



# **LS21 系列可编程序控制器 ( PLC )**

## **用户手册【指令篇】**

杭州良石智能技术有限公司

修订记录

版本号	发布日期	修订内容
Ver1.0.0	2015-10-17	创建本文档















注：此处只记载针对文档本身的主要修订记录

- ：特殊标示符(表示指令已经完成)
- ：特殊标示符(表示指令未完成)

黄底色：表示因果关系

红色字：表示指令书写

目录

1	.指令目录 .....	7
1.1	基本指令目录 .....	7
1.2	梯形图步进指令目录 .....	8
1.3	应用指令目录 .....	8
2	.基本指令 .....	11
2.1	LD 系列连接到母线的指令 .....	11
2.1.1	 取指令 .....	11
2.1.2	 取反转指令 .....	11
2.1.3	 取脉冲上升沿指令 .....	12
2.1.4	 取脉冲下降沿指令 .....	12
2.2	AND 系列串联连接的指令 .....	13
2.2.1	 与指令 .....	13
2.2.2	 与反转指令 .....	13
2.2.3	 与脉冲上升沿指令 .....	14
2.2.4	 与脉冲下降沿指令 .....	15
2.3	OR 系列并联连接的指令 .....	15
2.3.1	 或指令 .....	15
2.3.2	 或反转指令 .....	16
2.3.3	 或脉冲上升沿指令 .....	17
2.3.4	 或脉冲下降沿指令 .....	17
2.4	回路块指令 .....	18
2.4.1	 回路块或指令 .....	18
2.4.2	 回路块与指令 .....	18
2.5	线圈控制指令 .....	19

2.5.1	<b>OUT</b>	输出指令 .....	19
2.5.2	<b>SET</b>	置位指令 .....	19
2.5.3	<b>RST</b>	复位指令 .....	20
<b>2.6</b>	<b>脉冲输出指令 .....</b>	<b>20</b>	
2.6.1	<b>PLS</b>	上升沿脉冲输出指令 .....	20
2.6.2	<b>PLF</b>	下降沿脉冲输出指令 .....	21
<b>2.7</b>	<b>主控指令 .....</b>	<b>22</b>	
2.7.1	<b>MC</b>	主控指令 .....	22
2.7.2	<b>MCR</b>	主控复位指令 .....	22
<b>2.8</b>	<b>堆栈指令 .....</b>	<b>23</b>	
2.8.1	<b>MPS</b>	进栈指令 .....	23
2.8.1	<b>MRD</b>	读栈指令 .....	23
2.8.1	<b>MPP</b>	出栈指令 .....	24
<b>2.9</b>	<b>其他基本指令 .....</b>	<b>25</b>	
2.9.1	<b>INV</b>	取反指令 .....	25
2.9.2	<b>NOP</b>	空操作指令 .....	25
2.9.3	<b>END</b>	结束指令 .....	25
<b>3</b>	<b>梯形图步进指令 .....</b>	<b>26</b>	
<b>3.1</b>	<b>步进梯形图指令 .....</b>	<b>26</b>	
3.1.1	<b>STL</b>	步进梯形图指令 .....	26
3.1.2	<b>RET</b>	返回指令 .....	26
<b>4</b>	<b>应用指令 .....</b>	<b>28</b>	
<b>4.1</b>	<b>程序流程指令 .....</b>	<b>28</b>	
4.1.1	<b>CJ</b>	条件跳转指令 .....	28
4.1.2	<b>CALL</b>	子程序调用指令 .....	29
4.1.3	<b>SRET</b>	子程序返回指令 .....	29
4.1.4	<b>FEND</b>	主程序结束指令 .....	30
4.1.5	<b>FOR</b>	循环范围开始指令 .....	31
4.1.6	<b>NEXT</b>	循环范围结束指令 .....	31

<b>4.2</b>	<b>传送与比较指令</b>	<b>32</b>
4.2.1	<b>CMP</b> 比较指令	32
4.2.2	<b>ZCP</b> 区域比较指令	32
4.2.3	<b>MOV</b> 传送指令	33
4.2.4	<b>SMOV</b> 位移动指令	33
4.2.5	<b>CML</b> 反相传送指令	34
4.2.6	<b>BMOV</b> 成批传送指令	35
4.2.7	<b>FMOV</b> 多点传送指令	36
4.2.8	<b>XCH</b> 交换指令	36
4.2.9	<b>BCD</b> BCD 交换指令	37
4.2.10	<b>BIN</b> BIN 交换指令	38
<b>4.3</b>	<b>四则逻辑运算指令</b>	<b>39</b>
4.3.1	<b>ADD</b> BIN 加法运算指令	39
4.3.2	<b>SUB</b> BIN 减法运算指令	39
4.3.3	<b>MUL</b> BIN 乘法运算指令	40
4.3.4	<b>DIV</b> BIN 除法运算指令	41
4.3.5	<b>INC</b> BIN 增加指令	41
4.3.6	<b>DEC</b> BIN 减少指令	42
4.3.7	<b>AND</b> 逻辑与指令	43
4.3.8	<b>OR</b> 逻辑或指令	43
4.3.9	<b>XOR</b> 逻辑异或指令	44
4.3.10	<b>NEG</b> 求补指令	44
<b>4.4</b>	<b>循环移位</b>	<b>45</b>
4.4.1	<b>ROR</b> 右回转指令	45
4.4.2	<b>ROL</b> 左回转指令	46
4.4.3	<b>RCR</b> 带进位右回转指令	46
4.4.4	<b>RCL</b> 带进位左回转指令	47
4.4.5	<b>SFTR</b> 位右移指令	48
4.4.6	<b>SFTL</b> 位左移指令	48
4.4.7	<b>WSFR</b> 字右移指令	49
4.4.8	<b>WSFL</b> 字左移指令	50
4.4.9	<b>SFWR</b> 移位写入指令	50
4.4.10	<b>SFRD</b> 移位读出指令	51
<b>4.5</b>	<b>数据处理指令</b>	<b>51</b>
4.5.1	<b>ZRST</b> 批次复位指令	51
4.5.2	<b>DECO</b> 译码指令	52

4.5.3	<b>ENCO</b>	编码指令 .....	53
4.5.4	<b>SUM</b>	ON 位数 .....	53
4.5.5	<b>BON</b>	ON 位判断 .....	54
4.5.6	<b>MEAN</b>	平均值 .....	54
4.5.7	<b>ANS</b>	报警的设置 .....	55
4.5.8	<b>ANR</b>	报警的设置 .....	56
4.5.9	<b>SQR</b>	BIN 数据开方运算 .....	56
4.5.10	<b>FLT</b>	BIN 整数->2 进制浮点转换指令 .....	57
<b>4.6</b>	<b>高速处理</b>	.....	<b>57</b>
4.6.1	<b>PLSY</b>	脉冲输出指令 .....	58
4.6.2	<b>PWM</b>	脉宽调制指令 .....	59
4.6.3	<b>PWM</b>	带加减速脉冲输出指令 .....	59
<b>4.7</b>	<b>外围设备 I/O</b>	.....	<b>60</b>
4.7.1	<b>FROM</b>	BFM 读出指令 .....	60
4.7.2	<b>TO</b>	BFM 写入指令 .....	61
<b>4.8</b>	<b>外围设备 SER</b>	.....	<b>61</b>
4.8.1	<b>ASCII</b>	HEX->ASCII 转换指令 .....	61
4.8.2	<b>CCD</b>	校验码指令 .....	62
4.8.3	<b>PID</b>	PID 运算指令 .....	63
<b>4.9</b>	<b>浮点数指令</b>	.....	<b>64</b>
4.9.1	<b>ECMP</b>	二进制浮点比较指令 .....	64
4.9.2	<b>EZCP</b>	二进制浮点数区域比较指令 .....	65
4.9.3	<b>EBCD</b>	二进制浮点转十进制浮点 .....	65
4.9.4	<b>EBIN</b>	十进制浮点转二进制浮点 .....	66
4.9.5	<b>EADD</b>	二进制浮点数加法指令 .....	66
4.9.6	<b>ESUB</b>	二进制浮点数减法指令 .....	67
4.9.7	<b>EMUL</b>	二进制浮点数乘法指令 .....	67
4.9.8	<b>EDIV</b>	二进制浮点数除法指令 .....	68
4.9.9	<b>ESQR</b>	十进制浮点开方 .....	68
4.9.10	<b>INT</b>	十进制浮点转 BIN 整数 .....	69
4.9.11	<b>SIN</b>	浮点运算 .....	70
4.9.12	<b>COS</b>	浮点运算 .....	70
4.9.13	<b>TAN</b>	浮点运算 .....	71
<b>4.10</b>	<b>上下字节变换</b>	.....	<b>71</b>

4.10.1	<b>SWAP</b> 上下字节变换指令 .....	71
<b>4.11</b>	<b>时钟连算指令</b> .....	<b>72</b>
4.11.1	<b>TCMP</b> 时钟数据比较指令 .....	72
4.11.2	<b>TZCP</b> 时钟数据区间比较指令 .....	72
4.11.3	<b>TADD</b> 时钟数据加法指令 .....	73
4.11.4	<b>TSUB</b> 时钟数据减法指令 .....	74
4.11.5	<b>TRD</b> 时钟数据读出指令 .....	74
4.11.6	<b>TWR</b> 时钟数据写入指令 .....	75
<b>4.12</b>	<b>接点比较指令</b> .....	<b>75</b>
4.12.1	<b>LD♦</b> 母线连接的触点比较指令 .....	75
4.12.2	<b>AND♦</b> 串联连接的触点比较指令 .....	76
4.12.3	<b>OR♦</b> 并联连接的触点比较指令 .....	77

# 1 .指令目录

## 1.1 基本指令目录

指 令 助 记 符	名称	功能	FX2N 系列	页码
LD	取	a 触点的逻辑运算开始	●	
LDI	取反转	a 触点的逻辑运算开始	●	
LDP	取脉冲上升沿	检出上升沿逻辑运算开始	●	
LDF	取脉冲下降沿	检出下降沿逻辑运算开始	●	
AND	与	串联 a 触点	●	
ANI	与反转	串联 b 触点	●	
ANDP	与脉冲上升沿	上升沿检出串联连接	●	
ANDF	与脉冲下降沿	下降沿检出串联连接	●	
OR	或	并联 a 触点	●	
ORI	或反转	并联 b 触点	●	
ORP	或脉冲上升沿	上升沿检出并联连接	●	
ORF	或脉冲下降沿	下降沿检出并联连接	●	
ORB	回路块或	串联回路块的并联连接	●	
ANB	回路块与	并联回路块的串联连接	●	
OUT	输出	线圈驱动	●	
SET	置位	线圈驱动保持	●	
RST	复位	线圈驱动清除	●	
PLS	上升沿脉冲	上升沿检出输出	●	
PLF	下降沿脉冲	下降沿检出输出	●	
MC	主控	公共串联点的连接线圈指令	●	
MCR	主控复位	公共串联点的清除指令	●	
MPS	进栈	压入堆栈，运算储存	●	
MRD	读栈	读取堆栈，储存读出	●	
MPP	出栈	弹出堆栈，储存读出与复位	●	
INV	反转	运算结果取反	●	
NOP	空操作	无动作	●	
END	结束	顺序控制程序结束	●	



1.2 梯形图步进指令目录

指令助记符	名称	功能	FX2N系列	页码
STL	步进梯形图	步进梯形图开始	●	
RET	返回	步进梯形图结束	●	

1.3 应用指令目录

分类	FNC NO.	指令助记符	功能	32位支持	脉冲执行支持	FX2N系列	页码	分类	FNC NO.	指令助记符	功能	32位支持	脉冲执行支持	FX2N系列	页码
程序流程	0	CJ			●	●		数据处理	40	ZRST	批次复位		●	●	
	1	CALL	子程序调用		●	●			41	DECO	译码		●	●	
	2	SRET	子程序返回		●	●			42	ENCO	编码		●	●	
	3	IRET	中断返回			●			43	SUM	ON 位数	●	●	●	
	4	EI	终端许可			●			44	BON	ON 位数判定	●	●	●	
	5	DI	中断禁止			●			45	MEAN	平均值	●	●	●	
	6	FEND	主程序结束			●			46	ANS	信号报警器置位			●	
	7	WDT	监控定时器		●	●			47	ANR	信号报警器复位		●	●	
	8	FOR	循环范围开始			●			48	SQR	BIN 开方	●	●	●	
	9	NEXT	循环范围终了			●			49	FLT	BIN 整数-2 进制浮点数	●	●	●	
传送与比较	10	CMP	比较	●	●	●		高速处理	50	REF	输入输出刷新		●	●	
	11	ZCP	区域比较	●	●	●			51	REFF	滤波器调整		●	●	
	12	MOV	传送	●	●	●			52	MTR	矩阵输入			●	
	13	SMOV	移位传送		●	●			53	HSCS	比较置位(高速计数器)	●		●	
	14	CML	倒转传送	●	●	●			54	HSCR	比较复位(高速计数器)	●		●	
	15	BMOV	一并传送		●	●			55	HSZ	区间比较(高速计数器)	●		●	
	16	FMOV	多点传送	●	●	●			56	SPD	脉冲密度			●	
	17	XCH	交换	●	●	●			57	PLSY	脉冲输出	●		●	
	18	BCD	BCD 转换	●	●	●			58	PWM	脉冲调制			●	
	19	BIN	BIN 转换	●	●	●			59	PLSR	带加减速的脉冲输出	●		●	
四则运算	20	ADD	BIN 加法	●	●	●		方便指令	60	IST	初始化状态			●	
	21	SUB	BIN 减法	●	●	●			61	SER	数据查找	●	●	●	
	22	MUL	BIN 乘法	●	●	●			62	ABSD	凸轮控制(绝对方式)	●		●	
	23	DIV	BIN 除法	●	●	●			63	INCD	凸轮控制(增量方式)			●	
	24	INC	BIN 加一	●	●	●			64	TTMR	示教定时器			●	
	25	DEC	BIN 减一	●	●	●			65	STMR	特殊定时器			●	

循环移位	26	WAND	逻辑字与	●	●	●		外围设备I/O	66	ALT	交替输出		●	●	
	27	WOR	逻辑字或	●	●	●			67	RAMP	斜坡信号			●	
	28	WXOR	逻辑字异或	●	●	●			68	ROTC	旋转工作台控制			●	
	29	NEG	求补码	●	●	●			69	SORT	数据排列			●	
	30	ROR	循环右移	●	●	●			70	TKY	数字键输入	●		●	
	31	ROL	循环左移	●	●	●			71	HKY	16 键输入	●		●	
	32	RCR	带进位循环右移	●	●	●			72	DSW	数字式开关			●	
	33	RCL	带进位循环左移	●	●	●			73	SEGD	7 段译码		●	●	
	34	SFTR	位右移		●	●			74	SEGL	7 段码按时间分割显示			●	
	35	SFTL	位左移		●	●			75	ARWS	箭头开关			●	
外围设备SER	36	WSFR	字右移		●	●		外围设备	76	ASC	ASCII 码变换			●	
	37	WSFL	字左移		●	●			77	PR	ADCI 码打印输出			●	
	38	SFWR	移位写入		●	●			78	FROM	BFM 读出	●	●	●	
	39	SFRD	移位读出		●	●			79	TO	BFM 写入	●	●	●	
分类	FNC NO.	指令助记符	功能	32 位支持	脉冲执行支持	FX2N 系列	页码	分类	FNC NO.	指令助记符	功能	32 位支持	脉冲执行支持	FX2N 系列	页码
外围设备	80	RS	串行数据传输			●		外围设备	170	GRY	格雷码变换	●	●	●	
	81	PRUN	8 进制位传送	●	●	●			171	GBIN	格雷码逆变换	●	●	●	
	82	ASCI	HEX-ASCII 转换		●	●			176	RD3A	模拟块读出*		●	●	
	83	HEX	ASCII-HEX 转换		●	●			177	WR3A	模拟块写入*		●	●	
	84	CCD	校验码		●	●			180	EXTR	与三菱变频器通讯*	●	●	●	
	85	VRRD	电位器读出		●	●			224	LD=	(S1) = (S2)	●		●	
	86	VRSC	电位器刻度		●	●			225	LD>	(S1) > (S2)	●		●	
	87								226	LD<	(S1) < (S2)	●		●	
	88	PID	PIC 运算			●			228	LD<>	(S1) ≠ (S2)	●		●	
	88								229	LD<=	(S1) ≤ (S2)	●		●	
浮点数	110	ECMP	2 进制浮点数比较	●	●	●		接点比较	230	LD>=	(S1) ≥ (S2)	●		●	
	111	EZCP	2 进制浮点数区间比	●	●	●			232	AND=	(S1) = (S2)	●		●	
	118	EBCD	2 进制浮点数-10 进制浮点数转换	●	●	●			233	AND>	(S1) > (S2)	●		●	
	119	EBIN	10 进制浮点数-2 进制浮点数转换	●	●	●			234	AND<	(S1) < (S2)	●		●	
	120	EADD	2 进制浮点数加法	●	●	●			236	AND<>	(S1) ≠ (S2)	●		●	
	121	ESUB	2 进制浮点数减法	●	●	●			237	AND<=	(S1) ≤ (S2)	●		●	
	122	EMUL	2 进制浮点数乘法	●	●	●			238	AND>=	(S1) ≥ (S2)	●		●	
	123	EDIV	2 进制浮点数除法	●	●	●			240	OR=	(S1) = (S2)	●		●	
	127	ESQR	2 进制浮点数开方	●	●	●			241	OR>	(S1) > (S2)	●		●	
	129	INT	2 进制浮点数-BIN 整	●	●	●			242	OR<	(S1) < (S2)	●		●	
									244	OR<>	(S1) ≠ (S2)	●		●	
									245	OR<=	(S1) ≤ (S2)	●		●	

	130	SIN	浮点数 SIN 运算	●	●	●			246	OR>=	(S1) ≥ (S2)	●		●	
	131	COS	浮点数 COS 运算	●	●	●									
	132	TAN	浮点数 TAN 运算	●	●	●									
定位	147	SWAP	上下字节变换	●	●	●									
	155	ABS	ABS 现在值读出*	●		●									
	156	ZRN	原点回归#	●											
	157	PLSY	可变度的脉冲输出#	●											
	158	DRVI	相对定位#	●											
时钟运算	159	DRVA	绝对定位#	●											
	160	TCMP	时钟数据比较		●	●									
	161	TZCP	时钟数据区间比较		●	●									
	162	TADD	时钟数据加法		●	●									
	163	TSUB	时钟数据减法		●	●									
	166	TRD	时钟数据读出		●	●									
	167	TWR	时钟数据写入		●	●									
	169	HOUR	计时仪*	●		●									

## 2 .基本指令

### 2.1 LD 系列连接到母线的指令

#### 2.1.1 LD 取指令

功能说明：将触点连接到母线上，检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

LD S.

