

Projetando Controladores Digitais com FPGA

César da Costa

Novatec Editora

Capítulo 1

Introdução

No primeiro capítulo é feita uma introdução geral sobre controladores digitais baseados em microprocessador tradicional, seu princípio de funcionamento, ciclo de trabalho da CPU, tempos típicos de processamento, tempos de atraso, técnicas de programação e a linguagem Ladder. O objetivo deste capítulo não é ensinar o leitor a projetar controladores digitais microprocessados, mas sim apresentar algumas características de seu funcionamento, que auxiliarão os leitores no desenvolvimento de sistemas de controle digitais.

1.1 Controladores digitais tradicionais

Podem-se dividir os controladores digitais tradicionais em dois tipos: controladores digitais dedicados e controladores lógicos programáveis, também conhecidos como CLPs.

Os controladores digitais dedicados são basicamente aplicações de microcontroladores em tarefas de controle, cujo propósito específico é executar um algoritmo de controle gravado em sua memória ROM e, em geral, comunicar-se com dispositivos externos para troca de informações e atuação neles. São, por exemplo, a medição e o controle de temperatura, ligar /desligar motores, ler um teclado em sua entrada etc. Programa-se a ROM com códigos de máquina ou linguagem Assembler. O projetista deve conhecer bem o conjunto de instruções do microcontrolador utilizado, seu hardware e a tarefa de controle a ser realizada.

O desenvolvimento de software em linguagem de baixo nível (Assembler), para controladores digitais dedicados, é muito intuitivo e pessoal; raramente os projetistas envolvidos desenvolvem seus programas dentro das regras de programação estruturada.

Tal fato destaca a dificuldade de alteração do projeto pelo usuário final; sempre há necessidade da presença do projetista original, o que comercialmente pode ter suas vantagens para o fabricante do controlador. Entretanto, do ponto de vista técnico, contraria a grande flexibilidade oferecida pelos controladores lógicos programáveis (CLPs), ou seja, para um mesmo hardware existe uma infinidade de aplicações especificadas em software, passíveis de alterações pelo usuário final.

O desenvolvimento dos controladores lógicos programáveis (CLPs) começou em 1968, em resposta a uma requisição de engenharia da Divisão Hidráulica da General Motors (GM). Naquela época, a GM frequentemente passava dias, ou até semanas, alterando os sistemas de controle baseados em relés, sempre que ocorriam mudanças em um modelo de carro ou eram introduzidas modificações na linha de montagem.

Para reduzir o alto custo de instalação decorrente dessas alterações, a especificação do controlador digital necessitava de um sistema de estado sólido, microprocessado, com a flexibilidade de um computador, mas que pudesse ser programado e mantido pelos engenheiros e técnicos do chão de fábrica. Também era preciso que suportasse o ar poluído, a vibração, o ruído elétrico e os extremos de umidade e temperatura encontrados normalmente num ambiente industrial.

O CLP é um dispositivo de estado sólido, microprocessado, capaz de armazenar instruções para a implementação de funções de controle, tais como sequência lógica, temporização e contagem, além de realizar operações lógicas e aritméticas, manipulação de dados e comunicação em rede.

A Figura 1.1 ilustra a arquitetura básica de um CLP, formado por uma fonte de alimentação, uma Unidade Central de Processamento (CPU), um sistema de memória e circuitos de entrada e saída.

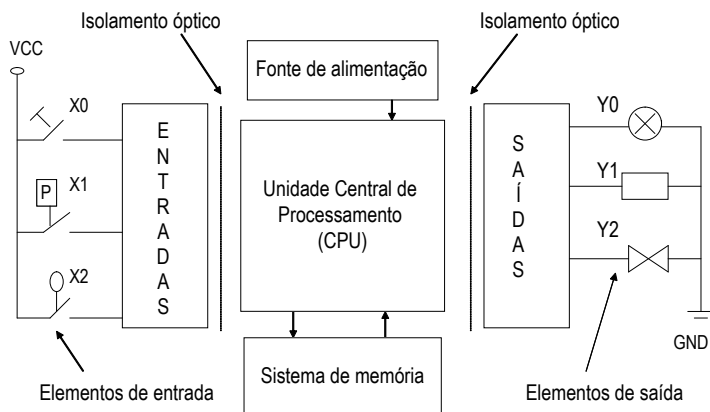


Figura 1.1 – Arquitetura básica de um CLP.

