



CLP

GPM-18

Manual de Programação

Prefácio

Características do Controlador Lógico-Programável GPM-18

A programação do CLP GPM tem as seguintes características:

- **Suporta 2 tipos de linguagem de programação**
Você pode programar o CLP GPM tanto em linguagem Ladder como em logica Booleana (lista de instruções).
- **Funções básicas**
O CLP GPM suporta não somente funções básicas de intertravamento, mas também funções de alto nível tais como movimentação de dados, comparação, operações aritméticas, loop, shift entre outras. Além disto também dispõe de funções de interrupção, contador alta velocidade , saída rápida para controle de posicionamento e comunicação serial.
- **Endereçamento indireto (Indexação)**
Adicione o indexador após um bit ou registrador de dados (ex.. X3[D100]D0[D100]) para realizar o acesso indireto ao endereço. Por exemplo, quando D100=0, X3[D100] significa X3 , D0[D100] significa D0; quando D100=2, X3[D100] significa X5 , D0[D100] significa D2.
- **Contadores alta velocidade de uma ou duas fases**
Os contadores alta velocidade no GPM dispõe de um dispositivo de interrupção independente para alguns pontos de entrada digital, logo eles são independentes do ciclo de scan, o contador pode receber pulso de até 200KHZ.
- **Instruções de comunicação MODBUS**
Com as instruções de comunicação MODBUS o CLP pode facilmente se comunicar com diversos equipamentos, desde que estes comuniquem no protocolo MODBUS.
- **Saídas de Pulso Alta Velocidade**
O GPM possui 2 saídas de alta velocidade, cada saída pode controlar segmentos em seqüência e os segmentos e números de pulsos podem ser ajustados livremente. A frequência dos pulsos pode atingir 400KHz.

1-1 . Resumo do CLP GPM e forma de programação

Introdução

Controlador lógico-programável GPM

- 18 E/S
- Memória FlashROM
- Relógio/Calendário tempo real – possui bateria de lítio no evento de falta de energia
- Porta de comunicação MODBUS.
- Várias funções de alto nível que facilitam a programação

Formas de Programação

Programação em Lista de Instruções – Boleana

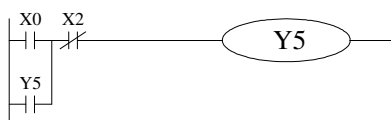
A programação em lista de instruções é a que usa os comandos “LD”, “AND”, “OUT” etc. Este é o formato básico de compilação do programa, porém não é conveniente para seu entendimento.

E.x. :	Passo	Instrução	ID
	0	LD	X000
	1	OR	Y005
	2	ANI	X002
	3	OUT	Y005

Programação em Ladder

Utiliza a lógica de intertravamento de relés que é de mais fácil interpretação do programa. Ao mesmo tempo é possível monitorar a condição do CLP diretamente no desenho do circuito.

Ex. :



O programa elaborado e compilado com uma desta forma de programação é armazenado no CLP. A edição de uma das formas acarretará na conversão automática da outra forma também.

1-2 . Especificações

Item	Especificação
Tensão de Isolação	Maior que 2MΩ – 500VCC
Imunidade a ruído	1000V 1uS pulsos por minuto
Temperatura Ambiente	0°C~60°C
Umidade Ambiente	5%~95%
COM 1	RS-232, Programação do CLP e IHM
COM 2	RS-485, comunicação com inversores, controladores de temper., etc.
Aterramento	Use terra independente

Item		Especificação
		10 Entradas / 8 saídas
Forma de execução do programa		Scan em loop, scan tempo fixo
Forma de programação		Ladder ou lista de instruções
Velocidade de execução		0.5us / passo
Proteção falta de energia		FlashROM e bateria de lítium
Capacidade de programação		8000 passos
Relés internos (M)		8512 pontos
Temporizador (T)	Quantidade	620 temporizadores
	Espec.	Temporiz. 100mS : faixa tempo 0.1~3276.7 segundos Temporiz. 10mS: faixa tempo 0.01~327.67 segundos Temporiz. 1mS: faixa tempo 0.001~32.767 segundos
Contador (C)	Quantidade	635 contadores
	Espec.	Contador 16 bits : K0~32767 Contador 32 bits : K0~2147483647
Registradores de dados (D)		8512 words
Registrador FlashROM (FD)		2048 words
Funções do contador alta velocidade		Contagem alta velocidade, saída de pulsos e interrupção
Ajuste do tempo de scan		0~99mS
Senha de proteção		6 bits ASCII
Funções auto-diagnose		Auto diagnose ao ligar, watchdog e verificação de sintaxe

2-1 . Mapeamento da memória do GP

A memória do CLP GP é dividida em diversos tipos de contatos/bobinas (bits) e registradores de dados. Segue abaixo o significado de cada tipo:

Entrada (X) e Saída (Y)

- As entradas digitais são denominadas por **X**. O GPM-18 possui 10 entradas digitais e elas são mapeadas desde a X0 a X7 e de X10 até a X11. O contato de uma entrada, por exemplo X3 pode ser usado inúmeras vezes no programa.
- As saídas digitais são denominadas por **Y**. O CLP GPM-18 possui 8 saídas digitais e elas são mapeadas desde Y0 até Y7. O contato de uma saída pode ser usado inúmeras vezes no programa, porém sua bobina só pode ser usada uma única vez.

Relé Auxiliar Interno (M)

- Relé auxiliar interno do CLP, é utilizado para elaboração das lógicas de intertravamento. Este tipo de relé não atua as saídas físicas. É só para uso interno.
- Os relés utilizados como retentivos podem manter a condição (ON ou OFF) no caso de desligamento do CLP.

Status (S)

- Relé utilizado para controle de lógica step ladder.
- Quando ele não é utilizado para monitorar o estado da lógica Step ladder poderá ser usado como relé interno.

Temporizador (T)

- Os temporizadores são do tipo retardo e tem bases de tempo de 1ms , 10ms , 100ms etc. Quando o tempo atinge o valor pré-ajustado o contato do temporizador é acionado.
- Os temporizadores T100~T199 tem base de tempo de 100ms, o tempo corrido é acumulativo e caso a lógica antes do temporizador seja desacionada o tempo corrido será acumulado

Contador (C)

- Os contadores são divididos em dois tipos de acordo com seu uso:

Uso interno na lógica: Com ou sem dados retentivos

Contador 16 bits: conta somente para cima de 0 a 32.767

Contador 32 bits: Conta para cima e para baixo de -2.147.483.648 a +2.147.483.647

Estes contadores são usados nas lógicas internas do CLP, usualmente seu tempo de resposta é inferior a 10Hz.

Contador de alta velocidade: Com ou sem dados retentivos

Contador 32 bits: Conta para cima e para baixo de -2.147.483.648 a +2.147.483.647

(contagem unidirecional para cima ou para baixo, contagem bidirecional em fase – encoder)

Utiliza entradas digitais específicas, pode contar pulsos de até 200KHz.

Registrador de Dados (D)

- O registrador de dados serve para armazenar valores dentro do CLP. Todos os registradores do GPM são de 16 bits (o bit mais significativo é o sinal). Combinando dois registradores (double word) pode-se trabalhar com até 32 bits. Os registradores de dados também podem ser retentivos ou não.

