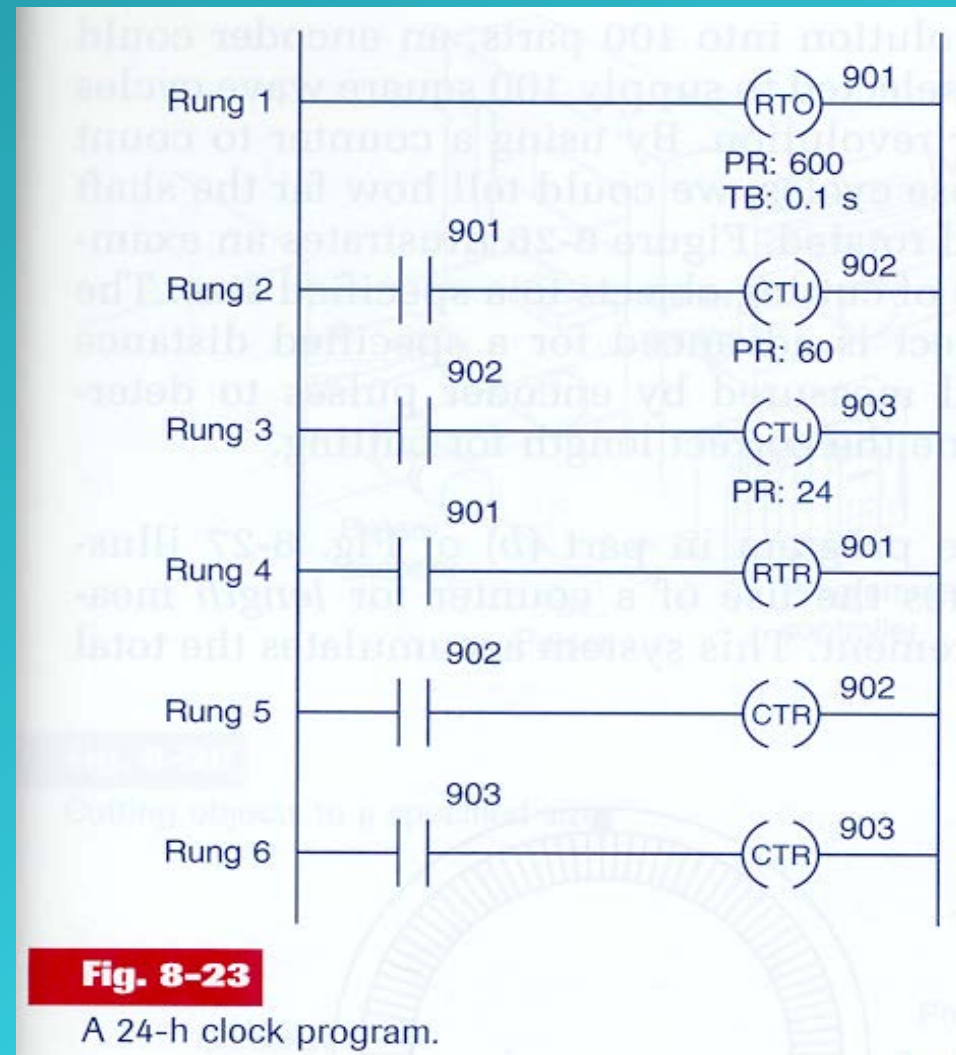


## Diagramas de contactos (cont.)

### Contadores em cascata

Exemplo:

relógio para 24 horas

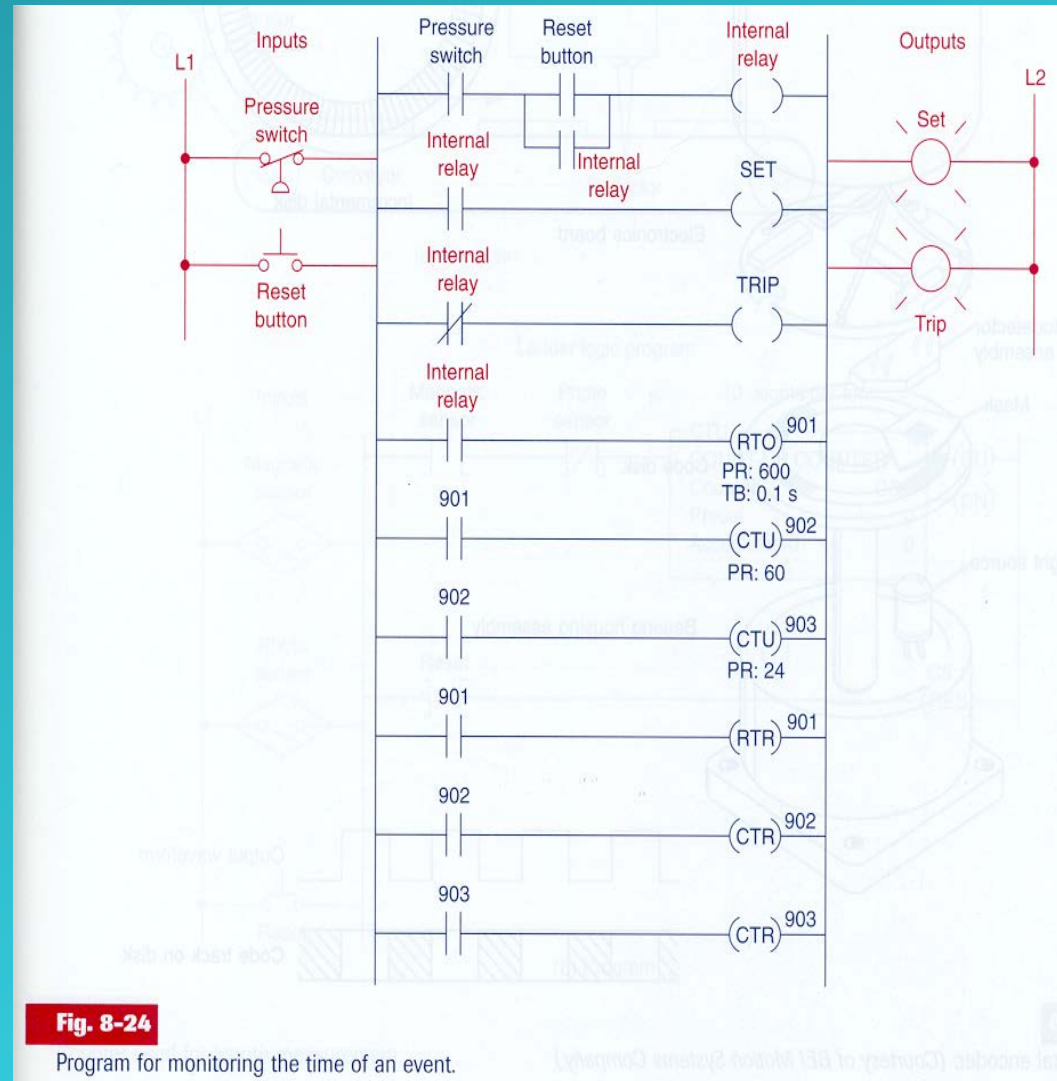


# Diagramas de contactos (cont.)

## Contadores em cascata

Exemplo:

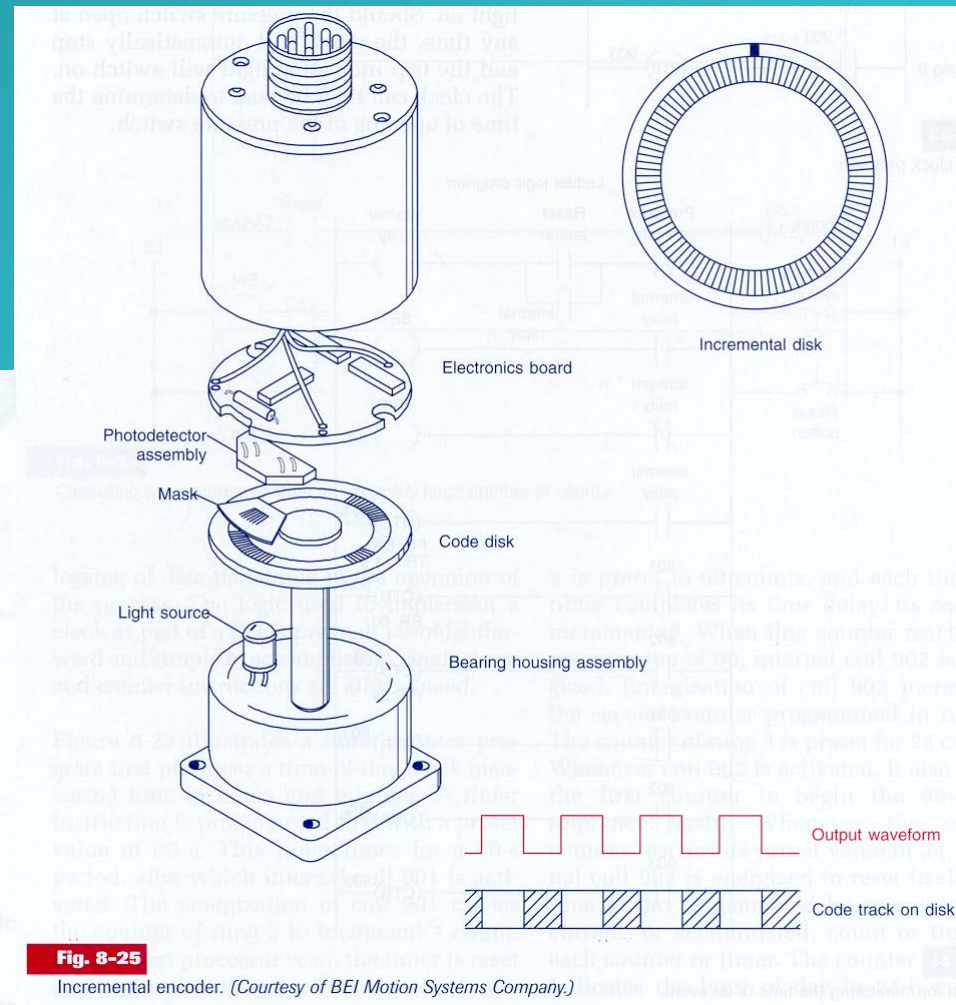
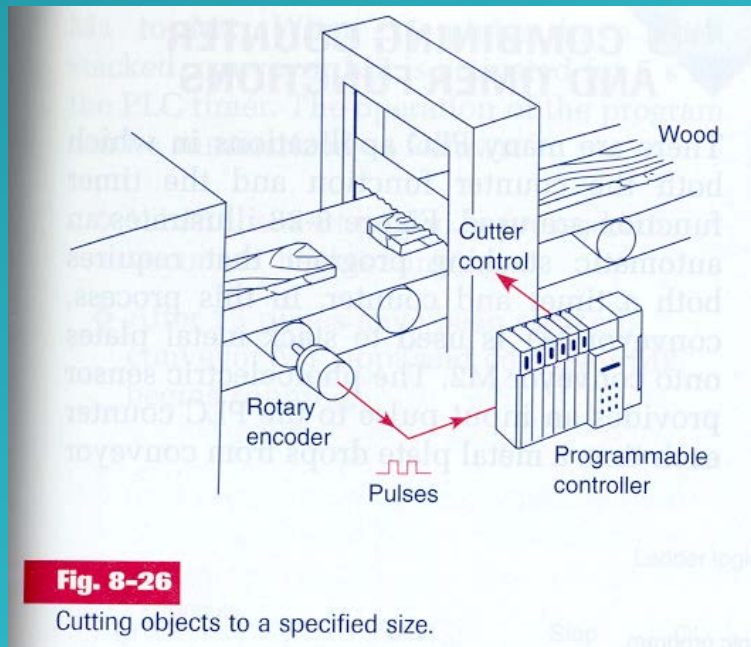
memorizar hora de um evento





# Diagramas de contactos (cont.)

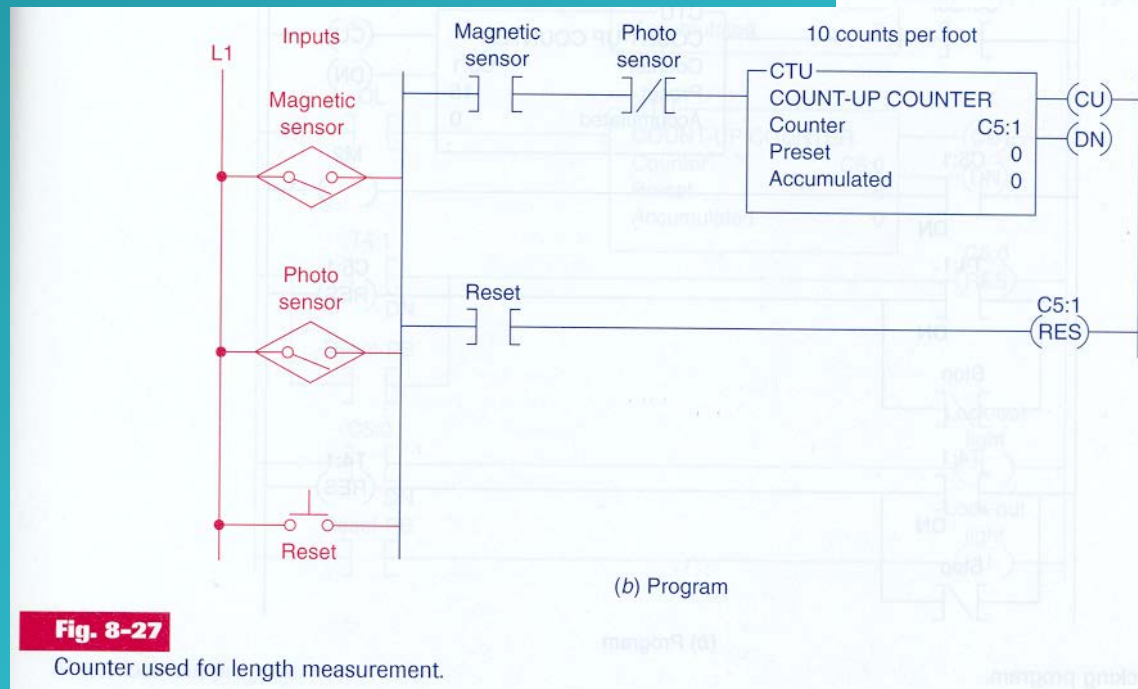
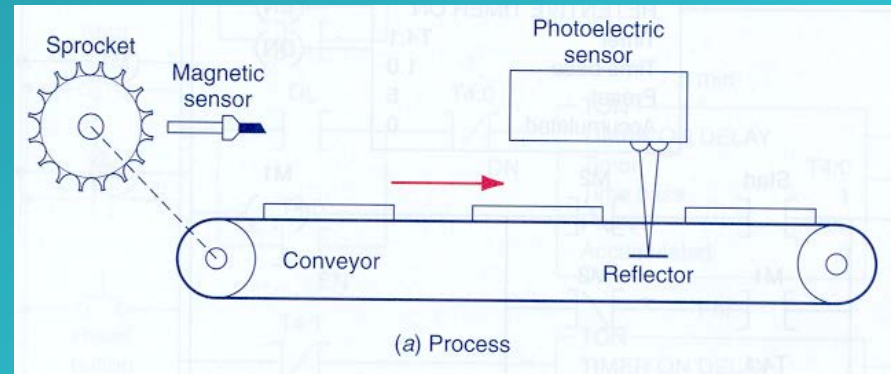
## Encoder incremental



# Diagramas de contactos (cont.)

## Encoder incremental

Exemplo: contador usado para medir um comprimento

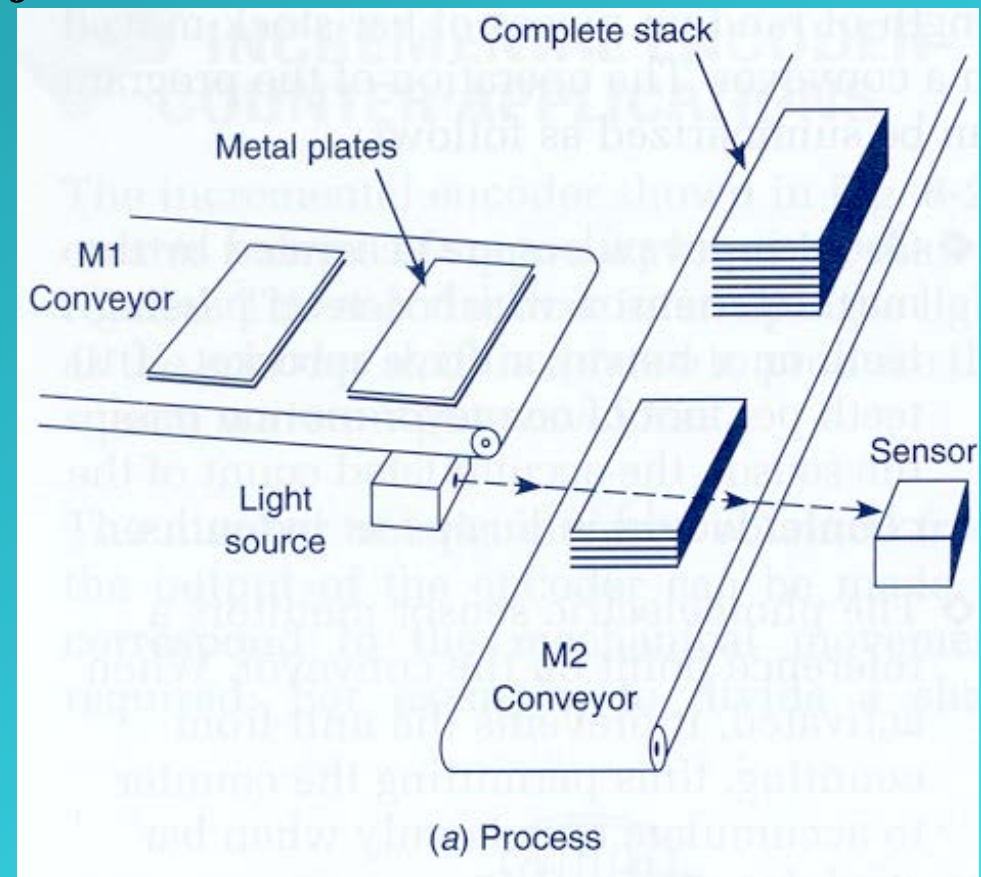


## Diagramas de contactos (cont.)

### Contadores usados em conjunto com *timers*

Exemplo:

- Após botão de *start* M1 começa a funcionar.
- Após 15 placas M1 para e M2 começa a funcionar.
- Após operar M2 por 5 segundos, parar e recomeçar sequencia.

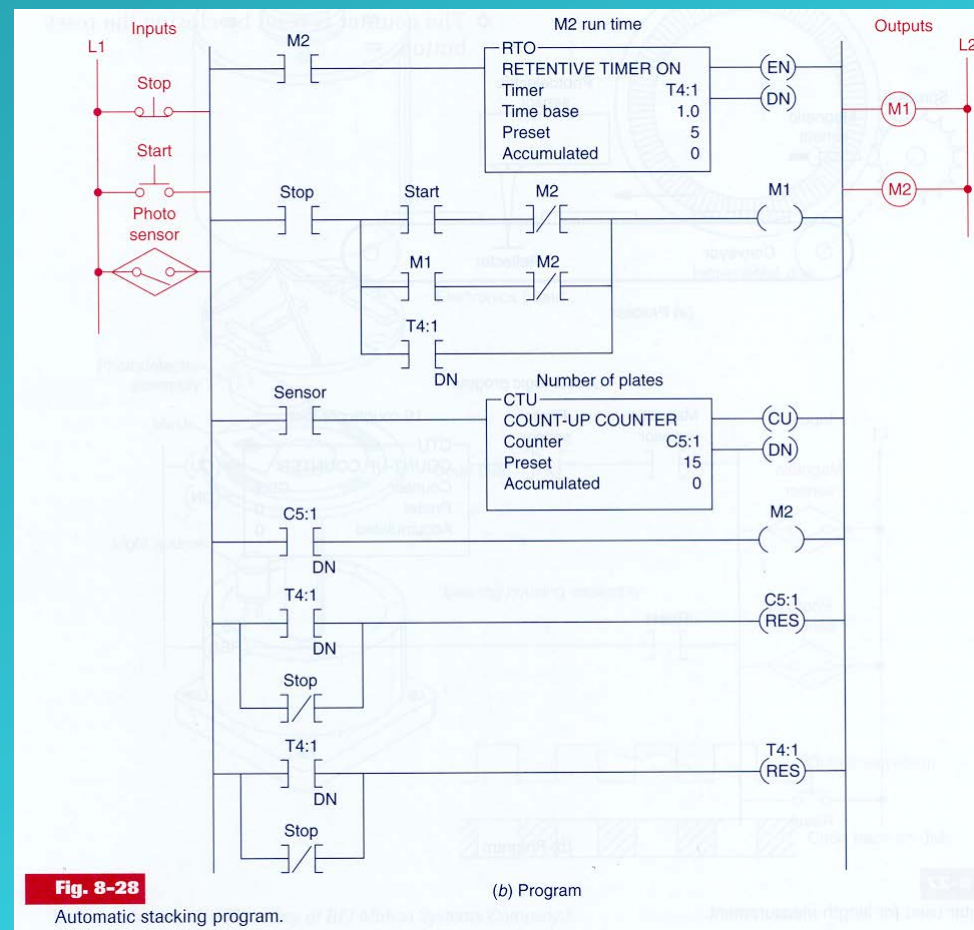


# Diagramas de contactos (cont.)

## Contadores usados em conjunto com *timers*

Exemplo:

- Após botão de *start* M1 começa a funcionar.
- Após 15 placas M1 para e M2 começa a funcionar.
- Após operar M2 por 5 segundos, parar e recomeçar sequencia.



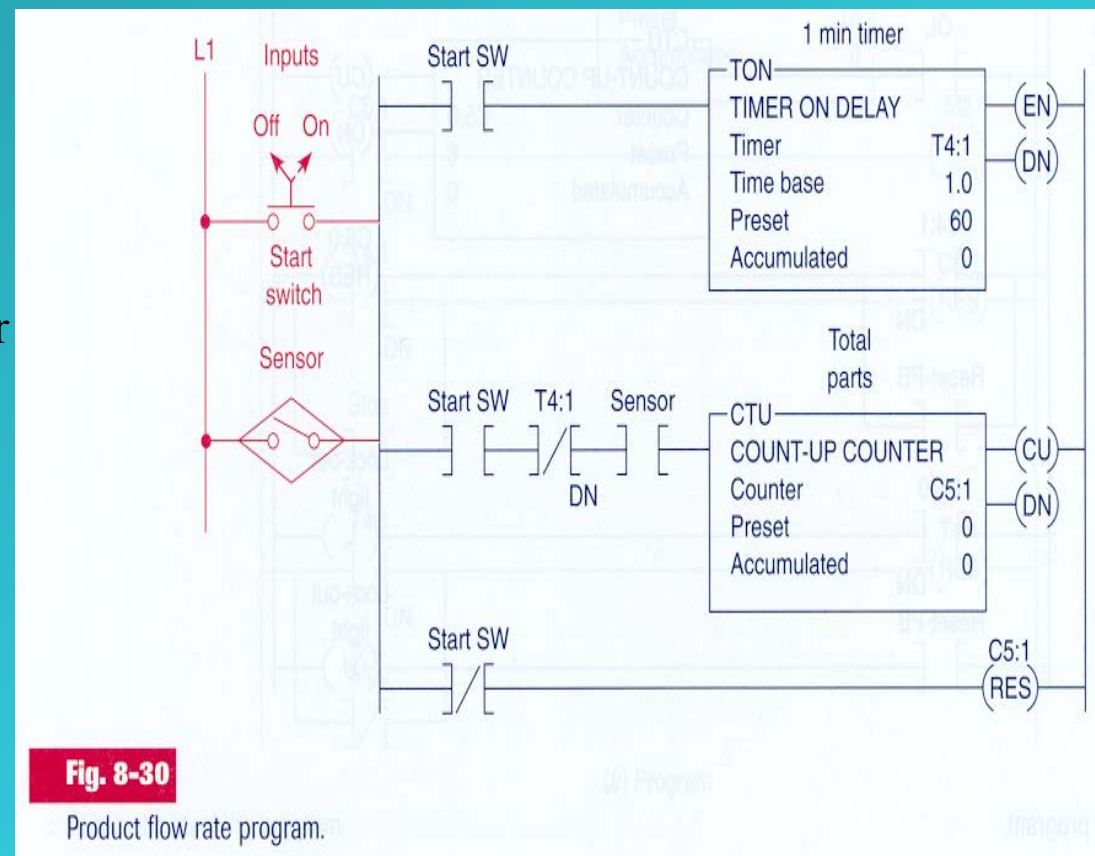


# Diagramas de contactos (cont.)

## Contadores usados em conjunto com *timers*

Exemplo:

- Após botão de *start* o timer e o contador são autorizados.
- O contador é incrementado por cada peça que é processada.
- O *timer* conta 1 minuto. No final o bit de enable interrompe a contagem do contador.

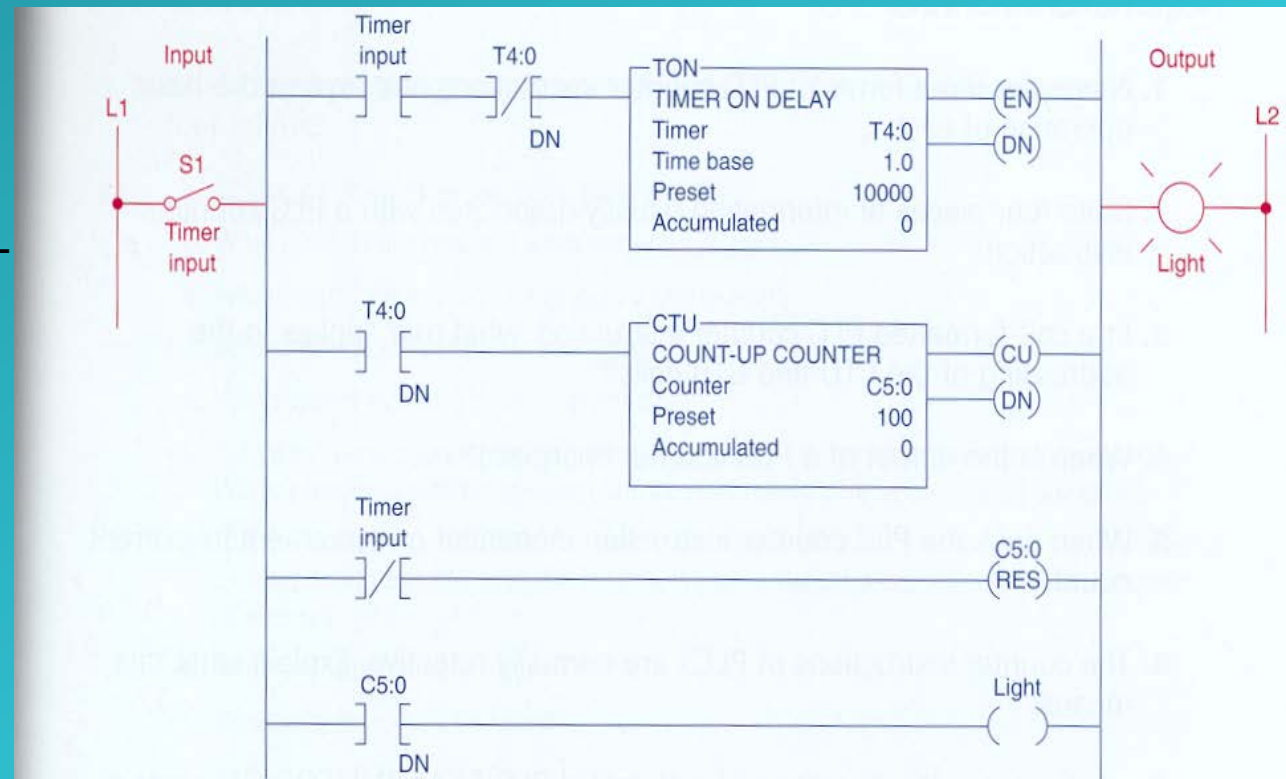


# Diagramas de contactos (cont.)

## Contadores usados em conjunto com *timers*

Exemplo:

timer a comandar um contador para ser possível obter intervalos de tempo grandes.



**Fig. 8-31**

Timer driving a counter to produce an extremely long time-delay period.

# Automação Industrial baseada em PLCs

7ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 7ª Aula

### Cap. 2 - Introdução aos PLCs

- Componentes constituintes dos PLCs.
- Estrutura interna e funcionamento.
- Interfaces de entrada e de saída.

### Cap. 3 - Linguagens de Programação de PLCs

- Linguagem *ladder*.
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.



# Linguagem DOLOG80

## PLC AEG A020 Plus:

### Entradas:

- 20 binárias com fotoacopladores
- 4 analógicas (8 bits, 0-10V)

### Saídas:

- 16 binárias por relés de 2A
- 1 analógica (8 bits, 0-10V)

Interface de programação: RS232

### Processador:

- 8031
- 2Kbytes de RAM
- 2K de EEPROM => 896 instruções.

**Tempo médio de ciclo: 6.5 ms**



# Linguagem DOLOG80

## PLC AEG A020 Plus:

### OPERANDOS

- I1 a I20                   entradas binárias
- Q1 a Q16               saídas binárias
- M1 a M128           memórias auxiliares
- T1 a T8               *timers* com base de tempo de 100ms
- T9 a T16 *timers* com base de tempo de 25ms
- C1 a C16              contadores de 16 *bits*



## Linguagem DOLOG80 (cont.)

### Programação por intermédio de Mnemónicas

- Funções Lógicas
- Funções Aritméticas
- Temporizadores
- Contadores

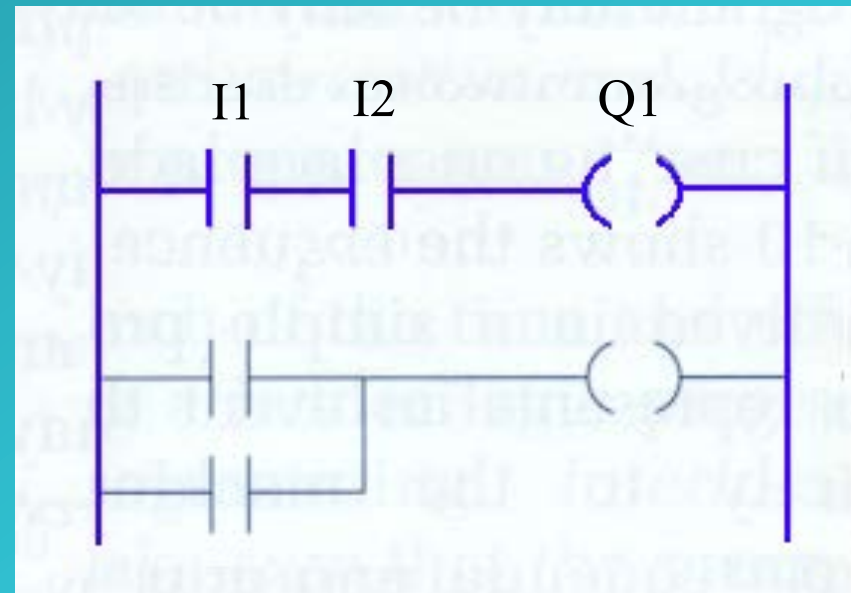


## Linguagem DOLOG80 (cont.)

### Funções Lógicas

AND:

AI1      carrega entrada 1  
AI2      *and* com entrada 2  
=Q1      resultado na saída 1  
...



Exemplo de uso da função AND



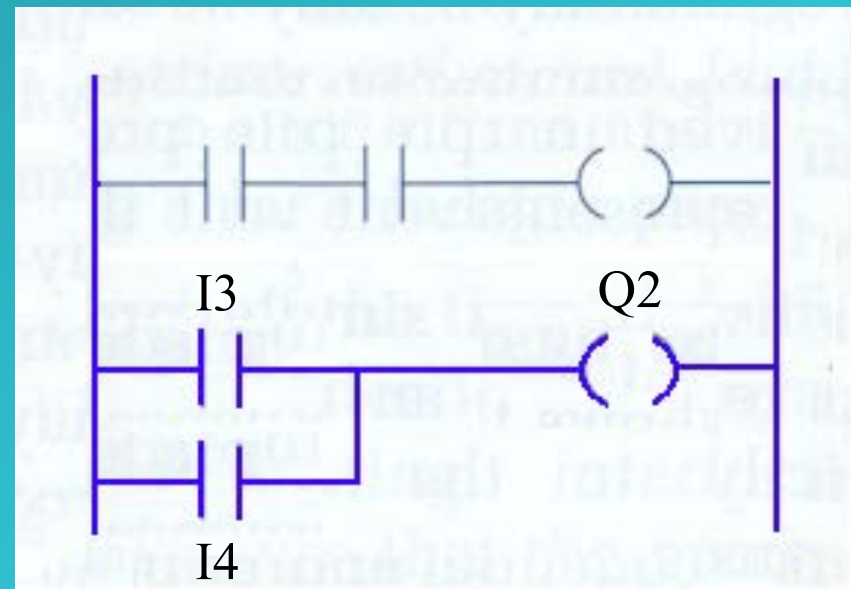
## Linguagem DOLOG80 (cont.)

### Funções Lógicas

OR:

...

AI3      carrega entrada 3  
OI4      *or* com entrada 4  
=Q2      resultado na saída 2  
PE      fim do programa

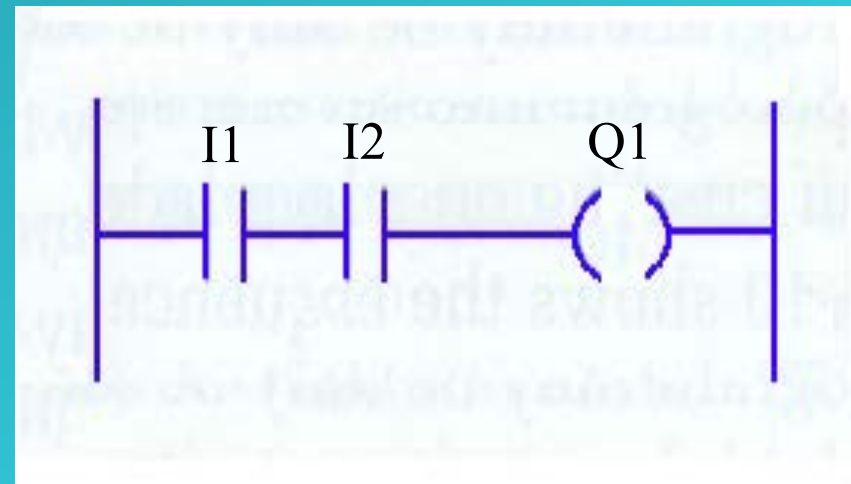


Exemplo de uso da função OR

## Linguagem DOLOG80 (cont.)

### Funcionamento no PLC

HE	programa pára, luz apaga
STN	apaga programa na EEPROM
M	introduz novo programa
1	endereço da primeira instrução
AI1	carrega entrada 1
AI2	<i>and</i> com entrada 2
=Q1	resultado na saída 1
PE	resultado na saída 1
E	fim de introdução de programa
PREC	força caracter de check
S	inicia execução do programa



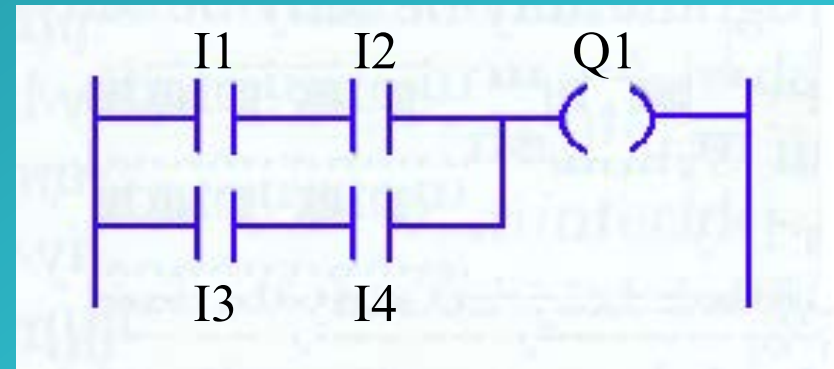
### Exemplo de uso da função AND

## Linguagem DOLOG80 (cont.)

Prioridades:

Pretende-se implementar:

$Q1 = (I1 \text{ and } I2) \text{ or } (I3 \text{ and } I4)$



1ª solução:

AI1  
AI2  
OI3  
AI4  
=Q1  
PE

Errada!...

2ª solução:

AI1  
AI2  
O(  
AI3  
AI4  
)  
=Q1  
PE

3ª solução:

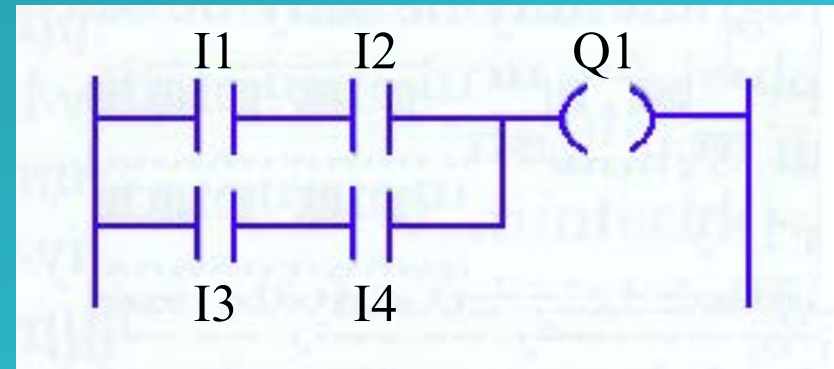
utilização de memórias  
auxiliares (markers)

## Linguagem DOLOG80 (cont.)

Marker (memórias auxiliares)

Pretende-se implementar:

$$Q1 = (I1 \text{ and } I2) \text{ or } (I3 \text{ and } I4)$$



3ª solução:

AI1

AI2

=M1      guarda resultado em M1

AI3

AI4

=M2      guarda resultado em M2

...

...

AM1      carrega M1

OM2      *or* com M2

=Q1      resultado em Q1

PE



## Linguagem DOLOG80 (cont.)

### Marker (memórias auxiliares)

- M1
- ... memórias auxiliares (livres)
- M122
- M123 sinaliza a “1” se *overload* ou curto-circuito
- M124 sinaliza a “1” se *undervoltage* ou NVRAM
- M125 sinal a pulsar à frequência de 1.25 Hz
- M126 sinal a pulsar à frequência de 2.5 Hz
- M127 sinal a pulsar à frequência de 5 Hz
- M128 sinal *resettable* (por exemplo para normalização)

## Linguagem DOLOG80 (cont.)

### Funções Lógicas

NOT:

nas entradas

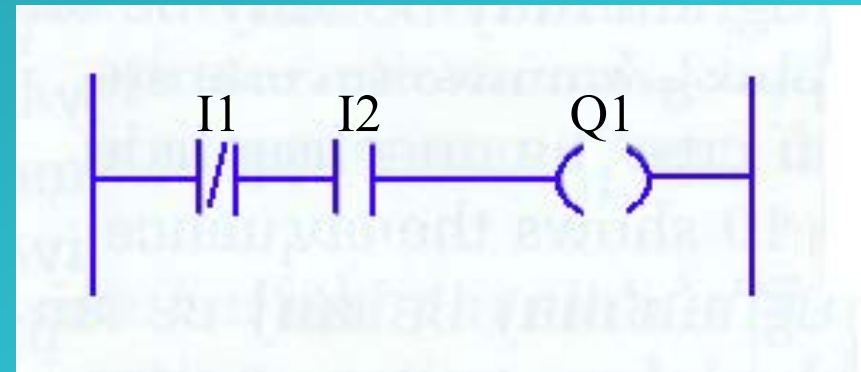
AN em vez de And

ON em vez de Or

ou nas saídas

=NQ

Exemplo:



ANI1 carrega entrada 1, negada

AI2 *and* com entrada 2

=Q1 resultado na saída 1

PE

# Linguagem DOLOG80 (cont.)

## Exemplo

AI1

AI2

A(

OM1

OI4

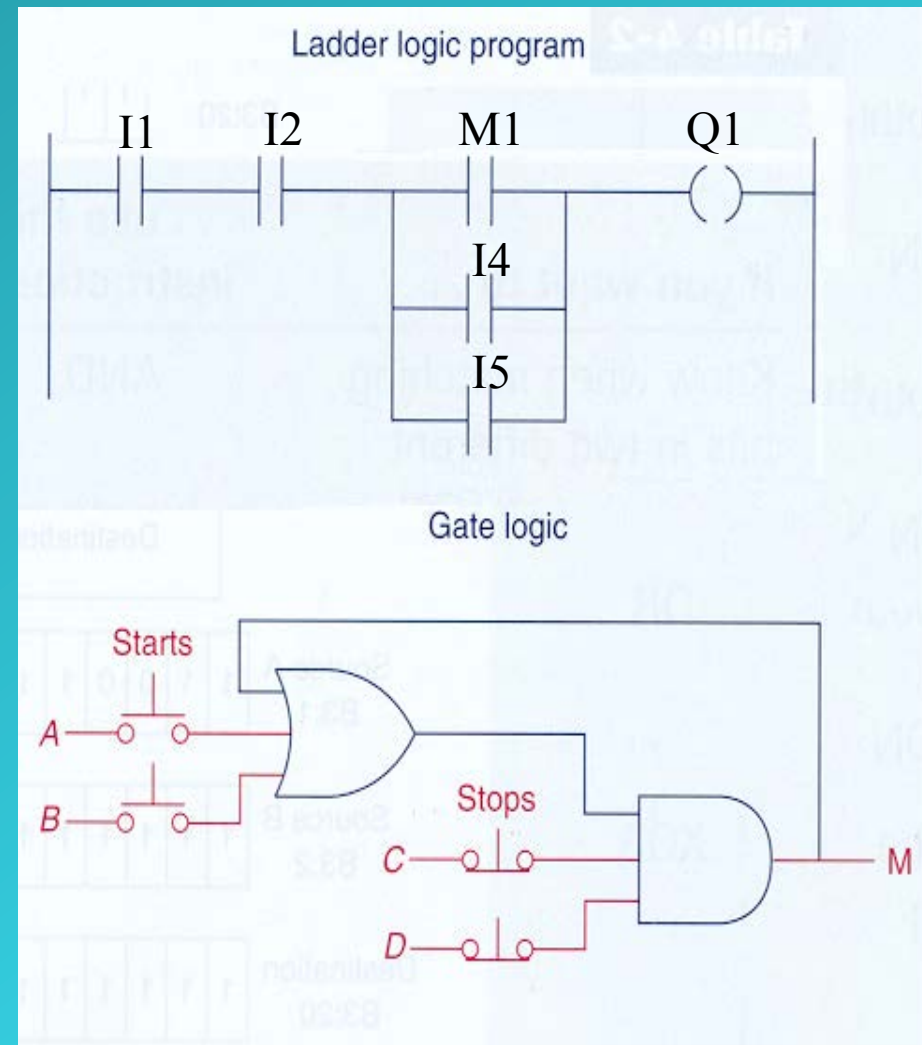
OI5

)=M1      guarda resultado em M1

AM1

=Q1

PE



## Linguagem DOLOG80 (cont.)

### Relés com *latch* e *unlatch*

AI1

SLQ1

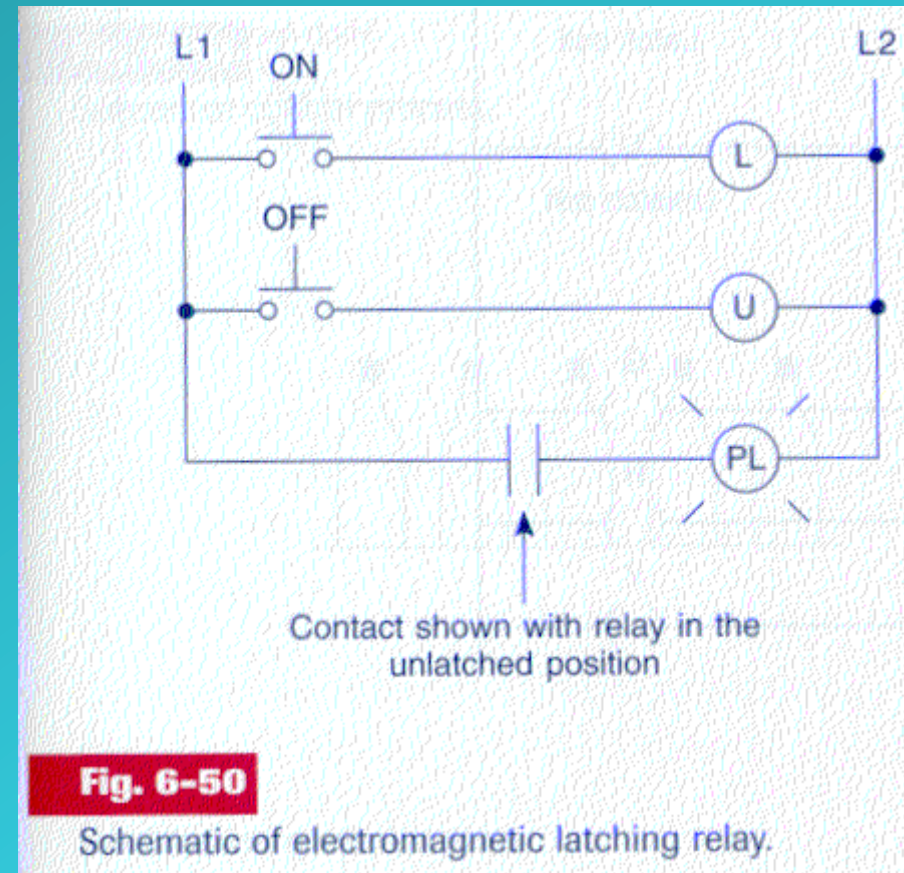
*set do latch*

AI2

RLQ1

*reset do latch*

PE



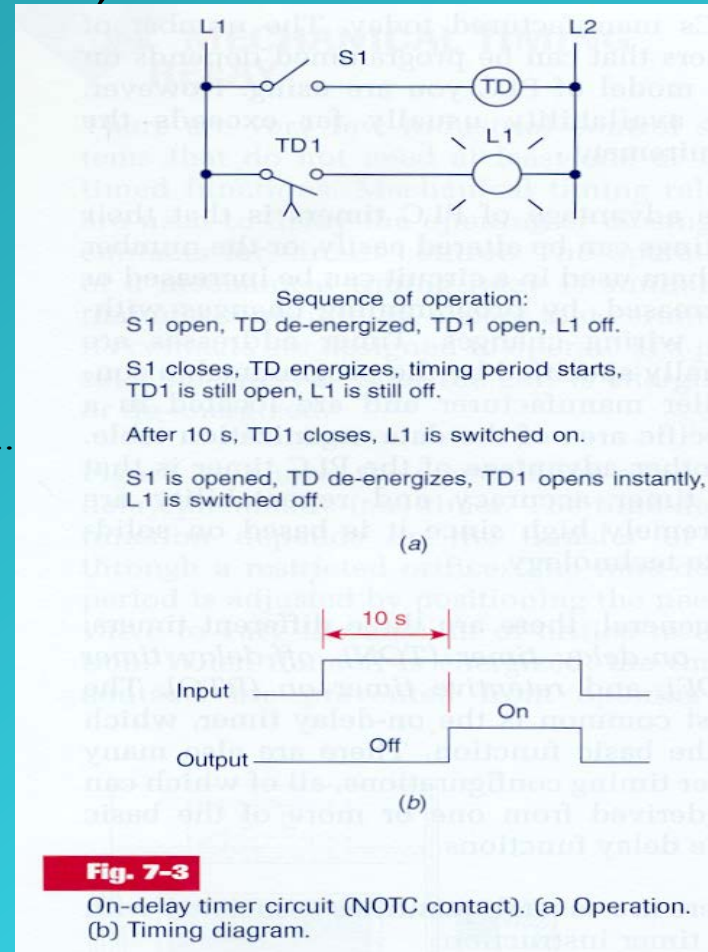
SW ON= I1; SW OFF=I2; PL=Q1



# Linguagem DOLOG80 (cont.)

## *Timers on delay*

- AI1
- =T1                      inicia timer
- AT1                      no fim da contagem do timer...
- =Q1                      activa saída
- LDV100
- =TSW1                    atraso de  $100 \times 100\text{ms} = 10\text{s}$
- PE

