

Twido PLC

指令对译速查手册

- Omron
- Mitsubishi
- Siemens



Schneider
 **Electric**
Building a New Electric World

前言

作为施耐德电气OEM市场的核心产品，Twido PLC正在各行各业被广泛应用。为有效减少在产品换型、客户开发的过程中，施耐德电气和代理商技术人员在指令翻译和特殊功能实现方面所花的精力，并帮助相关工程师提高客户开发的效率，小型自动化产品部门特将以往的指令对译经验收集归纳成册，以期对我们的现场工作有所帮助。

由于市面上主流PLC产品各有所长，我们选取了比较有代表性的三家产品：Siemens，Mitsubishi和Omron的相关指令，逐条对译。除极少数指令由于产品特性和特殊实现方式的原因无法翻译外，我们均尽可能列出了Twido指令的实现方法。

也借此机会，希望向各位展示Twido指令的优势所在：简单、易用、灵活。正如前文所说，各有所长。本手册的目的正是要在保留Twido长处的前提下，弥补不足之处。以求运筹帷幄之中，决胜千里之外。

由于时间因素和技术人员的经验原因，手册中难免有错漏之处，希望在实际使用过程中，各位能将发现的不足和自己的心得和我们共同分享；我们将在再版时加以补充和更正。

本手册是由小型自动化技术支持工程师荣东坡和石岩编写、全体部门同事协作完成的。如您在使用过程中发现任何问题或有任何意见和建议，请联系：

荣东坡

电话: 021-2401 2731

传真: 021-6485 3154

手机: 13611860153

邮件: dongpo.rong@cn.schneider-electric.com

张笑丰

小型自动化产品市场部经理

2007-10-18

目录

Omron

1、位操作

Out / out not.....	13
Set / reset	13
Keep	14
DIFU/DIFD.....	14

2、计时器、计数器

TIM, TIMH, TIMHH, TIML.....	15
CTBL注册高速计数比较表、INI计数模式控制	15
CNT, CNTR.....	16

3、移位指令

SFT	17
WSFT字移位.....	18
ASR算术右移	18
ROL循环左移	19
ROR循环右移	19

4、数据传送指令

MOV.....	20
MVN传送非.....	20

XFER	21
BSET	21
COLL	22
DIST	22
XCHG	23

5、比较指令

CMP	24
ZCP区域比较	24
CMPL双字比较	25
BCMP	25
TCMP表格比较	26

6、转换指令

BIN(BCD到二进制)	27
BCD(二进制到BCD)	27
BINL(双字BCD到双字二进制)	28
BCDL(双字二进制到双字BCD)	28
ASC (ASCII转换)	29
HEX (ASCII到16进制)	30
NEG二进制求补	31

7、二进制计算指令

ADB二进制加法	32
SBB二进制减法	32

MLB二进制乘法	33
DVB二进制除法	33

8、特殊数学指令

SRCH数据搜索	34
MAX寻找最大值	34
MIN寻找最小值	35
SUM求和	35

9、逻辑指令

COM求反	36
ANDW字与	36
OR字或	36
XOR字异或	37
XORN字异或非	37

Mitsubishi

1、传送和比较

CMP比较	39
BCD二进制-BCD转换	39
MOV传送	40
FMOV多点传送	40
CML反相传送	41
XCH交换	41

BMOV成批传送.....	42
ZCP区域比较.....	42
BIN(BCD-二进制转换)	43

2、四则逻辑运算

ADD(BIN加法运算)	44
MUL(BIN乘法运算)	44
SUB(BIN减法运算)	45
DIV(BIN除法运算).....	45
INC和DEC (BIN增减/BIN减少)	46
NEG 求补.....	46
WAND, WOR和WOR (字与, 字或, 字异或).....	47

3、旋转、移位

ROR右回转.....	48
ROL左回转	48
SFTR位右移	49
SFTL位左移.....	49
SFRD移位读出	50
WSFL字左移.....	50
SFWR移位写入.....	51
WSFR字右移	51

4、数据处理

ZRST全部复位.....	52
---------------	----

	BON(ON位判断)	52
	MEAN(平均值)	53
	SQR(BIN开放运算)	53
	FLT(BIN整数→浮点转换)	54
	SWAP(高低位字节对换)	54
5、方便命令		
	ALT(交替输出).....	55
	TTMR(示教定时器)	56
6、外部设备-I/O		
	ASC(ASCII转换)	57
7、外部机器-SER		
	PRUN(八进制位传送).....	58
	VRRD(电位器值读出)	58
	ASCI(HEX→ASCII转换)	59
	HEX(ASCII→HEX转换)	60
8、外部设备		
	GRY(将BIN数据转换为格雷码并传送)	61
	GBIN(将格雷码数据转换为BIN并传送)	62
9、浮点小数运算		
	DECMP(浮点比较).....	63

DEZCP(浮点区域比较)	63
EADD(浮点加法)	64
ESUB(浮点减法)	64
EMUL(浮点乘法)	65
EDIV(浮点除法)	65
DSIN(浮点SIN 运算)	66
ESQR(浮点数开平方)	66
DINT(浮点数→BIN整数)	67
DCOS(浮点COS运算)	67
DTAN(浮点TAN运算)	68

10、计数器和计时器

计时器(非保持型)	69
计数器(双向)	70
计数器(普通)	71
高速计数器	72
FX, FX2C中可用的高速计数器	73

Siemens

1、位逻辑指令

取反指令	75
负跳变触点指令	75
正跳变触点指令	76
复位指令	76

置位指令	77
SR置位触发器指令	78
RS复位触发器指令	79

2、时钟指令

读时钟指令	80
写时钟指令	81

3、比较指令

字节比较指令—等于	82
字节比较指令—不等于	82
字节比较指令—大于等于	83
字节比较指令—小于等于	83
字节比较指令—大于	84
字节比较指令—小于	84
单字比较—等于	85
单字比较—不等于	85
单字比较—大于等于	86
单字比较—小于等于	86
单字比较—大于	87
单字比较—小于	87
双字比较—等于	88
双字比较—不等于	88
双字比较—大于等于	89
双字比较—小于等于	89

双字比较—大于	90
双字比较—小于	90
实数比较—等于	91
实数比较—不等于	91
实数比较—大于等于	92
实数比较—小于等于	92
实数比较—大于	93
实数比较—小于	93

4、转换指令

单字转换为双字	94
双字转换为单字	94
双字转换为实数	95
BCD与整数的转化	95
四舍五入指令	96
取整指令	96
ASCII转16进制整数	97
整数转ASCII	98

5、计数器指令

增计数	100
减计数	100
增减计数器	101
高速计数器指令	102

6、浮点数计算

浮点数加法.....	103
浮点数减法.....	103
浮点数乘法.....	104
浮点数除法.....	104
求平方根	105
求正弦.....	105
求余弦.....	106
求正切.....	106
求对数.....	107
求指数.....	107

7、整数运算

整数加法	108
双字整数加法	108
整数减法	109
双整数减法.....	109
乘法	110
整数乘法	110
双整数乘法.....	111
除法	111
整数除法	112
双整数除法.....	112
字加1指令	113
双字加1指令.....	113

字减1指令	114
双字减1指令	114

8、逻辑操作

字取反	115
双字取反	115
字求与指令	116
双字求与指令	116
字求或指令	117
双字求或指令	117
字求异或	118
双字求异或	118

9、MOV指令

MOV字	119
MOV双字	119
MOV实数	120
BLKMOV字	120
BLKMOV 双字	121
WAP字节交换指令	121

10、PROGRAM CONTROL

FOR/NEXT循环	122
STOP指令	123

11、SHIFT/ROTATE

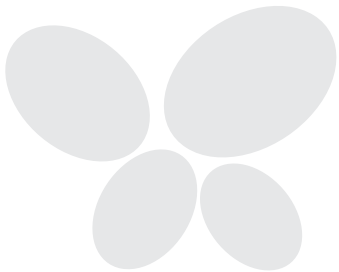
SHL_W字左移	124
SHL_DW双字左移	124
SHR_W字右移	125
SHR_DW双字右移	125
ROL_W字循环左移	126
ROL_DW双字循环左移	126
ROR_W字循环右移	127
ROR_DW双字循环右移	127

12、定时器

TON接通延时	128
TOF断开延时	128

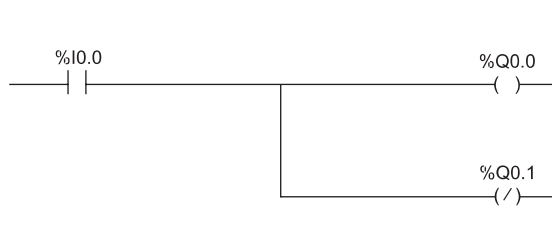
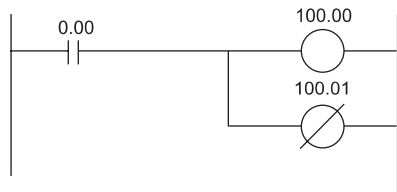
The Omron logo consists of a light gray, horizontally-oriented oval shape. Inside this oval, the word "Omron" is written in a black, sans-serif font, centered horizontally.

Omron

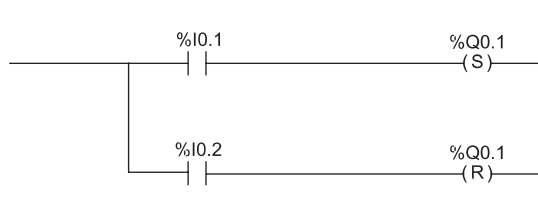
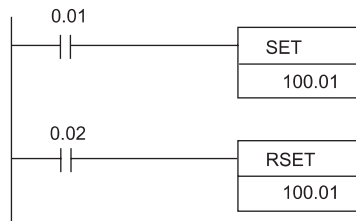


1、位操作

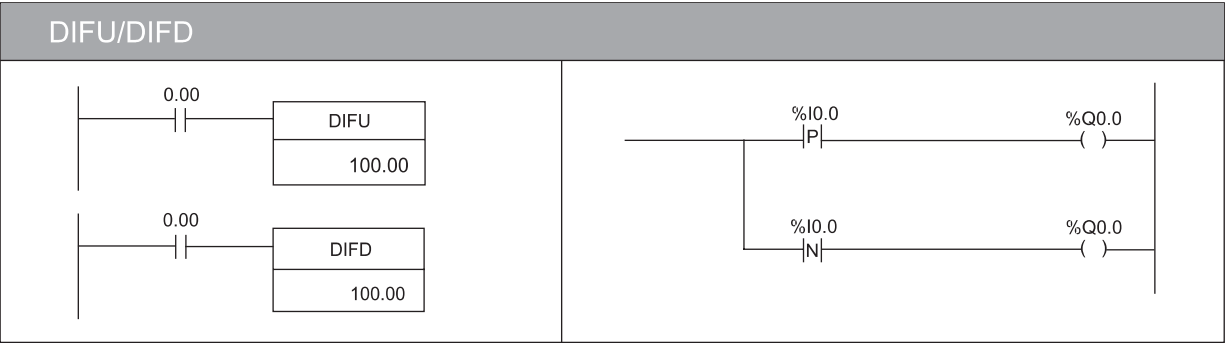
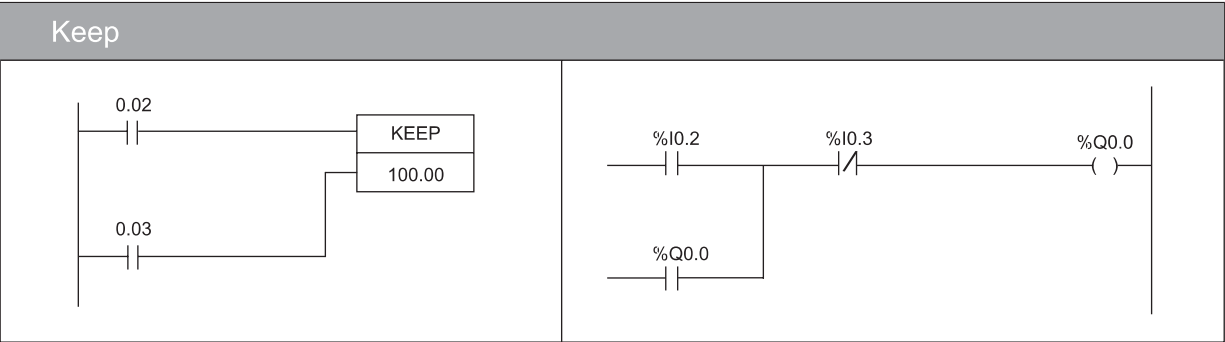
Out / out not



Set / reset

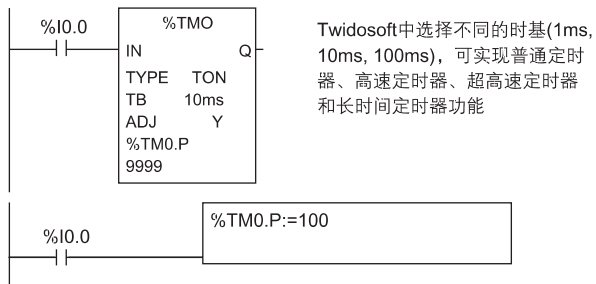
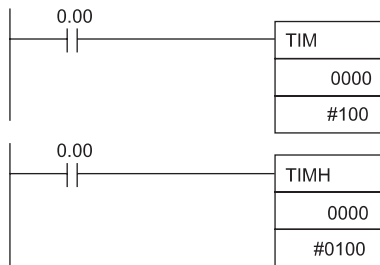


1、位操作

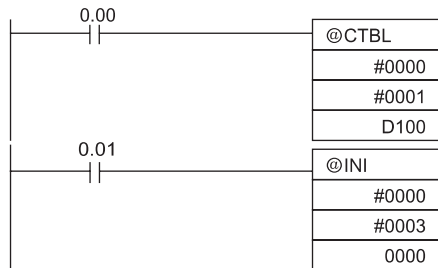


2、计时器、计数器

TIM, TIMH, TIMHH, TIML



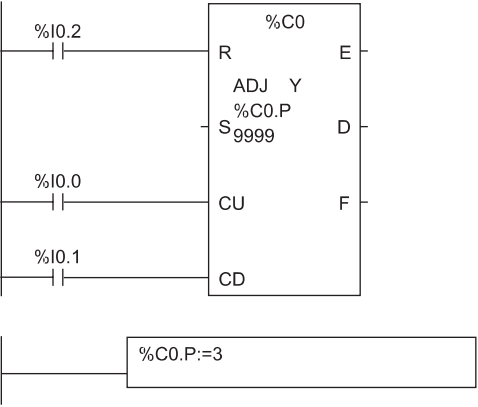
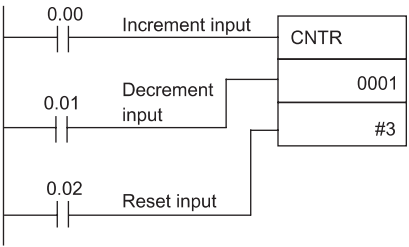
CTBL注册高速计数比较表、INI计数模式控制



在VFC功能块中配置阈值和计数器类型实现。

2、计时器、计数器

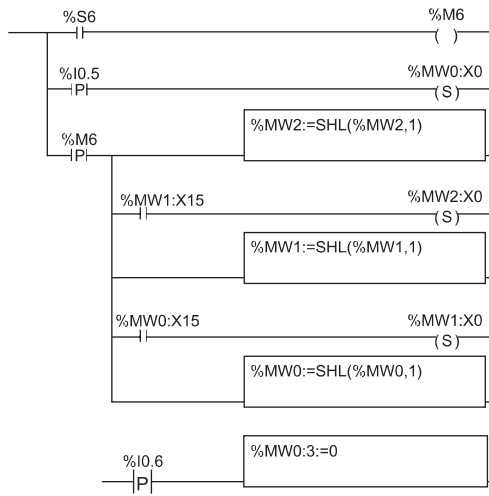
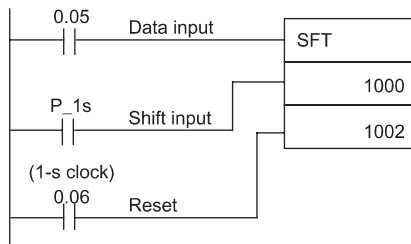
CNT, CNTR



Twidosoft中所有计数器均为双向计数器

3、移位指令

SFT



本例中以秒脉冲为移位条件，%I0.6为复位条件。输入%I0.5的状态被每秒移位。

3、移位指令

WSFT字移位

St: 起始字 E: 结束字
当执行条件置OFF时, WSFT(16)不被执行。
当执行条件置ON时, WSFT(16)
在以字为单位的St和E之间移位数据。0被写入St且E中内容将丢失。

E				St+1				St			
F	0	C	2	3	4	5	2	1	0	2	9

丢失 ←

E				St+1				St			
3	4	5	2	1	0	2	9	0	0	0	0

0000

```
%I0.0 |P| [%MW101:2:=%MW100:2] [%MW100:=0]
```

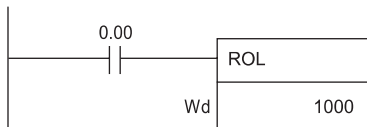
ASR算术右移

Wd 1000

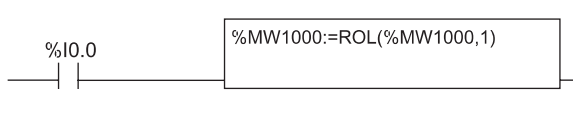
```
%I0.0 |P| [%MW1000:=SHR(%MW1000,1)]
```

