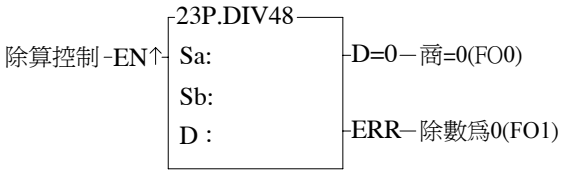
第 9 章：進階篇應用指令

FUN23 P DIV48	48 位元除法運算（48-BIT DIVISION） （將 Sa 除以 Sb 所得之商存到 D 去）	FUN23 P DIV48

指令說明

階梯圖符號



運算元

Sa：被除數之起頭暫存器號碼。
Sb：除數之起頭暫存器號碼。
D ：存放結果(商)之起頭暫存器號碼。
Sa，Sb，D 可結合 V、Z 作間接定址應用。

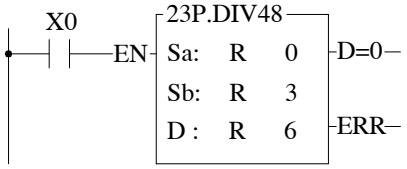
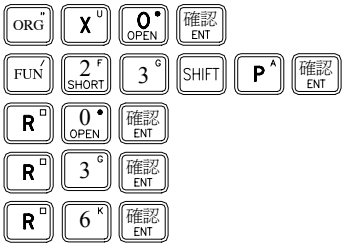
運算元 \ 範圍	HR	OR	SR	ROR	DR	XR
	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	V、Z
Sa	○	○	○	○	○	○
Sb	○	○	○	○	○	○
D	○	○	○*	○*	○	○

功能敘述

- 當除算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 Sa 除以 Sb 所得之商存到 D 去，同時若商爲 0，則 FO0 設爲 1，若除數 Sb=0 則錯誤旗號 FO1 設爲 1 且本指令不執行。
- 本指令爲 48 位元運算，所以 Sa，Sb，D 皆佔用連續三個暫存器。

程式範例

48 位元除法例

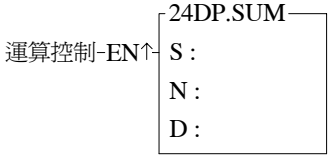
階梯圖	按鍵操作	簡碼指令
		ORG X 0 FUN 23P Sa： R 0 Sb： R 3 D： R 6

		R2	R1	R0
被除數 Sa		2147483647		
÷	除數 Sb	R5	R4	R3
		1234567		

R8	R7	R6
1739		

商

FUN24 **DP** SUM	總和計算 （SUM）	FUN24 **DP** SUM



S ：來源暫存器之起頭號碼

N：欲總和之暫存器個數
（由 S 開始連續 N 個）

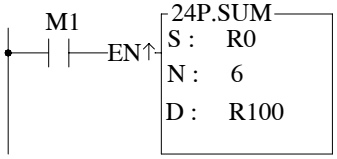
D：存放結果（總和）之暫存器號碼

S，N，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	1 | 511	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當運算控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 S 開始之連續 N 個 16 位元或 32 位元（**D**指令）暫存器作加法運算，得出總和，並將結果存入 D 所指定之暫存器。
- 當 N 之值爲 0 或大於 511 時，運算不執行。
- 通訊埠 1 或通訊埠 2 用來當作泛用 ASCII 通訊介面，如欲通訊對象之資料錯誤查核方式爲總和（Check-Sum）查核，則可使用此指令來產生總和值或利用此指令計算總和值並比對看是否資料有誤。

〈範例 1〉M1 由 OFF→ON 時，計算 16 位元總和

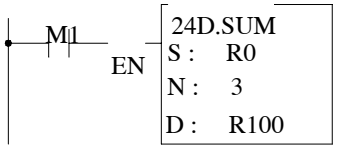


- 左圖範例係將 R0 開始之 6 個暫存器以 16 位元方式計算總和值，並將結果存入 R100 暫存器。

R0=0030H
R1=0031H
R2=0032H
R3=0033H
R4=0034H
R5=0035H

➔ R100=012FH

〈範例 2〉M1 ON 時，計算 32 位元總和

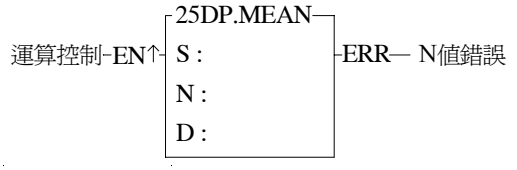


- 左圖範例係將 DR0 開始，以 32 位元方式計算總和值，並將結果存入 DR100（32 位元）暫存器內。

R1，R0=00310030H
R3，R2=00330032H
R5，R4=00410039H

➔ R101，R100=00A5009BH

FUN25 DP MEAN	取平均值 （MEAN）	FUN25 DP MEAN



S ：來源暫存器之起頭號碼

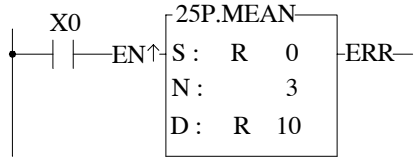
N ：欲平均之暫存器個數
（由 S 開始連續 N 個）

D ：存放結果（平均值）之暫存器號碼

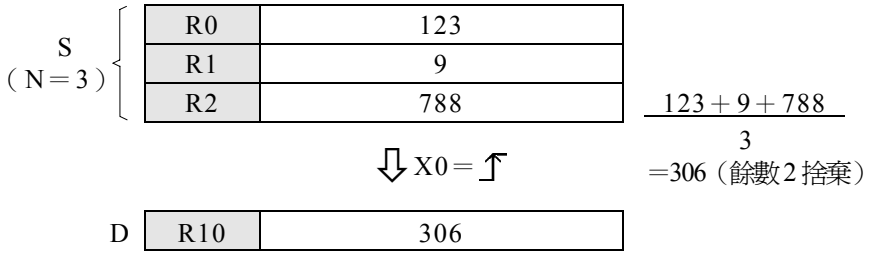
S，N，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V 、 Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

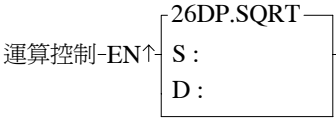
- 當運算控制〝EN〞＝1 或〝EN↑〞（ P指令）由 0→1 時，將 S 開始之連續 N 個 16 位元或 32 位元（ D指令）之數值相加再除以 N，所得之平均值（捨棄餘數）存入 D 所指定之暫存器。
- 以暫存器內容當 N 值時，若暫存器內容值非 2～256，則 N 值錯誤〝ERR〞設爲 1，且本指令不執行。



• 左圖範例爲求取自 R0 開始連續 3 個 16 位元暫存器之平均值，再將結果存於 16 位元暫存器 R10 中。



FUN26 DP SQRT	取平方根值 （SQUARE ROOT）	FUN26 DP SQRT



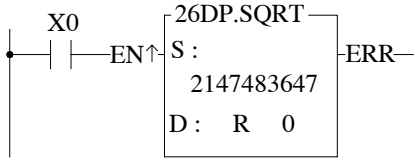
S ：求平方根之來源數值或其暫存器號碼

D：存放結果（平方根值）之暫存器號碼

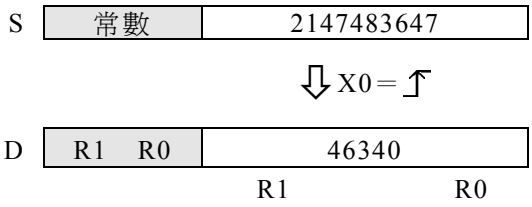
S，D 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16或32位元正數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 值或 S 所指定之暫存器內容值取平方根值（捨棄小數點以後之位數）後存入 D 所指定之暫存器內。
- 當 S 值爲暫存器內容值，而值爲負數，則 S 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例係將常數值 2147483647 取其平方根值，再將結果存到 DR0（R1,R0）去。

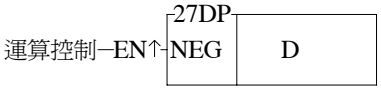


$$\sqrt{2147483647} = 46340.\underline{\underline{95}}$$

↑

小數點以後捨棄

FUN27 DP NEG	取負數 （NEGATION）	FUN27 DP NEG

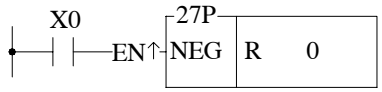


D：取負數之暫存器號碼

D 可結合 V、Z 作間接定址應用

運算元＼範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	V 、 Z
D	○	○	○	○	○	○	○	○*	○*	○	○

- 當運算控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 D 所指定之暫存器內容值取其負數（亦即取其 2 的補數）後存回原暫存器 D。
- 若 D 之內容值原爲負數，取負數之結果將變爲正數。



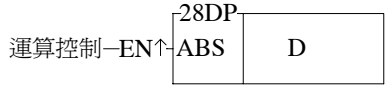
• 左圖程式係將暫存器 R0 之值取負數後再存回 R0 去。

D R0 12345 ☞3039H

⇩X0＝↾

D R0 −12345 ☞CFC7H

FUN28 DP ABS	取絕對值 （ABSOLUTE）	FUN28 DP ABS

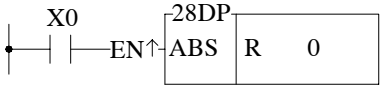


D：取絕對值之暫存器號碼

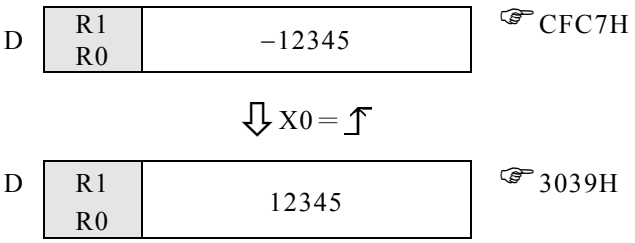
D 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	V 、 Z
D	○	○	○	○	○	○	○	○*	○*	○	○