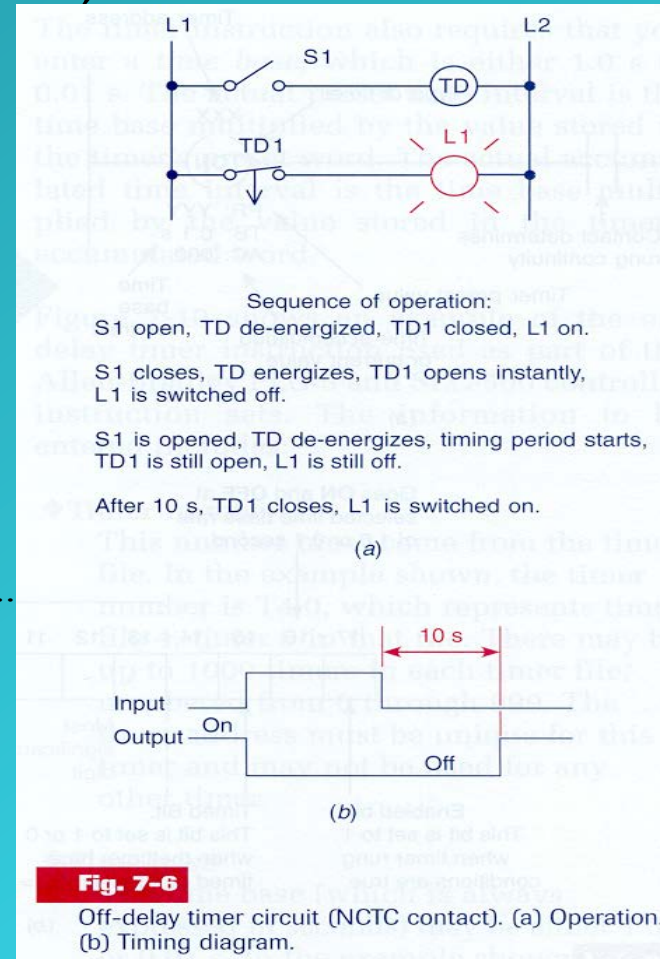


S1= I1; L1=Q1

# Linguagem DOLOG80 (cont.)

## Timers on delay

- AI1
- RLQ1                      desactiva saída
- ANI1
- =T2                          *switch off* activa timer
- AT2                          no fim da contagem do timer...
- SLQ1                        activa saída
- LDV100
- =TSW2                      atraso de  $100 \times 100\text{ms} = 10\text{s}$
- PE



S1= I1; L1=Q1

## Linguagem DOLOG80 (cont.)

### Contadores decrescentes de 16 bits

AI1

=P1

entrada de contagem

AC1

detecta fim de contagem

=Q1

AI2

=Z1

faz reset ao contador

LDV6

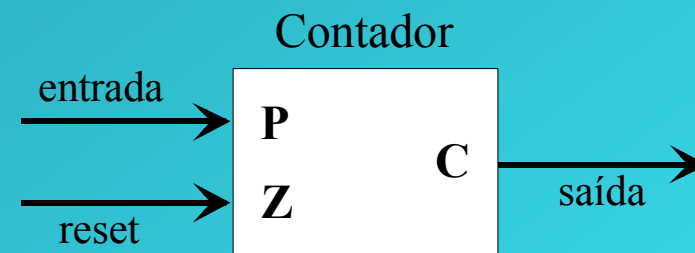
=CSW1

carrega valor de contagem

PE

(LDCAW1

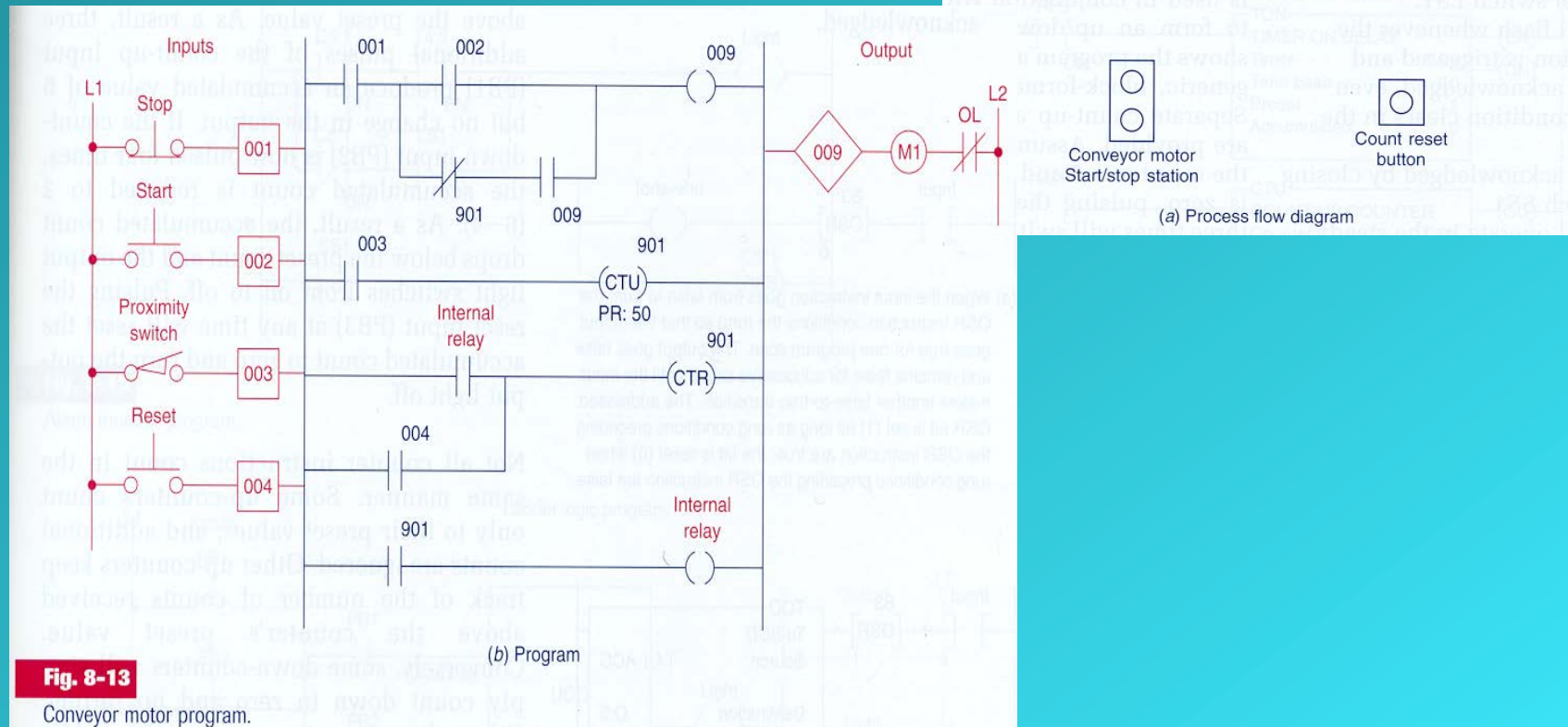
carrega o valor do contador)



## Linguagem DOLOG80 (cont.)

### Contadores

Exemplo:



**Fig. 8-13**

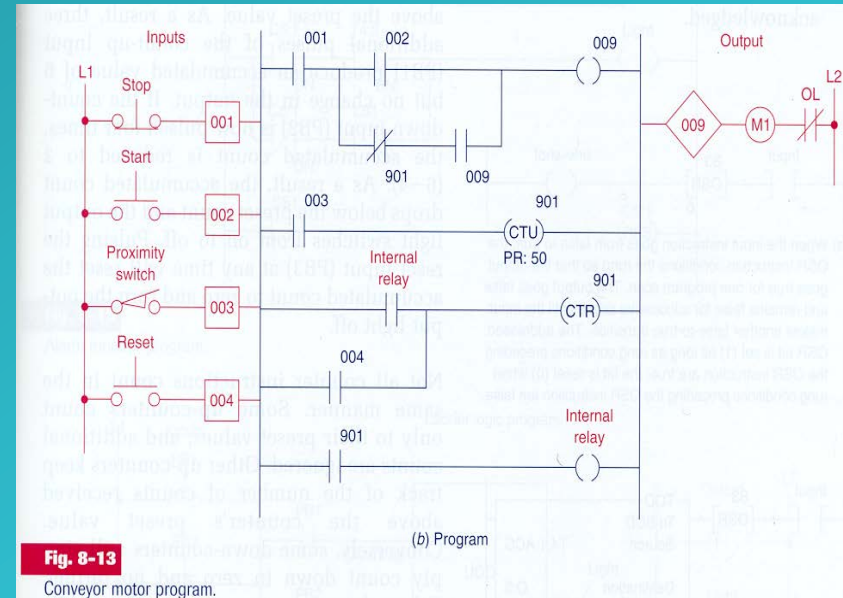
Conveyor motor program.



# Linguagem DOLOG80 (cont.)

## Exemplo de Contadores:

<b>ANI1</b>	<b>AI3</b>	<b>LDV50</b>
<b>A(</b>	<b>=P9</b>	<b>=CSW9</b>
<b>OI2</b>	<b>NO</b>	<b>PE</b>
<b>O(</b>	<b>OM1</b>	
<b>ANC9</b>	<b>OI4</b>	
<b>AQ9</b>	<b>=Z9</b>	
<b>)</b>	<b>NO</b>	
<b>)</b>	<b>AC9</b>	
<b>=Q9</b>	<b>=M1</b>	
<b>...</b>	<b>...</b>	



**Legenda:**

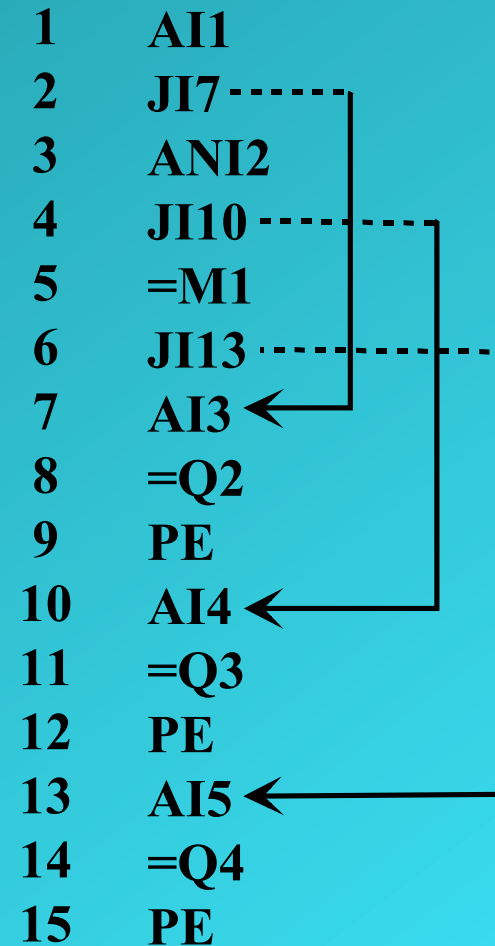
- Stop* = I1
- Start* = I2
- Proximidade* = I3
- Reset* = I4
- Contador* = C9
- Internal relay* = M1
- Motor* = Q9

## Linguagem DOLOG80 (cont.)

### Controlo de Fluxo:

### Saltos condicionais

Se a instrução anterior fôr verdadeira efectuar o salto para a linha indicada.



## Linguagem DOLOG80 (cont.)

### Funções Aritméticas

•Atribuição	LDV <sub>xxx</sub>
•Multiplicação	MULV <sub>xxx</sub>
•Adição	ADDV <sub>xxx</sub>
•Subtração	SUBV <sub>xxx</sub>
•Divisão	DIV <sub>xxx</sub>
•Teste de igualdade	EQV <sub>xxx</sub>
•Comparação	LTV <sub>xxx</sub> (menor que) GTV <sub>xxx</sub> (maior que)

Existem variáveis auxiliares de 16 bits MW1 a MW50 para armazenar estes valores. MW51 a MW54 contém o tempo desde que o PLC foi ligado (seg:min:horas:dias).

# Automação Industrial baseada em PLCs

8ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)



## 8ª Aula

### Cap. 3 - Linguagens de Programação de PLCs

- Linguagem *ladder*.
- Descrição dos componentes de uma linguagem típica (DOLOG80 e/ou STEP5).
- Estruturas de controlo de fluxo.

### Cap. 4 - GRAFCET

- A norma GRAFCET.
- Técnicas de modelação utilizando GRAFCET.

### Cap. 5 - Sistemas de Eventos Discretos

- Modelação de sistemas de eventos discretos.
- Autómatos.
- Redes de Petri.

# A norma GRAFCET (1975)

## *Sequential Function Chart I.E.C. 848 Standard* (1983 e 1988)

### Etapas (*steps*)

Inactiva

1

Activa

2

.

Inicial

3

Podem associar-se **acções** às etapas.

### Transições

Simple

(1)  R<sub>1</sub>

Joint

(2)  R<sub>2</sub>

Fork

(3)  R<sub>3</sub>

Joint e fork

(3)  R<sub>3</sub>

A cada transição pode associar-se uma função lógica de receptividade.

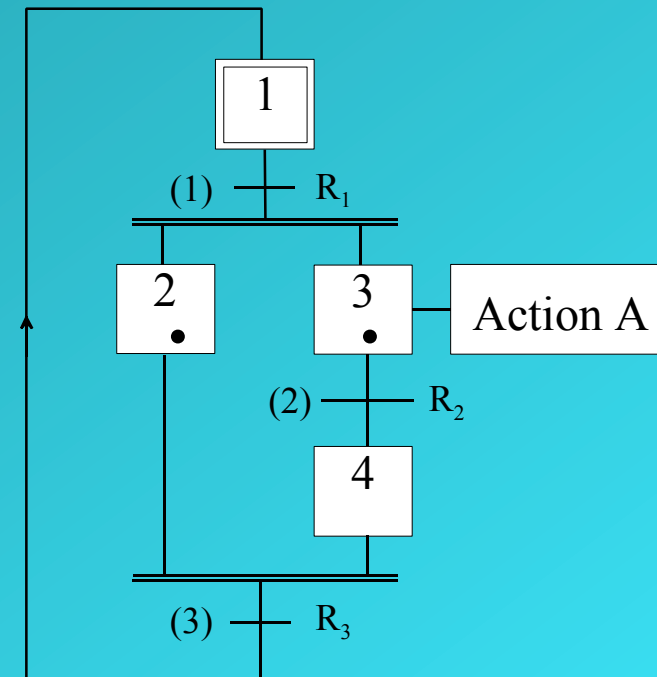
# A norma GRAFCET

## *Sequential Function Chart I.E.C. 848 Standard*

### Ligações orientadas (arcos)

Em um *grafcet*:

- Um arco pode ligar etapas a transições;
- Um arco pode ligar transições a etapas.
- Uma etapa pode não ter nenhuma transição de entrada (fonte);
- Uma etapa pode não ter nenhuma transição de saída (sorvedouro).
- O mesmo se passa para as transições.



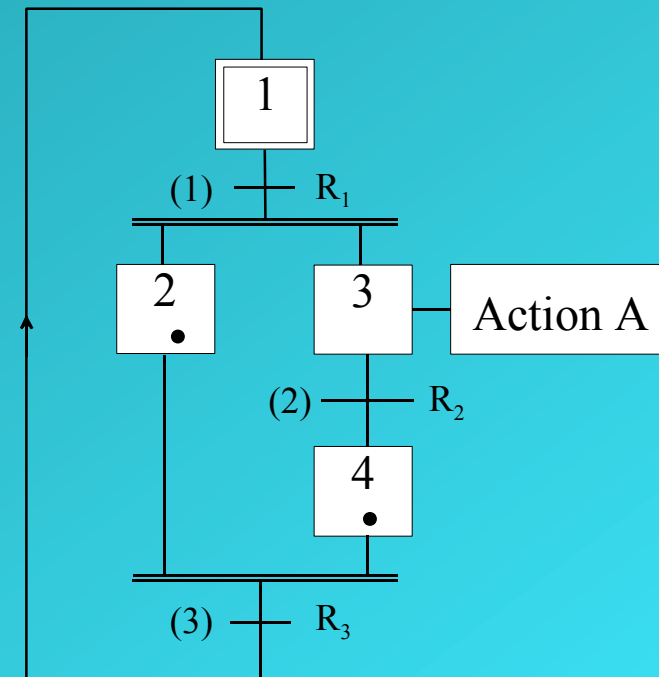
# A norma GRAFCET

## *Sequential Function Chart I.E.C. 848 Standard*

O conjunto de marcações de um *grafcet* constitui o **estado**.

**Questão:**

Como evolui o estado de um *grafcet*?





# A norma GRAFCET

## Evolução do estado de um grafcet:

- **Regra 1: Estado inicial**

Caracteriza-se pelas etapas activas no início de operação (pelo menos uma).

- **Regra 2: Transposição de uma transição**

Uma transição está permitida (ou activa) quando todas as etapas ligadas na sua entrada estão activas (caso contrário está inactiva). Uma transição só pode ser transposta se está activa e se a condição (função de receptividade) que lhe está associada for verdadeira.

- **Regra 3: Evolução das etapas activas**

A transposição de uma transição leva à desactivação de todas as etapas que a precedem e à activação de todas as etapas que lhe sucedem.

- **Regra 4: Simultanêidade da transposição de transições**

Todas as transições activas são transpostas simultaneamente.

- **Regra 5: Activação e desactivação simultânea de uma etapa**

Neste caso a activação é prioritária.

# A norma GRAFCET

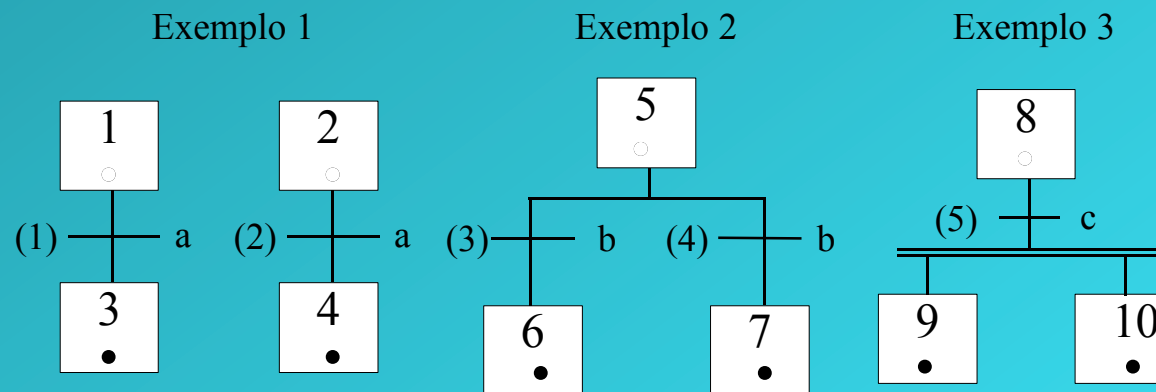
## Evolução do estado de um grafcet:

•**Regra 2a:**

Todas as transições activas são transpostas imediatamente.

•**Regra 4:**

Transições simultaneamente activas são transpostas simultaneamente.

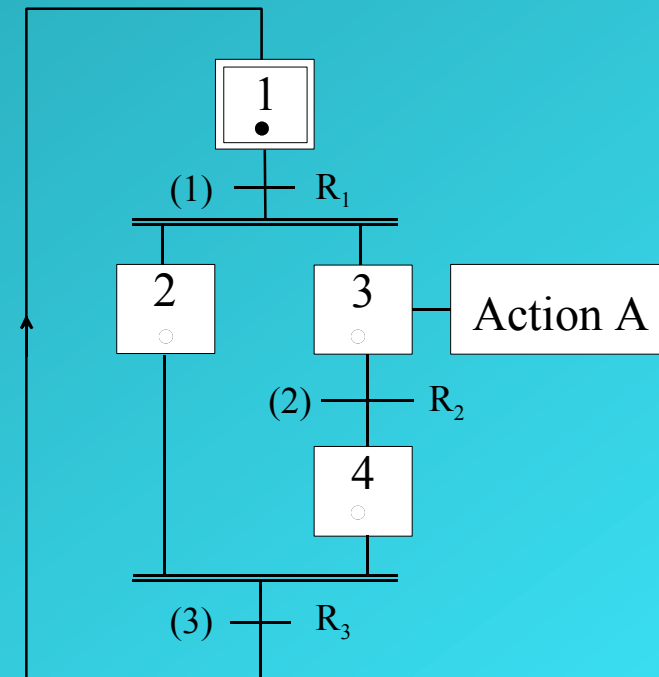


# A norma GRAFCET

## *Sequential Function Chart I.E.C. 848 Standard*

Exemplo de evolução do estado de um *grafcet*

(Acção ao nível. Também podem ser acções à transição)

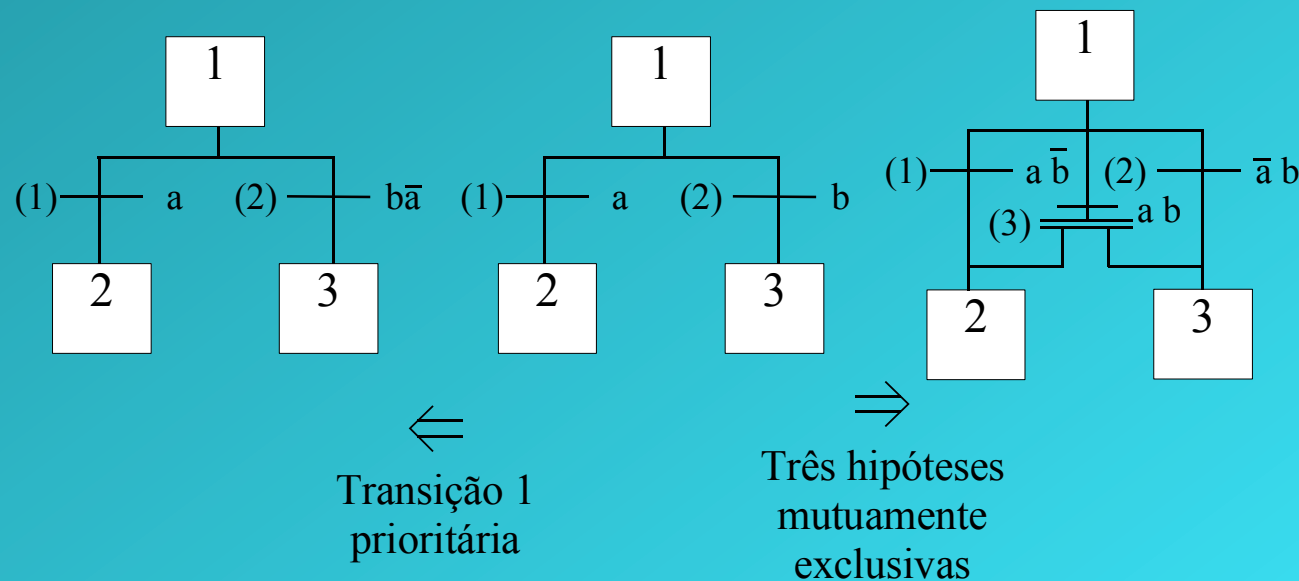


# A norma GRAFCET

## Evolução do estado de um grafcet:

### Conflitos:

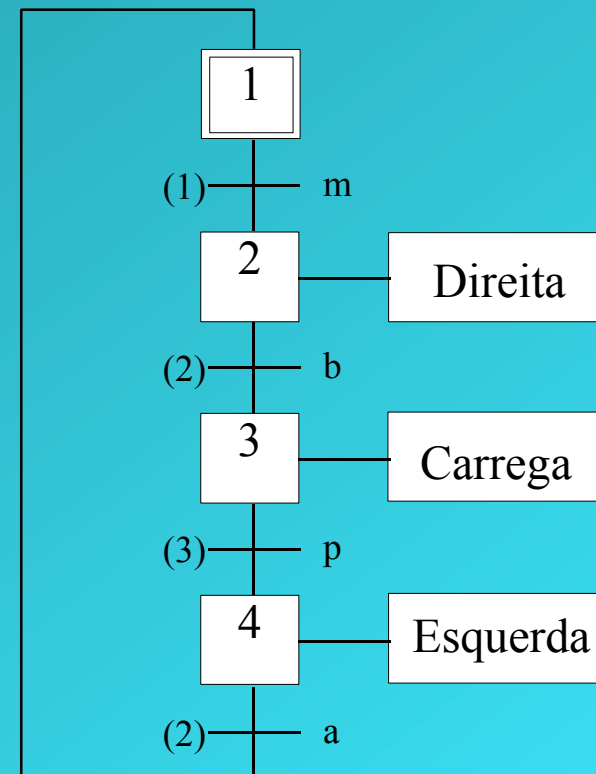
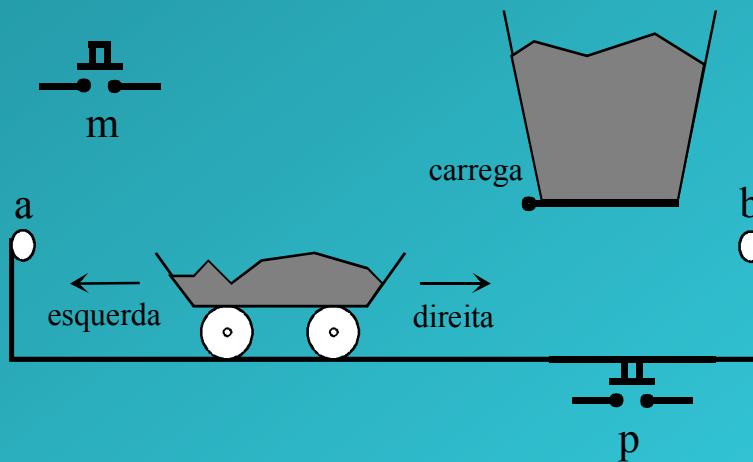
Existem quando a validação de mais do que uma transição depende de uma mesma etapa ou se várias funções de receptividade podem tornar-se verdadeiras ao mesmo tempo.





# A norma GRAFCET

Exemplo de modelação de um sistema de controlo



# A norma GRAFCET

Exemplo de modelação de um sistema de controlo

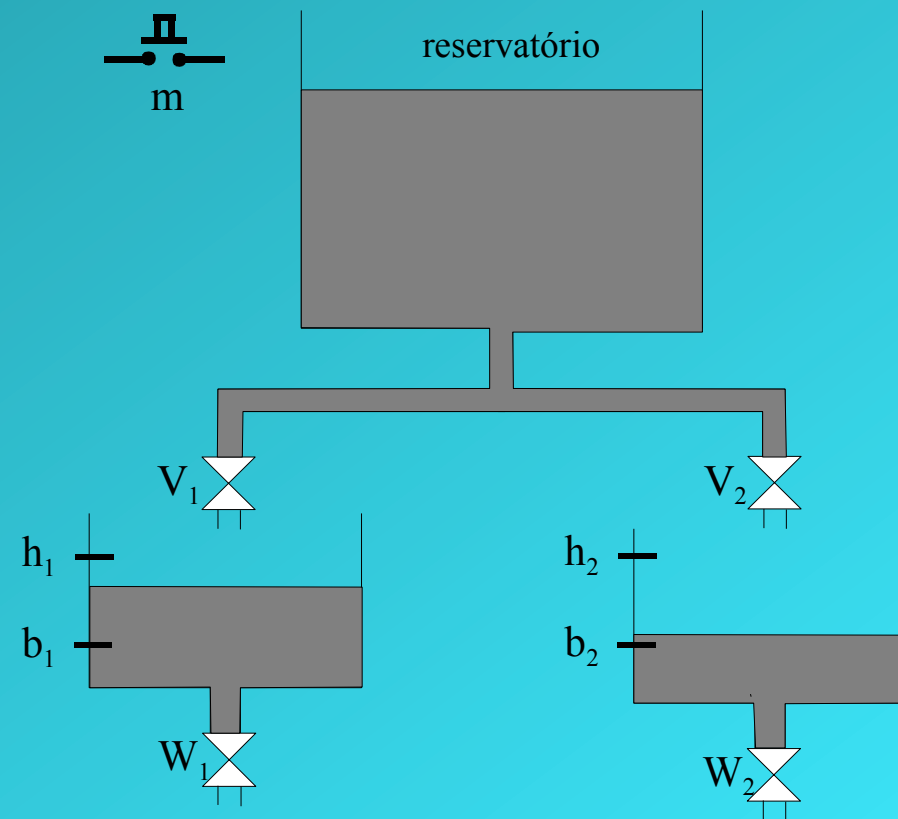
Sensores:

$m$

$b_1, h_1, b_2$  e  $h_2$

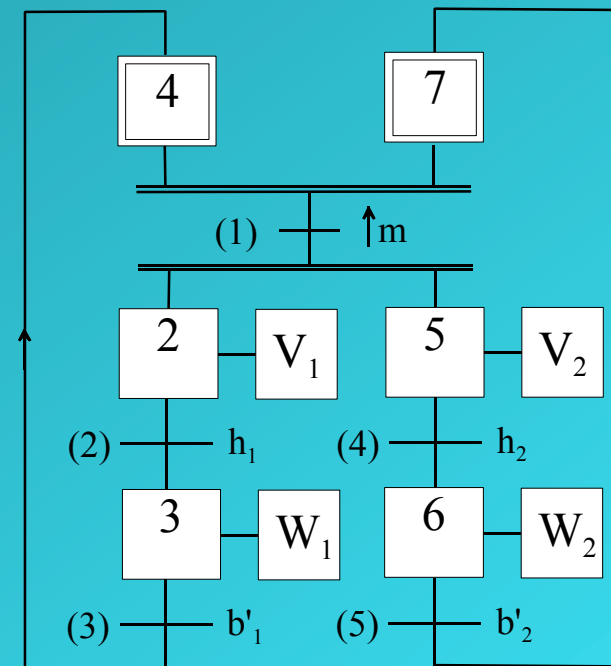
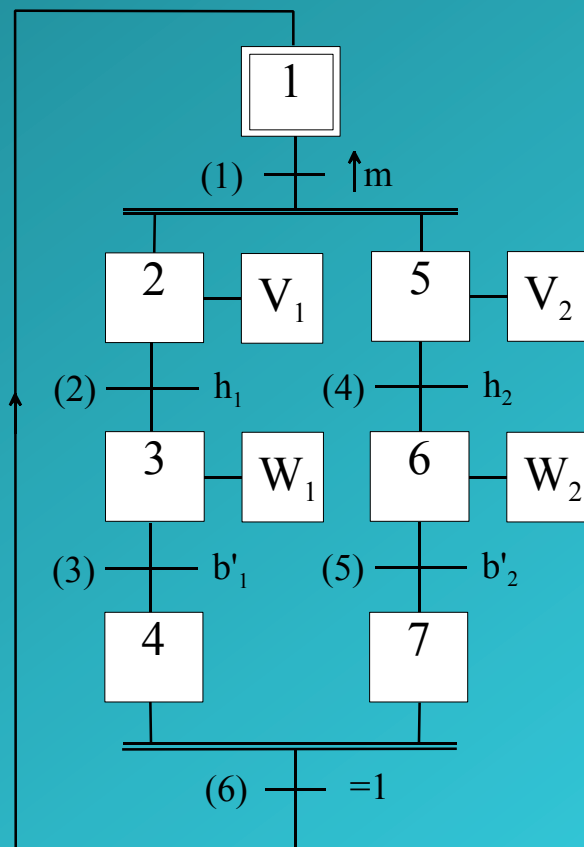
Actuadores:

$V_1, V_2, W_1$  e  $W_2$



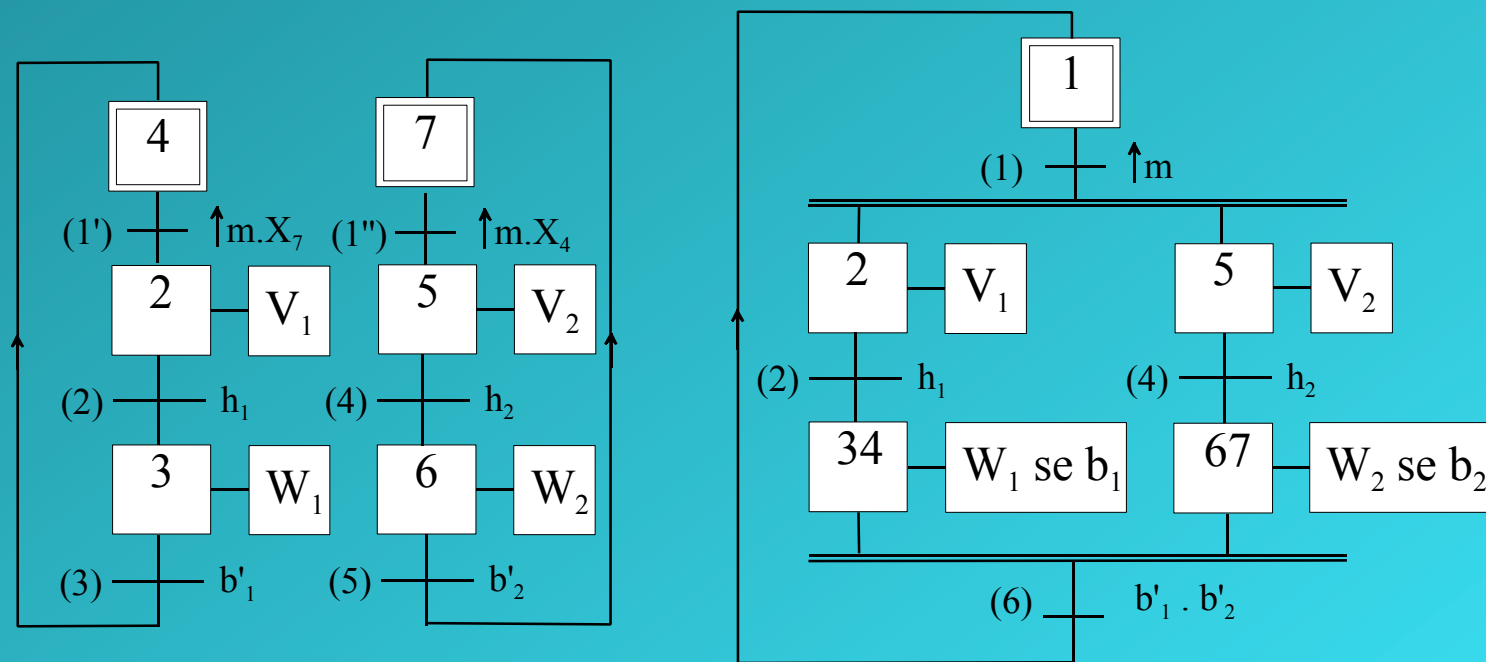
# A norma GRAFCET

Exemplo de modelação de um sistema de controlo



# A norma GRAFCET

Exemplo de modelação de um sistema de controlo





## A norma GRAFCET

### Eventos e propriedades

$$\uparrow a = \downarrow a'$$

$$\uparrow a . a = \uparrow a \quad \uparrow a . a' = 0 \quad \downarrow a . a' = \downarrow a \quad \downarrow a . a = 0$$

$$\uparrow a . \uparrow a = \uparrow a \quad \uparrow a . \uparrow a' = 0$$

$$\uparrow(a . b) = \uparrow a . b + \uparrow b . a \quad \uparrow(a + b) = \uparrow a . b' + \uparrow b + a'$$

$$\uparrow(a . b) . \uparrow(a . c) = \uparrow(a . b . c)$$

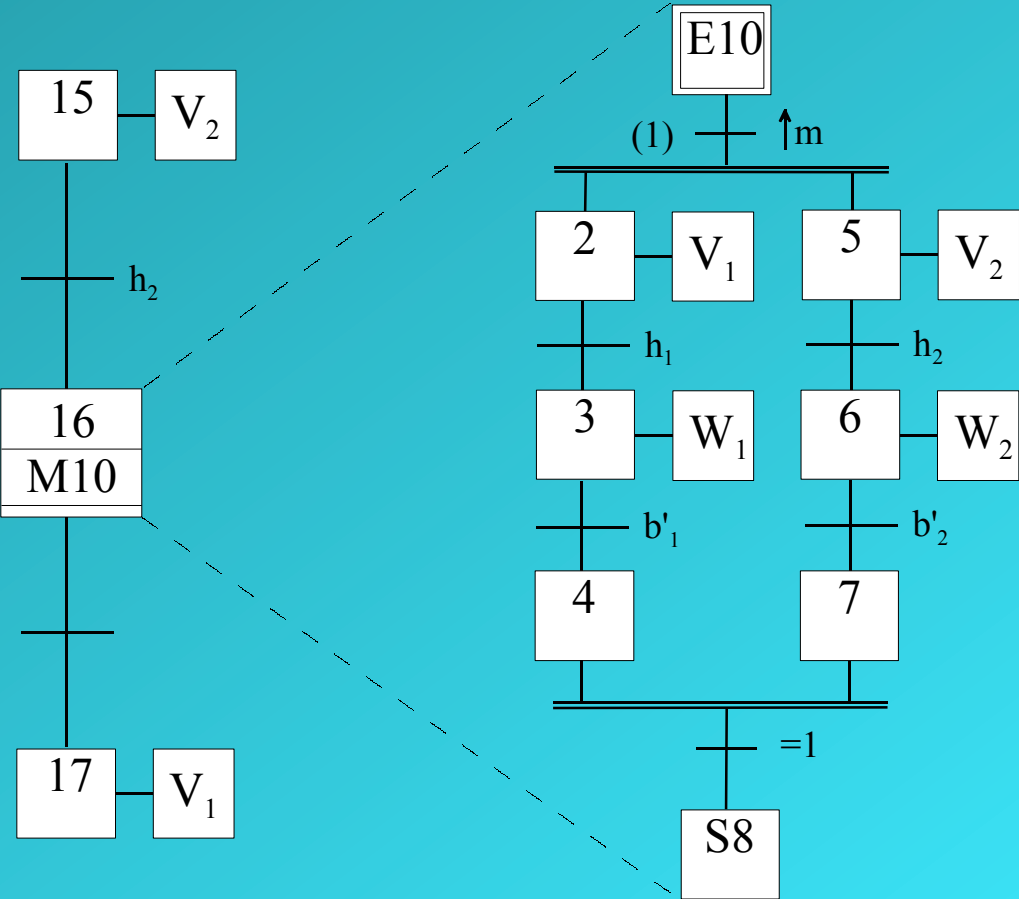
em geral, se os eventos a e b são independentes

$$\uparrow a . \uparrow b = 0$$

# A norma GRAFCET (1975)

*Sequential Function Chart I.E.C. 848 Standard* (1983 e 1988)

**Macro-etapas**



## A norma GRAFCET (1975)

*Sequential Function Chart I.E.C. 848 Standard* (1983 e 1988)

### **Outros mecanismos auxiliares**

### **Pseudo Macro-etapas**

### **Macro Acções**

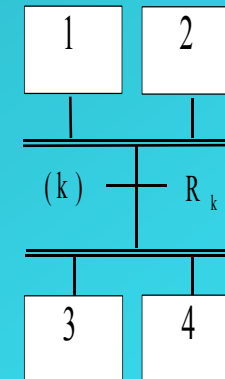
- **Forçar acções**
- **Trancar acções**
- **Mascarar acções**

# A norma GRAFCET (1975)

## Implementação de *GRAFCET* em DOLOG80

O estado de cada etapa é armazenado numa memória auxiliar.

<b>No arranque fazer:</b>	Por resultado de $R_k$	
AM128	em M100	
SLM <sub>x</sub>		
...	AM1	AM3
AM128	AM2	AM4
SLM <sub>y</sub>	AM100	RLM1
(etapas iniciais)	SLM3	AM3
RLM128	AM1	AM4
	AM2	RLM2
	AM100	
	SLM4	





# Automação Industrial baseada em PLCs

9ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 9ª Aula

### Cap. 5 - Sistemas de Eventos Discretos

- Modelação de sistemas de eventos discretos.
- Autómatos.
- Redes de Petri.

### Cap. 6 - Análise de Sistemas de Eventos Discretos

- Propriedades de SEDs.
- Metodologias para análise de SEDs.

### Cap. 7 - SEDs e Automação Industrial

- Relação GRAFCET/Redes de Petri.
- Análise das soluções de automação industrial por analogia com os sistemas de eventos discretos.

# Sistemas de Eventos Discretos

Caracterização genérica de um sistema como as relações de entrada/saída.

Espaço de estados:

$$\dot{x}(t) = f(x(t), u(t))$$

$$y(t) = g(x(t), u(t))$$

em tempo contínuo (ou em tempo discreto)

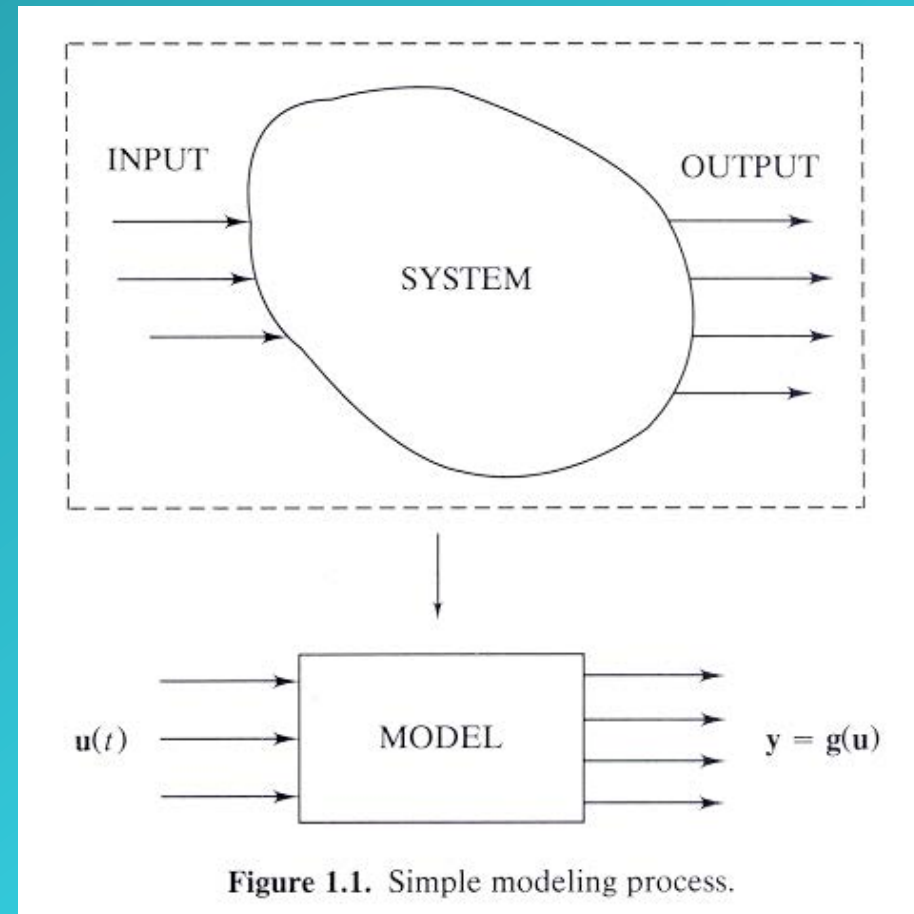
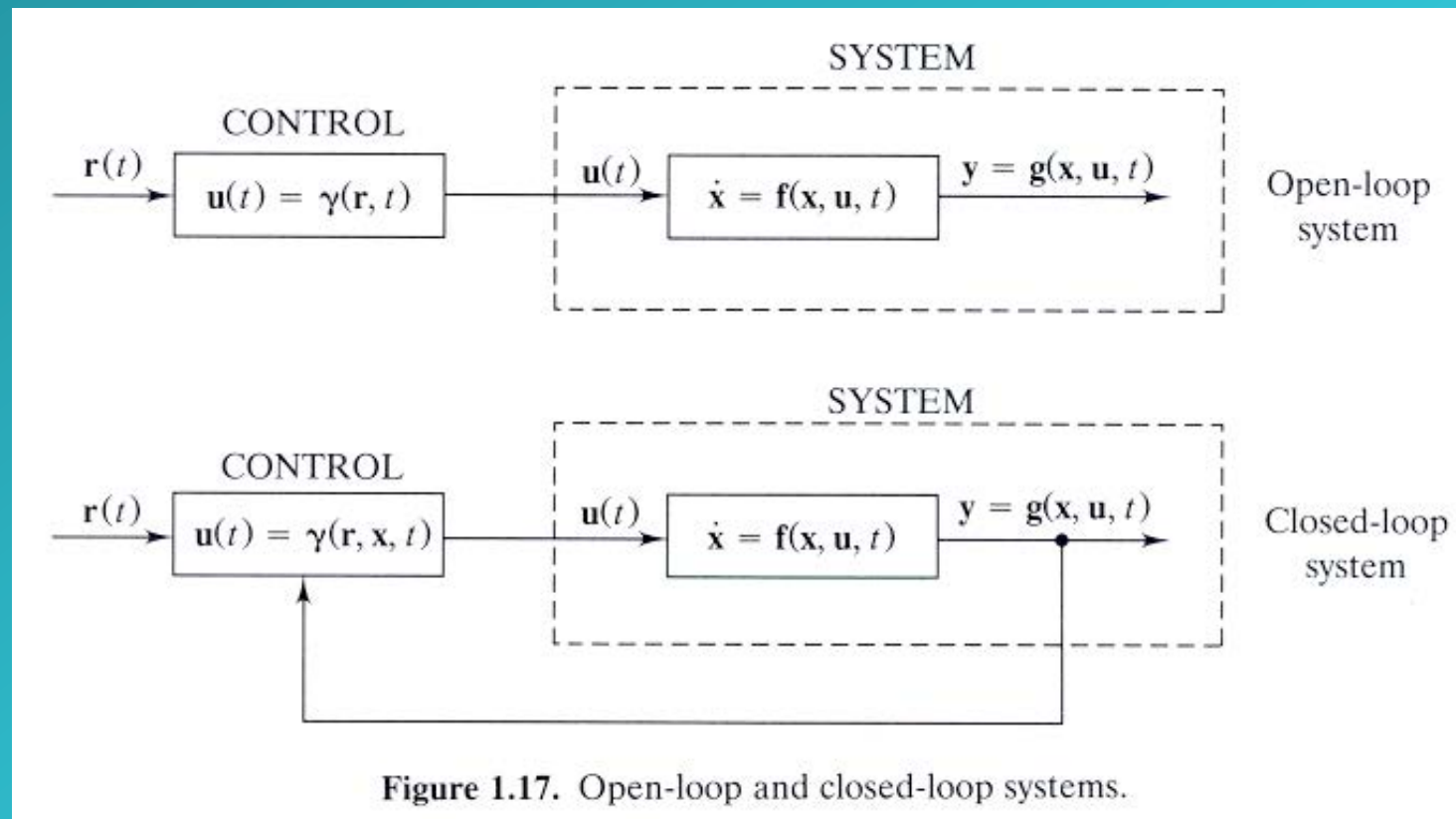


Figure 1.1. Simple modeling process.