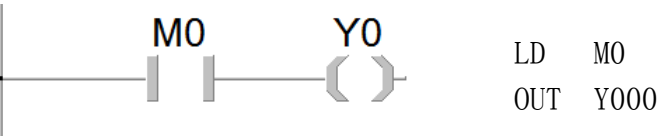
适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

当 ( M0 ) =0 时 , ( Y0 ) =0 ;

当 ( M0 ) =1 时 , ( Y0 ) =1 ;



#### 2.1.2 LDI 取反转指令

功能说明：将触点连接到母线上，检出触点不导通逻辑开始运算。

指令格式：( S-源操作数 )

LDI S.

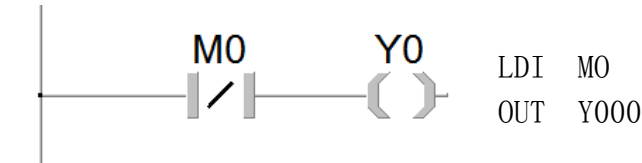
适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

当 ( M0 ) =0 时 , ( Y0 ) =1 ;

当 ( M0 ) =1 时 , ( Y0 ) =0 ;



2.1.3 **LDP** 取脉冲上升沿指令

功能说明：将触点连接到母线上，上升沿检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

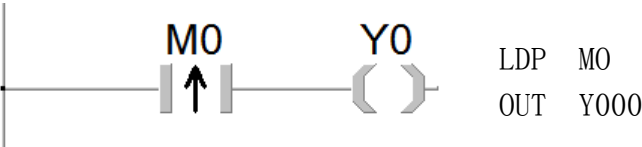
**LDP** S.

适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

梯形图运行第一个周期：( M0 ) =0 , ( Y0 ) =0 ;  
梯形图运行第二个周期：( M0 ) =1 , ( Y0 ) =1 ;  
梯形图运行第三个周期：( M0 ) =1 , ( Y0 ) =0 ;  
梯形图运行第四个周期：( M0 ) =0 , ( Y0 ) =0 ;  
梯形图运行第五个周期：( M0 ) =0 , ( Y0 ) =0 ;



2.1.4 **LDF** 取脉冲下降沿指令

功能说明：将触点连接到母线上，下降沿检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

**LDF** S.

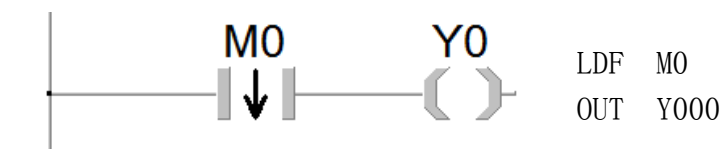
适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

梯形图运行第一个周期：( M0 ) =0 , ( Y0 ) =0 ;  
梯形图运行第二个周期：( M0 ) =1 , ( Y0 ) =0 ;  
梯形图运行第三个周期：( M0 ) =1 , ( Y0 ) =0 ;  
梯形图运行第四个周期：( M0 ) =0 , ( Y0 ) =1 ;

梯形图运行第五个周期 : ( M0 ) =0 , ( Y0 ) =0 ;



## 2.2 AND 系列串联连接的指令

### 2.2.1 AND 与指令

功能说明：串联连接一个触点，检出触点导通逻辑开始运算。

指令格式 : ( S-源操作数 )

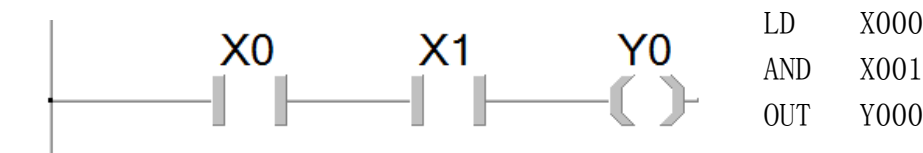
AND S.

适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

当 ( X0 ) =0 , ( X1 ) =0 时 , ( Y0 ) =0 ;  
当 ( X0 ) =0 , ( X1 ) =1 时 , ( Y0 ) =0 ;  
当 ( X0 ) =1 , ( X1 ) =0 时 , ( Y0 ) =0 ;  
当 ( X0 ) =1 , ( X1 ) =1 时 , ( Y0 ) =1 ;



### 2.2.2 ANI 与反转指令

功能说明：串联连接一个触点，检出触点不导通逻辑开始运算。

指令格式 : ( S-源操作数 )

ANDI S.

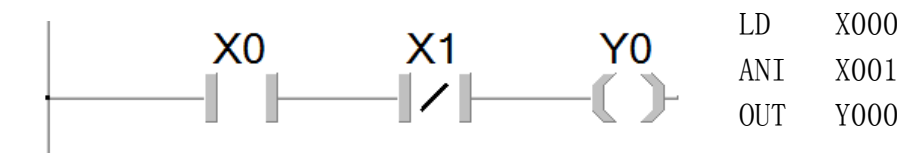
适用软元件：

	字软元件								位软元件			
--	------	--	--	--	--	--	--	--	------	--	--	--

S.						T	C			X	Y	M	S
----	--	--	--	--	--	---	---	--	--	---	---	---	---

举例：

当 ( X0 ) =0 , ( X1 ) =0 时 , ( Y0 ) =0 ;  
当 ( X0 ) =0 , ( X1 ) =1 时 , ( Y0 ) =0 ;  
当 ( X0 ) =1 , ( X1 ) =0 时 , ( Y0 ) =1 ;  
当 ( X0 ) =1 , ( X1 ) =1 时 , ( Y0 ) =0 ;



2.2.3 **ANDP** 与脉冲上升沿指令

功能说明：串联连接一个触点，上升沿检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

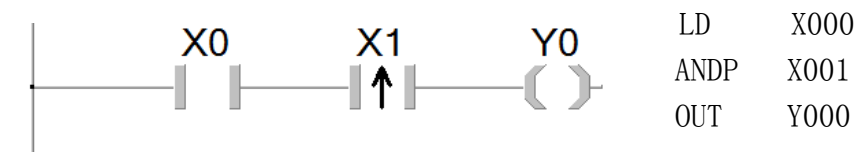
**ANDP** S.

适用软元件：

	字软元件									位软元件			
S.						T	C			X	Y	M	S

举例：

梯形图运行第一个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =0 ;  
梯形图运行第二个周期：( X0 ) =1, ( X1 ) =1 , ( Y0 ) =1 ;  
梯形图运行第三个周期：( X0 ) =1, ( X1 ) =1 , ( Y0 ) =0 ;  
梯形图运行第四个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =0 ;  
梯形图运行第五个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =0 ;  
梯形图运行第六个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ;  
梯形图运行第七个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =0 ;  
梯形图运行第八个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ;



2.2.4 **ANDF** 与脉冲下降沿指令

功能说明：串联连接一个触点，下降沿检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

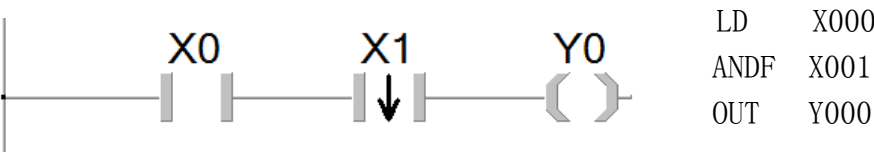
**ANDF** S.

适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

梯形图运行第一个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第二个周期：( X0 ) =1, ( X1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第三个周期：( X0 ) =1, ( X1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第四个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =1 ；  
梯形图运行第五个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第六个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第七个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第八个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ；



2.3 OR 系列并联连接的指令

2.3.1 **OR** 或指令

功能说明：并联连接一个触点，检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

**OR** S.

适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S



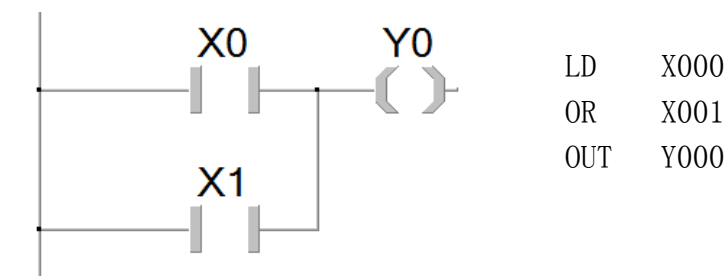
举例：

当 ( X0 ) =0 , ( X1 ) =0 时 , ( Y0 ) =0 ；

当 ( X0 ) =0 , ( X1 ) =1 时 , ( Y0 ) =1 ；

当 ( X0 ) =1 , ( X1 ) =0 时 , ( Y0 ) =1 ；

当 ( X0 ) =1 , ( X1 ) =1 时 , ( Y0 ) =1



2.3.2 **ORI** 或反转指令

功能说明：并联连接一个触点，检出触点不导通逻辑开始运算。

指令格式：( S-源操作数 )

**ORI** S.

适用软元件：

	字软元件								位软元件				
S.						T	C			X	Y	M	S

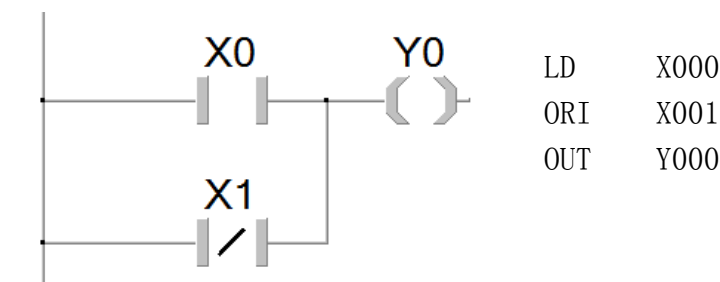
举例：

当 ( X0 ) =0 , ( X1 ) =0 时 , ( Y0 ) =1 ；

当 ( X0 ) =0 , ( X1 ) =1 时 , ( Y0 ) =0 ；

当 ( X0 ) =1 , ( X1 ) =0 时 , ( Y0 ) =1 ；

当 ( X0 ) =1 , ( X1 ) =1 时 , ( Y0 ) =1 ；



2.3.3 **ORP** 或脉冲上升沿指令

功能说明：并联连接一个触点，上升沿检出触点导通逻辑开始运算。

指令格式：(S-源操作数)

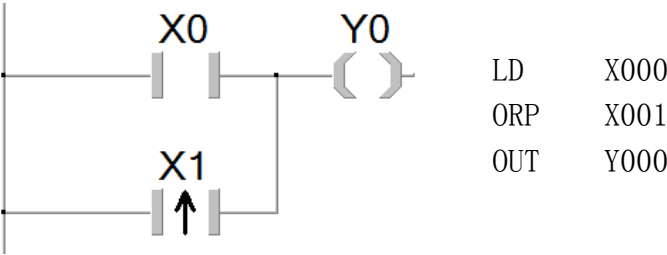
ORP S.

适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

梯形图运行第一个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第二个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =1 ；  
梯形图运行第三个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第四个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第五个周期：( X0 ) =1, ( X1 ) =0 , ( Y0 ) =1 ；



2.3.4 **ORF** 或脉冲下降沿指令

功能说明：并联连接一个触点，下降沿检出触点导通逻辑开始运算。

指令格式：( S-源操作数 )

ORF S.

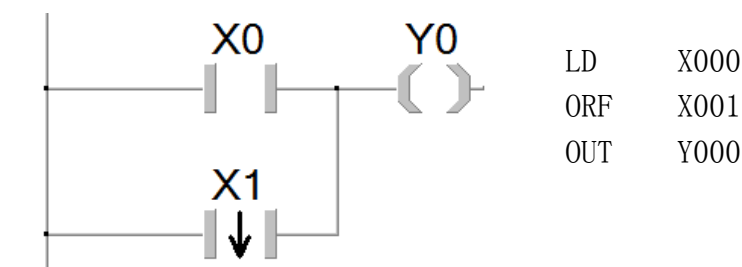
适用软元件：

	字软元件								位软元件			
S.						T	C		X	Y	M	S

举例：

梯形图运行第一个周期：( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第二个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第三个周期：( X0 ) =0, ( X1 ) =1 , ( Y0 ) =0 ；

梯形图运行第四个周期 : ( X0 ) =0, ( X1 ) =0 , ( Y0 ) =1 ;  
梯形图运行第五个周期 : ( X0 ) =0, ( X1 ) =0 , ( Y0 ) =0 ;  
梯形图运行第六个周期 : ( X0 ) =1, ( X1 ) =0 , ( Y0 ) =1 ;

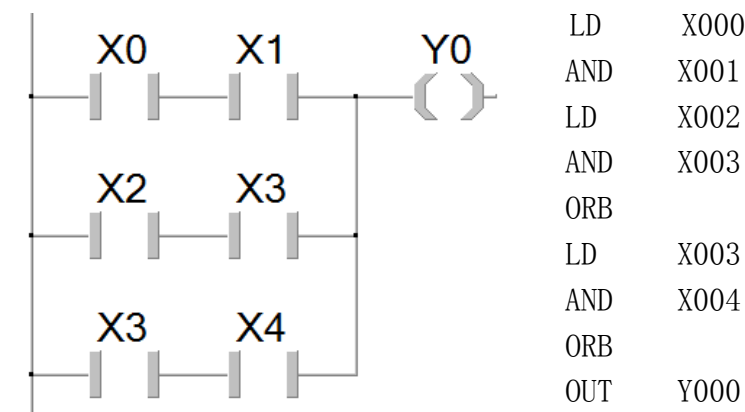


## 2.4 回路块指令

### 2.4.1 ORB 回路块或指令

功能说明：当串联电路（串联回路块）与前面的回路并联连接时，使用 ORB 回路块结束分支。

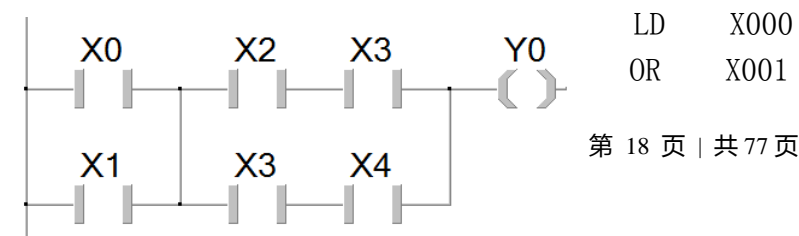
举例：



### 2.4.2 ANB 回路块与指令

功能说明：当并联电路（并联回路块）与前面的回路串联连接时，使用 ANB 回路块与指令与前面连接。

举例：



```
LD    X002
AND   X003
LD    X003
ORB
      ANB
      OUT  Y000
```

2.5 线圈控制指令

2.5.1 OUT 输出指令

功能说明：线圈驱动输出。

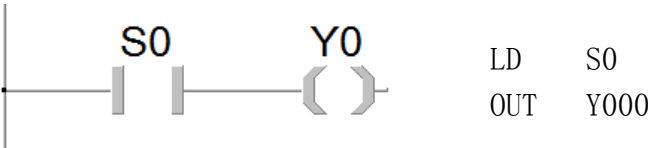
指令格式：( S-源操作数 )

```
OUT S.
```

适用软元件：

	字软元件										位软元件		
S.						T	C				Y	M	S

举例：  
当 ( S0 ) =0 时 , ( Y0 ) =0 ；  
当 ( S0 ) =1 时 , ( Y0 ) =1 ；



2.5.2 SET 置位指令

功能说明：线圈置位输出。

指令格式：( S-源操作数 )

```
SET S.
```

适用软元件：

	字软元件										位软元件		
S.											Y	M	S

举例：  
梯形图运行第一个周期：( M1 ) =0 , ( Y0 ) =0 ；  
梯形图运行第二个周期：( M1 ) =1 , ( Y0 ) =1 ；  
梯形图运行第三个周期：( M1 ) =0 , ( Y0 ) =1 ；