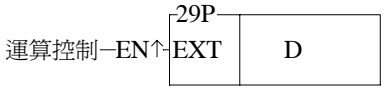
● 當運算控制〝EN〞=1 或〝EN↑〞（ P指令）由 0→1 時，將 D 指定之暫存器內容值取絕對值後寫回原暫存器 D。



• 左圖程式例係將暫存器 DR0 之值取其絕對值後再存回 DR0（R1,R0）去。



FUN29 P EXT	暫存器正負符號擴展 （SIGN EXTENTION）	FUN29 P EXT

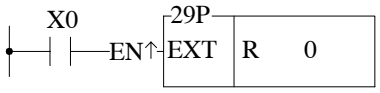


D：欲擴展正負符號之暫存器號碼

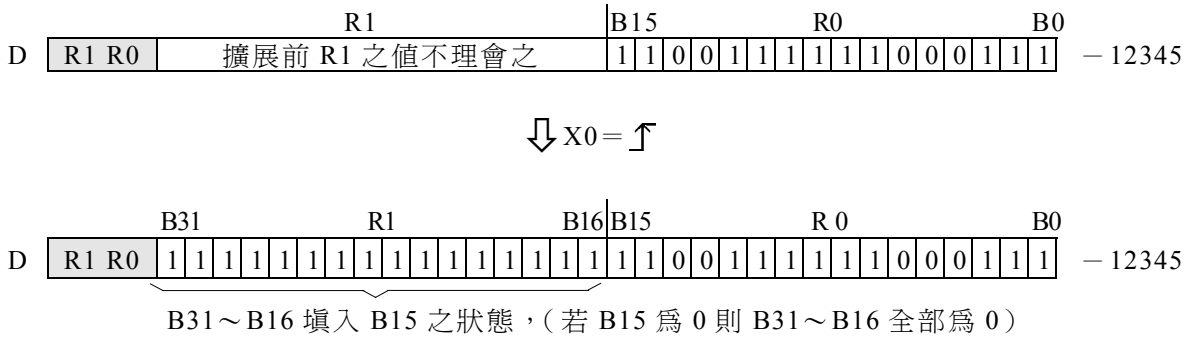
D 可結合 V、Z 作間接定址應用

運算元＼範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	V、Z
D	○	○	○	○	○	○	○	○*	○*	○	○

- 當運算控制〝EN〞=1 或〝EN↑〞（ P指令）由 0→1 時，將 D 所指定之數值，存入由 D+1 和 D 兩個連續 Word 組成之 32 位元暫存器。（兩者值相同只是原來爲 16 位元所表示之數值，而擴展後變成由 32 位元所表示之數值）。
- 本指令是將 16 位元之暫存器數值擴展爲等值之 32 位元暫存器數值（例如將 33FFH 變成 000033FFH），其功用主要在於將 16 位元數值和 32 位元數值作各種運算（＋，－，＊，／，CMP……）時，使用者之資料長度（表示位元）一致，才能進行上述之各種運算。

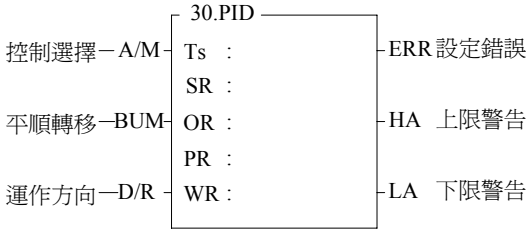


- 左圖程式例係將 16 位元之數值 R0 擴展爲等值之 32 位元數值後存到由 R0 本身和其左邊（高位）相鄰暫存器（R1）所構成之 32 位元暫存器（DR0=R1R0）去。



擴展前（16 位元） R0= CFC7H=－12345
擴展後（32 位元）R1R0=FFFFCFC7H=－12345
} 兩者實際數值相同

FUN30 PID	PID 運算便利指令	FUN30 PID



Ts ：PID 運算間隔時間

SR ：程控設定值起始暫存器號碼，共佔用 8 個暫存器

OR ：PID 輸出暫存器號碼

PR ：參數設定值起始暫存器號碼，共佔用 7 個暫存器

WR：本指令所需使用之工作暫存器起始號碼，共佔用 5 個暫存器，其它地方不可重覆使用。

運算元＼範圍	HR R0 ｜ R3839	ROR R5000 ｜ R8071	DR D0 ｜ D3071	K
Ts	○	○	○	1～3000
SR	○	○*	○	
OR	○	○*	○	
PR	○	○*	○	
WR	○	○*	○	

- PID 指令(FUN30)係將目前所量測之外界類比輸入值當作程控變數（Process Variable，簡稱 PV），將使用者所設定之設定值（Setpoint，簡稱 SP）與程控變數經由軟體 PID 數學式運算後，得到適宜之輸出控制值經由 D/A 類比輸出模組或再處理經由其它界面以控制受控程序在使用者所期望之設定範圍內。
- 數位化 PID 運算式如下：

$$M_n = [(1000/Pb)\times E_n] + \sum_{0}^{n} [(1000/Pb)\times T_i \times T_s \times E_n] - [(1000/Pb)\times T_d \times (PV_n - PV_{n-1})/T_s] + Bias$$

M_n = ： 〝n〞時之控制輸出量

Pb ： 比例帶（範圍：2~5000，單位爲 0.1%；K_c（增益）=1000/ Pb）

T_i ： 積分時間常數（範圍：0~9999，相當於 0.00~99.99 Repeats/Minute）

T_d ： 微分時間常數（範圍：0~9999，相當於 0.00~99.99 Minutes）

PV_n ： 〝n〞時之程控變數值

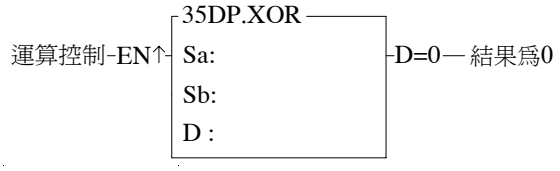
PV_{n-1} ： 〝n〞之上一次之程控變數值

E_n ： 〝n〞時之誤差=設定值（SP）－〝n〞時之程控變數值（PV_n）

T_s ： PID 運算之間隔時間（範圍：1~3000，單位：0.01S）

Bias ： 偏置輸出量（範圍：0~4095）

FUN35 DP XOR	邏輯互斥或（XOR）運算 （EXCLUSIVE OR）	FUN35 DP XOR



Sa：XOR 資料 a 或其暫存器號碼

Sb：XOR 資料 b 或其暫存器號碼

D ：存放 XOR 結果之暫存器號碼

Sa，Sb，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16或32位元 正、負數	V 、 Z
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當運算控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 Sa 和 Sb 資料作邏輯 XOR（Exclusive OR）運算，亦即將 Sa 和 Sb 之各對應位元（B0～B15 或 B0～B31）作比較，任一對應位元之狀態若不相同，則在 D 之該對應位元設爲 1，相同則爲 0。
- 若運算結果 D 之所有位元均爲 0，則結果爲 0 旗號〝D=0〞設爲 1。

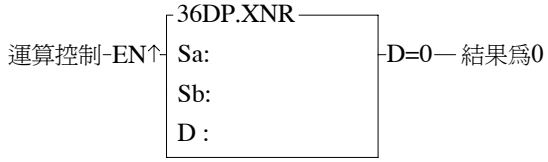
• 左圖程式例係將暫存器 R0 和 R1 作邏輯互斥或運算後將結果存到 R2 去

Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩ X0＝⏉

D	R2	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1

FUN36 DP XNR	邏輯互斥容（XNR）運算 （ENCLUSIVE OR）	FUN36 DP XNR



Sa ：XNR 資料 a 或其暫存器號碼

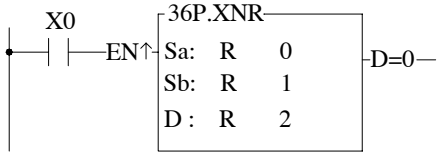
Sb ：XNR 資料 b 或其暫存器號碼

D ：存放 XNR 結果之暫存器號碼

Sa，Sb，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16或32位元 正、負數	V 、 Z
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當運算控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 Sa 和 Sb 資料作邏輯 XNR（Enclusive OR，即 Exclusive OR 之相反），亦即將 Sa 和 Sb 之各對應位元（B0～B15 或 B0～B31）作比較，任一對應位元之狀態相同，則 D 之該對應位元設爲 1，若不同則設爲 0。
- 若運算結果 D 之所有位元均爲 0，則結果爲 0 旗號〝D=0〞設爲 1。



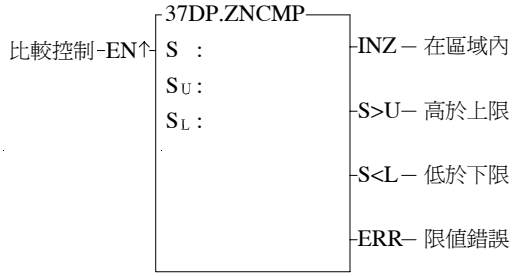
- 左圖程式範例係將暫存器 R0 和 R1 作邏輯互斥容運算後，再將所得結果存到暫存器 R2 去

Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩ X0 = ⇂

D	R2	1	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0

FUN37 DP ZNCMP	區域比較 （ZONE COMPARE）	FUN37 DP ZNCMP



S ：存放比較資料之暫存器號碼

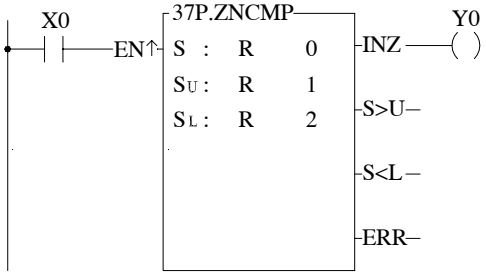
SU：區域上限值或上限值暫存器號碼

SL：區域下限值或下限值暫存器號碼

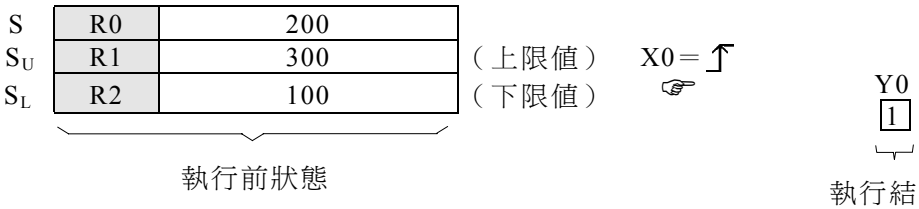
S，SU，SL 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16或32位元正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
S_U	○	○	○	○	○	○	○	○	○	○	○	○	○	○
S_L	○	○	○	○	○	○	○	○	○	○	○	○	○	○