A fonte de alimentação é o componente responsável pelo fornecimento adequado de energia elétrica para a CPU e para os circuitos de entrada e saída.

Os circuitos de entrada formam a interface pela qual os dispositivos enviam informações de campo para o CLP. As entradas podem ser digitais ou analógicas e são provenientes de elementos de campo, como sensores, botões, pressostatos, chaves fim-de-curso etc.

Os dispositivos de saída, tais como solenóides, relés, contadores, válvulas, luzes indicadoras e alarmes, estão conectados aos circuitos de saída do CLP. As saídas, de maneira similar às entradas, podem ser digitais ou analógicas e são geralmente isoladas do campo por meio de isoladores galvânicos, como acopladores ópticos e relés.

O sistema de memórias é constituído tipicamente por memórias ROM e RAM. O programa e os dados armazenados no sistema de memória são geralmente descritos a partir dos seguintes conceitos:

- **Memória residente (ROM):** contém os programas considerados parte integrante do sistema, permanentemente armazenados, que supervisionam e executam a seqüência de operações, as atividades de controle e a comunicação com os dispositivos periféricos, bem como outras atividades.
- **Memória do usuário (RAM):** armazena o programa aplicativo do usuário, ou seja, o programa de aplicação.
- **Memória de dados ou tabela de dados (RAM):** nessa área são armazenados os dados associados ao programa de controle, tais como valores de temporizadores, contadores, constantes etc.
- **Memória imagem das entradas e saídas (RAM):** área que reproduz o estado de todos os dispositivos de entrada e saída conectados ao CLP.

A CPU do CLP é formada pelo microprocessador e seus circuitos de controle e comunicação. O microprocessador é o elemento principal da arquitetura do controlador digital e tem como funções o controle dos barramentos, o gerenciamento das comunicações com a memória e os dispositivos de entrada e saída, e a execução das instruções.

O microprocessador interpreta os sinais de entrada, executa a lógica de controle segundo as instruções do programa de aplicação, realiza cálculos e executa operações lógicas, para, em seguida, enviar os sinais apropriados às saídas. De um modo simplificado, o ciclo de instrução do microprocessador utiliza a seqüência de atividades apresentadas na Figura 1.2.

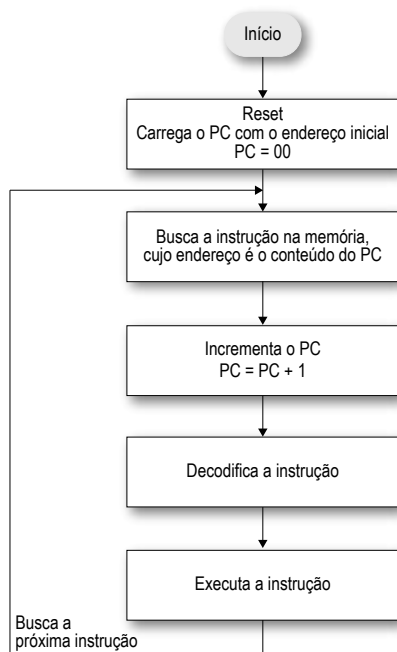


Figura 1.2 – Ilustração do ciclo de busca e execução de instrução.

Durante o ciclo de instrução, o microprocessador busca as instruções armazenadas na memória e executa cada uma delas. A execução do programa consiste na repetição seqüencial do processo de busca e execução das instruções.

Pode-se justificar a divisão do processamento da instrução em dois principais estágios, da seguinte forma:

1. a busca da instrução é uma operação comum para cada instrução, consiste na leitura de uma localização de memória e o posterior carregamento do seu conteúdo no registro de instrução;
2. a execução da instrução pode envolver várias operações e depende da natureza da instrução.

A Figura 1.3 apresenta um diagrama de tempos envolvidos na execução de um ciclo de instrução. Nota-se que o ciclo de instrução é igual à soma do ciclo de busca com o ciclo de execução; o ciclo de máquina M corresponde ao tempo que o microprocessador leva para acessar um endereço de memória ou um dispositivo de entrada e saída, e o estado T é um período de tempo que está vinculado ao clock do sistema. Também é a unidade de tempo que determina o tempo consumido na execução do ciclo de instrução.