2.5.3 **RST** 复位指令

功能说明：线圈复位不输出。

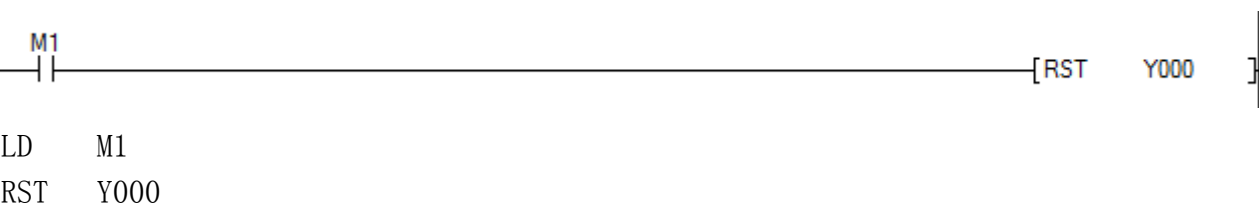
指令格式：( S-源操作数 )

**SET S.**

适用软元件：

	字软元件								位软元件		
S.									Y	M	S

举例：  
梯形图运行第一个周期：( M1 ) =0 , ( Y0 ) =1 ；  
梯形图运行第二个周期：( M1 ) =1 , ( Y0 ) =0 ；  
梯形图运行第三个周期：( M1 ) =0 , ( Y0 ) =0 ；



2.6 脉冲输出指令

2.6.1 **PLS** 上升沿脉冲输出指令

功能说明：仅在上升沿检出触点导通后的一个扫描周期，源操作数动作。

指令格式：( S-源操作数 )

**PLS S.**

适用软元件：

	字软元件										位软元件		
S.											Y	M	

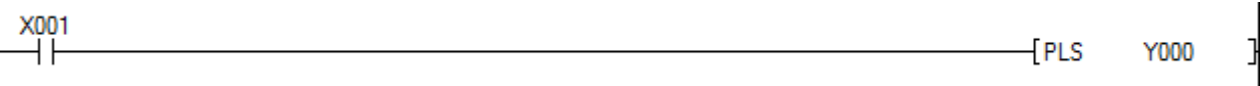
举例：

梯形图运行第一个周期： $(X1)=0,(Y0)=0$ ；

梯形图运行第二个周期： $(X1)=1,(Y0)=1$ ；

梯形图运行第三个周期： $(X1)=1,(Y0)=0$ ；

梯形图运行第四个周期： $(X1)=0,(Y0)=0$ ；



```
LD    X001
PLS   Y000
```

2.6.2 **PLF** 下降沿脉冲输出指令

功能说明：仅在下降沿检出触点导通后的一个扫描周期，源操作数动作。

指令格式： $(S-源操作数)$

```
PLF S.
```

适用软元件：

	字软元件										位软元件		
S.											Y	M	

举例：

梯形图运行第一个周期： $(X1)=1,(Y0)=0$ ；

梯形图运行第二个周期： $(X1)=0,(Y0)=1$ ；

梯形图运行第三个周期： $(X1)=0,(Y0)=0$ ；

梯形图运行第四个周期： $(X1)=1,(Y0)=0$ ；



```
LD    X001
PLF   Y000
```

2.7 主控指令

2.7.1 MC 主控指令

功能说明：触点导通，母线移动到 MC 触电之后。则执行源操作数 1 对应指针 N 一致的相距最近的，MC 与 MCR 指令之间的指令。与 MCR 指令联合使用。

指令格式：( S-源操作数 )

MC S1. S2.

( 注 1：0<=S1.<= 7 )；

( 注 2：可嵌套使用，非嵌套使用指针编号重复使用，多次使用时需注意 S2.不能重复 )；

( 注 3：MC 指令指令至相应 MCR 指令之间的程序，由执行变为不执行时，部分指令对应的软元件会复位。具体包括非累积定时器，OUT 指令驱动的 Y、M、S、T )；

适用软元件：

	指针
S1.	N

	字软元件								位软元件		
S2.									Y	M	

举例：( 见 MCR 主控复位指令举例 )

2.7.2 MCR 主控复位指令

功能说明：返回源操作数一致的 MC 指令的母线位置。

指令格式：( S-源操作数 )

MCR S.

( 注 1：0<=S.<= 7 )；

适用软元件：

	指针
S1.	N

举例：

梯形图运行第一个周期：( X0 ) =0 , ( M0 ) =0 , ( Y0 ) =0 , ( M5 ) =0 , ( S0 ) =0 , ( T0 ) =0 ；

梯形图运行第二个周期：( X0 ) =1 , ( M0 ) =0 , ( Y0 ) =0 , ( M5 ) =0 , ( S0 ) =0 , ( T0 ) 开始计时；

梯形图运行第三个周期 : ( X0 ) =1 , ( M0 ) =1 , ( Y0 ) =1 , ( M5 ) =1 , ( S0 ) =1 , ( T0 ) 开始计时 ;  
梯形图运行第四个周期 : ( X0 ) =0 , ( M0 ) =1 , ( Y0 ) =0 , ( M5 ) =0 , ( S0 ) =0 , ( T0 ) =0 ;



```
LD    X000
MC    NO      M100
LD    M0
OUT   Y000
LD    M0
OUT   M5
LD    M0
OUT   S0
LDI   T0
OUT   T0      K100
MCR   NO
```

2.8 堆栈指令

2.8.1 **MPS** 进栈指令

功能说明：并联回路块的串联连接，将指令运算中间结果压入堆栈。

举例：( 见 MPP 出栈指令举例 )

2.8.1 **MRD** 读栈指令

功能说明：并联回路块，将指令运算中间结果读出堆栈。

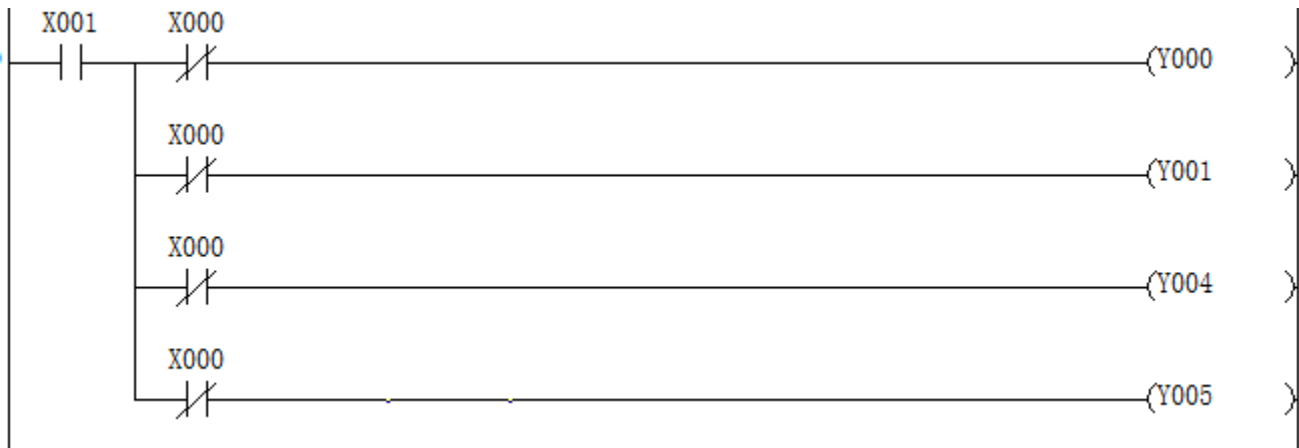


举例：( 见 MPP 出栈指令举例 )

2.8.1 **MPP** 出栈指令

功能说明：并联回路块的串联连接，将指令运算中间结果读出堆栈并将其复位。

举例：  
梯形图运行第一个周期：( X1 ) =0 , ( Y0 ) =0 , ( Y1 ) =0 , ( Y4 ) =0 , ( Y5 ) =0 ;  
梯形图运行第二个周期：( X1 ) =1 , ( Y0 ) =1 , ( Y1 ) =1 , ( Y4 ) =1 , ( Y5 ) =1 ;  
梯形图运行第三个周期：( X1 ) =0 , ( Y0 ) =0 , ( Y1 ) =0 , ( Y4 ) =0 , ( Y5 ) =0 ;



```
LD      X001
MPS
ANI     X000
OUT     Y000
MRD
ANI     X000
OUT     Y001
MRD
ANI     X000
OUT     Y004
MPP     X000
OUT     Y005
```

## 2.9 其他基本指令

### 2.9.1 **INV** 取反指令

功能说明：运算结果反转。

举例：

当 ( S0 ) =0 时 , ( Y0 ) =1 ;

当 ( S0 ) =1 时 , ( Y0 ) =0 ;



```
LD    S0
INV
OUT   Y000
```

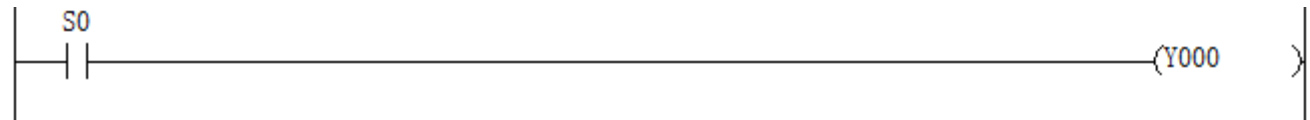
### 2.9.2 **NOP** 空操作指令

功能说明：没有任何操作。

举例：

当 ( S0 ) =0 时 , 无动作 ;

当 ( S0 ) =1 时 , 无动作 ;



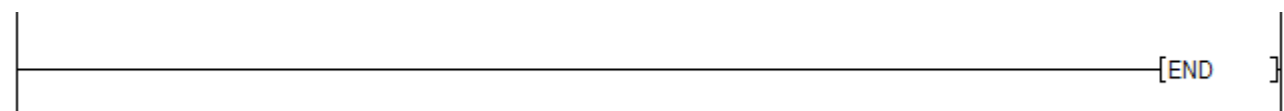
```
LD    S0
NOP
OUT   Y000
```

### 2.9.3 **END** 结束指令

功能说明：输入及输出数据处理，并返回梯形图 0 步。

举例：

返回梯形图 0 步；



```
END
```

# 3 .梯形图步进指令

## 3.1 步进梯形图指令

### 3.1.1 **STL** 步进梯形图指令

功能说明：利用内软元件状态（S 软元件），在顺序控制上面进行工序步进形控制的指令，与 RET 指令联合使用。

指令格式：( S-源操作数 )

**STL S.**

适用软元件：

	字软元件								位软元件			
S.												S

举例：

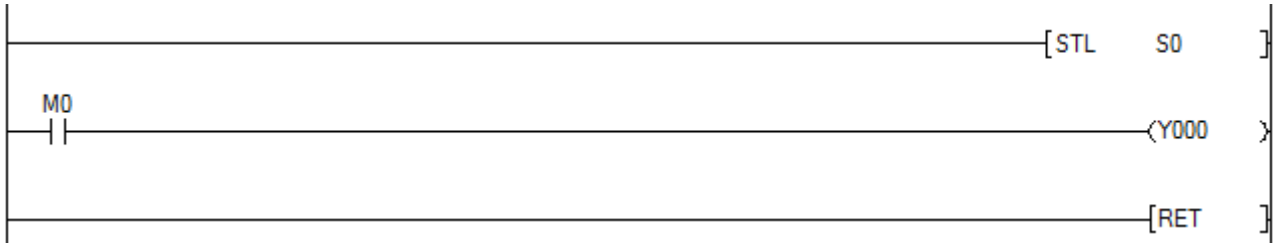
S0 为 1 时允许执行 STL S0 至 RET 中间的这一段程序：

当 ( S0 ) =0 , ( M0 ) =0 时 , ( Y0 ) =0 ；

当 ( S0 ) =0 , ( M0 ) =1 时 , ( Y0 ) =0 ；

当 ( S0 ) =1 , ( M0 ) =0 时 , ( Y0 ) =0 ；

当 ( S0 ) =1 , ( M0 ) =1 时 , ( Y0 ) =1 ；



STL S0  
LD M0  
OUT Y000  
RET

### 3.1.2 **RET** 返回指令

功能说明：返回表示状态流程（STL 指令）的结束。

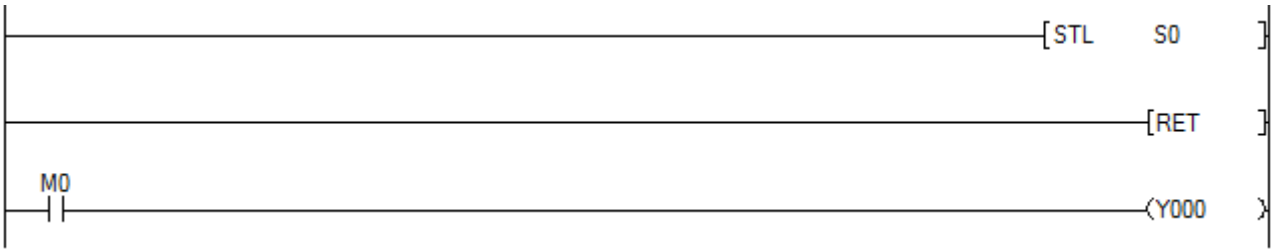
指令格式：

**RET**

举例：

RET 指令之下的程序不受 STL 指令控制：

当 ( S0 ) =0 , ( M0 ) =0 时 , ( Y0 ) =0 ;  
当 ( S0 ) =0 , ( M0 ) =1 时 , ( Y0 ) =1 ;  
当 ( S0 ) =1 , ( M0 ) =0 时 , ( Y0 ) =0 ;  
当 ( S0 ) =1 , ( M0 ) =1 时 , ( Y0 ) =1 ;



STL S0  
RET  
LD M0  
OUT Y000

# 4 .应用指令

## 4.1 程序流程指令

### 4.1.1 CJ 条件跳转指令

功能说明：跳转至目标指针程序处，开始执行。

- CJ - 16 位指令，源操作数为指针

指令格式：( S-源操作数 )

CJ S.