# Sistemas de Eventos Discretos

Cadeia aberta vs cadeia fechada com *feedback*





# Sistemas de Eventos Discretos

Exemplo de cadeia fechada com *feedback*

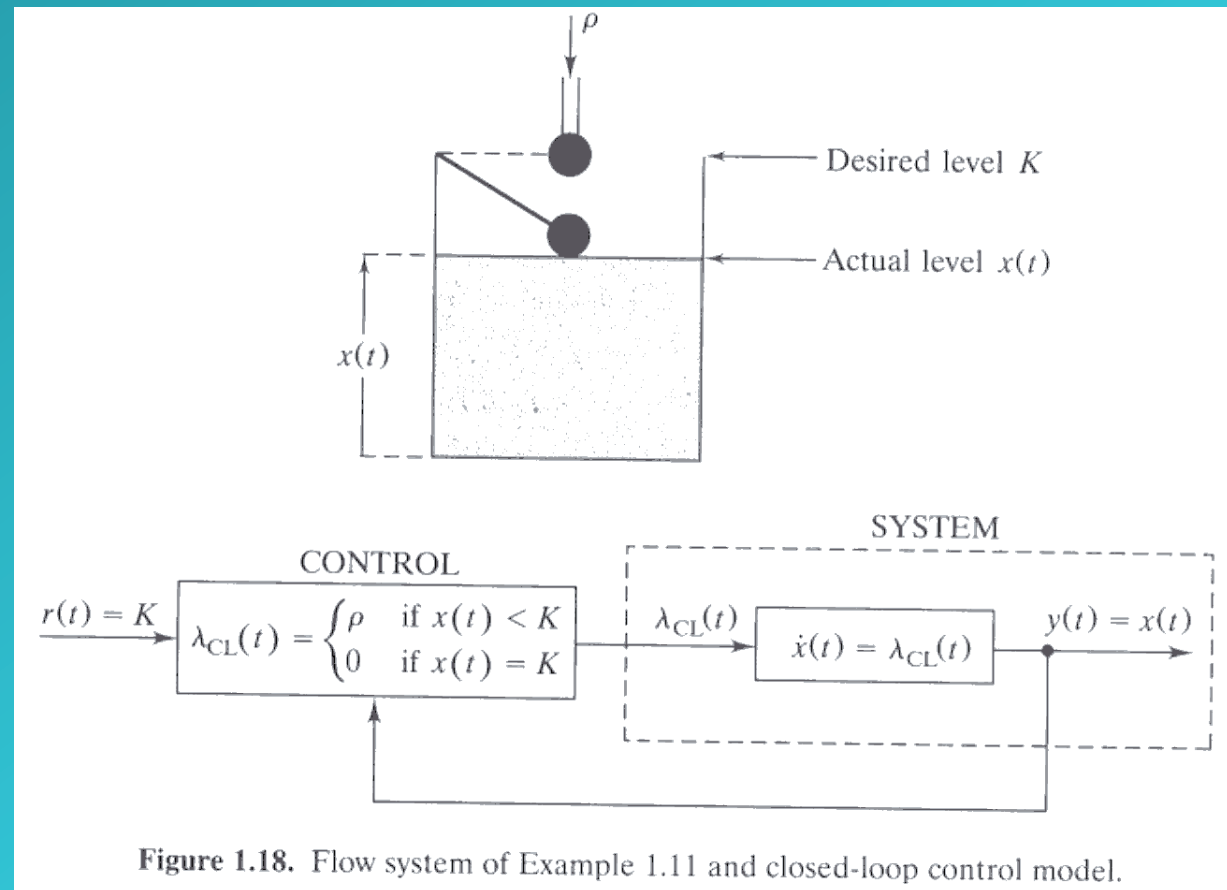


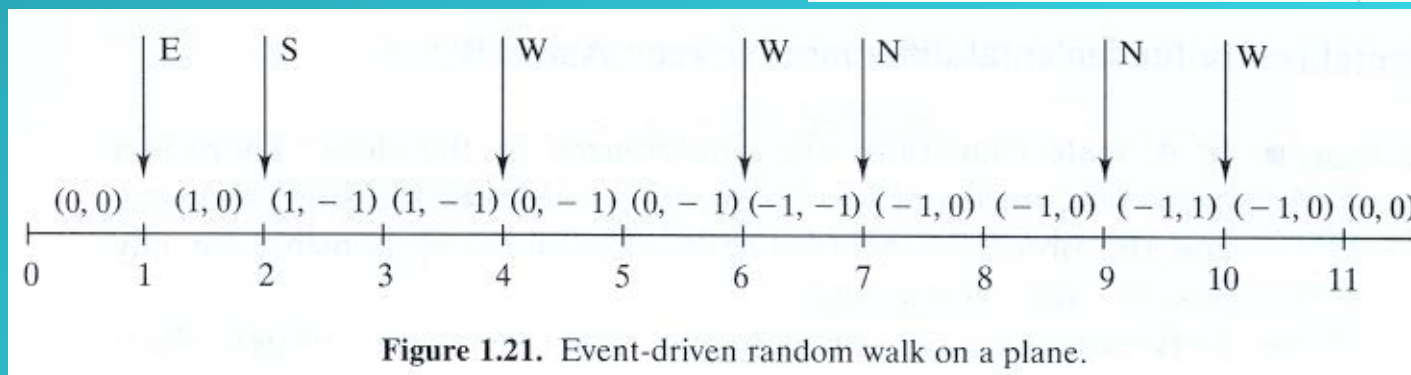
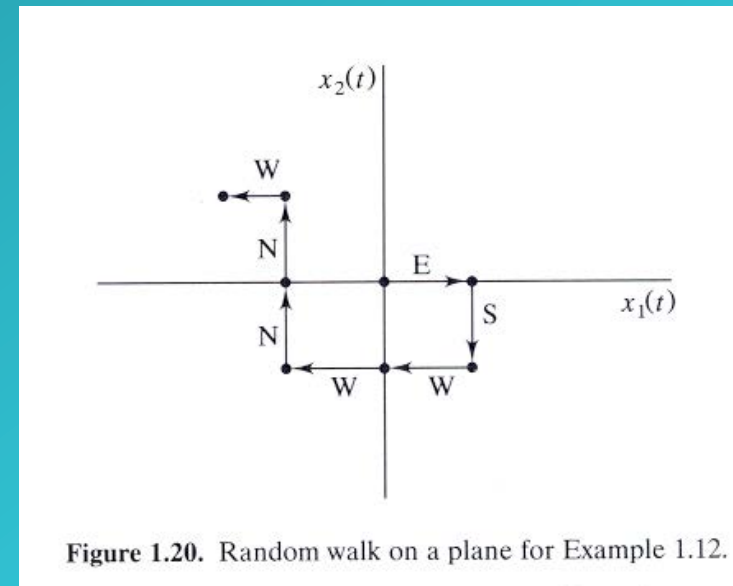
Figure 1.18. Flow system of Example 1.11 and closed-loop control model.

# Sistemas de Eventos Discretos

Sistemas de Eventos Discretos

Conjunto de eventos:

$$\mathbf{E} = \{N, S, E, W\}$$



## Sistemas de Eventos Discretos

Características dos sistemas com variáveis contínuas

1. Espaço de estados é contínuo
2. O mecanismo de transição de estado é *time-driven*

Características dos sistemas de Eventos Discretos

1. Espaço de estados é discreto
2. O mecanismo de transição de estado é *event-driven*

# Taxonomia dos Sistemas

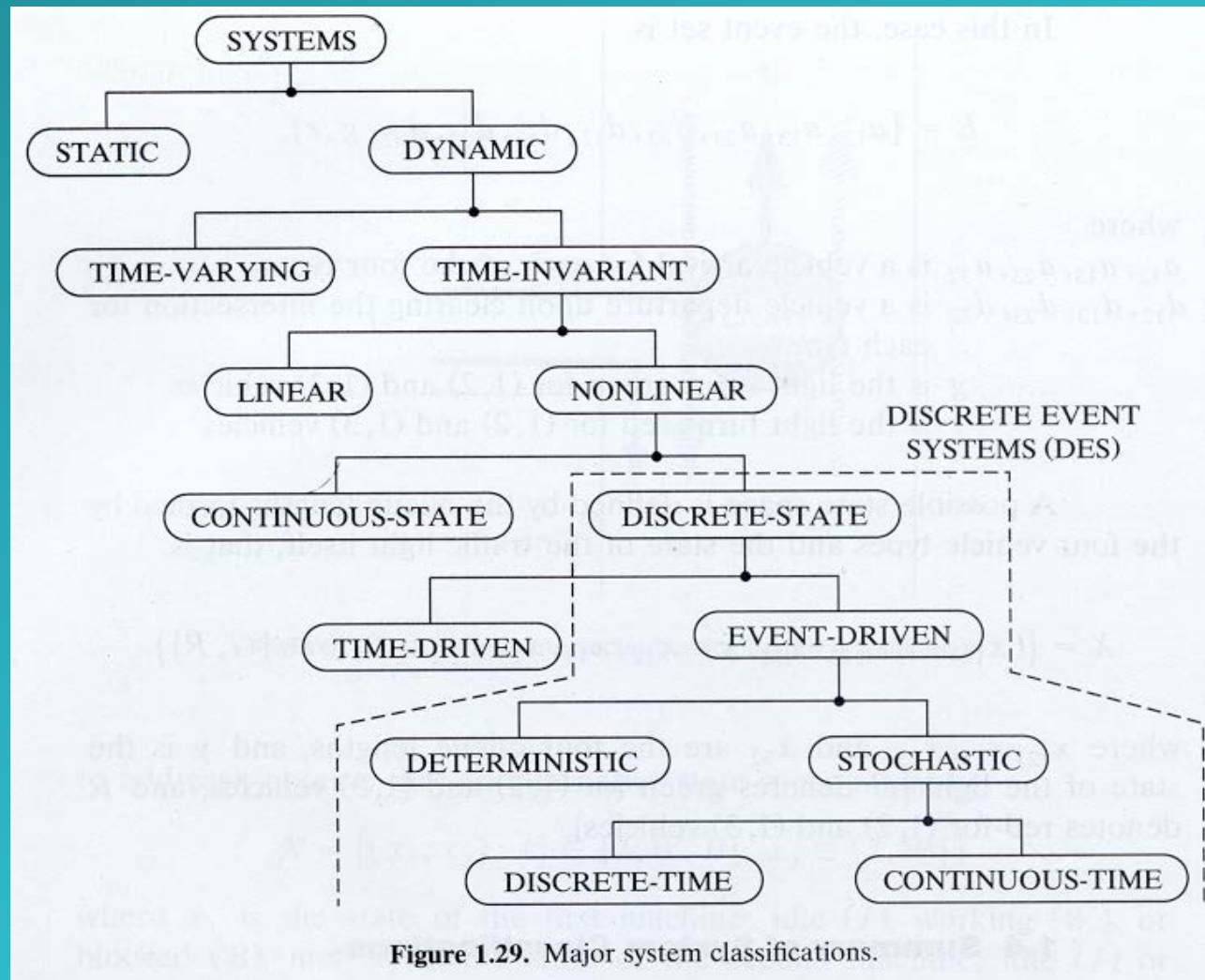


Figure 1.29. Major system classifications.



# Objectivos da Teoria de Sistemas

- Modelação e Análise
- *Design* e síntese
- Controlo
- Avaliação de desempenho
- Optimização

## Áreas de aplicação de Sistemas de Eventos Discretos

- Sistemas Operativos e Computadores
- Redes de Telecomunicações
- Automação
- Sistemas de filas de espera

## Sistemas de Eventos Discretos Não Temporizados

Metodologias típicas de modelação: autómatos e Redes de Petri

### Teoria de Autómatos e Linguagens

**Definição:** Uma linguagem  $L$ , definida sobre um alfabeto  $E$  é um conjunto de *strings* de eventos de  $E$ .

Exemplo:

$$E = \{\alpha, \beta, \gamma\}$$

$$L_1 = \{\alpha\alpha, \alpha\beta, \gamma\beta\alpha\}$$

$$L_2 = \{\text{todas as strings de comprimento 3}\}$$

# Sistemas de Eventos Discretos Não Temporizados

## Teoria de Autómatos e Linguagens

**Definição:** Um autômato (com estado) finito é um *5-tuplo*

$$(E, X, f, x_0, F)$$

onde:

**E** - alfabeto finito

**X** - conjunto finito de estados

**f** - função de transição de estado      **f: X x E -> X**

**x<sub>0</sub>** - estado inicial      **x<sub>0</sub> ∈ X**

**F** - conjunto de estados finais      **F ⊆ E**

# Sistemas de Eventos Discretos Não Temporizados

## Teoria de Autómatos e Linguagens

Exemplo de um autómato

$$(E, X, f, x_0, F)$$

$$E = \{\alpha, \beta, \gamma\}$$

$$X = \{x, y, z\}$$

$$x_0 = x$$

$$F = \{x, z\}$$

$$f(x, \alpha) = x$$

$$f(y, \alpha) = x$$

$$f(z, \alpha) = y$$

$$f(x, \beta) = z$$

$$f(y, \beta) = y$$

$$f(z, \beta) = z$$

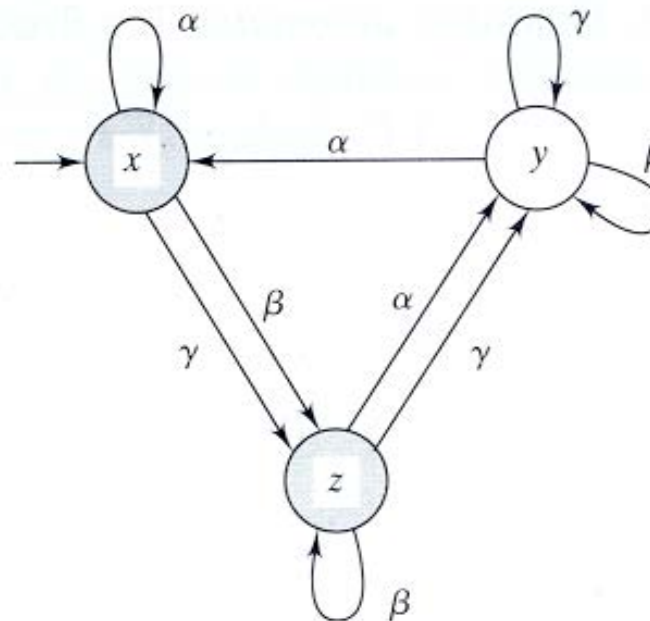


Figure 2.1. State transition diagram for Example 2.3.



# Sistemas de Eventos Discretos Não Temporizados

## Teoria de Autómatos e Linguagens

Exemplo de um autómato estocástico

$$(E, X, f, x_0, F)$$

$$E = \{\alpha, \beta\}$$

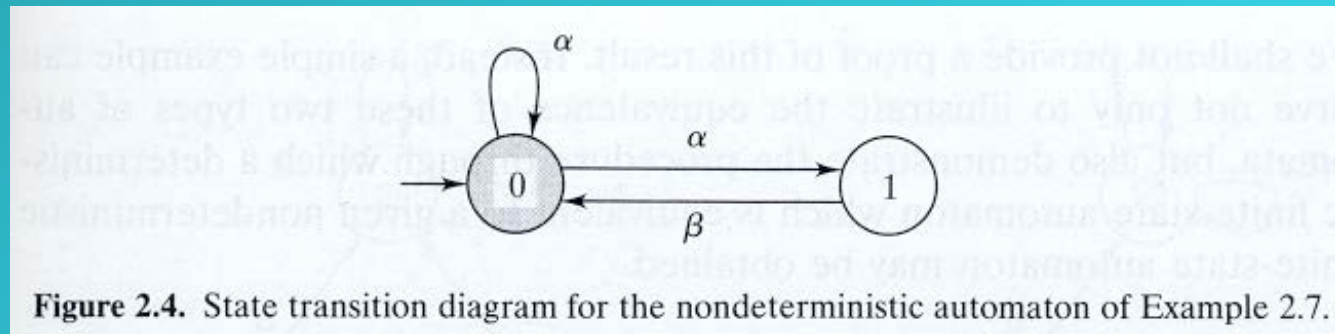
$$X = \{0, 1\}$$

$$x_0 = 0$$

$$F = \{0\}$$

$$f(0, \alpha) = \{0, 1\} \quad f(0, \beta) = \{\}$$

$$f(1, \alpha) = \{\} \quad f(1, \beta) = 0$$



# Sistemas de Eventos Discretos Não Temporizados

## Teoria de Autómatos e Linguagens

Exemplo de um autómato para validar uma sequência de eventos

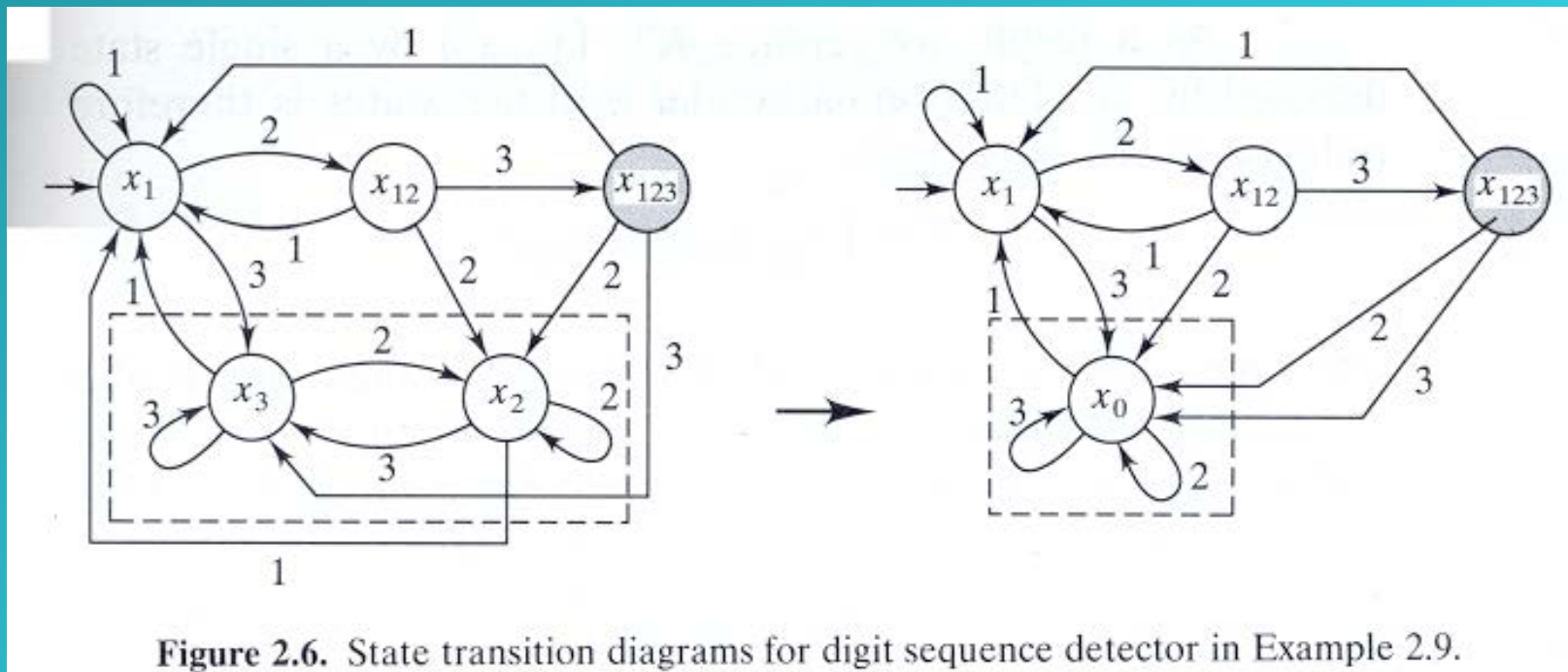


Figure 2.6. State transition diagrams for digit sequence detector in Example 2.9.

# Sistemas de Eventos Discretos Temporizados

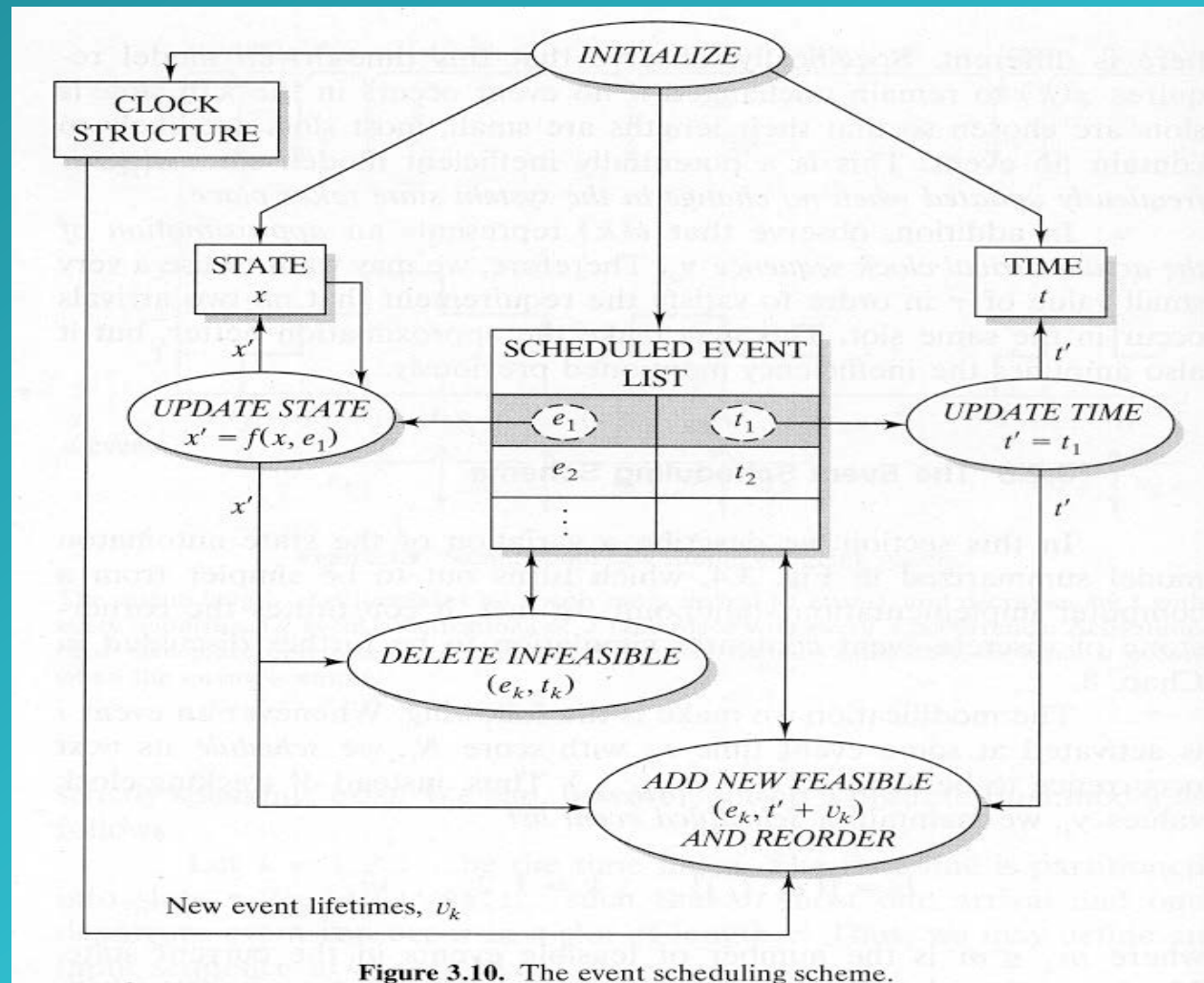


Figure 3.10. The event scheduling scheme.

# Automação Industrial baseada em PLCs

## 10ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)



## 10ª Aula

### **Cap. 5 - Sistemas de Eventos Discretos**

- Modelação de sistemas de eventos discretos.
- Autómatos.
- **Redes de Petri.**

### **Cap. 6 - Análise de Sistemas de Eventos Discretos**

- Propriedades de SEDs.
- Metodologias para análise de SEDs.

### **Cap. 7 - SEDs e Automação Industrial**

- Relação GRAFCET/Redes de Petri.
- Análise das soluções de automação industrial por analogia com os sistemas de eventos discretos.

## Sistemas de Eventos Discretos Não Temporizados

### Redes de Petri - Carl Adam Petri (início da década de 60)

**Definição:** Uma rede de Petri marcada é um *5-tuplo*

$$(\mathbf{P}, \mathbf{T}, \mathbf{A}, \mathbf{w}, \mathbf{x}_0)$$

onde:

$\mathbf{P}$  - conjunto de lugares

$\mathbf{T}$  - conjunto de transições

$\mathbf{A}$  - conjunto de arcos  $\mathbf{A} \subset (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$

$\mathbf{w}$  - função de peso  $\mathbf{w}: \mathbf{A} \rightarrow \mathbf{N}$

$\mathbf{x}_0$  - marcação inicial  $\mathbf{x}_0: \mathbf{P} \rightarrow \mathbf{N}$

# Sistemas de Eventos Discretos Não Temporizados

## Redes de Petri

Regras a observar:

- Um arco pode ligar lugares a transições;
- Um arco pode ligar transições a lugares.
- Um lugar pode não ter nenhuma transição de entrada (fonte);
- Uma lugar pode não ter nenhuma transição de saída (sorvedouro).
- O mesmo se passa para as transições.

# Sistemas de Eventos Discretos Não Temporizados

## Exemplo de Rede de Petri e sua Representação Gráfica

$$(P, T, A, w, x_0)$$

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

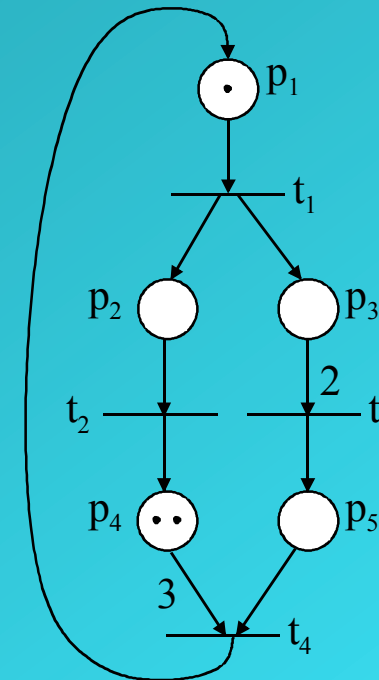
$$A = \{(p_1, t_1), (t_1, p_2), (t_1, p_3), (p_2, t_2), (p_3, t_3), (t_2, p_4), (t_3, p_5), (p_4, t_4), (p_5, t_4), (t_4, p_1)\}$$

$$w(p_1, t_1)=1, w(t_1, p_2)=1, w(t_1, p_3)=1, w(p_2, t_2)=1$$

$$w(p_3, t_3)=2, w(t_2, p_4)=1, w(t_3, p_5)=1, w(p_4, t_4)=3$$

$$w(p_5, t_4)=1, w(t_4, p_1)=1$$

$$x_0 = \{1, 0, 0, 2, 0\}$$





## Sistemas de Eventos Discretos Não Temporizados

### Definição alternativa de uma rede de Petri

Uma rede de Petri marcada é um *5-tuplo*

$$(P, T, I, O, \mu_0)$$

onde:

**P** - conjunto de lugares

**T** - conjunto de transições

**I** - função de entrada das transições  $I: P \rightarrow T^\infty$

**O** - função de saída das transições  $O: T \rightarrow P^\infty$

$\mu_0$  - marcação inicial  $\mu_0: P \rightarrow \mathbf{N}$

# Sistemas de Eventos Discretos Não Temporizados

## Exemplo de Rede de Petri e sua Representação Gráfica

Definição alternativa

$$(P, T, I, O, \mu_0)$$

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2, t_3, t_4\}$$

$$I(t_1) = \{p_1\}$$

$$I(t_2) = \{p_2\}$$

$$I(t_3) = \{p_3, p_3\}$$

$$I(t_4) = \{p_4, p_4, p_4, p_5\}$$

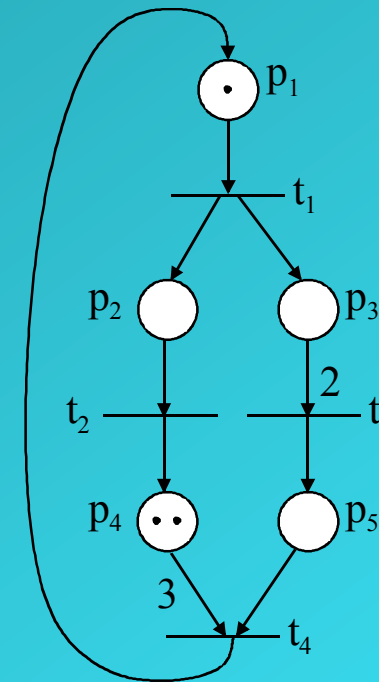
$$O(t_1) = \{p_2, p_3\}$$

$$O(t_2) = \{p_4\}$$

$$O(t_3) = \{p_5\}$$

$$O(t_4) = \{p_1\}$$

$$\mu_0 = \{1, 0, 0, 2, 0\}$$



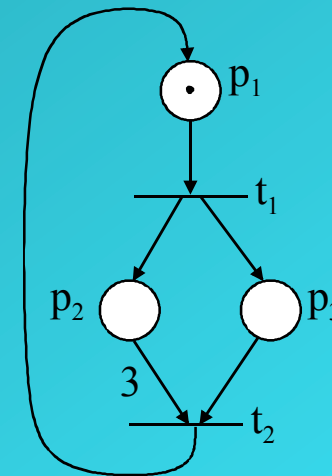
# Sistemas de Eventos Discretos Não Temporizados

## Redes de Petri

O conjunto de marcações possíveis de uma Rede de Petri constitui o **espaço de estados**.

**Questão:**

Como evolui o estado de uma rede de Petri?



## Sistemas de Eventos Discretos Não Temporizados

### Redes de Petri

Uma transição  $t_j \in T$  está permitida (*enabled*) se:

$$\forall p_i \in P: \mu(p_i) \geq \#(p_i, I(t_j))$$

Uma transição  $t_j \in T$  permitida pode disparar (*fire*), resultando uma nova marcação:

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$$



# Sistemas de Eventos Discretos Não Temporizados

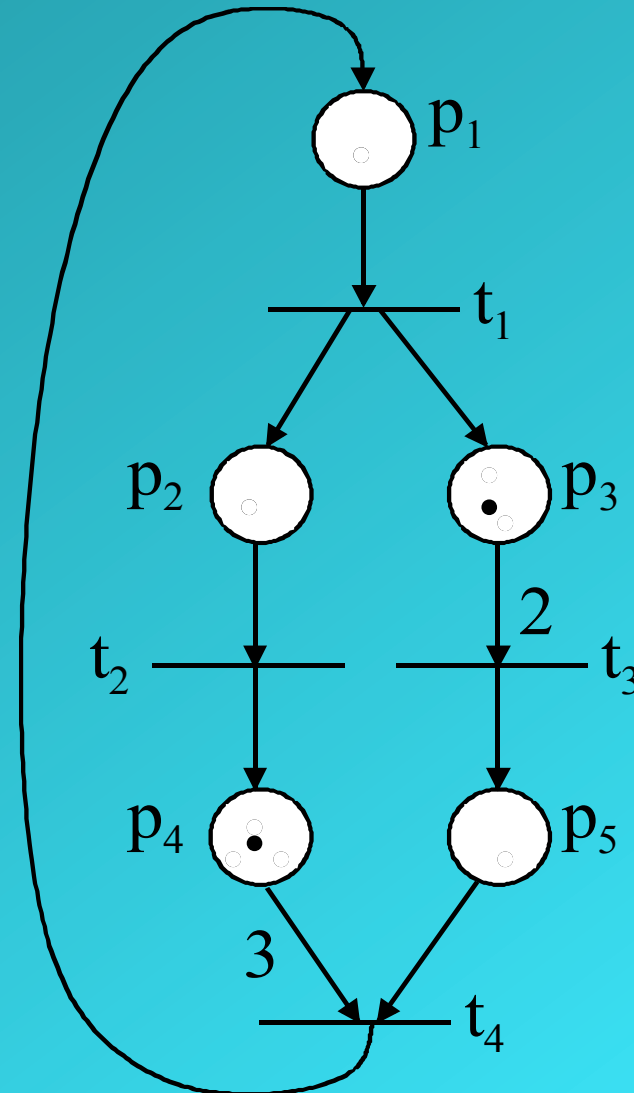
## Redes de Petri

Exemplo de funcionamento de uma rede de Petri

Marcação inicial:

$$\mu_0 = \{1, 0, 1, 2, 0\}$$

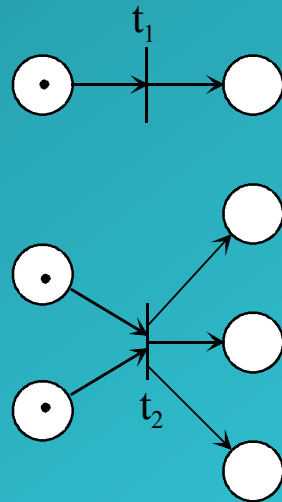
Este sistema de eventos discretos encontra-se bloqueado.  
Tem *deadlocks*!



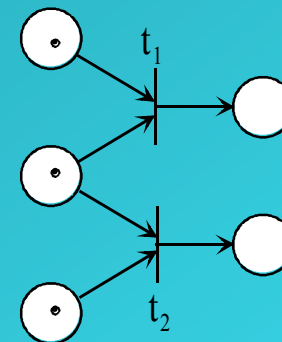
# Sistemas de Eventos Discretos Não Temporizados

## Redes de Petri

Concorrência



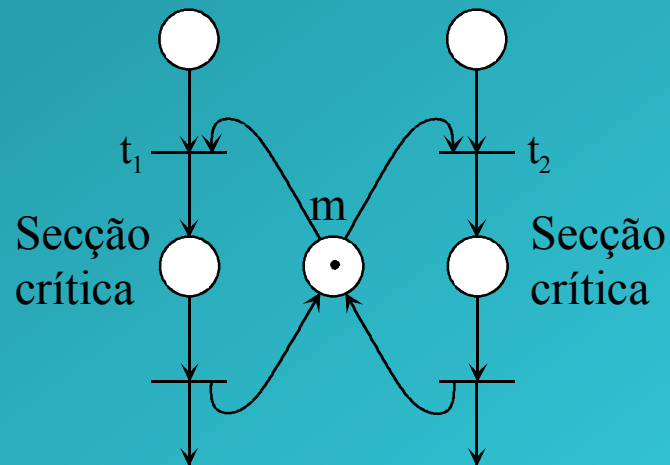
Conflito



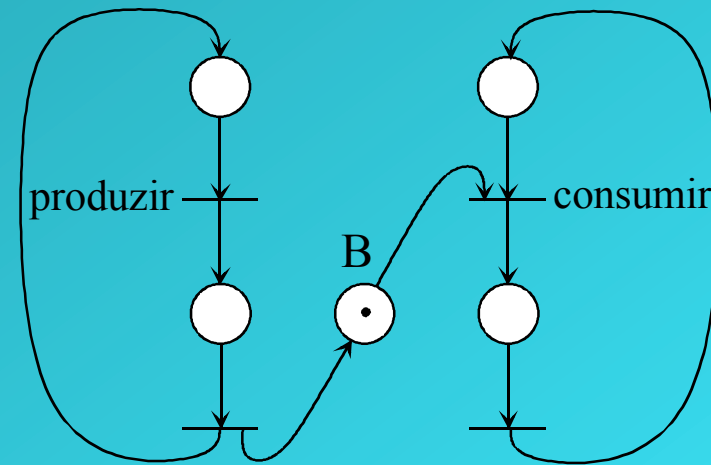
# Sistemas de Eventos Discretos Não Temporizados

## Redes de Petri

Exclusão Mútua



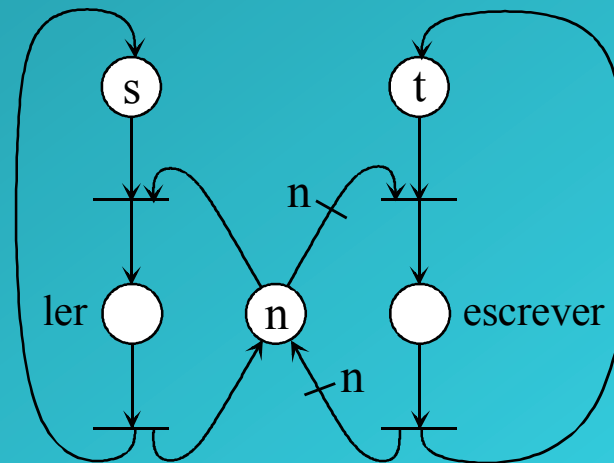
Produtor/Consumidor



# Sistemas de Eventos Discretos Não Temporizados

## Redes de Petri

Leitores/Escritores



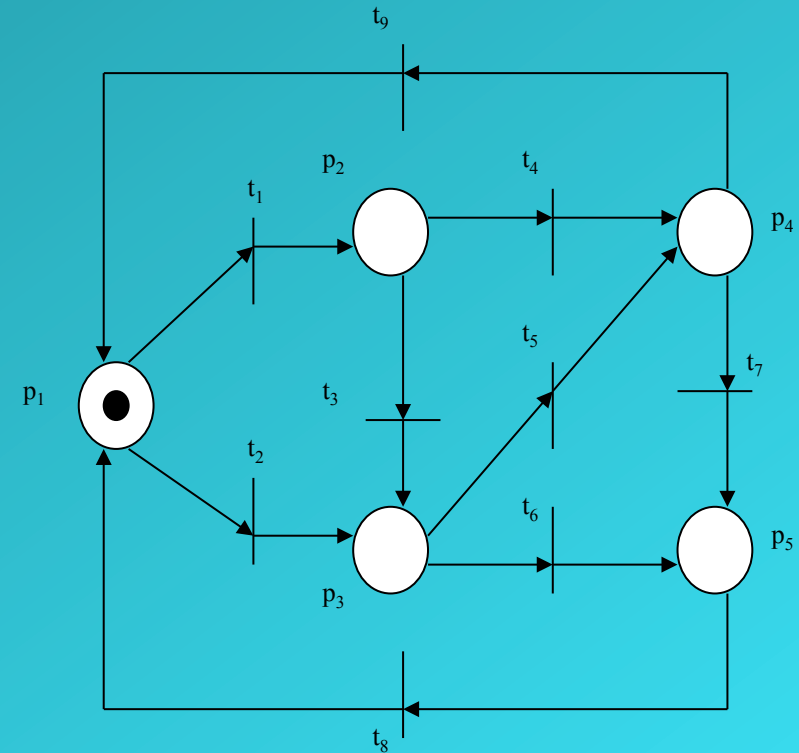


## Sistemas de Eventos Discretos Não Temporizados

### Exemplo de um sistema modelado com Redes de Petri

Uma máquina automática de venda de refrigerantes aceita apenas moedas de 50\$00 e 100\$00 e vende 2 tipos de refrigerantes: refrigerante A, que custa 150\$00 e refrigerante B, que custa 200\$00.

Assuma que na modelação da máquina são omitidas as transições relativas à devolução de moedas.

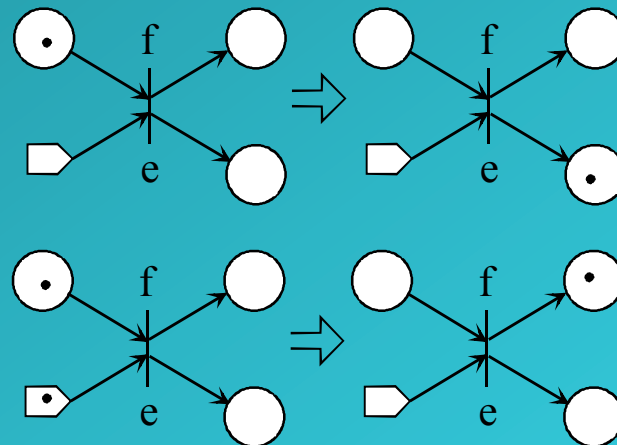


$p_1$ : máquina com zero escudos  
 $t_1$ : introdução de moeda de 50 escudos;  
 $t_8$ : saída de refrigerante do tipo B.

# Sistemas de Eventos Discretos Não Temporizados

## Extensões às Redes de Petri

### Interruptores (*Switches*) [Baer 1973]

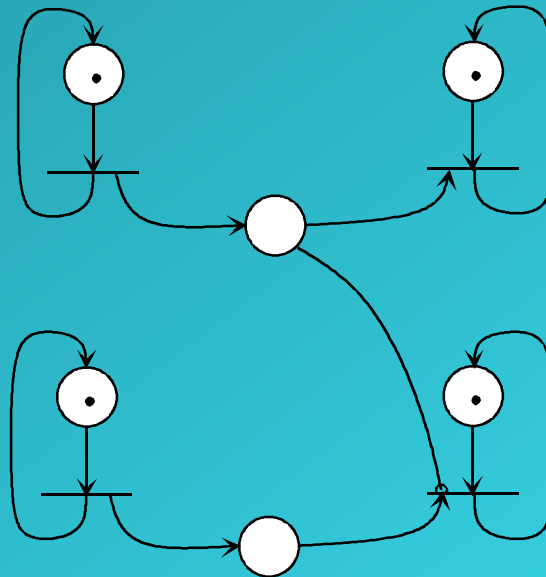


É possível implementar com redes normais.

# Sistemas de Eventos Discretos Não Temporizados

## Extensões às Redes de Petri

### Arcos Inibidores



É possível implementar com redes normais.

# Sistemas de Eventos Discretos Não Temporizados

## Subclasses das Redes de Petri

### **Máquinas de Estados:**

São redes de Petri onde cada lugar tem exactamente 1 arco de entrada e um arco de saída.

### **Grafos Marcados**

São redes de Petri onde um lugar é entrada para uma transição e saída de uma transição.



## Sistemas de Eventos Discretos Não Temporizados

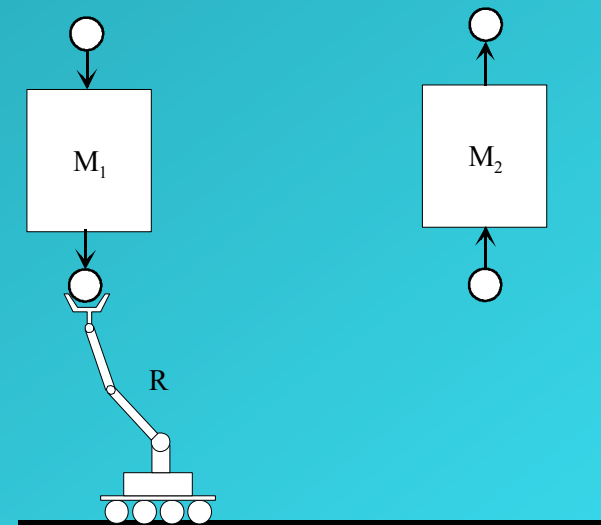
### Exemplo de SED:

Sistema de manufactura composto por 2 máquinas ( $M_1$  e  $M_2$ ) e um manipulador (R) que retira uma peça acabada de maquinar em  $M_1$  e a transporta para  $M_2$ .

Não há buffers em nenhuma máquina, pelo que se uma peça chegar a  $M_1$  e esta estiver ocupada a peça será rejeitada.

Se o manipulador chegar a  $M_2$  e esta estiver a processar uma peça, o manipulador tem de esperar.

Tempos de maquinação:  $M_1=0.5s$ ;  $M_2=1.5s$ ;  $R_{M_1 \rightarrow M_2}=0.2s$ ;  $R_{M_2 \rightarrow M_1}=0.1s$ ;



## Sistemas de Eventos Discretos Não Temporizados

### Exemplo de SED (cont.):

Estados de

$M_1$	$x_1$
$M_2$	$x_2$
R	$x_3$

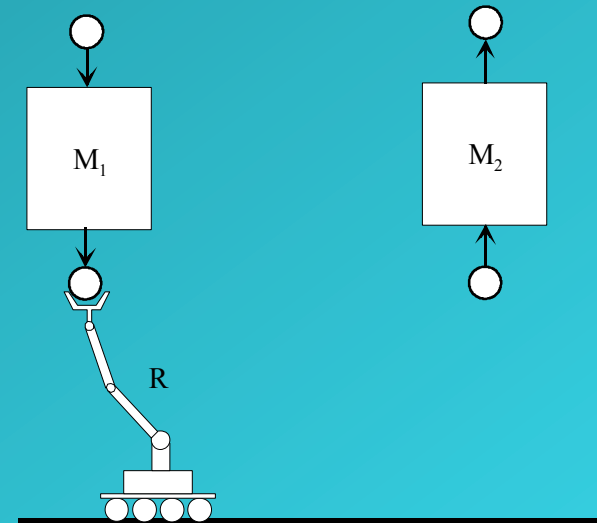
Chegada de peças:

$$a(t) = \begin{cases} 1 & \text{em } \{0.1, 0.7, 1.1, 1.6, 2.5\} \\ 0 & \text{em todos os outros instantes} \end{cases}$$

$x_1 = \{\text{Idle, Busy, Waiting}\}$

$x_2 = \{\text{Idle, Busy}\}$

$x_3 = \{\text{Idle, Carrying, Returning}\}$

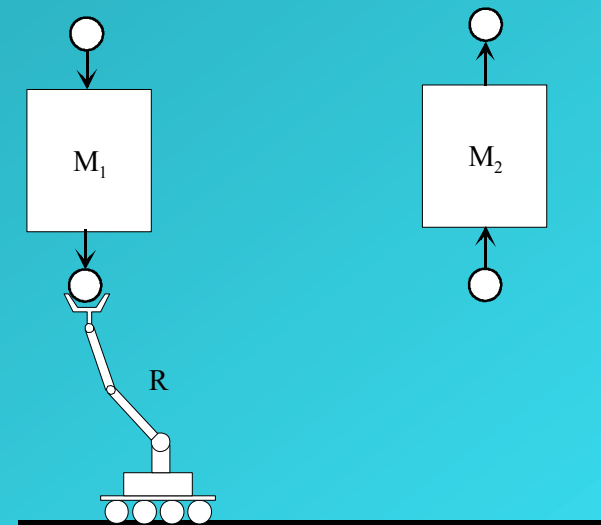


## Sistemas de Eventos Discretos Não Temporizados

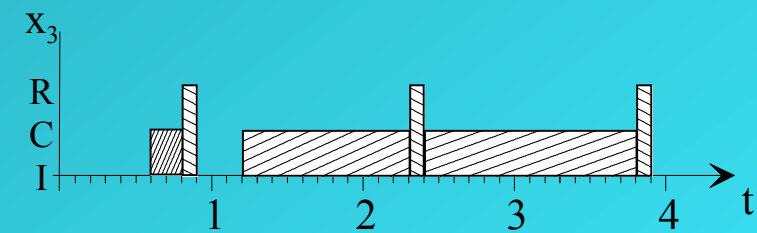
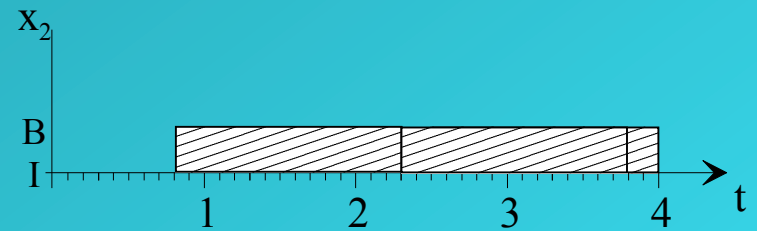
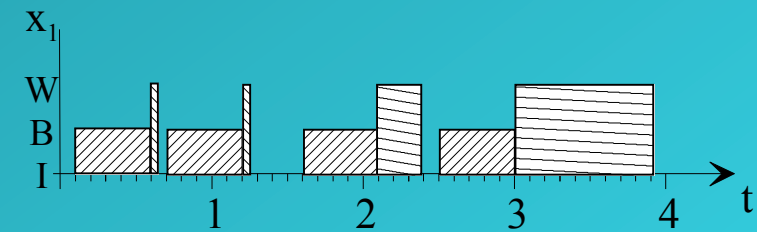
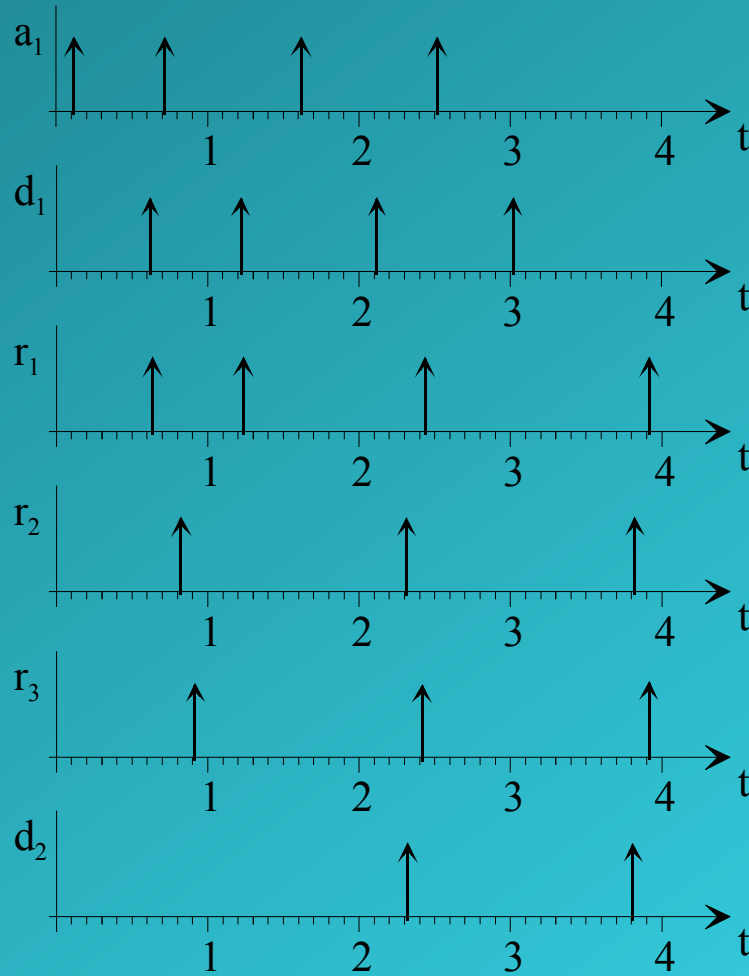
### Exemplo de SED (cont.):

Definição de eventos:

- $a_1$  - carregar peça em  $M_1$
- $d_1$  - fim de processamento em  $M_1$
- $r_1$  - carregar manipulador
- $r_2$  - descarregar manipulador e carregar  $M_2$
- $d_2$  - fim de processamento em  $M_2$
- $r_3$  - manipulador na base



# Sistemas de Eventos Discretos Não Temporizados



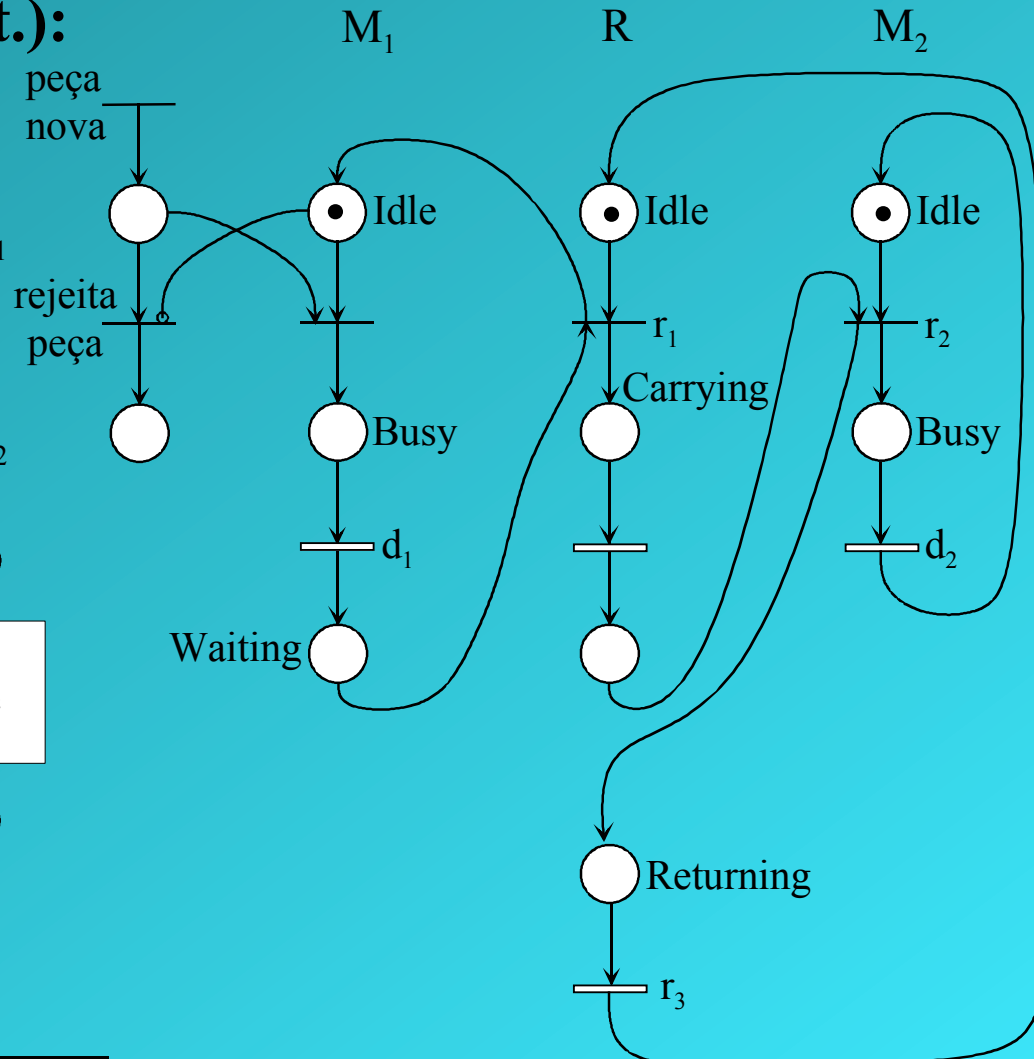
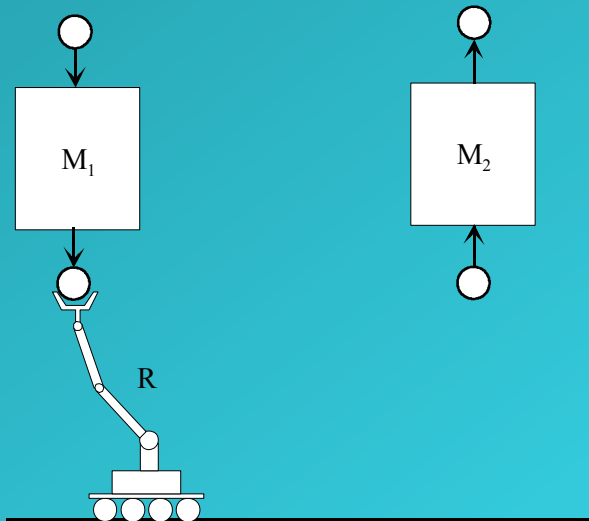


# Sistemas de Eventos Discretos Não Temporizados

## Exemplo de SED (cont.):

Eventos

- $a_1$ - carregar peça em  $M_1$
- $d_1$ - fim de processamento em  $M_1$
- $r_1$ - carregar manipulador e carregar  $M_2$
- $r_2$ - descarregar manipulador e carregar  $M_2$
- $d_2$ - fim de processamento em  $M_2$
- $r_3$ - manipulador na base



# Automação Industrial baseada em PLCs

## 11ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 11<sup>a</sup> Aula

### Cap. 5 - Sistemas de Eventos Discretos

- Modelação de sistemas de eventos discretos.
- Autómatos.
- Redes de Petri.