Constantes (K) e (H)

- São utilizadas de diversos modos na programação do CLP. O **K** significa valor decimal e o **H** valor hexadecimal. Eles são utilizados em contadores, temporizadores e funções de alto nível.

Ponteiros (P) e (I)

- Os ponteiros são utilizados como derivação ou interrupção do programa. O ponteiro (P) é utilizado nas funções jump ou subrotina. O ponteiro (I) é utilizado nas interrupções de hardware ou de tempo.

2-2 . Lista de alocação de memória

Mnemônico	Nome	Faixa	Pontos
X	Entradas digitais	X000~X007, X010~X011	10 entradas
Y	Saídas digitais	Y000~Y007	8 saídas
M	Relés internos	M0~M2999 【M3000~M7999】	8000
		M8000~M8511 – uso especial	512
S	Step ladder	S0~S511 【S512~S1023】	1024
T	Temporiza- dores	T0~T99 : 100ms não retentivo	620
		T100~T199 : 100ms retentivo	
		T200~T299 : 10ms não accumulation	
		T300~T399 : 10ms retentivo	
		T400~T499 : 1ms não retentivo	
		T500~T599 : 1ms retentivo	
		T600~T618 : 1ms com interrupção de tempo preciso	
C	Contadores	C0~C299 : 16 bits contadores só sobe	635
		C300~C598 : 32 bits Contadores sobe e desce	
		C600~C634 : Contadores alta velocidade	
D	Registradores de dados	D0~D3999 【D4000~D7999】	8000
		Para uso especial D8000~D8511	512
FD	Registradores FlashROM	FD0~FD1535	1536
		Para uso especial FD8000~FD8511	512

◆ NOTAS:

- ✘1. A área de memória marcada com **【 】** é a área retentiva definida na fábrica; os elementos D、M、S、T、C podem ter o tamanho de sua área retentiva mudados. Para detalhes veja a tabela abaixo.
- ✘2. A área de memória FlashROM é toda retentiva e não necessita da bateria para isto.

Ajuste das áreas retentivas:

Elemento	Registrador de ajuste	Função	Valor ajustado de fábrica	Faixa de ajuste
D	FD8202	Início da área de D retentivos	4000	D4000~D8000
M	FD8203	Início da área de M retentivos	3000	M3000~M8000
T	FD8204	Início da área de T retentivos	620	Não ajustável
C	FD8205	Início da área de C retentivos	320	C320~C640
S	FD8206	Início da área de S retentivos	512	S512~S1024

2-3 . Formato dos dados

O CLP GPM tem 5 formas diferentes de apresentação dos valores de acordo com a sua aplicação. Veja abaixo:

《DEC》 DEC : NÚMERO DECIMAL

Ajuste dos valores de temporizadores e contadores (Constante K)

- A identificação do número de relés internos (M), temporizadores (T), contadores (C) e status (S) .
- Determina valor em uma instrução de alto nível (Constante K)

《HEX》 HEX : NÚMERO HEXADECIMAL

- Mesmo que o número DEC, este é usado para determinar valor em uma instrução de alto nível (Constante H)

《BIN》 BIN : NÚMERO BINÁRIO

Os números utilizados na programação do CLP são DEC ou HEX, porém internamente eles são transformados em binários. Ao se monitorar um valor pode-se fazê-lo em DEC ou HEX.

《OCT》 OCT : NÚMERO OCTAL

- Os pontos de entrada e saída digital do CLP são identificados por número octal, por ex , 1-7 , 10-17 , . . . 70-77 , 100-107. As entradas do CLP são de X0 a X7, X10 e X11.

《código BCD》 BCD : BINARY CODE DECIMAL

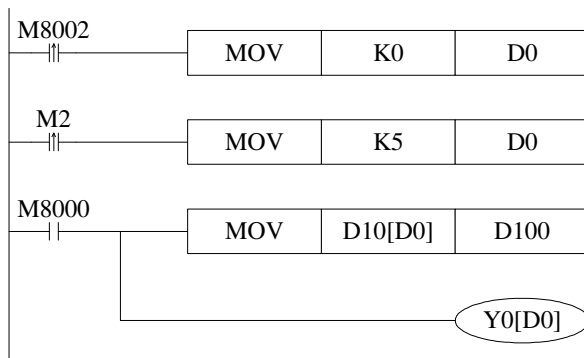
- O BCD é um método de numeração que utilize 4 bits para representar um número decimal de 0~9

《Ponto flutuante (float)》

- O CLP GPM tem a capacidade de executar operação de alta precisão com ponto flutuante.

2-4 . Alguns princípios de acesso a registradores e bits

1、 Um registrador de dados pode ser usado como indexador (endereçamento indireto)



Neste exemplo, quando $D0=0$, então $D100=D10$, e $Y0$ é ON ;

Quando M2 vai de OFF para ON, $D0=5$, então $D100=D15$, e $Y5$ é ON.

Quando $D10[D0]=D[10+D0]$, $Y0[D0]=Y[0+D0]$.

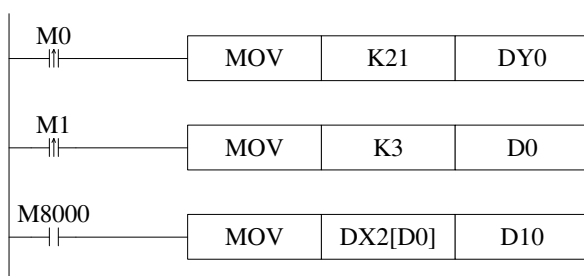
- Indexação de registradores: $DXn[Dm]$ significa $DX[n+Dm]$;
- A indexação de variáveis só é feita usando-se o registrador D.

2、 Acesso a bits através de registrador

As entradas X, saídas Y, relés internos M podem ser acessados através de registradores de 16 bits. Por ex. $DX0$ significa $X0\sim X17$ na forma de registrador de 16 bits. $DM20$ significa $M20\sim X37$ na forma de reg. de 16 bits.

Formato : Adicione D antes do mnemônico do bit. Ex. $DY0$

Registradores de combinação de bits disponíveis : DX, DY, DM, DS, DT, DC



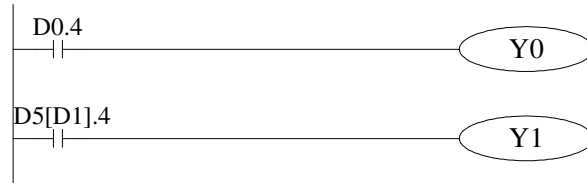
No exemplo acima, quando M0 vai de OFF para ON, o valor do registrador $DY0$ composto pelos bits $Y0\sim Y17$ fica igual a 21 , i.e. $Y0$, $Y2$ e $Y4$ vão para ON.

Antes de M1 ser ativado, quando $D0=0$, $DX2[D0]$ significa um registrador composto por $X2\sim X21$;

Quando M1 vai de OFF para ON, $D0=3$, então $DX2[D0]$ significa um registrador composto por $X5\sim X24$

3、 Acesso ao bit de um registrador

Formato : Dn.m



No exemplo acima, D0.4 significa que quando o bit No.4 de D0 é 1, Y0 é ON ;

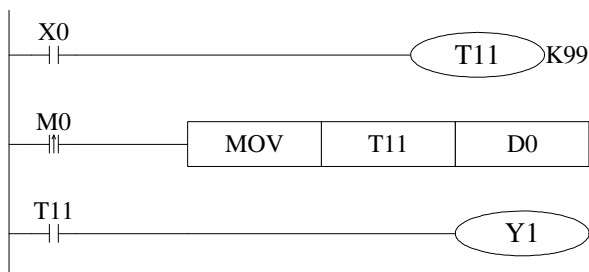
D5[D1].4 significa acesso a bit com indexação, se D1=5, isto diz que D5[D1] significa o 4º bit de D10.