3、移位指令

ROL循环左移

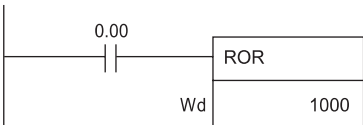


注意：在执行条件0N时，ROL将操作字所有位向左移一位，把CY移入操作字第0位，并把操作字的第15位移入CY

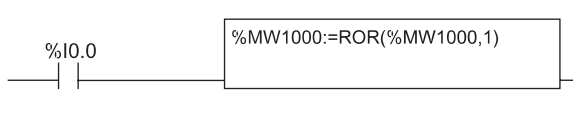


注意：在使用移位指令前请注意Omron PLC的CY状态，指令转换时做适当处理。

ROR循环右移



注意：在执行条件0N时，ROR将操作字所有位向右移一位，把CY移入操作字第15位，并把操作字的第0位移入CY



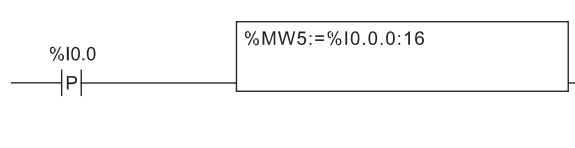
注意：在使用移位指令前请注意Omron PLC的CY状态，指令转换时做适当处理。

4、数据传送指令

MOV

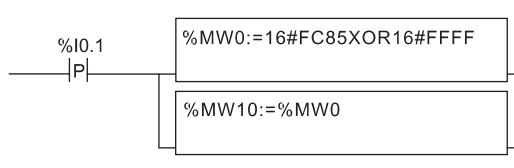


当IR 00000从OFF到ON变化时，@MOV(21)被用来把IR0001中的内容复制到HR 05。

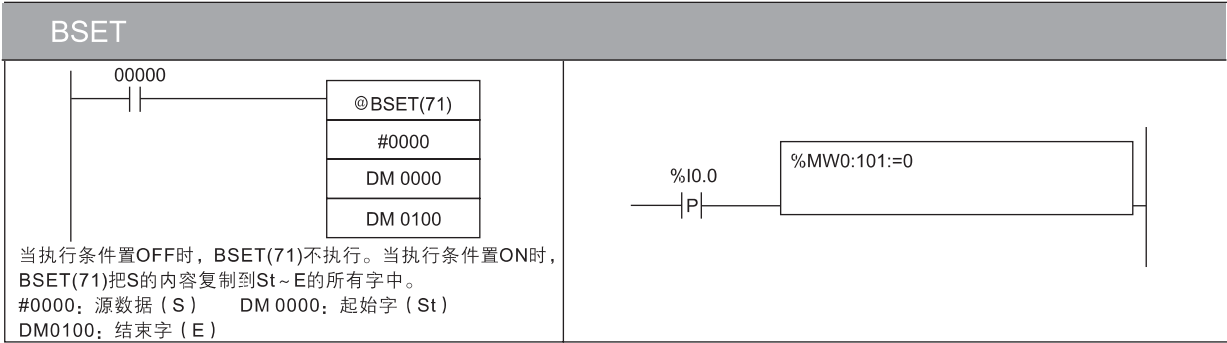
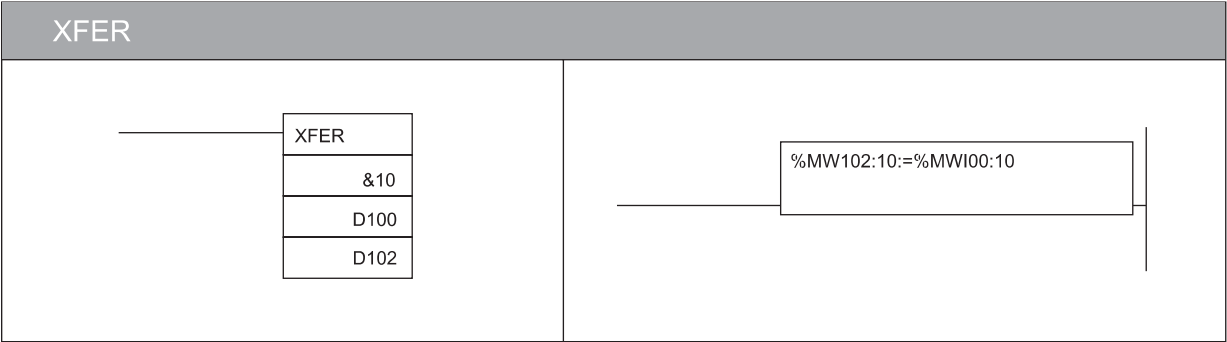


请自行注意使用对应的内存字。

MVN传送非

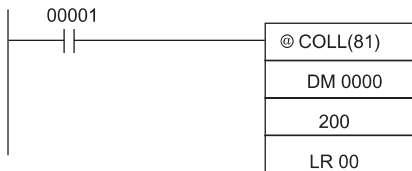


4、数据传送指令

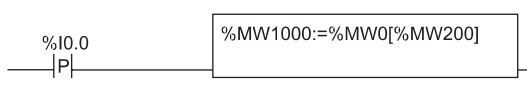


4、数据传送指令

COLL

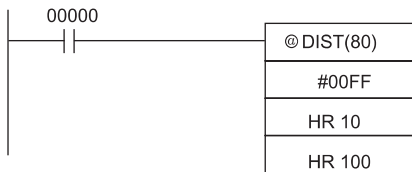


注意：IR200控制字，当其第12~15位为0~7数值时，用于单字分配操作，指令可如下翻译。当其内容为9或时，用于堆栈操作。具体请参考OMRON手册。

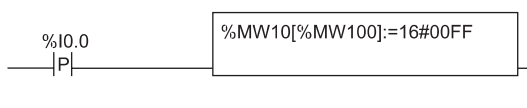


注意：此处%mw0对应 DM0000，%MW200对应IR200，%MW1000对应LR00

DIST

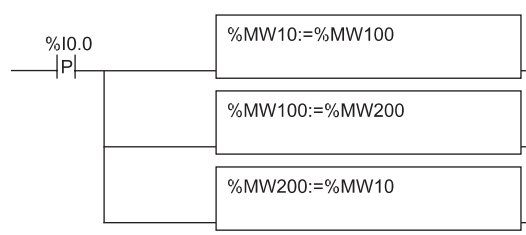
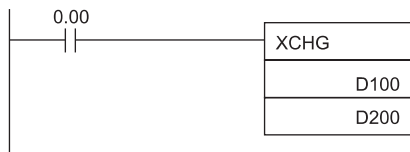


注意：HR100控制字，当其第12~15位为0~8数值时，用于单字分配操作，指令可如下翻译。当其内容为9时，用于堆栈操作。该字内容为堆栈的指针。具体请参考OMRON手册。



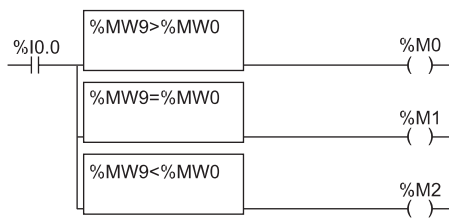
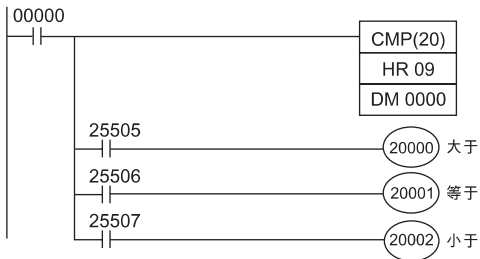
4、数据传送指令

XCHG



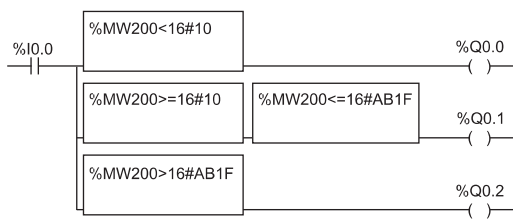
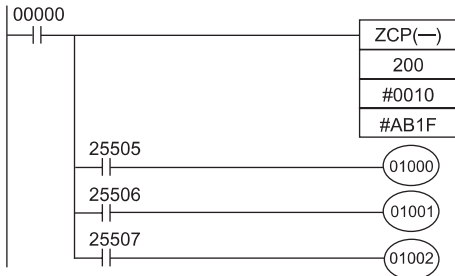
5、比较指令

CMP



注意：此处%MW9对应HR9，%MW0对应DM0000

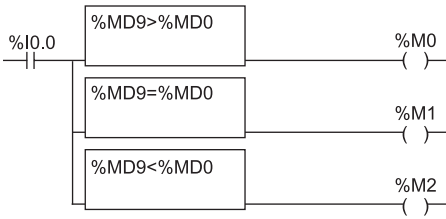
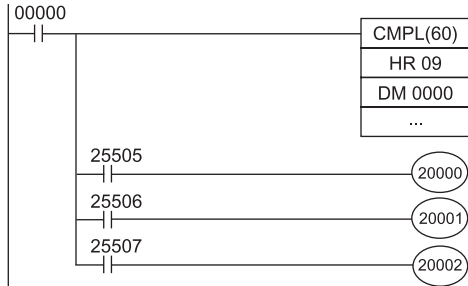
ZCP 区域比较



注意：此处%MW200对应IR200。双字区域比较指令ZCPL与ZCP类似，将本例中的操作数改为双字%MD即可，请自行使用。

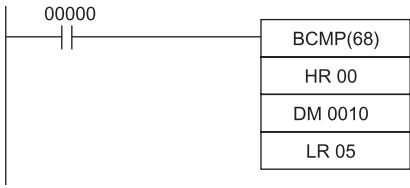
5、比较指令

CMPL双字比较



注意：此处%MD9对应HR09+HR10；%MD0对应DM0000+DM0001

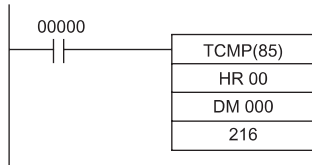
BCMP



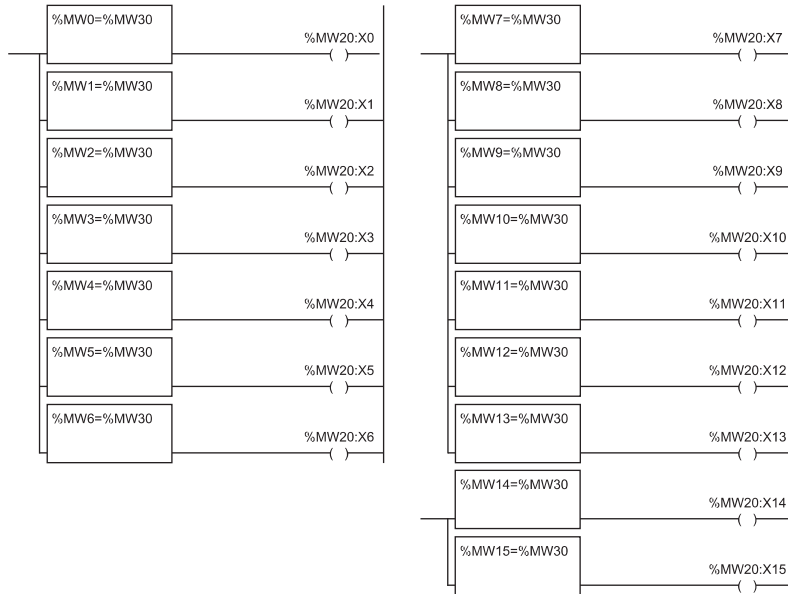
与TCMP指令类似，请参考TCMP翻译程序段。

5、比较指令

TCMP表格比较



当执行条件为ON时，TCMP(85)将CD和TB，TB+1，TB+2，...，和TB+15的内容进行比较。如果CD和任何这些字的内容相等时，其R中对应位被置位



注意：此处%MW0对应DM0000，%MW30对应HR00，%MW20对应IR216

6、转换指令

BIN(BCD到二进制)

—

BIN(23)

S

R

%MW10:=BTI(%MW20)

—

本例中%MW20为S，%MW10为R

BCD(二进制到BCD)

—

BCD(24)

S

R

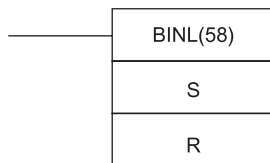
%MW10:=ITB(%MW20)

—

本例中%MW20为S，%MW10为R

6、转换指令

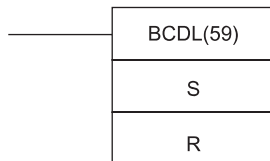
BINL(双字BCD到双字二进制)



%MD10:=BTI(%MD20)

本例中%MD20为S, %MD10为R

BCDL(双字二进制到双字BCD)

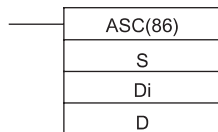


%MD10:=ITB(%MD20)

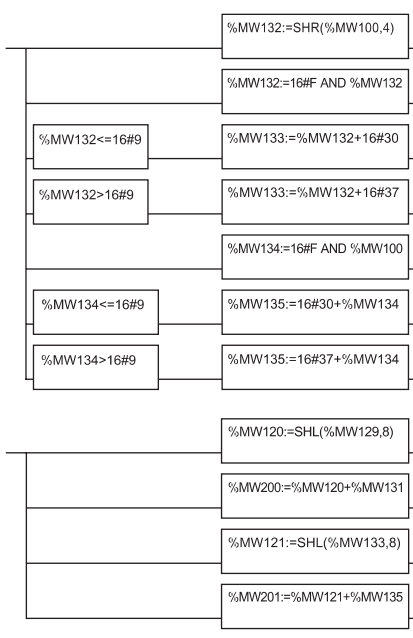
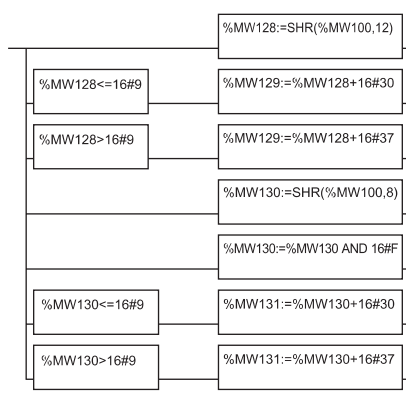
本例中%MD20为S, %MD10为R

6、转换指令

ASC (ASCII转换)



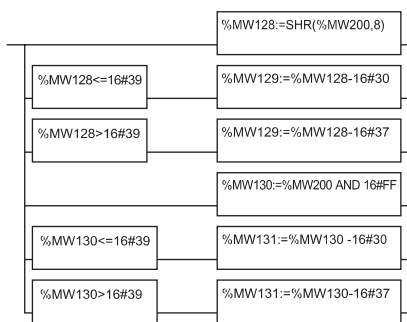
当执行条件为ON时，ASC(86)把S指定的数字转换成对应的8位ASCII码并将其放入从D起始的目的字中



注意：此处%MW100对应S，%MW200对应D，默认控制字为16#0030时的情形。其余情况请根据OMRON手册自行对结果进行处理。

30

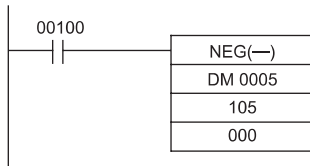
—	HEX(—)
	S
	Di
	D



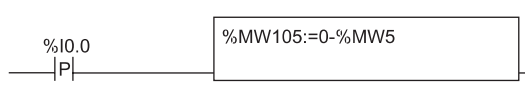
	%MW132:=SHR(%MW201,8)
%MW132<=16#39	%MW133:=%MW132-16#30
%MW132>16#39	%MW133:=%MW132-16#37
	%MW134:=16#FF AND %MW201
%MW134<=16#39	%MW135:=%MW134 -16#30
%MW130>16#39	%MW135:=%MW134-16#37
	%MW120:=SHL(%MW129,12)
	%MW121:=SHL(%MW131,8)
	%MW122:=SHL(%MW133,4)
	%MW100:=%MW120+%MW121
	%MW100:=MW100+%MW122
	%MW100:=%MW100+%MW135

6、转换指令

NEG二进制求补



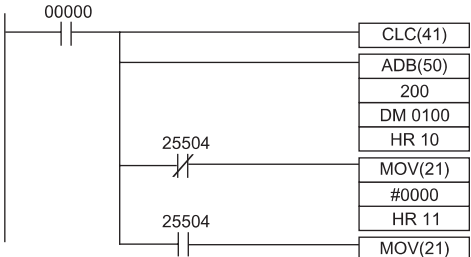
把源字(S)的四位十六进制内容转换为它的二进制补码并且把结果输出到结果字(R)中。这项操作和从0000减去S并把结果输出到R有一样效果；它计算带符号二进制负数的绝对值。