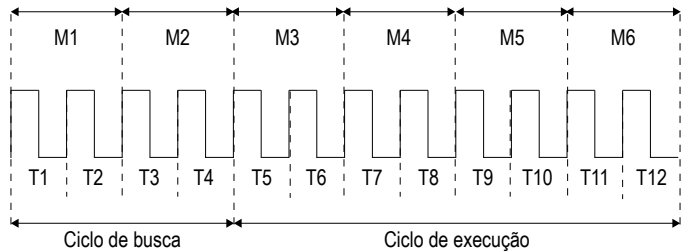


Figura 1.3 – Tempos envolvidos no ciclo de instrução.

Ao se analisar o diagrama de tempo da Figura 1.3, verifica-se que o ciclo de busca consome cerca de 40% dos estados totais de um ciclo de instrução, em uma tarefa que não está diretamente relacionada com o controle do processo, no qual o microprocessador está inserido.

Para melhor entendimento do tempo gasto com o ciclo de busca, vejamos a execução da soma de dois números pelo microprocessador, conforme mostra a Figura 1.4. Considerando-se a utilização de um clock de 4 MHz pelo microprocessador, será gerado um estado T com período de 0,25 µs.

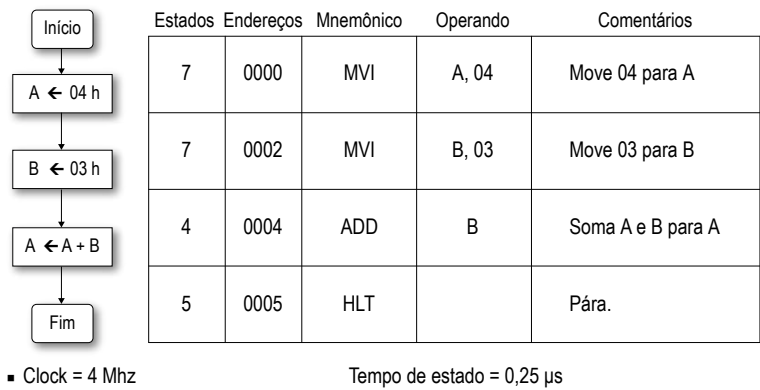


Figura 1.4 – Execução da soma de dois números pelo microprocessador.

Como a operação de somar os dois números consome um total de 23 estados, temos um ciclo total de instrução de 5,75 µs, sendo que cerca de 2,3 µs (40% do ciclo total) são gastos com ciclos de busca de instruções em memória.

O ciclo de busca é um tempo de processamento perdido, presente em todos os ciclos de instruções dos controladores digitais microprocessados, que não pode ser eliminado, pois a arquitetura interna dos microprocessadores tradicionais não permite a sua reconfiguração por hardware ou software. Este assunto, reconfiguração de hardware, voltará a ser abordado no Capítulo 3.

1.2 Princípio de funcionamento

O CLP funciona segundo um programa permanentemente armazenado em memória ROM, que executa um ciclo de scan chamado scan time e consiste de uma série de operações realizadas de forma seqüencial e repetida. A Figura 1.5 apresenta, em forma de fluxograma, as principais fases do ciclo de scan de um CLP. Os elementos mais importantes de um ciclo de scan são:

- **Atualização das entradas:** durante a varredura das entradas, o CLP examina os dispositivos externos de entrada quanto à presença ou à ausência de tensão, isto é, um estado “energizado” ou “desenergizado”. O estado das entradas é atualizado e armazenado temporariamente em uma região da memória chamada “tabela imagem das entradas”.
- **Execução do programa:** durante a execução do programa, o CLP examina as instruções do programa de controle (armazenado na memória RAM), usa o estado das entradas armazenadas na “tabela imagem das entradas” e determina se uma saída será ou não “energizada”. O estado resultante das saídas é armazenado em uma região da memória RAM chamada “tabela imagem das saídas”.
- **Atualização das saídas:** baseado nos estados dos bits da “tabela imagem das saídas”, o CLP “energiza” ou “desenergiza” seus circuitos de saída, que exercem controle sobre dispositivos externos.
- **Realização de diagnósticos:** ao final de cada ciclo de scan, a CPU verifica as condições do CLP, ou seja, se ocorreu qualquer falha em algum de seus componentes internos (fonte, circuitos de entrada e saída, memória etc.).