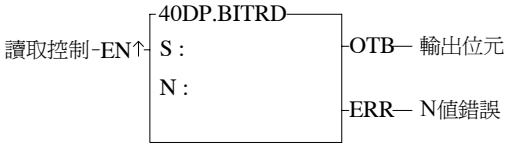
- 當比較控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，執行 S 與上限 SU 及下限 SL 之比較，若 S 介於上限值與下限值之間（SL≦S≦SU），則在區域內旗號〝INZ〞設爲 1，若 S 之值大於上限 SU，則高於上限旗號〝S＞U〞設爲 1，若 S 之值小於下限 SL，則低於下限旗號〝S＜L〞設爲 1。
- 上限 SU 應大於下限 SL，若 SU＜SL，則限值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例係將 R0 之值和由 R1 和 R2 所構成之上、下限區域作比較，設 R0～R2 之數值如下圖左，則可獲得如下圖右之執行結果。
- 若輸出結果需要爲不在區域內，則可用 OUT NOT Y0 即可。



FUN40 DP BITRD	位元資料讀取 （BIT READ）	FUN40 DP BITRD



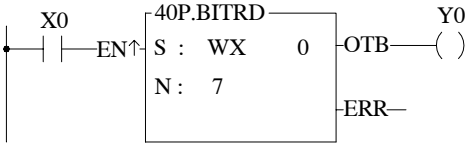
S ：欲讀取位元之資料或其暫存器號碼

N：指定 S 資料中第 N 個位元資料被讀出

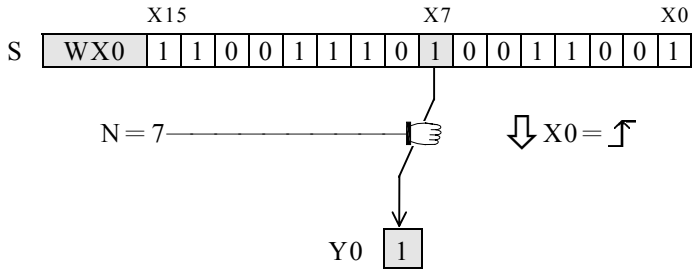
S，N 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16或32位元 正、負數	V 、 Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

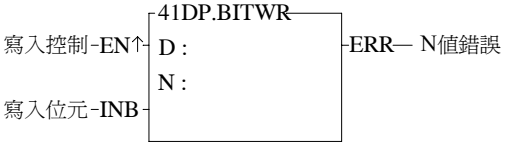
- 當讀取控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 資料中之第 N 個位元（BIT）取出送到輸出位元〝OTB〞去。
- 當讀取控制〝EN〞或〝EN↑〞（P指令）=0 時，輸出位元〝OTB〞之狀態保持上次執行之結果（M1919=0）或清除爲 0（M1919=1）。
- N 之值在 16 位元指令時有效範圍爲 0～15，在 32 位元（D指令）時則爲 0～31，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例係自 WX0（X0～X15）中讀取第 7 個位元（X7）之狀態，再將之送到 Y0 去，其結果如下：



FUN41 **D P** BITWR	位元資料寫入 (BIT WRITE)	FUN41 **D P** BITWR



D：欲寫入位元之暫存器號碼

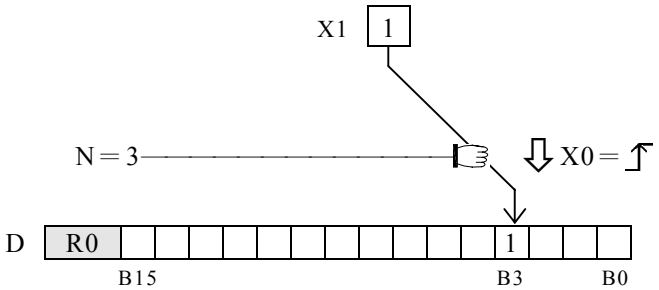
N：指定將寫入位元 INB 之狀態寫入 D 中第 N 個位元處

D，N 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	0 0 | 或 | 15 31	V 、 Z
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

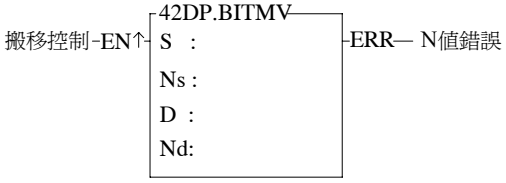
- 當寫入控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將寫入位元（INB）之狀態寫入 D 中 N 所指定之位元去。
- N 之值在 16 位元指令時有效範圍爲 0～15，在 32 位元（**D**指令）時則爲 0～31，若超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。

• 左圖程式範例係將寫入位元 INB 之狀態寫到 R0 中之 B3 去，假設 X=1，其執行結果如下圖。



B3 以外之位元保持不變

FUN42 DP BITMV	位元資料搬移 （BIT MOVE）	FUN42 DP BITMV



S ：搬移之來源資料或其暫存器號碼

Ns：指定 S 中之 Ns 位元爲來源位元

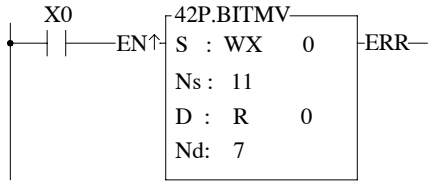
D ：搬移之目的暫存器號碼

Nd：指定 D 中之 Nd 位元爲目的位元

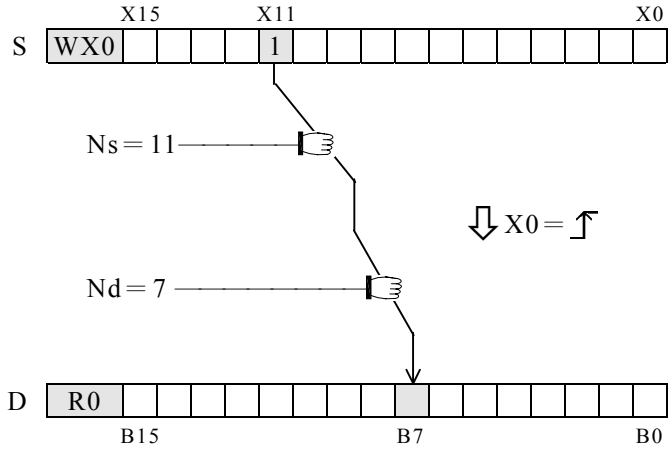
S，Ns，D，Nd 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16或32位元正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

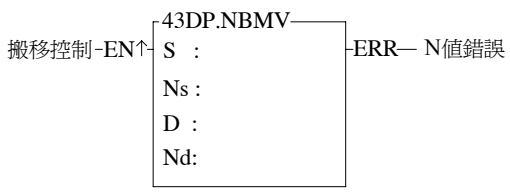
- 當搬移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 S 中 Ns 所指定之位元狀態搬移至 D 中 Nd 所指定之位元去。
- Ns 或 Nd 之值在 16 位元指令時有效範圍爲 0～15，在 32 位元（**D**指令）時則爲 0～31，超出此範圍，則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例係將 S 中之 B11（即 X11）之狀態搬移到 D 中 B7 之位置去，D 中除被寫入之位元 B7 外，其他位元狀態不變。



FUN43 DP NBMV	位數（NIBBLE）搬移 （NIBBLE MOVE）	FUN43 DP NBMV



S ：搬移之來源資料或其暫存器號碼

Ns：指定 S 中之第 Ns 個位數爲來源位數

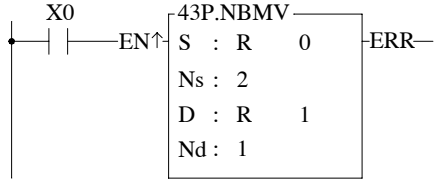
D ：搬移之目的暫存器號碼

Nd：指定 D 中之第 Nd 個位數爲目的位數

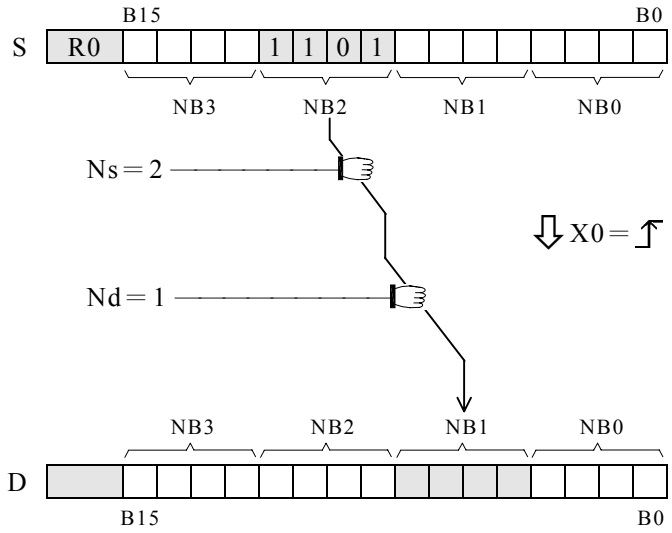
S，Ns，D，Nd 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16或32位元正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○