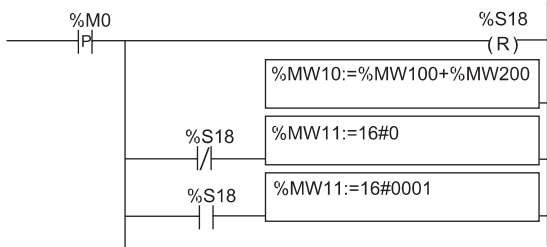
注意：本例中%MW5对应DM0005，%MW105对应IR105

7、二进制计算指令

ADB二进制加法

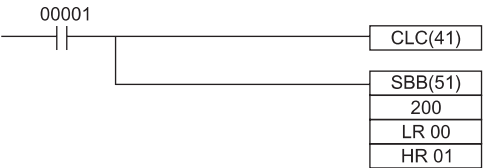


当执行条件为ON时，ADB(50)把Au，Ad和CY的内容相加并把结果输出到R。如果结果大于FFFF，CY将被置位。IR200：被加字(Au) Dm0100：加字(Ad) HR10：结果字(R)
注意：HR10用于保存计算结果，HR11用于保存进位标志

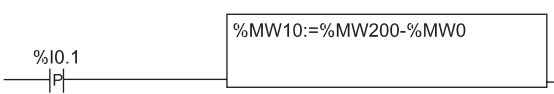


注意：本例中%M0对应HR10，%MW100对应DM0100，%MW200对应IR200

SBB二进制减法



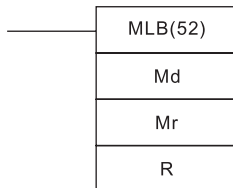
注意：本例中%M0对应HR10，%MW100对应DM0100，%MW200对应IR200



注意：本例中%M0对应HR10，%MW100对应DM0100，%MW200对应IR200

7、二进制计算指令

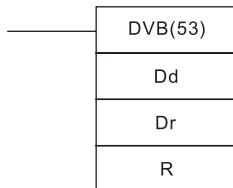
MLB



```
%MW9:=%MW0*%MW100
```

注意：本例中%MW0对应Md，%MW100对应Mr，%MW9对应R！对于OMRON PLC，乘法运算的结果低四位放入R，高四位放入R+1中；对于Twido，只采用一个单字保存运算结果，超出该字范围后，结果值无意义。

DVB



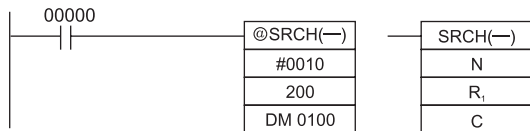
```
%MW10:=%MW20/%MW30
```

```
%MW11:=%MW20 REM %MW30
```

注意：本例中%MW20对应Dd，%MW30对应Dr，%MW10对应R！对于OMRON PLC，除法运算的商放入R，余数放入R+1中；对于Twido，需采用两条指令实现。本例中商放在%MW10中，余数放在%MW11中。

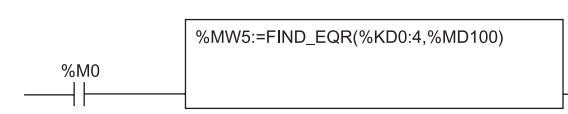
8、特殊数学指令

SRCH数据搜索



注意：

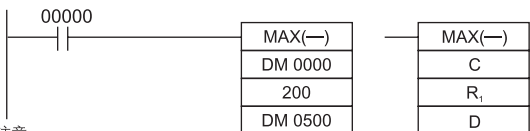
1. #0010为字表长度，IR200为字表首字，DM0100为要搜索的数据，DM0101内容为含有搜索字的最小地址。
2. 对于DM区域中的一个地址，地址字被写入C+1中。例如，含有比较数据的最小地址是DM0114，则#0114被写入C+1中。
3. 对于在另一数据区域中的一个地址，从搜索开始计的地址数被写入C+1。例如，含有比较数据的最小地址是IR114，搜索范围内第一个搜索到的字是IR014，则#0100被写入C+1。



注意：

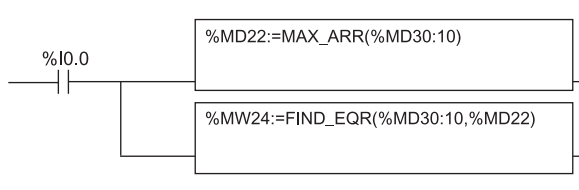
对于Twido PLC，只能对双字或浮点数查表。本例中%KD0对应IR200，%MD100对应DM0100，字表长度为4，查表结果存放在%MW5中。需要注意的是，查表结果返回的不是具体的地址，而是该数据在字表中的位置。例如，假如%MD100的值与%KD4相等，则%MW5的值为3。

MAX寻找最大值



注意：

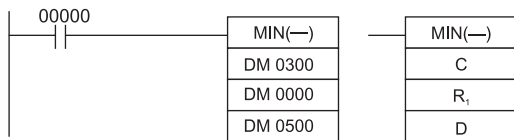
1. DM0000为字表长度，IR200为字表首字，DM0500为要最大数值，DM0501内容为含有搜索字的最小地址。
2. 对于DM区域中的一个地址，地址字被写入D+1中。例如，含有最大值的内存地址是DM0114，则#0114被写入D+1中。
3. 对于在另一数据区域中的一个地址，从搜索开始计的地址数被写入D+1。例如，如果含有最大值的内存地址是IR114，且搜索范围的第一个字是IR014，则#0100写入D+1。



注意：对于Twido PLC，只能对双字或浮点数操作。本例中%MD30对应IR200，%MD22对应DM0500，存放查表结果。字表长度为10，对应DM0000。%MW24中存放的是该数据在字表中的位置。同SRCH指令的返回结果。

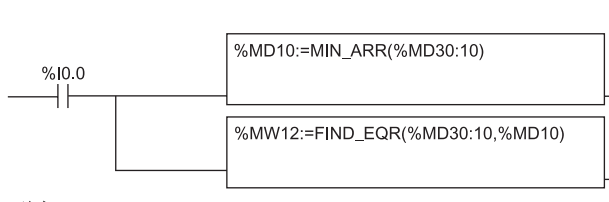
8、特殊数学指令

MIN寻找最小值



注意：

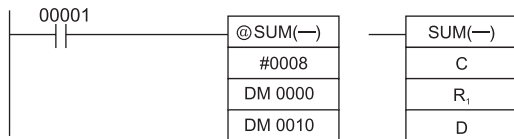
1. DM0300为字表长度，DM0200为字表首字，DM0500为要最大数值，DM0501内容为含有搜索字的最小地址。
2. 对于DM区域中的一个地址，地址字被写入C+1中。例如，包含最小值的内存地址是DM0114，则#0114被写入D+1中。
3. 对于在另一类数据区域中的一个地址，从搜索开始计的地址数被写入D+1。例如，如果含有最小值的内存地址是IR114，且搜索范围的第一个字是IR014，则#0100写入D+1。



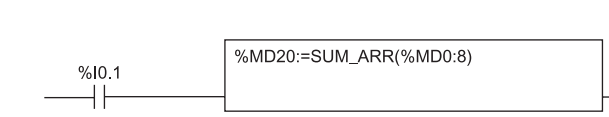
注意：

对于Twido PLC，只能对双字或浮点数操作。本例中%MD30对应DM0000，%MD10对应DM0500，存放查表结果。字表长度为10，对应DM0300的内容。%MW12中存放的是该数据在字表中的位置。同SRCH指令的返回结果。

SUM求和

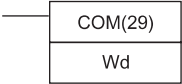
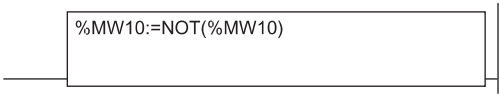
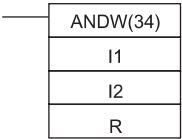
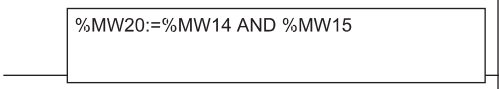
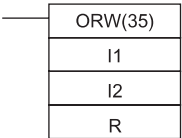
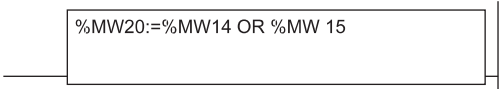


C为字表长度，R1为字表首字，D为目的首字。



注意：对于Twido PLC，只能对双字或浮点数操作。

9、逻辑指令

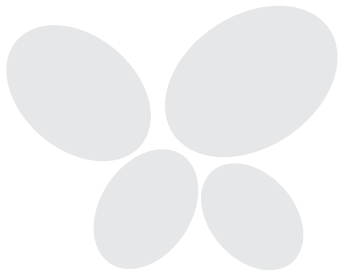
COM求反	
	
ANDW 字与	
	 %MW14对应I1, %MW15对应I2, %MW20对应R
OR 字或	
	 %MW14对应I1, %MW15对应I2, %MW20对应R

9、逻辑指令

XOR字异或							
<table><tr><td>XORW(36)</td></tr><tr><td>I1</td></tr><tr><td>I2</td></tr><tr><td>R</td></tr></table>	XORW(36)	I1	I2	R	<table><tr><td>%MW20:=%MW14 XOR %MW15</td></tr></table> %MW14对应I1， %MW15对应I2， %MW20对应R	%MW20:=%MW14 XOR %MW15	
XORW(36)							
I1							
I2							
R							
%MW20:=%MW14 XOR %MW15							
XORN字异或非							
<table><tr><td>XNRW(37)</td></tr><tr><td>I1</td></tr><tr><td>I2</td></tr><tr><td>R</td></tr></table>	XNRW(37)	I1	I2	R	<table><tr><td>%MW20:=%MW14 XOR %MW15</td></tr><tr><td>%MW20:=NOT(%MW20)</td></tr></table> %MW14对应I1， %MW15对应I2， %MW20对应R	%MW20:=%MW14 XOR %MW15	%MW20:=NOT(%MW20)
XNRW(37)							
I1							
I2							
R							
%MW20:=%MW14 XOR %MW15							
%MW20:=NOT(%MW20)							

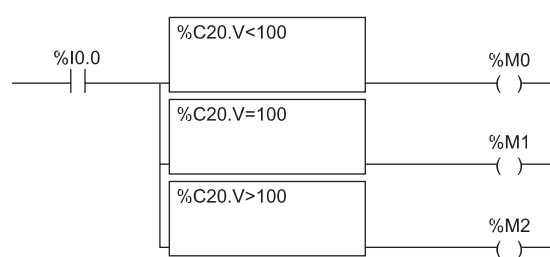
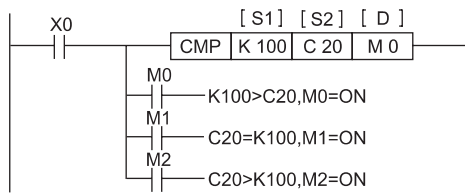


Mitsubishi

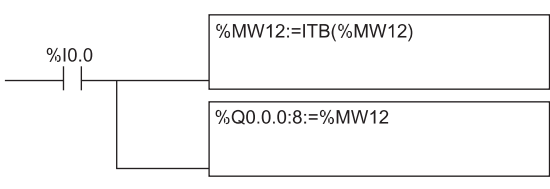


1、传送和比较

CMP比较

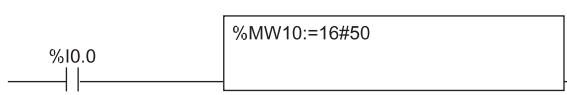


BCD二进制-BCD转换

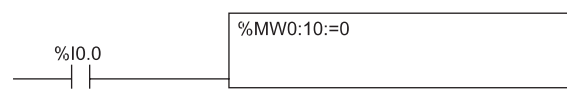


1、传送和比较

MOV传送



FMOV多点传送



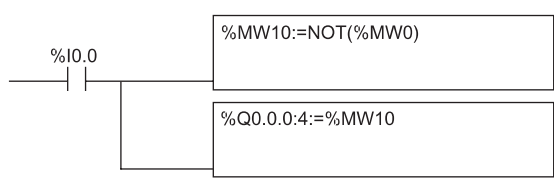
源数据内容被复制到目标数据区的每一个寄存器内

1、传送和比较

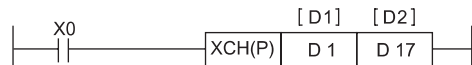
CML反相传送



将S内容取反后传送到D中

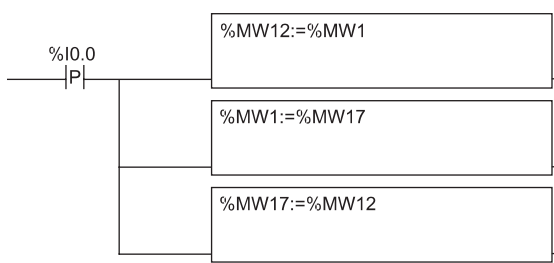


XCH交换



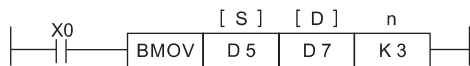
注意：

m8160为ON时，且D1和D2为同一内部字时，高8位和低8位将进行交换。

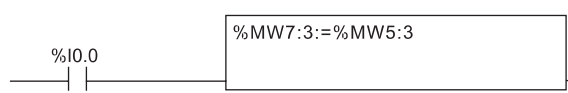


1、传送和比较

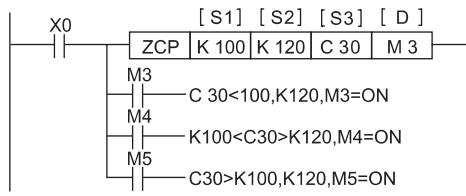
BMOV成批传送



复制连续数据区到指定区域



ZCP区域比较

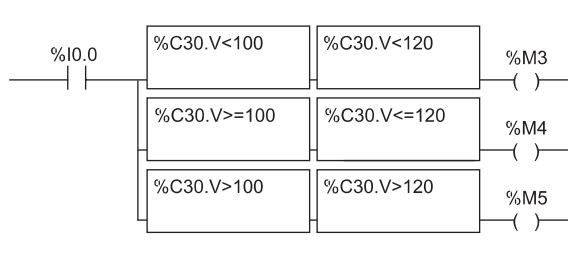


S1的内容不得大于S2的内容

S3 小于 S1a和 S2-位元件D 为 ON

S3 等于 或介于 S1 和 S2之间 -位元件D+1为 ON

S3 大于 S1 和 S2 -位元件D+2 为ON



1、传送和比较

BIN(BCD-二进制转换)



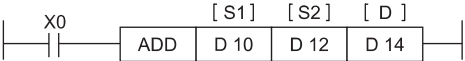
%I0.10

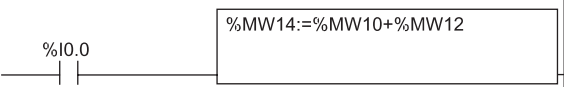
%MW13:=%I0.0:8

%MW13:=BTI(%MW13)

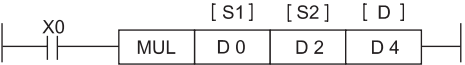
2、四则逻辑运算

ADD(BIN加法运算)

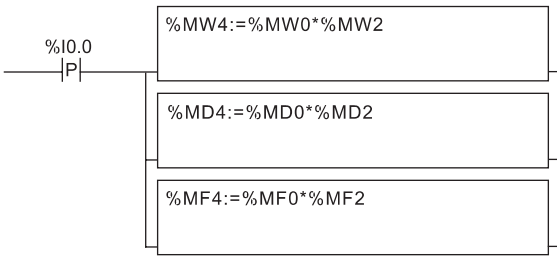




MUL(BIN乘法运算)



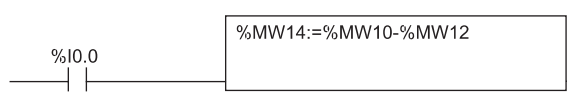
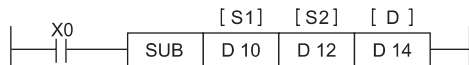
注意：运算结果保存在D4，D5中



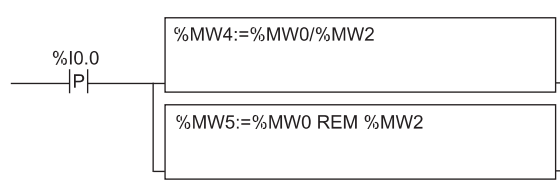
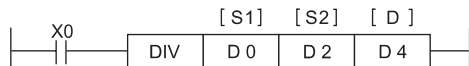
注意：在本例中，为防止溢出，可使用双字或浮点字进行运算

2、四则逻辑运算

SUB(BIN减法运算)



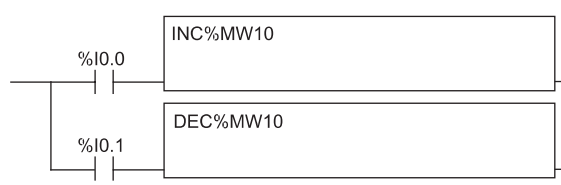
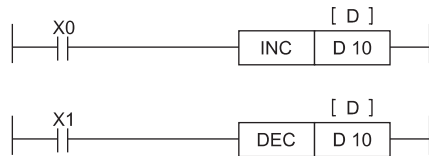
DIV(BIN除法运算)