( 注 1：0<=S.<=127 );

( 注 2：指针 P63 固定指向梯形图结束 END );

适用软元件：

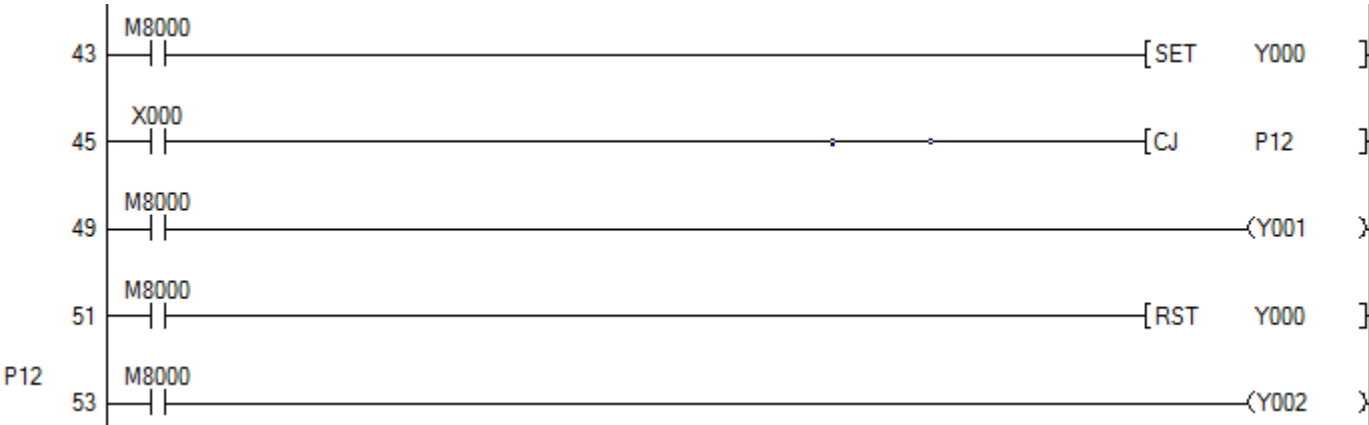
	指针
S.	P

举例：

梯形图运行第一个周期：( X0 ) =0 , ( Y0 ) =1 , ( Y1 ) =0 , ( Y2 ) =1 ；

梯形图运行第二个周期：( X0 ) =1 , ( Y0 ) =0 , ( Y1 ) =1 , ( Y2 ) =1 ；

梯形图运行第三个周期：( X0 ) =1 , ( Y0 ) =1 , ( Y1 ) =1 , ( Y2 ) =1 ；



```
LD      M8000
SET      Y000
LD      X000
CJ       P12
LD      M8000
```

RST        Y000  
P12  
LD        M8000  
OUT       Y002

4.1.2      **CALL**    子程序调用指令

功能说明：跳转至目标指针子程序处，开始执行。与 SRET 指令联合使用。

- CALL - 16 位指令，源操作数为指针

指令格式：( S-源操作数 )

**CALL S.**

( 注 1：0<=S.<=127 );

( 注 2：指针 P63 固定指向梯形图结束 END );

适用软元件：

	指针
S.	P

举例：( 见 SRET 子程序返回指令举例 )

4.1.3      **SRET**    子程序返回指令

功能说明：跳转出子程序，进入主程序，开始执行。与 CALL 指令联合使用。

- SRET - 单独指令

指令格式：

**SRET**

举例：

梯形图运行第一个周期：( X0 ) =0 , ( Y0 ) =0 , ( Y1 ) =1 , ( Y2 ) =0 ;

梯形图运行第二个周期：( X0 ) =1 , ( Y0 ) =1 , ( Y1 ) =1 , ( Y2 ) =1 ;

梯形图运行第三个周期：( X0 ) =0 , ( Y0 ) =0 , ( Y1 ) =1 , ( Y2 ) =1 ;



```
LD      M8000
RST     Y000
LD      X000
SALL    P13
LD      M8000
OUT     Y001
FEND
P13
LD      M8000
OUT     Y001
LD      M8000
SET     Y000
```

4.1.4 **FEND** 主程序结束指令

功能说明：主程序结束，执行输入输出处理，监视定时器刷新，返回第 0 步程序。

- **FEND** - 单独指令

指令格式：

```
FEND
```

举例：

主程序结束，做输入输出等处理，返回第 0 步程序；

```
FEND
```

4.1.5 **FOR** 循环范围开始指令

功能说明：FOR 指令与 NEXT 指令之间的程序，按 S.指定次数重复执行，与 NEXT 指令联合使用。

- FOR - 16 位指令，源操作数为 16 位

指令格式：( S-源操作数 )

FOR S.

( 注 1：1<=S.<=32767,当 S.<1 时按循环 1 次处理 );

( 注 2：FOR 指令最多嵌套 5 层 );

适用软元件：

	字软元件									位软元件			
S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z	X	Y	M	S

举例：( 见 NEXT 循环范围结束指令举例 )

4.1.6 **NEXT** 循环范围结束指令

功能说明：FOR 指令循环范围结束，判断决定跳转至 FOR 指令继续循环或结束循环继续执行程序。

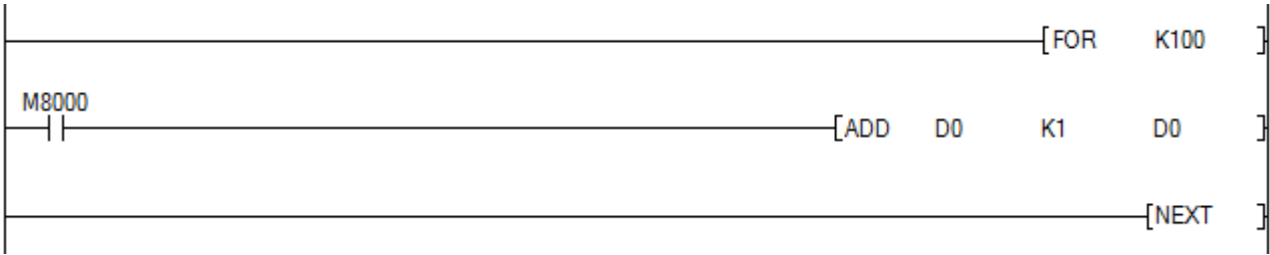
- NEXT - 单独指令

指令格式：( S-源操作数 )

NEXT

举例：

操作前：( D0 ) =0；



```
FOR      K100
LD       M8000
ADD      D0      K1      D0
NEXT
```

操作后：( D0 ) =100；



## 4.2 传送与比较指令

### 4.2.1 **CMP** 比较指令

功能说明：将 S1.与 S2.的内容进行比较，并将比较结果送到 D.及其后两个软元件上。

- **CMP** - 16 位指令，源操作数与目标操作数为 16 位
- **DCMP** - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

**CMP S1. S2. D.**

**DCMP S1. S2. D.**

( 注 1：所有源操作数都被看成二进制值处理 )；

( 注 2：D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用 )；

适用软元件：

		字软元件									位软元件		
S1. S2.		K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z			
D.											Y	M	S

举例：

**CMP K100 C10 M2**

操作后：当 K100> ( C10 ) 时，( M2 ) =1，( M3 ) =0，( M4 ) =0；

当 K100= ( C10 ) 时，( M2 ) =0，( M3 ) =1，( M4 ) =0；

当 K100< ( C10 ) 时，( M2 ) =0，( M3 ) =0，( M4 ) =1；

### 4.2.2 **ZCP** 区域比较指令

功能说明：将 S.与 S1.和 S2.的内容进行比较，并将比较结果送到目的操作数及其后两个软元件上。

- **ZCP** - 16 位指令，源操作数与目标操作数为 16 位
- **DZCP** - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

**CMP S1. S2. S. D.**

**DCMP S1. S2. S. D.**

( 注 1：所有源操作数都被看成二进制值处理 )；

( 注 2：D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用 )；

( 注 3：源操作数 1 不得大于源操作数 2，当不满足该条件时，默认两个比较值都为源操作数 1 )；

适用软元件：

字软元件												位软元件			
S1, S2, S,	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z						
D,												Y	M	S	

举例：

ZCP K100 K120 C10 M2

操作后：当 K100>( C10 ) 时 , ( M2 ) =1 , ( M3 ) =0 , ( M4 ) =0 ；

当 K100<= ( C10 ) <=K120 时 , ( M2 ) =0 , ( M3 ) =1 , ( M4 ) =0 ；

当 K120<( C10 ) 时 , ( M2 ) =0 , ( M3 ) =0 , ( M4 ) =1 ；

4.2.3 **MOV** 传送指令

功能说明：将 S.传送到 S.。

- MOV - 16 位指令，源操作数与目标操作数为 16 位
- DMOV - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

MOV S. D.

DMOV S. D.

( 注：源操作数为常数时，将被自动转换成 BIN 码 )；

适用软元件：

字软元件												位软元件			
S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z						
D.		KnX	KnY	KnM	KnS	T	C	D	V, Z						

举例：

操作前：( D0 ) =0x1234, ( D11 ) =0;

MOV D0 D11

操作后：( D0 ) =0x1234, ( D11 ) =0x1234;

操作前：( D1,D0 ) =0x12345678, ( D12,D11 ) =0;

DMOV D0 D11

操作后：( D1,D0 ) =0x12345678, ( D12,D11 ) =0x12345678;

4.2.4 **SMOV** 位移动指令

功能说明：将 S.的 BCD 码转换值从其第 m1 位起的低 m2 位部分向 D. ( BCD 码转换值 ) 开始传送，然后将其转换回 BIN 码。当驱

动 M8168 执行时不进行 BCD 码转换，直接传送。

- SMOV - 16 位指令，操作数、源操作数与目标操作数为 16 位。

指令格式：( S-源操作数，D-目标操作数，m、n-操作数 )

```
SMOV S. m1 m2 D. n
```

( 注 1：m1、m2、n=1~4，并且 n>=m2，S.<9999，D.<9999 )；

( 注 2：当驱动 M8168 时，S.与 D.中的数据不转换为 BCD 码，按十六进制的格式移动 )；

适用软元件：

				字软元件						位软元件						
S.					KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.						KnY	KnM	KnS	T	C	D	V, Z				
m1	m2	n	K, H													

举例：

操作前：( D10 ) =1234，( D2 ) =5678，( M8168 ) =0；

```
SMOV D10 K4 K2 D2 K3
```

操作后：( D10 ) =1234，( D2 ) =5128，( M8168 ) =0；

操作前：( D10 ) =0x6789，( D2 ) =0x1234，( M8168 ) =1；

```
SMOV D10 K4 K2 D2 K2
```

操作后：( D10 ) =0x6789，( D2 ) =0x1267，( M8168 ) =1；

### 4.2.5 CML 反相传送指令

功能说明：将 S.的各位反相 ( 0 变 1，1 变 0 ) 后，传送到 D.。

- CML - 16 位指令，源操作数与目标操作数为 16 位
- DCML - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

```
CML S. D.
```

```
DCML S. D.
```

( 注：源操作数为常数时，将被自动转换成 BIN 码 )；

适用软元件：

	字软元件									位软元件			
S.	K, H	KnX	KnY	KnS	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z				

举例：

操作前：( D0 ) =0x6969 , ( D10 ) =0 ；

**CML D0 D10**

操作后：( D0 ) =0x6969 , ( D10 ) =0x9696 ；

操作前：( D1 , D0 ) =0xA5A59696 , ( D11 , D10 ) =0 ；

**DCML D0 D10**

操作后：( D1 , D0 ) =0xA5A59696 , ( D11 , D10 ) =0x5A5A6969 ；

### 4.2.6 **BMOV** 成批传送指令

功能说明：将以 S.指定的软元件为开头的 n 个软元件，向以 D.指定的软元件开头的 n 个软元件成批传送。扩展指令，当驱动 M8024 时执行，反向传输。

- **BMOV** - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

**BMOV S. D. n**

( 注 1：n <= 512，在超过软元件编号范围时，在可能的范围内传送 )；

( 注 2：当 n<0 时会置位运算错误标志位 M8067 )；

( 注 3：当驱动 M8024 时，执行指令传送方向反转 )；

( 注 4：带有位指定的软元件（如：K1M0）时，源操作数和目标操作数要采用相同的位数 )；

适用软元件：

	字软元件								位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D				
D.			KnY	KnM	KnS	T	C	D				
N	K, H							D				

举例：

操作前：( D7 ) =0x1234 , ( D8 ) =0x5678 , ( D20 ) =0 , ( D21 ) =0, ( M8024 ) =0 ；

**BMOV D7 D20 K2**

操作后：( D7 ) =0x1234 , ( D8 ) =0x5678 , ( D20 ) =0x1234 , ( D21 ) =0x5678, ( M8024 ) =0 ；

操作前: ( D5 ) =0 , ( D6 ) =0 , ( D10 ) =0x6789 , ( D11 ) =0xABCD, ( M8024 ) =1 ；

**BMOV D5 D10 K2**

操作后：( D5 ) =0x6789 , ( D6 ) =0xABCD , ( D10 ) =0x6789 , ( D11 ) =0xABCD, ( M8024 ) =1 ；

4.2.7 **FMOV** 多点传送指令

功能说明：将 S.的内容向以 D.为开头的 n 个软元件进行重复传送。操作后从 D.开始的 n 个软元件的内容都一样。

- FMOV - 16 位指令，操作数、源操作数与目标操作数为 16 位
- DFMOV - 32 位指令，操作数、源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数，n-传送数据个数 )

FMOV S. D. n

DFMOV S. D. n

- ( 注 1：n <= 512 );
- ( 注 2：当 n<0 时会置位运算错误标志位 M8067 );
- ( 注 3：超过目标软元件号的范围，向可能的范围传送 );

适用软元件：

	字软元件									位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z				
n	K, H												

举例：

操作前：( D20 ) =0 , ( D21 ) =0 , ( D22 ) =0 ；

FMOV K12 D20 K3

操作后：( D20 ) =12 , ( D21 ) =12 , ( D22 ) =12 ；

操作前：( D11 , D10 ) =0 , ( D13 , D12 ) =0 , ( D15 , D14 ) =0 ；

DFMOV K12 D10 K3

操作后：( D11 , D10 ) =12 , ( D13 , D12 ) =12 , ( D15 , D14 ) =12 ；

4.2.8 **XCH** 交换指令

功能说明：将 S.与 D.数据交换。扩展指令，当驱动 M8160 且 S.与 D.是统一软元件时，该操作数低八位与高八位交换。

- XCH - 16 位指令，目标操作数为 16 位
- DXCH - 32 位指令，目标操作数为 32 位

指令格式：( D-目标操作数 )

XCH D1. D2.

DXCH D1. D2.

- ( 注 1：对于 C200 及其他高级计数器不能用 XCH 操作，视为出错 );
- ( 注 2：当驱动 M8160 时 ,并且 S.与 D.是同一软元件时 ,低 8 位与高 8 位可进行交换。非同一软元件时会置位运算错误标志位 M8067 ( 扩展功能与 SWAP 指令功能相同 ) );

(注3：当驱动 M8160 时，S 与 D 的软元件编号不同时，出错标志 M8067 变为 ON 状态，该指令无法执行)；

适用软元件：

		字软元件								位软元件			
D1.	D2.			KnY	KnM	KnS	T	C	D	V, Z			

举例：

操作前：( D10 ) =20 , ( D12 ) =30 , ( M8160 ) =0 ；  
**XCH D10 D12**  
操作后：( D10 ) =30 , ( D12 ) =20 , ( M8160 ) =0 ；

操作前：( D11 , D10 ) =0x12345678 , ( D13 , D12 ) =0xABCDEF21 , ( M8160 ) =0 ；  
**XCH D10 D12**  
操作后：( D11 , D10 ) =0xABCDEF21 , ( D13 , D12 ) =0x12345678 , ( M8160 ) =0 ；

操作前：( D10 ) =0x695A , ( M8160 ) =1 ；  
**XCH D10 D10**  
操作后：( D10 ) =0x5A69 , ( M8160 ) =1 ；

操作前：( D11,D10 ) =0x1234695A , ( M8160 ) =1 ；  
**DXCH D10 D10**  
操作后：( D11,D10 ) =0x34125A69 , ( M8160 ) =1 ；

4.2.9 **BCD** BCD 交换指令

功能说明：将 S 中的二进制数转换成 BCD 码送到 D 中。

- BCD - 16 位指令，源操作数与目标操作数为 16 位
- DBCD - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

**BCD S. D.**  
**DBCD S. D.**

(注：当 BIN 指令源操作数超出 0~9999 范围，或当 DBIN 指令源操作数超出 0~99999999 范围，会置位运算错误标志位 M8067)；

适用软元件：

		字软元件								位软元件			
S.			KnX	KnY	KnM	KnS	T	C	D	V, Z			
D.			KnY	KnM	KnS	T	C	D	V, Z				

举例：

操作前：( D10 ) =18 , ( D12 ) =0 ；

BCD D10 D12

操作后：( D10 ) =18 , ( D12 ) =0x18 ；

操作前：( D11 , D10 ) =43981 , ( D13 , D12 ) =0 ；

DBCD D10 D12

操作后：( D11 , D10 ) =43981 , ( D13 , D12 ) =0x43981 ；

4.2.10 **BIN** BIN 交换指令

功能说明：将 S 中的 BCD 码转换成二进制数送到 D 中。

- BIN - 16 位指令，源操作数与目标操作数为 16 位
- DBIN - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

BIN S. D.

DBIN S. D.