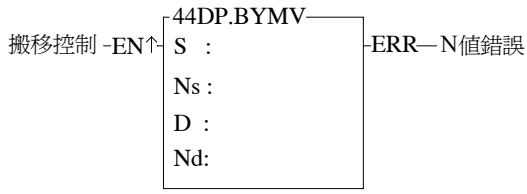
- 當搬移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 S 中第 Ns 個位數（Nibble：爲 4 個位元所組合。由暫存器之最低位元 B0 起每連續 4 個位元形成一個 Nibble，即 B0～B3 爲第 0 個位數，B4～B7 爲第 1 個位數，……）搬移到 D 中 Nd 所指定的那個位數去。
- Ns 或 Nd 在 16 位元指令時，有效範圍爲 0～3，在 32 位元（**D**指令）時則爲 0～7，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例係將 S 中之第 2 個位數 NB2(即 B8～B11）搬到 D 中之第 1 個位數 NB1（即 B4～B7）去，D 中之其他位數則保持不變。



FUN44 DP BYMV	位元組（BYTE）搬移 （BYTE MOVE）	FUN44 DP BYMV



S ：搬移之來源資料或其暫存器號碼

Ns：指定 S 中之第 Ns 個位元組爲來源位元組

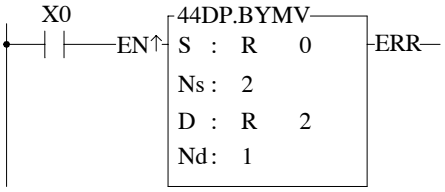
D ：搬移之目的暫存器號碼

Nd：指定 D 中之第 Nd 個位元組爲目的位元組

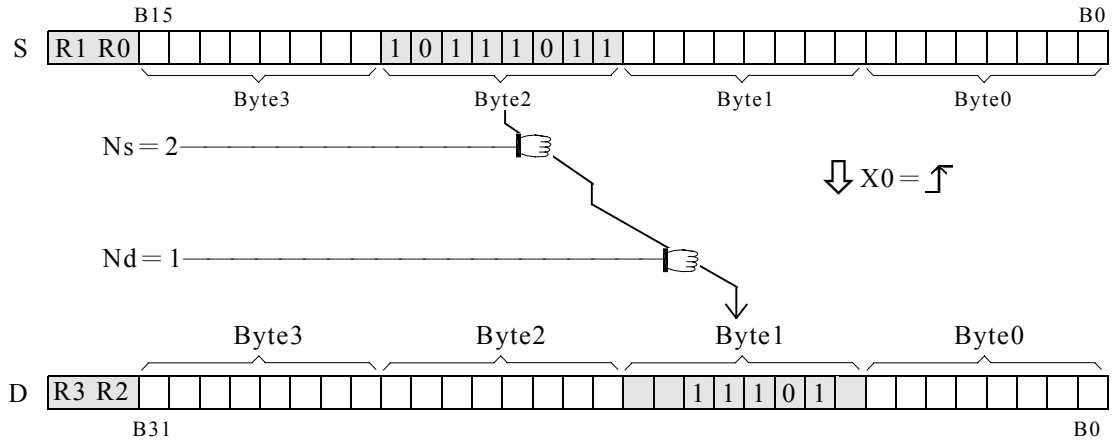
S，Ns，D，Nd 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16或32位元正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○

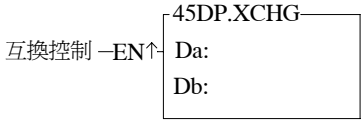
- 當搬移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 S 中第 Ns 個位元組（Byte：爲 8 個位元組成，由暫存器之最低位元 B0 起每連續 8 個位元形成一個 Byte，即 B0～B7 爲第 0 個位元組，B8～B15 爲第 1 個位元組……）搬移到 D 中第 Nd 個位元組處。
- Ns 或 Nd 在 16 位元指令時有效範圍爲 0～1，在 32 位元（**D**指令）時則爲 0～3，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例係將 S（由 R1R0 所構成之 32 位元暫存器）中之第 2 個位元組（即 B16～B23）搬移到 D（由 R3R2 構成之 32 位元暫存器）中之第 1 個位元組去，D 中之其他位元組則保持不變。



FUN45 **D P** XCHG	資料互換 （EXCHANGE）	FUN45 **D P** XCHG



Da：互換之暫存器 A 之號碼

Db：互換之暫存器 B 之號碼

Da，Db 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	V、Z
Da	○	○	○	○	○	○	○	○*	○*	○	○
Db	○	○	○	○	○	○	○	○*	○*	○	○

● 當互換控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 16 位元或 32 位元（**D**指令）之暫存器 Da 和暫存器 Db 之內容互換。

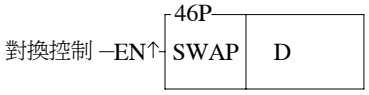
• 左圖程式範例係將 16 位元暫存器 R0 和 R1 之資料內容互換。

		B15															B0
Da	R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Db	R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

⇩X0＝⤒

		B15															B0
Da	R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Db	R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FUN46 P SWAP	位元組（BYTE）資料對換 （BYTE SWAP）	FUN46 P SWAP

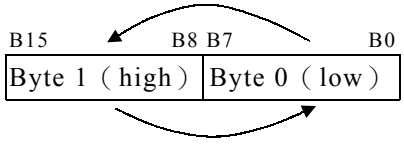


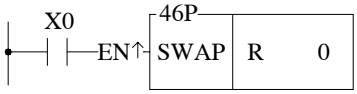
D：執行位元組資料對換之暫存器號碼

D 可結合 V、Z 作間接定址應用

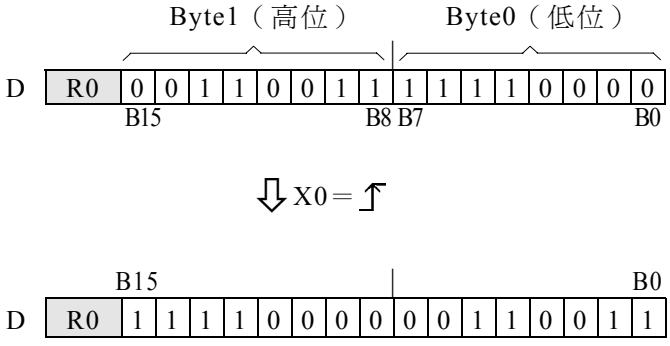
運算元 \ 範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	V 、 Z
D	○	○	○	○	○	○	○	○*	○*	○	○

● 當對換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 D 所指定之 16 位元暫存器之低位元組 Byte 0（B0～B7）和高位元組 Byte 1（B8～B15）之資料對換。

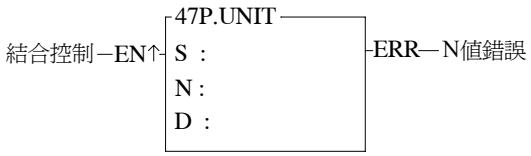




• 左圖程式係將 R0 之低位元組（B0～B7）和高位元組（B8～B15）之資料互換，其結果如下。



FUN47 P UNIT	位數（NIBBLE）資料結合 （NIBBLE UNITE）	FUN47 P UNIT



S：欲被結合之來源暫存器起頭號碼

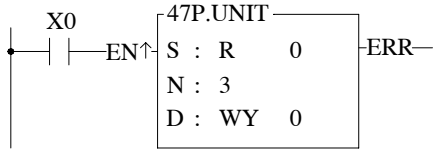
N：欲結合之位數

D：存放結合資料之暫存器號碼

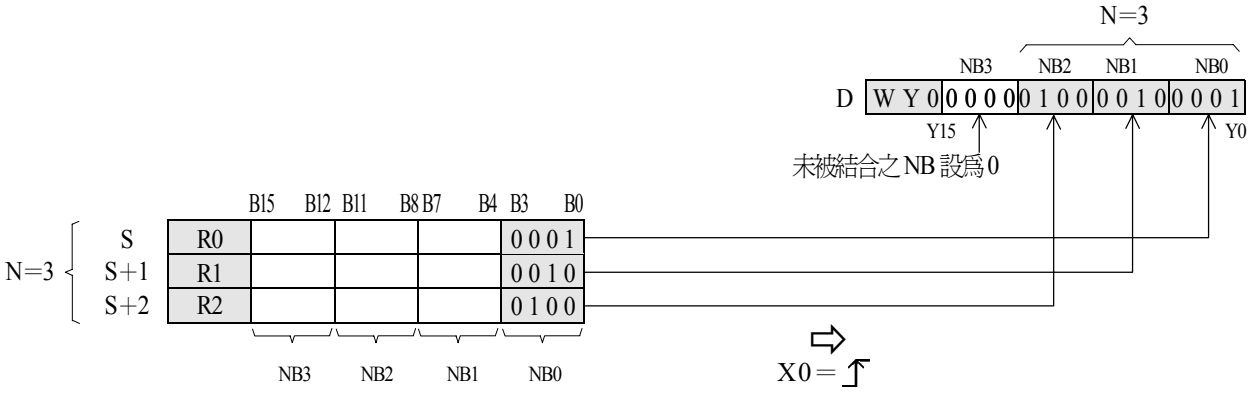
S，N，D 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	1 ｜ 4	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

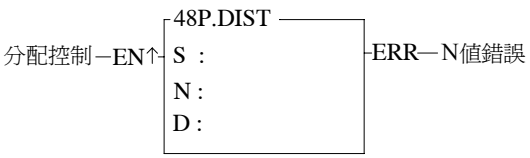
- 當結合控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，取出由 S 開始之連續 N 個暫存器之最低位數 NB0（Nibble：爲 4 位元所組成，由暫存器之最低位元 B0 起往左每連續 4 個位元構成一位數，即 B0～B3 爲第 0 個位數 NB0，B4～B7 爲第 1 個位數 NB1，……）並將之由低位往高位之順序依序填到 D 中之 NB0，NB1，……NBn-1，D 中未被填入之位數則填入 0。
- 本指令只提供 WORD（16 位元）指令，因此最多只有 4 個 Nibble，故 N 之有效範圍爲 1～4，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例係將 R0，R1 和 R2 三個暫存器之 NB0 取出填到 WY0 暫存器中之 NB0～NB2 去。