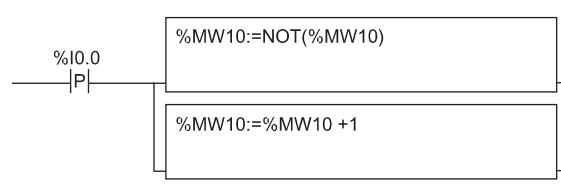
注意：商和余数分别保存在D4，D5中

2、四则逻辑运算

INC和DEC(BIN增减/BIN减少)

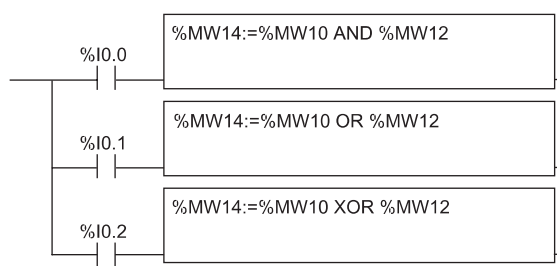
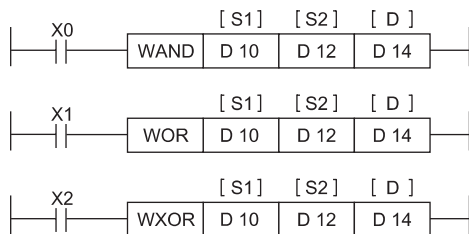


NEG 求补



2、四则逻辑运算

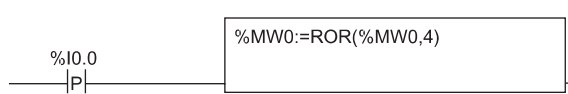
WAND, WOR和WOR(字与, 字或, 字异或)



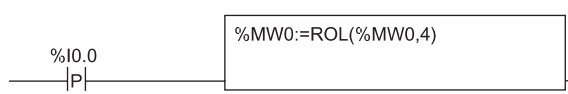
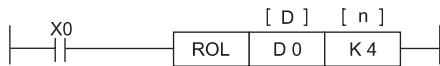
注意：XOR配合CML指令，可实现XORN即‘异或非’功能

3、旋转、移位

ROR右回转



ROL左回转

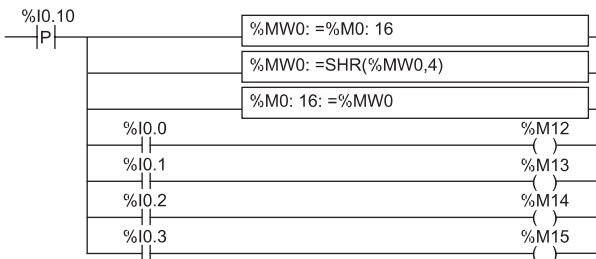


3、旋转、移位

SFTR位右移



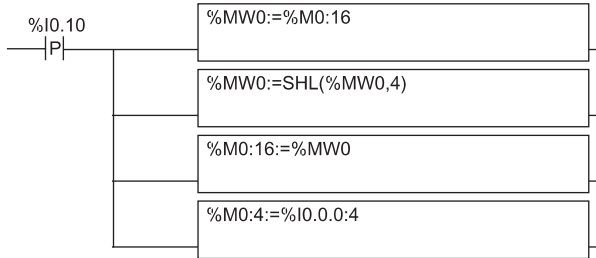
对于n1位长度的位元件进行n2位的右移动的指令



SFTL位左移

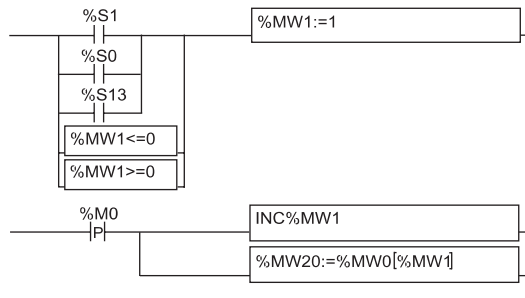


对于n1位长度的位元件进行n2位的左移动的指令，对应图示请参考SFTR翻译。

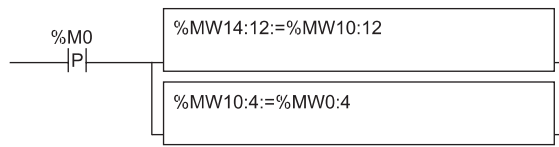


3、旋转、移位

SFRD移位读出



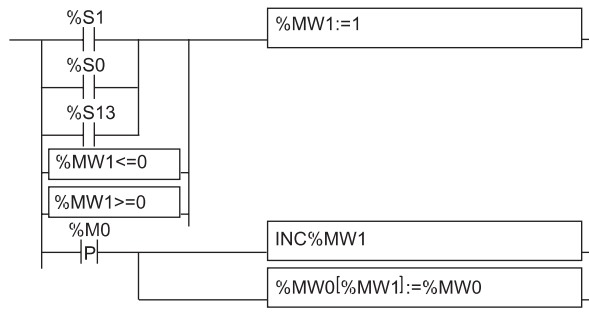
WSFL字左移



注意：三菱PLC指令执行过程请参考该手册；此处翻译只针对该指令特例，非通用翻译。

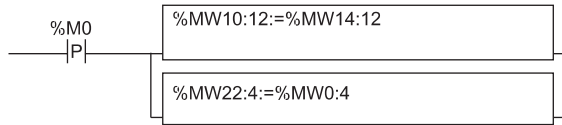
3、旋转、移位

SFWR 移位写入



注意：三菱PLC指令执行过程请参考该手册；此处翻译只针对该指令特例，非通用翻译。

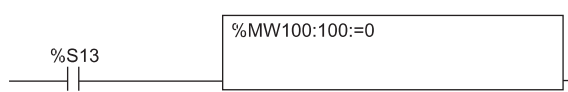
WSFR 字右移



注意：三菱PLC指令执行过程请参考该手册；此处翻译只针对该指令特例，非通用翻译。

4、数据处理

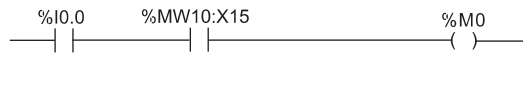
ZRST全部复位



BON(ON位判断)



D10的第15位为ON时，M0为ON

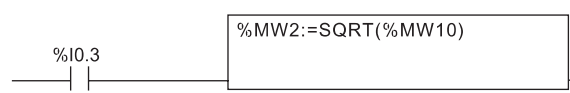
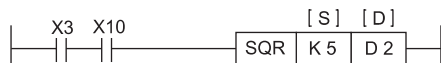


4、数据处理

MEAN(平均值)



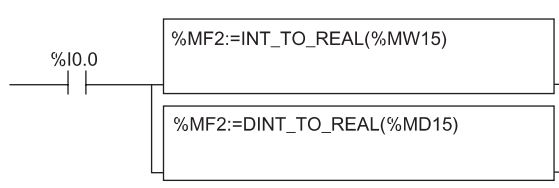
SQR(BIN开放运算)



Twidosoft中，源操作数不能为立即值

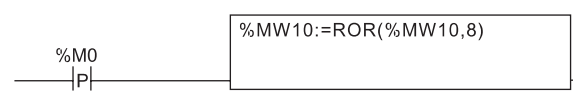
4、数据处理

FLT(BIN整数→浮点转换)



本例中包含单字和双字的转换示例。

SWAP(高低位字节对换)

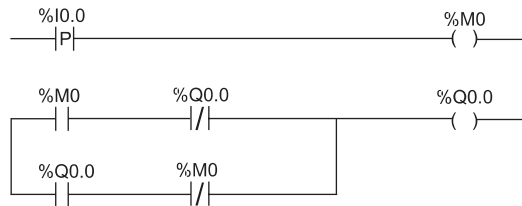


5、方便命令

ALT(交替输出)



X0第一次为ON, M0为ON
X0第二次为ON, M0为OFF
X0第三次为ON, M0为ON

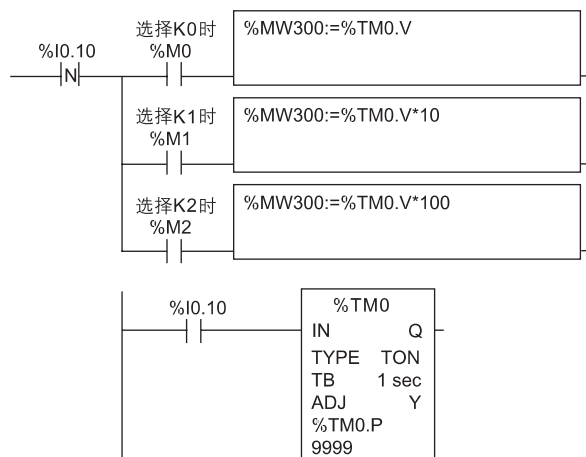


5、方便命令

TTMR(示教定时器)

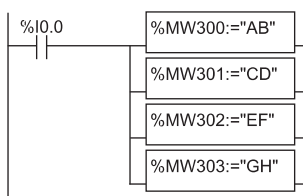
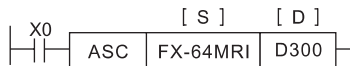


D301测定按钮X010的按动时间并乘以由n指定的倍率存入D300中。由此通过按钮可以调整定时器的设定时间。

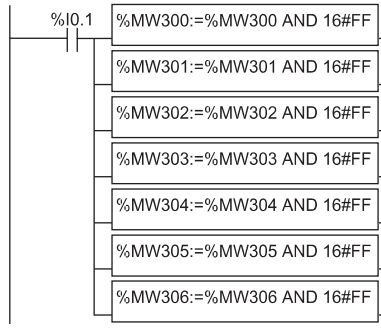
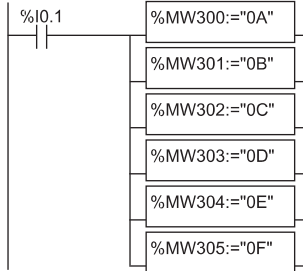


6、外部设备 · I/O

ASC(ASCII转换)

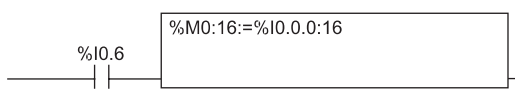


当三菱内部位M8161置ON时，程序应转换为：



7、外部机器・SER

PRUN(八进制位传送)

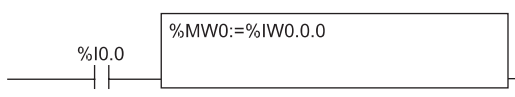


请注意根据内部位地址分配的不同，相关位的状况。

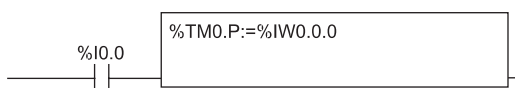
VRRD(电位器值读出)



将电位器0的模拟值读到D0中

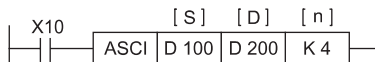


或者在用于模拟定时器的時候：



7、外部机器・SER

ASCII(HEX→ASCII转换)



将%MW100高字节转为ASCII码

	%MW128:=SHR(%MW100,12)
%MW128<=16#9	%MW129:=%MW128+16#30
%MW128>16#9	%MW129:=%MW128+16#37
	%MW130:=SHR(%MW100,8)
	%MW130:=%MW130AND16#F
%MW130<=16#9	%MW131:=%MW130+16#30
%MW130>16#9	%MW131:=%MW130+16#37

将%MW100低字节转为ASCII

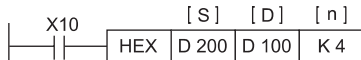
	%MW132:=SHR(%MW100,4)
	%MW132:=16#FAND%MW132
%MW132<=16#9	%MW133:=%MW132+16#30
%MW132>16#9	%MW133:=%MW132+16#37
	%MW134:=16#FAND%MW100
%MW134<=16#9	%MW135:=16#30+%MW134
%MW134>16#9	%MW135:=16#37+%MW134

转换后的数据放在%MW200和%MW201中

	%MW120:=SHL(%MW129,8)
	%MW200:=%MW120+%MW131
	%MW121:=SHL(%MW133,8)
	%MW201:=%MW121+%MW135

7、外部机器・SER

HEX(ASCII→HEX转换)



将%MW200转为ASCII码

	%MW128:=SHR(%MW200,8)
%MW128<=16#39	%MW129:=%MW128-16#30
%MW128>16#39	%MW129:=%MW128-16#37
	%MW130:=%MW200 AND 16#FF
%MW130<=16#39	%MW131:=%MW130-16#30
%MW130>16#39	%MW131:=%MW130-16#37

%MW201转为ASCII

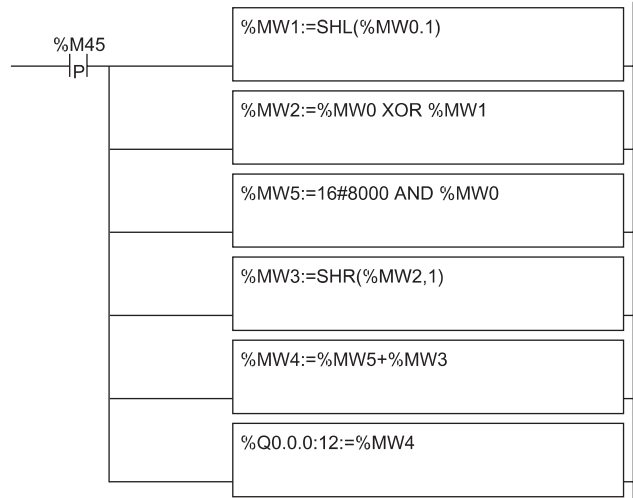
	%MW132:=SHR(%MW201,8)
%MW132<=16#39	%MW133:=%MW132-16#30
%MW132>16#39	%MW133:=%MW132-16#37
	%MW134:=16#FF AND %MW201
%MW134<=16#39	%MW135:=%MW134-16#30
%MW134>16#39	%MW135:=%MW134-16#37

结果存放在%MW100中

	%MW120:=SHL(%MW129,12)
	%MW121:=SHL(%MW131,8)
	%MW122:=SHL(%MW133,4)
	%MW100:=%MW120+%MW121
	%MW100:=%MW100+%MW122
	%MW100:=%MW100+%MW135

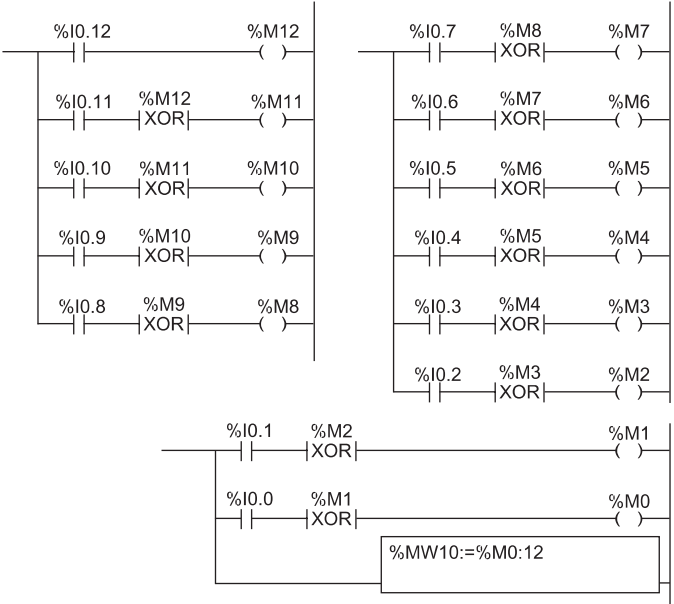
8、外部设备

GRY(将BIN数据转换为格雷码并传送)



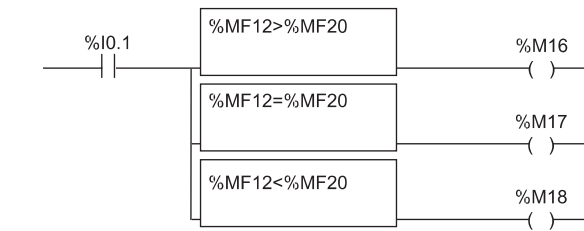
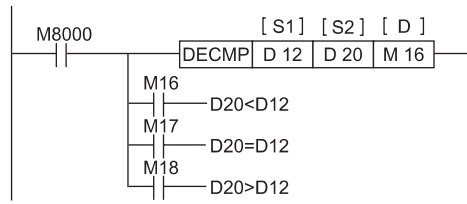
8、外部设备

GBIN(将格雷码数据转换为BIN并传送)



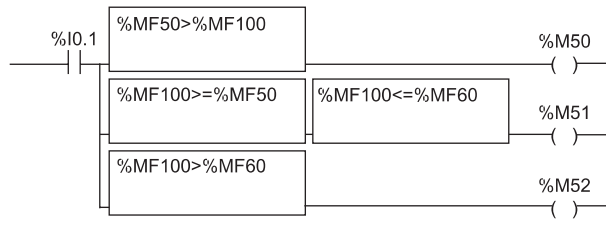
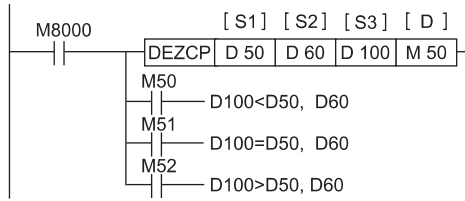
9、浮点小数运算