( 注 1：当 BIN 指令目标操作数超出 0~9999 范围，或当 DBIN 指令目标操作数超出 0~99999999 范围，会置位运算错误标志位 M8067 与预算错误锁存标志位 M8068 )；

( 注 2：当源操作数的数据不是 BCD 码时，会发生 M8067 ( 运算错误 )，M8068 ( 运算错误锁存 ) 将不工作 )；

适用软元件：

	字软元件									位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z				

举例：

操作前：( D0 ) =0x1234 , ( D12 ) =0 ；

BIN D10 D12

操作后：( D0 ) =0x1234 , ( D12 ) =1234 ；

操作前：( D1 , D0 ) =0x12345678 , ( D13 , D12 ) =0 ；

DBIN D10 D12

操作后：( D1 , D0 ) =0x12345678 , ( D13 , D12 ) =12345678 ；

4.3 四则逻辑运算指令

4.3.1 ADD BIN 加法运算指令

功能说明：将 S1.与 S2.数据进行二进制加法后传递到 D.中。

- ADD - 16 位指令，源操作数与目标操作数为 16 位
- DADD - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

ADD S1. S2. D.

DADD S1. S2. D.

- ( 注 1：运算结果为 0 时，0 标志 M8020 会动作 )；
- ( 注 2：运算结果超过 32767 ( 16 位运算 ) 或 2147483647 ( 32 位运算 ) 时，进位标志 M8022 会动作 )；
- ( 注 3：运算结果不满-32768 ( 16 位运算 ) 或-2147483648 ( 32 位运算 ) 时，借位标志 M8021 会动作 )；

适用软元件：

		字软元件									位软元件			
S1. S2.		K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.				KnY	KnM	KnS	T	C	D	V, Z				

举例：

操作前：( D0 ) =2 , ( D10 ) =0；

ADD K2 D0 D10

操作后：( D0 ) =2 , ( D10 ) =4；

操作前：( D1 , D0 ) =99999 , ( D11 , D10 ) =0；

DADD K99999 D0 D10

操作后：( D1 , D0 ) =99999 , ( D11 , D10 ) =199998；

4.3.2 SUB BIN 减法运算指令

功能说明：将 S1.的数据减去 S2.的数据后传递到 D.中。

- SUB - 16 位指令，源操作数与目标操作数为 16 位
- DSUB - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

SUB S1. S2. D.

DSUB S1. S2. D.

- ( 注 1：运算结果为 0 时，0 标志 M8020 会动作 )；



(注 2：运算结果超过 32767 ( 16 位运算 ) 或 2147483647 ( 32 位运算 ) 时，进位标志 M8022 会动作)；

(注 3：运算结果不满-32768 ( 16 位运算 ) 或-2147483648 ( 32 位运算 ) 时，借位标志 M8021 会动作)；

适用软元件：

		字软元件									位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.				KnY	KnM	KnS	T	C	D	V, Z				

举例：

操作前：( D0 ) =555 , ( D10 ) =55 , ( D20 ) =0 ；

SUB D0 D10 D20

操作后：( D0 ) =555 , ( D10 ) =55 , ( D20 ) =500

操作前：( D1 , D0 ) =777777 , ( D11 , D10 ) =34567 , ( D21 , D20 ) =0 ；

DSUB D0 D10 D20

操作后：( D1 , D0 ) =777777 , ( D11 , D10 ) =34567 , ( D21 , D20 ) =743210；

4.3.3 **MUL** BIN 乘法运算指令

功能说明：将 S1和 S2.的数据相乘后传递到 D.中。

- MUL - 16 位指令，源操作数为 16 位，目标操作数为 32 位
- DMUL - 32 位指令，源操作数为 32 位，目标操作数为 64 位

指令格式：( S-源操作数，D-目标操作数 )

MUL S1. S2. D.

DMUL S1. S2. D.

(注 1：运算结果为 0 时，0 标志 M8020 会动作)；

(注 2：D.是位元件时，可以进行 K1~K8 的位指定，指定 K4 时，智能求得乘积运算的低 16 位)；

(注 3：32 位指令且目标操作数使用位软元件时，只能得到低 32 位地址)；

(注 4：不能指定 Z 作为 D.)；

适用软元件：

		字软元件									位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.				KnY	KnM	KnS	T	C	D	V, Z 限于 16 位计算可指定				

举例：

操作前：( D0 ) =11111 , ( D10 ) =11112 , ( D21 , D20 ) =0 ；

MUL D0 D10 D20

操作后：( D0 ) =11111 , ( D10 ) =11112 , ( D21 , D20 ) =123465432 ；

操作前：( D1 , D0 ) = 777777 , ( D11 , D10 ) = 222222 , ( D23 , D22 , D21 , D20 ) = 0 ;

**DMUL D0 D10 D20**

操作后：( D1 , D0 ) = 777777 , ( D11 , D10 ) = 222222 , ( D23 , D22 , D21 , D20 ) = 172839160494 ;

#### 4.3.4 **DIV** BIN 除法运算指令

功能说明：将 S1. 中内容整除 S2. 中的内容，并将商和余数分别放入 D. 和 D. 的下一个编号的软元件中。

- **DIV** - 16 位指令，源操作数与目标操作数为 16 位
- **DDIV** - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

**DIV S1. S2. D.**

**DDIV S1. S2. D.**

( 注 1：除数为 0 时置位运算错误标志位 M8067，不能执行指令 )；

( 注 2：将位软元件指定为 D. 时，无法得到余数 )；

( 注 3：不能指定 Z 作为 D. )；

适用软元件：

		字软元件								位软元件			
S1. S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z 限于 16 位计算可指定				

举例：

操作前：( D0 ) = 25 , ( D10 ) = 10 , ( D20 ) = 0 , ( D21 ) = 0 ;

**DIV D0 D10 D20**

操作后：( D0 ) = 25 , ( D10 ) = 10 , ( D20 ) = 2 , ( D21 ) = 5 ;

操作前：( D1 , D0 ) = 777777 , ( D11 , D10 ) = 222222 , ( D21 , D20 ) = 0 , ( D23 , D22 ) = 0 ;

**DDIV D0 D10 D20**

操作后：( D1 , D0 ) = 777777 , ( D11 , D10 ) = 222222 , ( D21 , D20 ) = 3 , ( D23 , D22 ) = 111111 ;

#### 4.3.5 **INC** BIN 增加指令

功能说明：D. 的内容加一。

- **INC** - 16 位指令，目标操作数为 16 位
- **DINC** - 32 位指令，目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

INC D.

DINC D.

适用软元件：

	字软元件								位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z			

举例：

操作前：( D0 ) =300；

INC D0

操作后：( D0 ) =301；

操作前：( D1 , D0 ) =800000；

DINC D0

操作后：( D1 , D0 ) =800001；

4.3.6 **DEC** BIN 减少指令

功能说明：D.的内容自减一。

- DEC - 16 位指令，目标操作数为 16 位
- DDEC - 32 位指令，目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEC D.

DDEC D.

适用软元件：

	字软元件								位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z			

举例：

操作前：( D0 ) =300；

INC D0

操作后：( D0 ) =299；

操作前：( D1 , D0 ) =800000；

DINC D0

操作后：( D1 , D0 ) =799999；

4.3.7 AND 逻辑与指令

功能说明：将 S1.的数据与 S2.的数据按位与，并将结果传递到 D.中。

- WAND - 16 位指令，源操作数与目标操作数为 16 位
- DAND - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

WAND S1. S2. D.  
DAND S1. S2. D.

适用软元件：

		字软元件										位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z					
D.				KnY	KnM	KnS	T	C	D	V, Z					

举例：

操作前：( D0 ) =0xF , ( D10 ) =0x5 , ( D20 ) =0 ；

WAND D0 D10 D20

操作后：( D0 ) =0xF , ( D10 ) =0x5 , ( D20 ) =0x5 ；

操作前：( D1 , D0 ) =0xFFFFF000 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0 ；

DAND D0 D10 D20

操作后：( D1 , D0 ) =0xFFFFF000 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0xF000 ；

4.3.8 OR 逻辑或指令

功能说明：将 S1.的数据与 S2.的数据按位或，并将结果传递到 D.中。

- WAND - 16 位指令，源操作数与目标操作数为 16 位
- DAND - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

WOR S1. S2. D.  
DOR S1. S2. D.

适用软元件：

		字软元件										位软元件			
--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--

S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z					

举例：

操作前：( D0 ) =0xF , ( D10 ) =0x5 , ( D20 ) =0 ；

**WOR D0 D10 D20**

操作后：( D0 ) =0xF , ( D10 ) =0x5 , ( D20 ) =0xF ；

操作前：( D1 , D0 ) =0x1234FF00 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0 ；

**DOR D0 D10 D20**

操作后：( D1 , D0 ) =0x1234FF00 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0x1234FFFF ；

### 4.3.9 **XOR** 逻辑异或指令

功能说明：将 S1.的数据与 S2.的数据按位异或,并将结果传递到 D.中。

- **WXOR** - 16 位指令，源操作数与目标操作数为 16 位
- **DXOR** - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

**WXOR S1. S2. D.**

**DXOR S1. S2. D.**

适用软元件：

		字软元件									位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D	V, Z					

举例：

操作前：( D0 ) =15 , ( D10 ) =5 , ( D20 ) =0 ；

**WXOR D0 D10 D20**

操作后：( D0 ) =15 , ( D10 ) =5 , ( D20 ) =10 ；

操作前：( D1 , D0 ) =0x1234FF00 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0 ；

**DXOR D0 D10 D20**

操作后：( D1 , D0 ) =0x1234FF00 , ( D11 , D10 ) =0xFFFF , ( D21 , D20 ) =0x123400FF ；

### 4.3.10 **NEG** 求补指令

功能说明：D.的内容中各位先取反然后在加一，将其结果在存入原先的软元件中。

- **NEG** - 16 位指令，目标操作数为 16 位
- **DNEG** - 32 位指令，目标操作数为 32 位

指令格式：( D-目标操作数 )

**NEG D.**

**DNEG D.**

适用软元件：

	字软元件								位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z			

举例：

操作前：( D0 ) =111；

**NEG D0**

操作后：( D0 ) =-111；

操作前：( D1, D0 ) =800000；

**DNEG D0**

操作后：( D1, D0 ) =-800000；

## 4.4 循环移位

### 4.4.1 **ROR** 右回转指令

功能说明：D.的内容右移 n 位，超出寄存器范围的 n 位补入低位部分。

- **ROR** - 16 位指令，操作数与目标操作数为 16 位
- **DROR** - 32 位指令，操作数与目标操作数为 32 位

指令格式：( D-目标操作数, n-操作数 )

**ROR D. n**

**DROR D. n**

( 注 1：16 位指令时，n<=16，32 位指令时，n<=32 )；

( 注 2：补入的最后一位放入进位标志位 M8022 )；

( 注 3：在位指定软元件情况下，只有 K4 ( 16 位指令 ) 和 K8 ( 32 位指令 ) 是有效的 )；

适用软元件：

	字软元件								位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z			
N	K, H											

举例：

操作前：( D0 ) = 0xFF08；

**ROR D0 K4**

操作后：( D0 ) = 0x8FF0；

操作前：( D1, D0 ) = 0x12345678；

**DROR D0 K4**

操作后：( D1, D0 ) = 0x81234567；

#### 4.4.2 **ROL** 左回转指令

功能说明：D.的内容左移 n 位，超出寄存器范围的 n 位补入高位部分。

- **ROL** - 16 位指令，操作数与目标操作数为 16 位
- **DROL** - 32 位指令，操作数与目标操作数为 32 位

指令格式：( D-目标操作数, n-操作数 )

**ROL D. n**

**DROL D. n**

( 注 1：16 位指令时，n<=16，32 位指令时，n<=32 )；

( 注 2：补入的最后一位放入进位标志位 M8022 )；

( 注 3：在位指定软元件情况下，只有 K4 ( 16 位指令 ) 和 K8 ( 32 位指令 ) 是有效的 )；

适用软元件：

	字软元件								位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z			
n	K, H											

举例：

操作前：( D0 ) = 0x1234；

**ROL D0 K4**

操作后：( D0 ) = 0x2341；

操作前：( D1, D0 ) = 0x12345678；

**DROL D0 K20**

操作后：( D1, D0 ) = 0x67812345；

#### 4.4.3 **RCR** 带进位右回转指令

功能说明：D.中的数据连带进位标志位 M8022，一起向右移动 n 个二进制位，移出的低位连带标志位 M8022 的数据循环进入 D.的

高位，最后移出的位值移入标志位 M8022。

- RCR - 16 位指令，操作数与目标操作数为 16 位
- DRCR - 32 位指令，操作数与目标操作数为 32 位

指令格式：( D-目标操作数，n-操作数 )

RCR D. n

DRCR D. n

- ( 注 1：16 位指令时，n<=16，32 位指令时，n<=32 )；
- ( 注 2：进位标志位 M8022，会参与到指令操作中)；
- ( 注 3：在位指定软元件情况下，只有 K4 ( 16 位指令 ) 和 K8 ( 32 位指令 ) 是有效的)；

适用软元件：

	字软元件									位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z				
N	K, H												

举例：

操作前：( D0 ) =0xF0C5 , ( M8022 ) =1；

RCR D0 K4

操作后：( D0 ) =0xBF0C , ( M8022 ) =0；

操作前：( D1 , D0 ) =0xF0000C5 , ( M8022 ) =1；

DRCR D0 K4

操作后：( D1 , D0 ) =0xBF0000C , ( M8022 ) =0；

#### 4.4.4 RCL 带进位左回转指令

功能说明：D.中的数据连带进位标志位 M8022，一起向左移动 n 个二进制位，移出的高位连带标志位 M8022 的数据循环进入 D.的低位，最后移出的位值移入标志位 M8022。

- RCL - 16 位指令，操作数与目标操作数为 16 位
- DRCL - 32 位指令，操作数与目标操作数为 32 位

指令格式：( D-目标操作数，n-操作数 )

RCL D. n

DRCL D. n

- ( 注 1：16 位指令时，n<=16，32 位指令时，n<=32 )；
- ( 注 2：进位标志位 M8022，会参与到指令操作中)；
- ( 注 3：在位指定软元件情况下，只有 K4 ( 16 位指令 ) 和 K8 ( 32 位指令 ) 是有效的)；



适用软元件：

	字软元件									位软元件			
D.			KnY	KnM	KnS	T	C	D	V, Z				
N	K, H												

举例：

操作前：( D0 ) =0xB30F , ( M8022 ) =0；

RCL D0 K4

操作后：( D0 ) =0x30F5 , ( M8022 ) =1；

操作前：( D1 , D0 ) =0xB300000F , ( M8022 ) =0；

DRCL D0 K4

操作后：( D1 , D0 ) =0x300000F5 , ( M8022 ) =1；

4.4.5 SFTR 位右移指令

功能说明：D.开始的 n1 个位软元件，右移 n2 位。高位补 n2 位的 S.。

- SFTR - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

SFTR S. D. n1 n2

( 注：n2<=n1<=1024 )；

适用软元件：

	字软元件									位软元件			
S.										X	Y	M	S
D.											Y	M	S
n1 n2	K, H												

举例：

操作前：( X1 ) =0 , ( X0 ) =1 , ( M3 ) =0 , ( M2 ) =1 , ( M1 ) =1 , ( M0 ) =1；

SFTR X0 M0 K4 k2

操作后：( X1 ) =0 , ( X0 ) =1 , ( M3 ) =0 , ( M2 ) =1 , ( M1 ) =0 , ( M0 ) =1；

4.4.6 SFTL 位左移指令

功能说明：D.开始的 n1 个位软元件，左移 n2 位。低位补 n2 位的 S.。

- SFTL - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

SFTL S. D. n1 n2

( 注：n2<=n1<=1024 );

适用软元件：

	字软元件									位软元件			
S.										X	Y	M	S
D.											Y	M	S
n1 n2	K, H												

举例：

操作前：( X1 ) =1 , ( X0 ) =1 , ( M4 ) =1 , ( M3 ) =1 , ( M2 ) =0 , ( M1 ) =1 , ( M0 ) =0 ;

SFTL X0 M0 K5 k2

操作后：( X1 ) =1 , ( X0 ) =1 , ( M4 ) =0 , ( M3 ) =1 , ( M2 ) =0 , ( M1 ) =1 , ( M0 ) =1 ;

4.4.7 WSFR 字右移指令

功能说明：将以 D.为首地址字元件组合向右移动 n2 为，其高位由 n2 位字元件组合 S.移入，移出的 n2 个低位被舍弃，而字元件组合 S.保持原值不变。

- WSFR - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

WSFR S. D. n1 n2

( 注：n2<=n1<=512 );

适用软元件：

	字软元件									位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D					
D.			KnY	KnM	KnS	T	C	D					
n1 n2	K, H												

举例：

操作前：( D20 ) =25 , ( D21 ) =15 , ( D0 ) =20 , ( D1 ) =30 , ( D2 ) =40 , ( D3 ) =50 , ( D4 ) =60 ,  
( D5 ) =70 , ( D6 ) =80 , ( D7 ) =90 ;