- O acesso a bit de registrador com indexação é : Dn[Dm].x

4、 Diferenças entre interpretação das variáveis T e C

Para os registradores de temporizadores (T) e contadores (C), Tn/Cn pode significar registrador de dados ou bit dependendo da instrução aonde é aplicado.

T e C podem indicar o status do temporizador ou contador (no caso de bit), ou pode significar o valor atual do temporizador ou contador (no caso de registrador).



No exemplo acima , MOV T11 D0 , T11 é um registrador de dados ;

LD T11 , T11 é um bit.

5、 Interrupções: P , I

P significa o ponteiro utilizado nas funções J e/ou CALL.;

I é o ponteiro da interrupção.

3-1 . Temporizadores e funções [T]

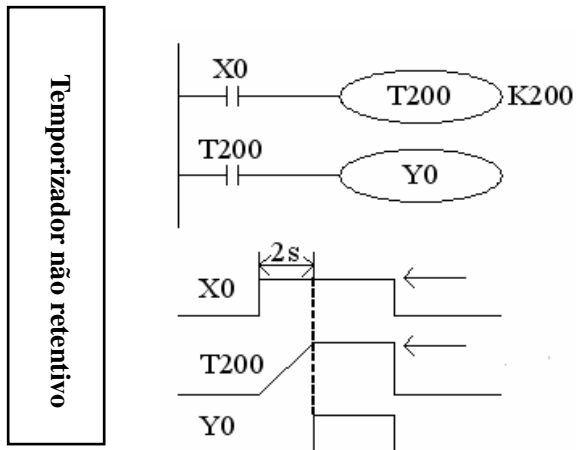
Número de temporizadores

Veja abaixo na tabela os temporizadores [T] possíveis de serem utilizados.

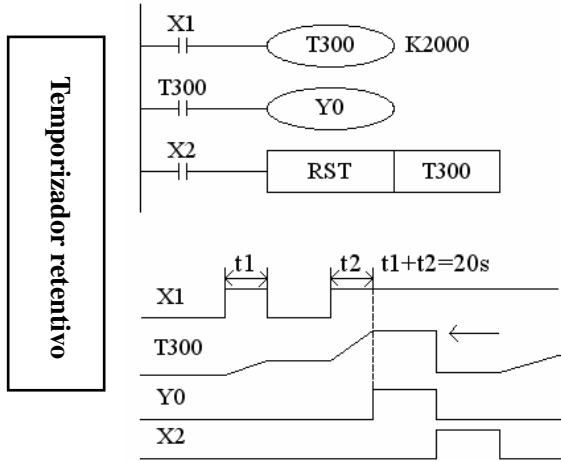
100ms não retentivo (16 bits)	T0~T99
100ms retentivo (16 bits)	T100~T199
10ms não retentivo (16 bits)	T200~T299
10ms retentivo (16 bits)	T300~T399
1ms não retentivo (16 bits)	T400~T499
1ms retentivo (16 bits)	T500~T599

Funções

Os temporizadores recebem pulsos de 1ms, 10ms, 10ms internamente no CLP. Quando o valor escolhido é alcançado, o sinal de saída é ativado. Use este sinal para acionar uma saída. Para determinar um tempo fixo utilize uma constante K no programa, É possível utilizar um registrador (D) para definir um tempo variável.



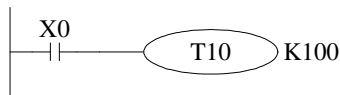
Se a entrada X000 estiver acionada, o temporizador T200 inicia a contagem do tempo. O T200 é um temporizador de 10ms, ou seja, o valor do tempo desejado terá que ser multiplicado por esse tempo. Se o valor escolhido for igual à K200, a saída do temporizador ocorrerá após 2 segundos ($200 \cdot 10\text{ms} = 2\text{s}$). O temporizador só temporiza enquanto existir a entrada X000, caso o sinal de entrada vá para o nível baixo o temporizador é zerado.



Se a entrada X001 estiver habilitada o temporizador T300 começa a contar o tempo. Quando o tempo é alcançado ($K2000 \times 10\text{ms} = 20\text{s}$), a saída do T300 é acionada. Observe que durante a contagem do tempo, a entrada X001 é desabilitada, mas o tempo não é zerado devido a retenção do valor. Quando habilitada novamente o tempo recomeça de onde parou, observe a figura. Quando a entrada X002 é habilitada, o tempo e a saída do temporizador são zerados.

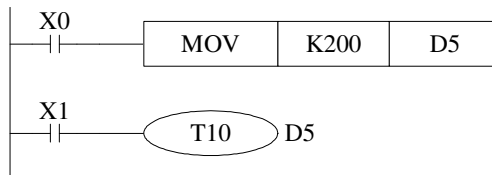
Método de uso do temporizador

«Valor constante (K)»



T10 é um temporizador com unidade de tempo de 100ms. K100 é uma constante de multiplicação, logo $0.1\text{s} \times 100 = 10\text{s}$ tempo de trabalho.

«Valor indireto (D)»



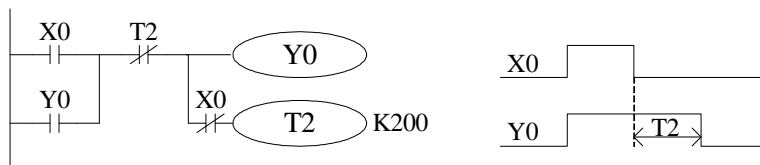
O valor indireto consiste na entrada do tempo para o temporizador através de um registrador (D). Veja o exemplo ao lado.

Valore de referência

Existem 600 temporizadores (retentivos e não retentivos, conforme tabela acima) do T0~T599 de 16 “bits” que abrange os valores de 0 á 32767. Se o valor do temporizador for superior á 32767, o temporizador deixará de cronômetrar e seu estado permanecerá o mesmo.

Exemplos

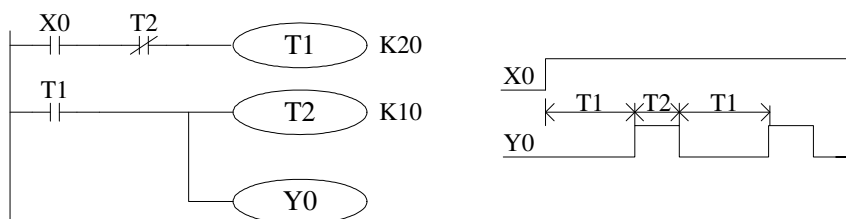
«Saída com retardo na desenergização»



Quando X000 é habilitada, a saída Y000 é acionada ;

Se X000 é desabilitada (1->0), O temporizador T2 de 20 segundos inicia a

《Pisca - pisca》



Se X000 ativado, Y000 é alternado de acordo com os tempos definidos.

T1 controla o nível baixo de Y000, T2 controla o nível alto de Y000. A saída é alternada enquanto existir o sinal alto de X000.

3-2 . Contadores e funções [C]

Números de contadores

Veja na tabela abaixo os contadores (C), existentes no CLP.

