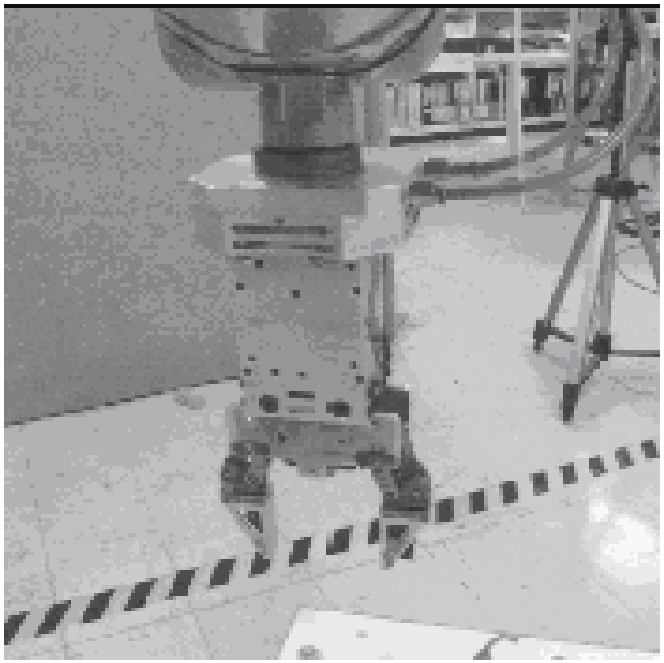
Tópicos em Sistema Digitais



Prof. Adilson Gonzaga

Elementos Básicos

Switches (CHAVES)

Representações

Non-locking

Locking

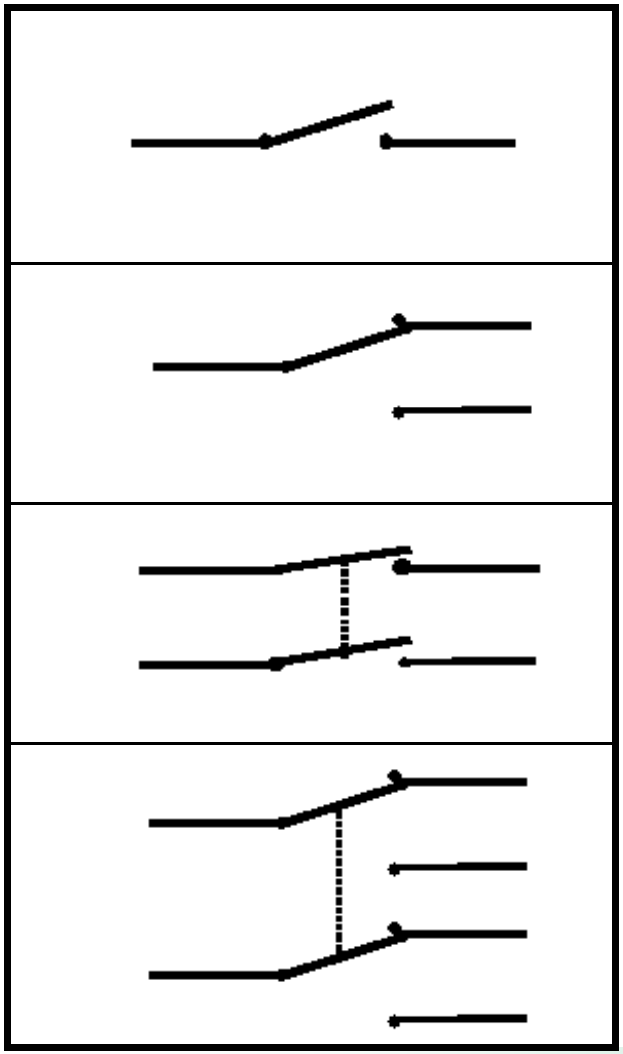
Normally Open

Normally Closed

Elementos Básicos

Switches (CHAVES)

Configurações



SPST -- **Single Pole Single Throw**
(Single Pole on-off)

SPDT -- **Single Pole Double Throw**
(SPCO - Single Pole Changeover
Single Pole on-on)

DPST -- **Double Pole Single Throw**
(Double Pole on-off)

DPDT -- **Double Pole Double Throw**
(DPCO - Double Pole Changeover
Double Pole on-on)

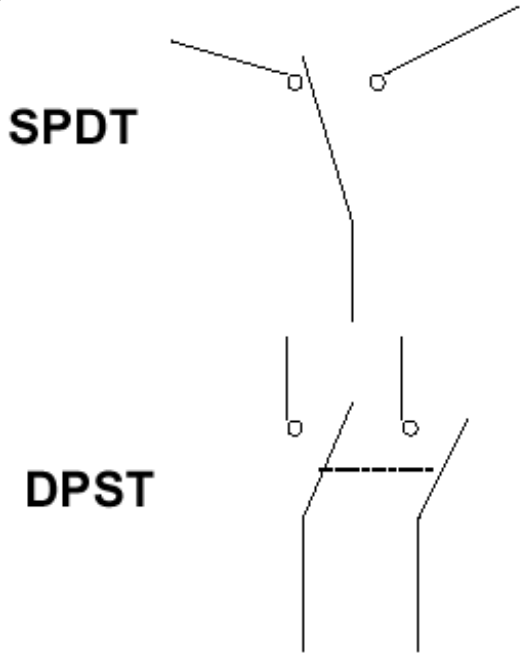
Elementos Básicos

Switches (CHAVES)

Termos

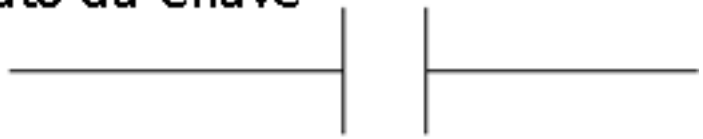
Pole = Número de partes móveis que se conectam ou
Número de circuitos individuais

Throw = Número de Estados



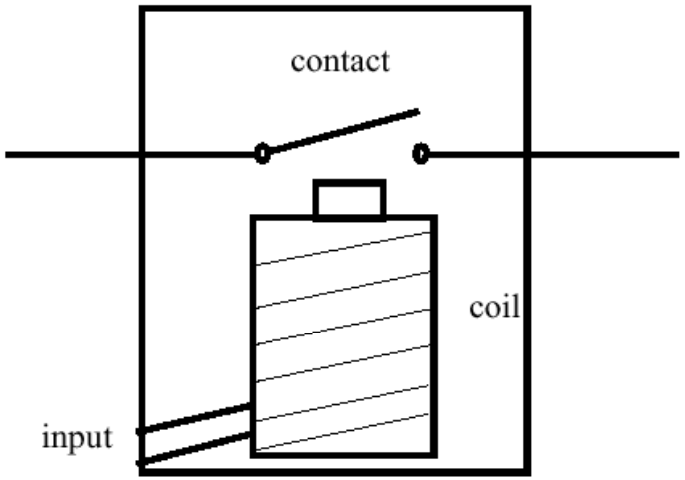
Representação esquemática

Contato da Chave



Elementos Básicos

Relays (Relês)

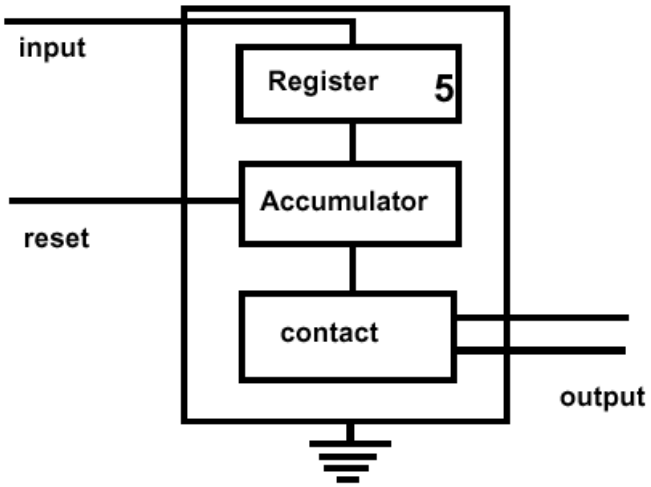


Chave cuja operação é ativada de maneira eletromagnética.

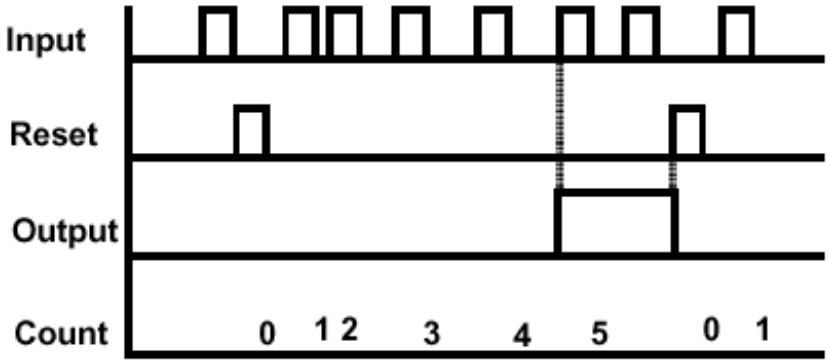
Representação esquemática

Elementos Básicos

Counters (Contadores)



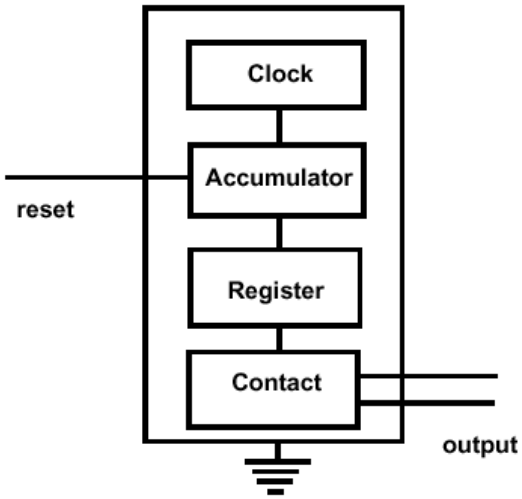
Os Contadores Digitais tem suas saídas na forma de contatos de Relês, quando uma contagem pré-estabelecida é atingida



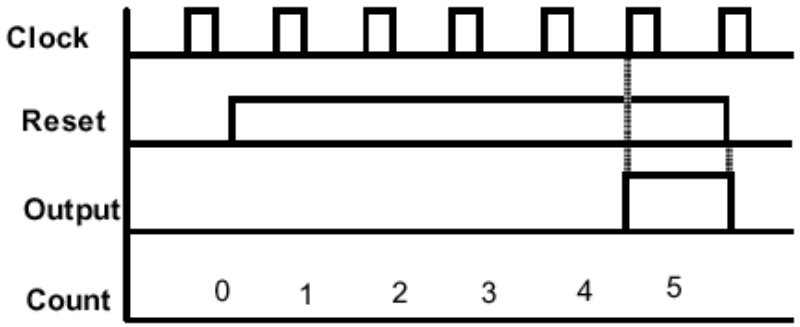
Ex: Contagem de 5 subidas de borda.

Elementos Básicos

Timers (Temporizadores)



Um Temporizador consiste de um clock interno, um registrador de contagem e um acumulador.



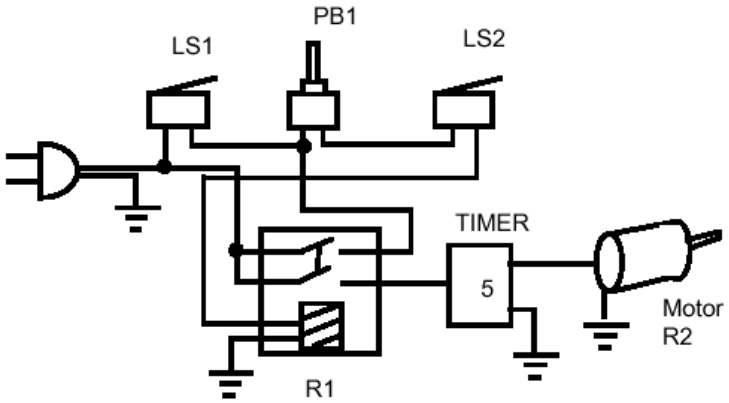
Time 5 seconds.