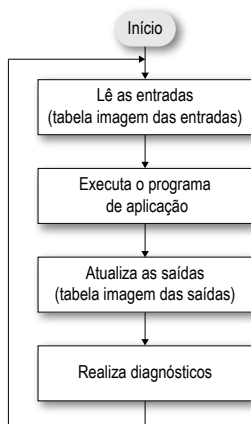


Figura 1.5 – Fluxograma básico de um ciclo de scan.

1.2.1 Ciclo de trabalho da CPU

O CLP executa cada linha do programa de forma seqüencial e não volta atrás para executar a linha anterior, até que se faça a próxima varredura do programa. Ele não executa loops ou desvios, como na programação tradicional. Seu processamento segue a seqüência do ciclo de scan apresentado na Figura 1.5, que não permite retroceder na seqüência de execução do programa.

As linhas são normalmente ordenadas de forma a configurar uma seqüência de eventos, ou seja, a linha mais acima é o primeiro evento e, assim, sucessivamente.

A Figura 1.6 exhibe uma seqüência de eventos em um programa em linguagem Ladder, que controla um estacionamento de veículos. Tanto nos diagramas elétricos quanto nos programas em linguagem Ladder, o estado das instruções de entrada (condição) de cada linha determina a seqüência em que as saídas são acionadas.

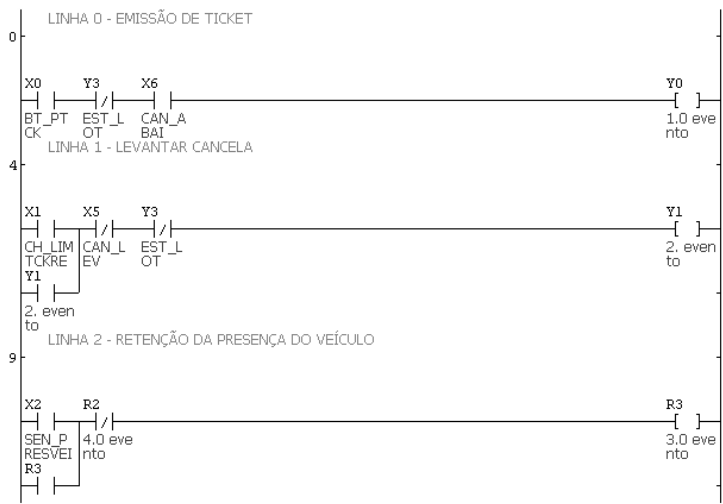


Figura 1.6 – Seqüência de eventos em programa Ladder.

A Tabela 1.1 ilustra os tempos típicos de execução de algumas instruções da linguagem Ladder, por exemplo, de um CLP fabricado pela empresa Allen Bradley.

Segundo a Allen Bradley, o tempo em microssegundos para executar uma instrução quando sua condição é verdadeira é maior que o tempo necessário para executar a mesma instrução quando sua condição é falsa.

Tabela 1.1 – Tempos típicos de execução de instruções do CLP

Tipo de instrução	Tempo de execução (μ s) (instrução falsa)	Tempo de execução (μ s) (instrução verdadeira)
Verificar se uma entrada está "energizada"	1,54	1,72
Verificar se uma entrada está "desenergizada"	1,54	1,72
"Energizar" uma saída	4,43	4,43
Temporizador (ON)	30,38	38,34
Contador (UP)	26,67	29,28
Contador (DOWN)	27,22	32,19
Comparação igual	6,60	21,52
Comparação maior que	6,60	23,60

1.2.2 Tempos típicos de processamento

O tempo de processamento é o espaço de tempo que o CLP leva para detectar uma entrada e “energizar” a saída correspondente. Os componentes do tempo de processamento incluem: tempo de atualização das entradas, tempo de execução do programa, tempo de atualização das saídas e tempo de housekeeping da CPU.

O tempo de housekeeping da CPU é o tempo, a cada início de ciclo de scan, que a CPU verifica as condições iniciais do CLP, ou seja, se ocorreu alguma falha em um dos seus componentes internos (fonte, circuito de E/S, memória etc.).

A Tabela 1.2 apresenta um exemplo dos tempos típicos envolvidos no processamento de um ciclo de scan de um micro CLP, da empresa Allen Bradley.