Contador positivo de 16 “bits”	C000~C299
Contadores positivos e negativos de 32 “bits”	C300~C598 (C300, C302... C598). (utilizar sempre múltiplos de 2). Os contadores podem ser ímpares.
Contador de alta velocidade	C600~C634 (C600, C602...C634). (utilizar sempre múltiplos de 2). Os contadores podem ser ímpares.

Características dos contadores

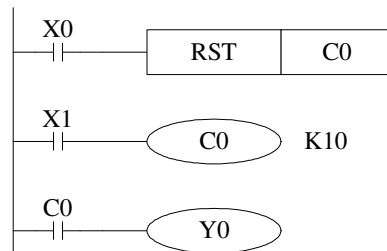
Veja abaixo na tabela algumas características dos contadores de 16 “bits” e 32 “bits”:

Tamanho	Contador 16 bits	Contador 32 bits
Direção da contagem	Positiva	Positiva/negativa
Valores min/máx	De 1 à 32767	De -2,147,483,648 à +2,147,483,647
Tipo de dados	Constante K ou registradores	Constantes K ou registradores, mas, aos pares.
Saídas	Aciona a saída após o alcançar o valor escolhido.	Incrementa ou decrementa.
Reset	Quando executado o comando de Reset , O contador retorna a zero e sua saída retorna ao valor de origem.	

Função

Contadores de 16 "bits" – Uso comum do contador

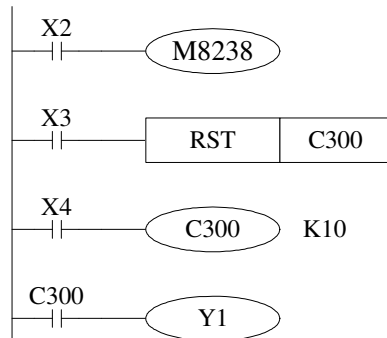
O contador de incremento de 16 "bits" pode ser ajustado para valores de K1 a K32767 (constante decimal).



Quando o CLP é alimentado, os valores dos contadores são colocados com o valor 0, salvo o contador que utiliza dados retentivos. O contador que utiliza dados retentivos mantém os mesmos valores que tinha antes de ser desenergizado.

- A cada pulso da entrada X001 em C0, o contador é incrementado. Quando a contagem alcançar o valor escolhido "K10", ou a décima contagem, a saída é ativada. Quando o valor da contagem é alcançado, os pulsos de X001 não incrementarão mais o contador ficando com o valor escolhido.
- Para resetar, basta dar um pulso na entrada X000 em série com a instrução RST, o valor do contador é inicializado e a saída volta ao seu estado inicial.
- Para ajustar o valor do contador, insira um valor através de uma constante "K" ou utilize um registrador interno do CLP, movendo o valor desejado para o registrador escolhido.

Para os contadores de incremento de 32 “bits”, com limites de K1 a K2.147.483,647 (constante decimal). É necessário o uso do relé especial M8238, que direciona a contagem em positivo/negativo (C300~C498). Veja o exemplo abaixo.

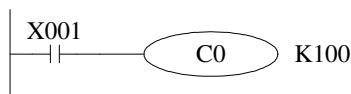


- Se a entrada X2 habilita M8238, então a contagem será decrescente. Caso contrário, a contagem será crescente.
- Escolha um valor positivo para o contador através de uma constante K ou de um registrador D. Esse valor deverá estar dentro da faixa de abrangência dos 32 “bits”. Para um registrador, exemplo D0, utilize D0 e D1 para formar os 32 “bits” de dados. Se a entrada X004 for pulsada no contador C300, a contagem é incrementada ou decrementada de acordo com X2.
- Quando a entrada X3 é habilitada, a instrução RST é executada, o contador é inicializado e a saída também.
- Para os contadores retentivos, o valor e a saída do contador não são perdidos quando o CLP for desenergizado.
- Você também poderá utilizar o contador de 32 “bits” como um registrador de 32 “bits”. Mas não é possível usá-lo com uma instrução de 16 “bits”

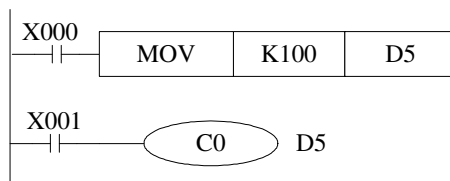
Exemplos

◆ Contador de 16“bits”

«Valor constante (K)»



«Usando um registrador (D) , através da instrução MOV»

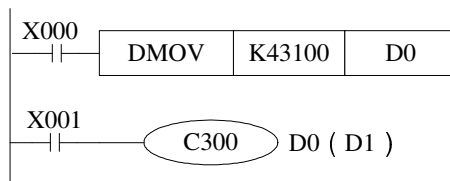


◆ **Contador de 32 “bits”**

«Valor constante (K)»



«Usando um registrador (D) , através da instrução DMOV»



Entrada de valores

A contagem dos contadores T0 á T599 é de 16 “*bits*”, linear incremental que varia de K0 á K32767. Caso o valor escolhido seja alcançado K32767, o contador para a contagem, e seus estados são inicializados.

3-3 . Informações adicionais

«Ordem de ação de entrada/saída, relé e tempo de resposta»

➤ Disposição das entradas

Antes de ser executado o programa no CLP, todas as entradas são colocadas para 0 na área de imagem. Durante a execução do programa, mesmo que o estado de uma entrada mude o conteúdo da memória imagem das entradas não será alterado. No entanto este conteúdo será alterado no scan seguinte.

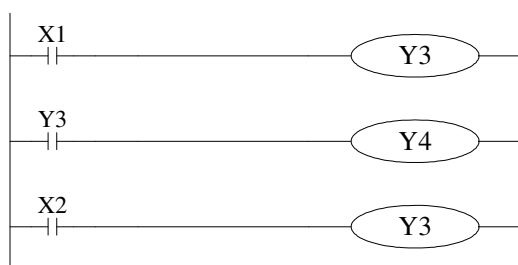
➤ Disposição das saídas

Uma vez realizada a execução de todas as instruções, os dados de “*bits*” são transferidos da área de imagem para a área de memória. Estas serão as saídas atuais no CLP. As saídas físicas do CLP são modificadas em função do tempo de resposta.

«Não é aceito pulso de sinal de entrada rápida»

Um pulso de entrada não pode ser maior que o tempo de ciclo. Se considerarmos um tempo de entrada com filtro de 10ms, em um ciclo também de 10ms, então uma entrada necessitará de 20ms por borda. Logo, temos $1000 / (20+20) = 25\text{Hz}$ de limite de entrada. Mas, esta condição pode ser ajustada para algumas entradas através instruções de função especial.

«Saída duplicada»



Como mostrado na figura ao lado, veja a saída Y003 sendo utilizado numa duplicidade por duas entradas, vamos supor que as entradas se encontram na seguinte condição:

Ex. X001="1", X002="0"

No primeiro instante, X001="1", este valor é transportado para a área de imagem, A saída Y004 é atualizada neste instante para "1", porque Y003 da condição para isto.

Mas, a próxima linha faz da entrada X002="0", e a área de imagem de Y003="0".

Então, os estados das saídas sempre serão: Y003="0" e Y004="1".

Sempre que necessitar utilizar uma saída duplicada, mantenha uma prioridade na linha ou utilize uma lógica combinacional para executar a instrução.