É utilizado para temporizar eventos, fechando os contatos após um tempo pré-programado.

Lógica de Relês

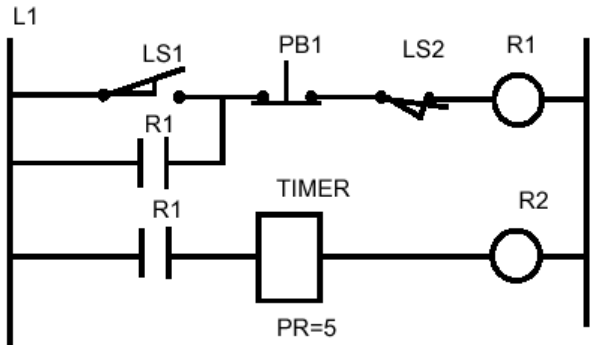
Exemplo de controle de processo:

Um processo inicia ligando um motor (R2), cinco segundos após uma peça tocar uma chave de limite (LS1). O processo termina automaticamente quando a peça terminada toca uma segunda chave de limite (LS2). Uma chave de emergência (PB1) termina o processo a qualquer instante, quando for acionada.



Processo

Lógica



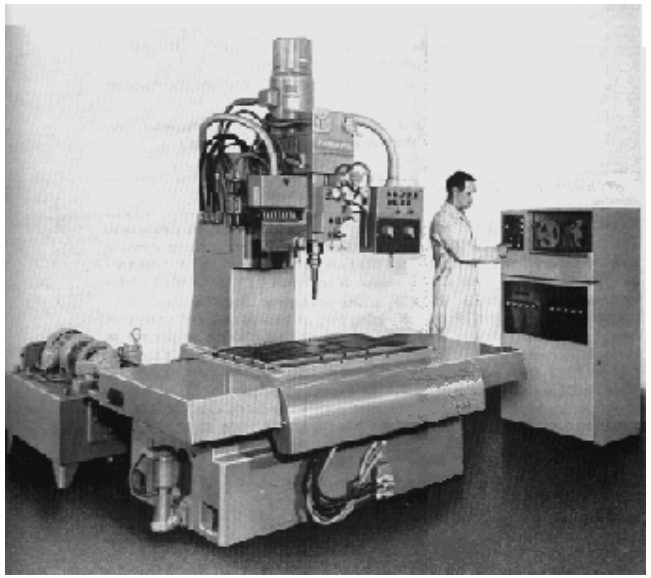
Comando Numérico Computadorizado (CNC):

- ***Uso de um computador para comandar o caminho da ferramenta cortante de um torno mecânico ou uma máquina fresadora. Com isto tem-se alta precisão no produto final e alta repetibilidade com um mesmo programa.***

❑ **O comando numérico computadorizado (CNC) é uma técnica que permite a operação automática de uma máquina por meio de uma série de instruções codificadas que contêm números, letras e outros símbolos.**

❑ **As máquinas CNC podem ser facilmente reprogramadas para atender a novos projetos e podem ser adaptadas a diferentes situações de produção.**

Comando Numérico Computadorizado (CNC):



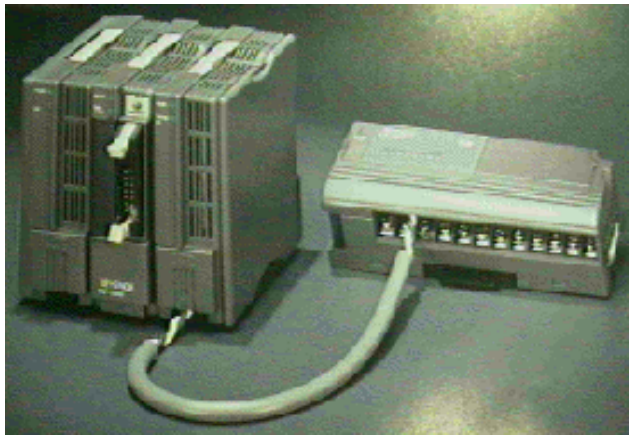
As primeiras máquinas fresadoras CN possuíam uma unidade de controle tão volumosa que precisava ficar fora da máquina.

Atualmente, máquinas CNC, possuem um comando numérico pequeno, normalmente embutido na própria máquina.

Controladores Lógicos Programáveis (CLP):

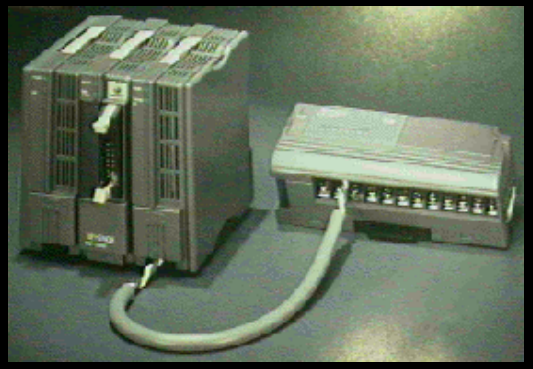
- ***(ou simplesmente Controladores Programáveis - CPs) são usados para controlar uma sucessão de eventos. Basicamente é um computador, que recebe sinais de sensores e/ou chaves, executa um programa e envia ordens a saídas, as quais acionarão elementos como motores, válvulas, etc...***

❑ **Antes do surgimento dos CLP's, as tarefas de comando e controle de máquinas e processos industrias eram feitas por relés eletromagnéticos, especialmente projetados para este fim.**



Controladores Lógicos Programáveis

CLP



PLC

Programmable Logic Controller

- **Dispositivo desenvolvido para substituir os circuitos a relê que realizavam a lógica seqüencial no controle de máquinas.**

- **O PLC trabalha "observando" suas entradas e dependendo dos seus estados, liga/desliga suas saídas.**

- **O usuário entra com um programa que fornece os resultados desejados.**

História:

- Os PLC's foram introduzidos no final da década de 60.

- No meio da década de 70 a tecnologia dominante nos PLC's eram as Máquinas de Estados Seqüenciais.

- Com o desenvolvimento dos Microprocessadores, os PLC's passaram a utilizá-los.

- A capacidade de comunicação dos PLC's começaram a aparecer em 1973.

Tipos de PLC:

Allen-Bradley PLC-3

Large Size PLC

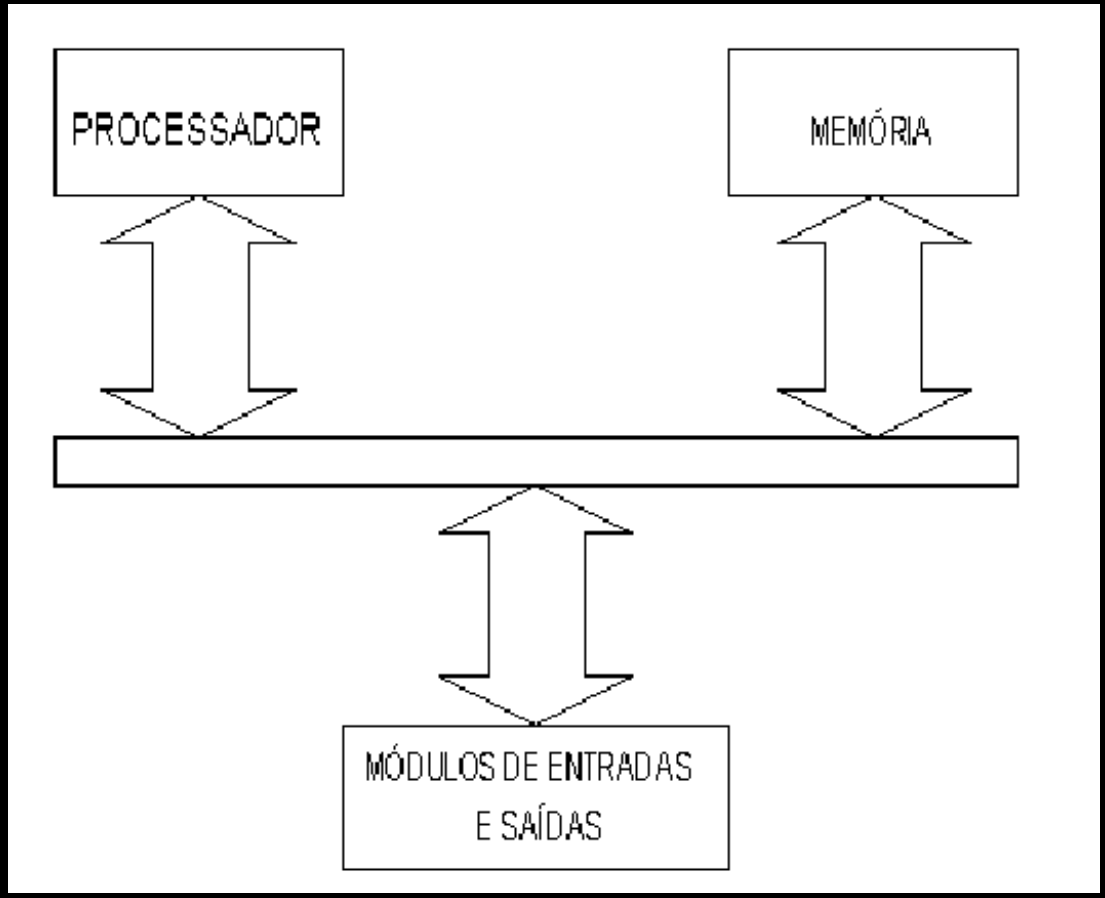
- Medidas --> 19” x 20” x 14,5”
- cerca de 10.000 pontos de I/O
- suporta todas as funções
- diversos protocolos de comunicação.

Allen-Bradley MicroLogix 1000

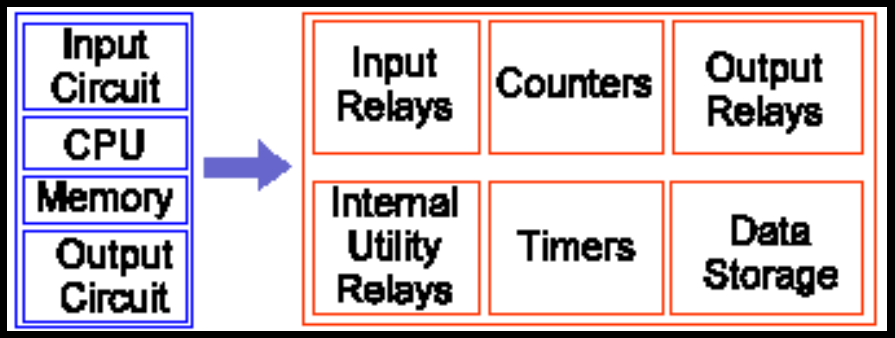
Small Size PLC

- Medidas --> 4,72” x 3,15” x 1,57”
- cerca de 32 pontos de I/O
- comunicação RS-232

Estrutura de um PLC:



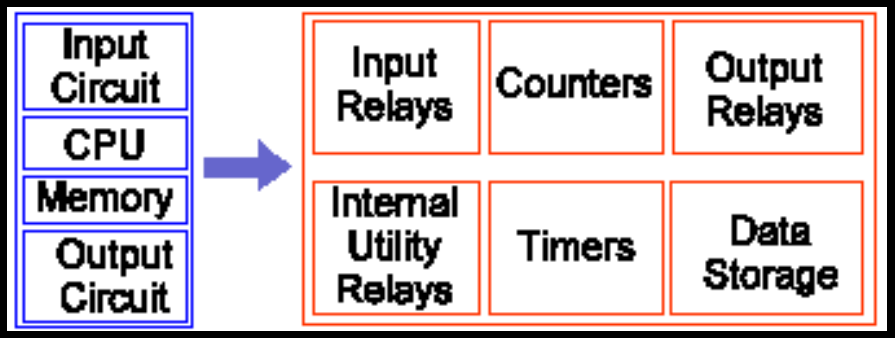
Partes constituintes de um PLC:



Input Relays -(Contatos) Conectados ao mundo externo. Existem fisicamente e recebem sinais de chaves, sensores, etc... (Tipicamente não são relês, mas transistores)

Internal Utility Relays - (Contatos) Não existem fisicamente. Não recebem sinais externos; são simulados e permitem aos PLC´s eliminarem relês externos.