FUN48 P DIST	位數（NIBBLE）資料分配 （NIBBLE DISTRIBUTE）	FUN48 P DIST



S：分配之來源資料或暫存器號碼

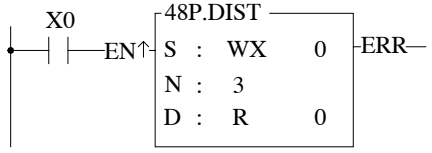
N：欲分配之位數的數目

D：存放分配資料之暫存器起頭號碼

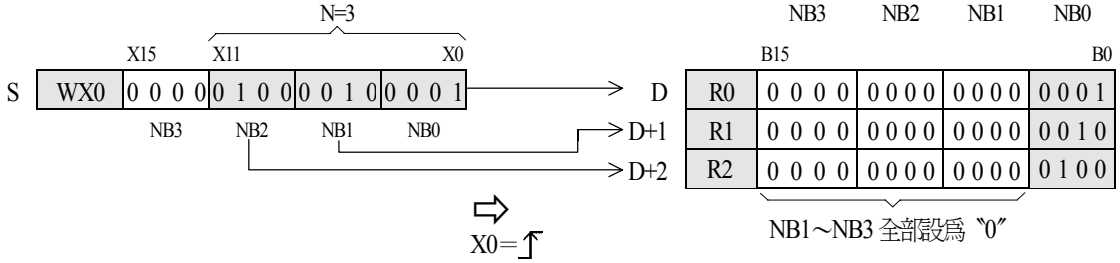
S，N，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16位元正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~4	○
D		○	○	○	○	○	○		○	○*	○*	○		○

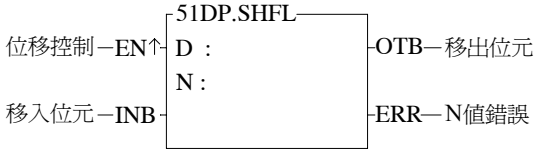
- 當分配控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 中自最低位數 NB0 開始之連續 N 個位數（位數：Nibble，係由 4 個位元所組成，由一暫存器之最低位元 B0 開始往左，每連續 4 個位元構成一位數，即 B0～B3 爲第 0 個位數 NB0，B4～B7 爲第 1 個位數，……），由低至高依序分配填至由 D 開始之 N 個暫存器之第 0 個位數 NB0 去。D 中各暫存器之 NB0 以外之位數則均填入 0。
- 本指令只提供 WORD（16 位元）指令，故最多只有 4 個 Nibble，所以 N 之有效值爲 1～4，超出此範圍，則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例係將 WX0 暫存器之 NB0～NB2 填寫到 R0～R2 三個連續暫存器之 NB0 去。



FUN51 DP SHFL	向左位移 （SHIFT LEFT）	FUN51 DP SHFL



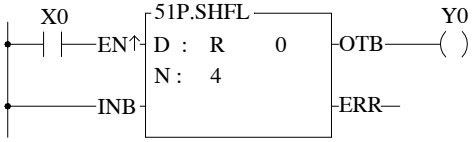
D：被位移之暫存器號碼墟

N：位移之位元數

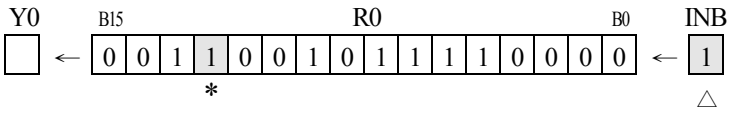
D，N 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	1 | 或 | 16 32	V、Z
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

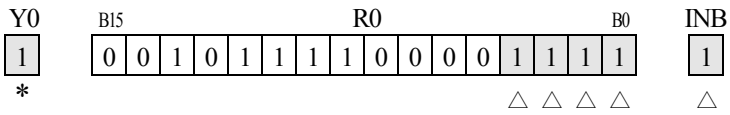
- 當位移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 D 暫存器之資料向左（由低位往高位）連續移動 N 個位元，在最低位元 B0 左移後，其空位以移入位元 INB 填補之，同時將移出位元 B15 或 B31（**D**指令）之狀態送至移出位元〝OTB〞去。
- N 之有效範圍在 16 位元指令爲 1～16，在 32 位元（**D**指令）則爲 1～32，若超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



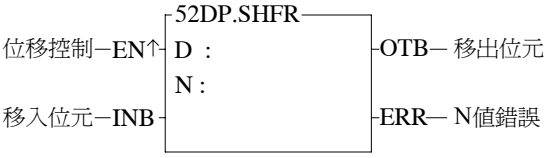
• 左圖程式範例係將暫存器 R0 之資料連續向左位移 4 個位元(4 次)，下圖爲其執行結果。



⇩X0＝⇑



FUN52 **D** **P** SHFR	向右位移 （SHIFT RIGHT）	FUN52 **D** **P** SHFR



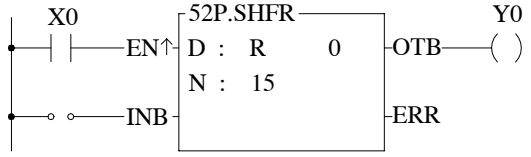
D：被位移之暫存器號碼

N：位移之位元數

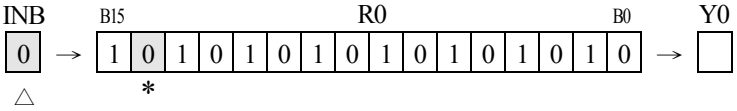
D，N 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	1 | 或 | 16 32	V、Z
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

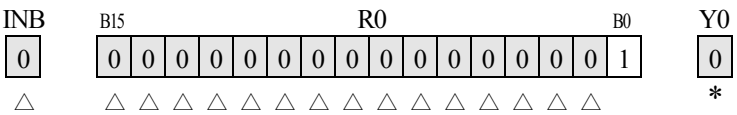
- 當位移控制〞EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 D 暫存器之資料向右（由高位往低位）連續移動 N 個位元，在最高位元 B15 或 B31（**D**指令）右移後，其空位由移入位元 INB 填補之，同時將移出位元 B0 之狀態送至移出位元〝OTB〞去。
- N 之有效範圍在 16 位元指令爲 1～16，在 32 位元（**D**指令）則爲 1～32，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



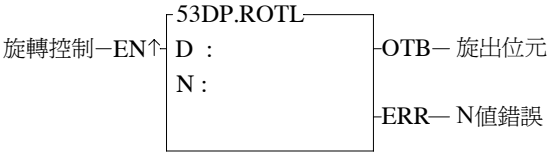
- 左圖程式範例係將暫存器 R0 之資料連續向右位移 15 個位元（15 次），下圖爲其執行結果。



⇩X0＝⇂



FUN53 DP ROTL	向左旋轉 （ROTATE LEFT）	FUN53 DP ROTL



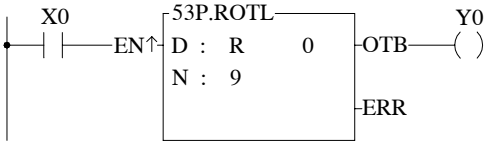
D：被旋轉之暫存器號碼

N：旋轉之位元數

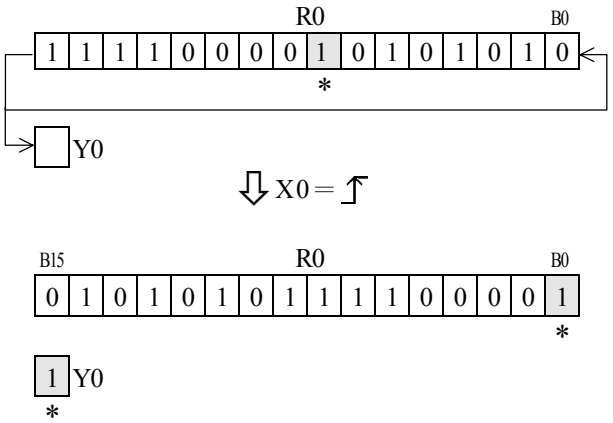
D，N 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	1 1 | 或 | 16 32	V、Z
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

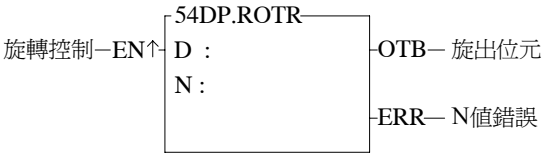
- 當旋轉控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 D 暫存器之資料向左（低位往高位，即 16 位元指令 B0→B1，B1→B2，……，B14→B15，B15→B0。32 位元指令則爲 B0→B1，B1→B2，……，B30→B31，B31→B0）連續旋轉 N 位元，同時並將旋出之 B15 或 B31（**D**指令）位元狀態送到旋出位元〝OTB〞去。
- N 之有效值在 16 位元指令爲 1～16，在 32 位元（**D**指令）則爲 1～32，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例係將暫存器 R0 之資料連續向左旋轉 9 次，下圖爲其執行結果。



FUN54 **D** **P** ROTR	向右旋轉 （ROTATE RIGHT）	FUN54 **D** **P** ROTR



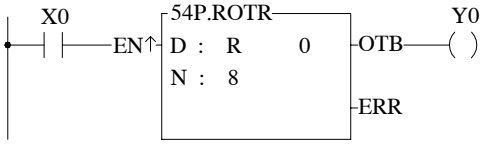
D：被旋轉之暫存器號碼

N：旋轉之位元數

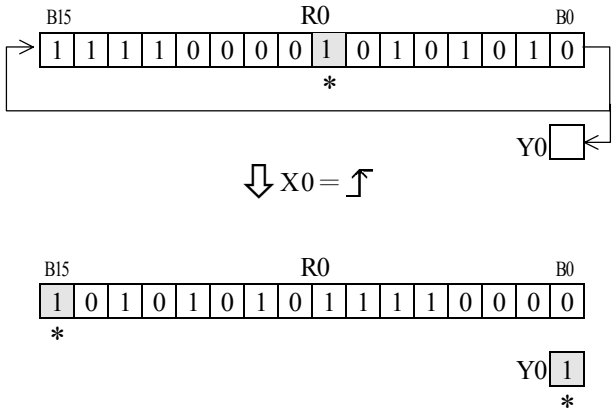
D，N 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	1 | 或 | 16 1 | 32	V、Z
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