WSFR D20 D0 K8 k2

操作后：( D20 ) =25 , ( D21 ) =15 , ( D0 ) =40 , ( D1 ) =50 , ( D2 ) =60 , ( D3 ) =70 , ( D4 ) =80 ,  
( D5 ) =90 , ( D6 ) =25 , ( D7 ) =15 ;

4.4.8 **WSFL** 字左移指令

功能说明：将以 D.为首地址字元件组合向左移动 n2 为，其低位由 n2 位字元件组合 S.移入，移出的 n2 个高位被舍弃，而字元件组合 S.保持原值不变。

- WSFL - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

WSFL S. D. n1 n2

( 注：n2<=n1<=512 );

适用软元件：

	字软元件									位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D					
D.			KnY	KnM	KnS	T	C	D					
n1 n2	K, H												

举例：

操作前：( D0 ) =1 , ( D10 ) =25 , ( D11 ) =20 , ( D12 ) =15 , ( D13 ) =10 , ( D14 ) =5 ;

WSFL D0 D10 K5 k1

操作后：( D0 ) =1 , ( D10 ) =1 , ( D11 ) =25 , ( D12 ) =20 , ( D13 ) =15 , ( D14 ) =10 ;

4.4.9 **SFWR** 移位写入指令

功能说明：为控制先进先出的数据写入指令。执行时 D.先复位为零后作计数用。按执行次数，S.的内容从目标操作数之后的一个开始依次往后放入寄存器中。直到计数到达 n-1 个后不再填充寄存器只计数。

- SFWR - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-源操作数 )

SFWR S. D. n

( 注 1：2<n<=512 );

( 注 2：执行次数超过 n-1 次进位标志 M8022 置位 );

适用软元件：

	字软元件									位软元件			
S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D					

n	K, H												
---	------	--	--	--	--	--	--	--	--	--	--	--	--

举例：

操作前：( D0 ) =0x5 , ( D10 ) =0 , ( D11 ) =0 , ( D12 ) =0 , ( D13 ) =0x123 , ( M8022 ) =0 ；

SFWR D0 D10 K3

操作一次后：( D0 ) =0x5 , ( D10 ) =1 , ( D11 ) =0x5 , ( D12 ) =0 , ( D13 ) =0x123 , ( M8022 ) =0 ；

改变 D0 的值再次重复操作：( D0 ) =0x7 ；

操作二次后：( D0 ) =0x7 , ( D10 ) =2 , ( D11 ) =0x5 , ( D12 ) =0x7 , ( D13 ) =0x123 , ( M8022 ) =0 ；

改变 D0 的值再次重复操作：( D0 ) =0x9 ；

操作三次后：( D0 ) =0x9 , ( D10 ) =2 , ( D11 ) =0x5 , ( D12 ) =0x7 , ( D13 ) =0x123 , ( M8022 ) =1 ；

4.4.10 **SFRD** 移位读出指令

功能说明：为控制先进先出的数据读出指令。S.每次执行减一，直至减至零。每次执行时，总执行次数 X，S.之后的第 X 个寄存器的内容，放入从 D.开始的第 X 个寄存器。直到 S.减 n-1 时不再减少也不再填充寄存器。

- SFRD - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

SFRD S. D. n

( 注 1：2<n<=512 )；

( 注 2：源操作数减至零后，再次执行零点标志 M8020 置位 )；

适用软元件：

	字软元件									位软元件			
S.			KnY	KnM	KnS	T	C	D					
D.			KnY	KnM	KnS	T	C	D	V, Z				
N	K, H												

举例：

操作前：( D10 ) =2 , ( D20 ) =0 , ( D11 ) =0x2 , ( D12 ) =0x3 , ( D13 ) =123 , ( M8020 ) =0 ；

SFWR D10 D20 K3

操作一次后：( D10 ) =1 , ( D20 ) =0x2 , ( D11 ) =0x2 , ( D12 ) =0x3 , ( D13 ) =123 , ( M8020 ) =0 ；

操作二次后：( D10 ) =0 , ( D20 ) =0x3 , ( D11 ) =0x3 , ( D12 ) =0x3 , ( D13 ) =123 , ( M8020 ) =0 ；

操作三次后：( D10 ) =0 , ( D20 ) =0x3 , ( D11 ) =0x3 , ( D12 ) =0x3 , ( D13 ) =123 , ( M8020 ) =1 ；

4.5 数据处理指令

4.5.1 **ZRST** 批次复位指令

功能说明：复位 S1.至 S2.的全部寄存器。

- **ZRST** - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( D-目标操作数 )

**ZRST D1. D2.**

( 注 1：D1.D2.需为同一种软元件且  $D1.<=D2.$ ，如果  $D1.>D2.$ 则只复位 D1.中的软元件 )；

( 注 2：该指令为 16 位指令，但 D1.D2.也适用 32 位计数器。但是不能 16 位 32 位混合，如 D1.为 16 位计数器，D2.位 32 位计数器 )；

适用软元件：

	字软元件								位软元件			
D1. D2.						T	C	D		Y	M	S

举例：

操作前：( D0 ) =0x1234, ( D1 ) =0x2345, ( D2 ) =0x3456, ( D3 ) =0x4567, ( D4 ) =0x5678;

**ZRST D0 D4**

操作后：( D0 ) =0, ( D1 ) =0, ( D2 ) =0, ( D3 ) =0, ( D4 ) =0;

## 4.5.2 **DECO** 译码指令

功能说明：由源址 S.所表示的二进制值 m 使终址 D.中编号为 m 的位元件或字元件中  $b_m$  位置 ON。S.的位数指定为  $2^n$ 。

- **DECO** - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-源操作数 )

**DECO S. D. n**

( 注 1： $1<=n<=8$  )；

( 注 2：m=0 时不处理，0~8 以外会置位运算错误标志位 M8067 )；

( 注 3：m=8 时，如果译码命令 D.为软元件时，其点数是  $28=256$  点 )；

适用软元件：

	字软元件									位软元件			
S.	K, H					T	C	D	V, Z	X	Y	M	S
D.						T	C	D			Y	M	S
n	K, H												

举例：

操作前：( D0 ) =2, ( Y0 ) =0, ( Y1 ) =0, ( Y2 ) =0, ( Y3 ) =0;

**DECO D0 Y0 K3**

操作后：( D0 ) =2, ( Y0 ) =0, ( Y1 ) =0, ( Y2 ) =1, ( Y3 ) =0;

### 4.5.3 ENCO 编码指令

功能说明：把 S. 中置 ON 的位元件或字元件中置 ON 的位的位置值转换成二进制整数传送到 D.。S 的位数指定为 2<sup>n</sup> 位；

● ENCO—16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：(S-源操作数，D-目标操作数，n-源操作数)

ENCO S. D. n

(注 1：1<=n<=8)；

(注 2：m=0 时不处理，0~8 以外会置位运算错误标志位 M8067)；

(注 3：S. 是位软元件时 n<=8, 是字软元件时 n<=4)；

(注 4：S. 内多个位是 1 时，忽略低侧，另外全部位都为 1 时出现运算出错)；

(注 5：n=8 时，编码指令的 S. 如果是位软件，其点数是 28=256 点)；

适用软元件：

	字软元件										位软元件			
S.						T	C	D	V, Z		X	Y	M	S
D.						T	C	C	V, Z					
N	K, H													

举例：

操作前：( M10 ) =0 , ( M11 ) =0 , ( M12 ) =0 , ( M13 ) =1 , ( M14 ) =0 , ( M15 ) =0 , ( M16 ) =0 ,  
( M17 ) =0 , ( D0 ) =0 ；

ENCO M10 D0 K3

操作后：( M10 ) =0 , ( M11 ) =0 , ( M12 ) =0 , ( M13 ) =0 , ( M14 ) =0 , ( M15 ) =0 , ( M16 ) =0 ,  
( M17 ) =0 , ( D0 ) =3 ；

### 4.5.4 SUM ON 位数

功能说明：S.在二进制下 1 的位数存入 D.中；

- SUM - 16 位指令，源操作数与目标操作数为 16 位
- DSUM - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

SUM S. D.

DSUM S. D.

适用软元件：

	字软元件										位软元件			
S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z					

D.			KnY	KnM	KnS	T	C	D	V, Z				
----	--	--	-----	-----	-----	---	---	---	------	--	--	--	--

举例：

操作前：( D0 ) =0x7F , ( D1 ) =0；

**SUM D0 D1**

操作后：( D0 ) =0x7F , ( D1 ) =7；

操作前：( D1 , D0 ) =0xFFFFFFFF , ( D3 , D2 ) =0；

**DSUM D0 D2**

操作后：( D1 , D0 ) =0xFFFFFFFF , ( D3 , D2 ) =32；

### 4.5.5 **BON** ON 位判断

功能说明：判断 S.第 n 位是否为零；

- **BON** - 16 位指令，操作数、源操作数与目标操作数为 16 位
- **DBON** - 32 位指令，操作数、源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数，n-源操作数 )

**BON S. D. n**

**DBON S. D. n**

( 注 1：0<=n<=31 )；

( 注 2：n 超出允许范围时，会置位运算错误标志位 M8067 )；

适用软元件：

	字软元件										位软元件		
S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.											Y	M	S
n	K, H												

举例：

操作前：( D0 ) =4 , ( M20 ) =0；

**BON D0 M20 K2**

操作后：( D0 ) =4 , ( M20 ) =1；

### 4.5.6 **MEAN** 平均值

功能说明：S.起 n 个软元件求平均值，放入 D.中；

- **MEAN** - 16 位指令，操作数、源操作数与目标操作数为 16 位
- **DMEAN** - 32 位指令，操作数、源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数，n-源操作数 )

MEAN S. D. n

DMEAN S. D. n

( 注 1：1<=n<=64 )；

( 注 2：n 超出允许范围时，会置位运算错误标志位 M8067 )；

适用软元件：

	字软元件									位软元件			
S.		KnX	KnY	KnM	KnS	T	C	D					
D.			KnY	KnM	KnS	T	C	C	V, Z				
N	K, H												

举例：

操作前：( D0 ) =3 , ( D1 ) =5 , ( D2 ) =7 , ( D3 ) =0；

MEAN D0 D3 K3

操作后：( D0 ) =3 , ( D1 ) =5 , ( D2 ) =7 , ( D3 ) =5；

操作前：( D1 , D0 ) =0x111111 , ( D3 , D2 ) =0x222222 , ( D5 , D4 ) =0x333333 , ( D7 , D6 ) =0；

DMEAN D0 D6 K3

操作后：( D1 , D0 ) =0x111111 , ( D3 , D2 ) =0x222222 , ( D5 , D4 ) =0x333333 , ( D7 , D6 ) =0x222222；

4.5.7 **ANS** 报警的设置

功能说明：用于驱动信号报警器的方便指令；

- ANS - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，M-目标操作数，D-源操作数 )

ANS S. M. D

适用软元件：

	字软元件									位软元件			
S.						T							
M.	K, H												
D													S

M；它的范围在 0 到 32767 之间超出范围会报错 M8067

D；它的范围是 S900 到 S999 之间超出范围会报错

举例：

操作前； ( D901 ) =0;



ANS T0 K10 S901

操作后：( T0=0 ； S901=0 ) → ( 计数到 T0=10； S901=0 ) → ( T0=0； S901=1 )  
定时 1 秒钟，导通一秒以后 S901 被置位，置位以后，不管 ANS 语句是否导通，S901 被一直动作。

M8049 动作时，将会把 S900 到 S999 中导通的，最小一个编号值给存到 D8049 中。

M8049 动作时，S900 到 S999 中任意一个为 ON 时，M8048，将动作报警（为 ON）

4.5.8 ANR 报警的设置

功能说明：S900 到 S999 指令 ANS 将报警给解除；

- ANR - 16 位指令指令格式：无操作数，无软元件
- ANRP - 16 位指令指令格式：无操作数，无软元件

指令格式：

ANR

适用软元件：

	字软元件										位软元件			

举例：

操作前；

ANR

操作后：当 ANR 导通时，会从 S900 到 S999（ON）状态最小编号，开始依次复位。

ANRP

操作后：当 ANRP 导通一次，会把 S900 到 S999（ON）状态最小编号，复位。

4.5.9 SQR BIN 数据开方运算

功能说明：进行平方根开方运算指令；

- SQR - 16 位指令，源操作数与目标操作数为 16 位
- DSQR - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

SQR S. D.

DSQR S. D.

( 注 1：进行平方根运算的指令 )；

( 注 2：仅在 S.是正数才有效，如果是负数时运算错误标志位 M8067 会工作，指令不被执行 )；

( 注 3：运算结果舍去小数为整数。舍去时，借位标志位会动作。)；

( 注 4：结果为零时，令标志位 M8020 会动作 )；

适用软元件：

	字软元件								位软元件			
S.	K, H							D				
D.								D				

举例：

操作前：(D11,D10)=4,(D13,D12)=0;

DSQR D10 D12

操作前：(D11,D10)=4,(D13,D12)=2.000;

4.5.10 FLT BIN 整数->2 进制浮点转换指令

功能说明：S.内的 BIN 整数值，转换为 2 进制浮点值放入 D.及其后一软元件；

- FLT - 16 位指令，源操作数与目标操作数为 16 位
- DFLT - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

FLT S. D.

DFLT S. D.

( 注：常数 K、H 在各浮点运算指令中被自动转换，因此在本 FLT 指令中不能使用 );

适用软元件：

	字软元件								位软元件			
S. D.								D				

举例：

操作前：( D0 ) =1 ,( D13 , D12 ) =0 ；

FLT D0 D12

操作后：( D0 ) =1 , float ( D13 , D12 ) =0x3F80 即二进制浮点数 1.000 ；

操作前：( D1 , D0 ) =1000000 ,( D13 , D12 ) =0 ；

DFLT D0 D12

操作后：( D1 , D0 ) =1000000 , float ( D13 , D12 ) =0x49742400 二进制浮点值 1.000e+006;

4.6 高速处理

脉冲输出类指令使用的特殊寄存器：

软元件	内容含义
M8029	指令执行完成标志位

M8144	Y2 停止脉冲输出的指令
M8145	Y0 停止脉冲输出的指令
M8146	Y1 停止脉冲输出的指令
M8147	Y0 脉冲输出中的监视 (BUSY/READY)
M8148	Y1 脉冲输出中的监视 (BUSY/READY)
M8149	Y2 脉冲输出中的监视 (BUSY/READY)

编号	位数	出厂值	内容含义
D8140 (低位)	32	0	Y0 输出位置当前值, 应用脉冲指令 PLSY, PLSR 时, 对脉冲输出值进行累加当前值
D8141 (高位)			
D8142 (低位)	32	0	Y1 输出位置当前值, 应用脉冲指令 PLSY, PLSR 时, 对脉冲输出值进行累加当前值
D8143 (高位)			
D8160 (低位)	32	0	Y2 输出位置当前值, 应用脉冲指令 PLSY, PLSR 时, 对脉冲输出值进行累加当前值
D8161 (高位)			
D8136 (低位)	32	0	Y0, Y1, Y2 输出脉冲合计数的累计值

4.6.1 PLSY 脉冲输出指令

功能说明：在 D.指定的输出口，输出一个脉冲串。脉冲串频率在源操作数 S1.内，数量在源操作数 S2.内，占空比为 50%；

- PLSY - 16 位指令，源操作数与目标操作数为 16 位
- DPLSY - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

PLSY S1. S2. D.

DPLSY S1. S2. D.

( 注 1：16 位指令：1<=S1<=32767，32 位指令：1<=S1<=100000 )；

( 注 2：16 位指令：0<=S2<=32767，32 位指令：0<=S2<=2147483647 )；

( 注 3：目标操作数必须为支持脉冲输出的 Y 软元件 )；

适用软元件：

		字软元件									位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.												Y		

举例：

```
PLSY K1000 K100 Y0
```

操作后：Y0 通道输出一个频率为 1000Hz，数量为 100 个，占空比为 50%的脉冲串；

4.6.2 **PWM** 脉宽调制指令

功能说明：在 D.指定的输出口，输出脉冲。脉冲的脉宽在源操作数 S1.内，周期在源操作数 S2.内；

- PWM - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

```
PWM S1. S2. D.
```

( 注 1：0<=S1<=32767 )；

( 注 2：1<=S2<=32767 )；

( 注 3：目标操作数必须为支持脉冲输出的 Y 软元件 )；

适用软元件：

		字软元件									位软元件			
S1.	S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.												Y		

举例：

```
PLSY K1 K2 Y1
```

操作后：Y1 通道输出一个周期为 2mS，脉宽为 1mS 的脉冲；

4.6.3 **PWM** 带加减速脉冲输出指令

功能说明：在 D.指定的输出口，输出一个开始时十级加速至最高频，结束时十级减速至最低频的脉冲串。脉冲串最高频率在源操作数 S1.内，总数量在源操作数 S2.内，加减速时间在 S3.内，占空比为 50%；

- PLSR - 16 位指令，源操作数与目标操作数为 16 位
- DPLSR - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

```
PLSR S1. S2. S3. D.
```

DPLSR S1. S2. S3. D.