Tabela 1.2 – Tempos típicos de um ciclo de scan

Descrição do processamento	Tempo máximo (μ s)
Tempo de atualização das entradas	8,0
Tempo de execução do programa	9,7
Tempo de atualização das saídas	8,0
Tempo de housekeeping (diagnósticos)	18,0
Tempo total de processamento (máximo)	43,75

1.2.3 Tempos de atraso no processamento

O ciclo de scan muitas vezes pode ocasionar problemas graves no controle de processos industriais, por não reconhecer uma entrada durante seu ciclo de funcionamento. Isso pode ocorrer com sinais de entrada de resposta rápida, como, por exemplo, sensores com resposta em torno de 10 kHz.

Dependendo do tempo de variação do estado lógico dos sinais de entrada, o CLP pode demorar mais tempo para acionar a saída ou mesmo nunca reconhecer uma entrada. Para ilustrar essa situação, a Figura 1.7 mostra a variação de três sensores de resposta rápida (sensor 1, sensor 2 e sensor 3) ligados na entrada de um CLP. As variações ocorrem durante três ciclos de varredura (scan).

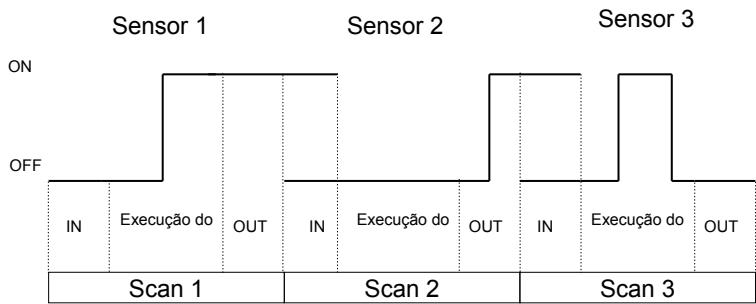


Figura 1.7 – Sensores de resposta rápida durante os ciclos de scan.

No primeiro ciclo de scan, o sensor 1 muda do estado desligado para ligado, durante a fase de execução do programa. Dessa forma, a transição do sensor 1 não é reconhecida nesse ciclo, sendo reconhecida somente na fase de leitura das entradas no segundo ciclo de scan.

O sensor 2 muda de estado na fase de atualização das saídas do segundo ciclo de scan; sua transição não é reconhecida durante esse ciclo, mas sim durante o terceiro ciclo de scan, durante a fase de atualização das entradas desse ciclo.

A mudança de estado do sensor 3 não será reconhecida em nenhum desses ciclos, porque tem resposta muito rápida e ocorre durante a fase de execução do programa no terceiro ciclo de scan.

Para evitar esse tipo de problema, o tempo do ciclo de scan do CLP deve ser menor que os tempos de amostragens dos sinais envolvidos no sistema.

1.3 Programação

A programação do CLP pode ser elaborada em várias linguagens de programação. A organização internacional IEC (International Electrotechnical Commission) é a responsável pela padronização das linguagens de programação para CLP, sendo a norma IEC 61131-3 Programming Languages a responsável pela classificação dessas linguagens, conforme ilustra a Tabela 1.3.

Tabela 1.3 – Classificação das linguagens de programação

Classes	Linguagens
Tabulares	Tabela de Decisão
Textuais	IL (Instruction List) ST (Structured Text)
Gráficas	LD (Diagrama de Relés) FBD (Function Block Diagram) SFC (Sequential Flow Chart)

A forma de programação pode ser remota (off-line) ou local (on-line), por meio de teclados especiais, interfaces gráficas ou por de microcomputador padrão IBM PC. A programação é executada e posteriormente transferida para o CLP, via porta de comunicação RS232C ou RS485.

1.3.1 Linguagem de diagrama de relés (Ladder)

Apesar das tentativas de padronização da norma IEC 61131-3, ainda não existe uma padronização rigorosa para programação em linguagem de diagrama de relés (Ladder Diagram). Em outras palavras, a linguagem Ladder de um fabricante de CLP não funciona no CLP de outro fabricante; o que existe é uma semelhança na representação gráfica dos diversos fabricantes, que representa esquematicamente o diagrama elétrico de fácil entendimento, com boa aceitação do mercado.

A linguagem de diagrama de relés (Ladder) é uma simbologia construída por linhas numa planilha gráfica, na qual cada elemento é representado como uma célula. Cada célula ou elemento gráfico é uma macroinstrução desenvolvida a partir de instruções do microprocessador.