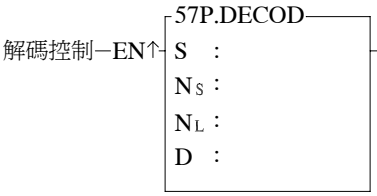
- 當旋轉控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 D 暫存器之位元資料向右（高位往低位，即 16 位元指令 B15→B14，B14→B13，……B1→B0，B0→B15。32 位元指令則爲 B31→B30，B30→B29，……B1→B0，B0→B31）連續旋轉 N 位元，同時並將旋出之 B0 位元狀態送到旋出位元〝OTB〞去。
- N 之有效值在 16 位元指令爲 1～16，在 32 位元（**D**指令）則爲 1～32，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例係將暫存器 R0 之資料連續向右旋轉 8 次，下圖爲其執行結果。



FUN57 P DECOD	解碼 （DECODE）	FUN57 P DECOD

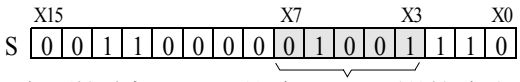


S ：解碼之來源資料暫存器號碼（16 位元）
NS：S 中欲被解碼之起始位元
NL：解碼值之長度（1～8 位元）
D ：存放解碼結果之暫存器起頭號碼
（2～256 點=1～16 Words）
S，NS，NL、D 可結合 V、Z 作間接定址應用

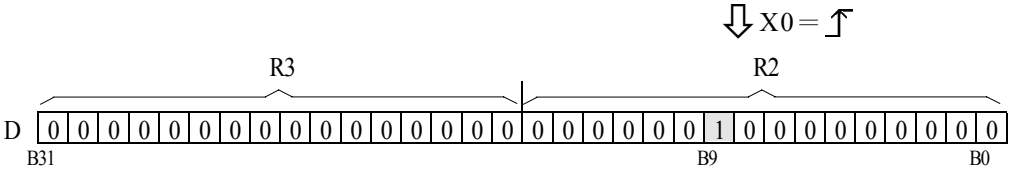
運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16位元 正、負數	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NS	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
NL	○	○	○	○	○	○	○	○	○	○	○	○	1～8	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 本指令所謂之解碼係在寬度爲 2^{N_L} 個單點（D）中，將 S 中位元 B_{N_S}～$B_{N_S+N_L-1}$（稱之爲解碼值，而 B_{N_S} 爲解碼值之起始位元，$B_{N_S+N_L-1}$ 則爲其終止位元）所指定的那個單點設爲 1，其他設爲 0。
- 當解碼控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 中 Ns 所指定之位元開始，往左（高位元方向）連續 NL 個位元資料（即 B_{N_S}～$B_{N_S+B_{N_L-1}}$）取出當作解碼值，並將解碼結果 D 之 2^{N_L} 個單點中，解碼值所指定的那個單點設爲 1，而其他單點全部設爲 0。
- 本指令只有 16 位元指令，S 只有 B0～B15，故 Ns 有效範圍爲 0～15，而解碼值長度 NL 限制爲 1～8 位元。故解碼結果 D 之寬度爲 $2^{1\sim8}$ 個點=2～256 點=1～16 Words（未滿 16 點仍佔 1 個 Word），若 Ns 或 NL 值超出上述範圍則範圍錯誤旗號〝ERR〞設爲 1，且本指令不執行。
- 若終止位元超出 S 之 B15，則往 S+1 之 B0 延伸。但終止位元不得超過該種類運算元之最高極限（各單點運算元之最後一點或各暫存器運算元之最後一個 Word 的 B15），若超出，則本指令只取起始位元 B_{N_S} 至其最高極限間之位元當解碼值。

• 左圖程式範例係自暫存器 WX0 中 X3 至 X7 連續 5 個位元的資料取出解碼後，將結果存到 R2 開始之 32 位元暫存器中。

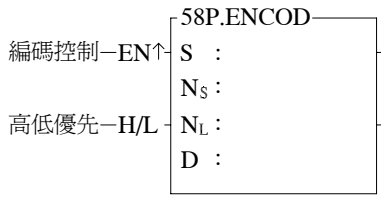


解碼值長度 NL =5，故爲 X3～X7（其值爲 9）



因 NL=5，故 D 之寬度爲 2^5=32 點=2 個 WORD，即 D 爲 R3R2 合成之 32 點寬度，而解碼值爲 01001=9，故 D 中之 B9（第 10 點）爲 1，其他點均爲 0。

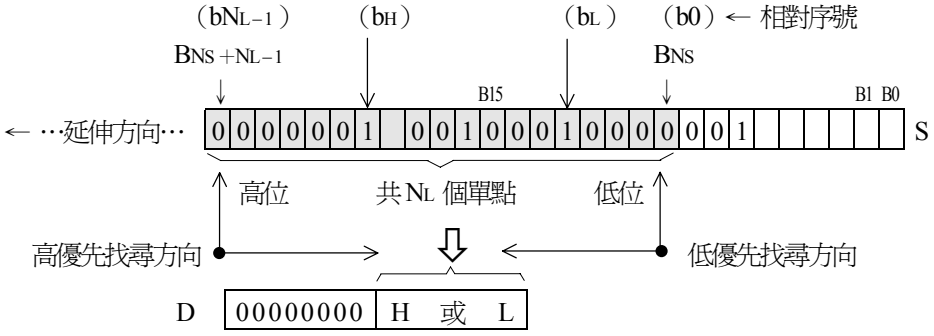
FUN58 P ENCOD	編碼 （ENCODE）	FUN58 P ENCOD



S ：被編碼之暫存器起頭號碼
NS：指定 S 中之一點爲編碼起始點
NL：編碼之單點數目（2～256 點）
D ：存放編碼結果之暫存器號碼
（1 個 Word）
S，NS，NL，D 可結合 V、Z 作間接定址應用

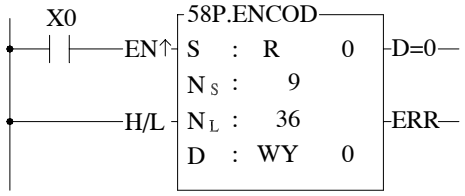
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16位元 正、負數	V 、 Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
NS	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
NL	○	○	○	○	○	○	○	○	○	○	○	○	2～256	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當編碼控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 中 NS 所指定之單點開始往左（高位方向）之連續 NL 個單點 BNS～BNS+NL-1（BNS 稱爲編碼起始點，其相對序號爲 b0，BNS+NL-1 則稱爲編碼終止點，相對序號爲 bNL-1）取出，由左往右作高優先（H／L=1 時）或由右往左作低優先（H／L=0 時）編碼（亦即找出第一個狀態爲 1 之單點，該單點之相對序號值即爲編碼值），再將編碼值存到編碼結果暫存器 D 之低位元組（B0～B7），而 D 之高位元組則填 0。

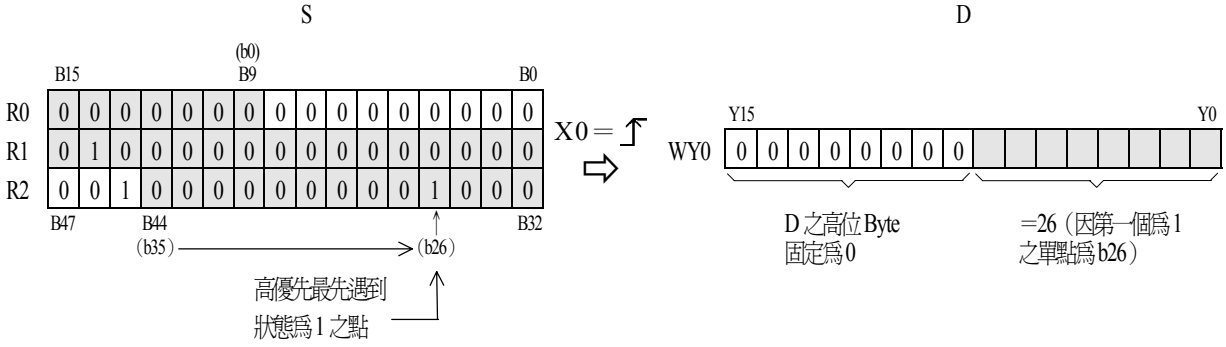


- 如上示意圖範例，若爲高優先編碼，將先找到 bH（值爲 12）；若爲低優先編碼則會先找到 bL（值爲 4）。在 NL 個單點中至少要有一個狀態爲 1。若全爲 0 則本指令不執行，同時將全部爲 0 旗號〝D=0〞設爲 1。
- 因 S 爲一 16 位元暫存器，故 NS 可爲 0～15，用以指定 S 中 B0～B15 之一點爲編碼起始點（b0）。而 NL 值可爲 2～256，是用以界定編碼終止點，即指定自起始點（b0）開始往左（高位元方向）連 NL 個單點爲編碼區域（即 b0～bNL-1）。NS 或 NL 值若超出上述範圍則本指令不執行，並將範圍錯誤旗號〝ERR〞設爲 1。
- 若編碼終止點（bNL-1）超出 S 之 B15，則繼續往 S+1，S+2，……延伸，但最大不能超過該種類運算元之最高極限（各單點運算元之最後一點或各暫存器運算元之最後一個 Word 的 B15），若超出則本指令只取 b0 至其最高極限間之單點當作編碼範圍。