DECMP(浮点比较)



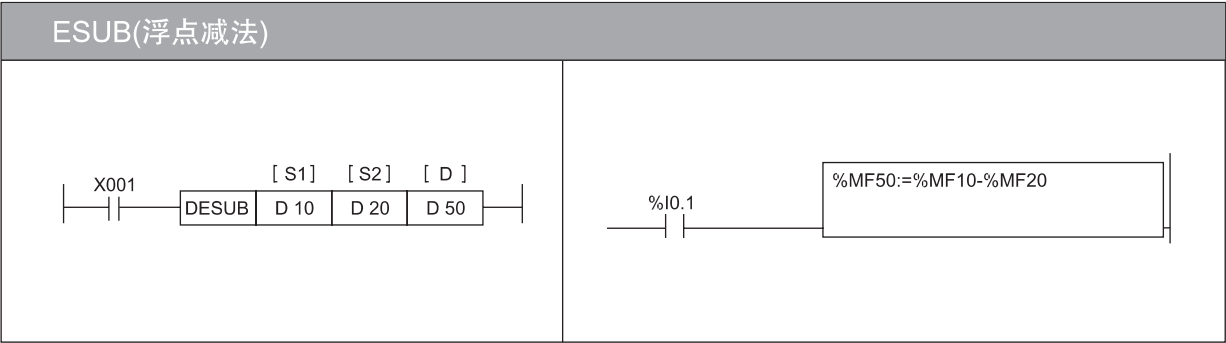
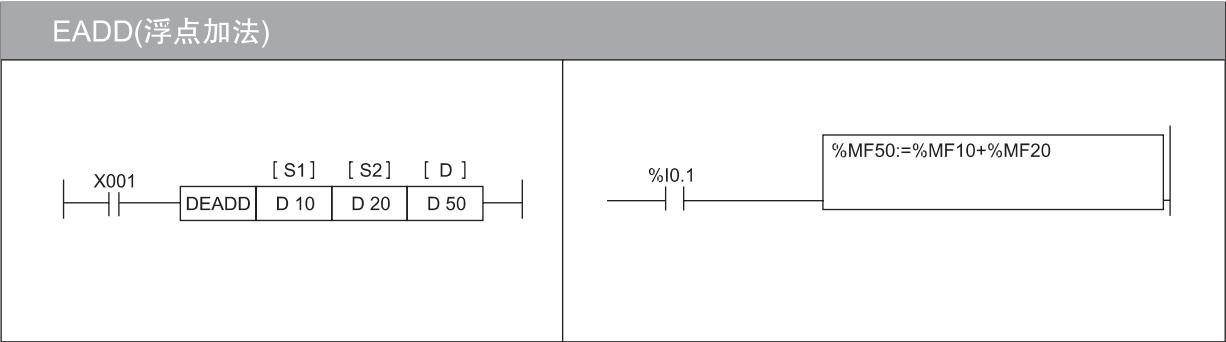
%m16,%m17,%m18用于下一步执行的条件；执行条件%I0.1为任意

DEZCP(浮点区域比较)



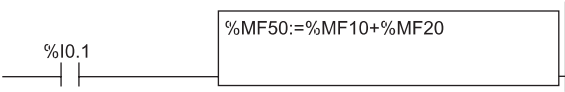
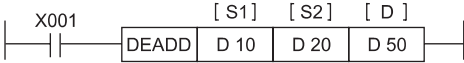
%MF50必须小于%MF60；执行条件%I0.1为任意

9、浮点小数运算

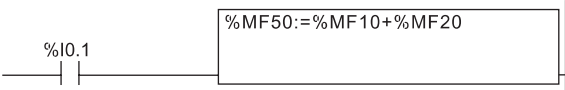
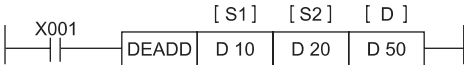


9、浮点小数运算

EMUL(浮点乘法)



EDIV(浮点除法)



9、浮点小数运算

DSIN(浮点SIN运算)



%I0.3

%MF314:=SIN(%MF510)

ESQR(浮点数开平方)

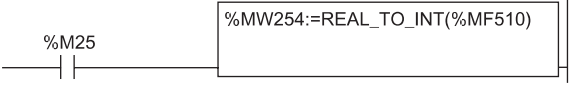


%I0.1

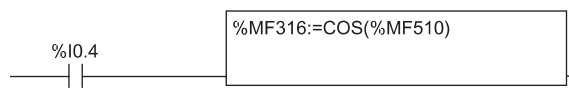
%MF510:=SQRT(%MF302)

9、浮点小数运算

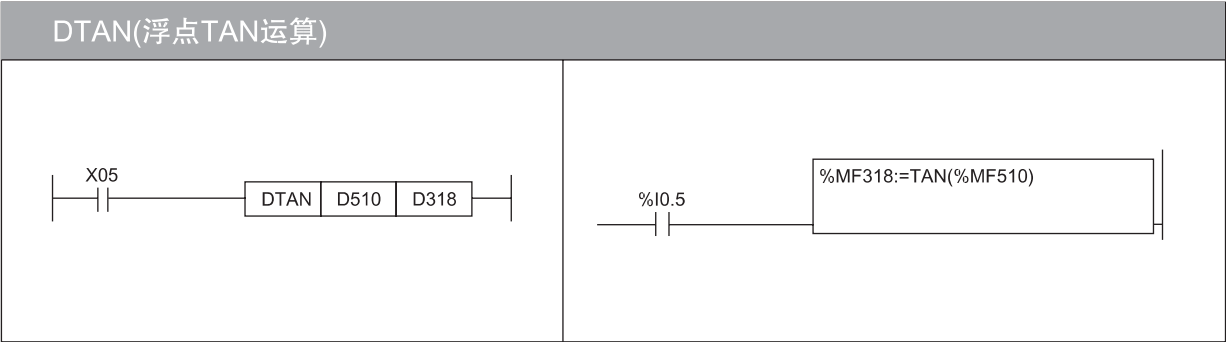
DINT(浮点数→BIN整数)



DCOS(浮点COS运算)

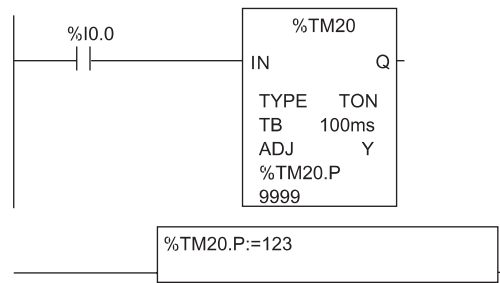


9、浮点小数运算



10、计数器和计时器

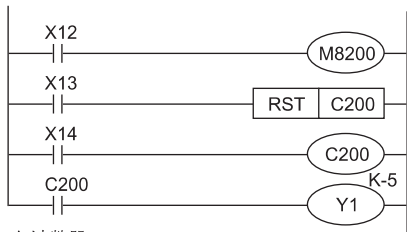
计时器(非保持型)



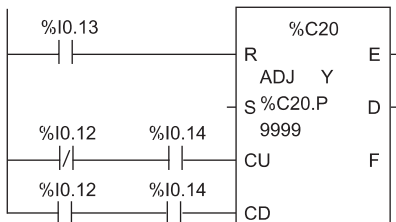
Timer Resolution	FX0(S)	FX0N	FX	FX(2C)	FX2N(C)
100 msec	56 (TO-55)	63 (TO-62)		200 (TO-199)	
10 msec	\24 (T32-55)	\31 (T32-62)		46 (T200-245)	
1 msec	N/A	1 (T63)		N/A	
Retentive 1 msec	N/A	N/A		4 (T246-249)	
Retentive 100 msec	N/A	N/A		6 (T250-255)	

10、计数器和计时器

计数器(双向)



C200为双向计数器。
X12不导通时加计数；X12导通时减计数。
请参考下表中的计数器功能分类。

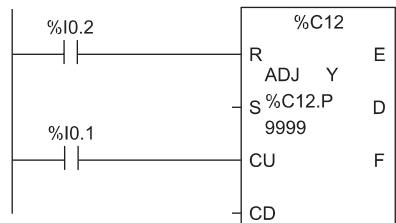
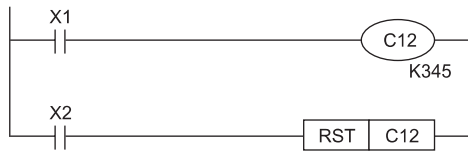


Twidosoft中所有计数器均为双向计数器，共128个

Counter Resolution	FX0(S)	FX0N	FX	FX(2C)	FX2N(C)
General 16bit up counter	14 (C0-13)	16 (C0-15)		100 (C0-99)	
Latched 16bit up counter	2 (C14-15)	16 (C16-31)		100 (C100-199)	
General 32bit bi-directional counter	N/A	N/A		20 (C200-219)	
Latched 32bit bi-directional counter	N/A	N/A		15 (T220-234)	

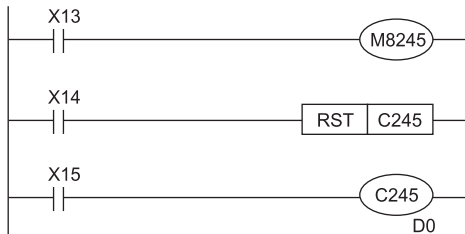
10、计数器和计时器

计数器(普通)



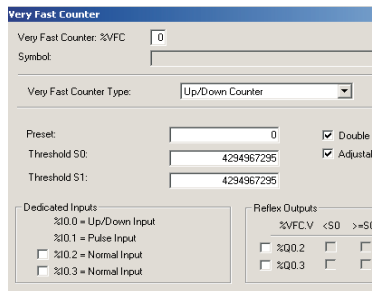
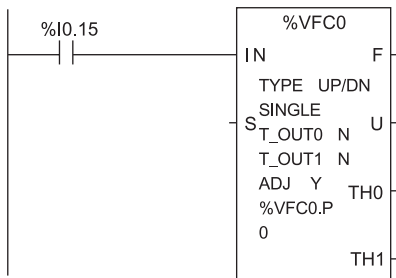
10、计数器和计时器

高速计数器



M8245得电时，C245减计数；M8245不得电则加计数。
X14得电时，C245计数值复位；同时对照下表X3同样可复位该计数器。
X2为计数脉冲。
X15用于启动/选定该计数器；X7为开始计数信号。

配置如下：



%I0.0得电减计数，否则加计数；%I0.1为脉冲输入端

10、计数器和计时器

FX, FX2C 中可用的高速计数器

I N P U T	1 Phase counter user start/reset						1 Phase counter assigned start/reset					2 Phase counter bi-directional					A/B Phase counter				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0 ◆	U/D						U/D			U/D		U	U		U		A	A		A	
X1 P		U/D					R			R		D	D		D		B	B		B	
X2 ◆			U/D					U/D			U/D		R		R			R		R	
X3 ◆				U/D				R		S	R			U	S	U			A		A
X4 P					U/D				U/D					D		D			B		B
X5 P						U/D			R					R		R			R		R
X6										S					S					S	
X7											S					S					S

Key:

U - up counter input

D - down counter input

R - reset counter (input)

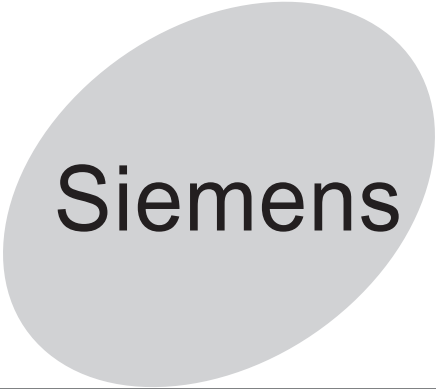
S - start counter (input)

A - A phase counter input

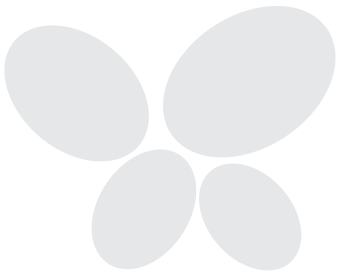
B - B phase counter input

C235

- Counter is backed up / latched

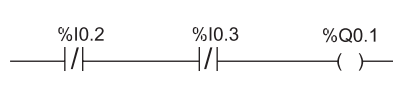
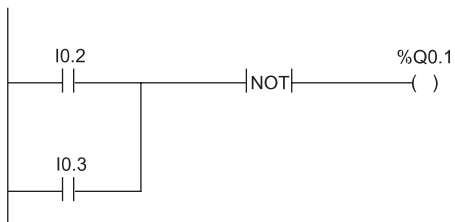
The Siemens logo, consisting of a light gray oval with the word "Siemens" in black text inside it, is positioned in the upper right quadrant of the image.

Siemens



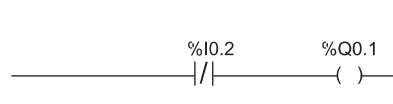
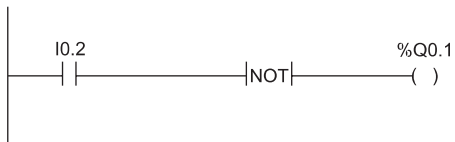
1、位逻辑指令

取反指令



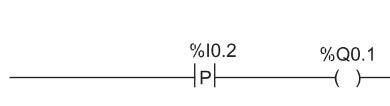
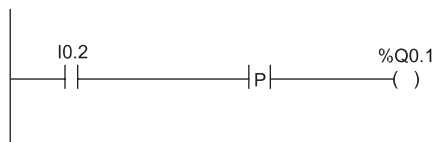
注意：二者的I/O地址定义的方式不同

负跳变触点指令



1、位逻辑指令

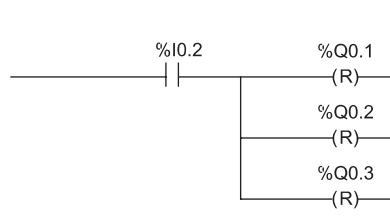
正跳变触点指令



复位指令



该复位指令可以对以Q0.1开始的连续3个输出位复位

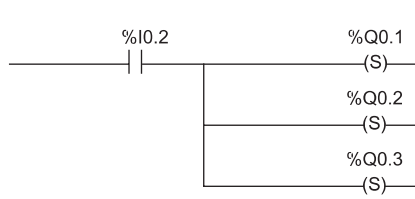


1、位逻辑指令

置位指令



该复位指令可以对以Q0.1开始的连续3个输出位置位

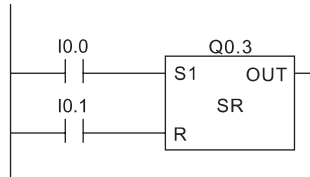


1、位逻辑指令

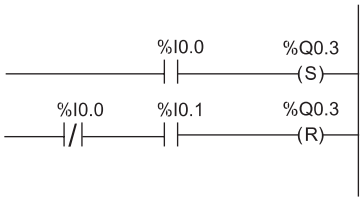
SR置位触发器指令

SR复位指令触发器的原理如下：

S1/%I0.0	R/%I0.1	OUT
0	0	保持前一个状态
0	1	0
1	0	1
1	1	1



RS复位指令是复位优先

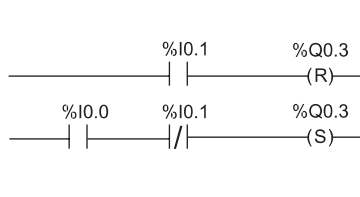
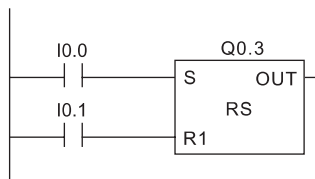


1、位逻辑指令

RS复位触发器指令

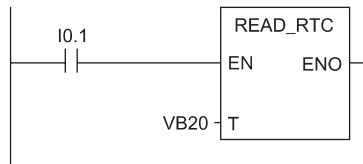
RS置位指令触发器的原理如下：

S1/%I0.0	R/%I0.1	OUT
0	0	保持前一个状态
0	1	0
1	0	1
1	1	0



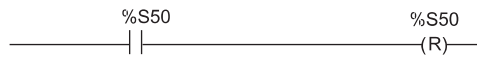
2、时钟指令

读时钟指令

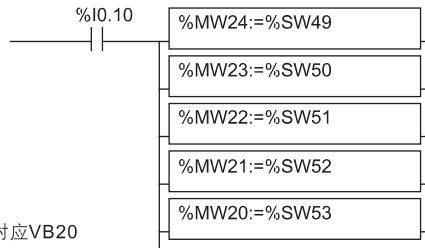


注：读时钟指令是从硬件时钟中读当前时间和日期，并把它装载到一个8字节、起始地址为VB20的时间缓冲区中。
由于221和222两种PLC需要扩展时钟卡，故相应读时钟指令应为READ_RTCX