### Cap. 6 - Análise de Sistemas de Eventos Discretos

- Propriedades de SEDs.
- Metodologias para análise de SEDs.

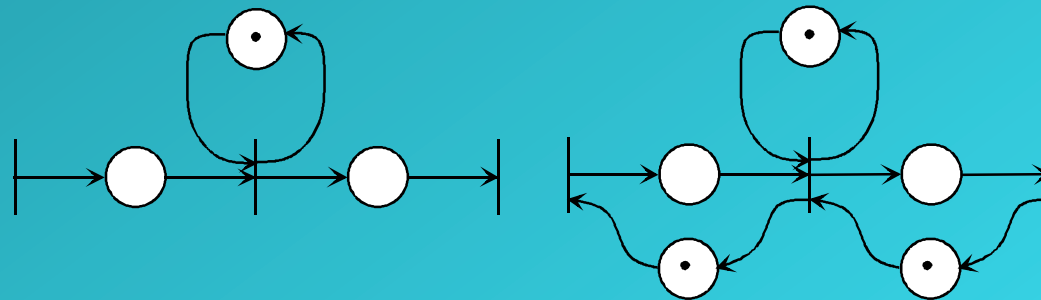
### Cap. 7 - SEDs e Automação Industrial

- Relação GRAFCET/Redes de Petri.
- Análise das soluções de automação industrial por analogia com os sistemas de eventos discretos.

## Propriedades dos Sistemas de Eventos Discretos

### Segurança

Um lugar  $p_i \in P$  de uma rede de Petri  $C=(P, T, I, O, \mu_0)$  é seguro se para todo o  $\mu' \in R(C, \mu_0)$ :  $\mu' \leq 1$ . Uma rede de Petri é segura se cada todos os seus lugares forem seguros.



Rede não segura

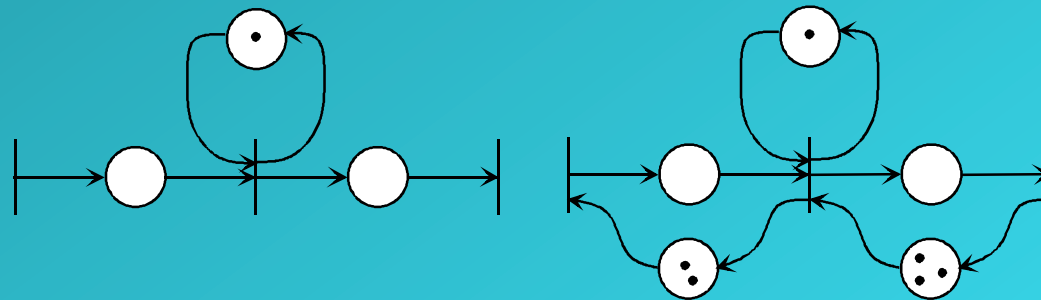
Rede segura



## Propriedades dos Sistemas de Eventos Discretos

### Limitação

Um lugar  $p_i \in P$  de uma rede de Petri  $C=(P, T, I, O, \mu_0)$  é  $k$ -limitado se para todo o  $\mu' \in R(C, \mu_0)$ :  $\mu' \leq k$ . Uma rede de Petri é  $k$ -limitada se todos os seus lugares forem  $k$ -limitados.



Rede não limitada

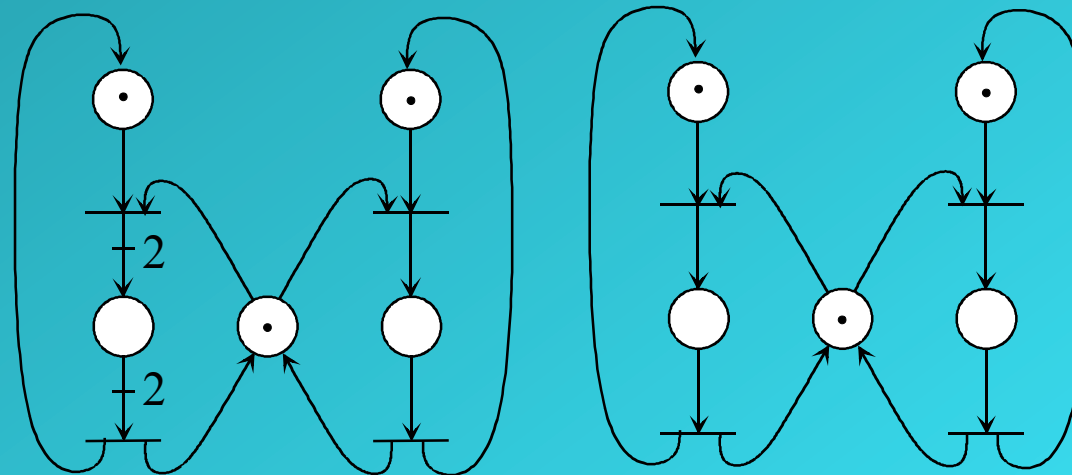
Rede 3-limitada

# Propriedades dos Sistemas de Eventos Discretos

## Conservação

Uma rede de Petri  $C=(P, T, I, O, \mu_0)$  é **estritamente conservativa** se para todo o  $\mu' \in R(C, \mu)$

$$\sum_{p_i \in P} \mu'(p_i) = \sum_{p_i \in P} \mu(p_i)$$



Rede conservativa

Rede não conservativa

## Propriedades dos Sistemas de Eventos Discretos

**Vivacidade das transições:** A transição  $t_j$  é viva de

**Nível 0** - se nunca pode ser disparada.

**Nível 1** - se é potencialmente disparável, isto é se existir  $\mu' \in R(C, \mu)$  tal que  $t_j$  está permitida em  $\mu'$ .

**Nível 2** - se, para cada inteiro  $n$ , existir uma sequência de disparos na qual ocorrem  $n$  disparos de  $t_j$ .

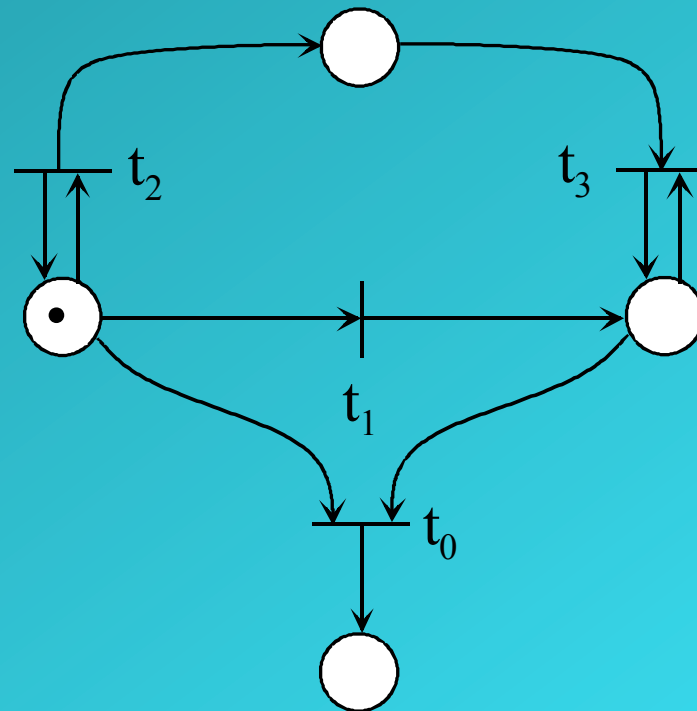
**Nível 3** - se existir uma sequência de disparos na qual ocorre um número infinito de disparos de  $t_j$ .

**Nível 4** - se para cada  $\mu' \in R(C, \mu)$  existir uma sequência  $\sigma$  que leva a que se torne permitida.

## Propriedades dos Sistemas de Eventos Discretos

### Exemplo de vivacidade das transições

- $t_0$  é de nível 0.
- $t_1$  é de nível 1.
- $t_2$  é de nível 2.
- $t_3$  é de nível 3.





## Propriedades dos Sistemas de Eventos Discretos

### Alcançabilidade ou Atingibilidade

Dada uma rede de Petri  $C=(P, T, I, O, \mu_0)$  com uma marcação  $\mu_0$  será que a marcação  $\mu' \in R(C, \mu)$  é alcançável?

### Métodos de análise:

- Por tentativa
- Árvore de alcançabilidade
- Método das matrizes

## Propriedades dos Sistemas de Eventos Discretos

### Árvore de alcançabilidade

Construir a árvore de marcações atingíveis. Constituída por nós:

- terminais
- interiores
- duplicados

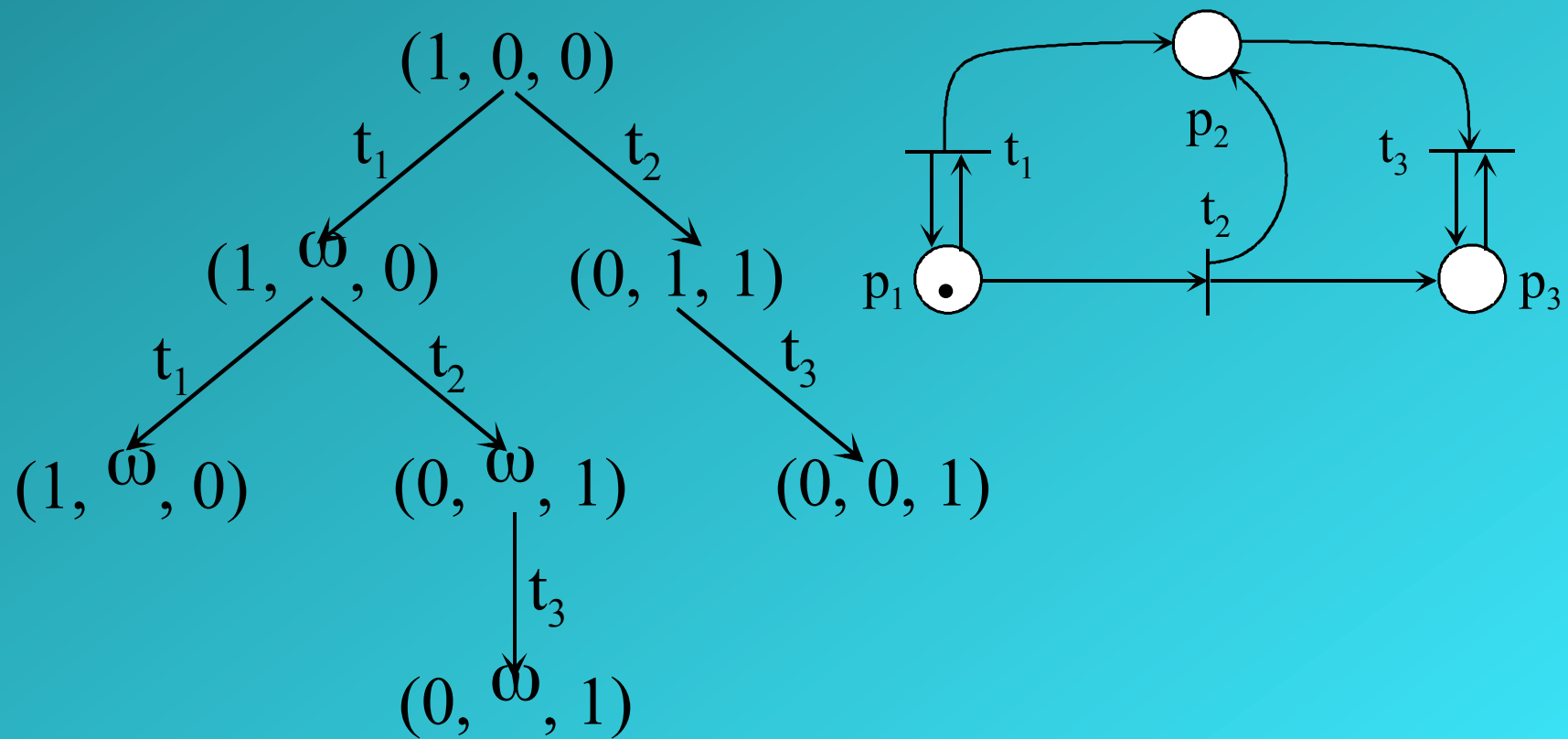
Introduz-se o símbolo infinito ( $\omega$ ) para obter árvores de tamanho finito.

### **Nota:**

Este método também pode ser usado para estudar as propriedades introduzidas anteriormente.

## Propriedades dos Sistemas de Eventos Discretos

Exemplo de construção da árvore de alcançabilidade:



## Propriedades dos Sistemas de Eventos Discretos

### Método das Matrizes (condição suficiente)

A evolução da marcação numa rede de Petri pode ser escrita de forma abreviada como:

$$\mu' = \mu + f(\sigma)D$$

onde:

- $\mu'$  - marcação a atingir
- $\mu$  - marcação inicial
- $f(\sigma)$  - vector de disparo de transições
- $D$  - matriz de transição. Contabiliza a distribuição de tokens

#### **Nota:**

Este método também pode ser usado para estudar a propriedade de conservação de uma rede de Petri.



## Propriedades dos Sistemas de Eventos Discretos

Exemplo de utilização do método das matrizes:

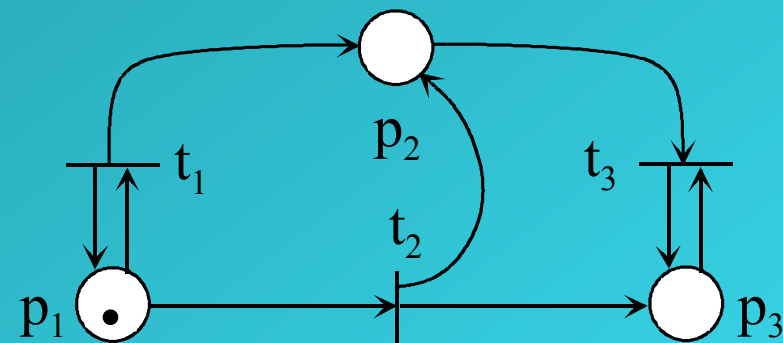
$$\mu' = \mu + f(\sigma)D$$

$$\mu' = (1, 3, 0)$$

$$\mu = (1, 0, 0)$$

$$f(\sigma) = (\sigma_a, \sigma_b, \sigma_c)$$

$$D = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$



$$\begin{cases} 1 = 1 - \sigma_b \\ 3 = \sigma_a + \sigma_b \\ 0 = \sigma_b \end{cases} \quad \begin{cases} \sigma_a = 3 \\ \sigma_b = 0 \\ \sigma_c = ? \end{cases}$$

**Verificar!**

# Automação Industrial baseada em PLCs

## 12ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

## 12ª Aula

### Cap. 5 - Sistemas de Eventos Discretos

- Modelação de sistemas de eventos discretos.
- Autómatos.
- Redes de Petri.

### Cap. 6 - Análise de Sistemas de Eventos Discretos

- Propriedades de SEDs.
- Metodologias para análise de SEDs.

### Cap. 7 - SEDs e Automação Industrial

- Relação GRAFCET/Redes de Petri.
- Análise das soluções de automação industrial por analogia com os sistemas de eventos discretos.

## Implementação de Redes de Petri com PLCs

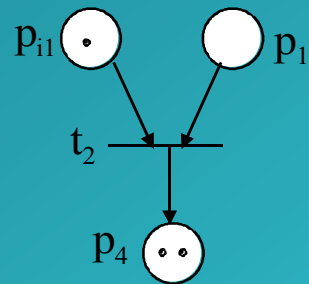
### Interface com o exterior:

- **Entradas de nível** - utilizar em funções associadas aos lugares e/ou transições
- **Entradas como eventos** - associar uma variável onde se fazem aparecer tokens associados à ocorrência dos eventos correspondentes
- **Saídas de nível** - comandar com funções das entradas, saídas ou outras variáveis
- **Eventos na saída** - gerar transições nas saídas correspondentes



## Implementação de Redes de Petri com PLCs

Exemplo de evento na entrada (flanco ascendente):



M21	I1	OM21	ONI1
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	1

M21 anterior valor de entrada I1  
 MW1 variável que representa o lugar associado a I1

...

OM21

ONI1

GIxx

LBV1

ADDMW1

xx

AI1

=M21

...

## Implementação de Redes de Petri com PLCs

Exemplo de geração de evento na saída (flanco ascendente):



M1      variável auxiliar iniciada a zero  
...  
AI1  
ANM1  
=Q1  
AI1  
=M1  
...

## Implementação de Redes de Petri com PLCs

Exemplo de geração de evento na saída (flanco descendente):



M1      variável auxiliar iniciada a zero  
...  
ANI1  
ANM1  
=Q1  
ANI1  
=M1  
...

## Implementação de Redes de Petri com PLCs

O número de tokens em cada lugar é armazenado numa memória auxiliar do tipo word (limitação a 65535 tokens por lugar).

No arranque fazer:

AM128

Gixx

PE

xx

LDV2

=MW4

LDV1

=MW5

...

Testar se  $t_4$  está activa

LDMW4

GTV2

=M1

LDMW5

GTV1

=M2

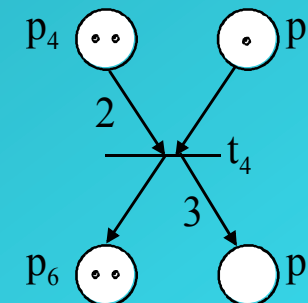
AM1

AM2

Giyy

...

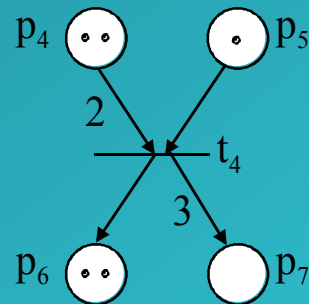
caso contrário continuar





## Implementação de Redes de Petri com PLCs

Se  $t_4$  está activa disparar  
e actualizar os tokens



yy

LDMW4  
SUBV2  
=MW4  
LDMW5  
SUBV1  
=MW5  
LDMW6  
ADDV1  
=MW6  
LDMW7  
ADDV3  
=MW7

**Associado a cada lugar pode existir uma função das entradas para actualizar as saídas (extensão).**

## **Análise de SEDs baseado na implementação de Redes de Petri com PLCs**

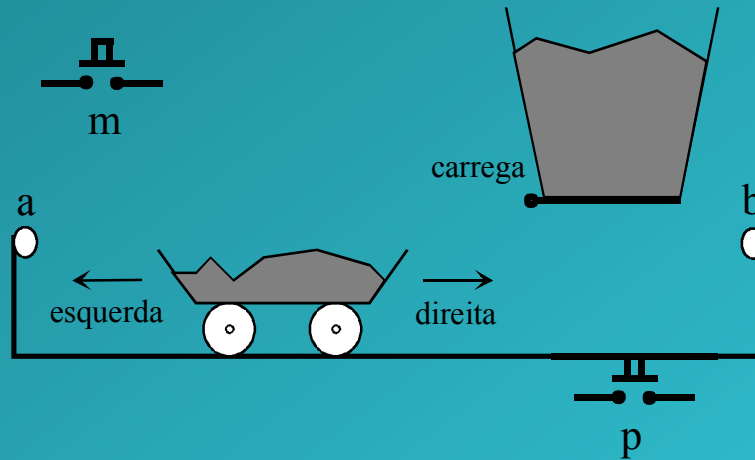
**Estudar as propriedades dos sistema de automação baseado na implementação de SEDs como Redes de Petri.**

- **Vivacidade**
- **Atingibilidade**
- **Segurança**

...

**Nota: no caso de se utilizar GRAFCET ou uma máquina de estados analisar a uma Rede de Petri equivalente.**

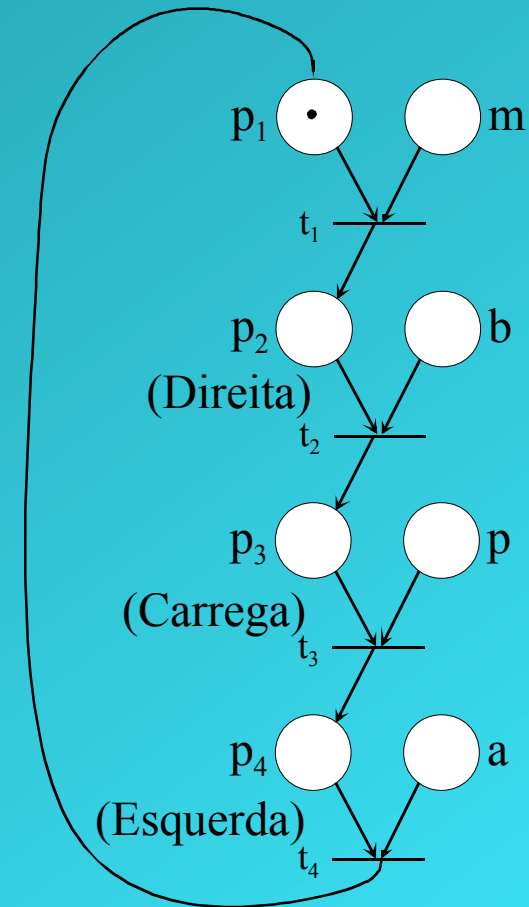
Exemplo de modelação de um sistema de controlo



Associado a um lugar podem ser actuadas saídas

Q1 saída correspondente a comando para a direita

LDMW4  
GE1  
=Q1



# Automação Industrial baseada em PLCs

## 13ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

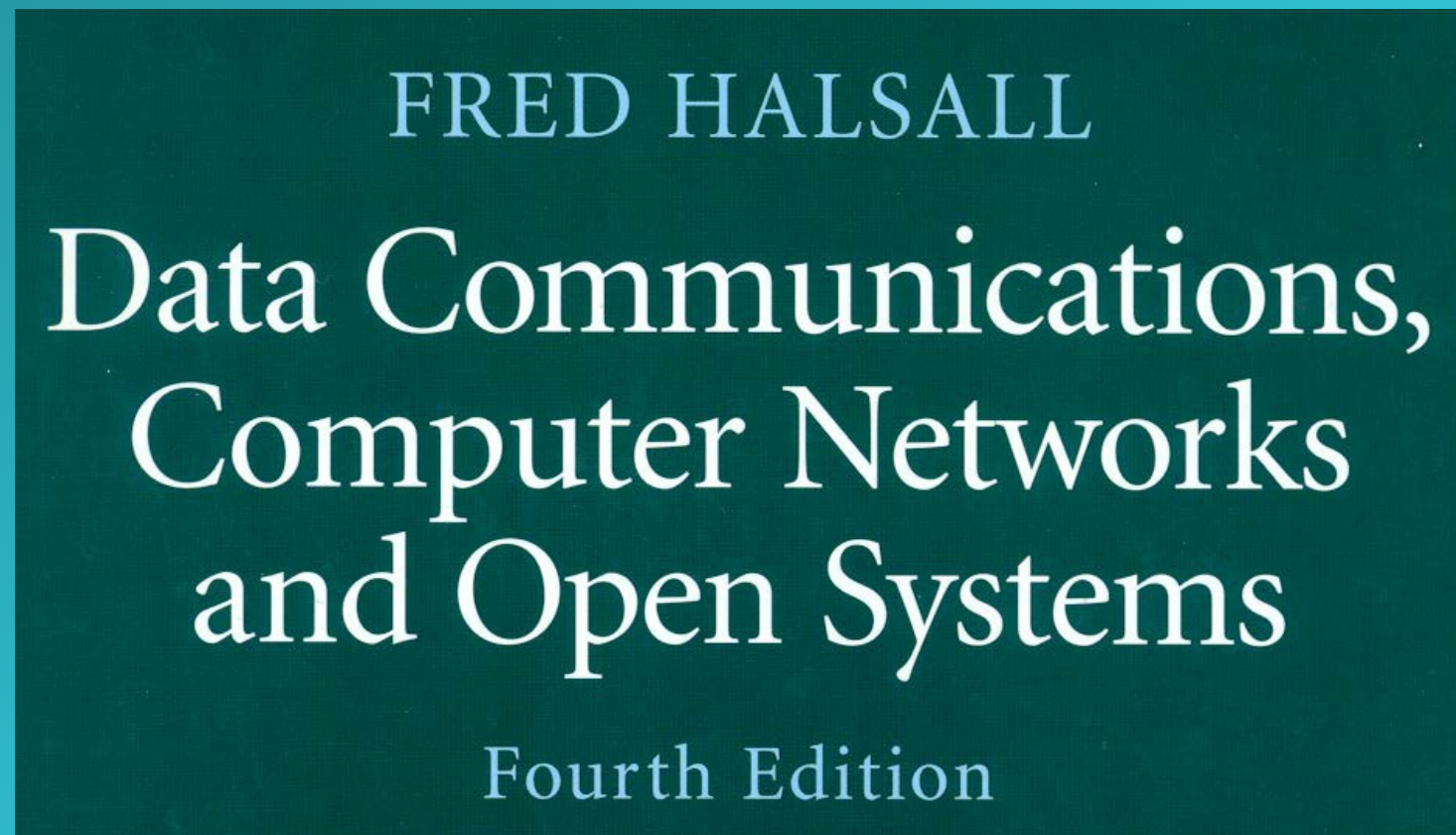
Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

# 13<sup>a</sup> Aula

Cap. 8 - Redes de Comunicações

Cap. 9 - Protocolos de Comunicações

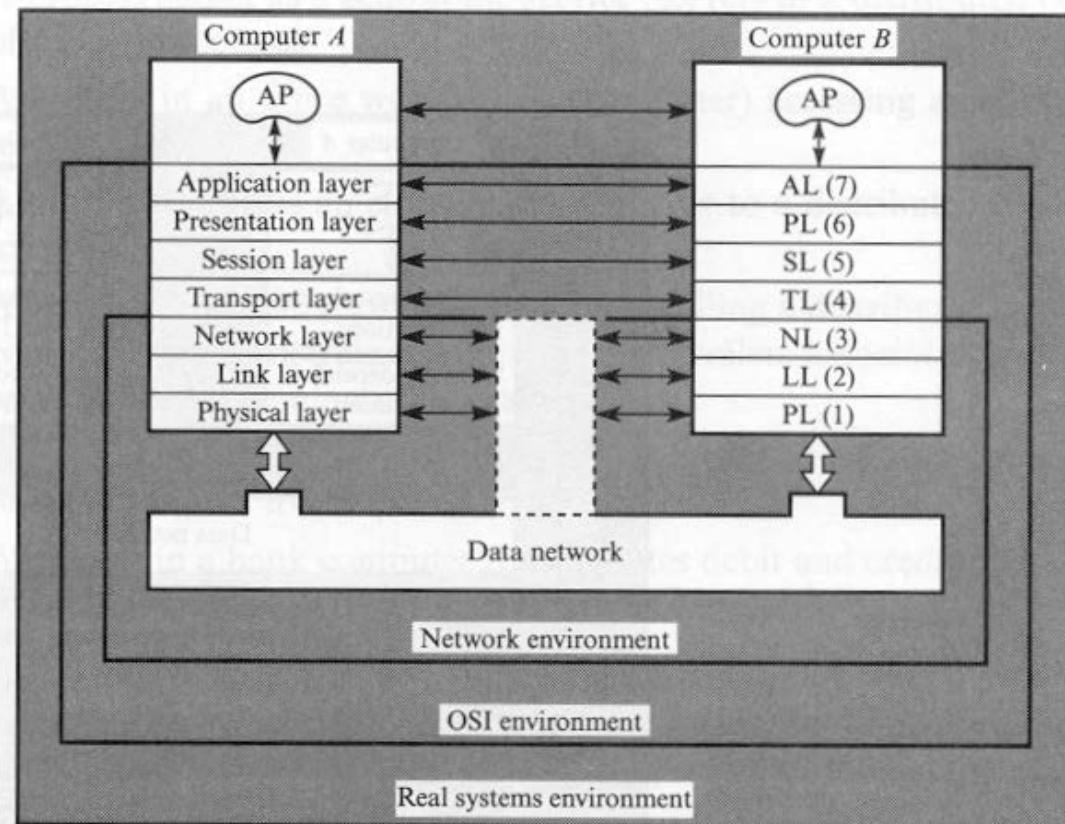




## Redes de Comunicações

**Objectivos: Comunicação entre duas aplicações residentes em dois computadores ligados em rede**

### Modelo OSI (Open Systems Interconnection)

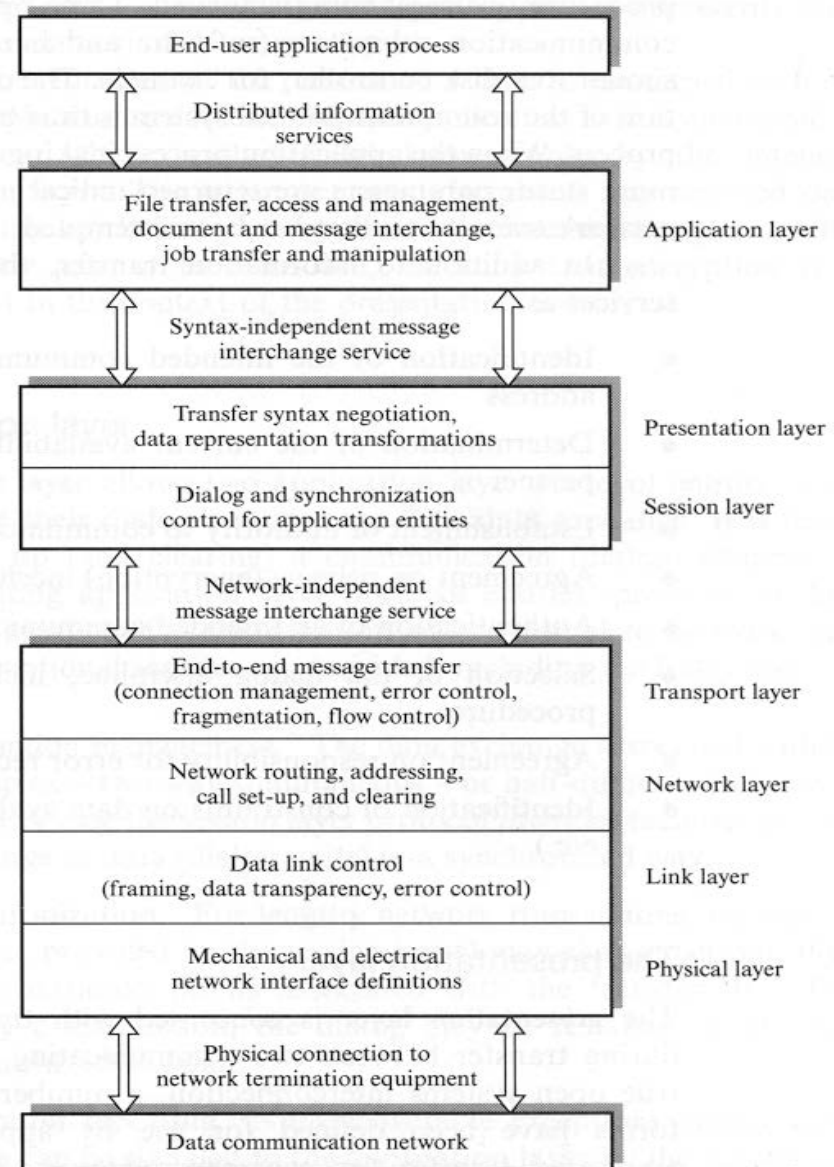


**Figure 1.10**  
Overall structure of the ISO reference model.

## Redes de Comunicações

### Definição dos níveis genéricos do protocolo:

- 7 níveis hierárquicos com objectivos independentes
- cada nível requer ferramentas e metodologias particulares.



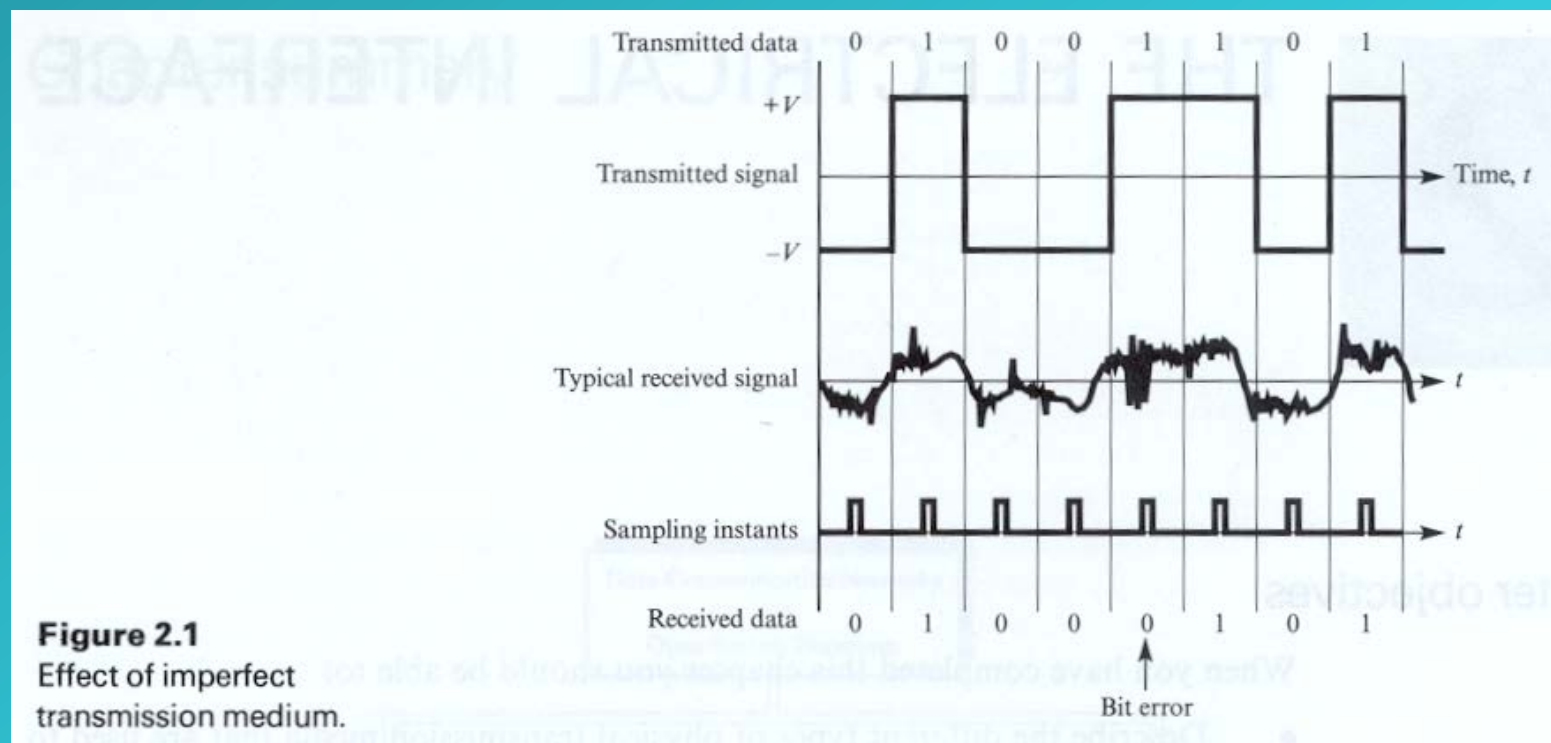
**Figure 1.11**  
Protocol layer  
summary.

## Redes de Comunicações

Interface com o nível físico

Problemas mecânicos e elétricos

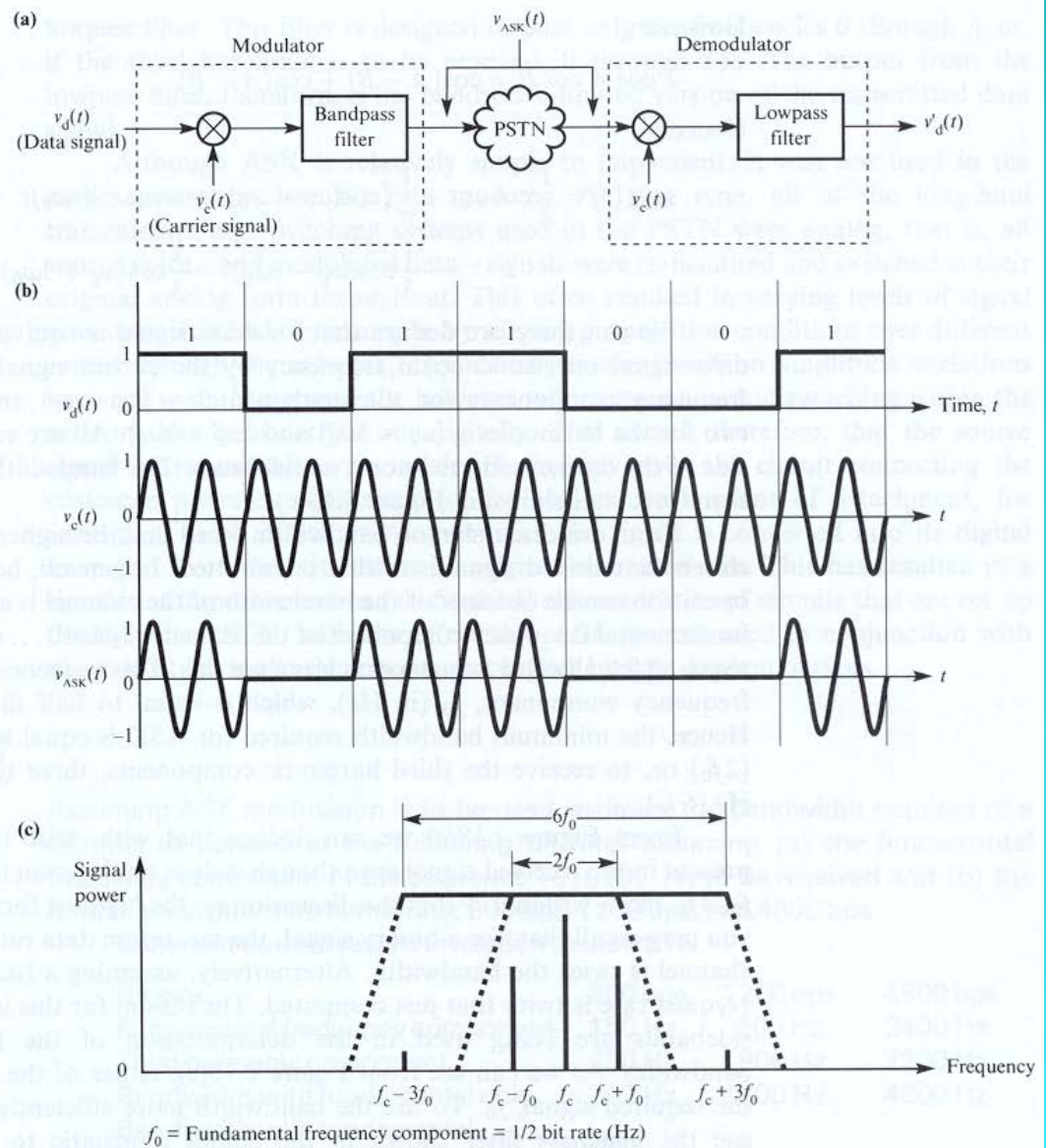
Problemas devido a *canal não ideal*



# Redes de Comunicações

Metologias de codificação  
ao nível do *bit*

Modulação ASK



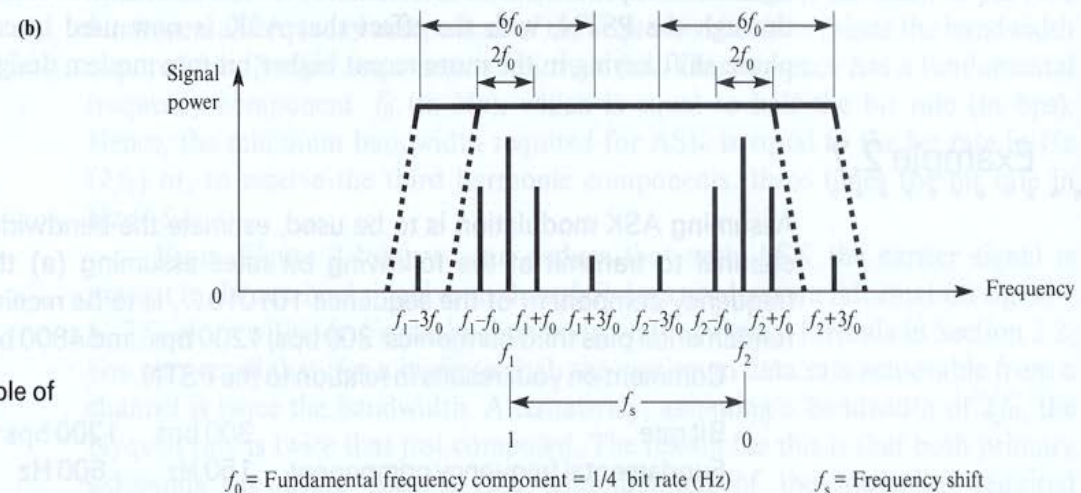
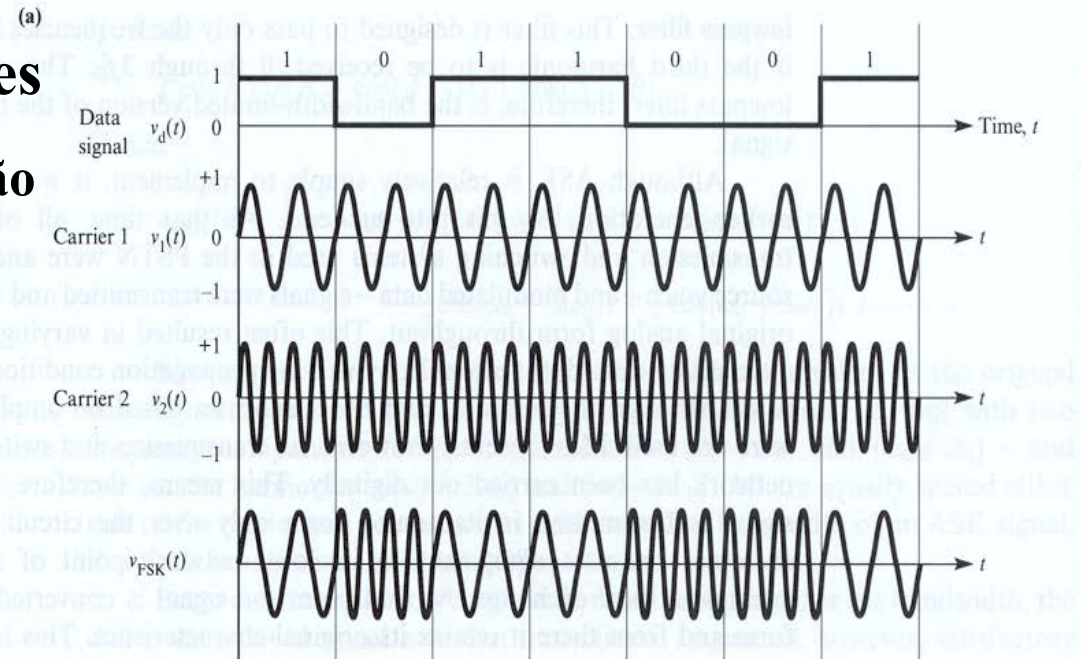
**Figure 2.18**  
Amplitude-shift keying: (a) circuit schematic; (b) waveform set; (c) bandwidth alternatives.