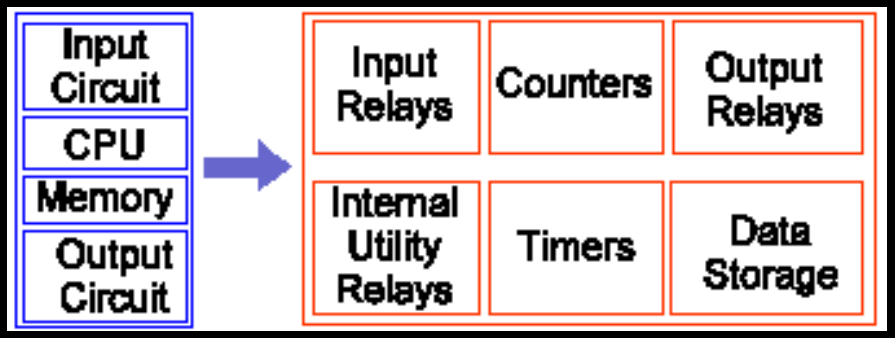
Partes constituintes de um PLC:



Counters - Não existem fisicamente. São contadores simulados e podem ser programados para contar pulsos "up" ou "down".

Timers - Não existem fisicamente. Incrementos variam de 1 ms a 1s.

Partes constituintes de um PLC:

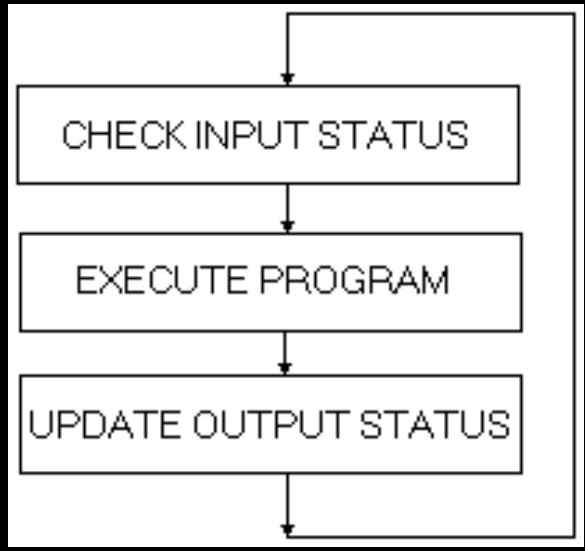


Output Relays - (bobinas) São conectados ao mundo externo. Existem fisicamente e enviam sinais on/off para solenóides, luzes, etc... (Podem ser transistores, relês ou triacs, dependendo do modelo)

Data Storage - São registradores dedicados ao armazenamento de dados. Podem ser temporários ou permanentes, mantendo informações quando o PLC está desligado.

Operação de um PLC:

- **Um PLC opera continuamente executando um programa**



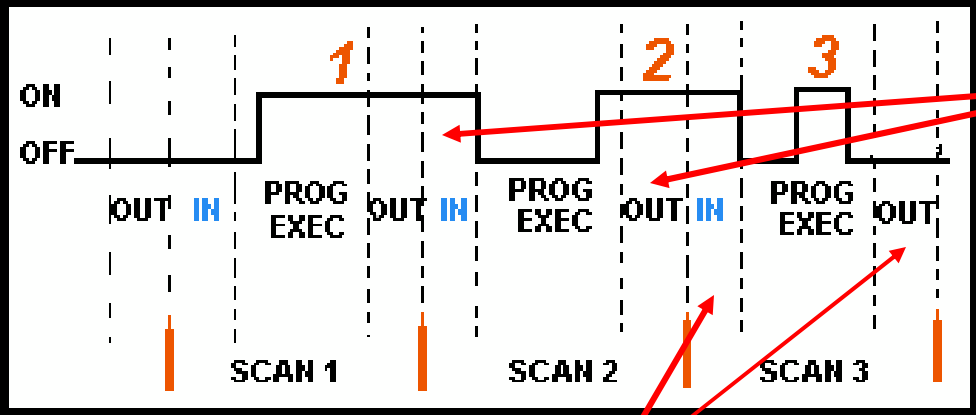
Check Input Status - O PLC verifica cada entrada se está "on" ou "off". Armazena os dados na memória para serem usados no próximo passo.

Execute Program - O PLC executa o programa armazenado, baseado nas entradas lidas. Armazena o resultado para ser usado no próximo passo.

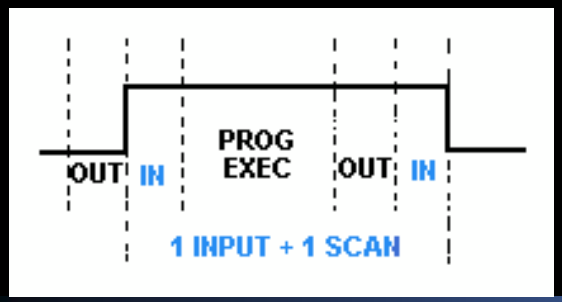
Update Output Status - Finalmente o PLC atualiza suas saídas baseado nos passos anteriores.

Tempo de Resposta do PLC:

Exemplo:



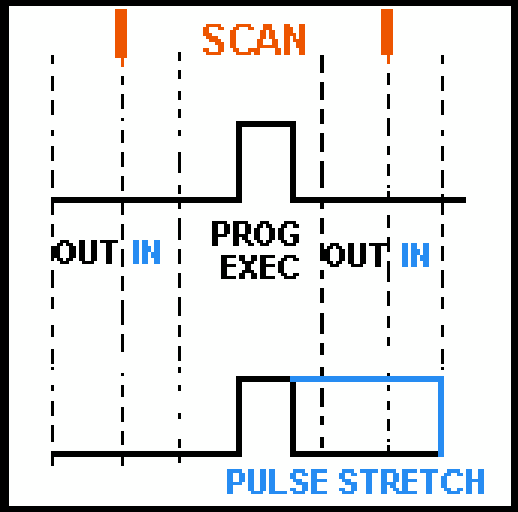
- **A entrada 1 só é "vista" na varredura 2.**
- **A entrada 2 só é "vista" na varredura 3.**
- **A entrada 3 nunca será detectada.**

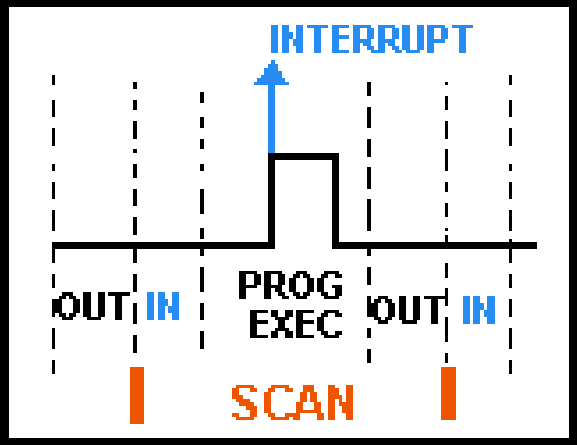


Regra: **A entrada deve estar "on" por no mínimo 1 "input delay time" + 1 "scan time".**

Técnicas para adequação de pulsos de entrada:

Esticamento do Pulso de Entrada:
(Pulse Strech Function)



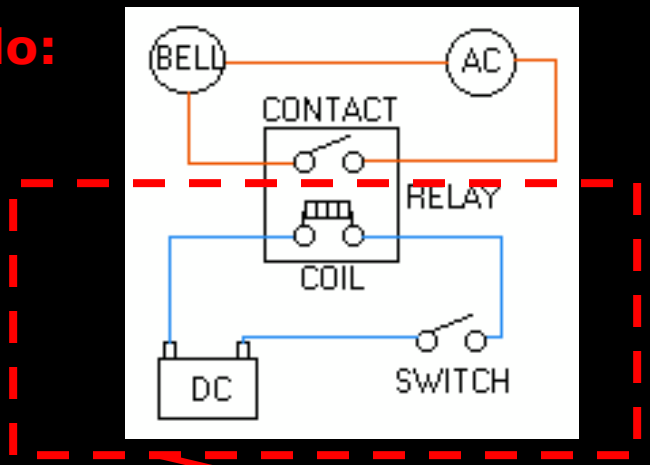


Interrupção:

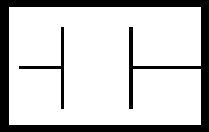
Linguagem LADDER

Linguagem gráfica utilizada para programação de CLP's.
O elemento básico é o relê.

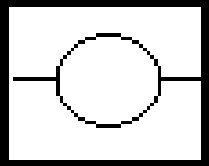
Exemplo:



Símbolos gráficos:

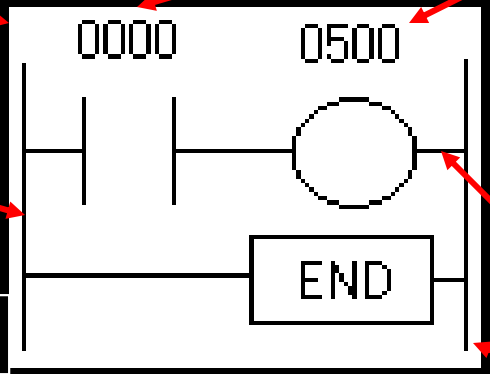


Contato da Chave



Bobina

Endereços no CLP



Alimentação DC

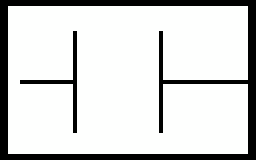
Degrau da Escada

Escada (LADDER)

Terra

Instruções Básicas:

Load

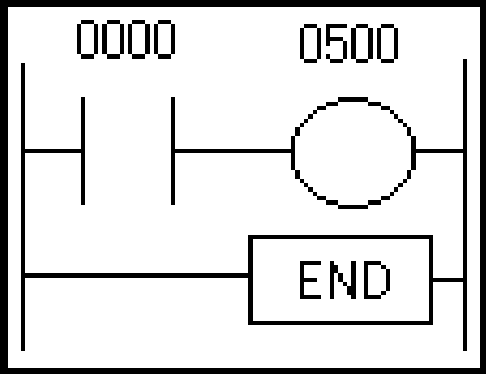


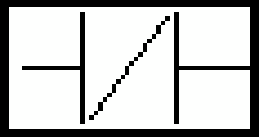
A instrução Load (LD) é um contato normalmente aberto. Também chamada de: Examine se "on" (XIO).

- **Quando a entrada física está "on" a instrução é verdadeira.**

- **Um sinal de entrada precisa estar presente para que o símbolo seja ativado.**

- **Pode ser usado para:**
  - ✓ **Entradas Internas**
  - ✓ **Entradas Externas**
  - ✓ **Saídas Externas**



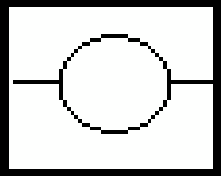


LoadBar

A instrução LoadBar (LD) é um contato normalmente fechado. Também chamada de: Examine se "closed" (XIC).