方法1：点击PLC菜单，再点击“查看控制器”菜单，出现查看控制器界面，在此界面中可查看PLC时钟



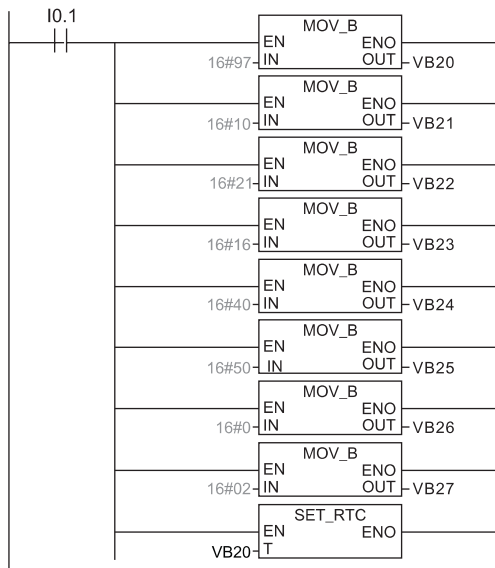
方法2：



%MW20对应VB20

2、时钟指令

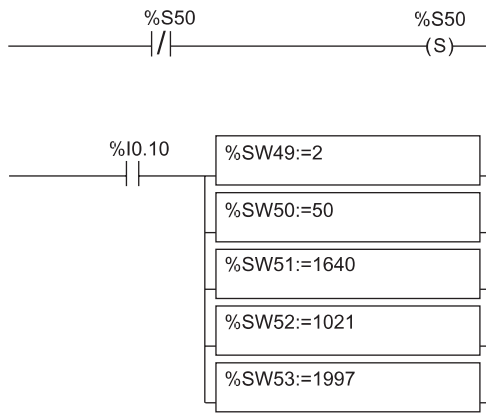
写时钟指令



注：写时钟指令是将当前时间和日期写到硬件时钟中，当前时钟存储在一个8字节、起始地址为VB20的时间缓冲区内。由于221和222两种PLC需要扩展时钟卡，故相应读时钟指令应为SET_RTCX

方法1：点击PLC菜单，再点击“查看控制器”菜单，出现查看控制器界面，在此界面中可设定PLC时钟

方法2：

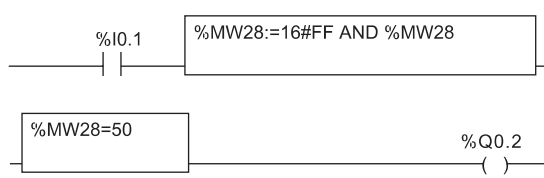


3、比较指令

字节比较指令—等于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等

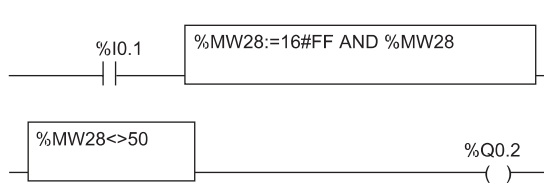


Twido不包含字节，故要进行字节比较，须取字的低位进行比较

字节比较指令—不等于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等



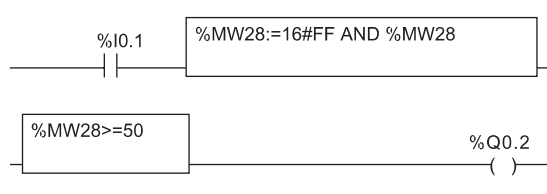
Twido不包含字节，故要进行字节比较，须取字的低位进行比较

3、比较指令

字节比较指令一大于等于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等

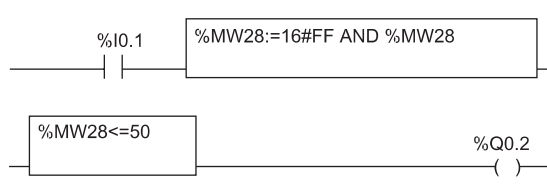


Twido不包含字节，故要进行字节比较，须取字的低位进行比较

字节比较指令一小于等于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等



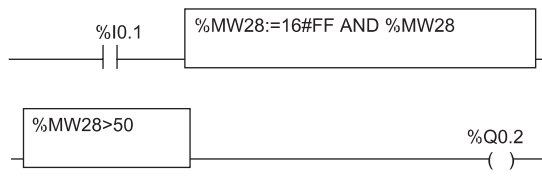
Twido不包含字节，故要进行字节比较，须取字的低位进行比较

3、比较指令

字节比较指令—大于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等

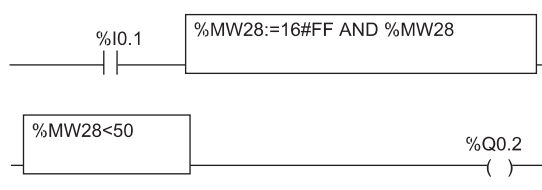


Twido不包含字节，故要进行字节比较，须取字的低位进行比较

字节比较指令—小于



任何字节或符合范围的常数都可以进行比较，
如IB,QB,SMB,MB等



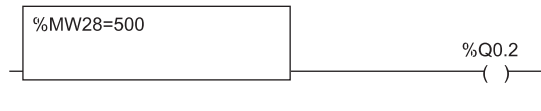
Twido不包含字节，故要进行字节比较，须取字的低位进行比较

3、比较指令

单字比较—等于



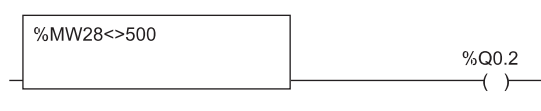
任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等



单字比较—不等于



任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等

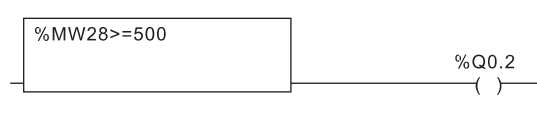


3、比较指令

单字比较—大于等于



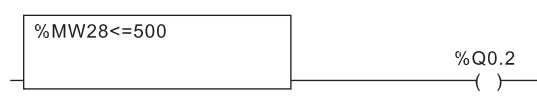
任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等



单字比较—小于等于



任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等

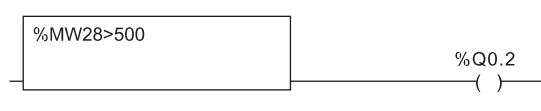


3、比较指令

单字比较—大于



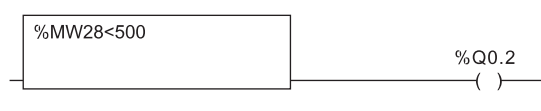
任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等



单字比较—小于



任何单字或符合范围的常数都可以进行比较，
如IW,QW,SMW,MW等

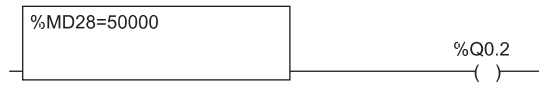


3、比较指令

双字比较—等于



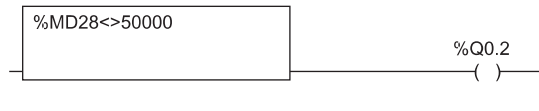
任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等



双字比较—不等于



任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等

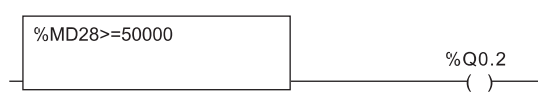


3、比较指令

双字比较—大于等于



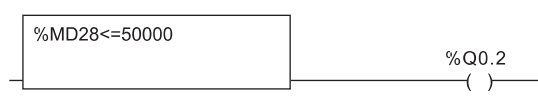
任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等



双字比较—小于等于



任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等

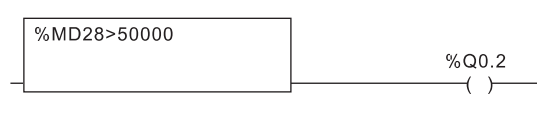


3、比较指令

双字比较—大于



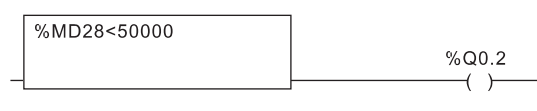
任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等



双字比较—小于



任何类型双字或符合范围的常数都可以进行比较，
如ID,QD,SMD,MD等

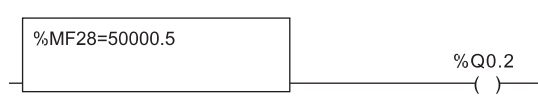


3、比较指令

实数比较—等于



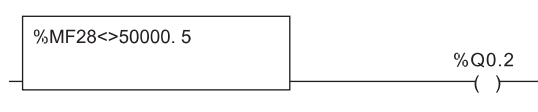
任何类型实数或符合范围的实数数都可以进行比较，
如ID,QD,SMD,MD等



实数比较—不等于



任何类型实数或符合范围的实数数都可以进行比较，
如ID,QD,SMD,MD等

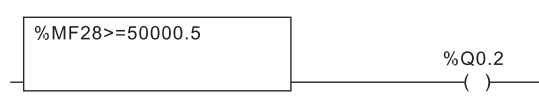


3、比较指令

实数比较—大于等于



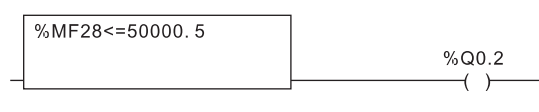
任何类型实数或符合范围的实数都可以进行比较，
如ID,QD,SMD,MD等



实数比较—小于等于



任何类型实数或符合范围的实数都可以进行比较，
如ID,QD,SMD,MD等

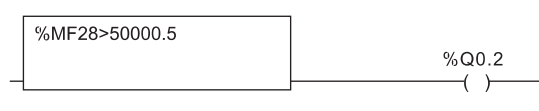


3、比较指令

实数比较—大于



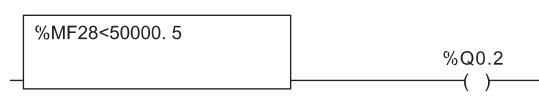
任何类型实数或符合范围的实数数都可以进行比较，
如ID,QD,SMD,MD等



实数比较—小于

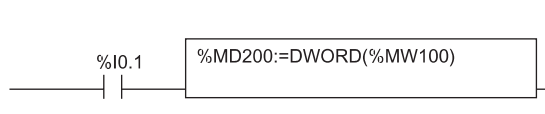
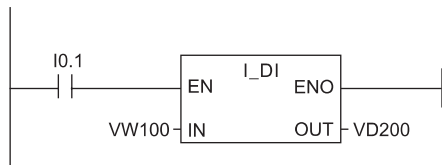


任何类型实数或符合范围的实数数都可以进行比较，
如ID,QD,SMD,MD等

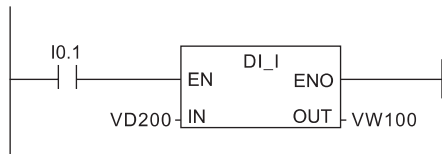


4、转换指令

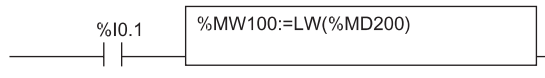
单字转换为双字



双字转换为单字

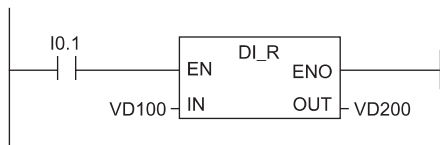


如果所转换的数值太大，以至于无法在输出中表示，
则溢出标志位置位，并且输出不会改变

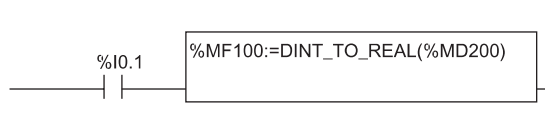


4、转换指令

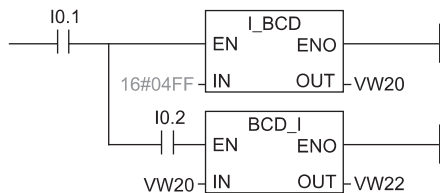
双字转换为实数



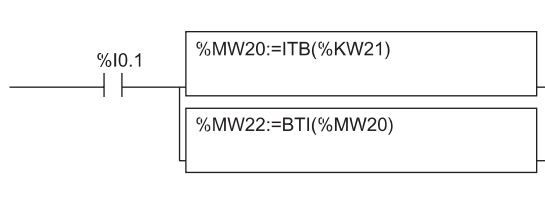
如果所转换的数值太大，以至于无法在输出中表示，
则溢出标志位置位，并且输出不会改变



BCD与整数的转化

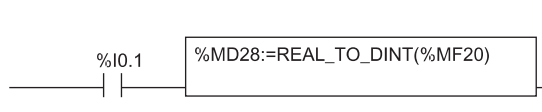
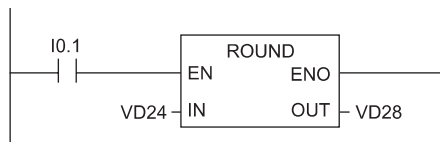


先将整数转化为BCD数，然后将BCD数转化为整数

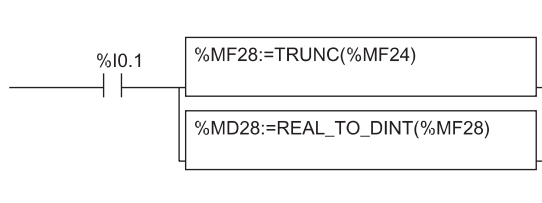
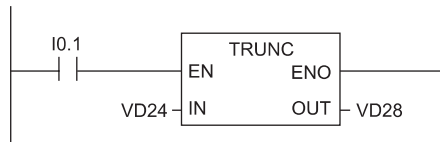


4、转换指令

四舍五入指令

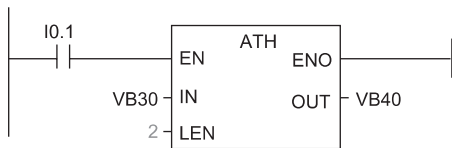


取整指令

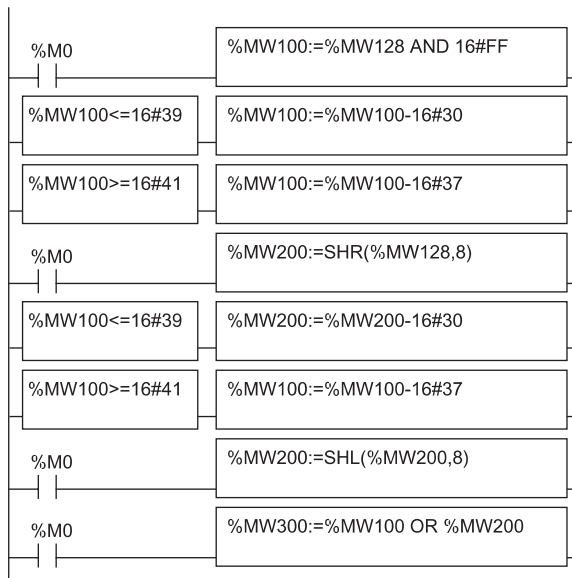


4、转换指令

ASCII转16进制整数



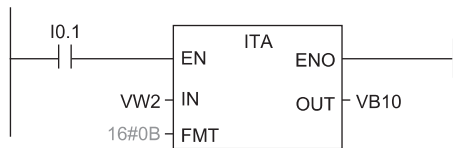
该指令是将VB30为首地址的ASCII字符串(长度为2个字节)转化为16进制整数,例如VB30,VB31的值分别为33, 45, 转换为整数就是3, E



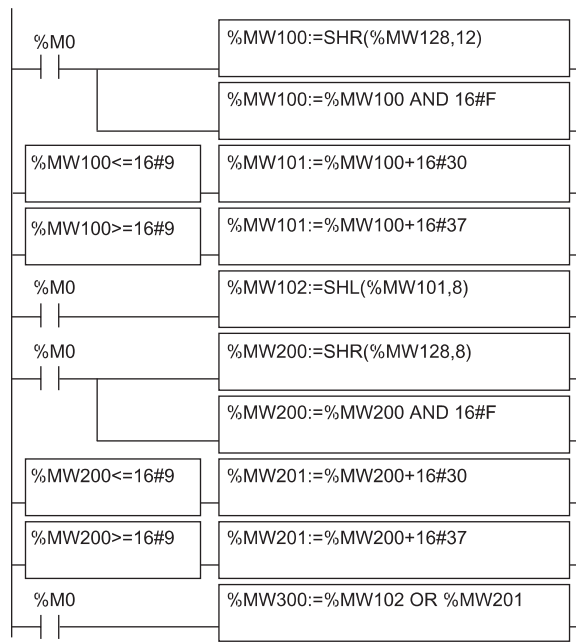
假设ASCII字符串33, 45存储在 %MW128中

4、转换指令

整数转ASCII

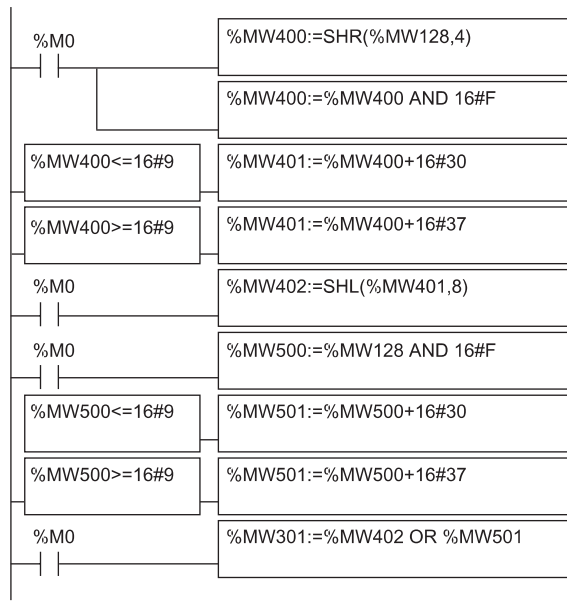


双整数转换成ASCII的对译指令也是类似，设VW2的值为“12AB”转换过来则是31，32，41，42，Twido对译指令的结果存储在%MW300和%MW301



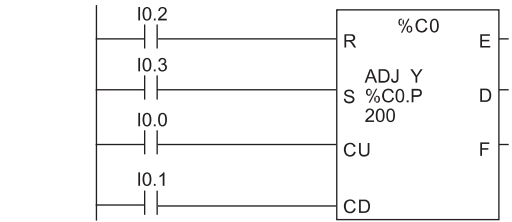
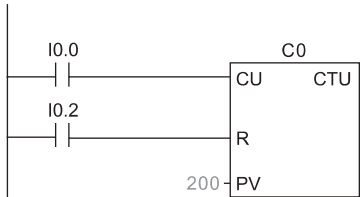
4、转换指令