4 . Instruções de programação básicas

Neste capítulo, descrevemos algumas instruções básicas e funções.

4-1 . Lista de instruções básicas.

4-2 . **【LD】** , **【LDI】** , **【OUT】**

4-3 . **【AND】** , **【ANI】**

4-4 . **【OR】** , **【ORI】**

4-5 . **【LDP】** , **【LDF】** , **【ANDP】** , **【ANDF】** , **【ORP】** , **【ORF】**

4-6 . Instruções de comparação

4-7 . **【ORB】**

4-8 . **【ANB】**

4-9 . **【MCS】** , **【MCR】**

4-10 . **【ALT】**

4-11 . **【PLS】** , **【PLF】**

4-12 . **【SET】** , **【RST】**

4-13 . **【OUT】** , **【RST】** (Para uso com os contadores também).

4-14 . **【NOP】** , **【END】**



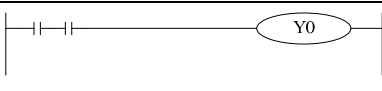
4-15 . Entre outros...

4-1 . Identificando as instruções básicas.**GPM-18 Tabela de instruções básicas**

Mnemônico	Função	Formato e Registros
LD (Contato aberto)	Operação inicial lógica contato NA (Normalmente aberto)	X, Y, M, S, T, C, Dn.m, FDn.m
LDI (Contato fechado)	Operação inicial lógica contato NF (Normalmente fechado)	X, Y, M, S, T, C, Dn.m, FDn.m
LDP (Pulso de borda)	Operação inicial lógica de borda pulso de subida	X, Y, M, S, T, C, Dn.m, FDn.m
LDF (Pulso de borda)	Operação inicial lógica de borda pulso de descida	X, Y, M, S, T, C, Dn.m, FDn.m
AND (E)	Lógica combinacional de contatos NA (Normalmente aberto)	X, Y, M, S, T, C, Dn.m, FDn.m
ANI (E-Invertida)	Lógica combinacional de contatos NF (Normalmente fechado) contacts	X, Y, M, S, T, C, Dn.m, FDn.m
ANDP (E pulsado, subida)	Lógica combinacional de contatos por borda de subida	X, Y, M, S, T, C, Dn.m, FDn.m
ANDF (E pulsado, descida)	Lógica combinacional de contatos por borda de descida	X, Y, M, S, T, C, Dn.m, FDn.m
OR (OU)	Lógica combinacional de contatos NA (Normalmente aberto)	X, Y, M, S, T, C, Dn.m, FDn.m
ORI (OU-Invertida)	Lógica combinacional de contatos NF (Normalmente aberto)	X, Y, M, S, T, C, Dn.m, FDn.m
ORP (OU-Pulsado, subida)	Lógica combinacional de contatos por borda de subida	X, Y, M, S, T, C, Dn.m, FDn.m
ORF (OU-Pulsado, descida)	Lógica combinacional de contatos por borda de descida	X, Y, M, S, T, C, Dn.m, FDn.m
ANB (Múltiplos E)	Lógica combinacional de contatos de entradas múltiplas	Não tem
ORB (Múltiplos OU)	Lógica combinacional de contatos de múltiplas entradas	Não tem
OUT (Saída)	Operação final de uma linha de instrução	Y, M, S, T, C, Dn.m
SET (Seta)	Condiciona um “bit” permanentemente “1”	Y, M, S, T, C, Dn.m
RST (Reset)	Condiciona um “bit” permanentemente “0”	Y, M, S, T, C, Dn.m

PLS (Detecção de borda subida)	Atua uma vez na borda de subida	X, Y, M, S, T, C, Dn.m
PLF (Detecção de borda descida)	Atua uma vez na borda de descida	X, Y, M, S, T, C, Dn.m
MCS (Início de nova linha)	Nova linha de instruções em série	Não tem
MCR (Retorno da linha inserida)	Retorno da linha em série inserida	Não tem
ALT (Alterna estado)	Alterna os estados da saída do dispositivo	X, Y, M, S, T, C, Dn.m
NOP (Sem operação)	Não executa durante o tempo da instrução	Não tem
END (FIM)	Força o programa para o final do “ <i>scan</i> ”	Não tem

4-2 . **[LD]** , **[LDI]** , **[OUT]**

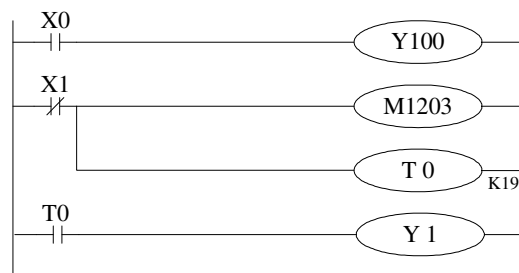
Mnemonic	Mnemônico	Função	Formato e registros
	LD (Contato aberto)	Lógica de contato aberto NA (Normalmente aberto)	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	LDI (Contato fechado)	Lógica de contato fechado NF (Normalmente fechado)	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	OUT (Saída)	Lógica de saída física, sinal ou relé interno.	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>

Modo de usar

- Coloque as instruções, por exemplo, (LD e LDI) do lado esquerdo da barra de linha, ou utilize um novo bloco usando a instrução ANB.
- A instrução OUT representa uma saída física, um relé interno auxiliar, temporizador, contador, etc. Não é possível usar uma entrada na instrução de saída.
- Não é possível usar um comando de saída paralela por muitas vezes.
- Para os temporizadores e ou contadores, depois de escolher a instrução de saída é necessário inserir uma constante K.
- O limite das variáveis deve ser respeitado. Veja abaixo na tabela.

Temporizador /Contador	Abrangência	Valores respectivos
Tempo de 1ms	1 ~ 32.767	0.001 ~ 32.767 segundos
Tempo de 10ms		0.01 ~ 32.767 segundos
Tempo de 100ms		0.1 ~ 32.767 segundos
Contador de 16 “bits”	1 ~ 32.767	Igual a abrangência
Contador de 32 “bits”	1 ~ 2.147.483.647	Igual a abrangência

Programa



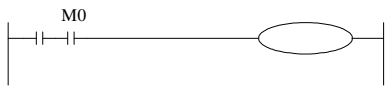

```

LD    X0
OUT   Y100
LDI   X1
OUT   M1203
OUT   T0
SP    K19
LD    T0
OUT   Y1

```

4-3 . 【AND】 , 【ANI】

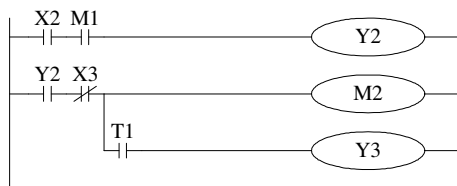
Mnemonic

Mnemônico	Função	Formato e registros
AND (E)	Lógica “E” de contato NA (Normalmente aberto)	 Registros :X, Y, M, S, T, C, Dn.m, FDn.m
ANI (E-Invertida)	Lógica “E-Invertida” de contato NF (Normalmente fechado)	 Registros :X, Y, M, S, T, C, Dn.m, FDn.m

Descrição

- Utilize as instruções “E” ou “E-Invertida” para uma seqüência de lógicas combinacional.
- Uma lógica combinacional “E” permite que uma saída ou uma instrução, seja executada somente se as entradas estiverem em nível alto (para um contato NA) ou nível baixo (para um contato NF) forem verdadeiros. Veja o exemplo abaixo.