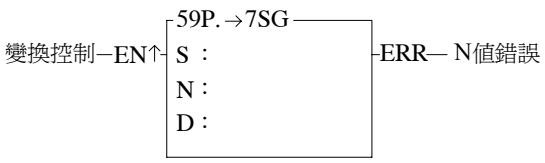
FUN58 P ENCOD	編碼 （ENCODE）	FUN58 P ENCOD



- 左圖程式例爲高優先編碼之範例，當 X0 由 0 →1 時，將 S（R0）中 Ns 所指之點 B9（b0）開始往左連續 36 個單點取出作高優先編碼（因 H／L=1），亦即自 b35（編碼終止點）開始往右找尋第一個狀態爲 1 之單點。本例之結果其相對序號爲 b26，故 D 之值爲 001AH=26，如下圖所示。



FUN59 P →7SG	7 段顯示碼變換 （7-SEGMENT CONVERSION）	FUN59 P →7SG



S：變換之來源資料或其暫存器號碼

N：指定 S 資料中連續 N+1 個位數（Nibble）

D：存放 7 段碼結果之起始暫存器號碼

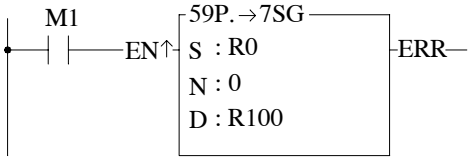
S，N，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16位元 正、負數	V 、 Z
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當變換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 中連續 N+ 1 個位數（Nibble：由連續 4 個位元所組成，即 S 之 B0～B3 爲位數 0，B4～B7 爲位數 1，……）轉換成 7 段顯示碼後，將之存入 D。D 中 7 段碼之擺放順序爲 a 段置於 B6，b 段置於 B5，…… g 段置於 B0，B7 不用而固定爲 0。請參閱 9-29 頁之〝7 段碼與顯示字型表〞。
- 因本指令只限 16 位元，因 S 只有 4 個 Nibble（NB0～NB3），故 N 之有效範圍爲 0～3，超出此範圍則 N 值錯誤旗號〝ERR〞設爲 1，且本指令不執行。
- N=0，代表一位數；N=1，代表二位數；N=2，代表三位數；N=3，代表四位數。
- 當使用永宏 7 段顯示器擴充模組（FB-7SG）且利用 FUN84（7SGMO）便利指令作解碼與非解碼之綜合使用時，可結合 FUN 59 與 FUN 84 兩指令而簡化程式之設計。（參考第 17 章範例說明）。

FUN59 P →7SG	7 段顯示碼變換 （7-SEGMENT CONVERSION）	FUN59 P →7SG

〈範例 1〉M1 由 OFF→ON 時，轉十六進制值爲 7 段顯示碼

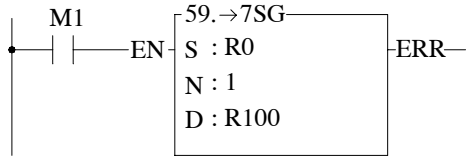


- 左圖範例係將 R0 的第 1 個位數（Nibble）轉換爲 7 段顯示碼並存放至 R100 的低位元組（Low Byte），而 R100 的高位元組（High Byte）保持不變。

原 R100＝0000H

R0＝0001H ➔ R100＝0030H（1）

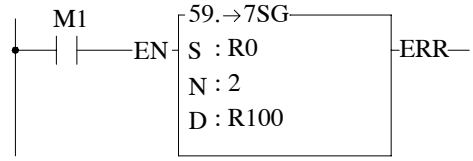
〈範例 2〉M1 ON 時，轉十六進制值爲 7 段顯示碼



- 左圖範例係將 R0 的第 1 和第 2 個位數轉換爲 7 段顯示碼並存放至 R100。
- R100 的低位元組存放第 1 位數。
- R100 的高位元組存放第 2 位數。

R0＝0056H ➔ R100＝5B5FH（56）

〈範例 3〉M1 ON 時，轉十六進制值爲 7 段顯示碼



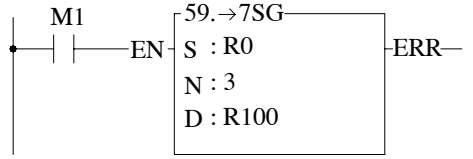
- 左圖範例係將 R0 的第 1 和第 2 和第 3 個位數轉換爲 7 段顯示碼，並存放至 R100 與 R101。
- R100 的低位元組存放第 1 位數。
- R100 的高位元組存放第 2 位數。
- R101 的低位元組存放第 3 位數。
- R101 的高位元組保持不變。

原 R101=0000H

R0=0A48H ➔ R100=337FH（48）

R101=0077H（A）

〈範例 4〉M1 ON 時，轉十六進制值爲 7 段顯示碼

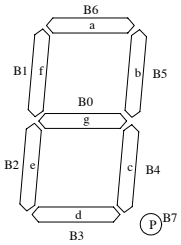


- 左圖範例係將 R0 的第 1～4 位數轉換爲 7 段顯示碼，並存放至 R100 與 R101。
- R100 的低位元組存放第 1 位數。
- R100 的高位元組存放第 2 位數。
- R101 的低位元組存放第 3 位數。
- R101 的高位元組存放第 4 位數。

R0=2790H ➔ R100=7B7EH（90）

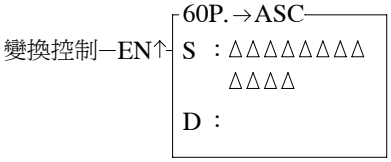
R101=6D72H（27）

FUN59 P →7SG	7 段顯示碼變換 （7-SEGMENT CONVERSION）	FUN59 P →7SG

S 之位數（4 位元）		7 段顯示器結構	D 之位元組（7 段顯示碼）								顯示字形
十六進制	二進制		B7 ●	B6 a	B5 b	B4 c	B3 d	B2 e	B1 f	B0 g	
0	0000		0	1	1	1	1	1	1	0	
1	0001		0	0	1	1	0	0	0	0	
2	0010		0	1	1	0	1	1	0	1	
3	0011		0	1	1	1	1	0	0	1	
4	0100		0	0	1	1	0	0	1	1	
5	0101		0	1	0	1	1	0	1	1	
6	0110		0	1	0	1	1	1	1	1	
7	0111		0	1	1	1	0	0	1	0	
8	1000		0	1	1	1	1	1	1	1	
9	1001		0	1	1	1	1	0	1	1	
A	1010		0	1	1	1	0	1	1	1	
B	1011		0	0	0	1	1	1	1	1	
C	1100		0	1	0	0	1	1	1	0	
D	1101		0	0	1	1	1	1	0	1	
E	1110		0	1	0	0	1	1	1	1	
F	1111		0	1	0	0	0	1	1	1	

7 段碼與顯示字型表

FUN60 P →ASC	ASCII 碼變換 （ASCII CONVERSION）	FUN60 P →ASC

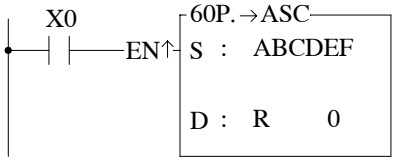


S ：欲變換成 ASCII 碼之文／數字

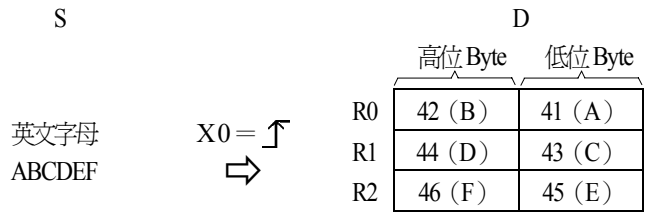
D ：存放 ASCII 碼結果之暫存器起頭號碼

運算元＼範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	文／數字
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	1～12 個英文或數字
S											○
D	○	○	○	○	○	○	○	○*	○*	○	

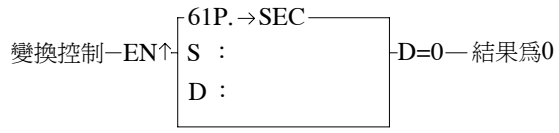
- 當變換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 之文／數字（最多可達 12 個字元）變換爲 ASCII 碼再存入由 D 起頭之暫存器內，每兩個字元將佔用一個 16 位元暫存器。
- 本指令之應用係將文／數字訊息先存於程式中，等某些條件發生時，再將此文／數字訊息變成 ASCII 碼送出給外界能接受 ASCII 碼之顯示裝置顯示之。



• 左圖程式將 ABCDEF 6 個英文字母轉換成 ASCII 碼，再將之存到 R0 開始之連續 3 個暫存器去。



FUN61 P →SEC	時：分：秒→秒 （HOUR：MINUTE：SECOND→SECOND）	FUN61 P →SEC



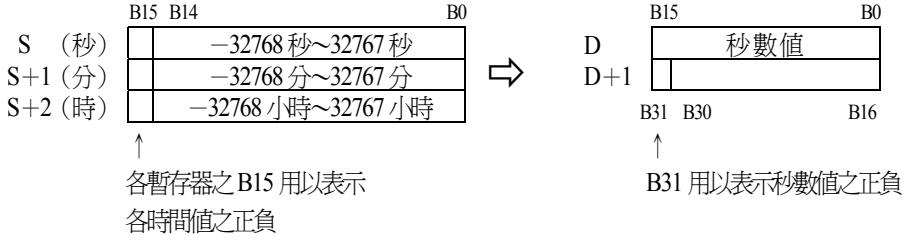
S ：欲變換之時間資料暫存器起頭號碼

D ：存放結果之暫存器起頭號碼

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	−117968399 | 117964799
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	

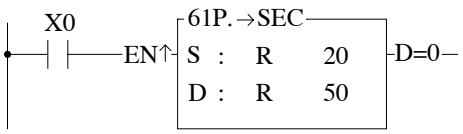
- 當變換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S～S+2 的〔時：分：秒〕資料轉換爲等值之秒數值後存入由 D 和 D+1 合併而成之 32 位元暫存器中。若結果=0，則〝D=0〞旗號設爲 1。