整数转ASCII



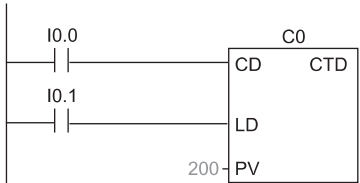
5、计数器指令

增计数

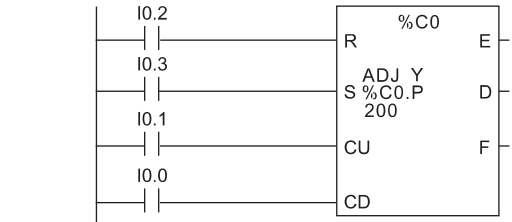


给CU端脉冲，则计数器增计数

减计数



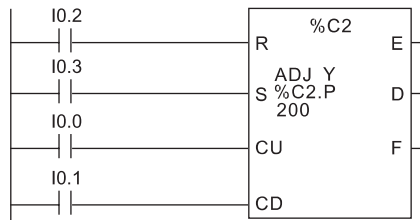
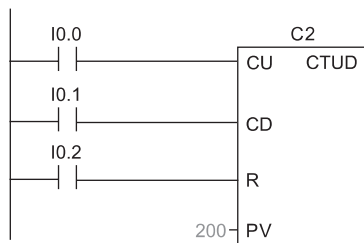
I0.1上升沿时，PV值存入计数器，通过I0.0给出减计数脉冲



给CD端脉冲，则计数器减计数

5、计数器指令

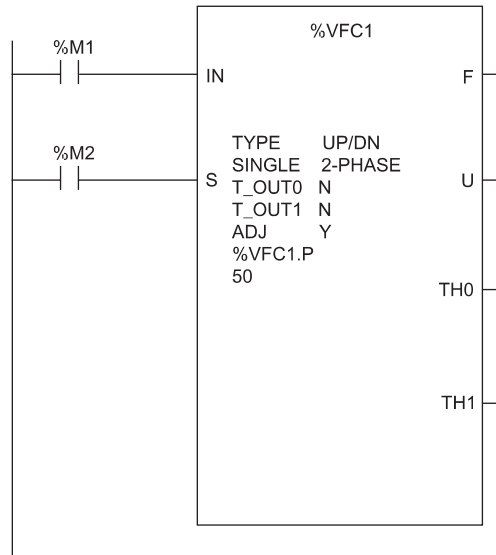
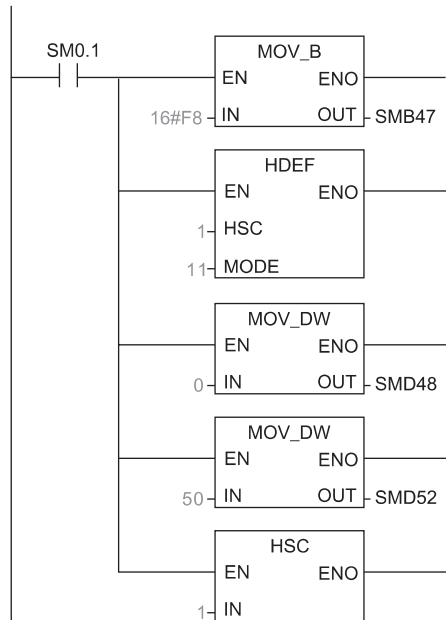
增减计数器



给CU端脉冲，则计数器增计数，给CD端脉冲，则减计数器

5、计数器指令

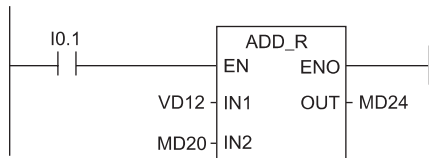
高速计数器指令



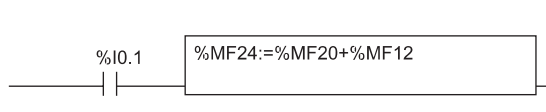
在VFC功能块中配置阈值和计数器类型实现

6、浮点数计算

浮点数加法

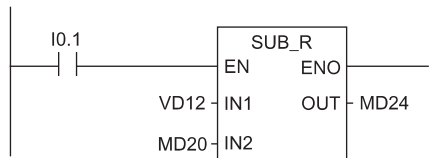


各种存储类型的实数变量都可以相加，结果也可存放在不同的存储器

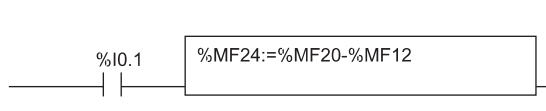


Twido实数只能存放在%MF存储器中

浮点数减法



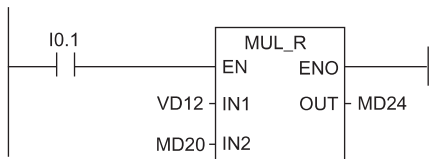
各种存储类型的实数变量都可以相减，结果也可存放在不同的存储器



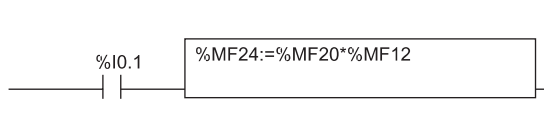
Twido实数只能存放在%MF存储器中

6、浮点数计算

浮点数乘法

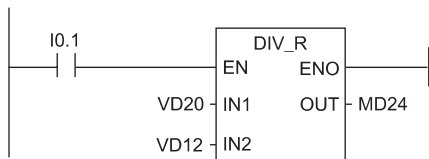


各种类型类型的实数变量都可以相乘，结果也可存放在不同的存储器

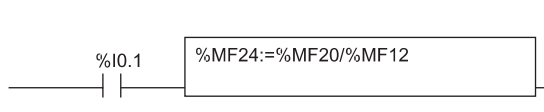


Twido实数只能存放在%MF中

浮点数除法



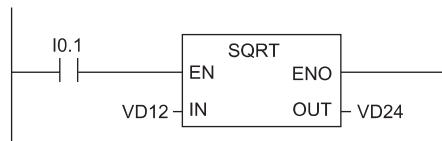
各种存储类型的实数变量都可以相除，结果也可存放在不同的存储器



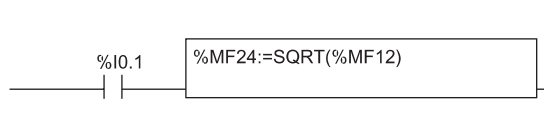
Twido实数只能存放在%MF中

6、浮点数计算

求平方根

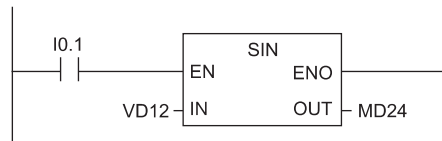


各种存储类型的实数变量都可以求平方根，结果也可存放在不同的存储器

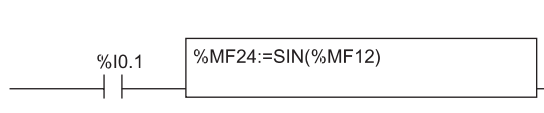


Twido实数只能存放在%MF中

求正弦



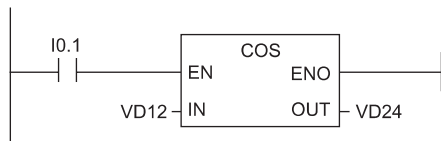
各种存储类型的实数变量都可以求正弦，结果也可存放在不同的存储器



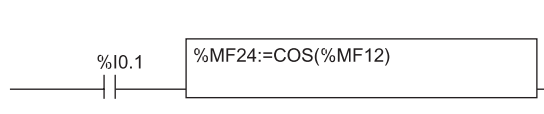
Twido实数只能存放在%MF中

6、浮点数计算

求余弦

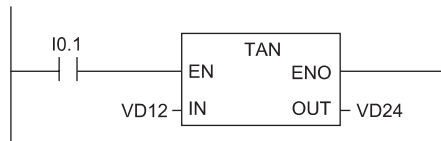


各种存储类型的实数变量都可以求余弦，结果也可存放在不同的存储器

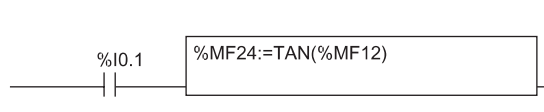


Twido实数只能存放在%MF中

求正切



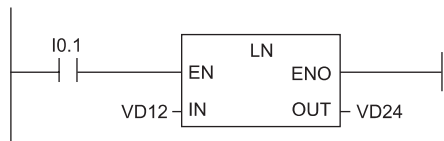
各种存储类型的实数变量都可以求正切，结果也可存放在不同的存储器



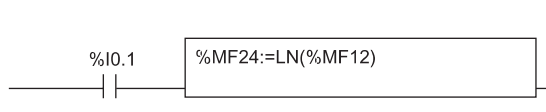
Twido实数只能存放在%MF中

6、浮点数计算

求对数

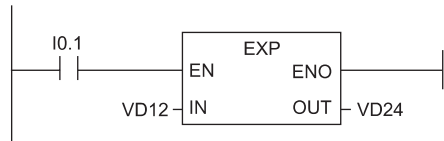


各种存储类型的实数变量都可以求对数，结果也可存放在不同的存储器

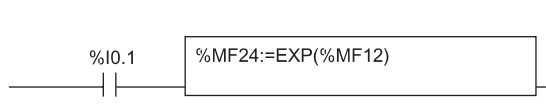


Twido实数只能存放在%MF中

求指数



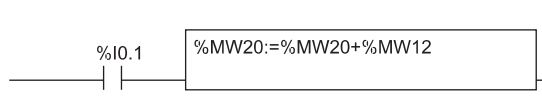
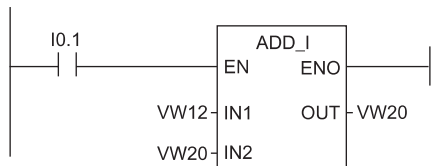
各种存储类型的实数变量都可以求指数，结果也可存放在不同的存储器



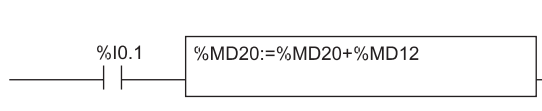
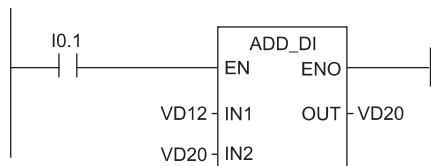
Twido实数只能存放在%MF中

7、整数运算

整数加法

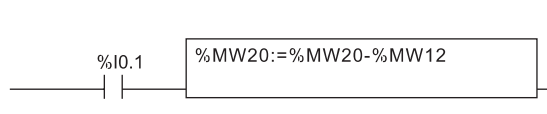
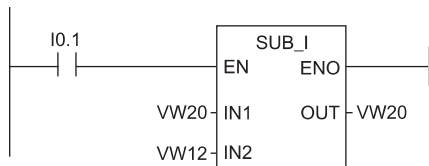


双字整数加法

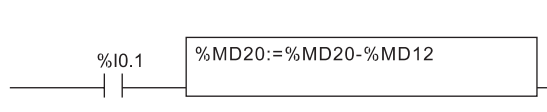
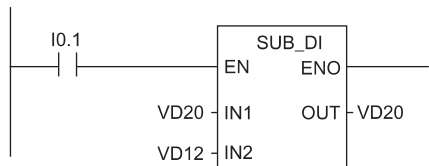


7、整数运算

整数减法

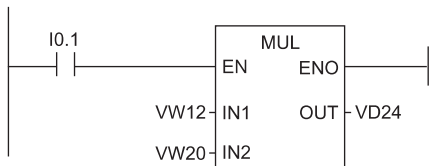


双整数减法

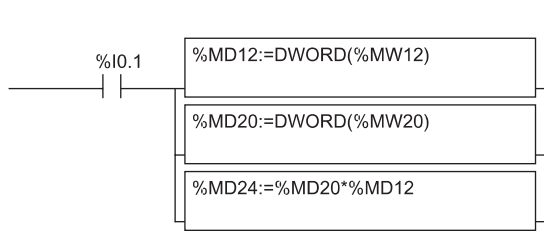


7、整数运算

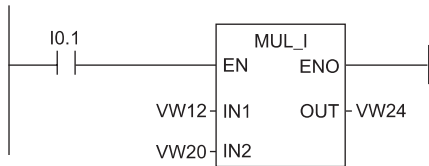
乘法



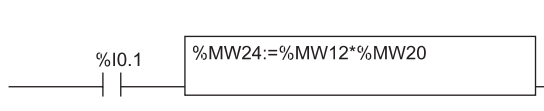
该指令要求两个输入值都为整数，输出值则为双整数



整数乘法

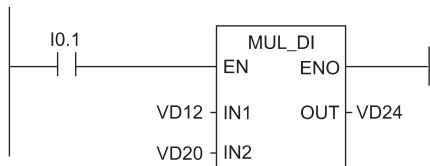


该指令要求两个输入值都为整数，输出值也为整数

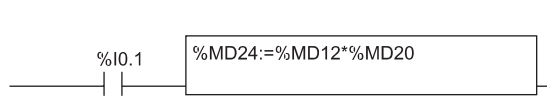


7、整数运算

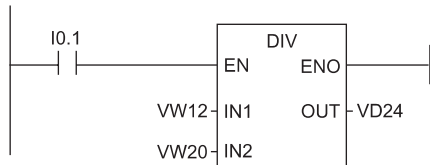
双整数乘法



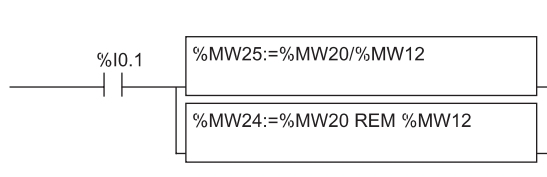
该指令要求两个输入值都为双整数，输出值也为双整数，
双整数的变量地址是4的倍数



除法



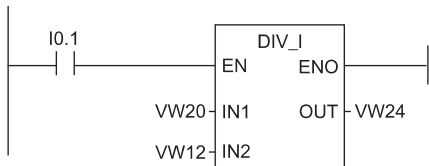
该指令要求两个输入值都为整数，输出值也为双整数，
双整数的变量地址都是4的倍数



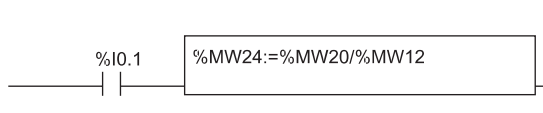
双字%MD24高地址存放余数,低地址存放商

7、整数运算

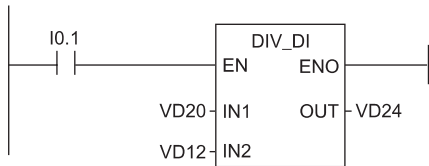
整数除法



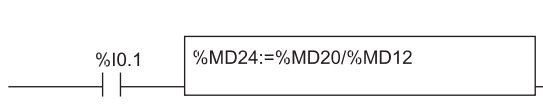
该指令要求两个输入值都为整数，输出值也为整数



双整数除法

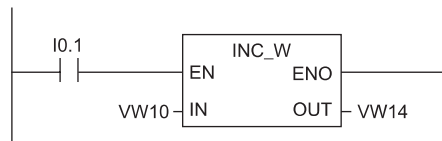


该指令要求两个输入值都为双整数，输出值也为双整数

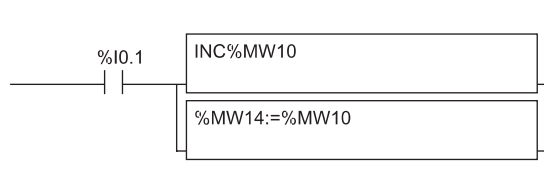


7、整数运算

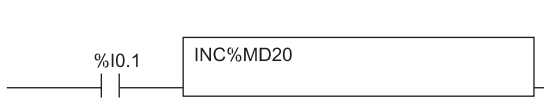
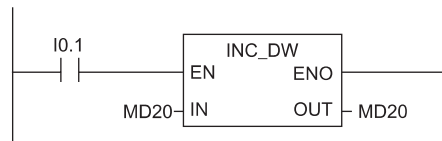
字加1指令