- **Quando a entrada física está "off" a instrução é verdadeira.**
- **Um sinal de entrada não deve estar presente para que o símbolo seja ativado.**
- **Pode ser usado para:**
  - ✓ **Entradas Internas**
  - ✓ **Entradas Externas**
  - ✓ **Saídas Externas (em alguns casos)**

Logic State	Load	LoadBar
0	False	True
1	True	False



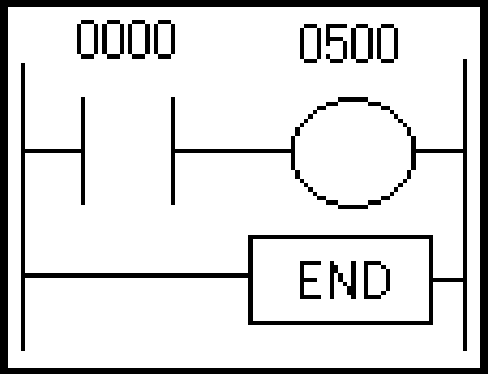
Out

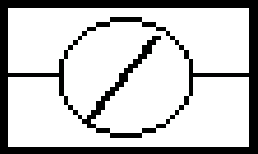
- **Quando existe um caminho de "instruções verdadeiras" que precedem esta instrução no degrau da escada, ela também será verdadeira.**

- **Quando a instrução é verdadeira ela está fisicamente "on".**

- **Pode ser considerada como uma saída normalmente aberta.**

- **Pode ser usada para bobinas internas e saídas externas.**





OutBar

- **Representa a bobina normalmente fechada de um relê.**

- **Quando existe um caminho "falso" que precede esta instrução no degrau da escada, ela será verdadeira.**

- **Quando a instrução é verdadeira ela está fisicamente "on".**

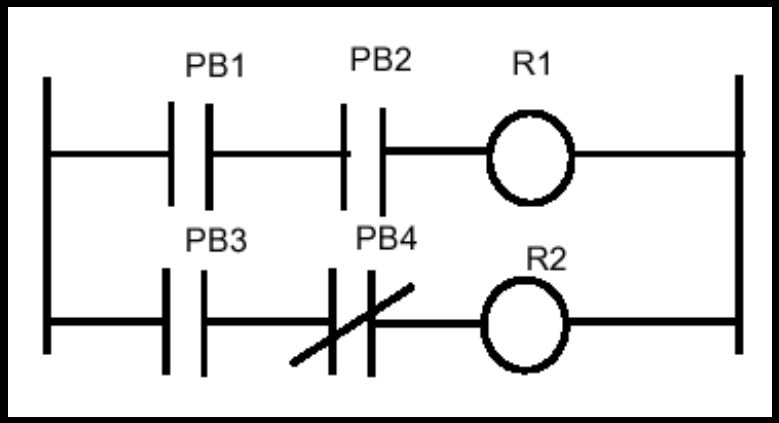
- **Pode ser considerada como uma saída normalmente fechada.**

- **Pode ser usada para bobinas internas e saídas externas.**

Logic State	Out	OutBar
0	False	True
1	True	False

Operações Lógicas:

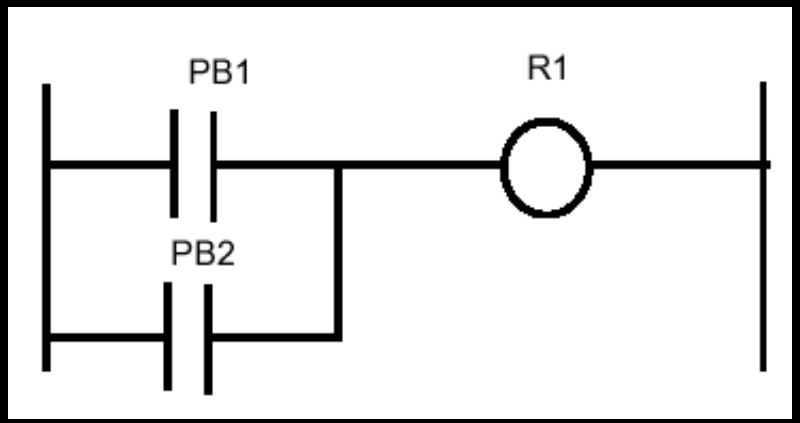
AND (E)



R1 = PB1 AND PB2

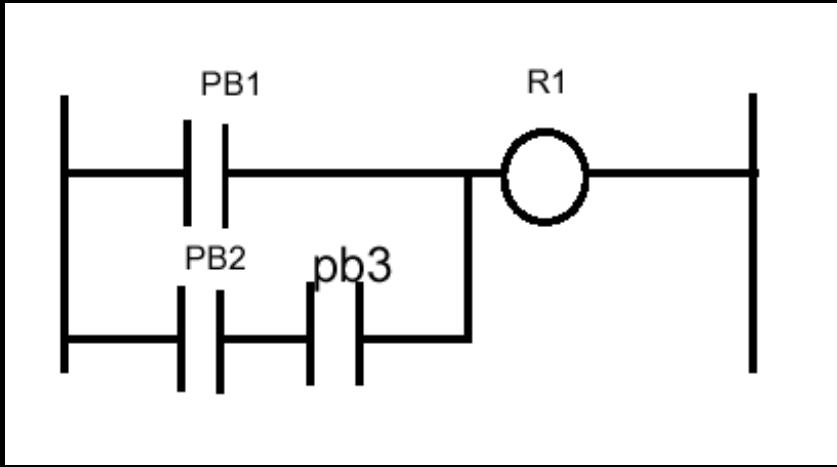
R2 = PB3 AND $\overline{\text{PB4}}$

OR (OU)



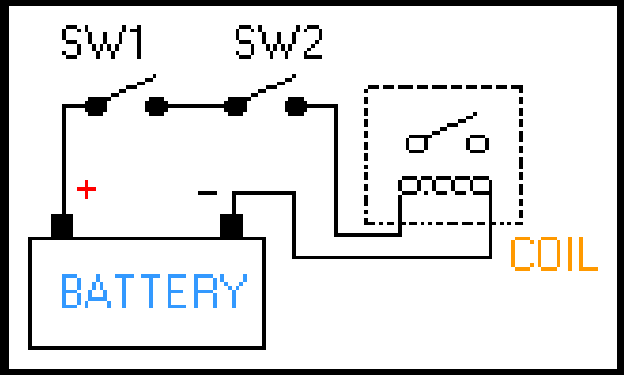
R1 = PB1 OR PB2

Operações Lógicas:

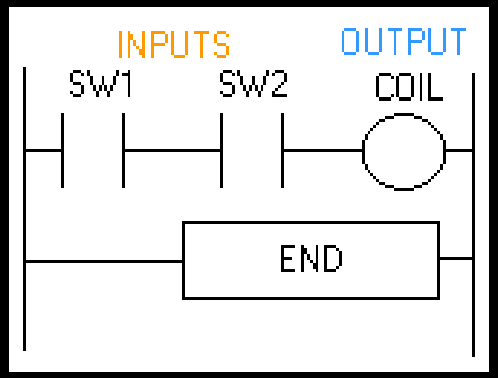


R1 = PB1 OR (PB2 AND PB3)

Exemplo1a:

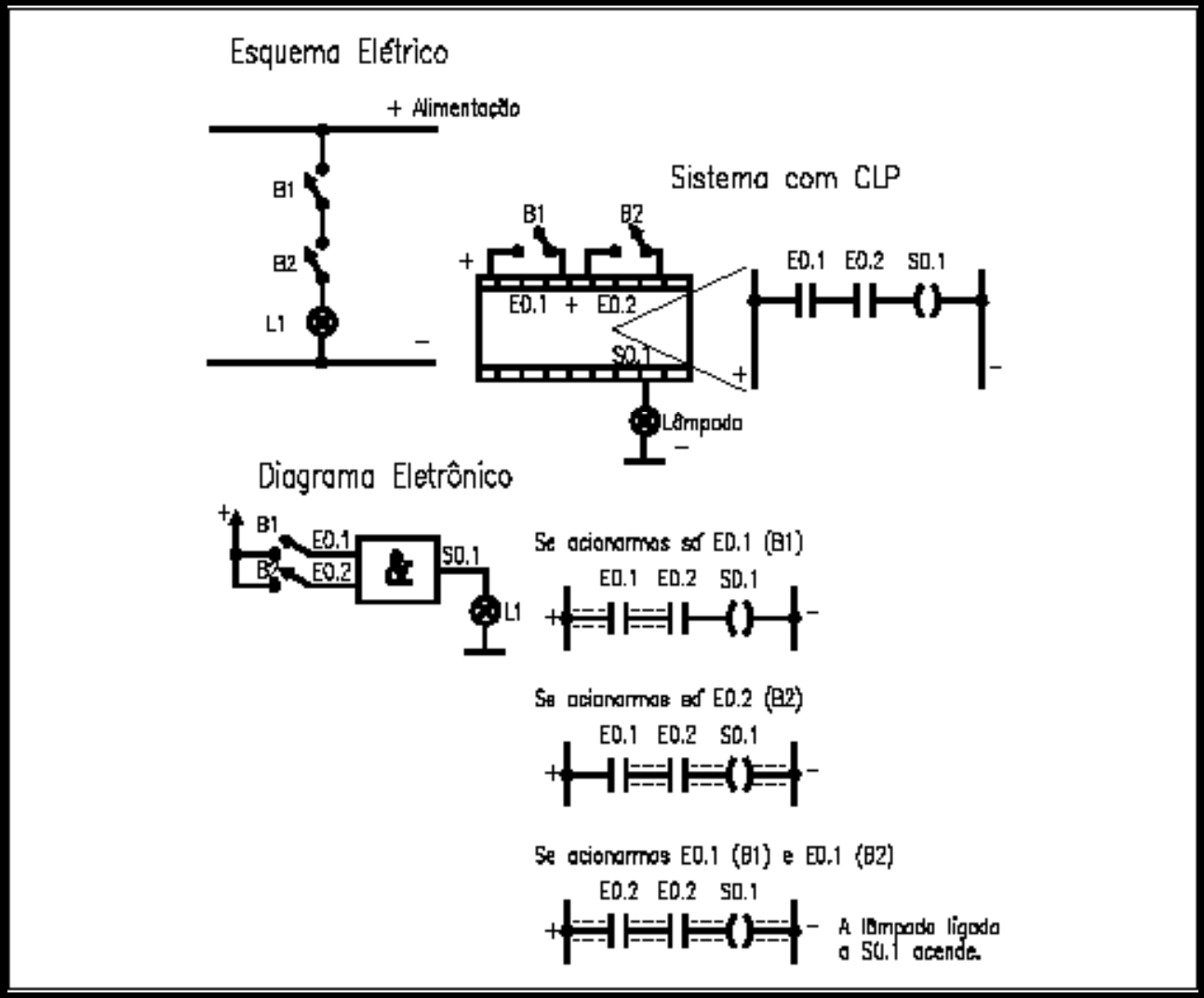


- **A primeira instrução em um degrau deve ser sempre ima instrução de entrada (Load, LoadBar) e a última deve ser sempre uma instrução de saída (Out, OutBar) ou sua equivalente.**