Programa



```

LD    X2
AND   M1
OUT   Y2
LD    Y2
ANI   X3
OUT   M2
AND   T1
OUT   Y3
    
```

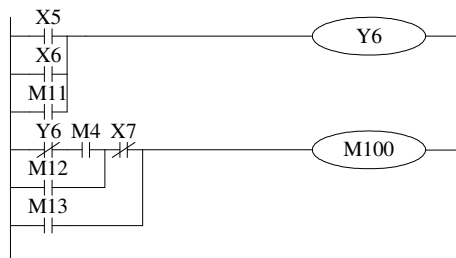

4-4 . **[OR]** , **[ORI]**

Mnemônico e Função	Mnemônico	Função	Formato e registros
Mnemônico e Função	OR (OU)	Lógica combinacional “OU” contato NA (Normalmente aberto)	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	ORI (OU-Invertida)	Lógica combinacional “OU” contato NF (Normalmente fechado)	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>

Descrição

- Use a instrução OR e ORI, quando houver a necessidade de mais de uma entrada dar condição de acionar uma saída ou uma função. Para inserir mais de um contato em uma lógica OU utilize a instrução ORB. Veja o exemplo abaixo.

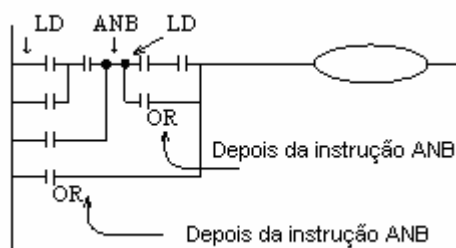
Programa



```

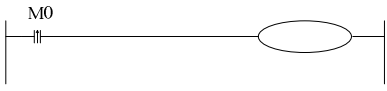

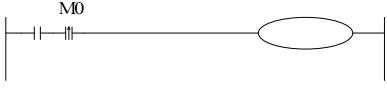



LD    X5
OR    X6
OR    M11
OUT   Y6
LDI   Y6
AND   M4
OR    M12
ANI   X7
OR    M13
OUT   M100
    
```

Relacionando com ANB



Uma ligação em paralelo com as instruções OR, ORI, e ANB é necessário usar as instruções LD, LDI sempre no início das instruções.

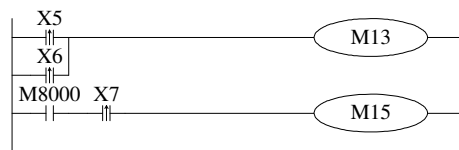
4-5 . **【LDP】** , **【LDF】** , **【ANDP】** , **【ANDF】** , **【ORP】** , **【ORF】**

Mnemônico e Função	Mnemônico	Função	Formato e registros
	LDP (Pulso de borda)	Operação inicial lógica contato NA (Normalmente aberto)	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	LDF (Pulso de borda)	Operação inicial lógica de borda Pulso de descida	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	ANDP (E pulsado, subida)	Lógica combinacional de contatos por borda de subida	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	ANDF (E pulsado, descida)	Lógica combinacional de contatos por borda de descida	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	ORP (OU-Pulsado, subida)	Lógica combinacional de contatos por borda de subida	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>
	ORF (OU-Pulsado, descida)	Lógica combinacional de contatos por borda de descida	 <p>Registros : X, Y, M, S, T, C, Dn.m, FDn.m</p>

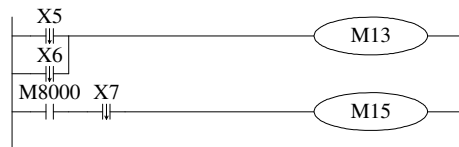
Descrição

- LDP, ANDP, ORP São ativados pelo “scan” quando mudam o estado de ”0” para “1”.
- LDF, ANDF, ORF São ativados pelo “scan” quando mudam o estado de ”1” para “0”.

Programa



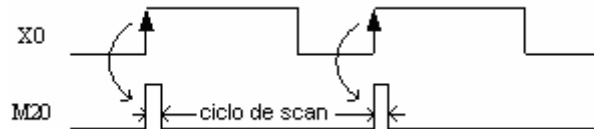
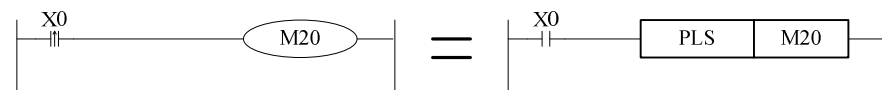
```
LDP X5
ORP X6
OUT M13
LD M8000
ANDP X7
OUT M15
```



```
LDF X5
ORF X6
OUT M13
LD M8000
ANDF X7
OUT M15
```

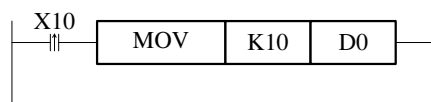
No diagrama acima quando X005 ~ X007 mudar o estado de "1" para "0" ou de "0" para "1", M13 ou M15 é ativado no primeiro scan do programa.

Saída

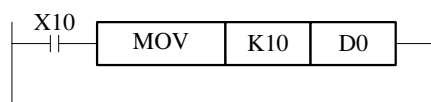


Nas duas condições, quando X0 mudar de "0" para "1", M20 é ativado no ciclo de "scan".

NOTA :




Quando X10 mudar de "0" para "1", a instrução MIV é executada apenas uma vez.



No exemplo ao lado, quando X10 mudar de "0" para "1", a instrução MOV é executada a cada ciclo de "scan".

4-7 . 【ORB】

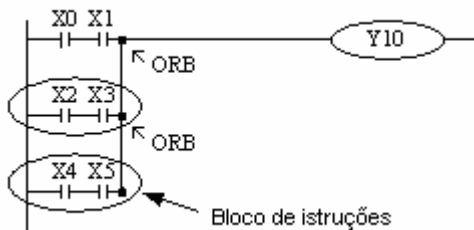
Mnemônico e Função

Mnemônico	Função	Formato e registros
ORB (Múltiplos OU)	Lógica combinacional de contatos de múltiplas entradas	 <p>Registros : Não tem</p>

Descrição

- Declarar no início da lógica, quando necessitar de múltiplas entradas.
- Uma instrução ORB é independente e não é associada com nenhum registrador.
- Sem limites para o uso da instrução ORB.
- Quando usar a instrução ORB em grandes quantidades, não ultrapassar de 8 instruções de LD e LDI nas definições dos blocos do programa (em ligações paralelas).

Programa




Método de programação sequencial recomendada:

```
LD X0
AND X1
LD X2
AND X3
ORB
LDI X4
AND X5
ORB
OUT Y10
```

Método de programação sequencial **não** recomendada:

```
LD X0
AND X1
LD X2
AND X3
LDI X4
AND X5
ORB
ORB
OUT Y10
```

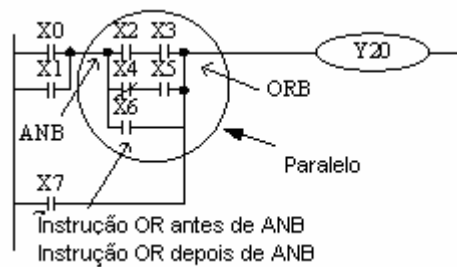
4-8 . 【ANB】

Mnemônico	Função	Formato and registros
ANB (Múltiplos E)	Lógica combinacional de contatos de entradas múltiplas	 Registros : Não tem

Descrição

- Para declarar o ponto inicial do circuito utilize as funções LD ou LDI. Depois de completar o circuito paralelo conecte-o com a instrução ANB.