# Redes de Comunicações

Metologias de codificação  
ao nível do *bit*

Modulação FSK



**Figure 2.19**  
Frequency shift  
keying: (a) principle of  
operation;  
(b) bandwidth  
alternatives.

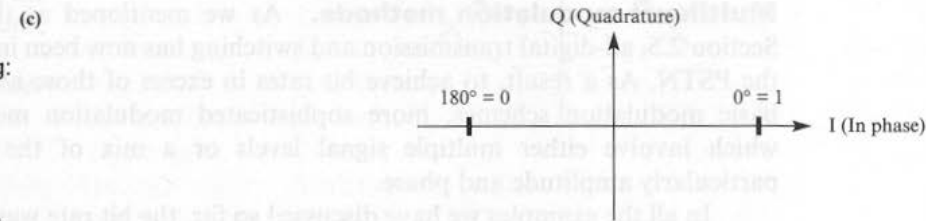
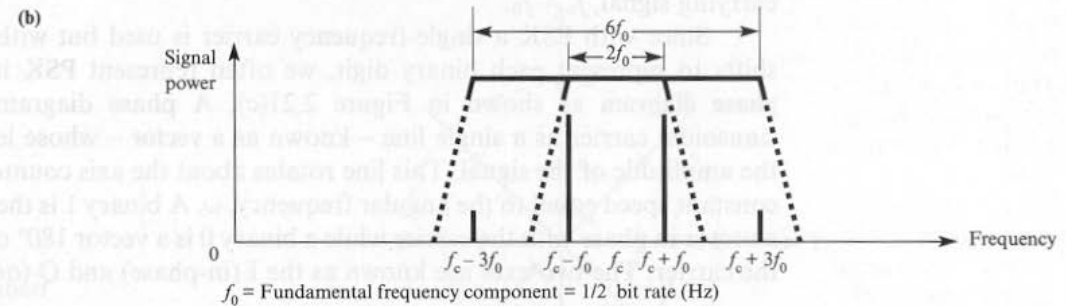
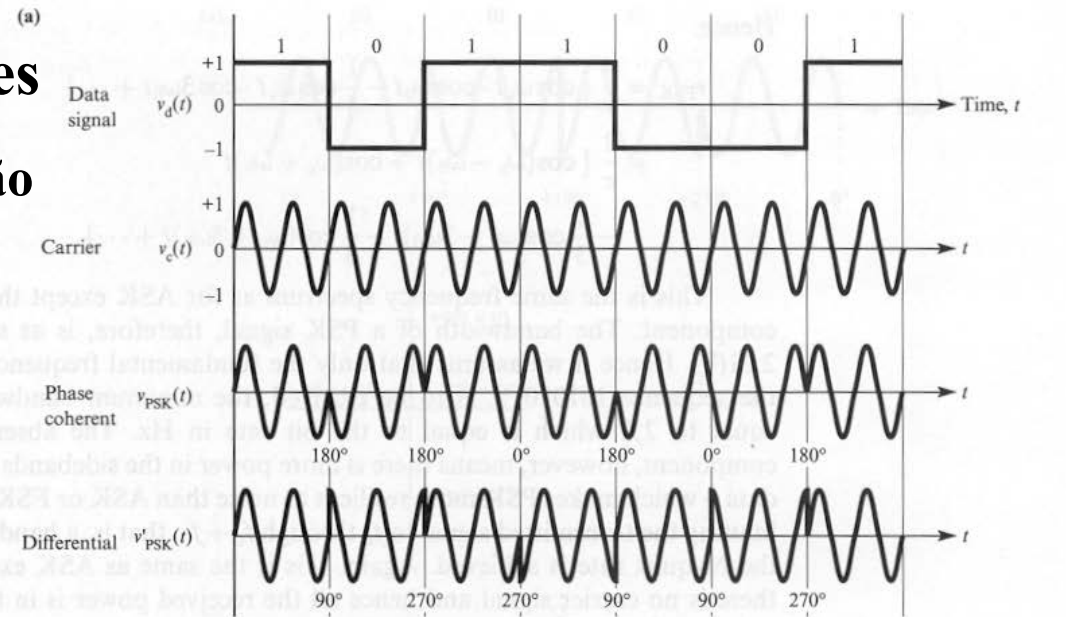
$f_0$  = Fundamental frequency component = 1/4 bit rate (Hz)       $f_s$  = Frequency shift



# Redes de Comunicações

## Metologias de codificação ao nível do *bit*

### Modulação PSK



**Figure 2.21**  
Phase-shift keying:  
(a) principle of operation;  
(b) bandwidth alternatives;  
(c) phase diagram.

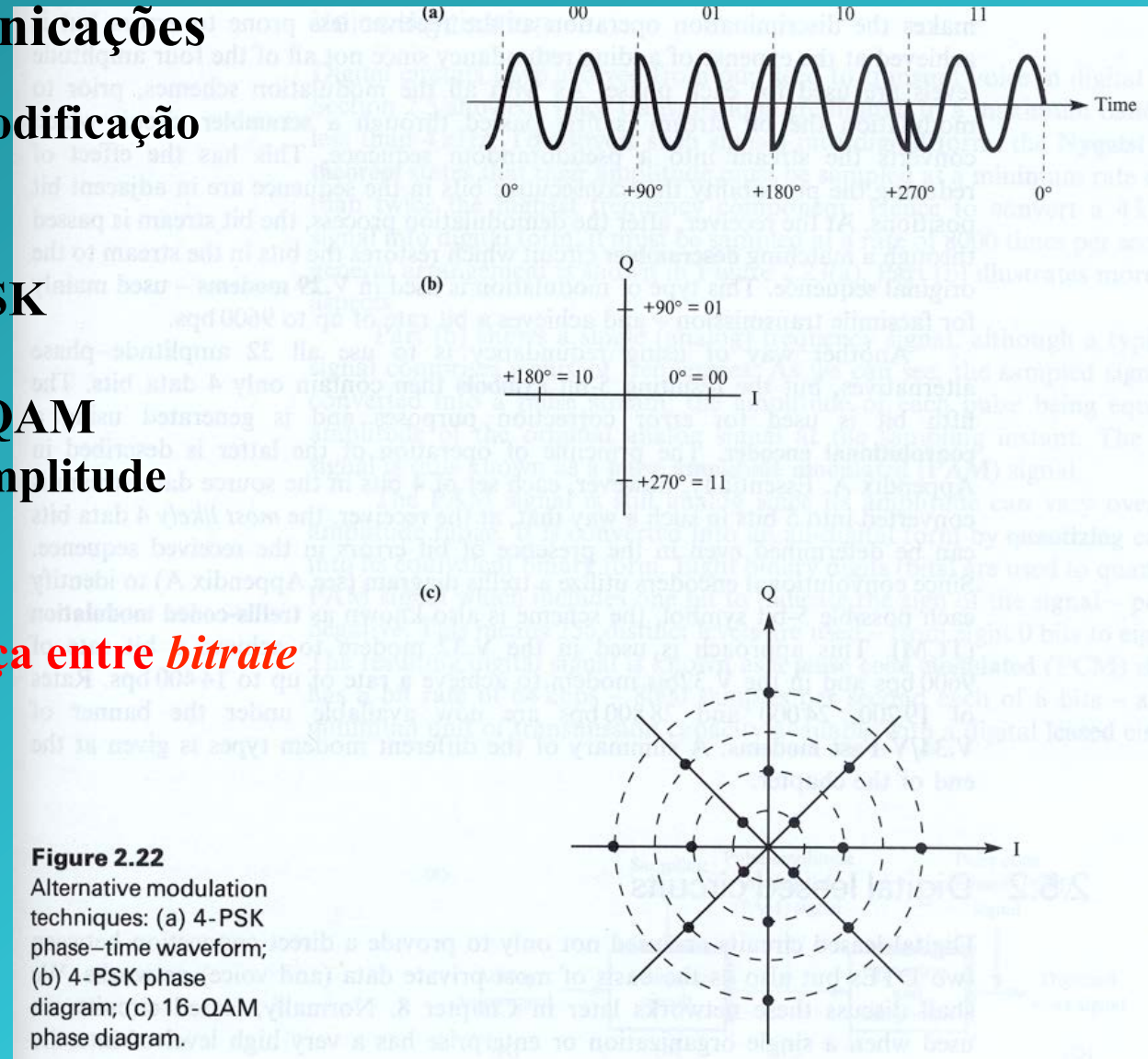
## Redes de Comunicações

Metologias de codificação  
ao nível do *bit*

Modulação 4-PSK

Modulação 16-QAM  
(Quadratura Amplitude  
Modulation)

**NOTA: Diferença entre *bitrate*  
e *baudrate***

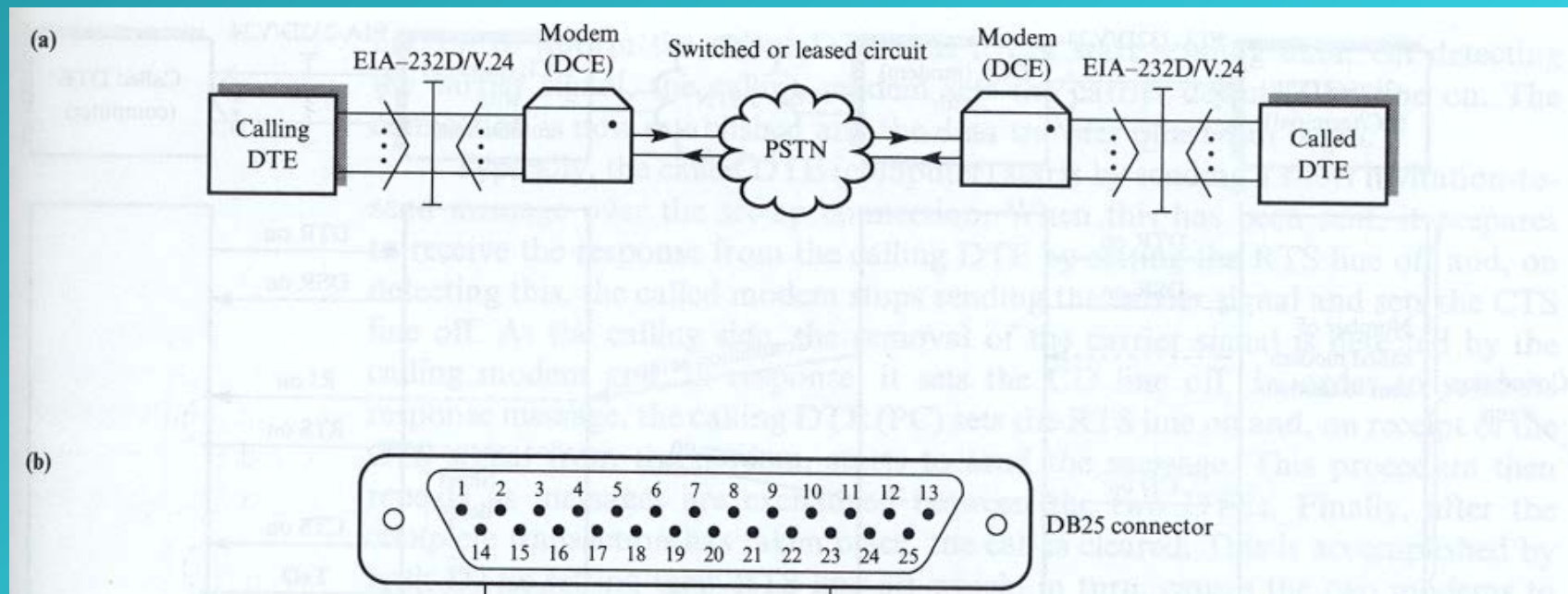


## Redes de Comunicações

Interligação mecânica e eléctrica

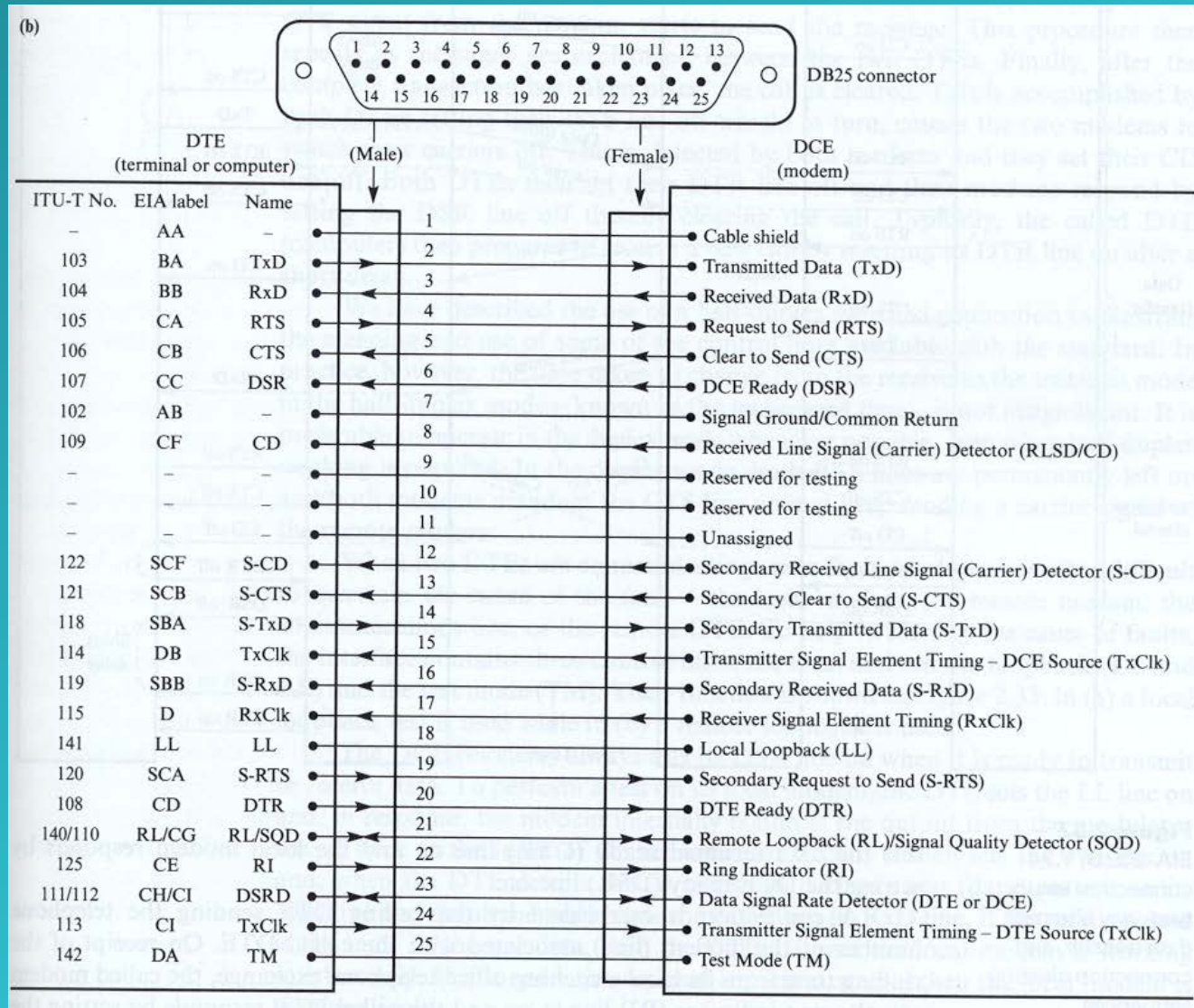
Exemplo: Norma EIA232D

(Ligação de computador a modem)

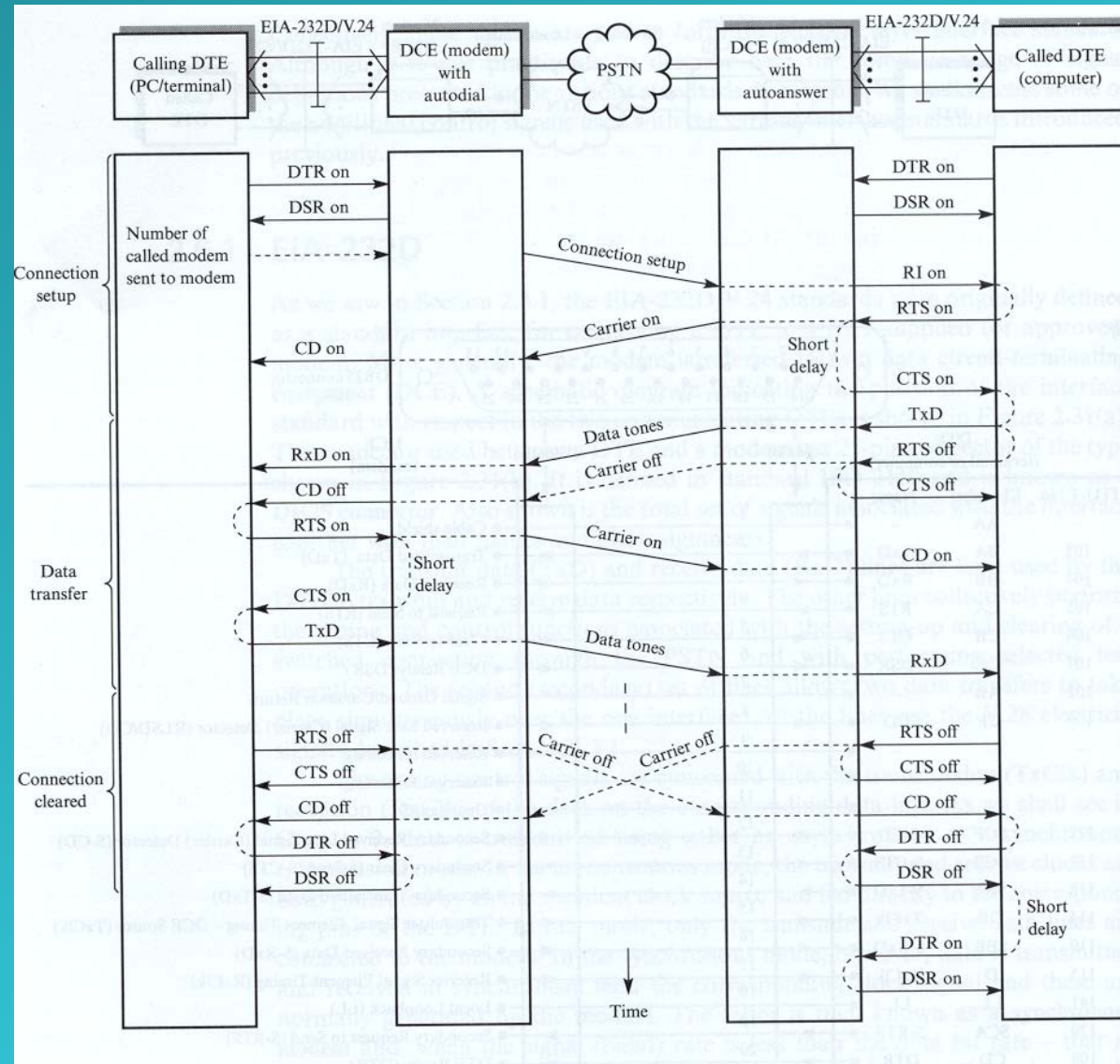




## Exemplo: Norma EIA232D (cont)



Exemplo: Protocolo associado à Norma EIA232D

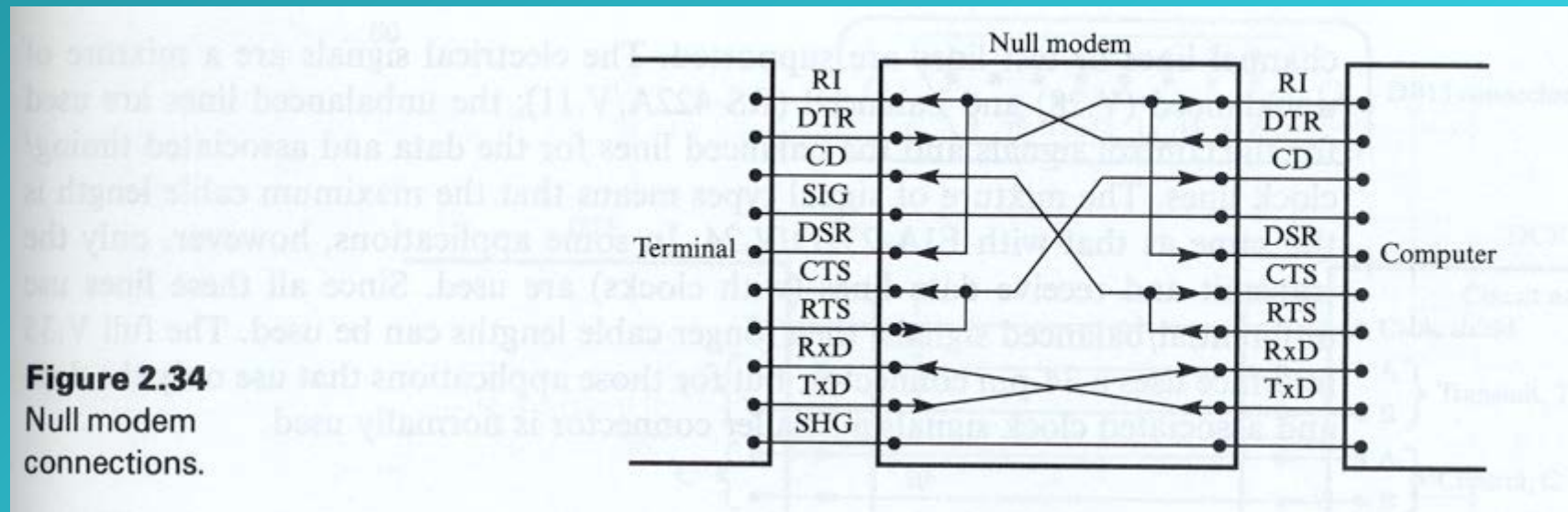




## Redes de Comunicações

Utilização para ligação entre dois computadores

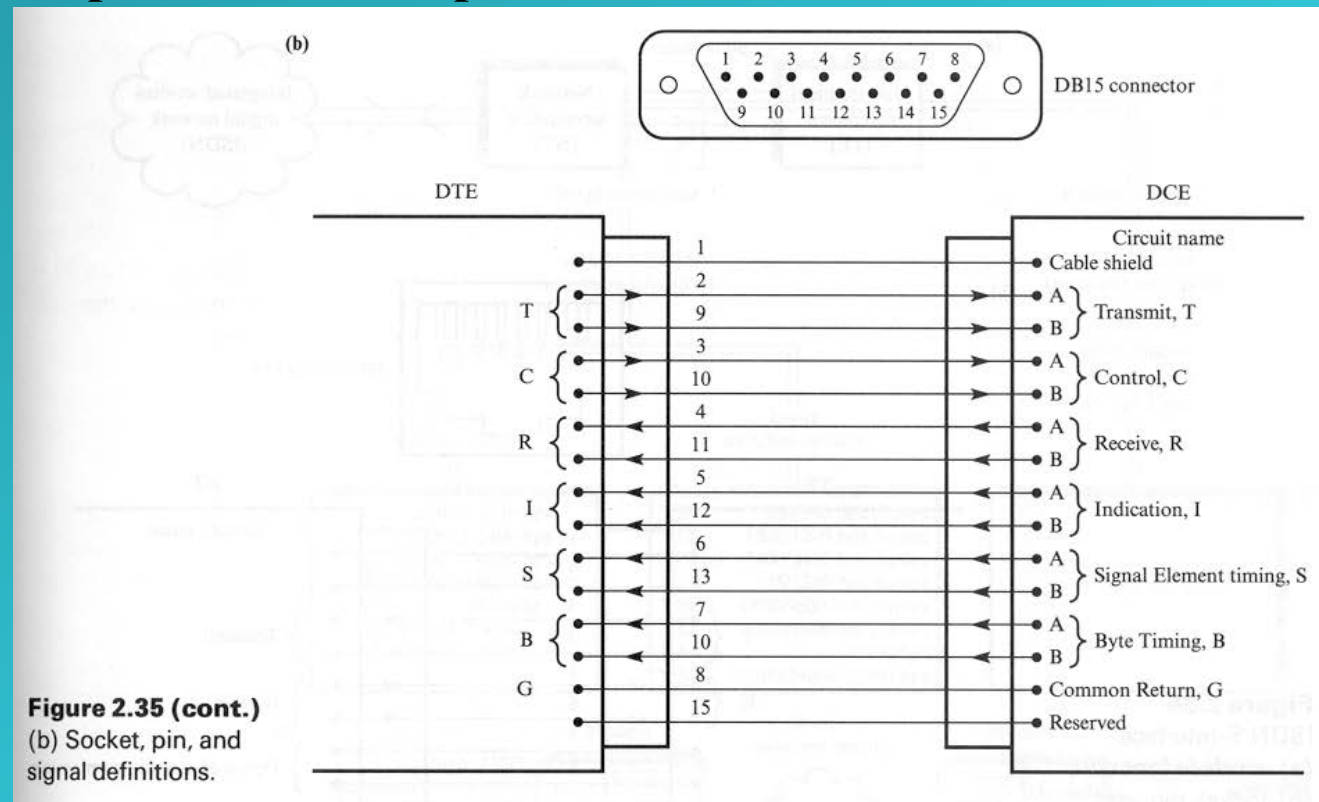
### *NULL MODEM*



# Redes de Comunicações

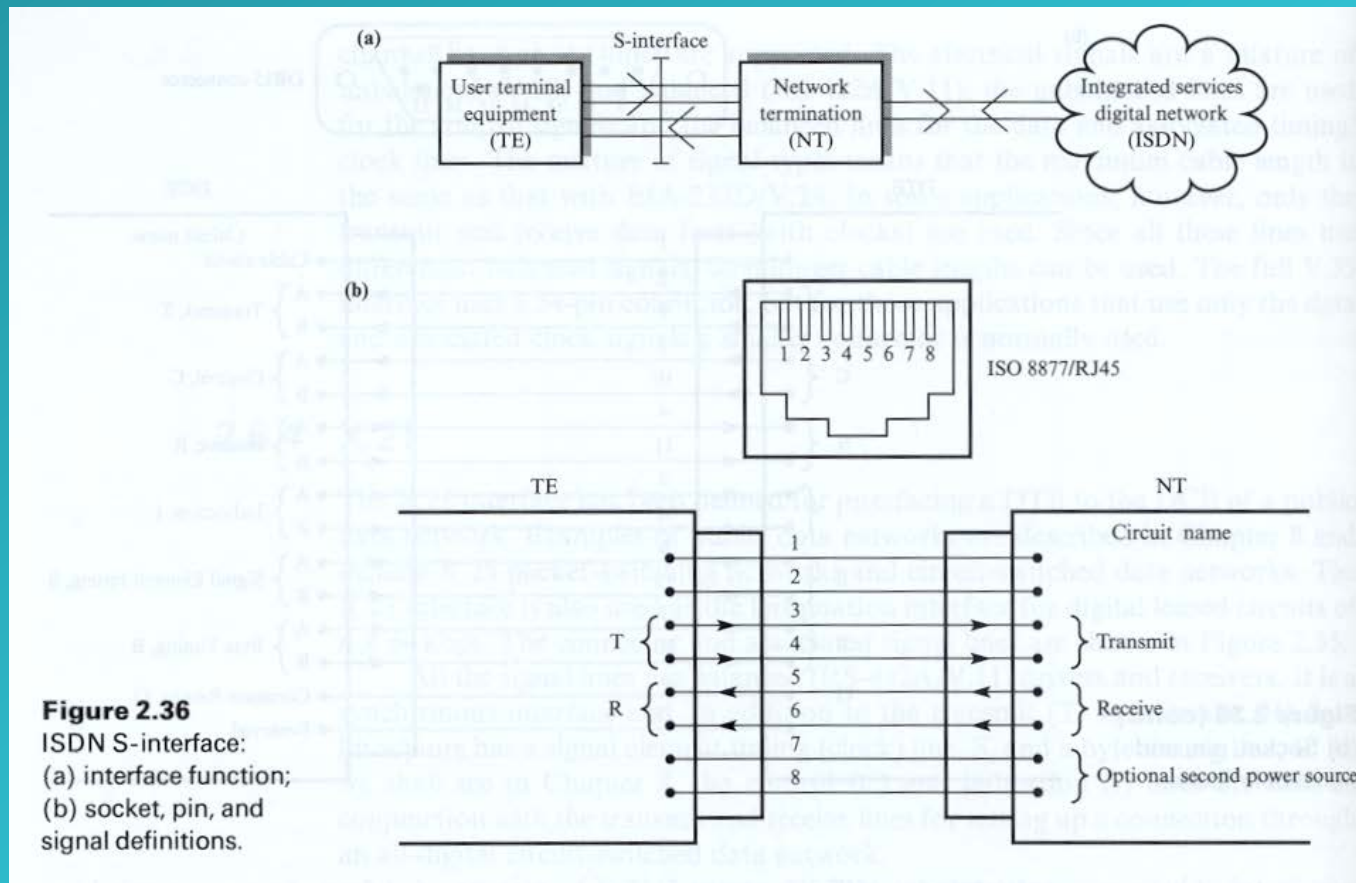
## Interligação mecânica e eléctrica

**Exemplo: Norma RDIS (ISDN) - Rede Digital Integrada de Serviços**  
**circuito duplo de 64kbts/s para voz mais 16 kbts/s de dados**



## Redes de Comunicações

### Outra norma de interligação mecânica e eléctrica em RDIS (ISO 8877)



**Figure 2.36**  
ISDN S-interface:  
(a) interface function;  
(b) socket, pin, and  
signal definitions.

# Redes de Comunicações

Metologias de codificação<sup>(b)</sup>  
ao nível de trama (*frame*)

Tabela de símbolos ASCII

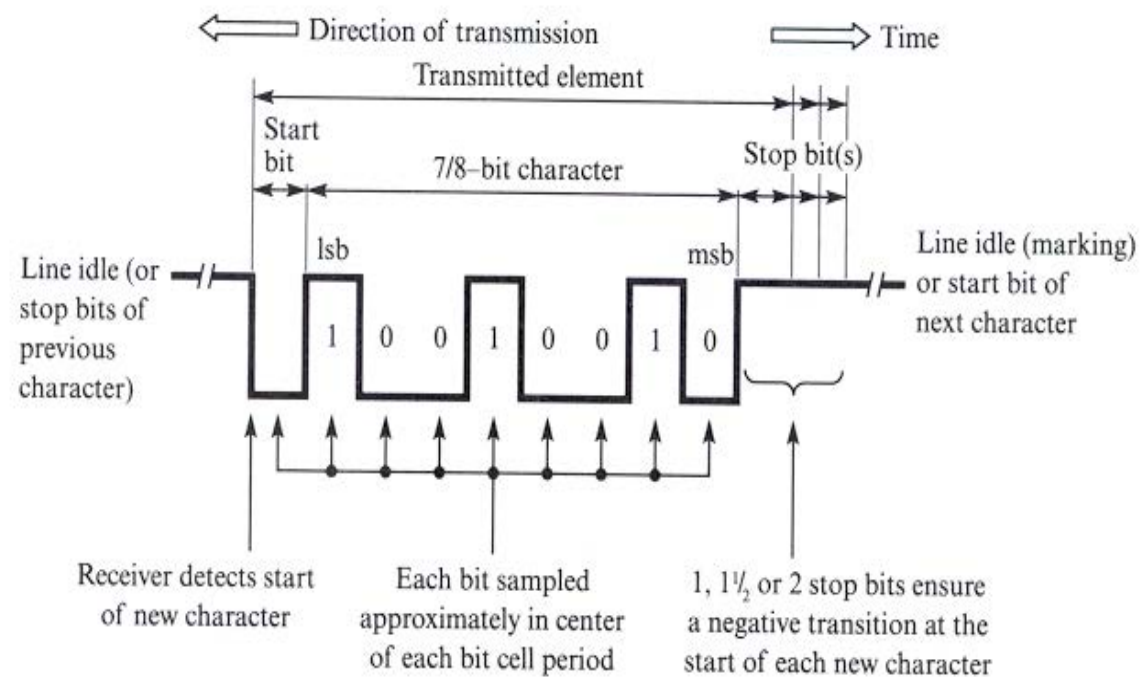
Bit positions				7	6	5	4	3	2	1	0	0	0	0	1	1	1	1
0	0	0	0	NUL	DLE	SP	0	@	P	\	p							
0	0	0	1	SOH	DC1	!	1	A	Q	a	q							
0	0	1	0	STX	DC2	"	2	B	R	b	r							
0	0	1	1	ETX	DC3	#	3	C	S	c	s							
0	1	0	0	EOT	DC4	\$	4	D	T	d	t							
0	1	0	1	ENQ	NAK	%	5	E	U	e	u							
0	1	1	0	ACK	SYN	&	6	F	V	f	v							
0	1	1	1	BEL	ETB	'	7	G	W	g	w							
1	0	0	0	BS	CAN	(	8	H	X	h	x							
1	0	0	1	HT	EM	)	9	I	Y	i	y							
1	0	1	0	LF	SUB	*	:	J	Z	j	z							
1	0	1	1	VT	ESC	+	;	K	[	k	{							
1	1	0	0	FF	FS	.	<	L	\	l								
1	1	0	1	CR	GS	-	=	M	]	m	}							
1	1	1	0	SO	RS	.	>	N	^	n	~							
1	1	1	1	SI	US	/	?	O	_	o	DEL							

Figure 3.1 (cont.)  
(b) ASCII/IA5.

## Redes de Comunicações

### Metologias de codificação ao nível de trama (*frame*)

### Construção de Tramas



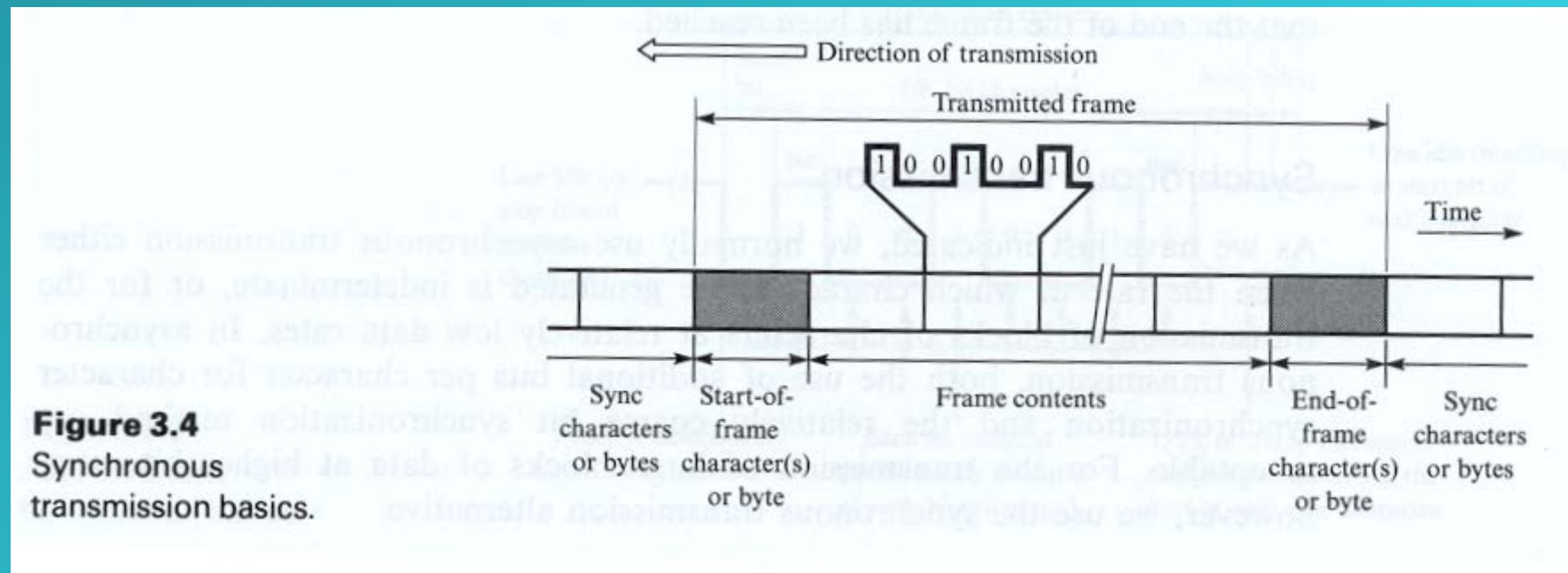
**Figure 3.3**  
Asynchronous transmission basics.



## Redes de Comunicações

Metologias de codificação  
ao nível de trama (*frame*)

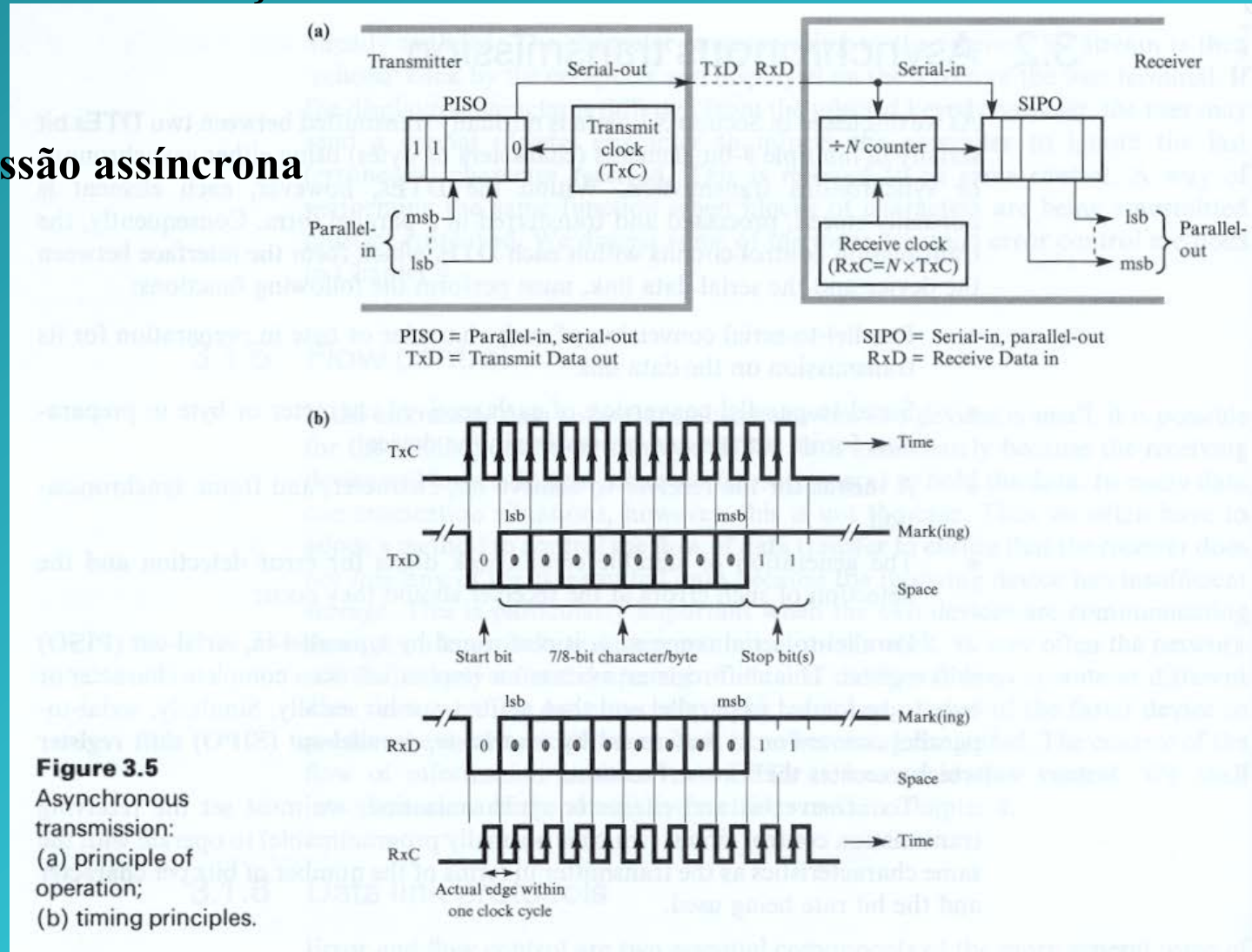
Construção de Tramas



**Figure 3.4**  
Synchronous  
transmission basics.

# Redes de Comunicações

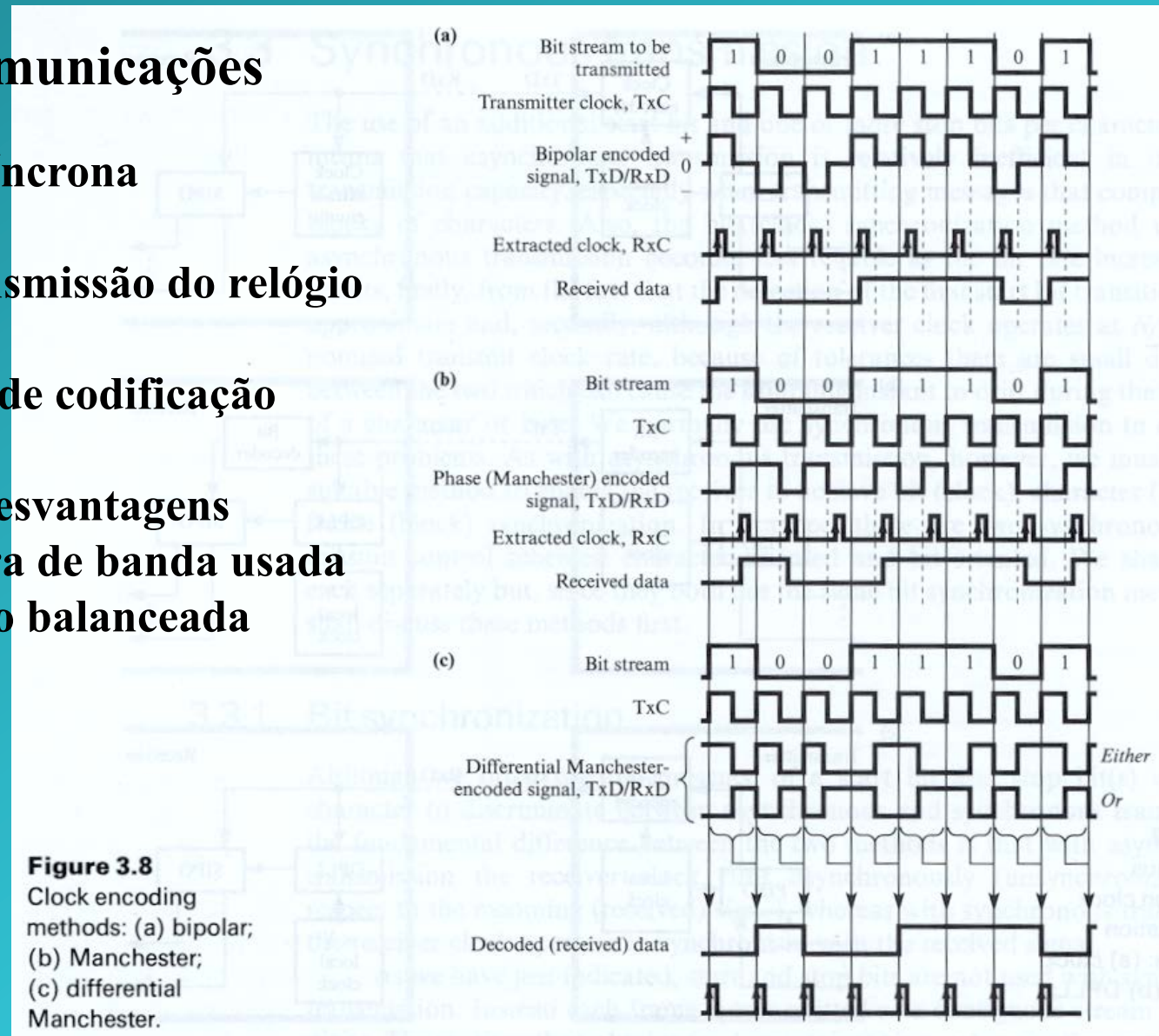
## Transmissão assíncrona



# Redes de Comunicações

## Transmissão síncrona

- implica a transmissão do relógio
- metodologias de codificação
- vantagens e desvantagens
  - largura de banda usada
  - ligação balanceada



# Automação Industrial baseada em PLCs

## 14ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

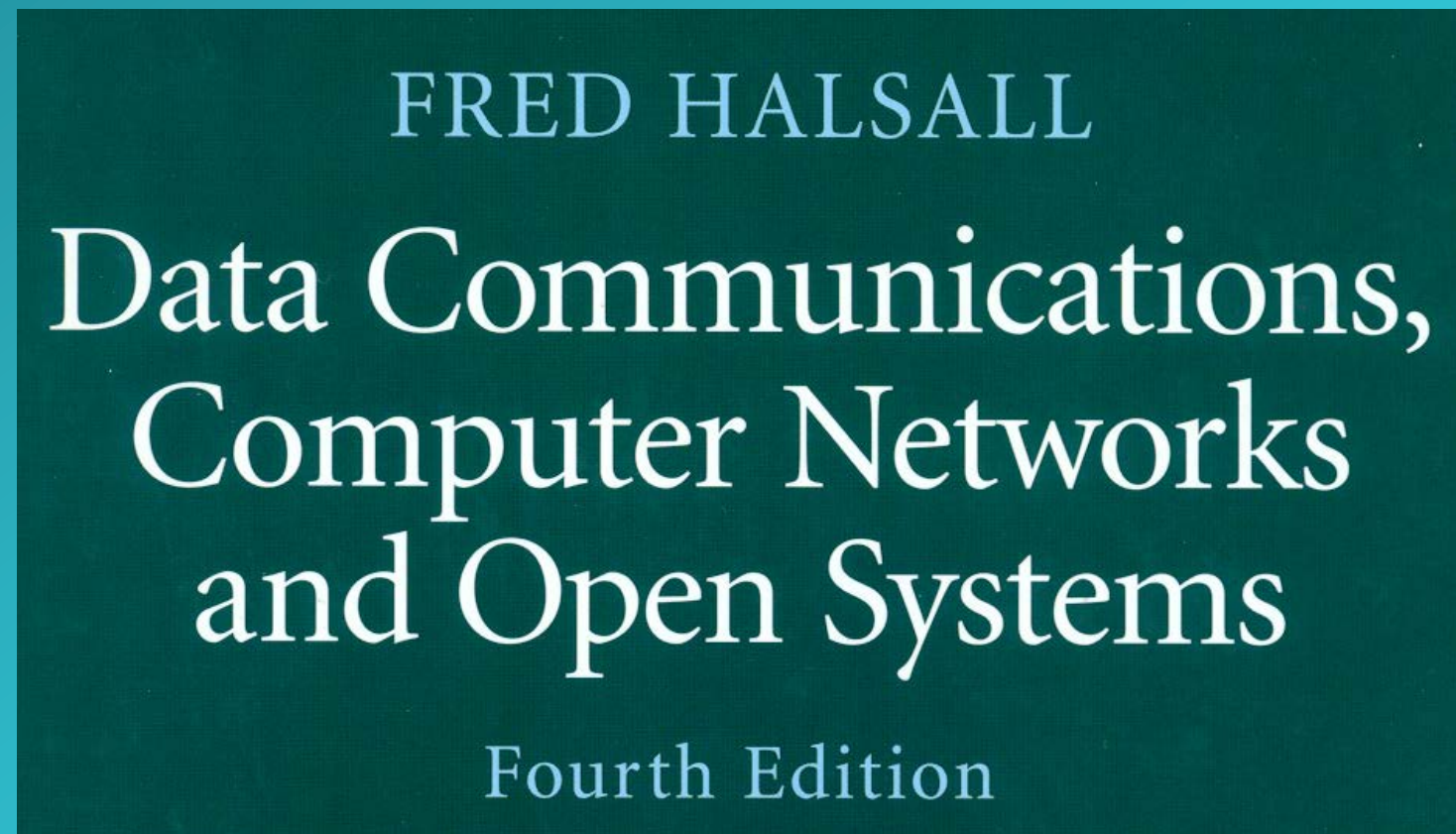
Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)



# 14<sup>a</sup> Aula

## Cap. 9 - Protocolos de Comunicações

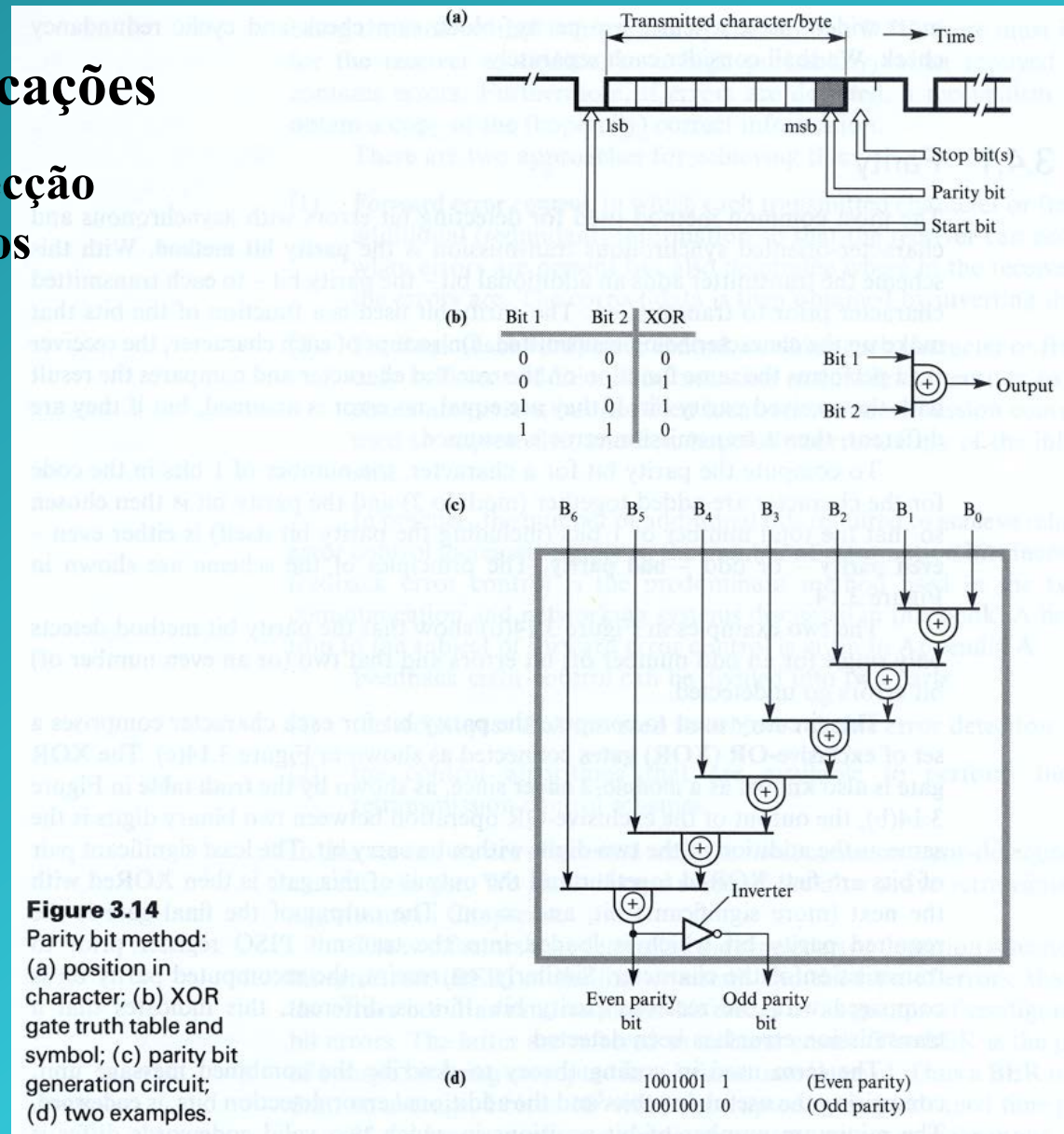




# Redes de Comunicações

## Metologias de detecção e correcção de erros

### Paridade



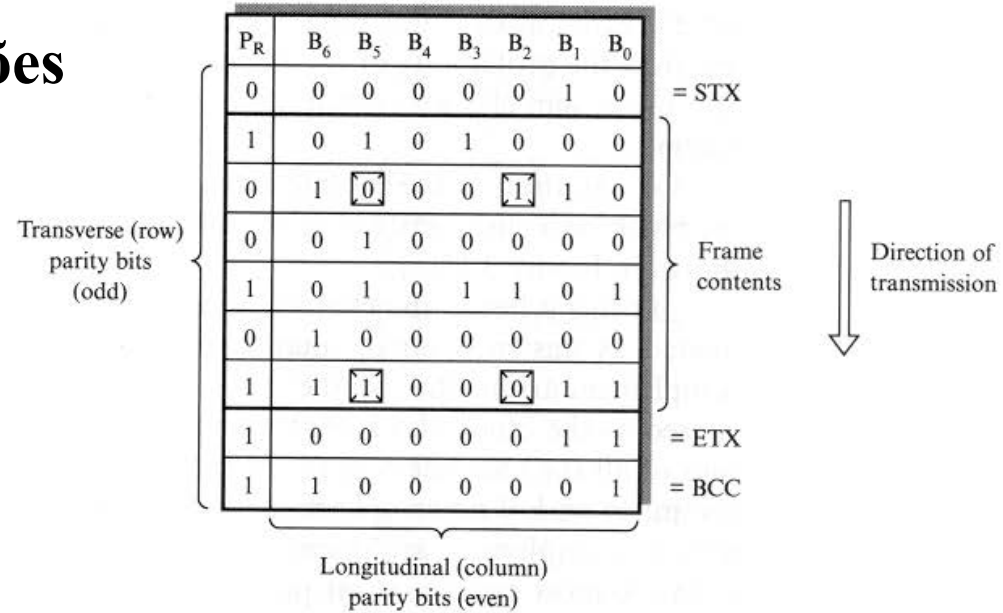
**Figure 3.14**  
Parity bit method:  
(a) position in character;  
(b) XOR gate truth table and symbol;  
(c) parity bit generation circuit;  
(d) two examples.