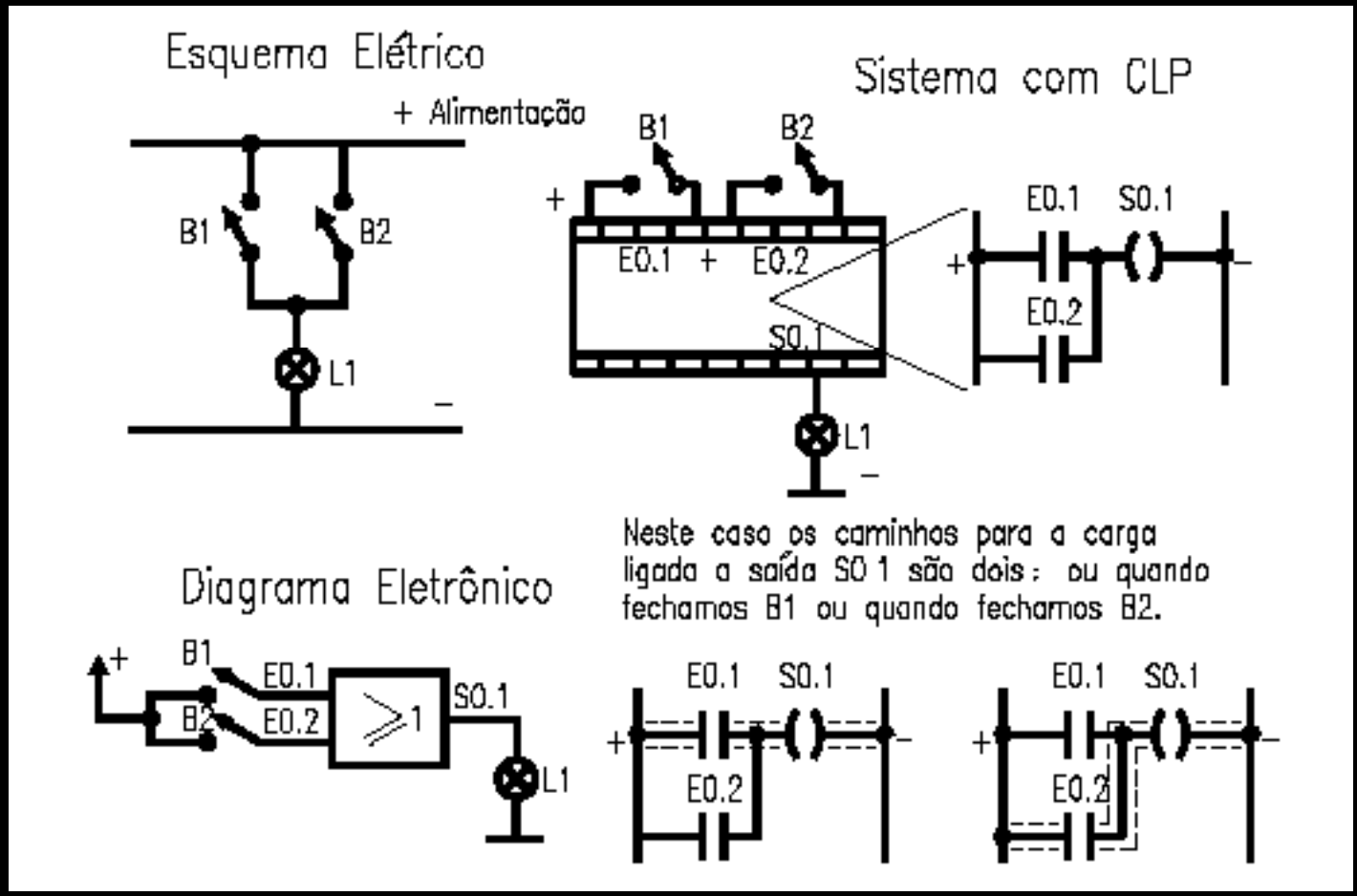
Exemplo1b:

a lâmpada L1 deve ser acesa apenas se os dois interruptores B1 e B2 forem acionados. Corresponde à operação lógica E.

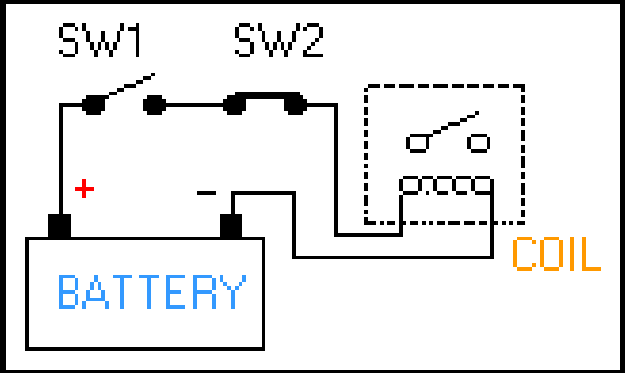


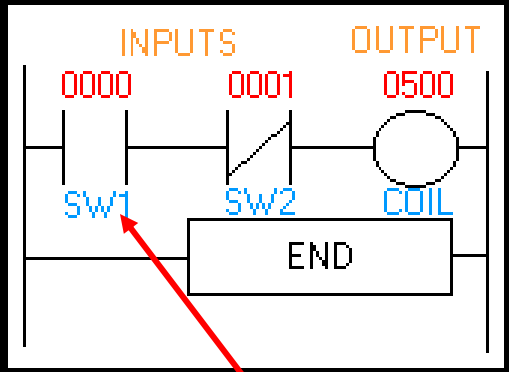
Exemplo1c:

ligar a lâmpada L1 se os interruptores B1 ou B2 forem acionados. Corresponde à operação lógica OU.



Exemplo2:





Registradores internos:

REGISTER 00															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
														1	0

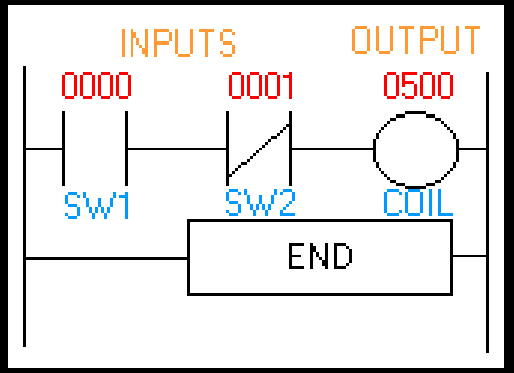
REGISTER 05															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
															0

- **A maioria dos PLC's usam registradores de armazenamento de 16 Bits.**

- **Os PLC's apenas energizam a saída quando todas as condições, no degrau da escada, forem verdadeiras.**

LOGICAL CONDITION OF SYMBOL			
LOGIC BITS	LD	LDB	OUT
Logic 0	False	True	False
Logic 1	True	False	True

SW1 deve estar em nível lógico 1 e SW2 deve ser 0, para que a saída seja verdadeira (bobina energizada).

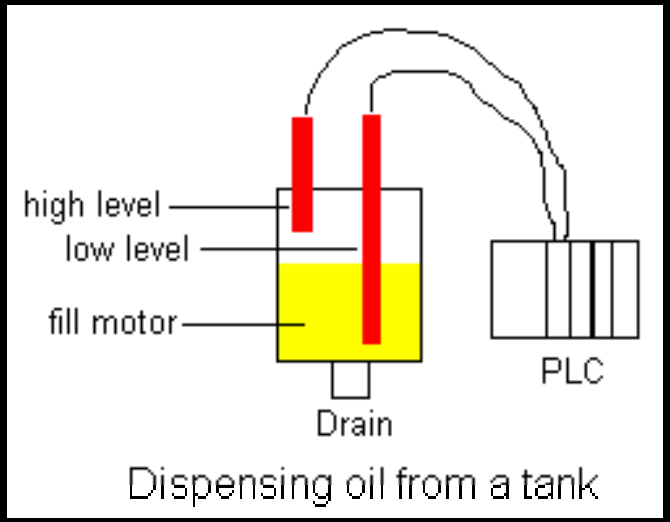


Inputs		Outputs	Register Logic Bits		
SW1(LD)	SW2(LDB)	COIL(OUT)	SW1(LD)	SW2(LDB)	COIL(OUT)
False	True	False	0	0	0
False	False	False	0	1	0
True	True	True	1	0	1
True	False	False	1	1	0

Exemplo3:

Controlador de Nível

Acionar um motor que bombeia óleo lubrificante para um recipiente até que o nível mais alto seja atingido. Desligar o motor e deixar drenar até que o nível mais baixo seja atingido.

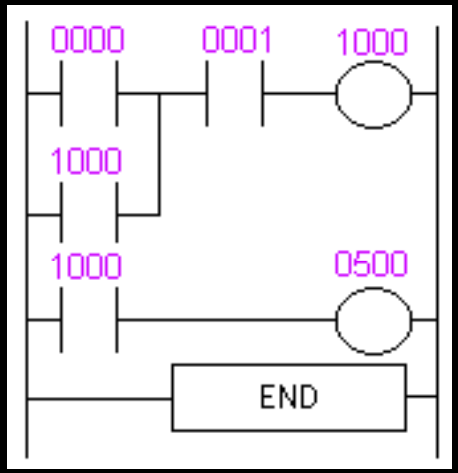


Os sensores são de fibra óptica.
ON → quando não estão imersos no óleo
OFF → quando estão imersos no óleo

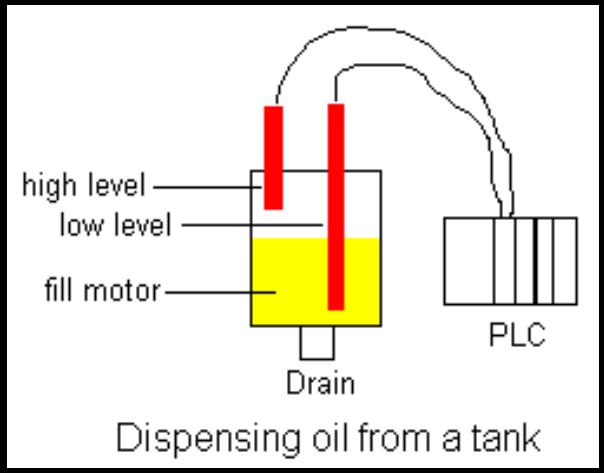
Atribuição de endereços:

Inputs	Address	Output	Address	Internal Utility Relay
Low	0000	Motor	0500	1000
High	0001			

Programação Ladder



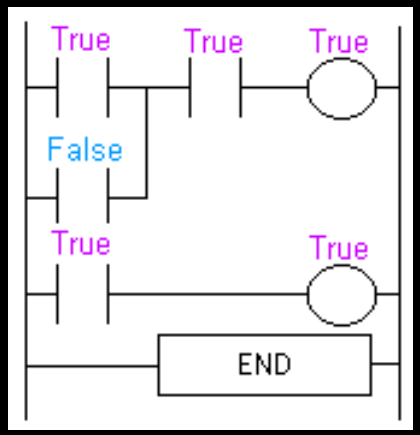
Processo



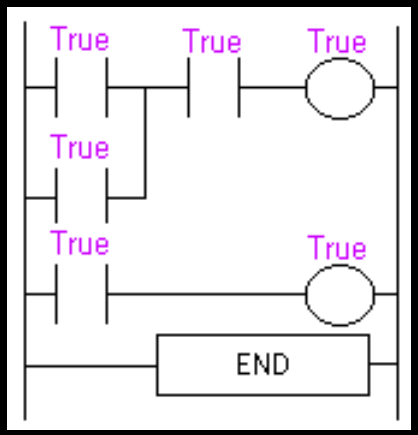
Inicialmente o tanque está vazio, os dois sensores estão ON.
0000 = 1 e 0001 = 1 → 1000 =1 (motor ligado → 0500 =1)
(0000 = 0 ou 1000 =1) e 0001 = 1 → 1000 = 1 → 0500 =1
0000 = 0 e 0001 = 0 → 1000 = 0 (motor desligado 0500 =0)
0000 = 0 e 0001 = 1 → 1000 =0 → 0500 =0 (drenando)

As Varreduras (scan) do programa:

Tanque Vazio

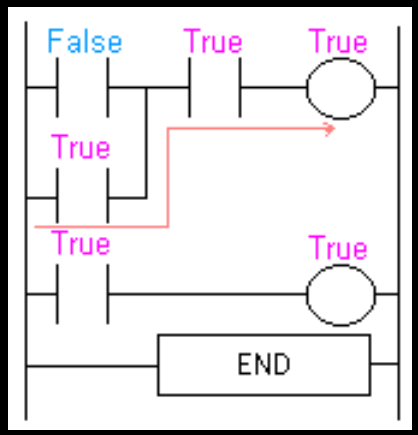


Scan 1



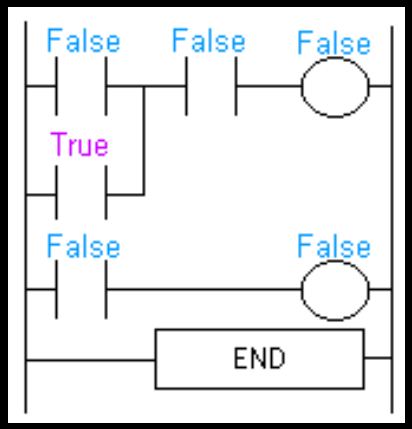
Scan 2 - 100

Óleo no sensor baixo

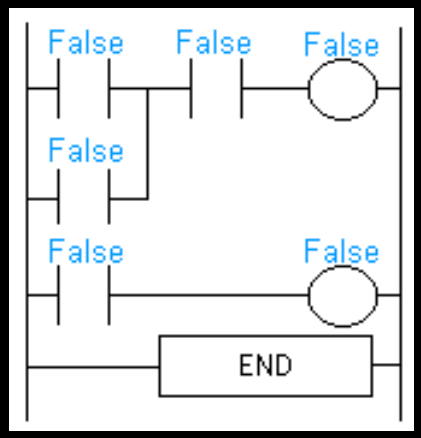


Scan 101-1000

Óleo no sensor alto

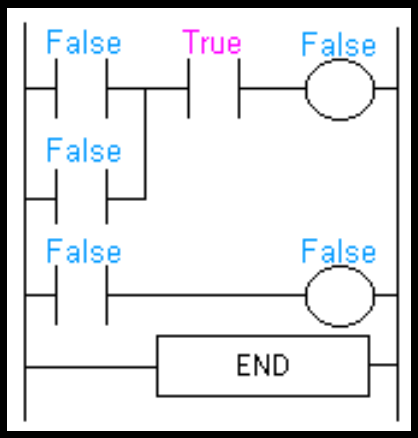


Scan 1001



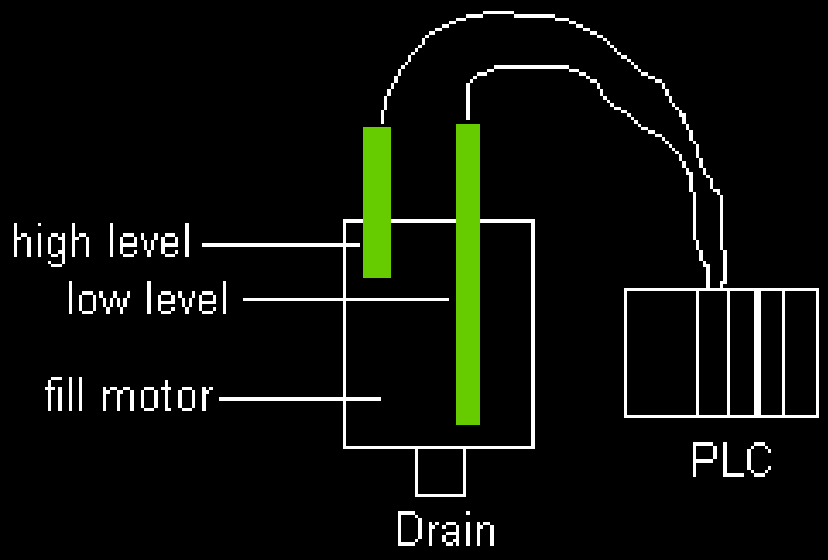
Óleo no sensor alto

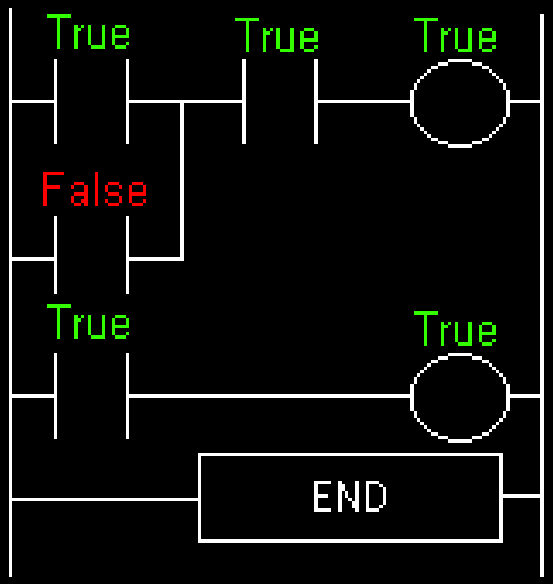
Scan 1002



Óleo abaixo do sensor alto

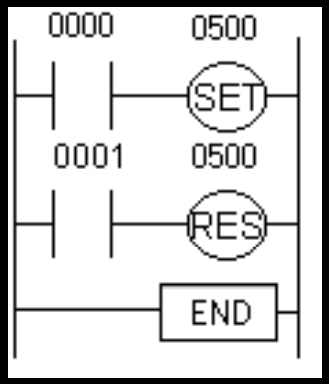
Scan 1050

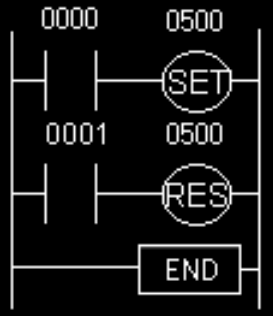

high level
low level
fill motor
Drain
PLC

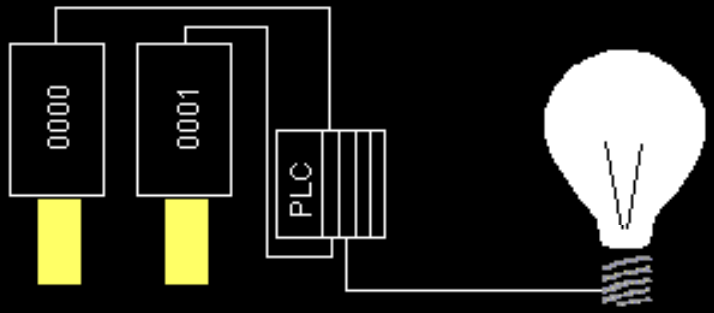

True
True
True
False
True
True
END

Instruções de Latch

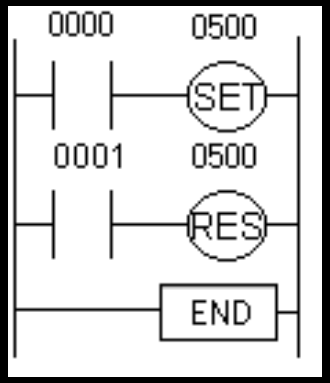
- **Permite utilizar chaves de contato momentâneo e programar o PLC tal que quando aperta-se uma, a saída liga e quando aperta-se a outra a saída desliga.**







- **A instrução de Latch é chamada de SET ou OTL. A instrução de Unlatch é chamada de RES (Reset) ou RST**



O que acontece se as duas chaves forem acionadas simultaneamente ?

Seqüência de Scanning:

- **A escada é sempre varrida de cima para baixo e da esquerda para a direita.**

- **Primeiro passo: ler as entradas → ambas estão ON**

- **Segundo passo: executar o programa**
→ primeiro degrau → 0000 = 1 → 0500 = SET
→ segundo degrau → 0001 = 1 → 0500 = RES

- **Terceiro passo: atualizar as saídas**
0500 = RES → a saída será RES (zerada)

Contadores:

- **Podem ser UP Counters, Down Counters ou UP/Down Counters**

- **Normalmente implementados em Software pelo PLC.**

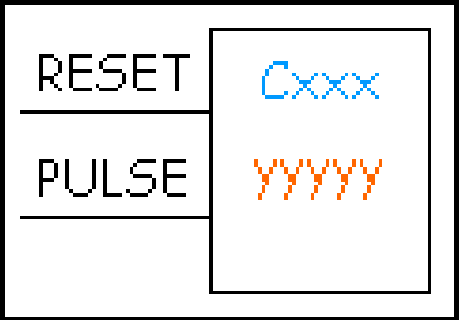
- **High Speed Counter → Implementado em Hardware.**

- **Regra: Usar o contador normal (por software) a menos que os pulsos de contagem sejam mais rápidos que 2 vêzes a freqüência de uma varredura (scan time).**

- **Ex: Se o "scan time" = 2 ms e os pulsos a serem contados são de 4 ms ou mais, usar o contador por software.**

Tipos de Contadores:

UP Counter

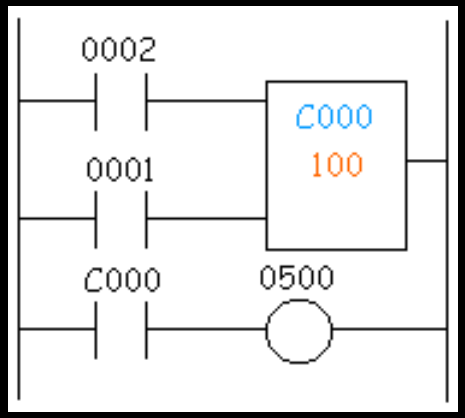


RESET → quando esta entrada é ativada (ON), a contagem acumulada é zerada.

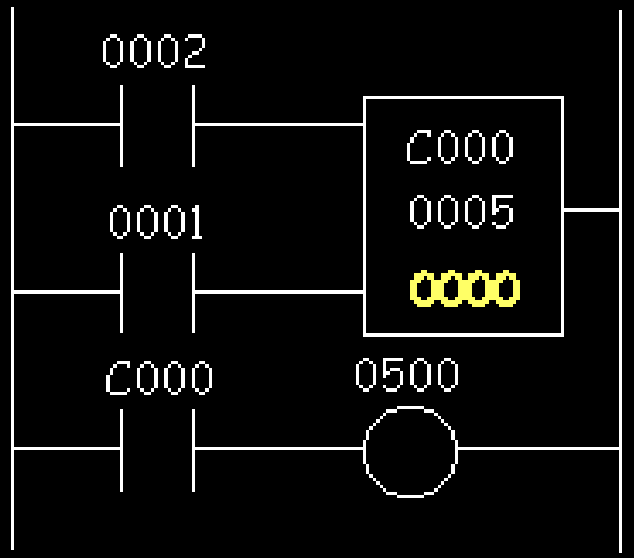
PULSE → entrada dos pulsos a serem contados.