Programa

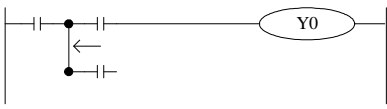



```

LD    X0
OR    X1
LD    X2      Início do ramo
AND   X3
LDI   X4      Início do ramo
AND   X5
ORB
OR    X6      Fim do bloco paralelo
ANB
OR    X7      Fim do bloco paralelo
OUT   Y20     Lógica "E"

```

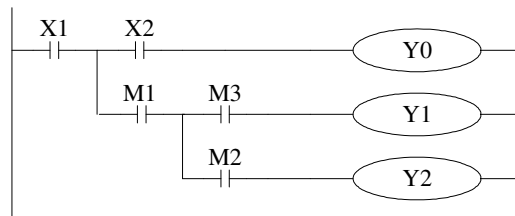
4-9 . 【MCS】 , 【MCR】

Mnemônico	Função	Formato e registros
MCS (Início de nova linha)	Nova linha de instruções em série	 Registros : Não tem
MCR (Retorno da linha inserida)	Retorno da linha em série inserida	 Registros : Não tem

Descrição

- Após a execução da instrução MCS, indica que uma nova linha será criada a partir da instrução (LD, LDI). Para retornar para a linha original use a instrução MCR.
- MCS, MCR são instruções para serem usados em conjunto.
- A lógica deve ser implementada entre as instruções MCS, MCR. O máximo de níveis de linha que podem ser inseridos, não pode ultrapassar de 10.

Programa



```

LD      X1
MCS
LD      X2
OUT     Y0
LD      M1
MCS
LD      M3
OUT     Y1
LD      M2
OUT     Y2
MCR
MCR
    
```


Início da segunda linha

Terceira linha

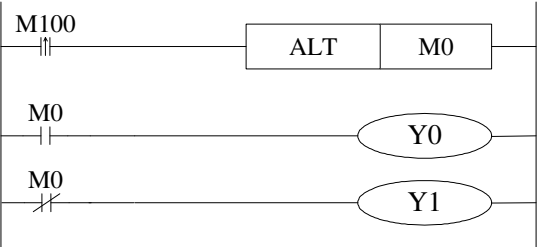
Retorno da terceira linha

Retorno da segunda linha

4-10 . 【ALT】

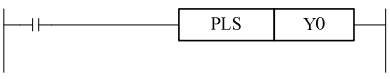
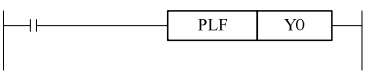
Mnemônico e função	Mnemônico	Função	Formato e registros
	ALT (Alterna estado)	Alterna os estados da saída do dispositivo	 Registros : Y, M, S, T, C, Dn.m

Descrição	
	É alternado o estado do dispositivo de destino em toda operação da instrução de ALT.

Programa		
	<pre> LDP M100 ALT M0 LD M0 OUT Y0 LDI M0 OUT Y1 </pre>	

4-11 . [PLS] , [PLF]

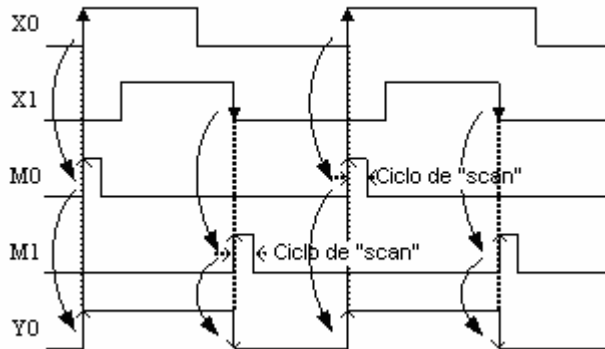
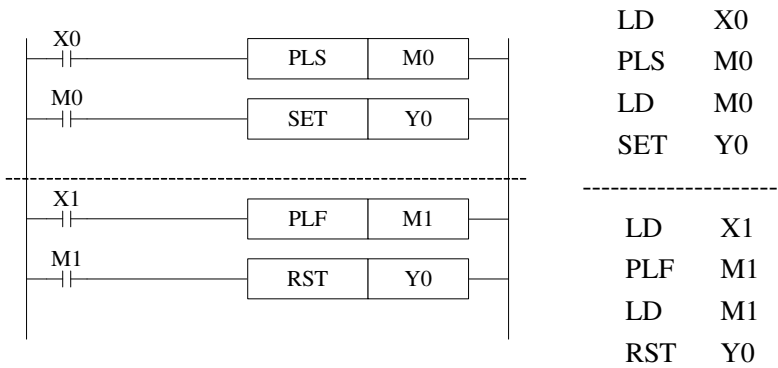
Mnemônico e Função

Mnemônico	Função	Formato e registros
PLS (Detecção de borda subida)	Atua uma vez na borda de subida	 Registros : Y, M, S, T, C, Dn.m
PLF (Detecção de borda descida)	Atua uma vez na borda de descida	 Registros : Y, M, S, T, C, Dn.m

Descrição


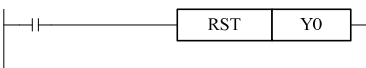
- Quando a instrução PLS é executada, as saídas atuam na borda de subida, ou, são modificadas.
- Quando a instrução PLS é executada, as saídas atuam na borda de descida, ou, são modificadas.
- Utilizado para capturar apenas um sinal por scan.

Programa



4-12 . [SET] , [RST]

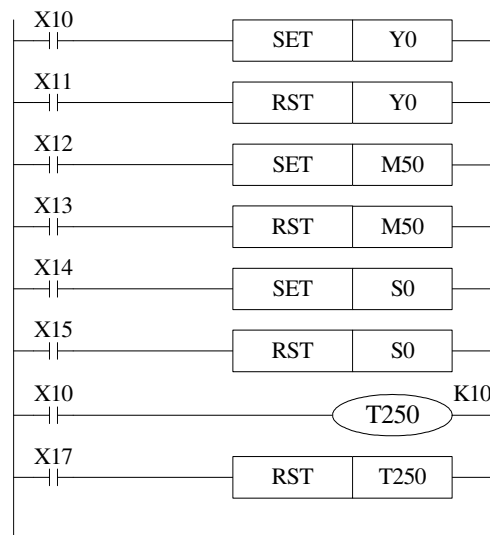
Mnemônico e Função

Mnemônico	Função	Formato e registros
SET (Seta)	Condiciona um “ <i>bit</i> ” permanentemente “1”	 Registros : Y, M, S, T, C, Dn.m
RST (Reset)	Condiciona um “ <i>bit</i> ” permanentemente “0”	 Registros : Y, M, S, T, C, Dn.m

Descrição

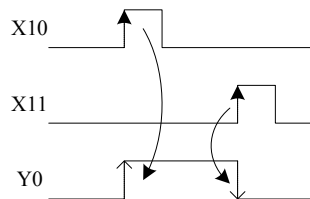
- A instrução SET força um registro para “1” garantindo o seu estado.
- A instrução RST força um registro para “0”, garantindo o seu estado.
- Use as instruções SET e RST, sempre em conjunto.