# Redes de Comunicações

## Metologias de detecção e correcção de erros

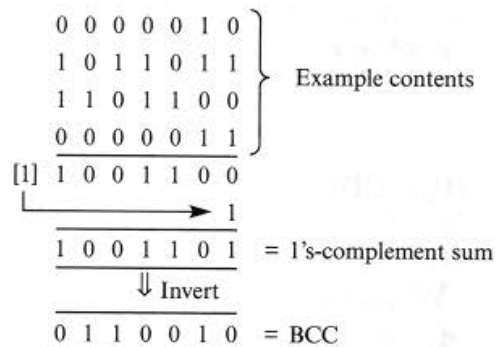
### Check sum

(a)

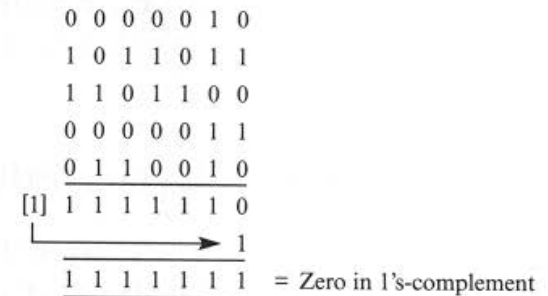


□ = Example of undetected error combination  
 P<sub>R</sub> = Row parity bit

(b) At sending side:



At receiving side:



**Figure 3.15**  
 Block sum check example: (a) row and column parity bits; (b) 1's complement sum.

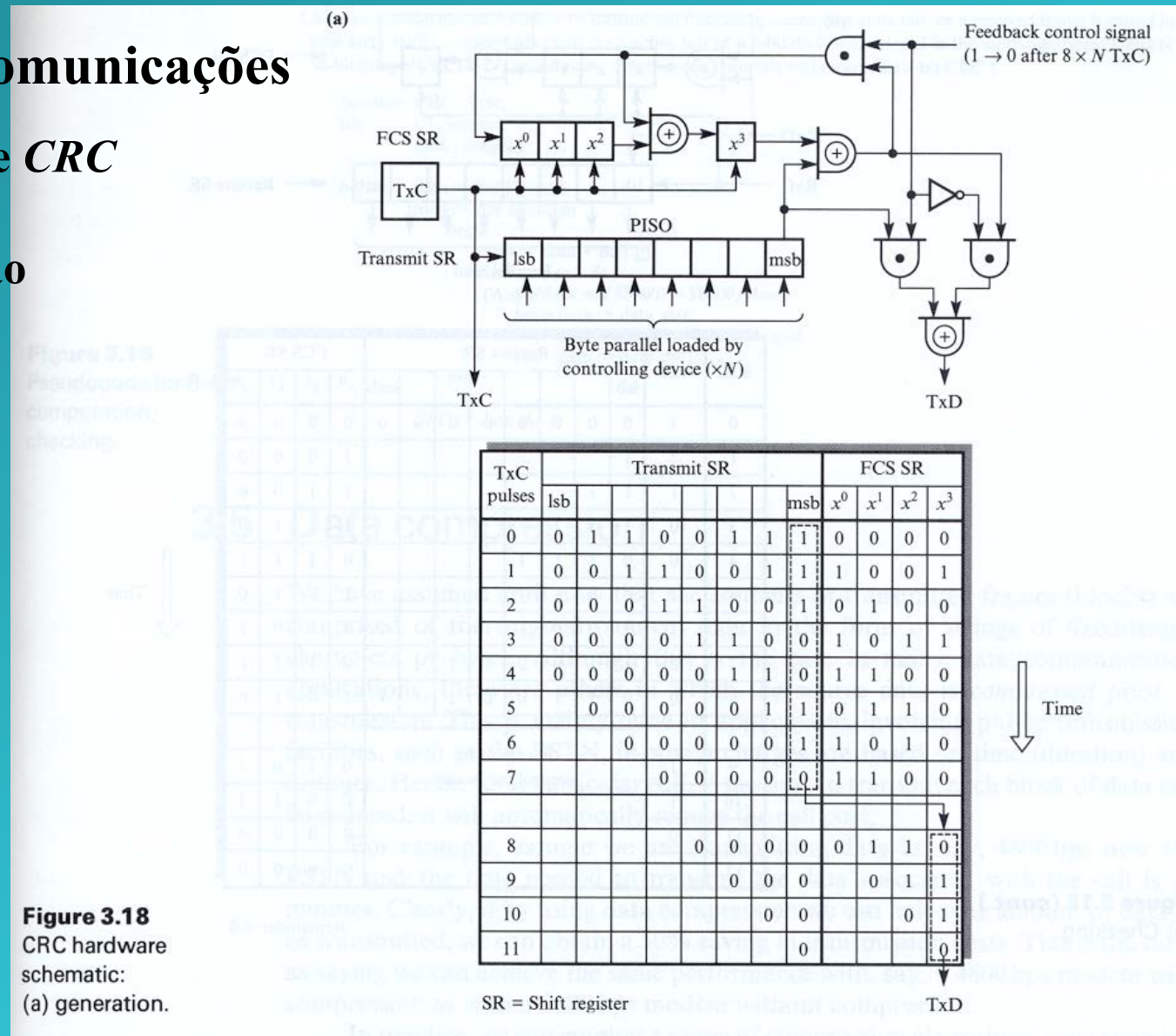




# Redes de Comunicações

## Exemplo de CRC

## Transmissão



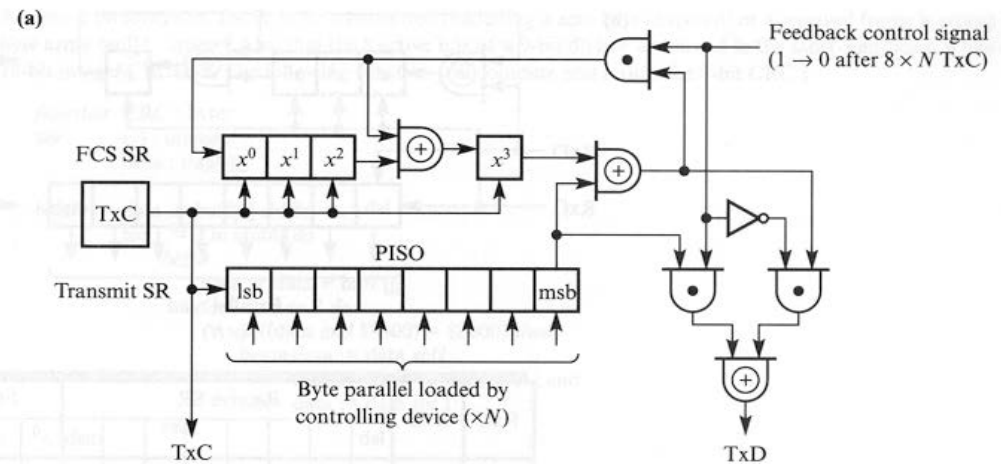
**Figure 3.18**  
CRC hardware schematic:  
(a) generation.



# Redes de Comunicações

## Exemplo de CRC

## Recepção



TxC pulses	Transmit SR							FCS SR			
	lsb						msb	$x^0$	$x^1$	$x^2$	$x^3$
0	0	1	1	0	0	1	1	0	0	0	0
1	0	0	1	1	0	0	1	1	0	0	1
2	0	0	0	1	1	0	0	1	0	1	0
3	0	0	0	0	1	1	0	0	1	0	1
4	0	0	0	0	0	1	1	0	1	1	0
5		0	0	0	0	0	1	1	0	1	1
6			0	0	0	0	0	1	1	0	1
7				0	0	0	0	0	1	1	0
8					0	0	0	0	0	1	1
9						0	0	0	0	1	1
10							0	0	0	0	1
11								0	0	0	0

**Figure 3.18**  
CRC hardware schematic:  
(a) generation.

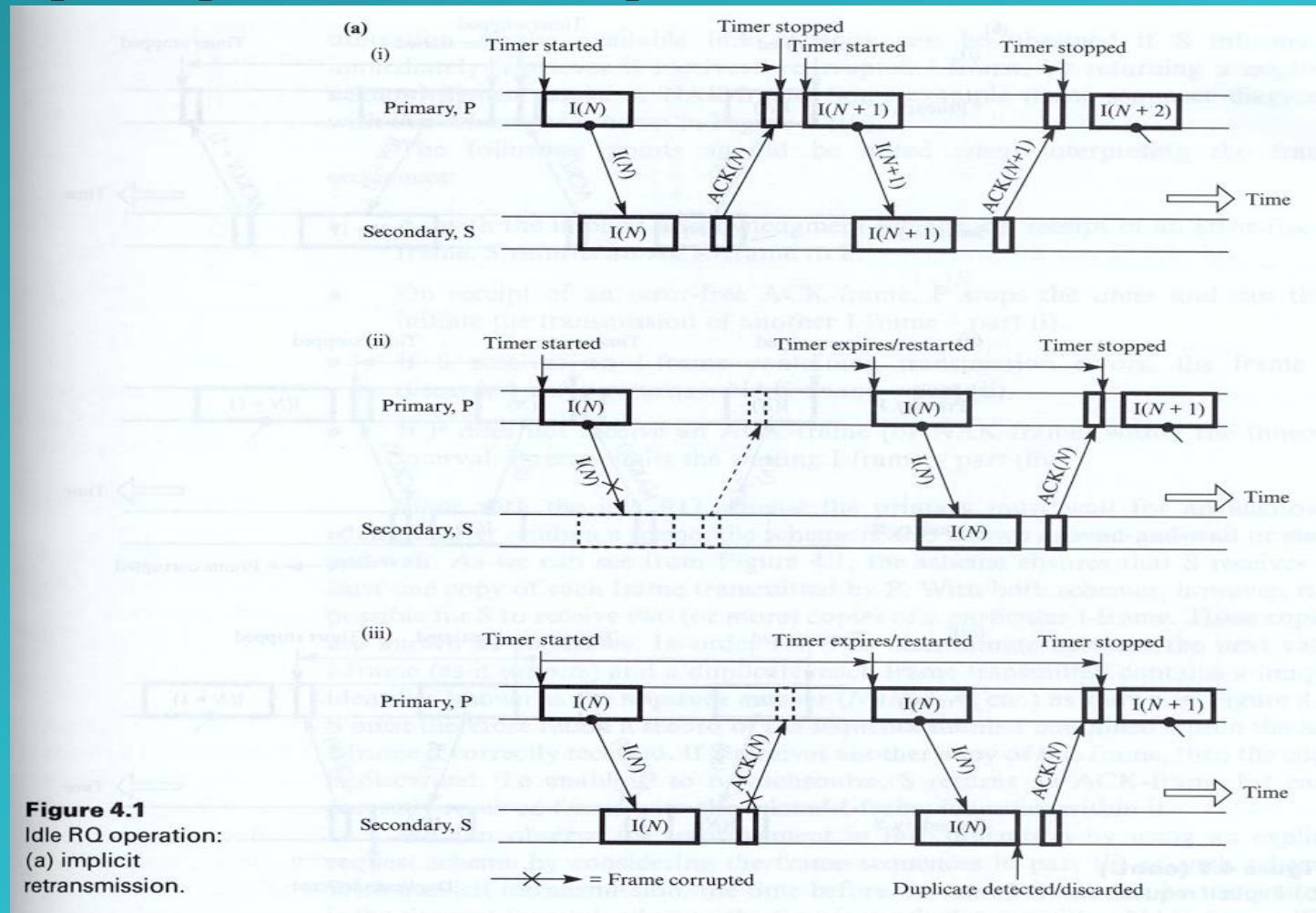
SR = Shift register

TxD

Time ↓

# Protocolos de Comunicações

## Idle repeat request IRQ - modo implícito



**Figure 4.1**  
Idle RQ operation:  
(a) implicit retransmission.

# Protocolos de Comunicações

## Idle repeat request IRQ - modo explícito

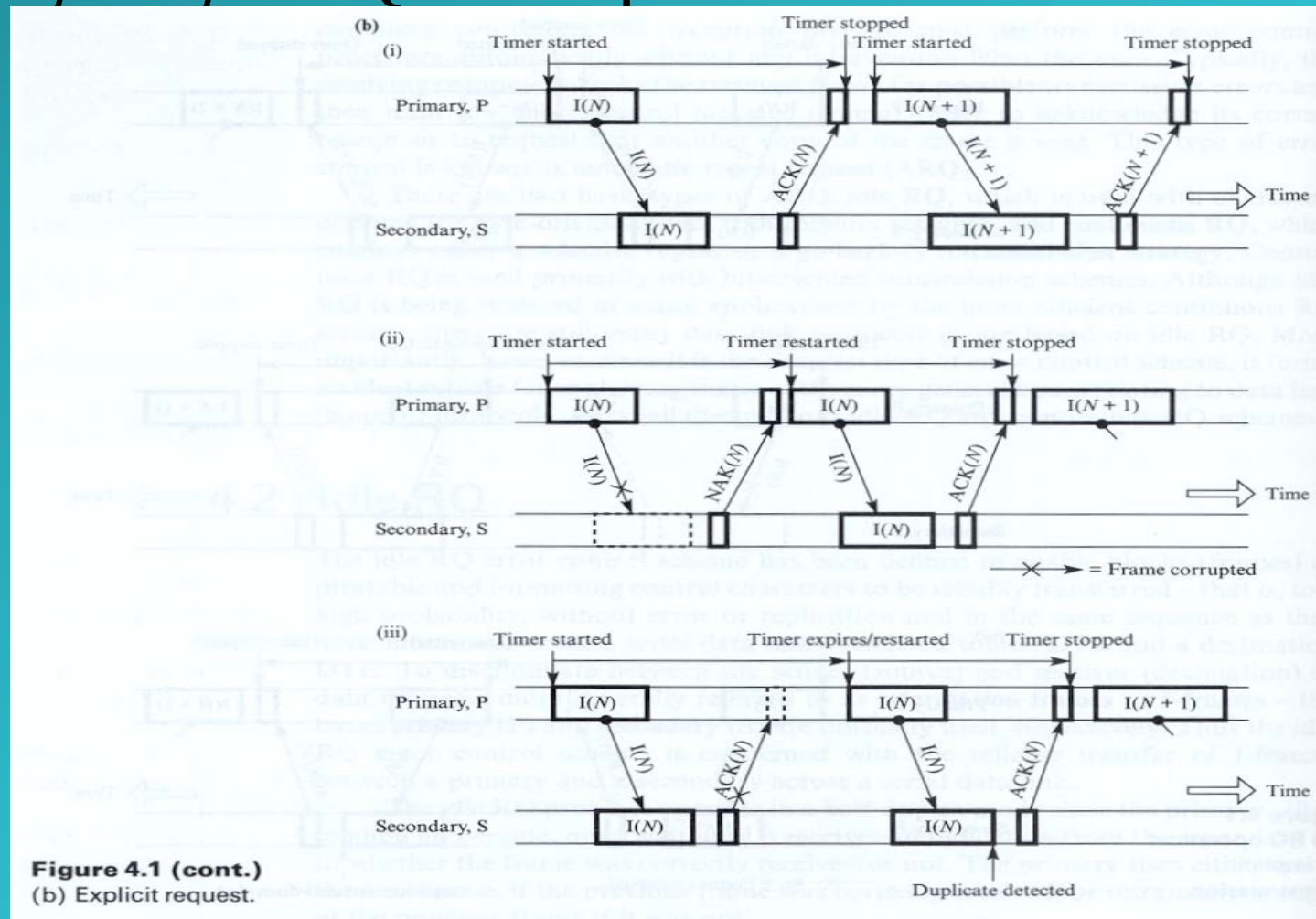
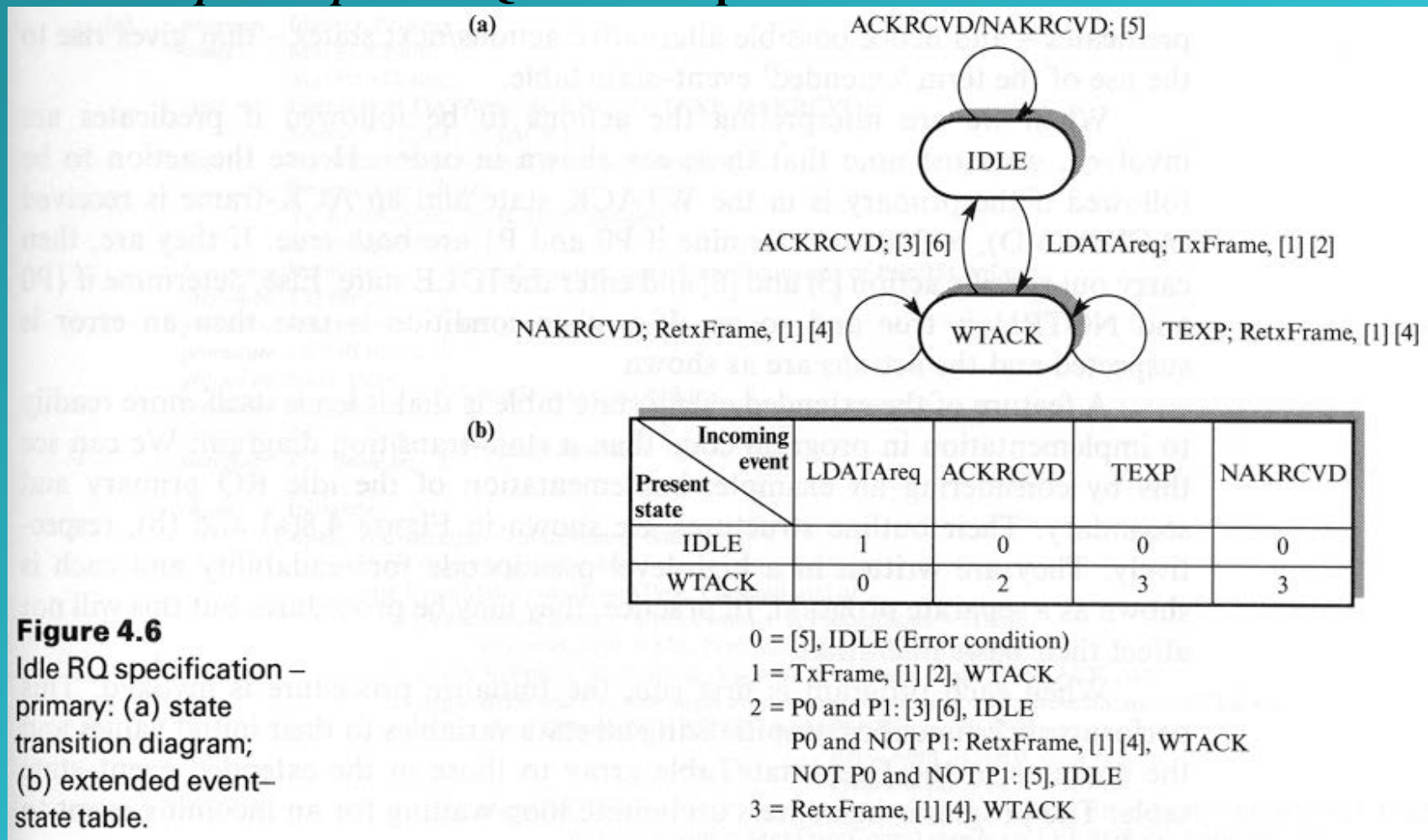


Figure 4.1 (cont.)  
(b) Explicit request.

# Protocolos de Comunicações

## Idle repeat request IRQ - modo explícito

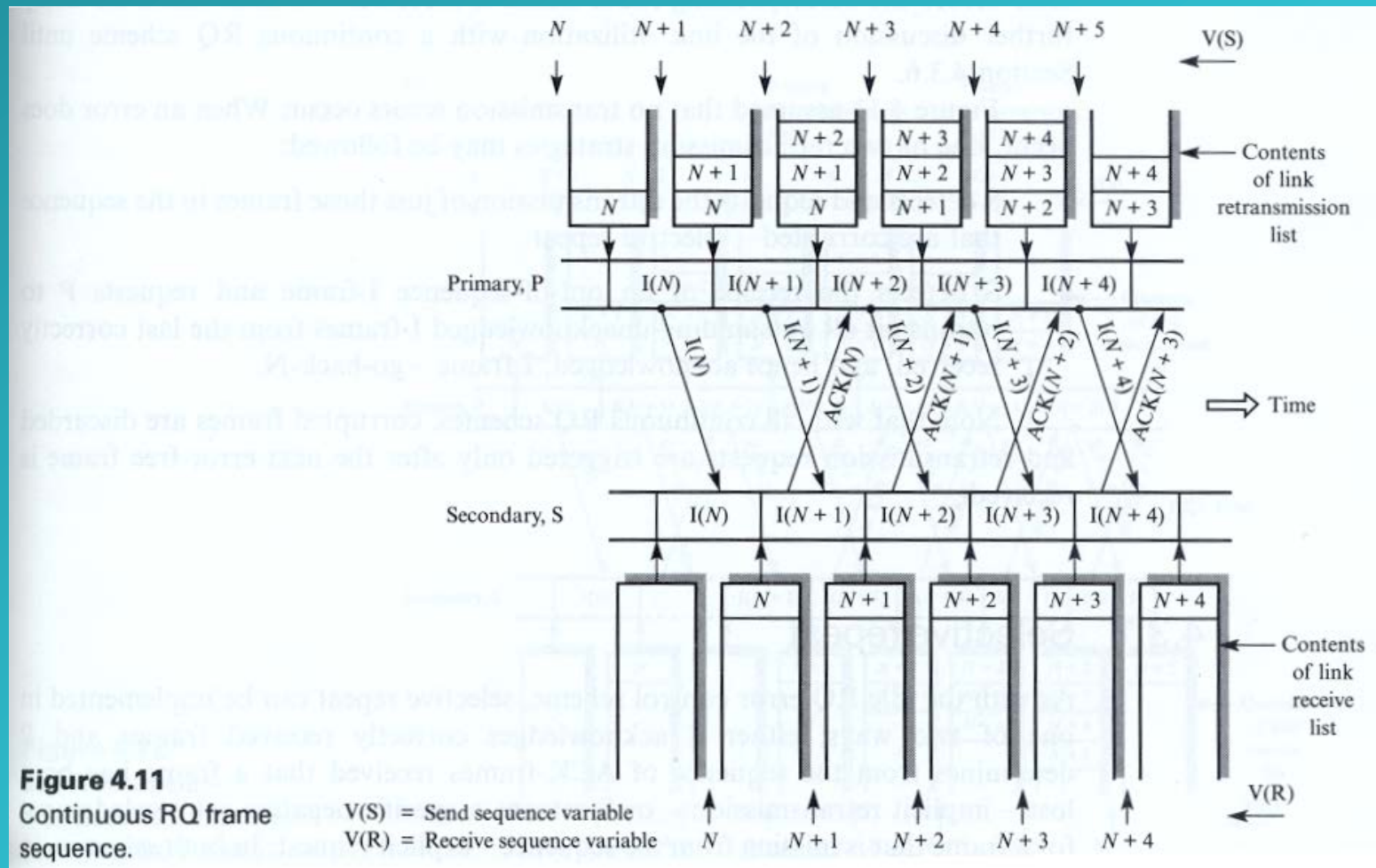


**Figure 4.6**  
 Idle RQ specification –  
 primary: (a) state  
 transition diagram;  
 (b) extended event–  
 state table.



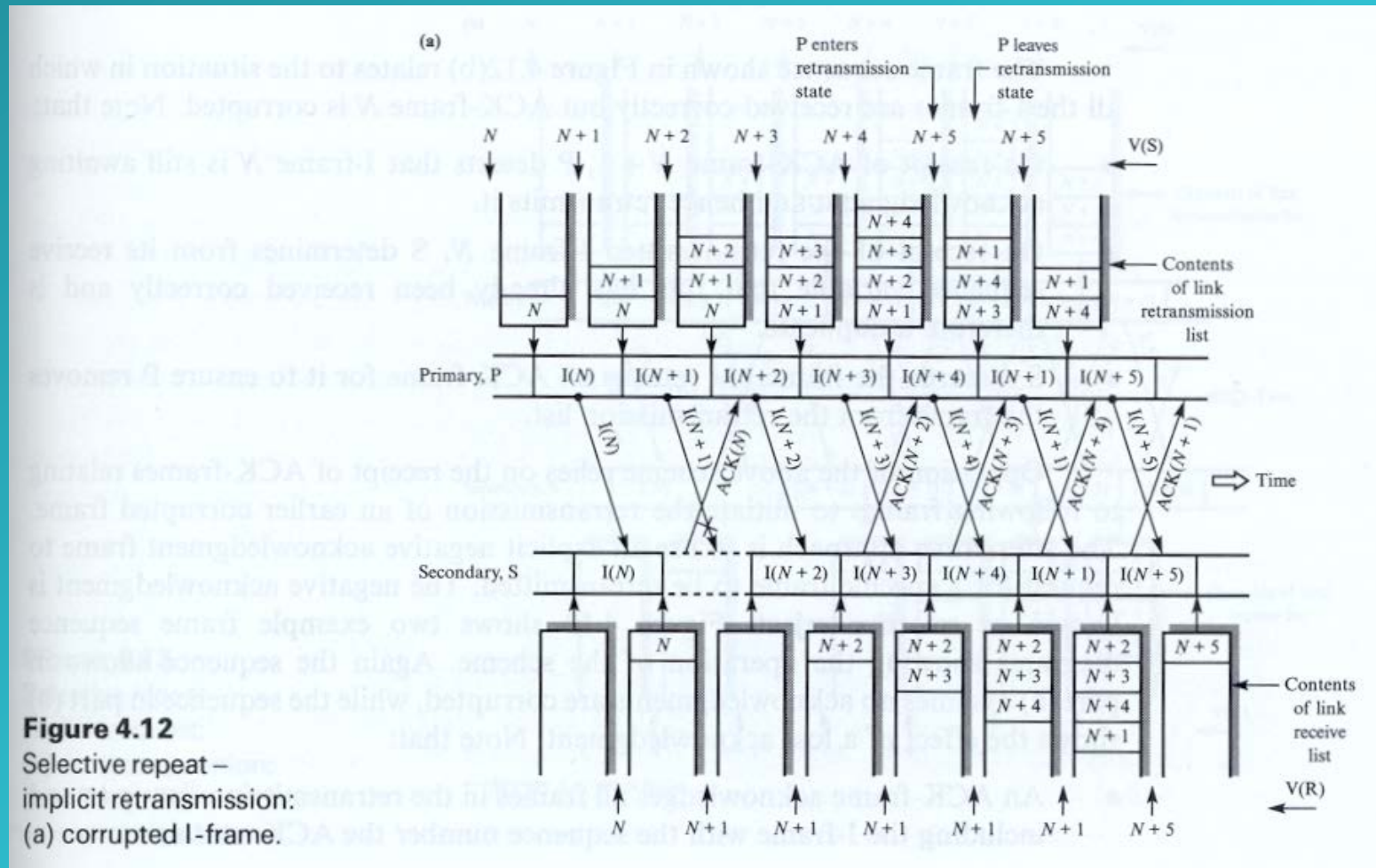
# Protocolos de Comunicações

## Continuous RQ



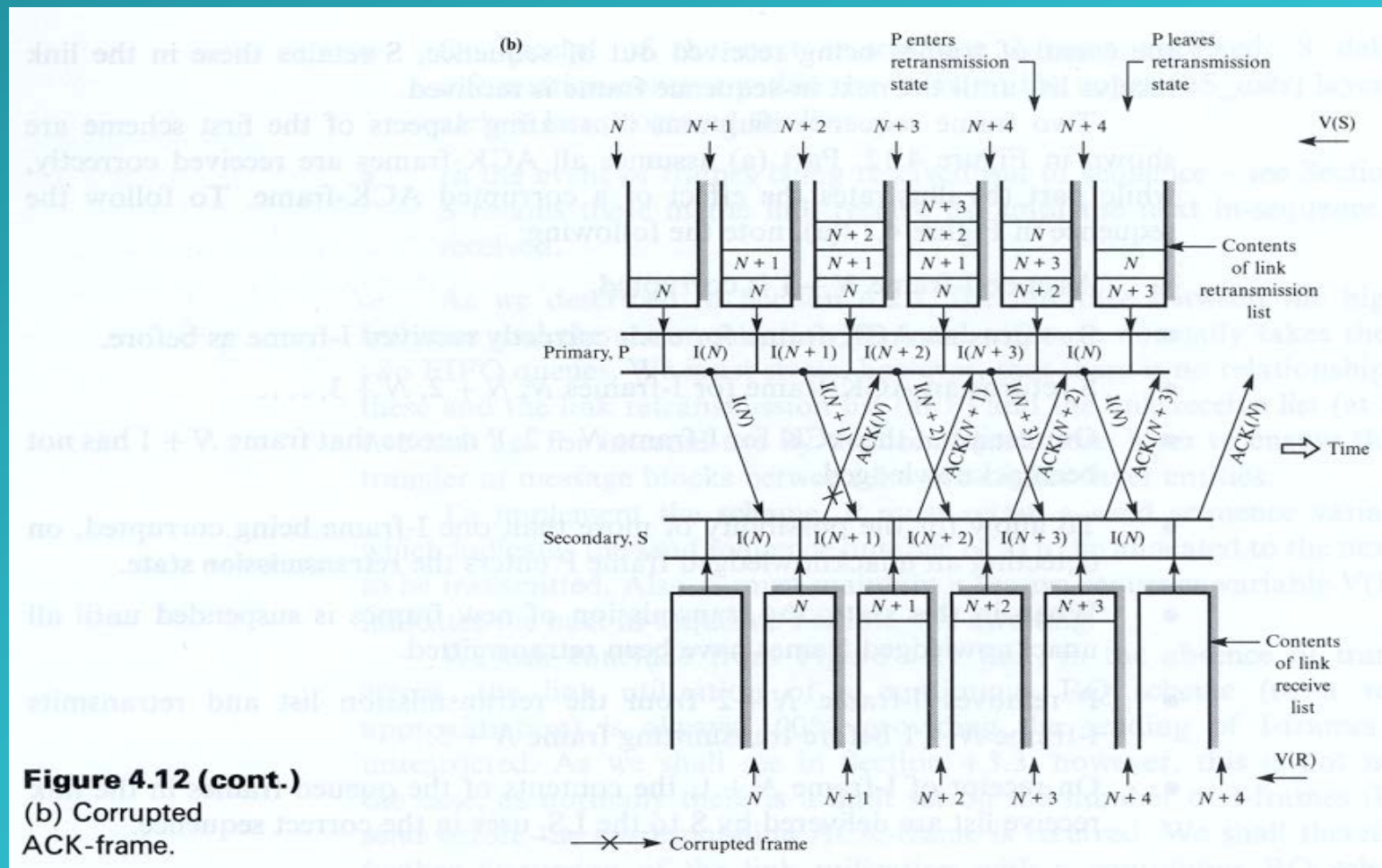
# Protocolos de Comunicações

## *Implicit Selective Repeat: trama I corrompida*



# Protocolos de Comunicações

## *Implicit Selective Repeat: ACK corrompido*

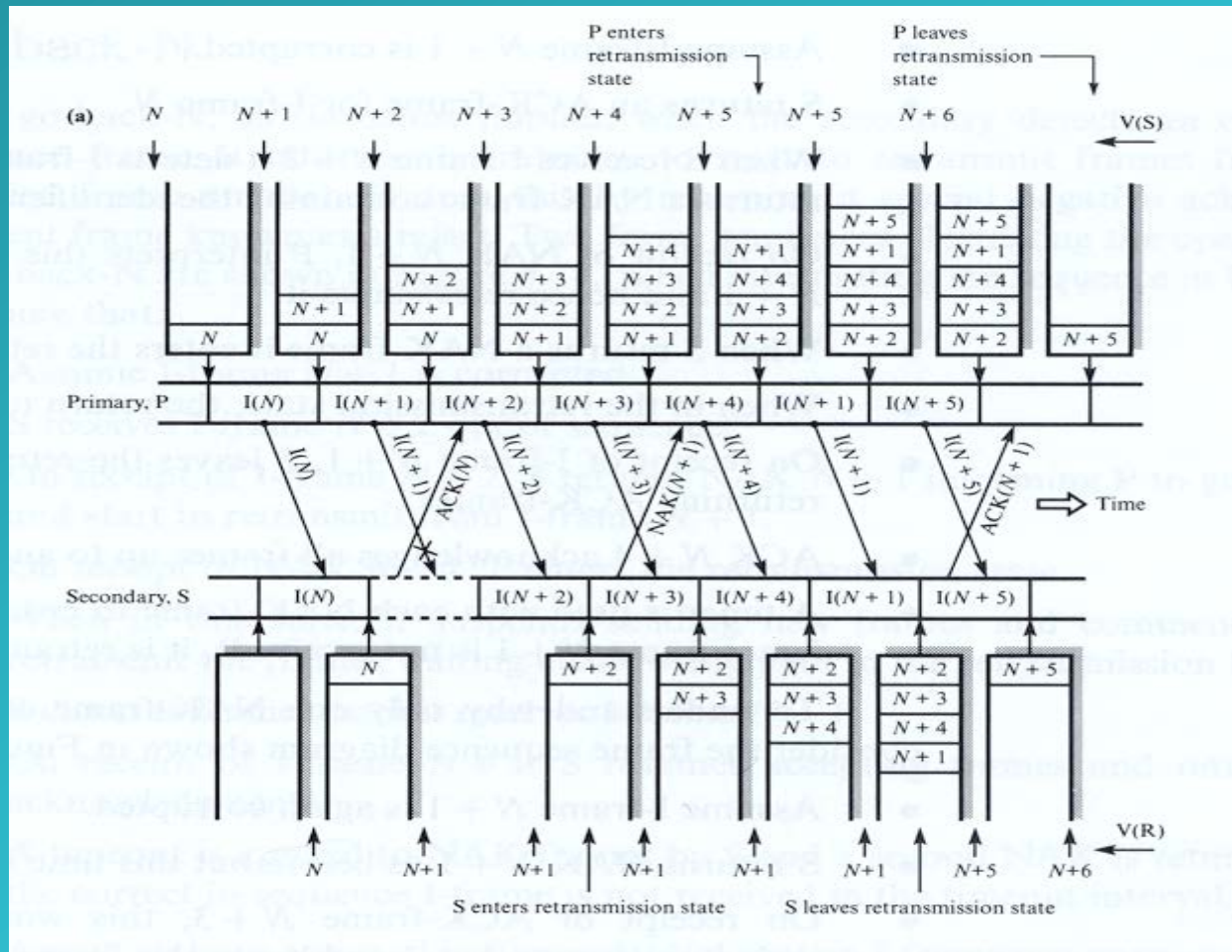


**Figure 4.12 (cont.)**  
 (b) Corrupted ACK-frame.



# Protocolos de Comunicações

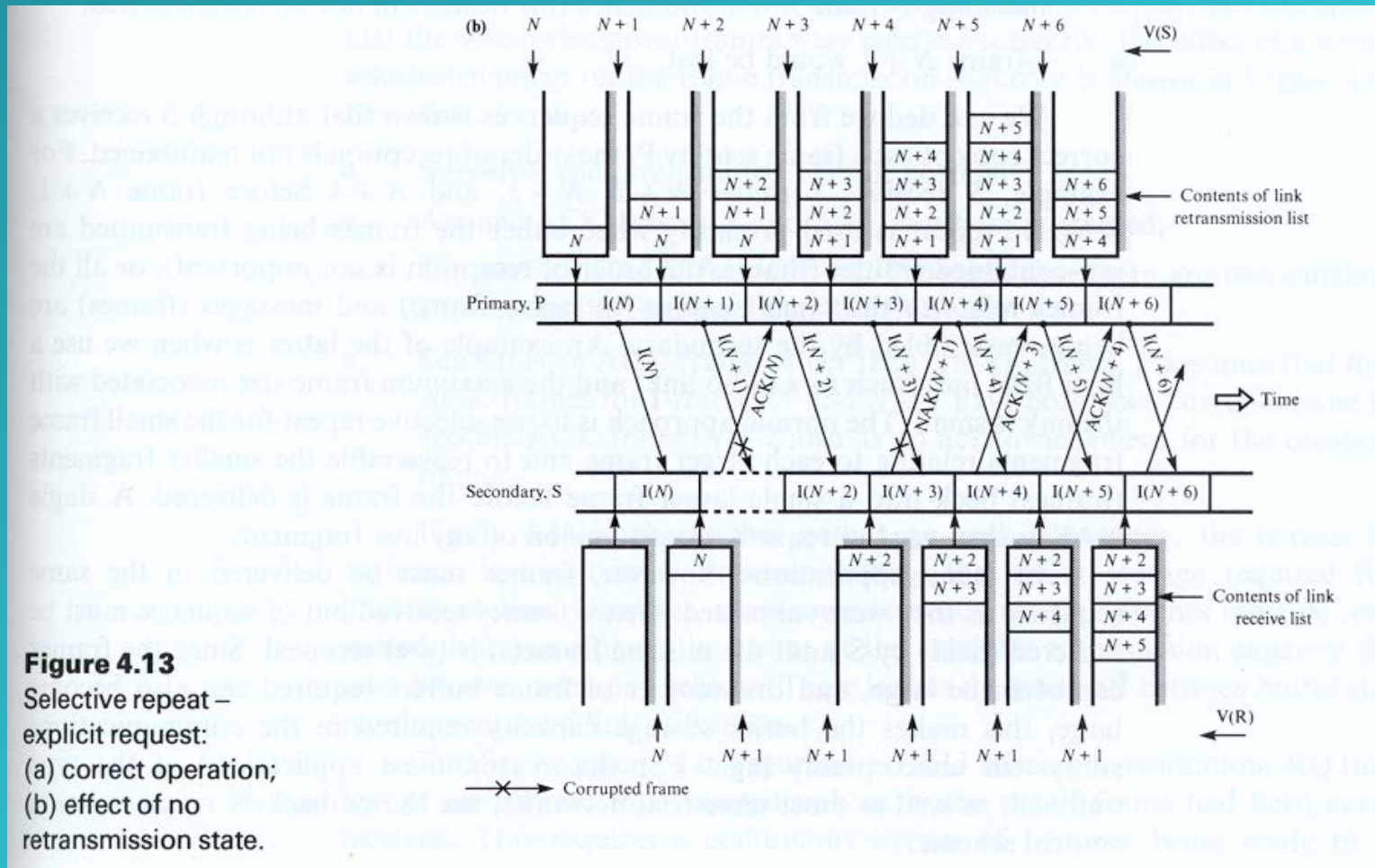
## *Explicit Selective Repeat: trama I corrompida*





# Protocolos de Comunicações

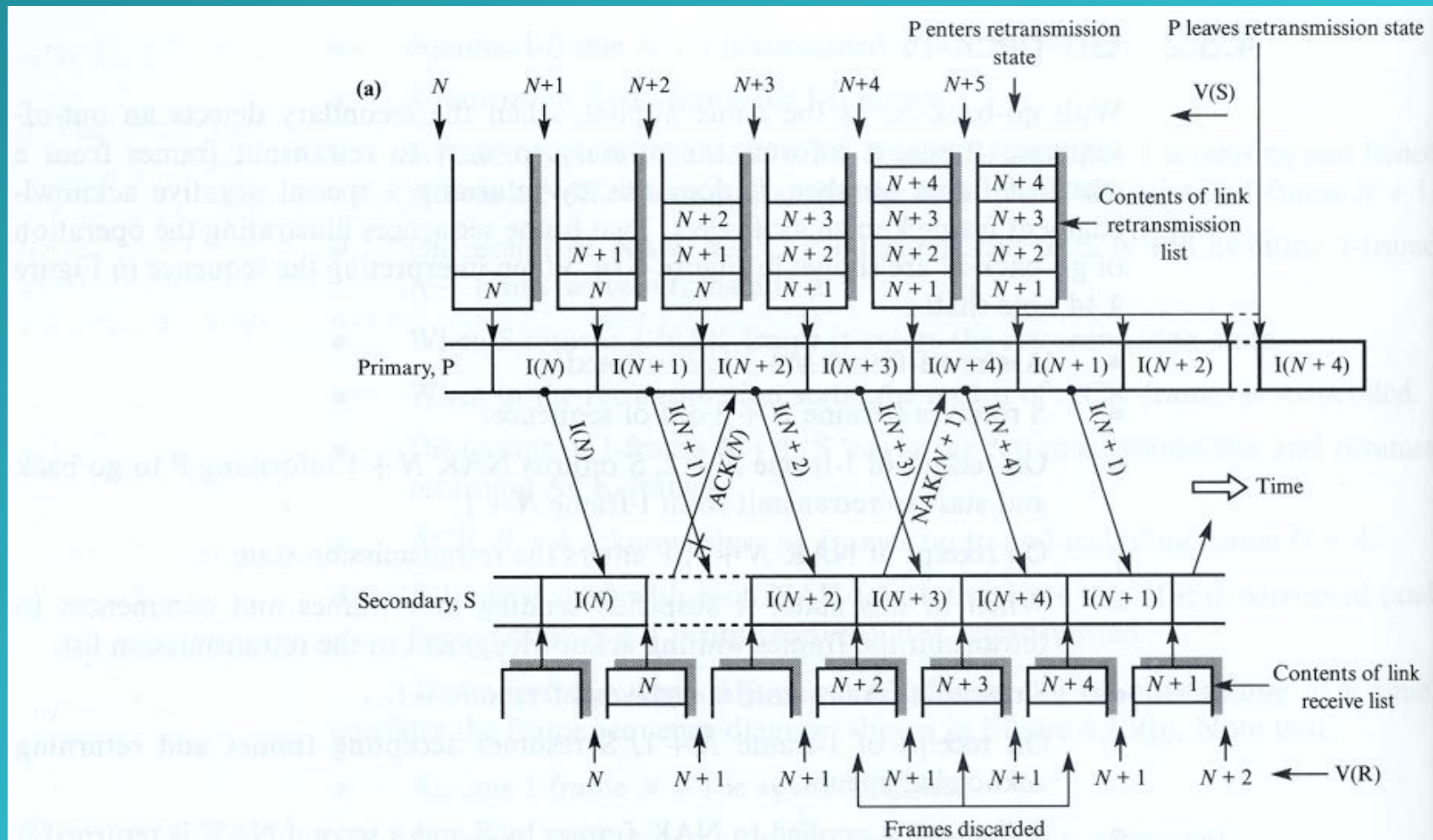
## *Explicit Selective Repeat: trama I corrompida e NAK corrompido*



**Figure 4.13**  
 Selective repeat –  
 explicit request:  
 (a) correct operation;  
 (b) effect of no  
 retransmission state.