(注 1：16 位指令：1<=S1<=32767，32 位指令：1<=S1<=100000)；

(注 2：16 位指令：110<=S2<=32767，32 位指令：110<=S2<=2147483647)；

(注 3：1<=S3<=5000)；

(注 4：目标操作数必须为支持脉冲输出的 Y 软元件)；

(注 5：参数设置时需要注意是否合理，例如：最高频很低加速时间内无法达到最高频，这些情况仍会有脉冲输出，但不会满足所有设置条件)；

适用软元件：

	字软元件									位软元件			
S1. S2. S3.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.											Y		

举例：

PLSR K10000 K10000 K50 Y2

操作后：Y2 通道输出一个总数量为 10000 个，占空比为 50%，频率开始时十级加速至 10000Hz，结束时十级减速的脉冲串；

4.7 外围设备 I/O

4.7.1 FROM BFM 读出指令

功能说明：从第 m1 号从机模块缓冲储存器（BFM）的 m2 起始地址的储存区，开始读出 n 个数据放入从本机 D.开始的软元件中。

- FROM - 16 位指令，操作数与目标操作数为 16 位
- DFROM - 32 位指令，操作数与目标操作数为 32 位

指令格式：( D-目标操作数，m、n-操作数 )

FROM m1 m2 D. n

(注 1：-1<=m1<=7)；

(注 2：0<=m2<=32767)；

(注 3：1<=n<=32767)；

(注 4：m1 为-1 时代表对本机操作)；

适用软元件：

			字软元件								位软元件			
m1	m2	n	K, H											
D.				KnY	KnM	KnS	T	C	D	V, Z				

举例：

FROM K2 K20 D5 K2

操作后：2 号从机的 BFM 存储器第 20、21 个数据，放入本机的 D5、D6 软元件；

4.7.2 TO BFM 写入指令

功能说明：从本机 S.开始的软元件，读出 n 个数据。放入第 m1 号从机模块缓冲存储器（BFM）以 m2 起始地址开始的储存区。

- TO - 16 位指令，操作数与源操作数为 16 位
- DTO - 32 位指令，操作数与源操作数为 32 位

指令格式：( S-目标操作数，m、n-操作数 )

TO m1 m2 S. n

( 注 1：-1<=m1<=7 )；

( 注 2：0<=m2<=32767 )；

( 注 3：1<=n<=32767 )；

( 注 4：m1 为-1 时代表对本机操作 )；

适用软元件：

			字软元件								位软元件			
m1	m2	n	K, H											
S.			K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z			

举例：

TO K2 K20 D5 K2

操作后：本机的 D5、D6 软元件数据,放入 2 号从机的 BFM 存储器第 20、21 个储存区；

4.8 外围设备 SER

4.8.1 ASCII HEX->ASCII 转换指令

功能说明：S.按照 16 进制，从低位开始共 n 位转化成 ASCII 码，转化后的 ASCII 码依次放入 D.。

- ASCII - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

ASCII S. D. n

( 注 1：1<=n<=256 )；

( 注 2：M8161=0 时，为 16 位转化模式。M8161=1 时，为 8 位转化模式。)；

适用软元件：

			字软元件								位软元件			

S.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
D.			KnY	KnM	KnS	T	C	D					
n	K, H												

举例：

操作前：( D0 ) =0x270F , ( D5 ) =0 , ( D6 ) =0 ;

ASCII D0 D5 K4

操作后：( D0 ) =0x270F , ( D5 ) =0x3732 , ( D6 ) =0x4630 ;

## 4.8.2 CCD 校验码指令

功能说明：M8161 为 0 时，以 S 指定的元件为起始的 n 点 8 位数据（16 位寄存器先低位后高位），将其总和与水平检验数据存储在 D 软元件与之后的一个软元件中。M8161 为 1 时只取低 8 位数据，其他操作与上述一致。

- CCD - 16 位指令，操作数、源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数，n-操作数 )

CCD S. D. n

( 注 1：1≤n≤256 )；

( 注 2：M8161=0 时，为 16 位转化模式。M8161=1 时，为 8 位转化模式。 )；

适用软元件：

	字软元件								位软元件				
S.		KnX	KnY	KnM	KnS	T	C	D					
D.			KnY	KnM	KnS	T	C	D					
n	K, H							D					

举例：

操作前：( M8161 ) =1 , ( D0 ) =37 , ( D1 ) =166 , ( D2 ) =77 , ( D10 ) =0 , ( D11 ) =206 ;

二进制每一位 1 的个数，为奇数时水平校验为 1，为偶数时水平校验为 0；

( D0 ) =37 -00100101

( D1 ) =166 -10100110

( D2 ) =77 -01001101

( D11 ) =206 -11001110

( D10 ) = ( D0 ) + ( D1 ) + ( D2 )

CCD D0 D10 K3

操作后：( M8161 ) =1 , ( D0 ) =37 , ( D1 ) =166 , ( D2 ) =77 , ( D10 ) =280 , ( D11 ) =206 ;

4.8.3 **PID** PID 运算指令

功能说明：用于进行 PID 控制的 PID 运算程序。达到采样时间的 PID 指令在其后的扫描时进行 PID 运算。参数中 S1.为设定目标值 SV，S2.为测量值 PV，S3.~S3.+24 控制参数，D.为输出值 MV。

- PID - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

PID S1. S2. S3. D.

( 注 1：需要占有自 S3.起始的 25 个数据寄存器 )；

( 注 2：若指定电池保持寄存器时，在可编程控制器 RUN 时，务必清除保持内容 )；

寄存器	功用	取值范围
S3.	采样时间 (Ts)	1~32767(ms) (但比运算周期短的时间无法执行)
S3. +1	动作方向 (ACT)	Bit0 0: 正动作 1: 逆动作 Bit1 0: 关闭输入变化量报警, 1: 打开 Bit2 0: 关闭输出变化量报警, 1: 打开 Bit3 不可用 Bit4: 关闭自动调谐, 1: 打开 Bit5: 关闭输出上下限, 1: 打开 Bit6~bit15 不可用
S3. +2	输入滤波常数 (α)	0~99[%], 0 时没有输入滤波
S3. +3	比例增益 (Kp)	1~32767[%]
S3. +4	积分时间 (TI)	0~32767(*100ms) 0 时作为∞处理
S3. +5	微分增益 (KD)	0~100[%] 0 时无积分增益
S3. +6	微分时间 (TD)	0~32767(*10ms) 0 时无微分处理
S3. +7	PID 运算的内部处理占用	
...		
S3. +19		
S3. +20		
S3. +21	输入变化量 (增侧) 报警设定值	0~32767 (S3+1 的 bit1=1 时有效)
S3. +22	输入变化量 (减侧) 报警设定值	0~32767 (S3+1 的 bit1=1 时有效)
S3. +22	输出变化量 (增侧) 报警设定值。 另外，输出上限设定值	0~32767 (S3+1 的 bit2=1 bit5=0 时有效) -32767~32767 (S3+1 的 bit2=0 bit5=1 时有效)
S3. +23	输出变化量 (减侧) 报警设定值。 另外，输出下限设定值	0~32767 (S3+1 的 bit2=1 bit5=0 时有效)



		-32767~32767 (S3+1 的 bit2=0 bit5=1 时有效)	
S3. +24	报警输出	Bit0 输入变化量上溢出 Bit1 输入变化量下溢出 Bit2 输出变化量上溢出 Bit3 输出变化量下溢出	S3+1 的 bit1=1 或 bit2=1 时有效

适用软元件：

	字软元件								位软元件			
S1. S2. S3. D.								D				

举例：  
操作前：目标值 SV(D0)，测定值 PV ( D1 )，参数 ( D100 )，输出值 MV ( D150 )；  
PID D0 D1 D100 D150

4.9 浮点数指令

4.9.1 ECMP 二进制浮点比较指令

功能说明：将 S1.与 S2.内的二进制浮点值进行比较，并将比较结果送到 D.及其后两个软元件上。  
• DECMP - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DECMP S1. S2. D.

( 注 1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理 )；  
( 注 2：D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用 )；

适用软元件：

	字软元件								位软元件			
S1. S2.	K, H							D				
D.										Y	M	S

举例：  
DECMP D0 D7 M3  
操作后：当 float (D1，D0)> float ( D8，D7 ) 时，( M3 ) =1，( M4 ) =0，( M5 ) =0；

当 float (D1 , D0)= float ( D8 , D7 ) 时 , ( M3 ) =0 , ( M4 ) =1 , ( M5 ) =0 ;  
当 float (D1 , D0)< float ( D8 , D7 ) 时 , ( M3 ) =0 , ( M4 ) =0 , ( M5 ) =1 ;

4.9.2      **EZCP**    二进制浮点数区域比较指令

功能说明：将 S.与 S1.和 S2.的二进制浮点值比较，并将比较结果送到 D.及其后两个软元件上。

- DEZCP - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEZCP    S1.    S2.    S.    D.

- ( 注 1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理 )；
- ( 注 2：源 S1.的内容不得大于源 S2.的内容，当源 S1.的内容大于源 S2.的内容时，系统默认把源 S2.的内容当成源 S1.的内容进行计算 )；
- ( 注 3：D.占有 3 点位软元件 )；
- ( 注 4：S1.<=S2. )；

适用软元件：

	字软元件								位软元件			
S10. S2.	K, H							D				
S.												
D.										Y	M	S

举例：

DEZCP K100 K120 D7 M2

操作后：当 K100> float ( D8 , D7 ) 时 , ( M2 ) =1 , ( M3 ) =0 , ( M4 ) =0 ;  
当 K100<= float ( D8 , D7 ) <=K120 时 , ( M2 ) =0 , ( M3 ) =1 , ( M4 ) =0 ;  
当 K120< float ( D8 , D7 ) 时 , ( M2 ) =0 , ( M3 ) =0 , ( M4 ) =1 ;

4.9.3      **EBCD**    二进制浮点转十进制浮点

功能说明：将 S.指定的软元件的二进制浮点值转化成十进制浮点值，存入 D.中及其后一个软元件中。

- DEBCD - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEBCD    S.    D.

- ( 注 1：二进制浮点数值为不易判别的数值，可以通过将其转化成十进制浮点值得外部设备很

容易的监控);

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：bin float ( D1 , D0 ) =1120403456 , ( D6 ) = 0 , ( D7 ) = 0 ;

DEBCD D0 D6

操作后：bin float ( D1 , D0 ) =1120403456 , ( D6 ) = 1000 , ( D7 ) = -1 ;

(D7 , D6)表示的是 1000\*10^(-1),值也为 100.

4.9.4 EBIN 十进制浮点转二进制浮点

功能说明：将 S. 及其后一个软元件指定的软元件的十进制浮点值转化成二进制浮点值，存入 D.中。

- DEBIN - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEBIN S. D.

注：十进制浮点数 S=D0; ( D1 , D0 ) =D0 \* 10^(D1) ;

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：( D0 ) =1 , ( D1 ) =2 , (D7 , D6) =0 , (D1 , D0)表示的是 1\*10^(2),值也为 100.

DEBIN D0 D6

操作后：( D0 ) =1 , ( D1 ) =2 , bin float (D7 , D6) =1120403456 ;

4.9.5 EADD 二进制浮点数加法指令

功能说明：将 S1.与 S2.内的二进制浮点值相加后，作为二进制浮点值存入 D.中。

- DEADD - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEADD S1. S2. D.

(注1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理)；

(注2：源数据和目的地址也可以指定同一元件号，此时，如用连续执行指令，就会在每个运算周期均相加，因此，请注意)；

适用软元件：

	字软元件								位软元件			
S1. S2.	K, H							D				
D.								D				

举例：

操作前：float ( D1 , D0 ) =5.250 , float ( D8 , D7 ) =2.500 , ( D11 , D10 ) =0；

DEADD D0 D7 D10

操作后：float ( D1 , D0 ) =5.250 , float ( D8 , D7 ) =2.500 , float ( D11 , D10 ) =7.750；

4.9.6 **ESUB** 二进制浮点数减法指令

功能说明：将 S1.指定的软元件的二进制浮点值减去 S2.指定的软元件的二进制浮点值，并将结果作为二进制浮点值存入 D.中。

- DESUB - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DESUB S1. S2. D.

(注1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理)；

(注2：源数据和目的地址也可以指定同一元件号，此时，如用连续执行指令，就会在每个运算周期均运行该指令，因此，请注意)；

适用软元件：

	字软元件								位软元件			
S1. S2.	K, H							D				
D.								D				

举例：

操作前：float ( D1 , D0 ) =5.750 , float ( D8 , D7 ) =2.500 , ( D11 , D10 ) =0；

DESUB D0 D7 D10

操作后：float ( D1 , D0 ) =5.750 , float ( D8 , D7 ) =2.500 , float ( D11 , D10 ) =3.250；

4.9.7 **EMUL** 二进制浮点数乘法指令

功能说明：将 S1.与 S2.内的二进制浮点值的积作为二进制浮点值存入 D.中。

- DEMUL - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEMUL S1. S2. D.

( 注：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理 )；

适用软元件：

	字软元件								位软元件			
S1. S2.	K, H							D				
D.								D				

举例：

操作前：float ( D1 , D0 ) =2.500 , float ( D8 , D7 ) =2.500 , ( D11 , D10 ) =0；

DEMUL D0 D7 D10

操作后：float ( D1 , D0 ) =2.500 , float ( D8 , D7 ) =2.500 , float ( D11 , D10 ) =6.250；

### 4.9.8 EDIV 二进制浮点数除法指令

功能说明：将 S1.指定的软元件的二进制浮点值除以 S2.指定的软元件的二进制浮点值，并将结果作为二进制浮点值存入 D.中。

- DEDIV - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DEDIV S1. S2. D.

( 注 1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理 )；

( 注 2：除数 S2.为 0 时，则置位运算错误标志位 M8067，指令不能执行 )；

适用软元件：

	字软元件								位软元件			
S1. S2.	K, H							D				
D.								D				

举例：

操作前：float ( D1 , D0 ) =6.250 , float ( D8 , D7 ) =2.500 , ( D11 , D10 ) =0；

DEDIV D0 D7 D10

操作后：float ( D1 , D0 ) =6.250 , float ( D8 , D7 ) =2.500 , float ( D11 , D10 ) =2.50；

### 4.9.9 ESQR 十进制浮点开方

功能说明：将 S.及其后的一个软元件表示的二进制浮点值进行开方根运算，存入 D. 及其后的一个软元件中。

- DESQR - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DESQR S. D.

(注1：常数 K、H 被指定为源数据时，自动转换成二进制浮点值处理)；

(注2：运算结果为零时，零标志号动作 M8020 )

(注2：源数据的内容只有正数有效，负数时运算错误标志位 M8067，指令不能执行)；

适用软元件：

	字软元件								位软元件			
S.	K, H							D				
D.								D				

举例：

操作前：float ( D1 , D0 ) =9.000 , float ( D7 , D6 ) =0;

DESQR D0 D6

操作后：float ( D1 , D0 ) =9.000 , float ( D7 , D6 ) =3.000 ；

4.9.10 INT 十进制浮点转 BIN 整数

功能说明：将 S. 及其后的一个软元件表示的软元件的十进制浮点值转成 BIN 整数，存入 D. 及其后的一个软元件中。

- INT - 16 位指令，源操作数为 16 位
- DINT - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DINT S. D.

(注1：运算结果为零时，零标志号动作 M8020 为 NO )

(注2：转化时不满 1 而舍去，借位标志号动作 M8021 为 NO )

(注3：运算结果超出以下范围而发生溢出时，进位标志位为 NO )

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：( D0 , D1 ) =123.6,(D7 , D6) =0 ；

DINT D0 D6

操作后：(D0 , D1 ) =123.6 ,(D7 , D6) =123 ；

4.9.11 SIN 浮点运算

功能说明：S.指定的角度（RAD）的 SIN 值，并传送到 D.中的指令。

- DSIN - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DSIN S. D.+

（注 1：源操作数保存的角度是弧度）

（注 2：计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度  
超出范围的产生报错 M8067 为 NO）

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：(D3, D2)=0.785（弧度），(D7，D6)=0；

DSIN D2 D6

操作后：(D3, D2)=0.785，(D7，D6)=0.707；

4.9.12 COS 浮点运算

功能说明：源操作数指定的角度（RAD）的 COS 值，并传送到目的地址中的指令。

- DCOS - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DCOS S. D.