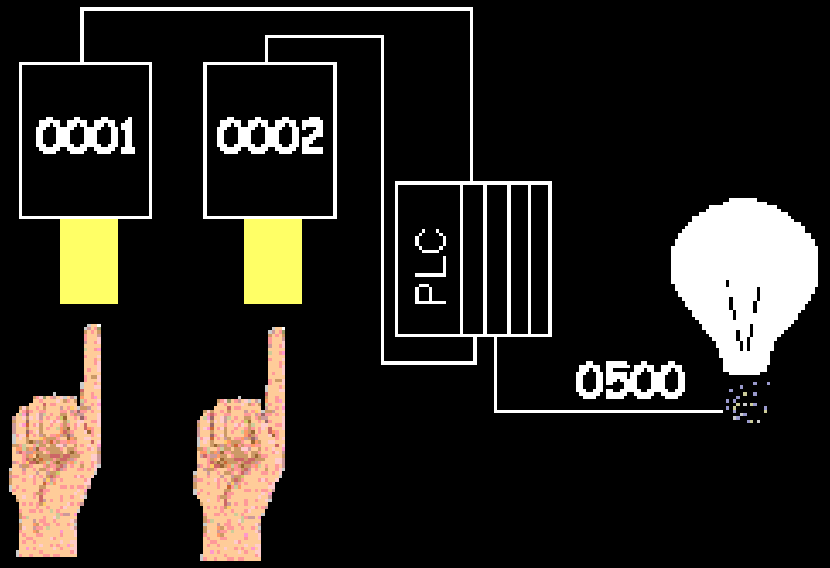
Cxxx → endereço ou Nome do contador

yyyyy → número de pulsos a serem contados

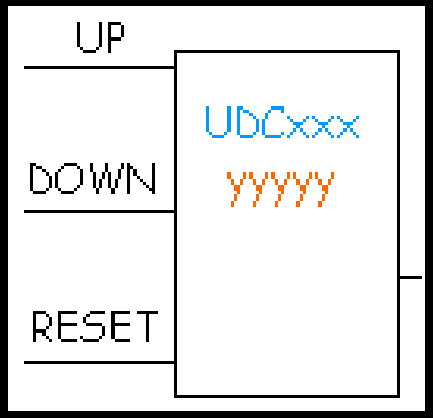


- **O contador 000 contará 100 pulsos na entrada 0001 e após ligará a saída 500.**
- **O sensor na entrada 0002 zerará a contagem.**

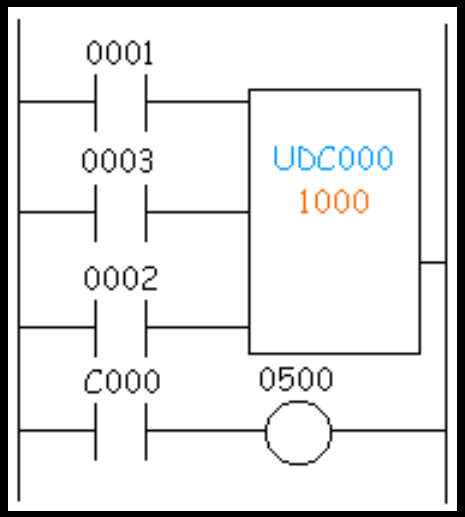

0002
C000
0005
0000
0001
C000
0500


0001
0002
PLC
0500

UP/Down Counter



Exemplo:



- **Na entrada 0001, quando o sensor detecta o objeto faz com que o contador incremente de um.**

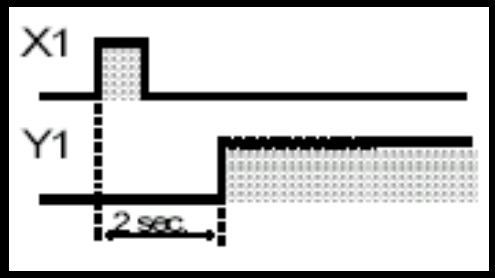
- **Na entrada 0002, quando o sensor detecta o objeto faz com que o contador decremente de um.**

- **Ao atingir 1000 pulsos, a saída 0500 é ativada.**

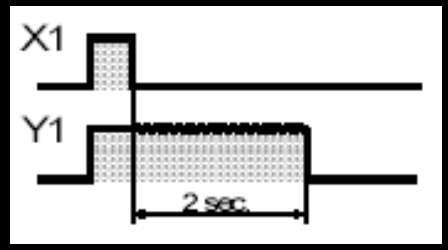
Temporizadores:

Tipos de operação:

On-Delay Timer: após sua entrada ser ativada, ele espera x-segundos para ativar a saída.

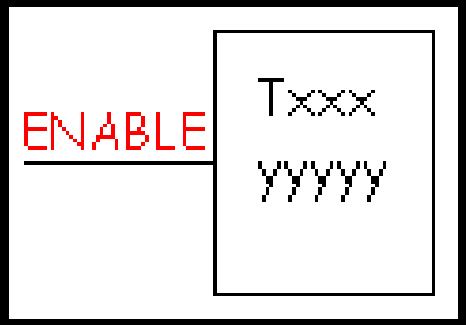


Off-Delay Timer: após sua entrada ser ativada, ele ativa a saída. Após a entrada ser desativada ele mantém a saída ativada por x-segundos antes de desativá-la.



Retentive or Accumulating Timer: necessita de duas entradas. Uma inicializa o evento, a outra zera. Este Timer mantém a contagem corrente quando a entrada deixa de estar ativada e continua quando reativada.

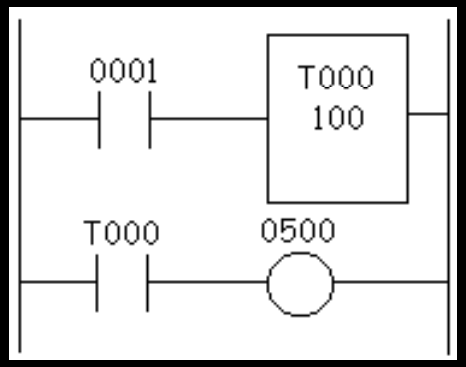
Símbolo da instrução:



Timer do tipo On-Delay.

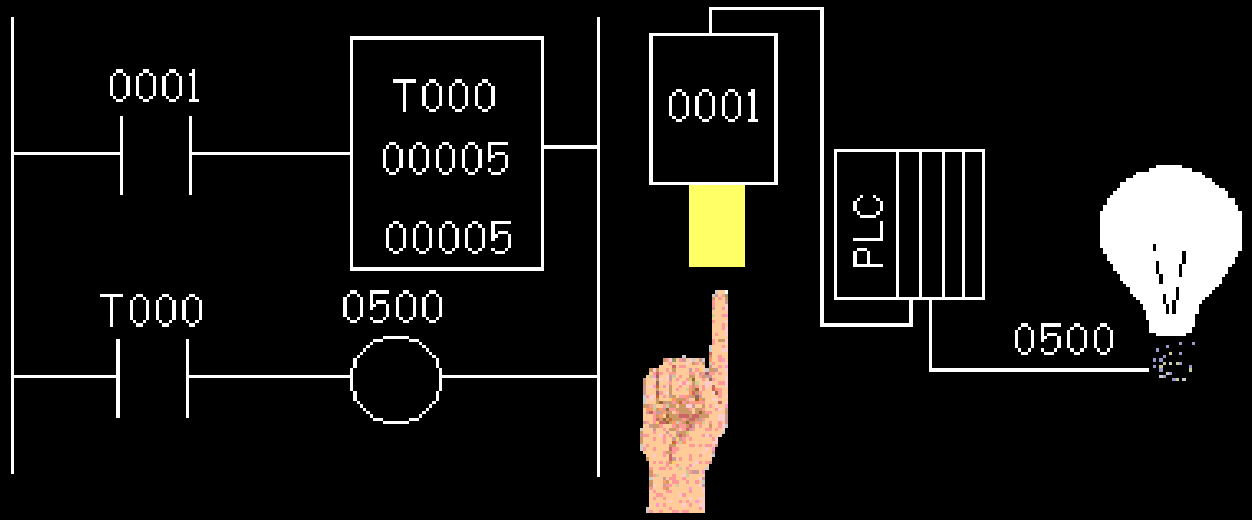
Txxx → Nome do Timer

yyyyy → Número de incrementos de tempo programado para ativar a saída.

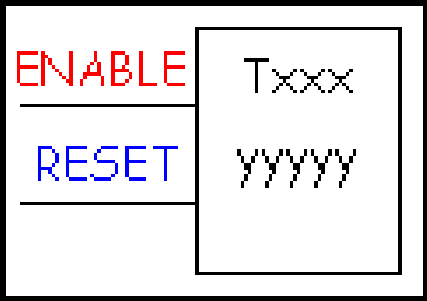


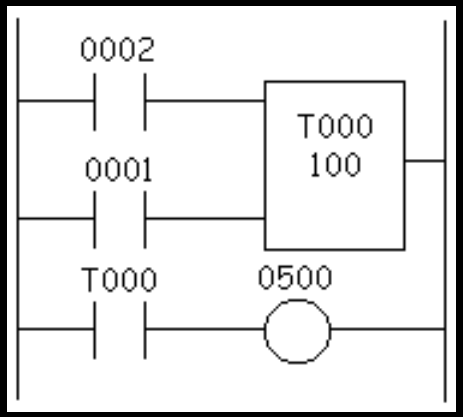
Quando a entrada 0001 é ativada, o Timer 000 começa a contar em incrementos fixos de tempo. Após atingir 100 destes incrementos sua saída ativa a saída 0500.

Se a entrada 0001 desativar antes do tempo programado, o Timer desativa sua saída.


0001
T000
00005
00005
T000
0500
0001
PLC
0500

Accumulating Timer:



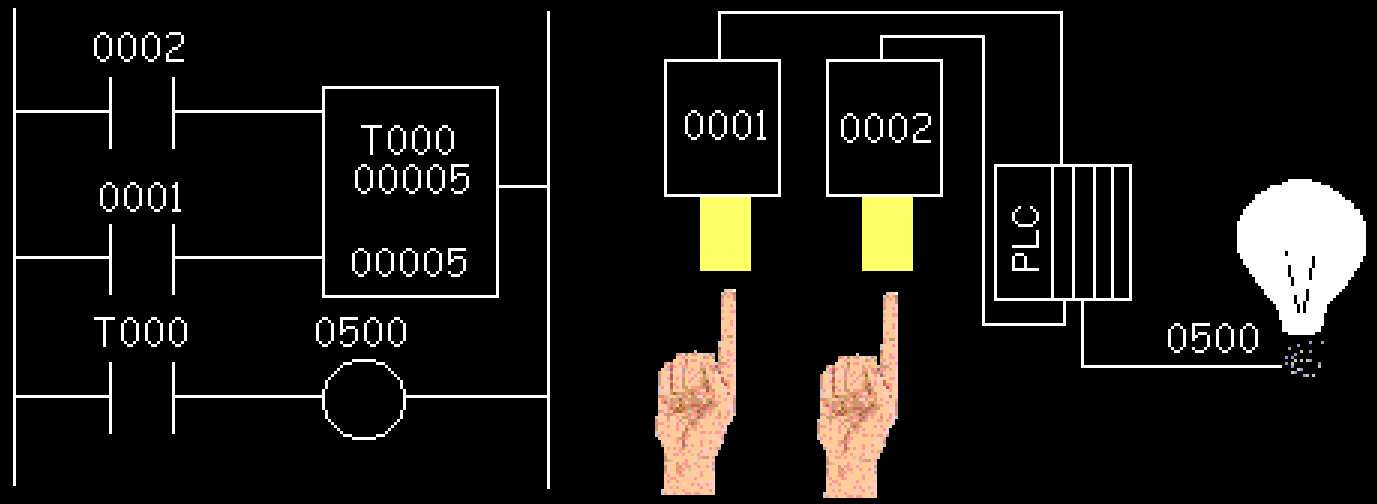


Quando a entrada 0002 do Timer 000 é ativada, ele inicia sua contagem.

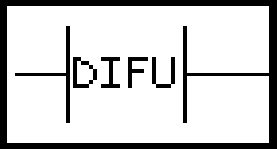
Se a entrada 0002 for desativada o timer mantém a contagem obtida, reiniciando quando a entrada 0002 voltar a ser ativada.

No fim de 100 incrementos de tempo, sua saída é ativada ativando a saída 0500.

Se a entrada 0001 for ativada, o Timer zera a contagem.