对于字节加1指令INC_B,也可以用该条对应的Twido指令对译

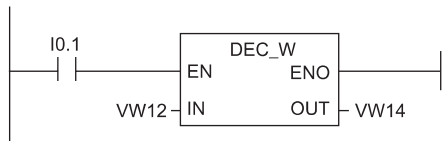


双字加1指令

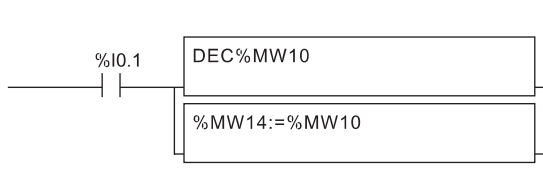


7、整数运算

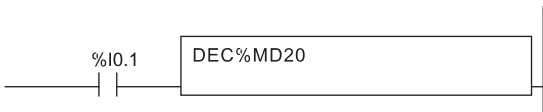
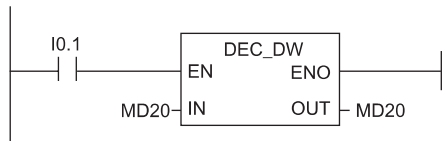
字减1指令



对于字节减1指令INC_B,也可以用该条对应的Twido指令对译

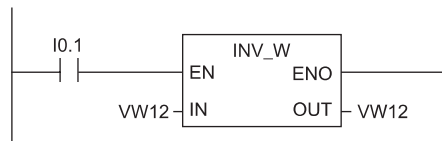


双字减1指令

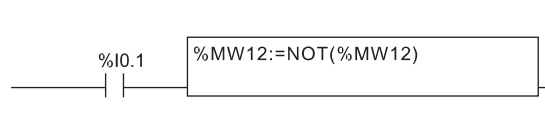


8、逻辑操作

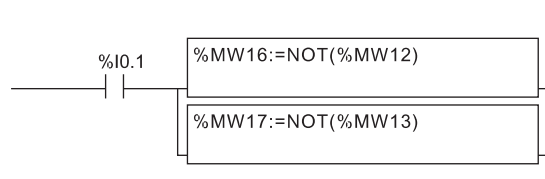
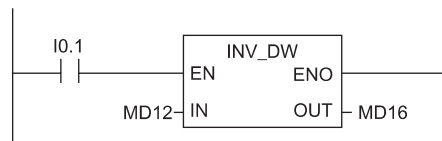
字取反



对于字节取反，Twido也可以用该条对应指令对译

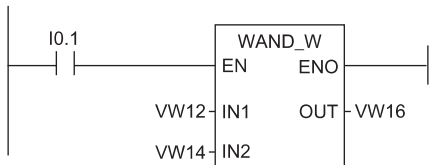


双字取反

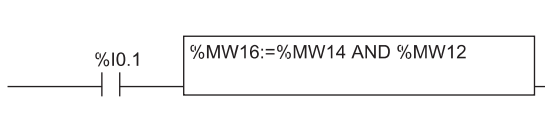


8、逻辑操作

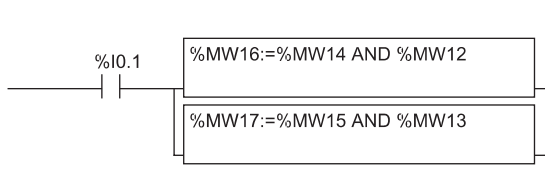
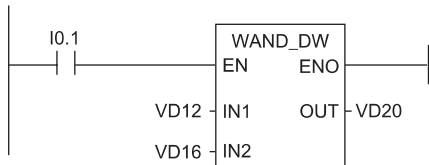
字求与指令



对于字节求与，Twido也可以用该条对应指令对译

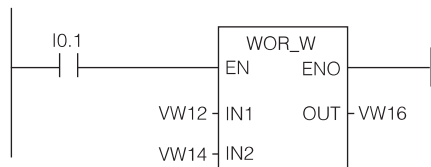


双字求与指令

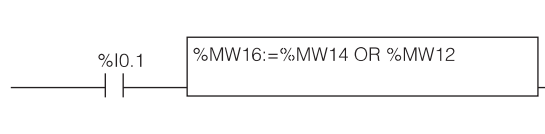


8、逻辑操作

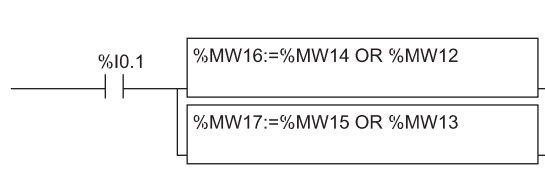
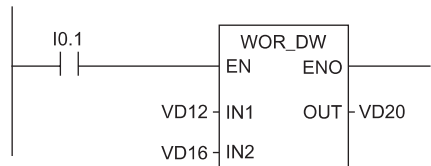
字求或指令



对于字节求或，Twido也可以用该条对应指令对译



双字求或指令



8、逻辑操作

字求异或

The diagram shows a normally open contact labeled I0.1 connected to the EN input of a function block labeled WXOR_W. The block has two inputs: IN1 (labeled VW12) and IN2 (labeled VW14). The output ENO is connected to a terminal, and the result is output to VW16.

对于字节求或，Twido也可以用该条对应

The diagram shows a normally open contact labeled %I0.1 connected to a Twido instruction box containing the code: %MW16:=%MW14 XOR %MW12.

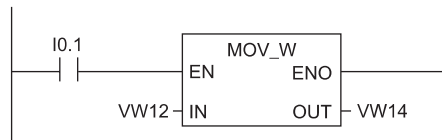
双字求异或

The diagram shows a normally open contact labeled I0.1 connected to the EN input of a function block labeled WXOR_DW. The block has two inputs: IN1 (labeled VD12) and IN2 (labeled VD16). The output ENO is connected to a terminal, and the result is output to VD20.

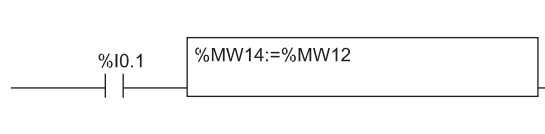
The diagram shows a normally open contact labeled %I0.1 connected to two Twido instruction boxes. The first box contains the code: %MW16:=%MW14 XOR %MW12. The second box contains the code: %MW17:=%MW15 XOR %MW13.

9、MOV指令

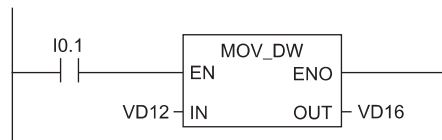
MOV字



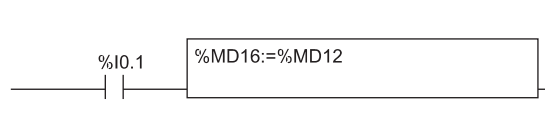
对于字节移动，也可以采用该条对应的Twido指令对译



MOV双字



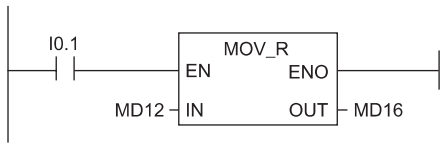
注：此处%MD16与%MD12是连续的



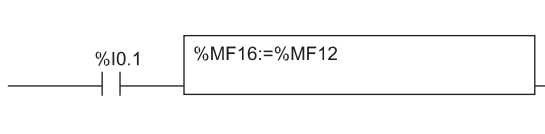
注：此处%MD16与%MD12间隔一个双字

9、MOV指令

MOV实数

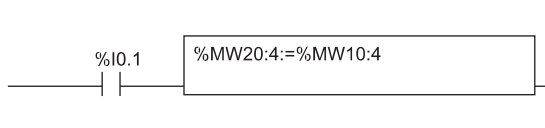
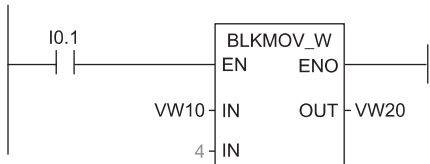


注：此处%MD16与%MD12是连续的



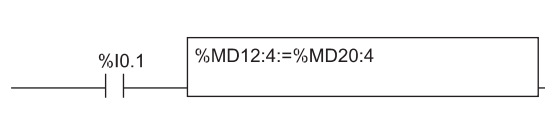
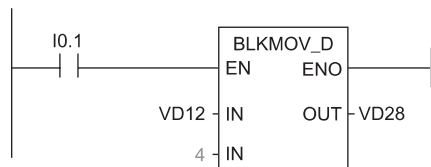
注：此处%MF16与%MF12间隔一个%MF14

BLKMOV字

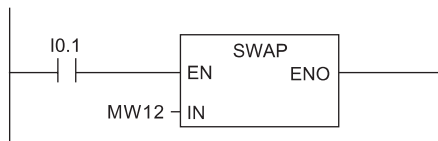


9、MOV指令

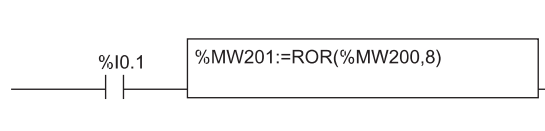
BLKMOV双字



WAP字节交换指令

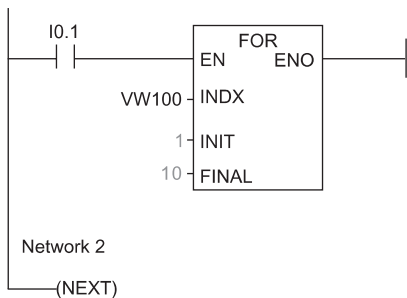


该指令是交换字的高低位字节

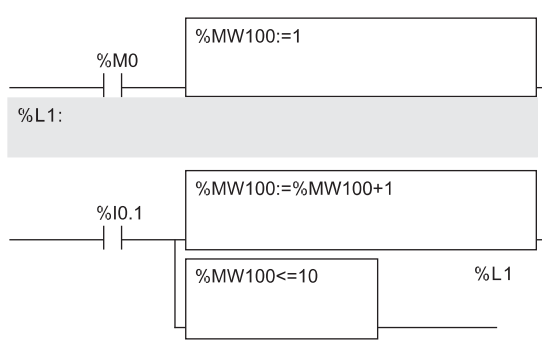


10、PROGRAM CONTROL

FOR/NEXT循环



FOR和NEXT指令可以描述需重复进行一定次数的循环体，每条FOR指令必须对应一条NEXT指令。该循环指令是按照INIT和FINAL循环执行FOR和NEXT之间的指令



10、PROGRAM CONTROL

STOP指令

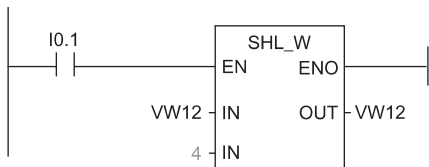


Input Configuration						
	Used By	Symbol	Filtering	Latch?	Run/Stop?	Deactivation
%I0.0			3 ms	☐	☐	
%I0.1			3 ms	☐	☒	
%I0.2			3 ms	☐	☐	Not used
%I0.3			3 ms	☐	☐	Not used
%I0.4			3 ms	☐	☐	Not used
%I0.5			3 ms	☐	☐	Not used

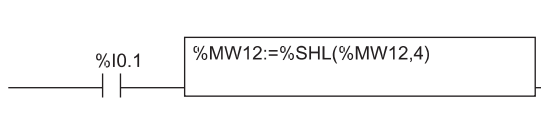
可利用外部输入端子控制PLC的通断，例如可设定%I0.1

11、SHIFT/ROTATE

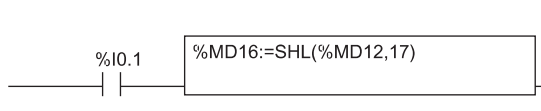
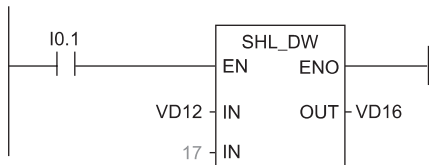
SHL_W字左移



对于字节左移指令SHL_B，也可以用Twido与此对应的指令对译，只是用Twido的字来代表S7-200的字节即可符合编程要求

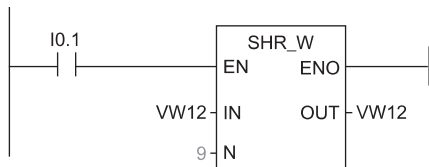


SHL_DW双字左移

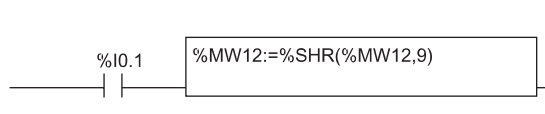


11、SHIFT/ROTATE

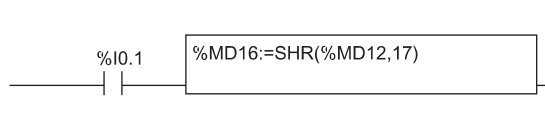
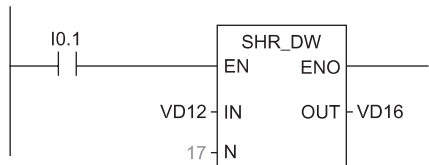
SHR_W字右移



对于字节右移指令SHR_B, 也可以用Twido与此对应的指令对译, 只是用Twido的字来代表S7-200的字节即可符合编程要求

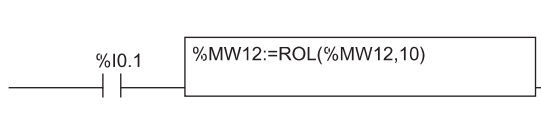
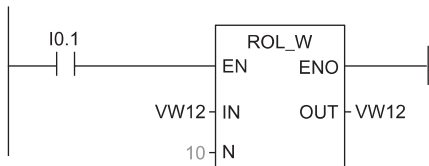


SHR_DW双字右移

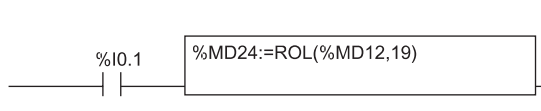
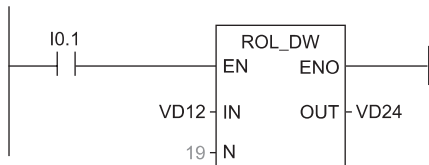


11、SHIFT/ROTATE

ROL_W字循环左移

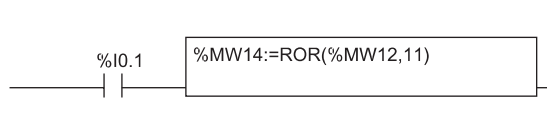
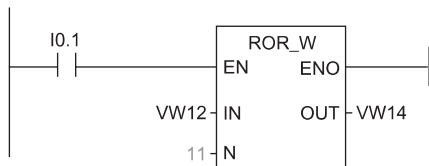


ROL_DW双字循环左移

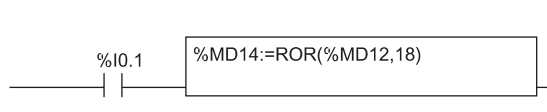
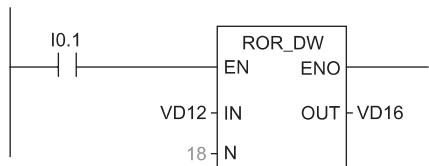


11、SHIFT/ROTATE

ROR_W字循环右移

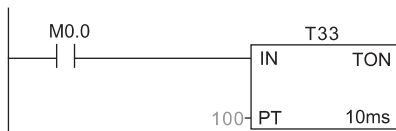


ROR_DW双字循环右移

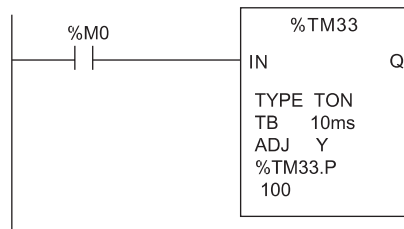


12、定时器

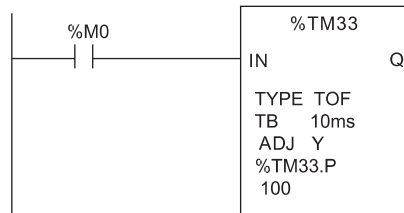
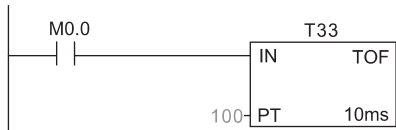
TON接通延时



S7-200的定时器的时基选择是由定时器号来决定的，
如果本例中T33仅对应10ms的时基



TOF断开延时



施耐德电气公司
Schneider Electric China
www.schneider-electric.cn

北京市朝阳区将台路2号
和乔丽晶中心施耐德大厦
邮编: 100016
电话: (010) 8434 6699
传真: (010) 8450 1130

Schneider Building ,Chateau Regency,
No.2 Jiangtai Road, Chaoyang District,
Beijing 100016 China.
Tel: (010) 8434 6699
Fax: (010) 8450 1130

由于标准和材料的变更，文中所述特性和本资料中的图像只有经过我们的业务部门确认以后，才对我们有约束。



本手册采用生态纸印刷