（注 1：源操作数保存的角度是弧度）

（注 2：计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度  
超出范围的产生报错 M8067 为 NO）

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：（D3，D2）= 45 \* 31415926/1800000000=0.785（45 度的角度）

DCOS D2 D6

操作后：(D7，D6)=0.707；

4.9.13 TAN 浮点运算

功能说明：源操作数指定的角度（RAD）的 TAN 值，并传送到目的地址中的指令。

- DTAN - 32 位指令，源操作数与目标操作数为 32 位

指令格式：( S-源操作数，D-目标操作数 )

DTAN S. D.

（注 1：源操作数保存的角度是弧度）

（注 2：计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度，当为 90 度和 270 度以及超出范围的会产生报错 M8067 为 NO）

适用软元件：

	字软元件								位软元件			
S.								D				
D.								D				

举例：

操作前：(D3，D2)=0.785（弧度），(D7，D6)=0；

DTAN D2 D6

操作后：(D3,D2)=0.785（弧度），(D7，D6)=1.000；

4.10 上下字节变换

4.10.1 SWAP 上下字节变换指令

功能说明：S.的各个低八位与高八位相互交换。

- SWAP - 16 位指令，源操作数为 16 位
- DSWAP - 32 位指令，源操作数为 32 位

指令格式：( S-源操作数 )

SWAP S.

DSWAP S.

适用软元件：

	字软元件								位软元件			
S.			KnY	KnM	KnS	T	C	D	V, Z			



举例：  
操作前：( D0 ) =0x1234；  
**SWAP D0**  
操作后：( D0 ) =0x3412；

操作前：( D0 ) =0x12345678；  
**DSWAP D0**  
操作后：( D0 ) =0x34127856；

## 4.11 时钟连算指令

### 4.11.1 **TCMP** 时钟数据比较指令

功能说明：S1.作时，S2.作分，S3.作秒。与将 S.起始的三点内的时钟数据进行比较，并将比较结果送到 D.及其后两个软元件上。  
● **TCMP** - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

**TCMP S1. S2. S3. S. D.**  
( 注 1：所有源操作数都被看成二进制值处理 )；  
( 注 2：D.自动占三点，如 D.为 M0 时，M0、M1、M2 自动被占用 )；

适用软元件：

	字软元件										位软元件		
S1. S2. S3.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
S.						T	C	D					
D.											Y	M	S

举例：  
**TCMP K10 K30 K50 D7 M0**  
操作后：当 10 时 30 分 50 秒> (( D7 ) 时 ( D8 ) 分 ( D9 ) 秒) 时，( M0 ) =1，( M1 ) =0，( M2 ) =0；  
当 10 时 30 分 50 秒= (( D7 ) 时 ( D8 ) 分 ( D9 ) 秒) 时，( M2 ) =0，( M3 ) =1，( M4 ) =0；  
当 10 时 30 分 50 秒< (( D7 ) 时 ( D8 ) 分 ( D9 ) 秒) 时，( M2 ) =0，( M3 ) =0，( M4 ) =1；

### 4.11.2 **TZCP** 时钟数据区间比较指令

功能说明：将 S.起始的三点内的时钟数据.与将 S1.起始的三点内的时钟数据和将 S2.起始的三点内的时钟数据进行比较，并将比较结果送到 D.及其后两个软元件上。  
● **TZCP** - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

TZCP S1. S2. S. D.

( 注 1：所有源操作数都被看成二进制值处理 )；

( 注 2：D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用 )；

( 注 3：源操作数 1 起始的三点内的时钟数据不得大于源操作数 2 起始的三点内的时钟数据，当不满足该条件时，默认两个比较值都为源操作数 1 起始的三点内的时钟数 )；

适用软元件：

			字软元件							位软元件			
S1.	S2.	S					T	C	D				
D.											Y	M	S

举例：

TZCP D10 D20 D0 M2

操作后：当 (( D10 ) 时 ( D11 ) 分 ( D12 ) 秒 ) > (( D0 ) 时 ( D1 ) 分 ( D2 ) 秒 ) , ( M2 ) =1 , ( M3 ) =0 , ( M4 ) =0 ;  
当 (( D10 ) 时 ( D11 ) 分 ( D12 ) 秒 ) <= (( D0 ) 时 ( D1 ) 分 ( D2 ) 秒 ) <= (( D20 ) 时 ( D21 ) 分 ( D22 ) 秒 ) , ( M2 ) =0 , ( M3 ) =1 , ( M4 ) =0 ;  
当 (( D0 ) 时 ( D1 ) 分 ( D2 ) 秒 ) < (( D20 ) 时 ( D21 ) 分 ( D22 ) 秒 ) , ( M2 ) =0 , ( M3 ) =0 , ( M4 ) =1 ;

4.11.3 TADD 时钟数据加法指令

功能说明：保存于 S1.起始的三点内的时钟数据，加上源 S2.起始的三点内的时钟数据，并将其结果保存于以 D.起始的三点软元件内。

- TADD - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

TADD S1. S2. D.

( 注 1：当运算结果为 0 时 ( 0 时 0 分 0 秒 )，零标志位 M8020 置位 )；

( 注 2：当运算结果超过 24 小时时，进位标志位 M8022 置位，将运算的结果减去 24 小时后，将该值作为运算结果保存 )；

适用软元件：

			字软元件							位软元件			
S1.	S2.	D.					T	C	D				

举例：

操作前：( D0 ) =10 , ( D1 ) =30 , ( D2 ) =10 , ( D10 ) =3 , ( D11 ) =10 , ( D12 ) =5 , ( D20 ) =0 , ( D21 ) =0 , ( D22 ) =0 , ( M8020 ) =0 , ( M8022 ) =0 ;

TADD D0 D10 D20

操作后：( D0 ) =10 , ( D1 ) =30 , ( D2 ) =10 , ( D10 ) =3 , ( D11 ) =10 , ( D12 ) =5 , ( D20 ) =13 , ( D21 ) =40 , ( D22 ) =15 , ( M8020 ) =0 , ( M8022 ) =0 ;

4.11.4 TSUB 时钟数据减法指令

功能说明：保存于 S1.起始的三点内的时钟数据，减去 S2.起始的三点内的时钟数据，并将其结果保存于以 D.起始的三点软元件内。

- TSUB - 16 位指令，源操作数与目标操作数为 16 位

指令格式：( S-源操作数，D-目标操作数 )

TSUB S1. S2. D.

( 注 1：当运算结果为 0 时 ( 0 时 0 分 0 秒 )，零标志位 M8020 置位 )；

( 注 2：当运算结果超过 0 小时时，借位标志位 M8021 置位，将运算的结果加上 24 小时后，将该值作为运算结果保存 )；

适用软元件：

			字软元件							位软元件			
S1.	S2.	D.					T	C	D				

举例：

操作前：( D0 ) =5 , ( D1 ) =20 , ( D2 ) =40 , ( D10 ) =18 , ( D11 ) =10 , ( D12 ) =5 , ( D20 ) =0 , ( D21 ) =0 , ( D22 ) =0 , ( M8020 ) =0 , ( M8021 ) =0 ；

TSUB D0 D10 D20

操作后：( D0 ) =5 , ( D1 ) =20 , ( D2 ) =40 , ( D10 ) =18 , ( D11 ) =10 , ( D12 ) =5 , ( D20 ) =11 , ( D21 ) =10 , ( D22 ) =35 , ( M8020 ) =0 , ( M8021 ) =1 ；

4.11.5 TRD 时钟数据读出指令

功能说明：将可编程控制器的实时时钟的时钟数据，按照年、月、日、时、分、秒、星期读入，以 D.为起始的七点数据寄存器中。

- TRD - 16 位指令，目标操作数为 16 位

指令格式：( D-目标操作数 )

TRD D.

适用软元件：

			字软元件							位软元件			
D.							T	C	D				

举例：

操作前：( D0 ) =0 , ( D1 ) =0 , ( D2 ) =0 , ( D3 ) =0 , ( D4 ) =0 , ( D5 ) =0 , ( D6 ) =0 ；

TRD D0

操作后：( D0 ) =2015 , ( D1 ) =12 , ( D2 ) =26 , ( D3 ) =11 , ( D4 ) =11 , ( D5 ) =9 , ( D6 ) =6 ；

4.11.6 TWR 时钟数据写入指令

功能说明：将时钟数据即以 D.为起始的七点数据寄存器，写入可编程控制器的实时时钟中。

- DESUB - 16 位指令，源操作数为 32 位

指令格式：( S-源操作数 ) 23

TWR S.

适用软元件：

	字软元件							位软元件			
S.						T	C	D			

举例：

操作前：( D0 ) =2015 , ( D1 ) =12 , ( D2 ) ==26 , ( D3 ) =11 , ( D4 ) =11 , ( D5 ) =9 , ( D6 ) =6 ；

TWR D0

操作后： 可编程控制器的时间变为，2015 年 12 月 26 日 11 时 11 分 9 秒星期六；

4.12 接点比较指令

4.12.1 LD◆ 母线连接的触点比较指令

功能说明：对 S1.与 S2.内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，连接母线的触点比较指令。

- LD◆ - 16 位指令，目标操作数为 16 位
- LDD◆ - 32 位指令，目标操作数为 32 位

指令格式：( S-源操作数 )

LD◆ S1. S2.

LDD◆ S1. S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
224	LD=	LDD=	S1. =S2.	S1. ≠S2.
225	LD>	LDD>	S1. >S2.	S1. ≤S2.
226	LD<	LDD<	S1. <S2.	S1. ≥S2.
228	LD<>	LDD<>	S1. ≠S2.	S1. =S2.
229	LD<=即≤	LDD<=即≤	S1. ≤S2.	S1. >=S2.
230	LD>=即≥	LDD>=即≥	S1. ≥S2.	S1. <S2.

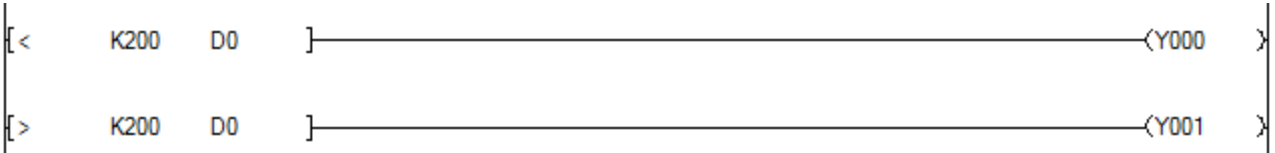
适用软元件：

	字软元件	位软元件
--	------	------

S1, S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z				
---------	------	-----	-----	-----	-----	---	---	---	------	--	--	--	--

举例：

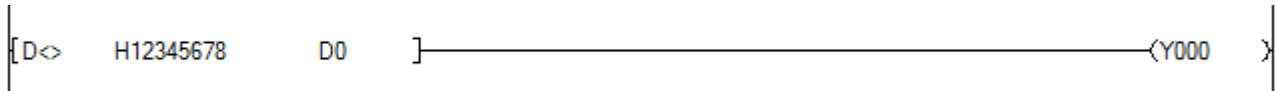
操作前：( D0 ) =100, ( Y0 ) =0 , ( Y1 ) =0 ；



LD< K200 D0  
OUT Y000  
LD> K200 D0  
OUT Y001

操作后：( D0 ) =100, ( Y0 ) =0 , ( Y1 ) =1 ；

操作前：( D0 ) =0X12345678, ( Y0 ) =0 ；



LDD<> H12345678 D0  
OUT Y000

操作后：( D0 ) =100, ( Y0 ) =0 ；

### 4.12.2 AND◆ 串联连接的触点比较指令

功能说明：对 S1. 与 S2. 内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，与其它接点串联连接的触点比较指令。

- AND◆—16 位指令，目标操作数为 16 位
- ANDD◆—32 位指令，目标操作数为 32 位

指令格式：(S-源操作数)

AND◆ S1. S2.  
ANDD◆ S1. S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
232	AND=	ANDD=	S1. =S2.	S1. ≠S2.
233	AND>	ANDD>	S1. >S2.	S1. ≤S2.
234	AND<	ANDD<	S1. <S2.	S1. ≥S2.

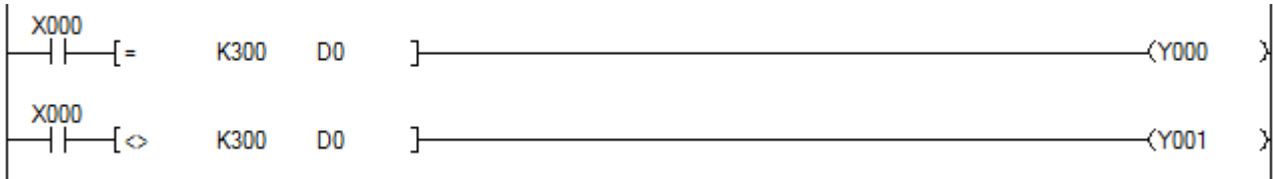
236	AND<>	ANDD<>	S1. ≠S2.	S1. =S2.
237	AND<=即≤	ANDD<=即≤	S1. ≤S2.	S1. >S2.
238	AND>=即≥	ANDD>=即≥	S1. ≥S2.	S1. <S2.

适用软元件：

	字软元件										位软元件			
S1. S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z					

举例：

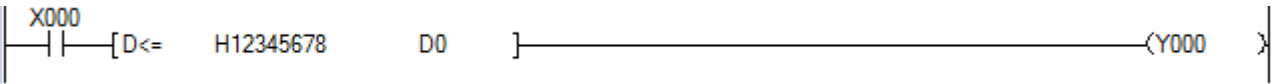
操作前：( X0 ) =1 , ( D0 ) =300, ( Y0 ) =0 , ( Y1 ) =0；



```
LD      X000
AND=    K300      D0
OUT      Y000
LD      X000
AND<>   K300      D0
OUT      Y001
```

操作后：( X0 ) =1, ( D0 ) =300, ( Y0 ) =1, ( Y1 ) =0；

操作前：( X0 ) =1, ( D0 ) =0X12345678, ( Y0 ) =0；



```
LD      X000
ANDD<=  H12345678      D0
OUT      Y000
```

操作后：( X0 ) =1, ( D0 ) =0x12345678, ( Y0 ) =1；

4.12.3 **OR◆** 并联连接的触点比较指令

功能说明：对 S1.与 S2.内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，与其它接点并联连接的触点比较指令。

- **OR◆** - 16 位指令，目标操作数为 16 位
- **ORD◆** - 32 位指令，目标操作数为 32 位

指令格式 : ( S-源操作数 )

OR◆ S1. S2.

ORD◆ S1. S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
240	OR=	ORD=	S1. =S2.	S1. ≠S2.
241	OR>	ORD>	S1. >S2.	S1. ≤S2.
242	OR<	ORD<	S1. <S2.	S1. ≥S2.
244	OR<>	ORD<>	S1. ≠S2.	S1. =S2.
245	OR<=即≤	ORD<=即≤	S1. ≤S2.	S1. >S2.
246	OR>=即≥	ORD>=即≥	S1. ≥S2.	S1. <S2.

适用软元件：

	字软元件										位软元件			
S1. S2.	K, H	KnX	KnY	KnM	KnS	T	C	D	V, Z					

举例：

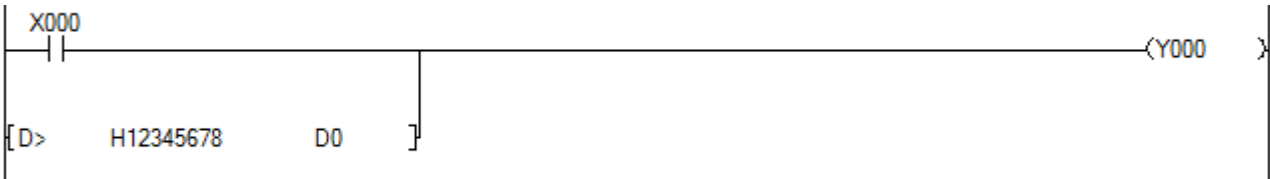
操作前 : ( X0 ) =0 , ( D0 ) =256, ( Y1 ) =0 ;



LD X000  
OR= K256 D0  
OUT Y000

操作后 : ( X0 ) =0 , ( D0 ) =256, ( Y1 ) =1 ;

操作前 : ( X0 ) =0 , ( D0 ) =0X12345678, ( Y0 ) =0 ;



LD X000

ORD>     H12345678     D0

OUT     Y000

操作后 : ( X0 ) =0 , ( D0 ) =0x12345678, ( Y0 ) =0 ;