0002
T000
00005
0001
00005
T000
0500
0001
0002
PLC
0500

Detector de Borda:



- **Pode ser acionado na descida ou na subida de borda.**

- **Ativo em apenas 1 varredura (scan).**

- **Recebe o nome de:**

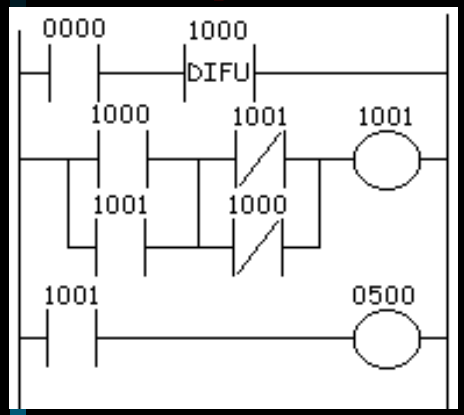
DIFU/DIFD (differentiate up/down)

SOTU/SOTD (single output up/down)

OSR (one-shot rising)

DF (differentiate)

Exemplo:



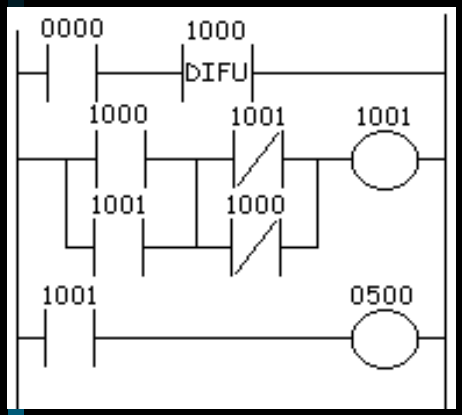
Scan 1

- **Rung 1:** 0000 = True --> 1000(DIFU) = True
- **Rung 2:** 1000(NO) = True , 1001(NO) = False, 1001(NC) = True, 1000(NC) = False
  [1000(NO) & 1001(NC)] = True --> 1001(OUT) = True
- **Rung 3:** 1001(NO) = True --> 500 = True

Scan 2

- **Rung 1:** 0000 = True --> 1000(DIFU) = False
- **Rung 2:** 1000(NO) = False , 1001(NO) = True,
  1001(NC) = False, 1000(NC) = True
  [1001(NO) & 1000(NC)] = True --> 1001(OUT) = True
- **Rung 3:** 1001(NO) = True --> 500 = True

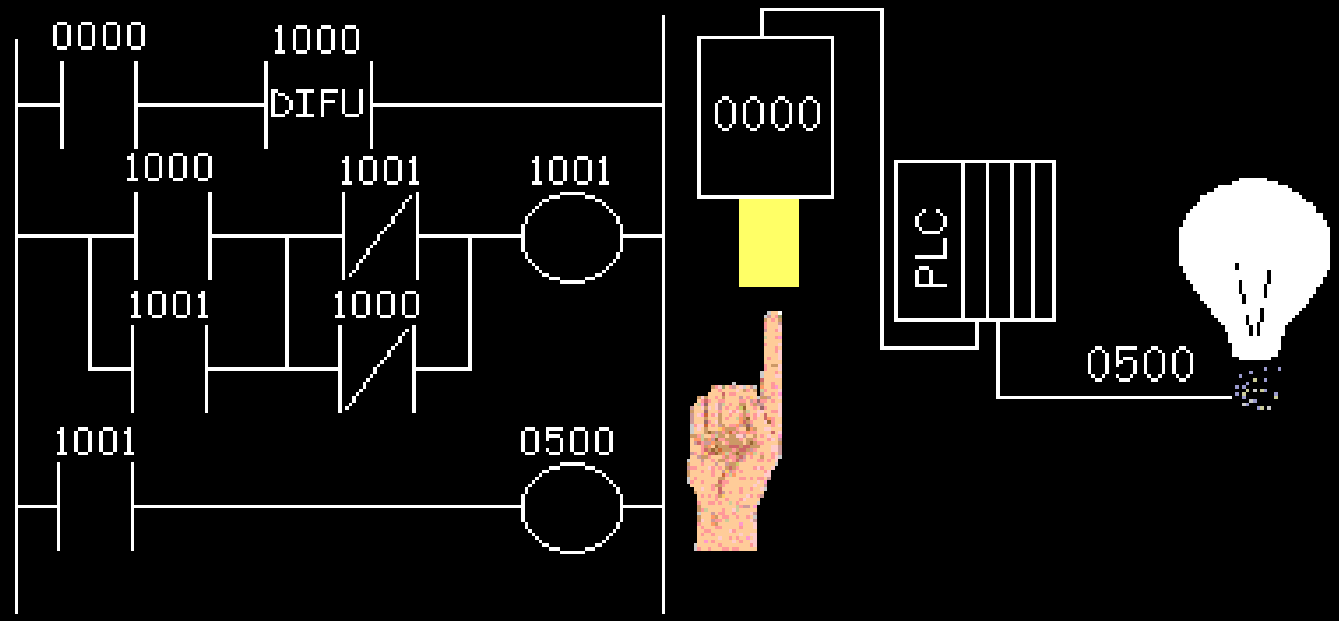
Scan 100



- **Rung 1:** 0000 = False --> 1000(DIFU) = False
- **Rung 2:** 1000(NO) = False , 1001(NO) = True, 1001(NC) = False, 1000(NC) = True
  [1001(NO) & 1000(NC)] = True --> 1001(OUT) = True
- **Rung 3:** 1001(NO) = True --> 500 = True

Scan 101

- **Rung 1:** 0000 = True --> 1000(DIFU) = True
- **Rung 2:** 1000(NO) = True , 1001(NO) = True, 1001(NC) = False, 1000(NC) = False
  [Não existe caminho True] --> 1001(OUT) = False
- **Rung 3:** 1001(NO) = False --> 500 = False


0000
1000
DIFU
1000
1001
1001
1001
1000
1001
0500
0000
PLC
0500

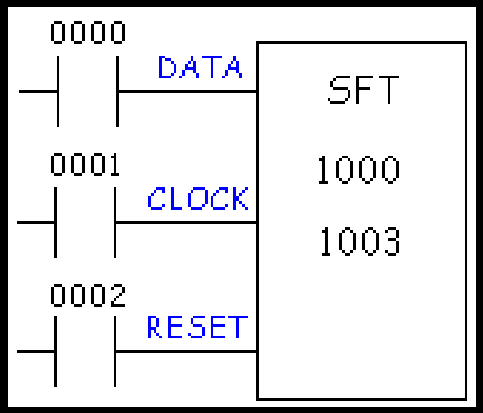
Shift Register:

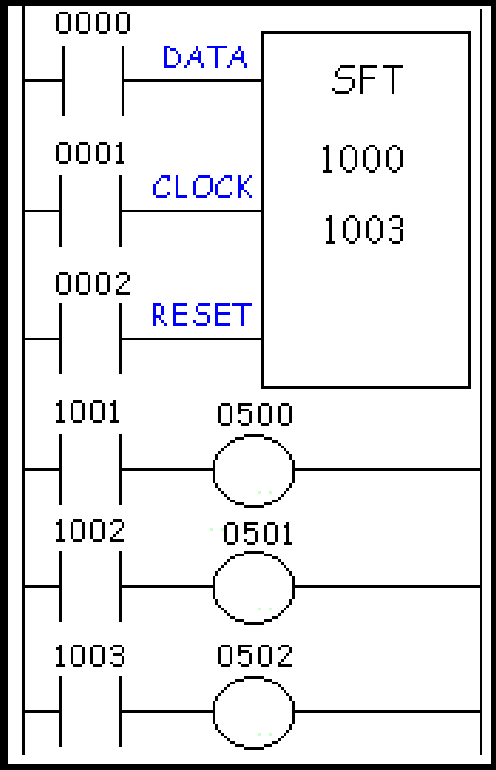
- **Data** - Captura o status (verdadeiro =1 ou falso = 0) na borda do **clock**.

- **Clock** - Desloca o bit dentro do shift register.

- **Reset** - Zera todos os bits dentro do registrador.

- Os endereços 1000 e 1003 são respectivamente o primeiro e o último endereço do bit deslocado.

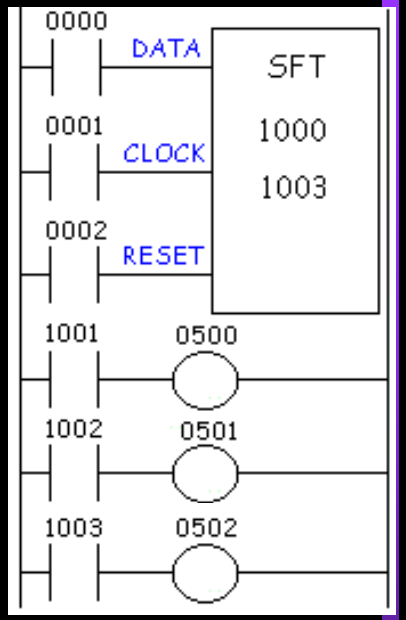




Exemplo: Máquina de encher casquinha de sorvete:

1. Verificar se a casquinha não está quebrada.
2. Colocar sorvete na casquinha (500=ON).
3. Adicionar castanhas (501=ON).
4. Adicionar gotas de chocolate (502=ON)

Se a casquinha estiver quebrada, não colocar nenhum dos itens.



- Um sensor detecta se a casquinha está quebrada (entrada 0000). Se 0000=ON a casquinha está boa
- Um encoder localizado na correia transportadora sincroniza o processo (entrada 0001).
- Um Push button na máquina, zera o resgistrador (entrada 0002)

Endereços dos bits do registrador:

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	0	0	0

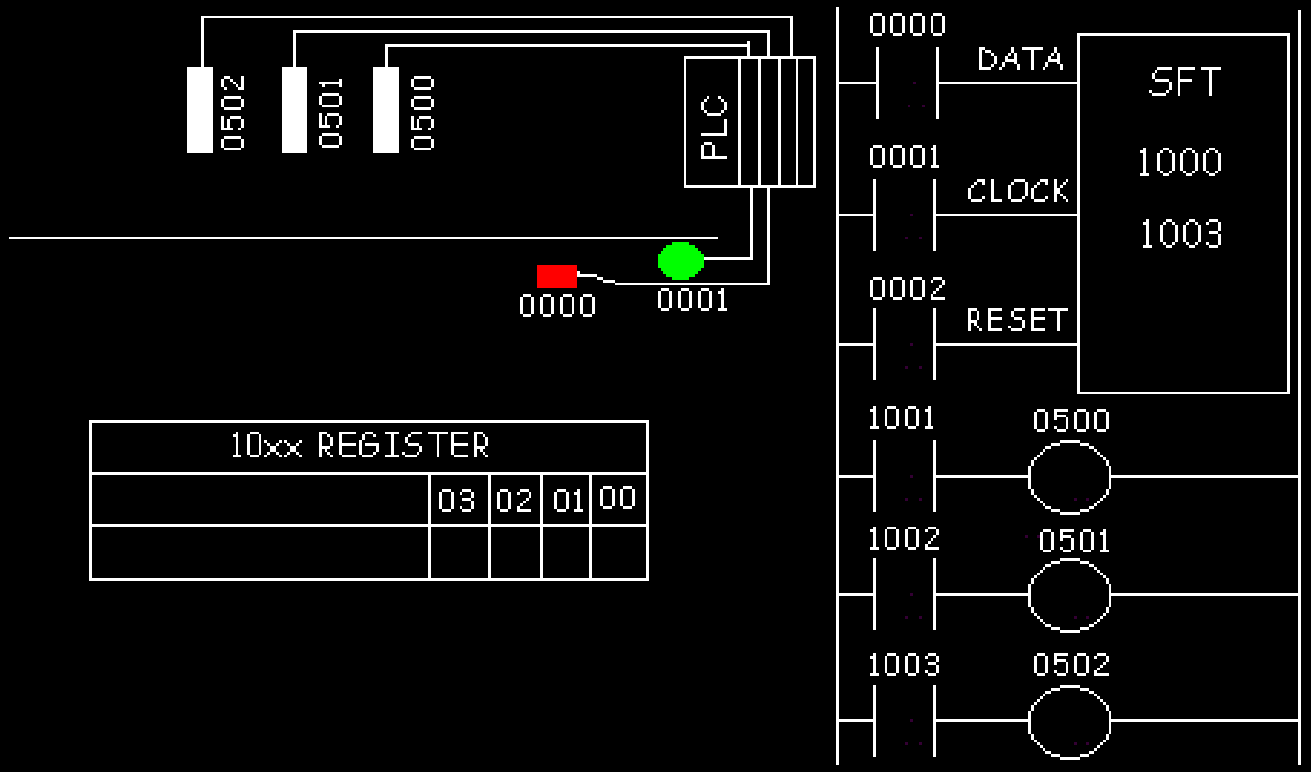
10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	0	0	1

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	0	1	0

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	1	0	1

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												1	0	1	1

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	1	1	0



10xx REGISTER				
	03	02	01	00