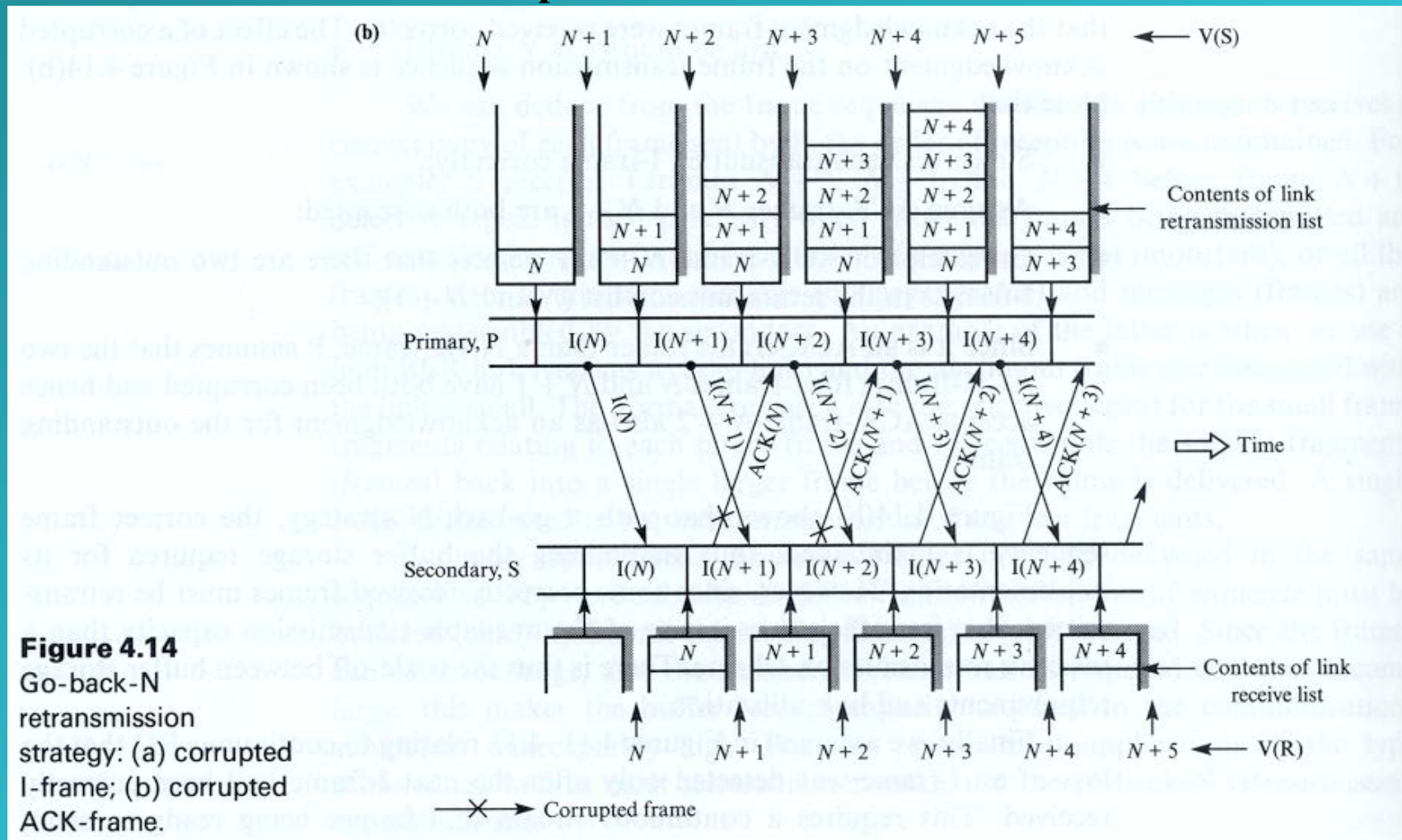
# Protocolos de Comunicações

## Go-back-N: trama I corrompida



# Protocolos de Comunicações

## Go-back-N: ACK corrompido



**Figure 4.14**  
Go-back-N  
retransmission  
strategy: (a) corrupted  
I-frame; (b) corrupted  
ACK-frame.

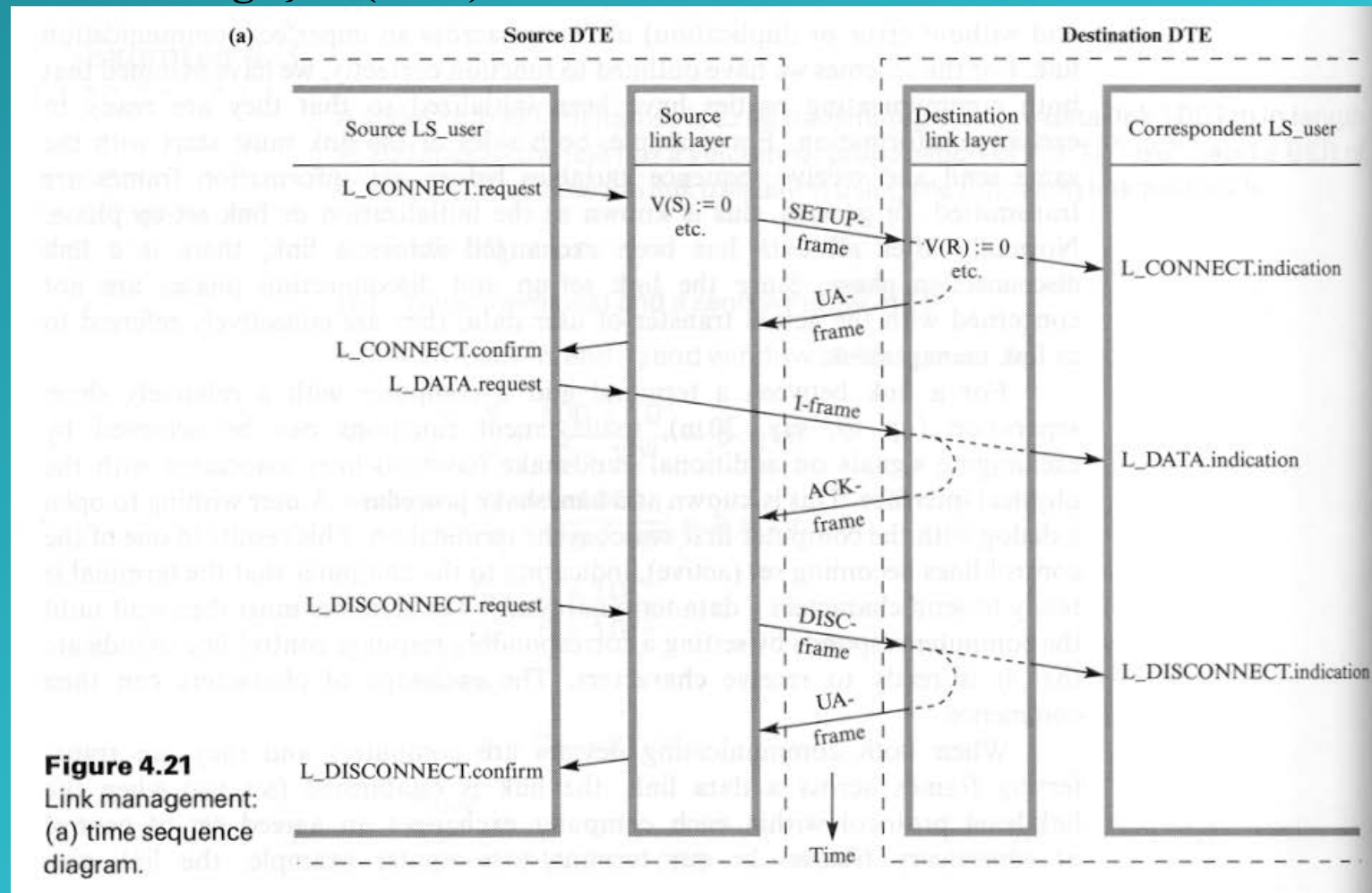






# Protocolos de Comunicações

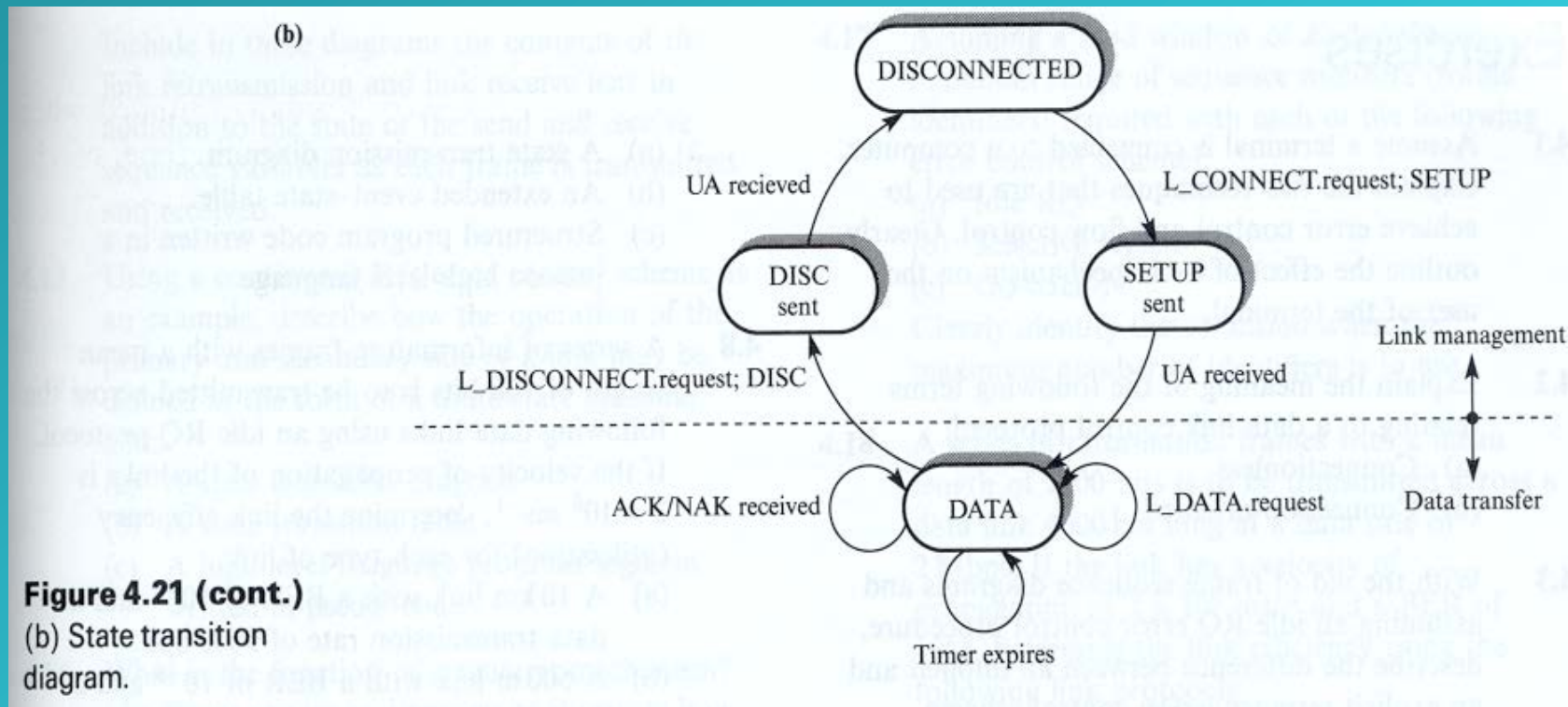
## Gestão da Ligação (*Link*)



# Protocolos de Comunicações

## Gestão da Ligação (*Link*)

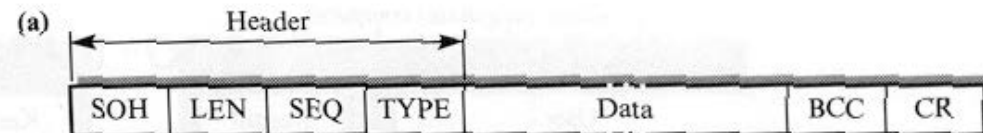
máquina de estados associada



## Protocolos de Comunicações

### Exemplo de protocolo para transferência de ficheiros

#### KERMIT



SOH = Marks start of a frame

LEN = The number of characters/bytes in the frame following this character up to and including BCC. The number is encoded in excess-32 notation using a single character in the range from ASCII # (decimal 35) to ASCII ~ (decimal 126) where # indicates a length of 3 (no data) and ~ indicates the maximum length of 91

SEQ = The send sequence number of the frame. The sequence number is incremented modulo-64 and is again encoded as a single character in the range ASCII SP (decimal 32) for zero to ASCII \_ (decimal 95) for 63

TYPE = The type of frame encoded using a single character:

S = Send initiation (parameters)

F = Filename

D = File data

Z = End of file

B = End of transaction

Y = Acknowledgment (ACK)

N = Negative acknowledgment (NAK)

E = Fatal error

Data = Frame contents

BCC = Block check character

CR = End of block marker (ASCII carriage return)

**Figure 5.4**

Kermit operation:

(a) frame format and types.

# KERMIT

Exemplo de envio de um ficheiro.

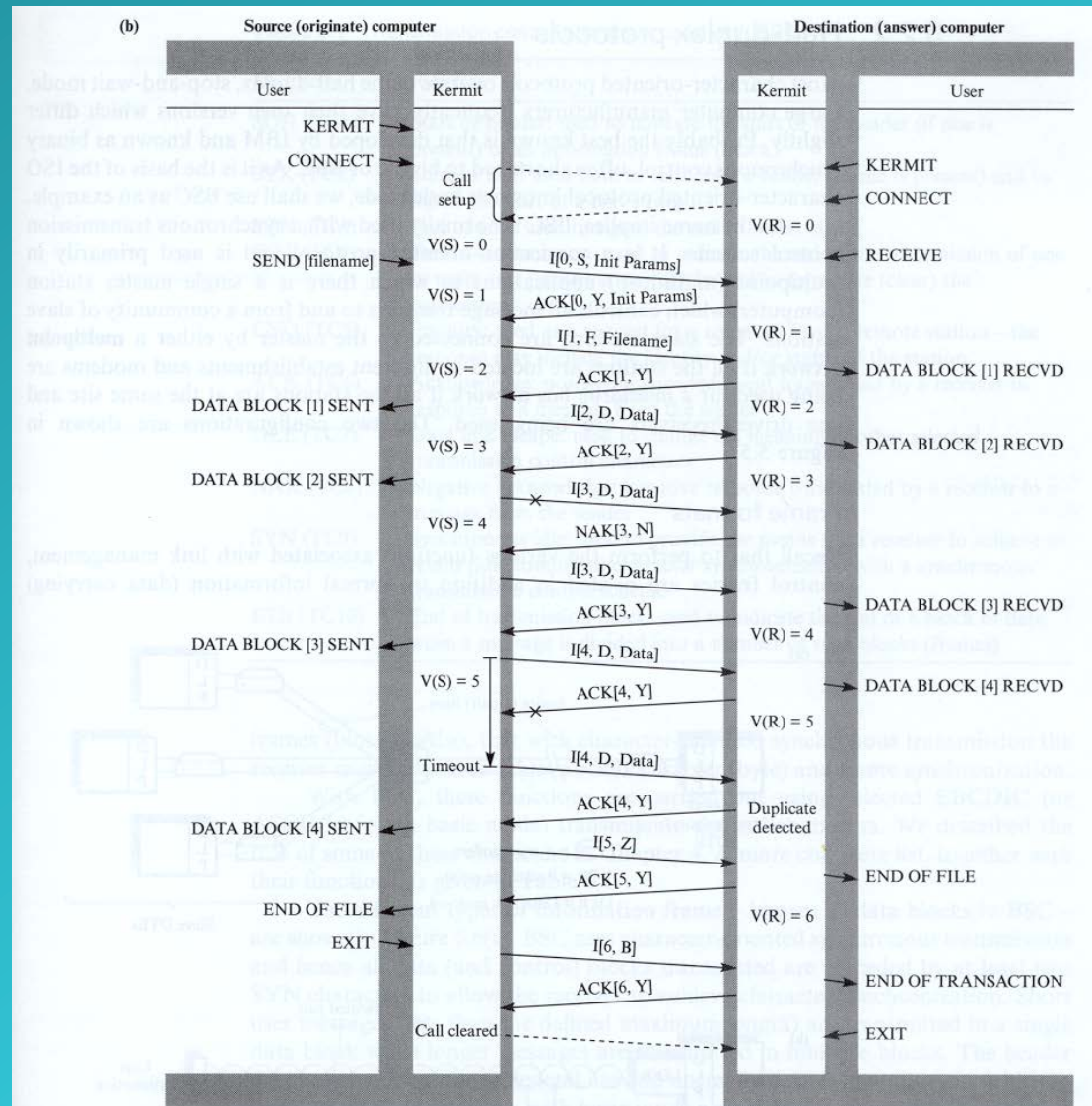
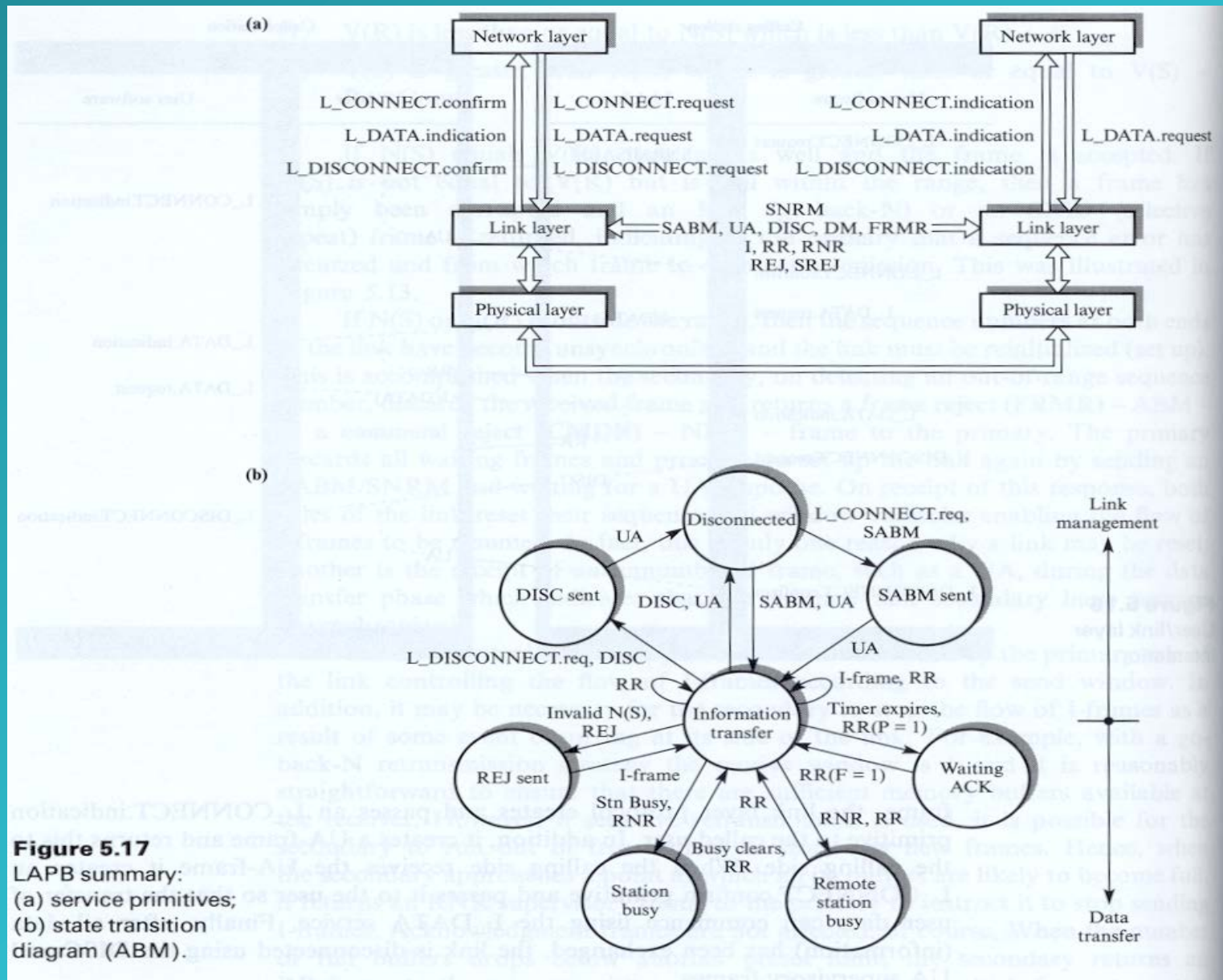


Figure 5.4 (cont.)  
(b) Frame sequences.

sequence number in each I-frame increments modulo 64 and the receive sequence number in each ACK (Y) and NAK (N) frame carries the same sequence number



LAPB



# Automação Industrial baseada em PLCs

## 15ª Aula

<http://www.isr.ist.utl.pt/~pjcro/aibplc.html>

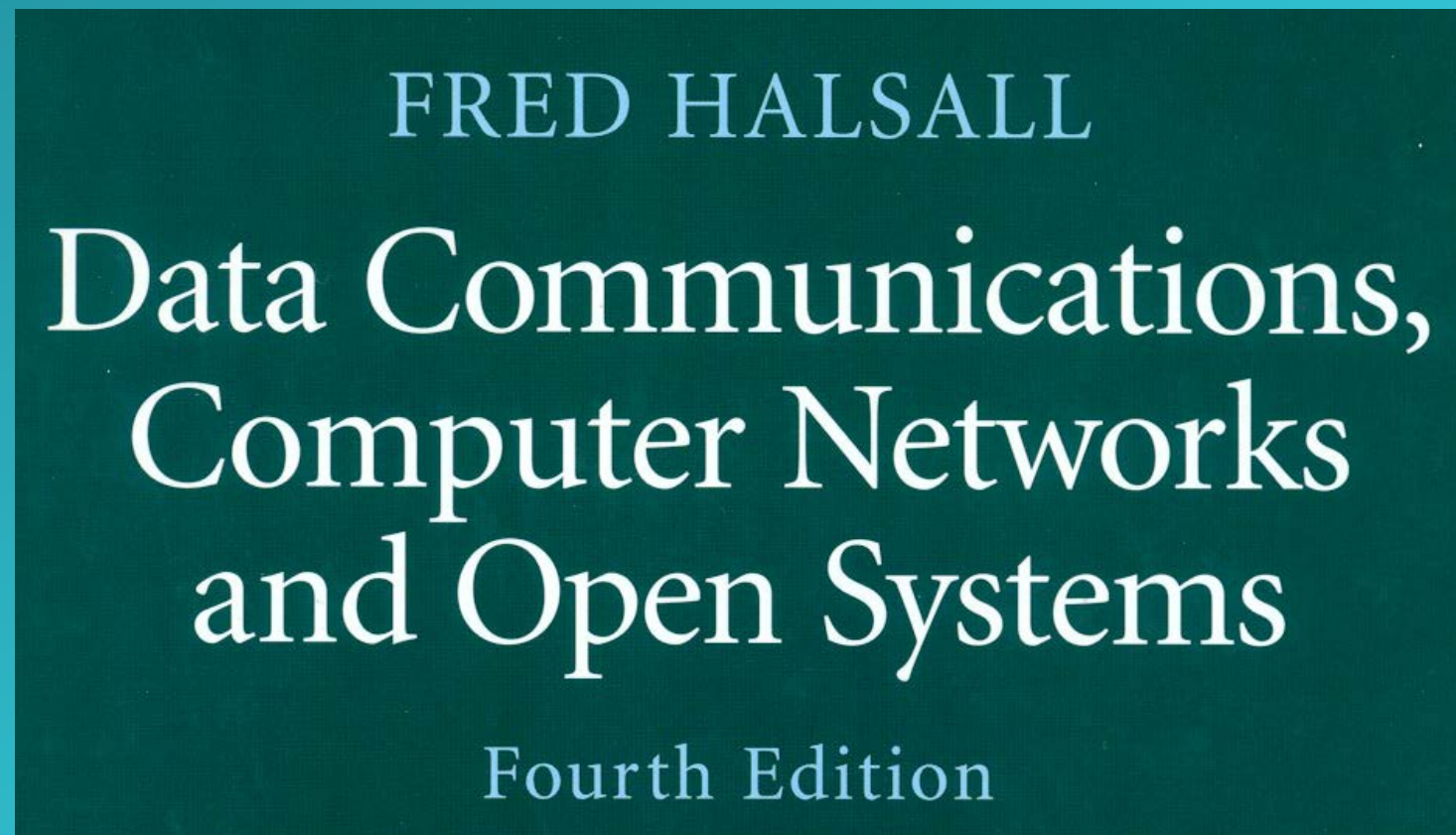
Docente: Paulo Jorge Oliveira

e-mail: [pjcro@isr.ist.utl.pt](mailto:pjcro@isr.ist.utl.pt)

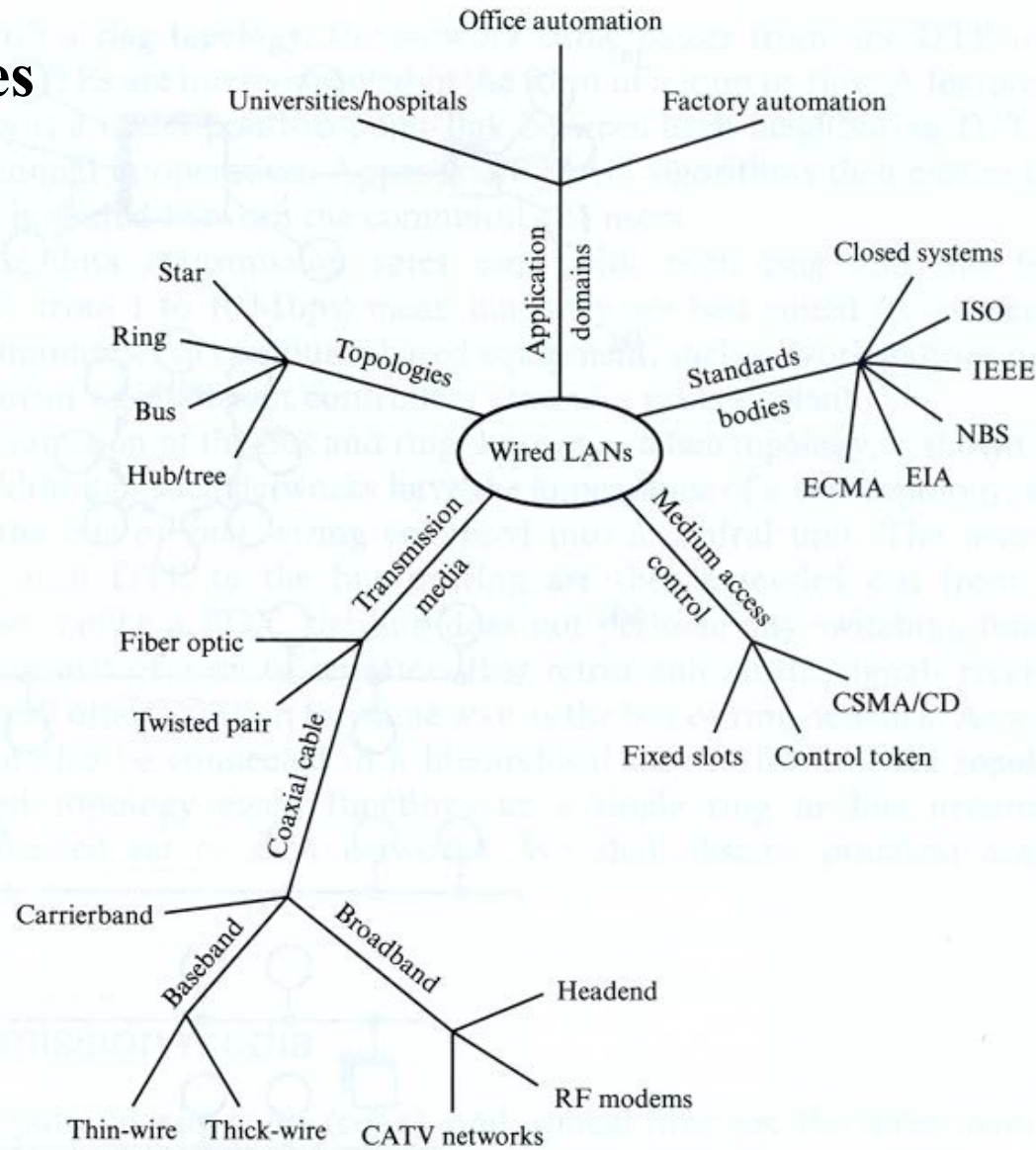
## 15ª Aula

Cap. 9 - Topologia de Redes

Cap. 9 - Protocolos de Comunicações (cont)



# Topologia de Redes

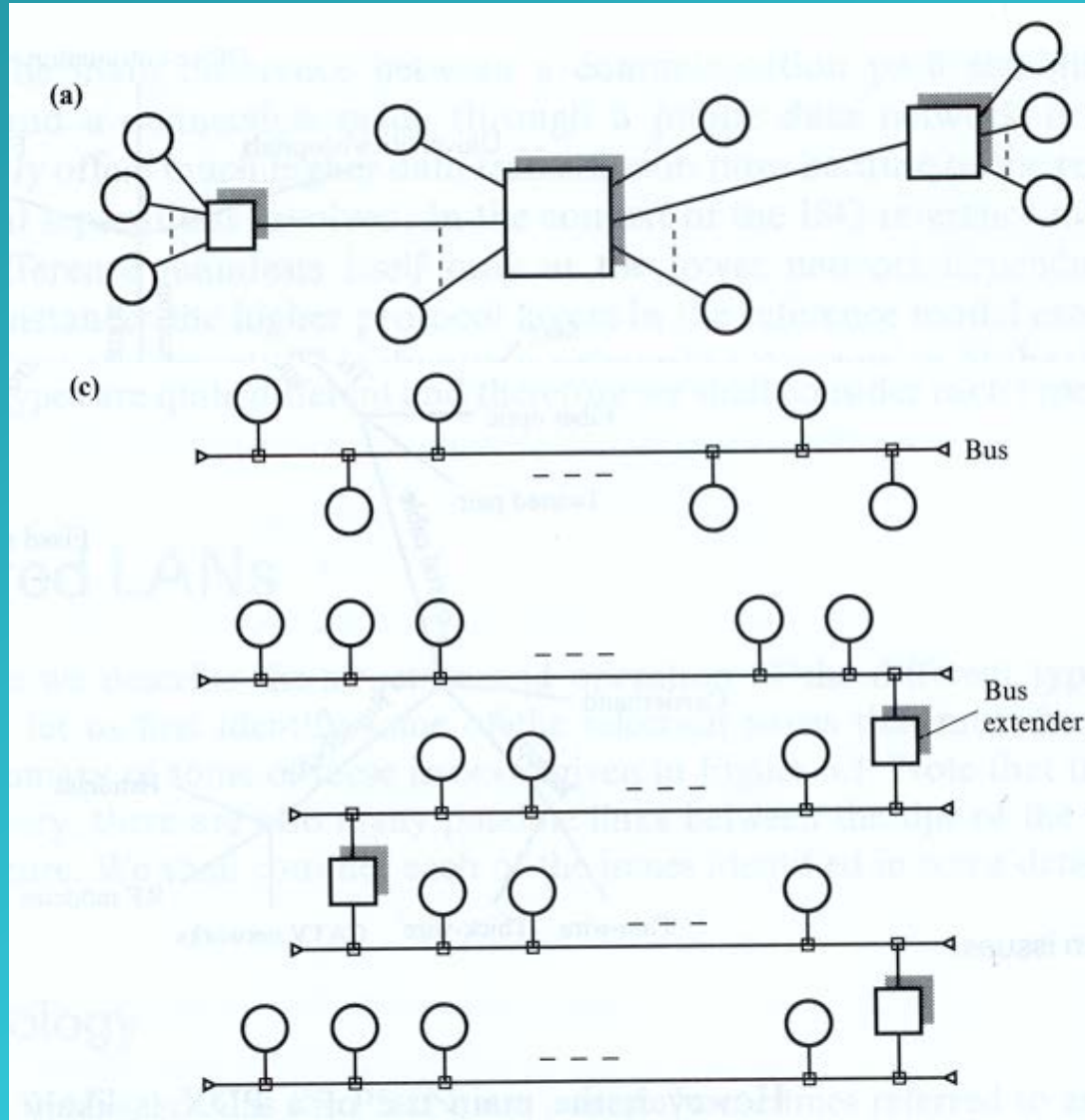


**Figure 6.1**  
LAN selection issues.



# Topologia de Redes

Estrela

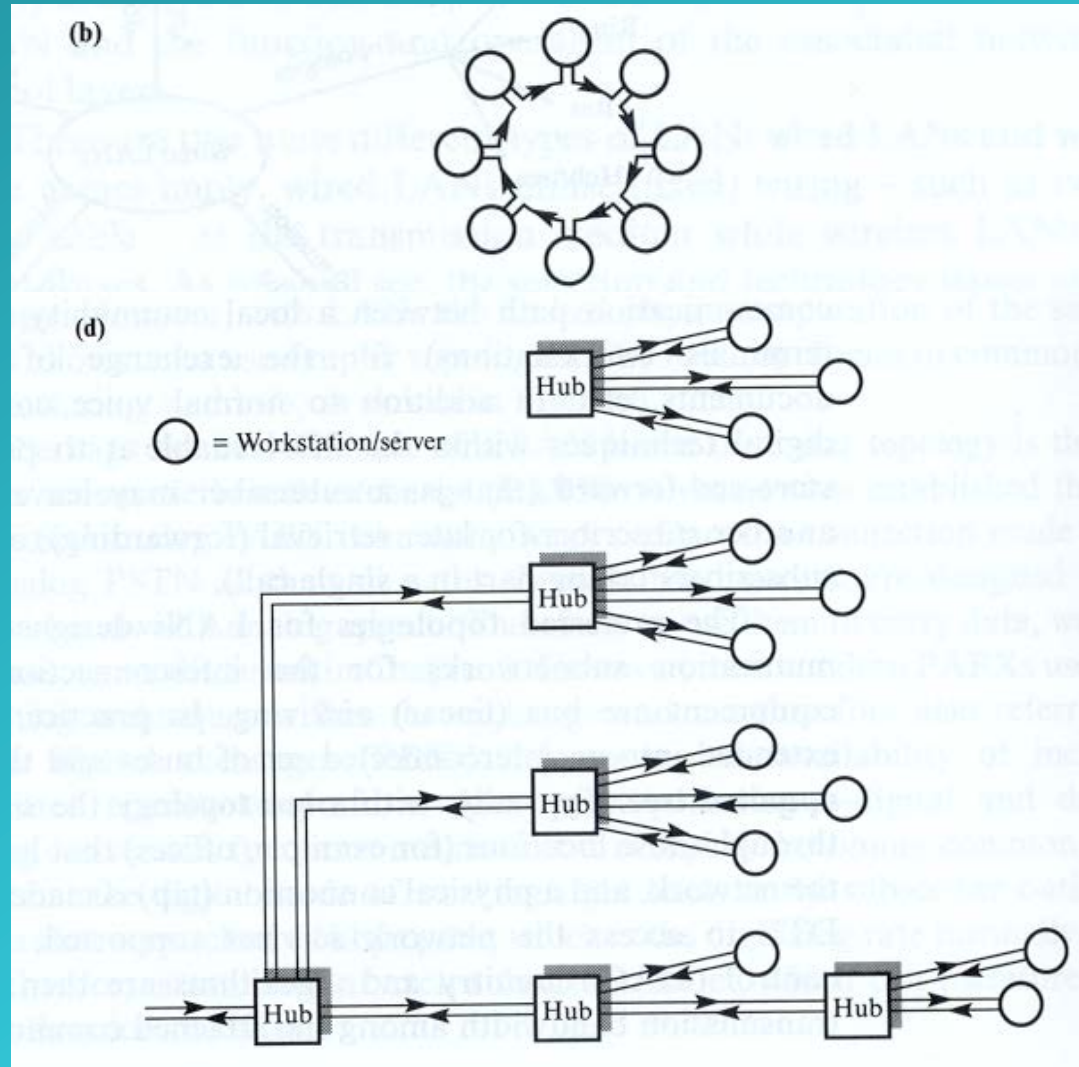


Barramento  
(bus)

# Topologia de Redes

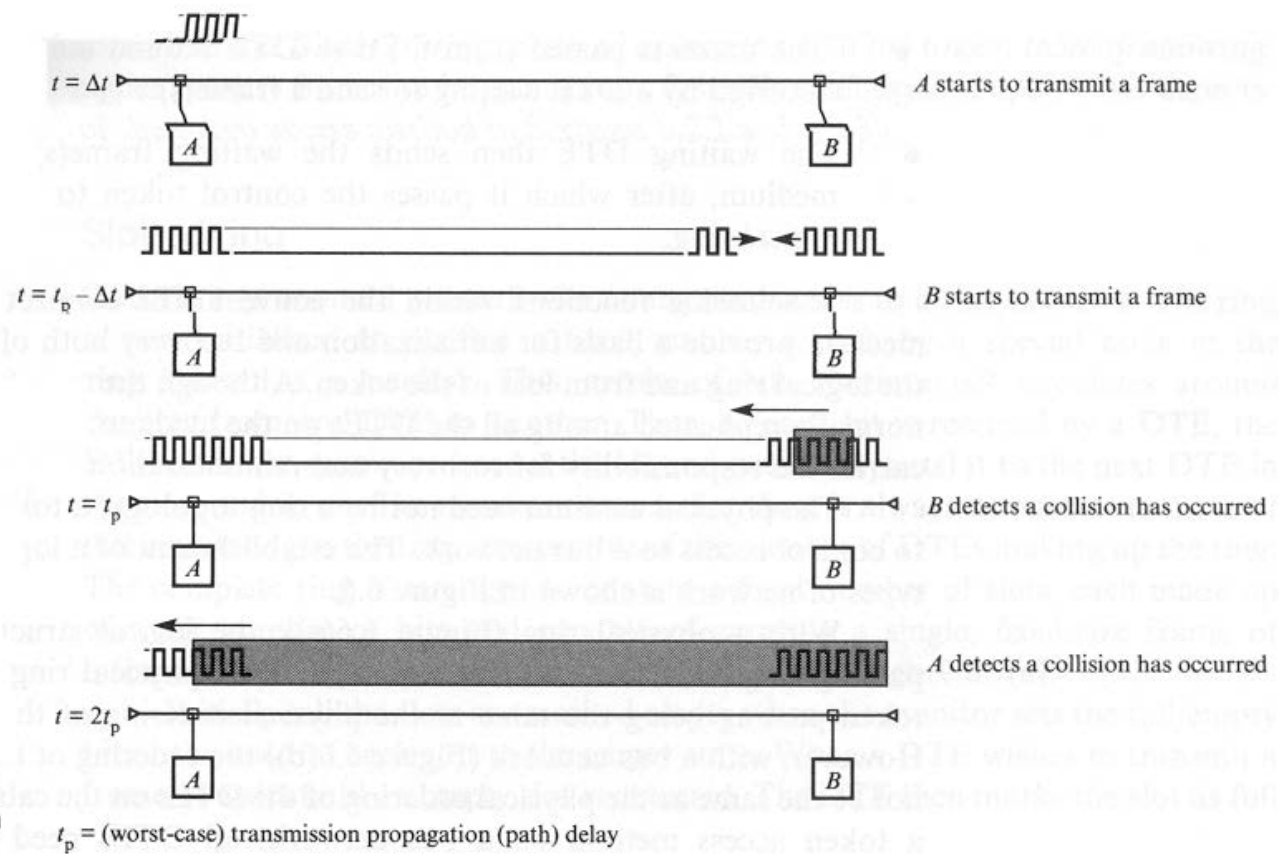
Anel  
(ring)

Hub/árvore



# Topologia de Redes

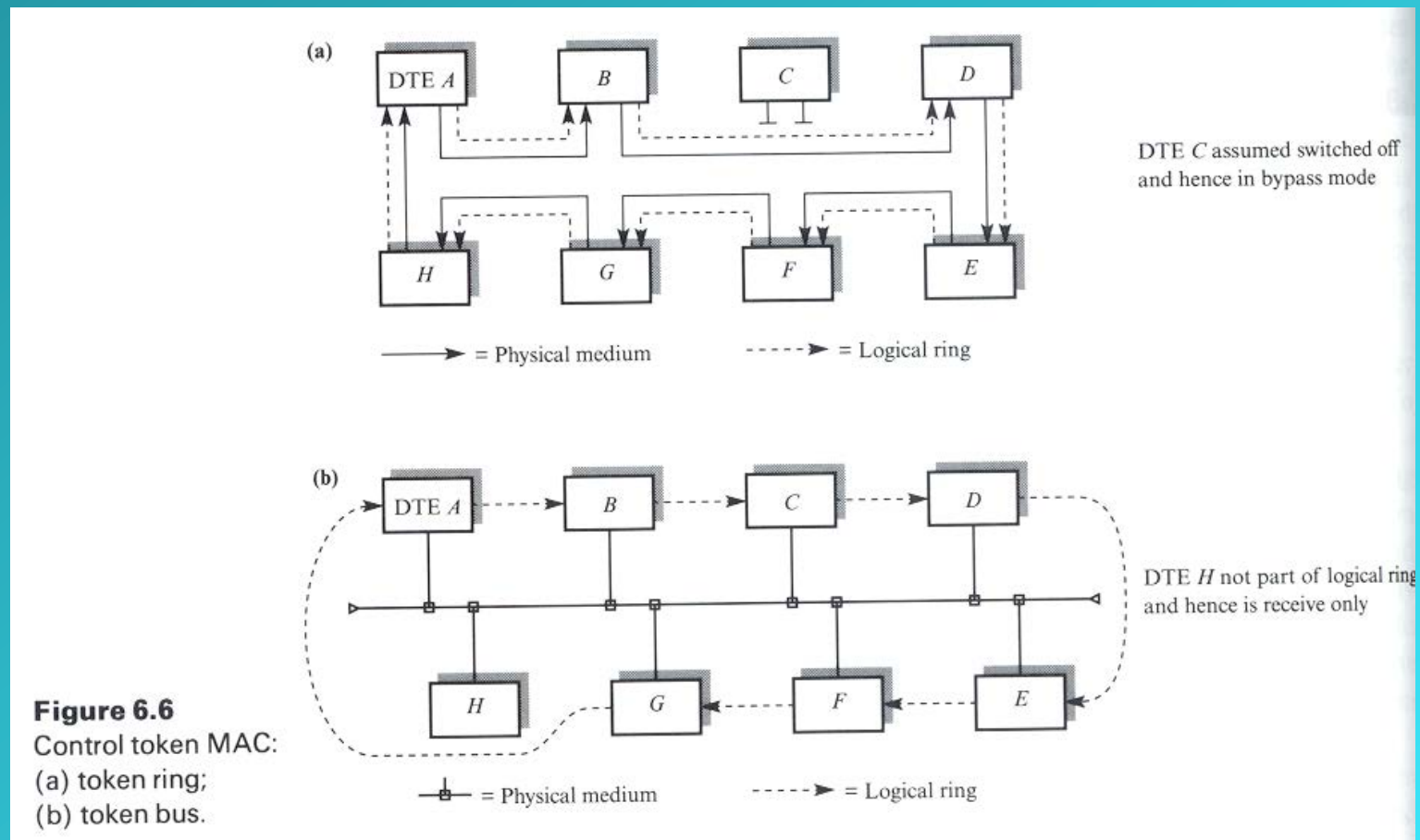
## Acesso ao meio físico CSMA/CD



**Figure 6.5**  
CSMA/CD collision  
schematic.

# Topologia de Redes

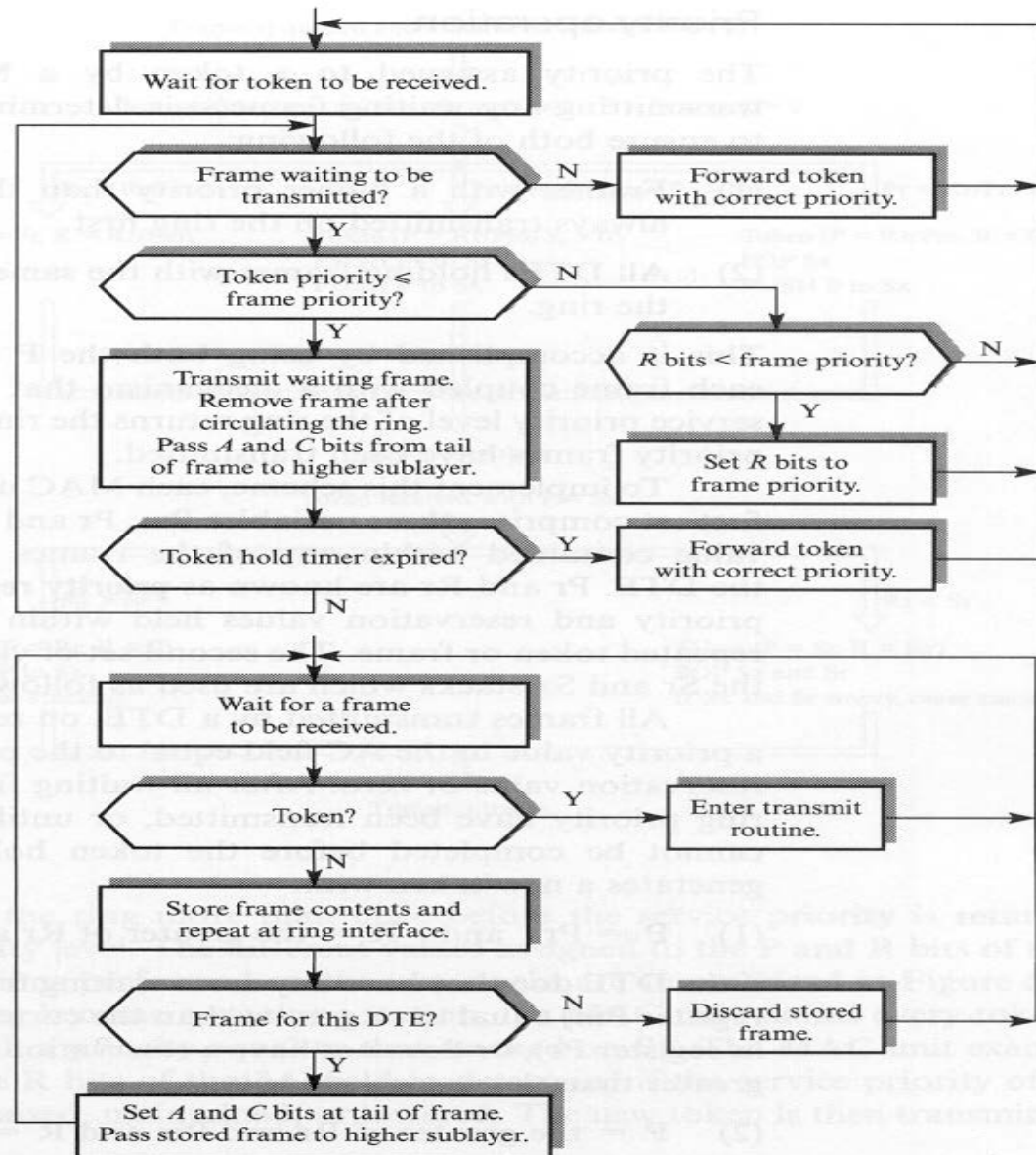
## Acesso ao meio físico *Token ring/bus*