Programa



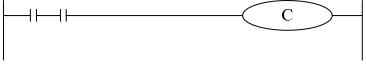

```

LD    X10
SET   Y0
LD    X11
RST   Y0
LD    X12
SET   M50
LD    X13
RST   M50
LD    X14
RST   S0
LD    X15
RST   S0
LD    X10
OUT   T250
SP    K10
LD    X17
RST   T250
    
```

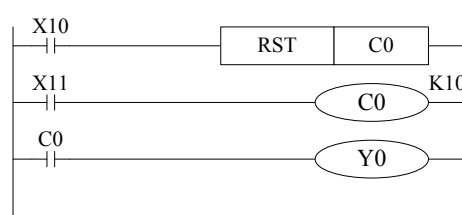


4-13 . 【OUT】 , 【RST】 para os contadores

Mnemônico e Funções

Mnemônico	Função	Formato e registros
OUT (OUT)	Resultado da lógica dos contatos. Finaliza a instrução.	 K or D
RST (Reset)	Condiciona um “bit” permanentemente “0”	

Programação

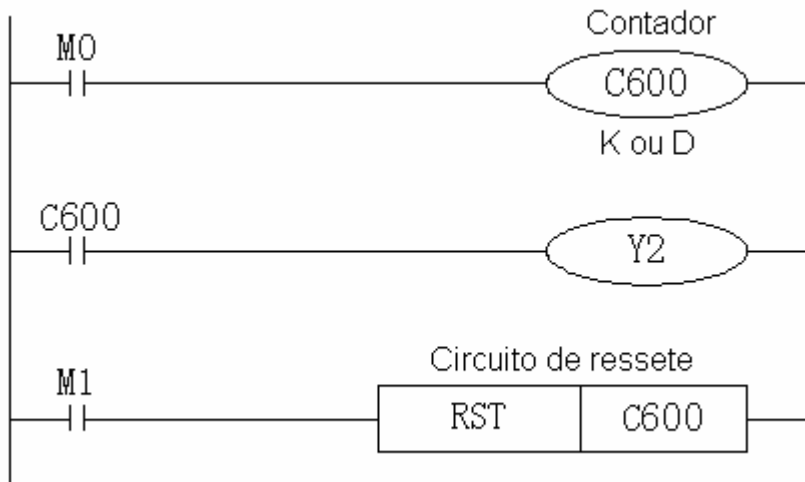


C0 é incrementado através da transição do pulso de X011 de “0” para “1”. Quando o valor da contagem K10 é alcançado, a saída do contador C0 é ativada. Caso a entrada X011 continue gerando pulsos o contador não fará mais contagens porque a sua saída já está ativada.

Para uso com os contadores retentivos. Sempre que for deserneizado o CLP, para um contador retentivo (abaixo) o seu valor permanecerá com o estado anterior, use o reset para limpar o “bit” do contador.

Para zerá-lo, foi utilizada a entrada X010 que utiliza o comando RST para reiniciar o valor do contador. Isso é necessário para reiniciar a contagem do contador e limpar a saída também.

Programando com o contador de alta velocidade



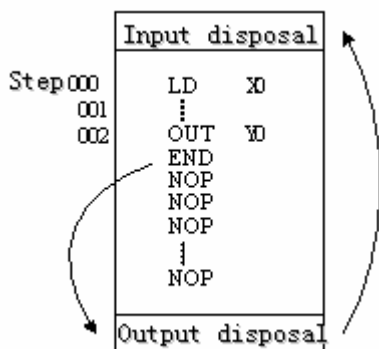
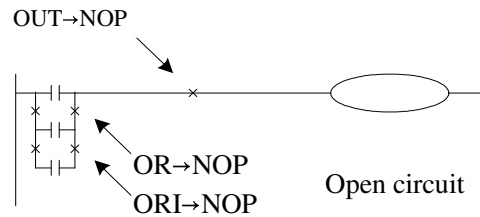
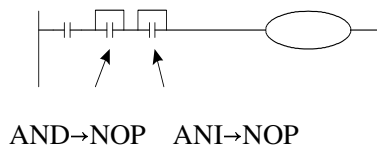
4-14 . 【NOP】 , 【END】

Mnemônico e Funções

Mnemônico	Função	Formato e registros
NOP (Sem operação)	Não executa durante o tempo da instrução	
END (FIM)	Força o programa para o final do "scan"	

Descrição

- A função NOP não efetua nenhuma operação. E



O CLP executa repetidamente a leitura das entradas, execução do programa e atualização das saídas (scan). O programa é executado a partir da primeira instrução até a instrução END. As funções após o END não são executadas.

4-15 . Pequenas notas para o início de programação

1- A estrutura de contatos e o número de passos

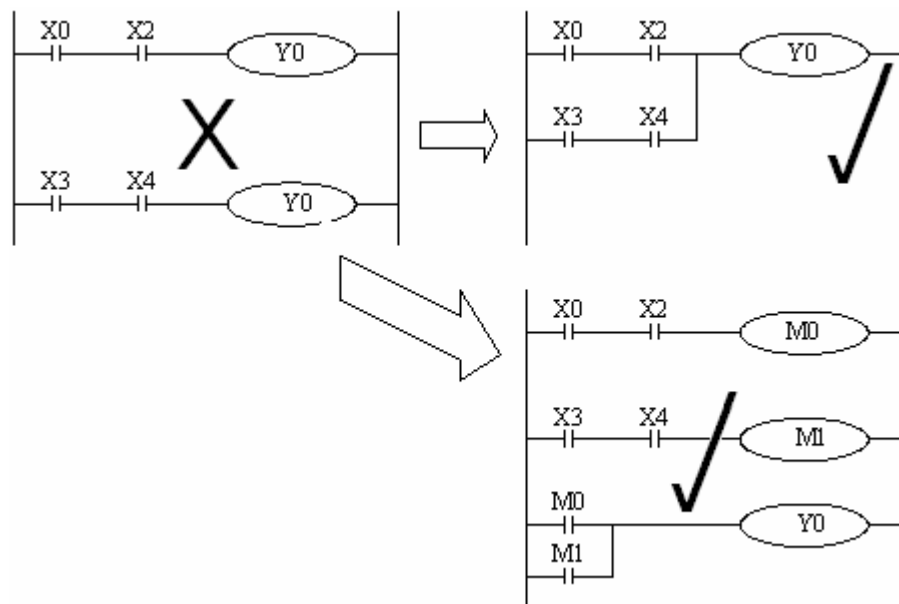
O programa é realizado em ordem seqüencial, de fácil construção e pode ser salvo de acordo com o avanço no desenvolvimento.

2- Seqüência da programação

A seqüência do programa é feito da seguinte maneira: **【de cima para baixo】** e **【da esquerda para a direita】** .

3- Duas saídas ativadas e uma solução

- Coloque as instruções em uma lógica combinacional “E” ou “OU”..., quando precisar ativar a mesma saída.
- Não é contra a regra usar duplamente a mesma saída, mas, este procedimento confunde e dificulta na interpretação do programa, por favor, modifique o programa conforme o desenho abaixo.



Existem diversas maneiras para ativar uma saída. O importante é garantir que a uma saída terá apenas uma linha para a sua execução, isto facilita muito no desenvolvimento do programa garantido toda a lógica em apenas uma linha.