A Figura 1.8 apresenta um painel de símbolos gráficos utilizados na programação Ladder pelo software FPCWIN GR, configurador dos CLPs FP0, FP1 e FP-M, fabricados pela empresa japonesa Matsushita Electric Works.

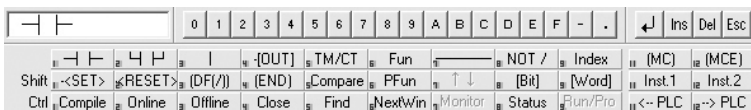


Figura 1.8 – Painel de programação Ladder do software FPWIN.

Um programa em linguagem Ladder assemelha-se bastante a um diagrama de contatos elétricos, em que os símbolos gráficos representam os dispositivos reais e a maneira como estão conectados.

O programa em linguagem Ladder utiliza símbolos semelhantes: nesse caso, os símbolos gráficos representam macroinstruções lógicas do programa de aplicação, armazenado na memória do usuário (RAM).

Não existe barra de alimentação, nem o fluxo de corrente ao longo do programa. Outra diferença é que em um diagrama elétrico descrevem-se os dispositivos como abertos ou fechados (“energizados” ou “desenergizados”). No programa em linguagem de diagrama de relés, as macroinstruções são condições lógicas verdadeiras ou falsas.

A Figura 1.9 apresenta um trecho de um programa em linguagem Ladder. As macroinstruções mais freqüentemente usadas são chamadas de “Normalmente Aberto” (NA), “Normalmente Fechado” (NF) e “Energizar Saída” (OUT). Elas são representadas na forma de símbolos gráficos colocados nas linhas do programa, sendo por isso também conhecidas como “simbologia de contatos de relés”.

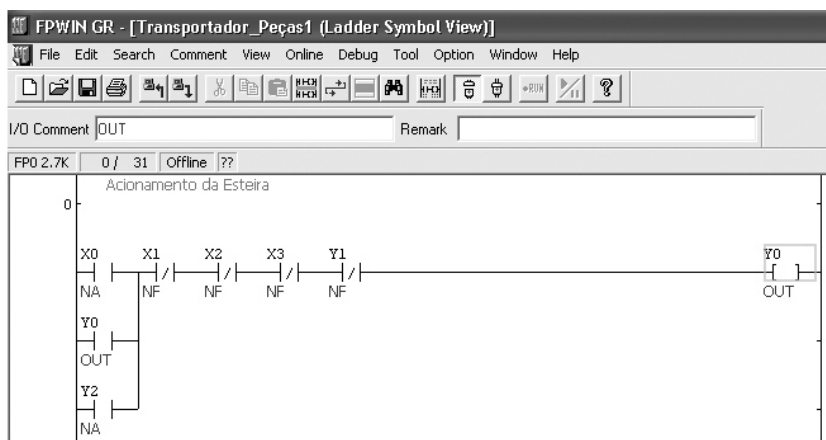


Figura 1.9 – Trecho do programa em linguagem Ladder.

1.4 Teste de avaliação

1. Quais as diferenças básicas (hardware e software) entre um controlador digital dedicado baseado em microcontrolador e um controlador lógico programável (CLP)?
2. O que é um ciclo de instrução do microprocessador?
3. Descreva por que o ciclo de busca é um tempo de processamento perdido, presente em todos os ciclos de instruções dos controladores digitais microprocessados.
4. Explique como funciona um ciclo de scan (scan time) de um CLP.
5. Apesar de sua flexibilidade, por que a arquitetura interna de um microcontrolador não pode ser reconfigurada pelo usuário final?
6. O que é uma linguagem Ladder?
7. Apesar de padronizada pela norma IEC 61131-3, por que a linguagem Ladder de um fabricante não funciona no CLP de outro fabricante?
8. Durante a execução de um ciclo de scan de um CLP, um sensor ligado em uma de suas entradas é atuado. Entretanto, o CLP não detecta a sua atuação. Explique o que pode ter ocorrido.
9. Seguindo a linguagem Ladder – macroinstrução normalmente aberta ($\overline{X3} \text{ } \overline{I}$), normalmente fechada ($\overline{X3} \text{ } I$) e energizar saída ($\overline{Y0} \text{ } I$) –, programe as seguintes expressões algébricas:
 - a) $\overline{B}C + CD + AD$
 - b) $\overline{A}\overline{B} + \overline{B}C$
 - c) $\overline{B}\overline{C} + C\overline{D} + \overline{A}BD$