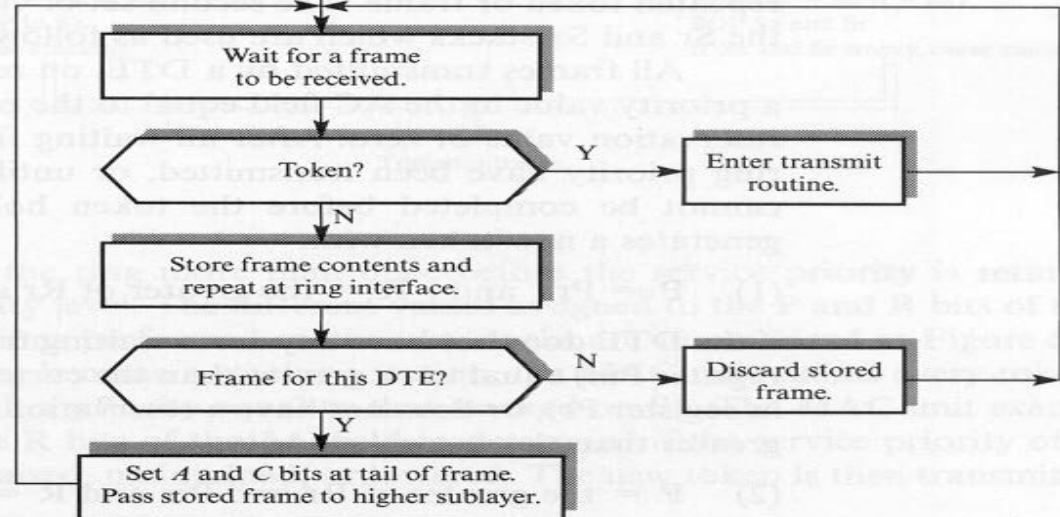


Topologia de Redes <sup>(a)</sup>

Acesso ao meio físico  
*Token ring/bus*



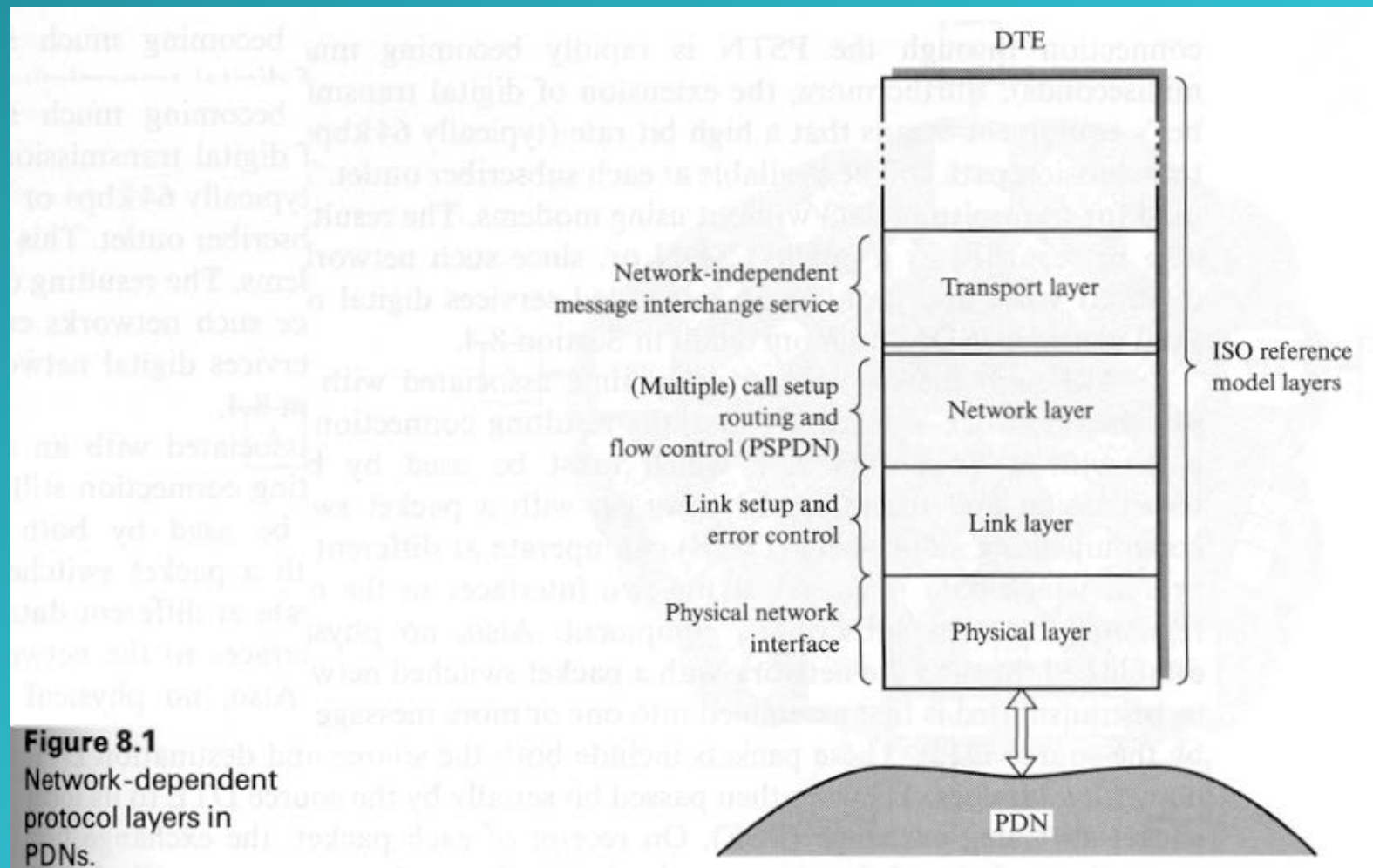
(b)



**Figure 6.15**  
Token ring MAC sublayer operation:  
(a) transmit;  
(b) receive.

# Topologia de Redes

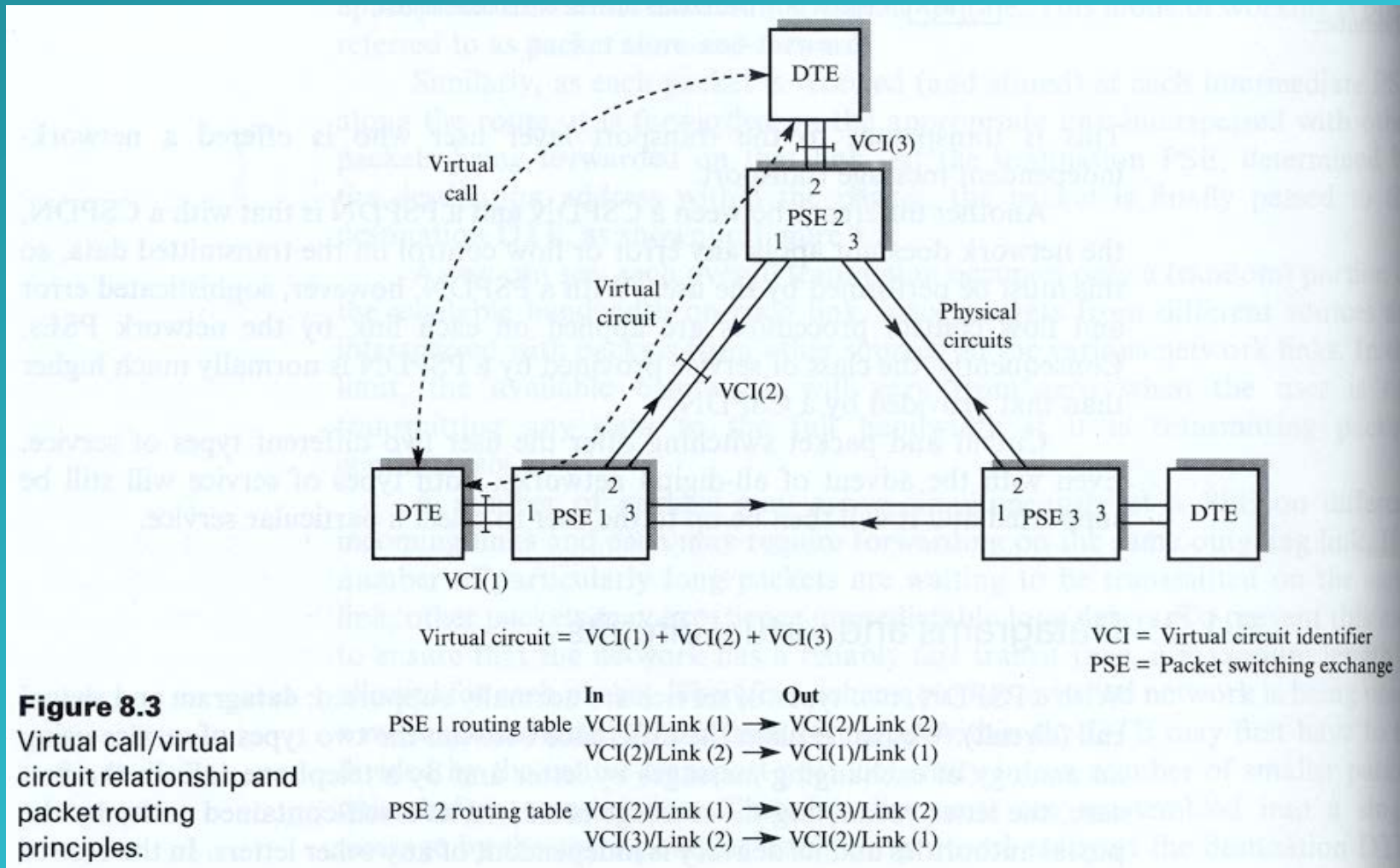
## Acesso a Redes Públicas



**Figure 8.1**  
Network-dependent  
protocol layers in  
PDNs.

# Topologia de Redes

## Acesso a Redes Públicas - chamadas virtuais

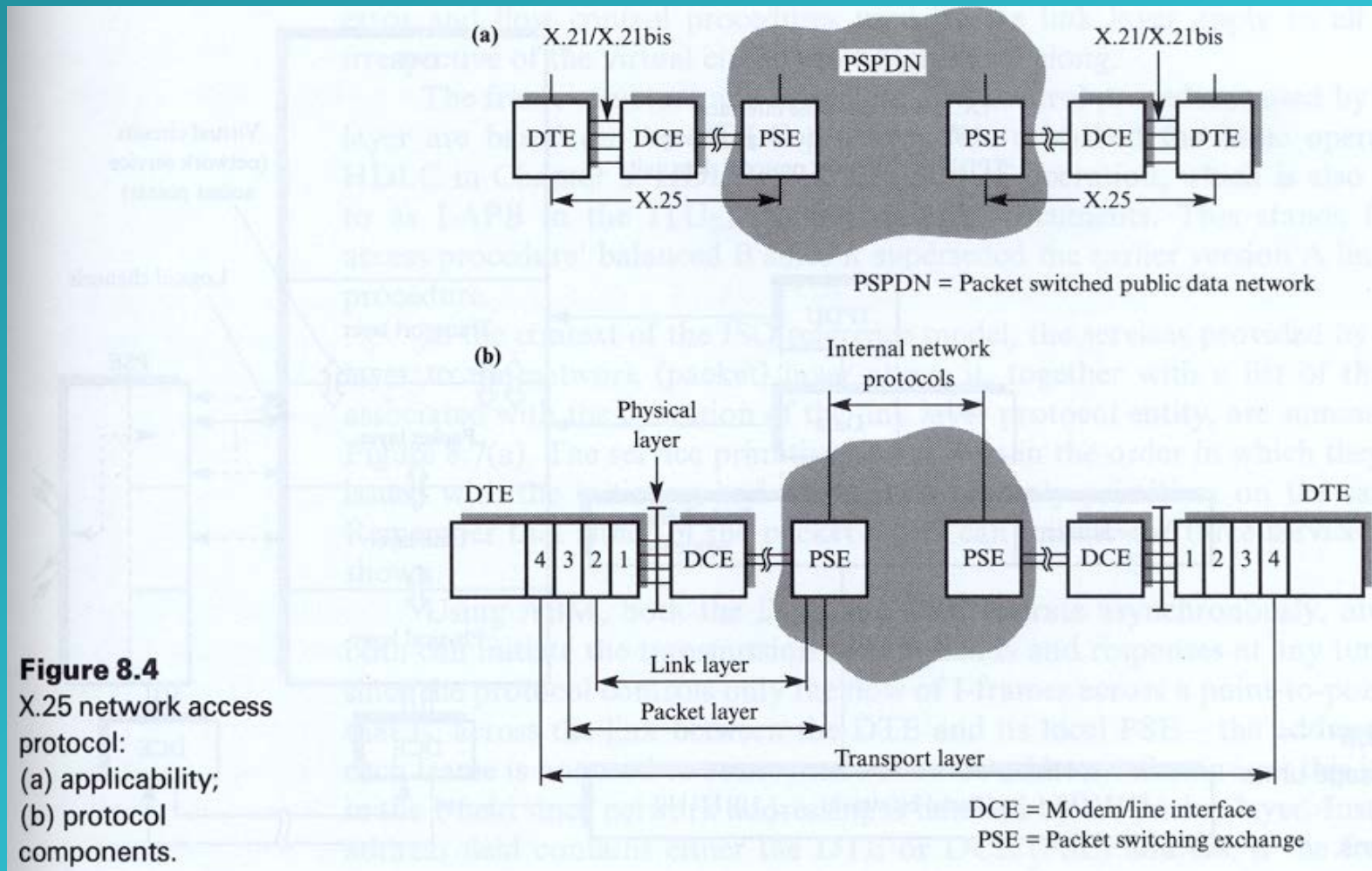


**Figure 8.3**  
Virtual call/virtual circuit relationship and packet routing principles.



# Topologia de Redes

## Acesso a Redes Públicas de Comutação de Pacotes

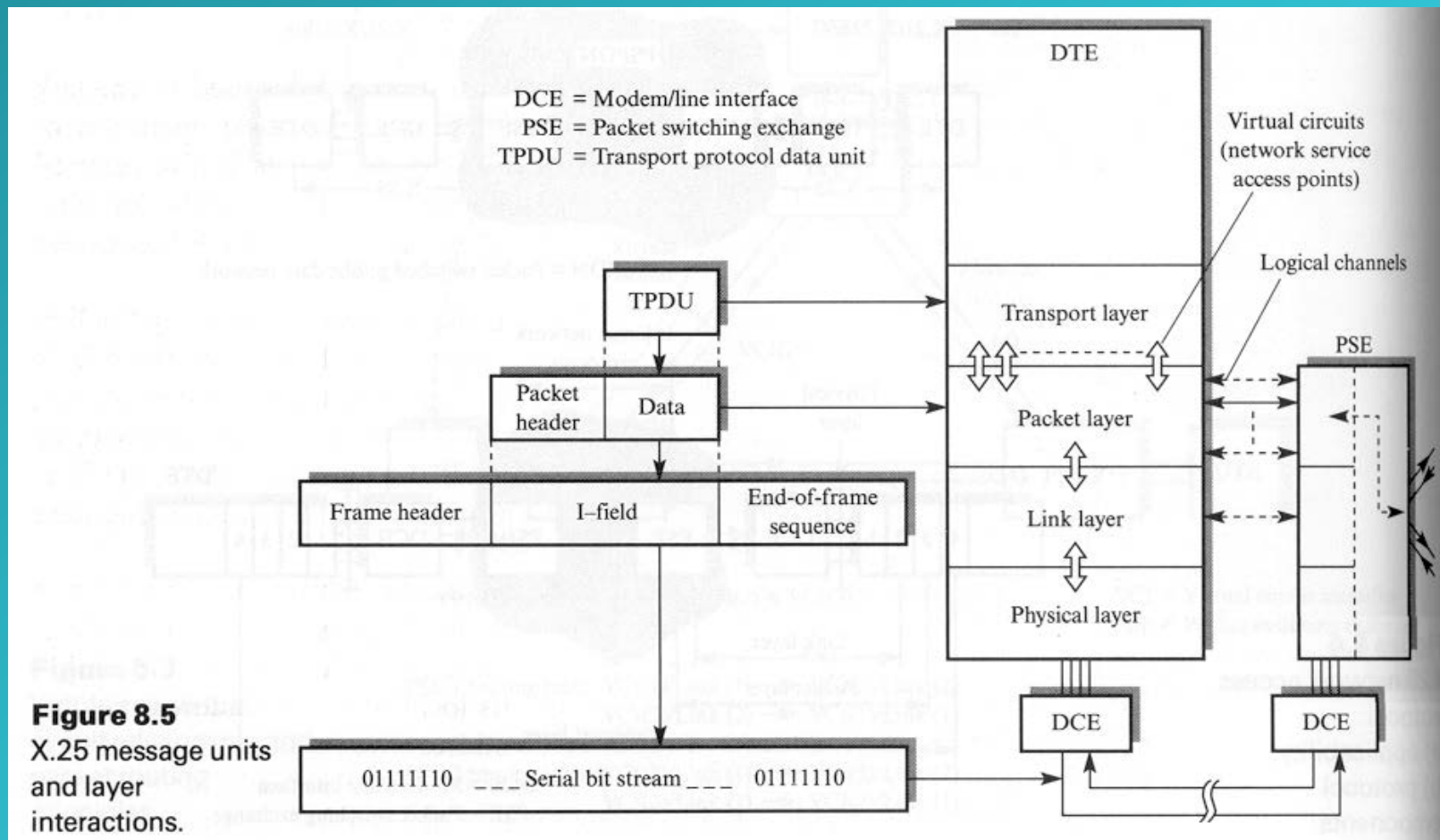


**Figure 8.4**  
 X.25 network access  
 protocol:  
 (a) applicability;  
 (b) protocol  
 components.



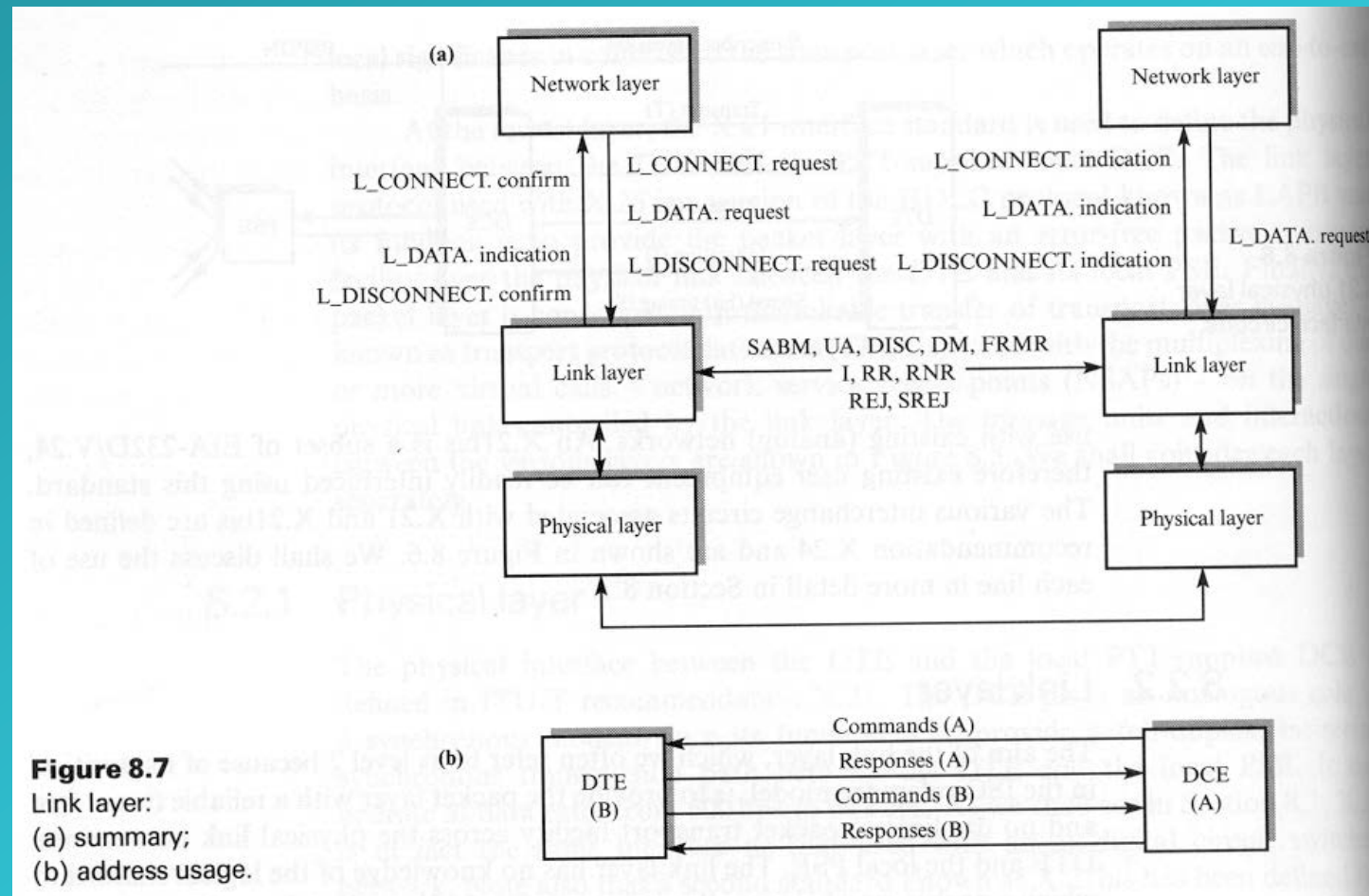
# Topologia de Redes

## Acesso a Redes Públicas de Comutação de Pacotes Nível físico



# Topologia de Redes

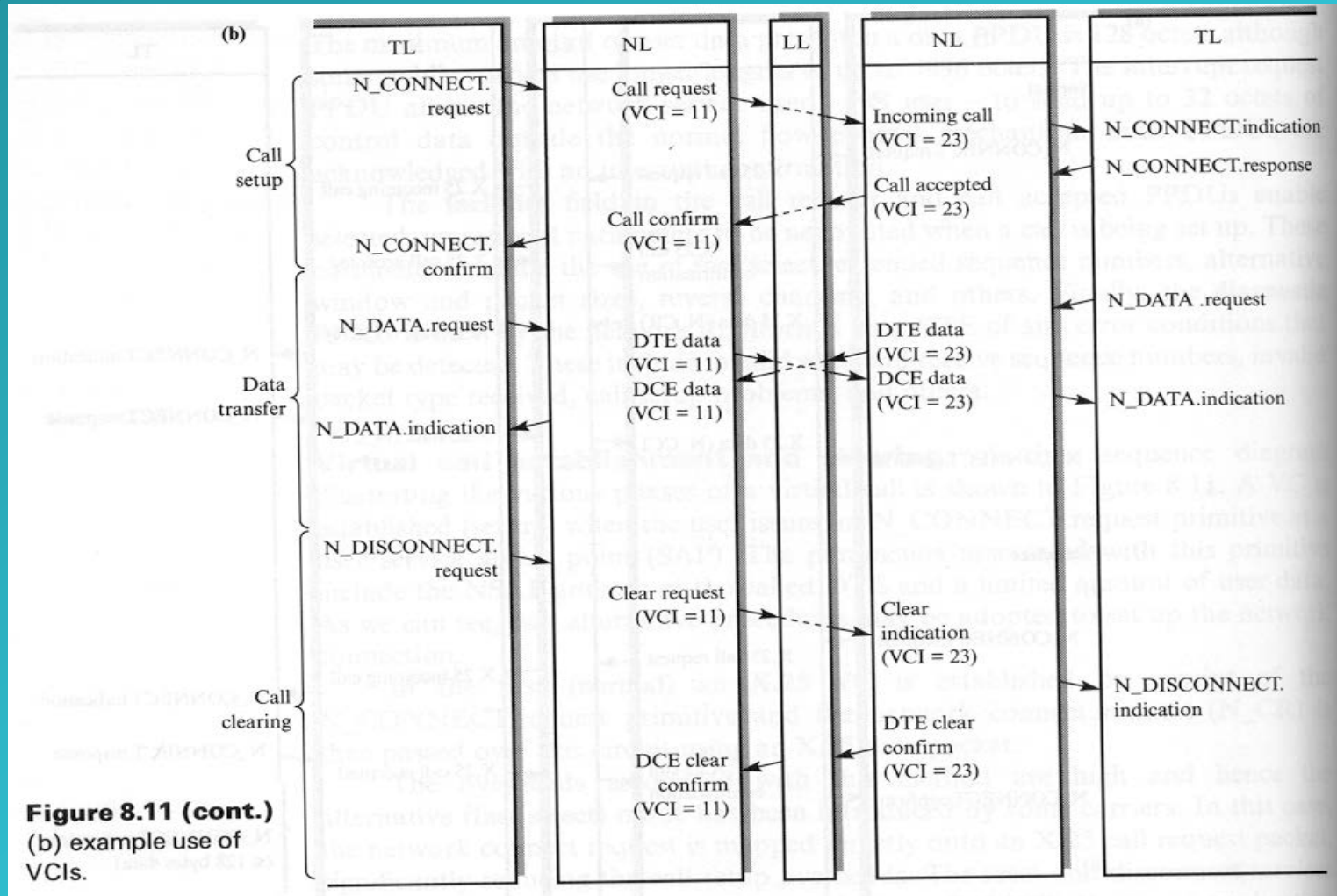
## Acesso a Redes Públicas de Comutação de Pacotes Nível de ligação



**Figure 8.7**  
Link layer:  
(a) summary;  
(b) address usage.

# Topologia de Redes

## Ligações virtuais

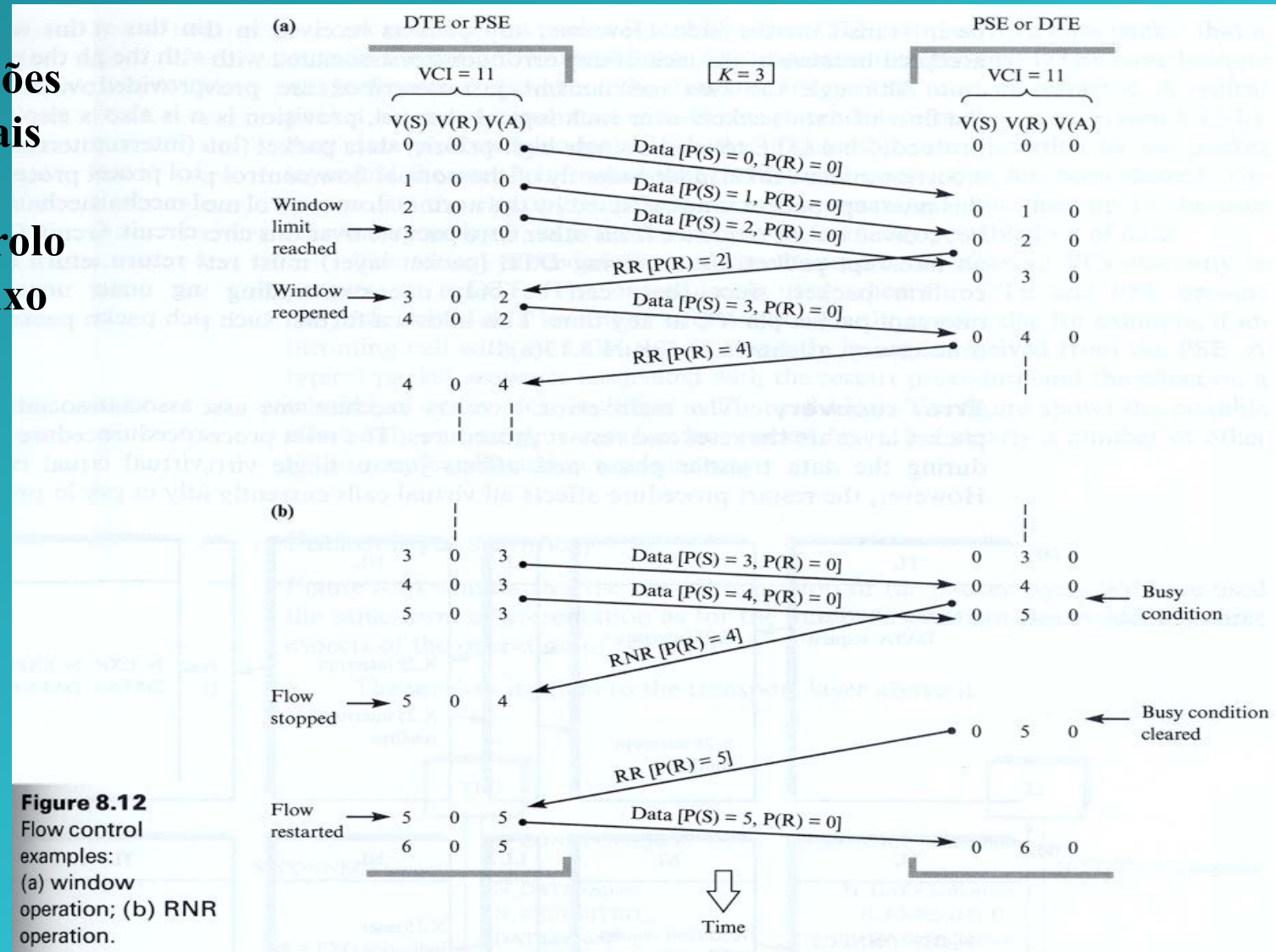


**Figure 8.11 (cont.)**  
(b) example use of VCIs.

# Protocolo IP

Ligações virtuais

Controlo de fluxo

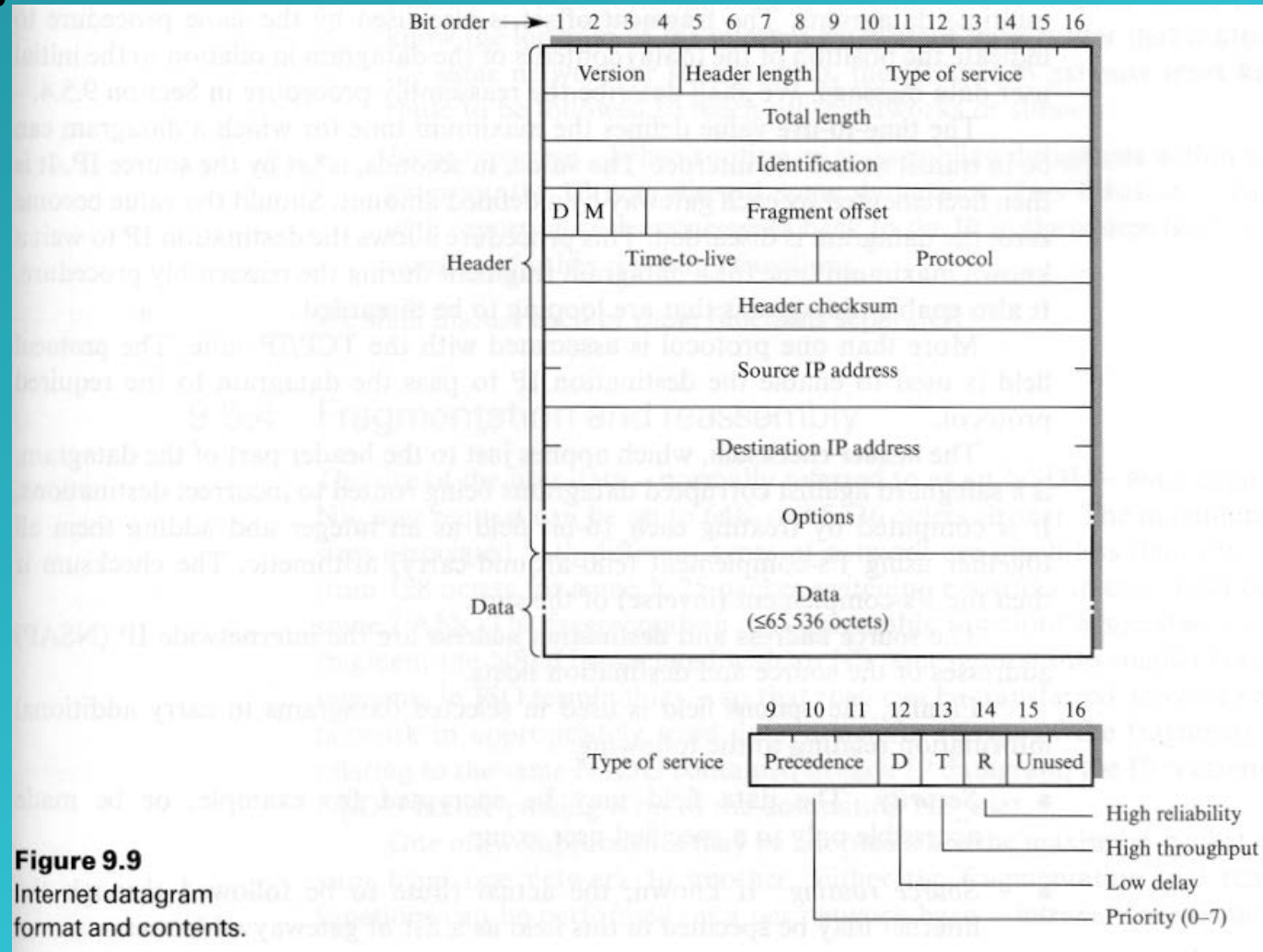


**Figure 8.12**  
Flow control examples:  
(a) window operation; (b) RNR operation.



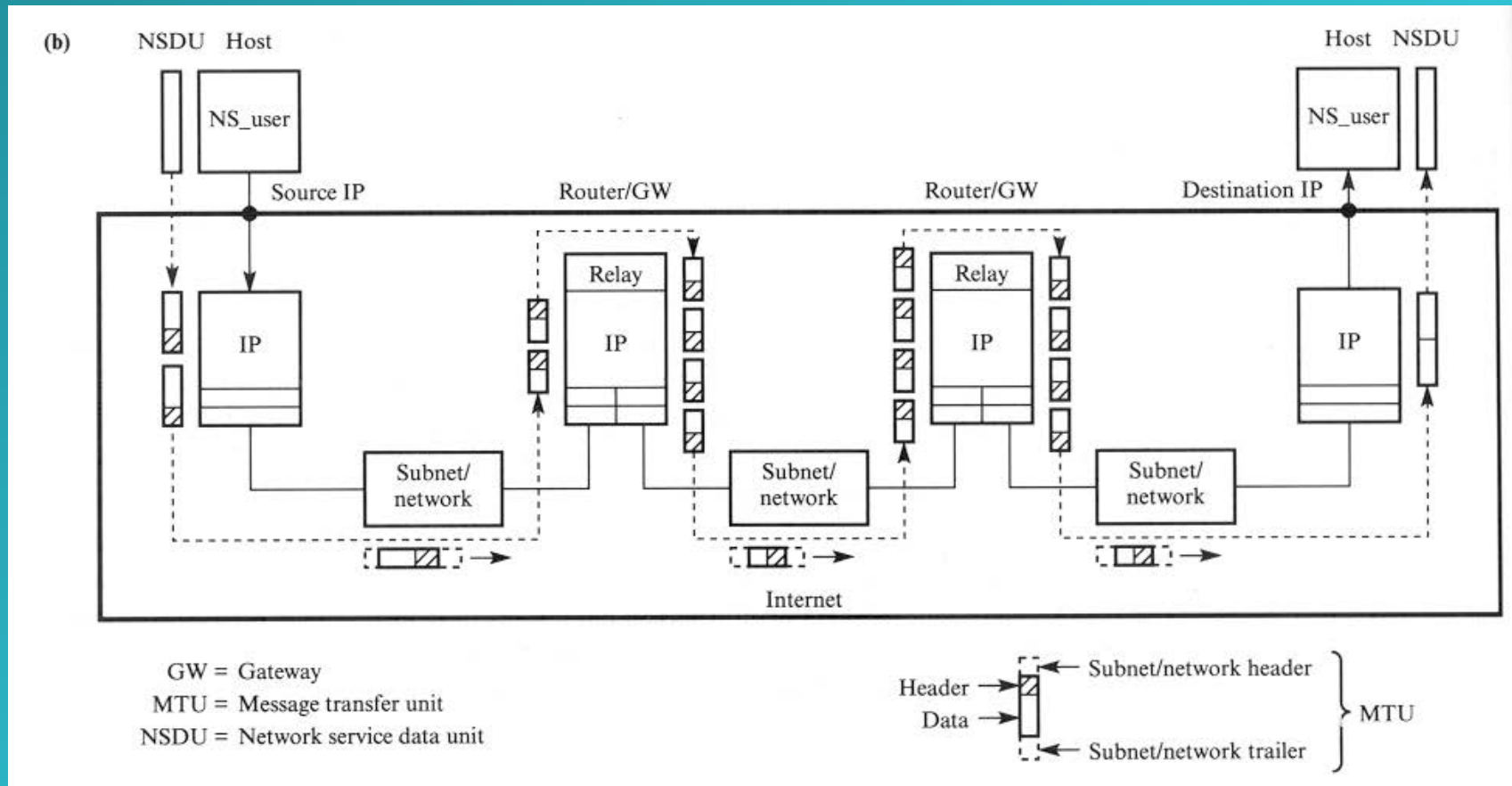
# Protocolo IP

## Datagramas



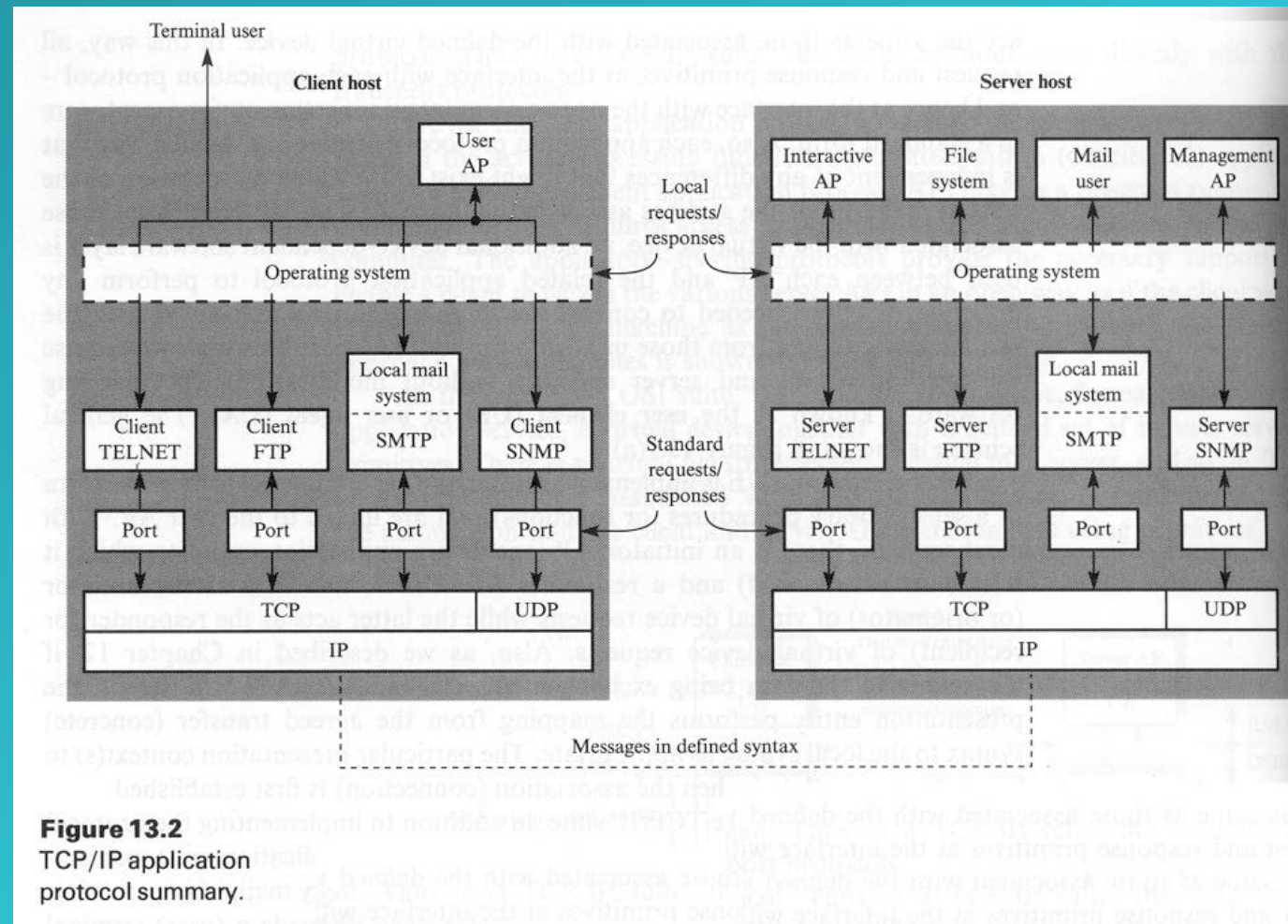
# Protocolo IP

## Routing



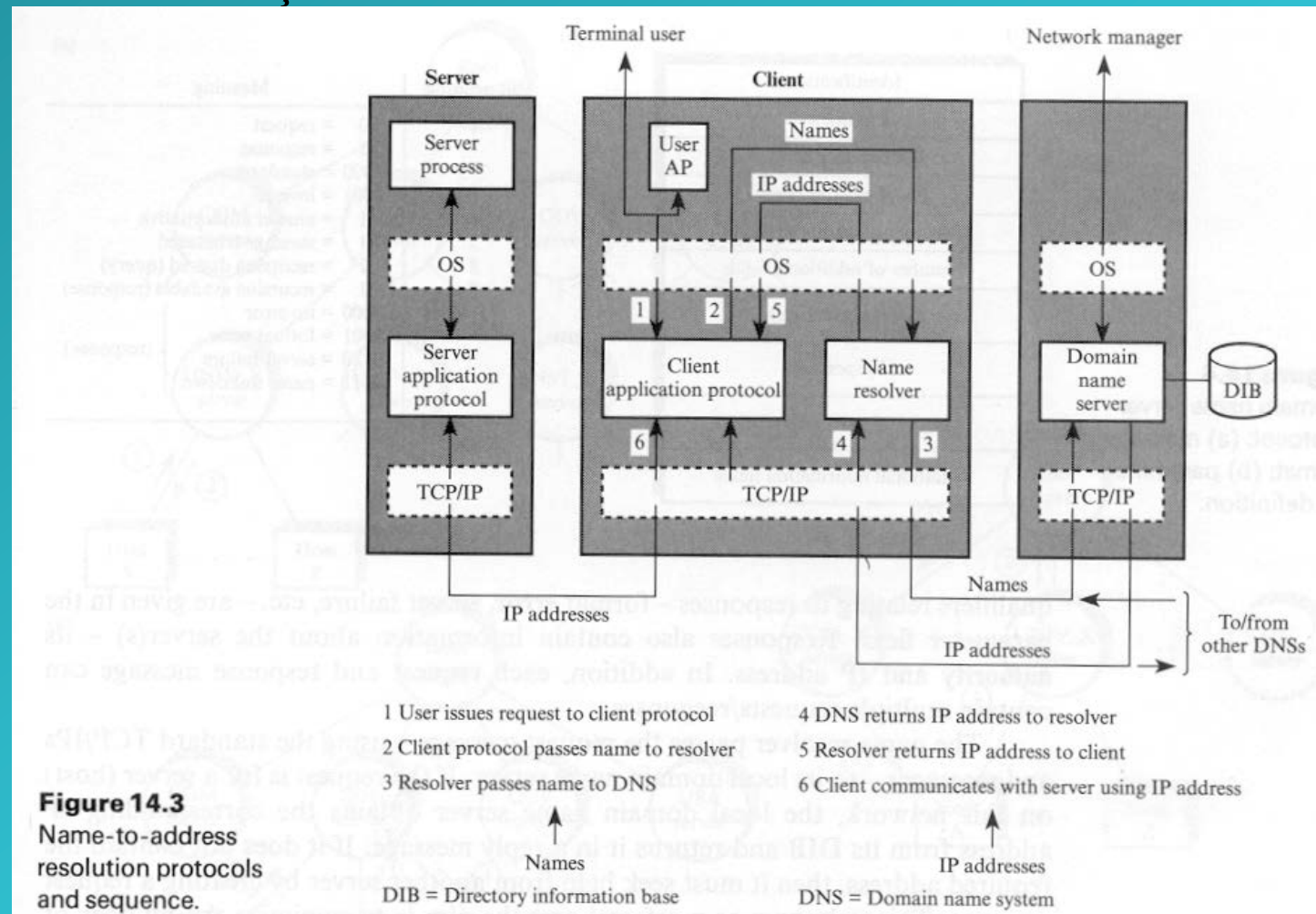
# Protocolo IP

## Tipos de serviços



# Protocolo IP

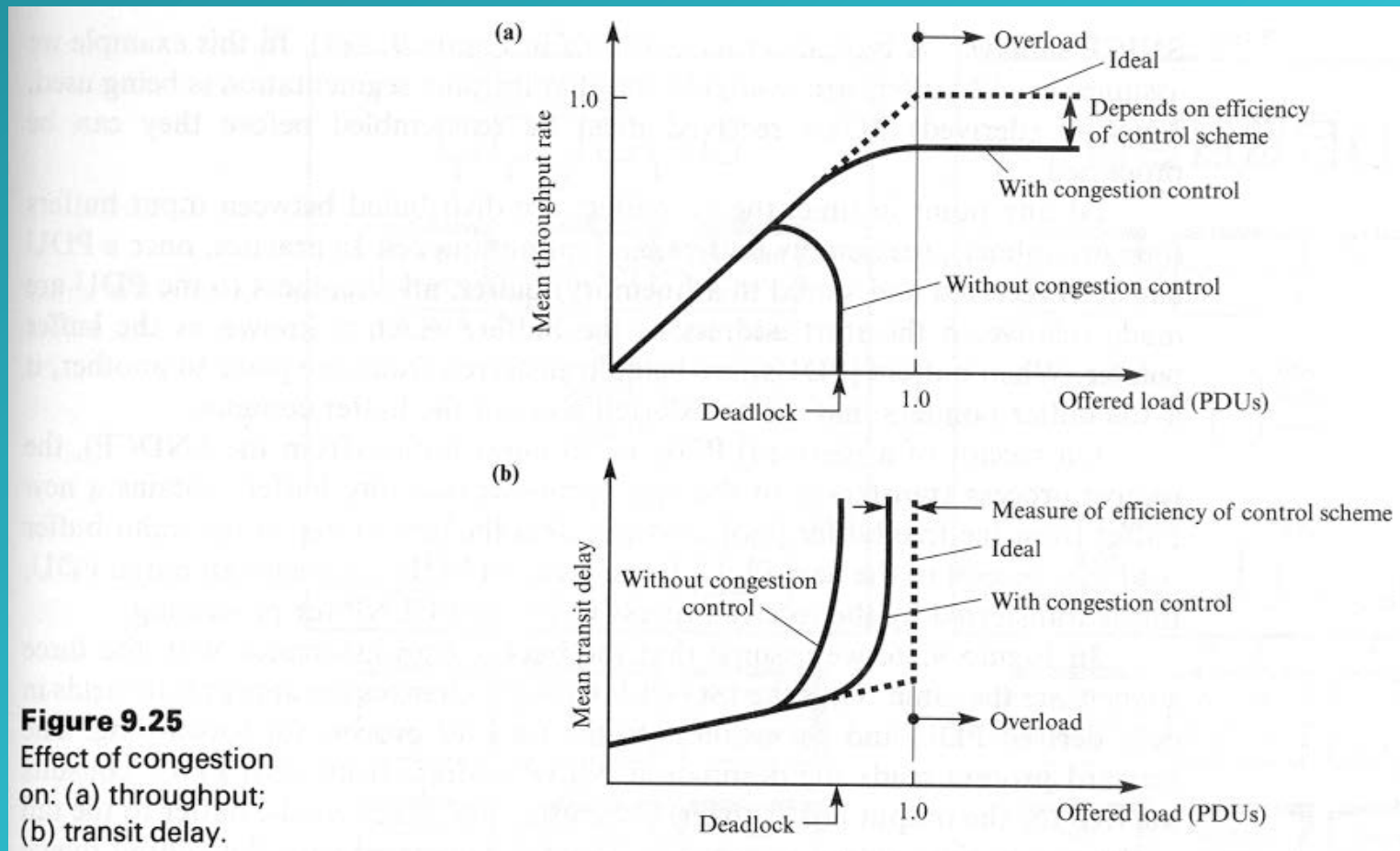
## Resolução de Endereços





# Protocolo IP

## Congestão nas Redes



**Figure 9.25**  
 Effect of congestion  
 on: (a) throughput;  
 (b) transit delay.

## Redes de banda larga

**Table 10.1** Bandwidth requirements of different media types.

<i>Media</i>	<i>Transaction type</i>	<i>Format</i>	<i>Sampling dimensions pixel, line, frame/s</i>	<i>Uncompressed bit rate</i>	<i>Compressed maximum bit rate</i>
Speech and music	Telephony		8 ksps ×8 bit/sample	64 kbps	8–32 kbps
	Teleconferencing		16 ksps ×8 bit/sample	128 kbps	48–64 kbps
	CD-audio		44.1 ksps × 16 bit/sample	705.6 kbps	128 kbps
Image	Normal resolution image	SVGA	640 pixel × 480 line ×8 bit/pixel	2.458 Mbits	24 k–245 kbits
		JPEG	720 pixel × 576 line ×16 bit/pixel	6.636 Mbits	104 k–830 kbits
	Very high resolution image		1280 pixel × 1024 line × 24 bit/pixel	31.46 Mbits	300 k–3 Mbits

## Redes de banda larga

Business video	Videophone	QCIF (H.261)	176 pixel × 144 line ×12 bit × 30 frame/s*	9.115 Mbps	p × 64 kbps (p = 1, 2)
		MPEG-4 (H.320)	176 pixel × 144 line ×12 bit × 10 frame/s	3.04 Mbps	64 kbps
	Video conferencing	CIF (H.261)	352 pixel × 288 line ×12 bit × 30 frame/s*	36.45 Mbps	m × 384 kbps (m = 1, 2, ... 5)
		MPEG-1 (PAL)	352 pixel × 288 line ×12 bit × 25 frame/s	30.4 Mbps	1.15 M–3 Mbps
		MPEG-1 (NTSC)	352 pixel × 240 line ×12 bit × 30 frame/s	30.4 Mbps	1.15 M–3 Mbps
Entertainment video	VCR	CIF (MPEG-2)	352 pixel × 240 line ×12 bit × 30 frame/s	30.4 Mbps	4 Mbps
	Broadcast television	MPEG-2 (PAL)	720 pixel × 576 line ×12 bit × 25 frame/s	124.4 Mbps	15 Mbps
		MPEG-2 (NTSC)	720 pixel × 480 line ×12 bit × 30 frame/s	124.3 Mbps	15 Mbps
	High quality television	HDTV	1920 pixel × 1080 line ×16 bit × 30 frame/s	994.3 Mbps	135 Mbps
		MPEG-3	1920 pixel × 1080 line ×12 bit × 30 frame/s	745.8 Mbps	20 M–40 Mbps

\* Frame rate can be 30, 15, 10, 7.5 frame/s.

CIF = Common intermediate format

MPEG = Moving pictures expert group

QCIF = Quarter common intermediate format

JPEG = Joint photographic experts group