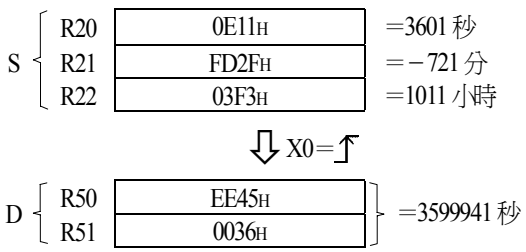
- 永宏 PLC 指令中，和〔時：分：秒〕時間相關之指令（FUN61 和 62），其時間資料之格式如下圖所示將自動合併連續 3 個暫存器（Word）來當時間值使用，其起頭第一個爲秒數（Second）暫存器，次一個爲分數（Minute）暫存器，最後一個則爲時數（Hour）暫存器。每個暫存器之 16 個位元中只有 B14～B0 用以表示時間值，而其最高位元 B15 則用以表示各該時間值之正、負。B15 爲 0 表示該時間爲正，B15 若爲 1 則表示該時間值爲負，B14～B0 之時間值是以二進制表示，當時間值爲負時，B14～B0 則以 2 的補數表示。運算之秒數結果爲〔時：分：秒〕三個暫存器之秒數加減結果。



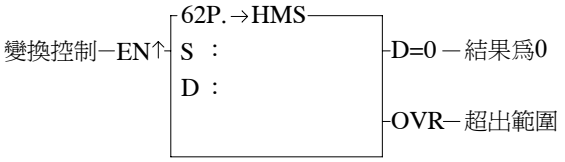
- 任一〔時：分：秒〕時間資料，除了以 FUN61 或 62 兩指令去存取時才會自動合併使用，其他指令均會將之視爲個別之一般暫存器，不會自動合併使用，3 個暫存器間沒有任何關係，因此您可個別對時、分、秒之任一資料運算，結果互不影響。

- 下圖程式例，本指令會將 R20 開始之 3 個資料視爲〔時：分：秒〕資料而將之轉換成等值之秒數值後再存入 R50～R51 所組成之 32 位元暫存器中，其結果如下圖右。





FUN62 P →HMS	秒數→時：分：秒 （SECOND→HOUR：MINUTE：SECOND）	FUN62 P →HMS

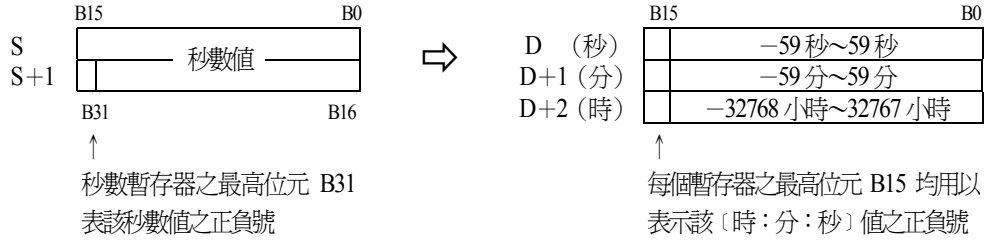


S ：欲變換之秒數資料暫存器起頭號碼

D ：變換結果〔時：分：秒〕存放之暫存器起頭號碼

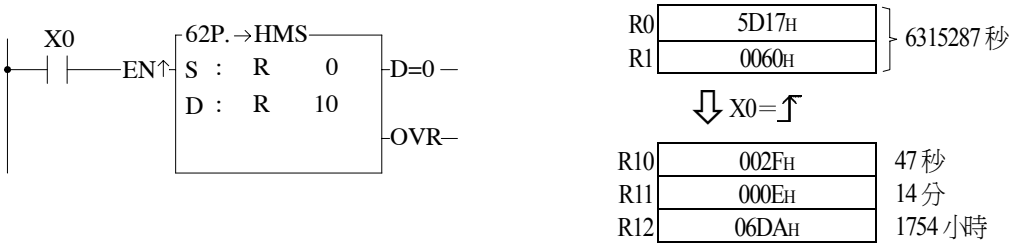
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	−117968399 | 117964799
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	

● 當變換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S～S+1 之 32 位元秒數資料轉換爲等值之〔時：分：秒〕時間值存入 D～D+2 三個連續之暫存器中，本指令所有資料均以二進碼表示（若爲負值則以 2 的補數表示）。

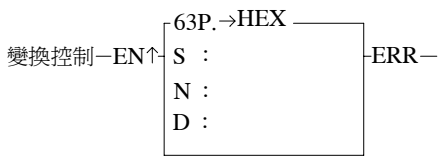


● 如上圖所示本指令轉成〔時：分：秒〕時間後，其〔分：秒〕值只可能爲−59～59，而時數則可爲−32768～32767 小時，因此 D 之最大極限是−32768 小時−59 分−59 秒至 32767 小時 59 分 59 秒，分別對應到 S 之秒數爲−117968399 秒～117964799 秒。若 S 值超出此範圍 D 將放不下，此時本指令便不執行，並將超出範圍旗號〝OVR〞設爲 1。若 S 爲 0 則結果爲零旗號〝D=0〞會設爲 1。

● 下圖程式爲本指令執行之結果範例，注意暫存器內容值均爲二進制值，其右邊爲其等效之 10 進制表示值。



FUN63 P →HEX	ASCII 碼轉換爲十六進制值	FUN63 P →HEX



S ：來源暫存器之起始號碼

N：欲轉 ASCII 碼爲十六進制值之個數

D：存放結果（十六進制值）之暫存器起始號碼

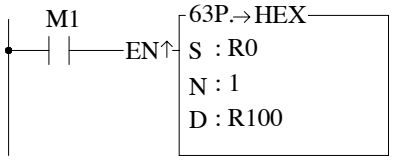
S，N，D 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	正數 16 位元	V 、 Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1～511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當換變控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 開始之連續 N 個 16 位元暫存器（Low Byte 有效）之 ASCII 碼轉換爲十六進制值，並將結果存入由 D 所指定開始之暫存器。每 4 個 ASCII 碼由一個暫存器儲存，未對應到 ASCII 碼之暫存器內容維持原值不變。
- 當 N 之值爲 0 或大於 511 時，運算不執行。
- 當 ASCII 碼錯誤時（非 30H～39H 或 41H～46H），輸出〝ERR〞ON。
- 此指令最大用途係將通訊埠 1 或通訊埠 2 所接收到外界 ASCII 週邊（以 ASCII 碼傳送數值給 PLC）之 ASCII 數碼轉換爲 CPU 能夠直接處理之十六進制值。

FUN63 P →HEX	ASCII 碼轉換爲十六進制值	FUN63 P →HEX

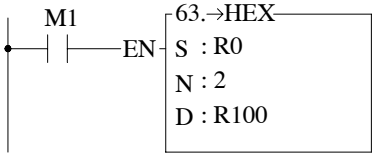
〈範例 1〉M1 由 OFF→ON 時，轉 ASCII 碼爲十六進制值



• 將 R0 之 ASCII 碼轉換爲十六進制值並存入 R100 之 Nibble 0（Nibble1～Nibble3 不變）

原　R100＝0000H
R0＝0039H（9）➔　R100＝0009H

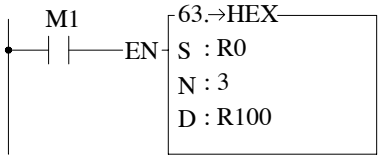
〈範例 2〉M1 ON 時，轉 ASCII 碼爲十六進制值



• 將 R0 與 R1 之 ASCII 碼轉換爲十六進制值並存入 R100 之低位元組（高位元組不變）

R0＝0039H（9）　原　R100＝0000H
R1＝0041H（A）➔　R100＝009AH

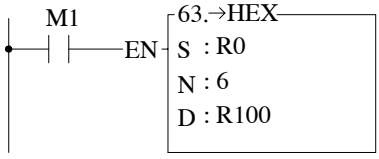
〈範例 3〉M1 ON 時，轉 ASCII 碼爲十六進制值



• 將 R0～R2 之 ASCII 碼轉換爲十六進制值並存入 R100（Nibble 3 不變）

R0＝0039H（9）　原　R100＝0000H
R1＝0041H（A）
R2＝0045H（E）➔　R100＝09AEH

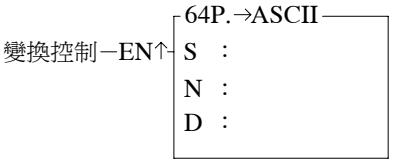
〈範例 4〉M1 ON 時，轉 ASCII 碼爲十六進制值



• 將 R0～R5 之 ASCII 碼轉換爲十六進制值並存入 R100～R101

R0＝0031H（1）　原　R100＝0000H
R1＝0032H（2）　　R101＝0000H
R2＝0033H（3）
R3＝0034H（4）
R4＝0035H（5）➔　R100＝3456H
R5＝0036H（6）　　R101＝0012H

FUN64 P →ASCII	十六進制值轉換爲 ASCII 碼	FUN64 P →ASCII



S ：來源暫存器之起始號碼

N：欲轉十六進制值爲 ASCII 碼之個數

D：存放結果（ASCII 碼）之暫存器起始號碼

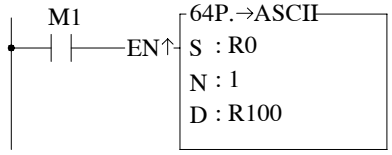
S，N，D 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	正數 16位元	V、Z
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 當變換控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 S 開始之暫存器連續 N 個位數（4 位元）之十六進制值轉換爲 ASCII 碼，並將結果存入 D 所指定開始暫存器之低位元組。(高位元組維持原值不變)。
- 當 N 之值爲 0 或大於 511 時，運算不執行。
- 此指令最大用途係將 PLC 處理完之數值資料轉換爲 ASCII 碼透過通訊埠 1 或通訊埠 2 傳送給 ASCII 週邊設備。

FUN64 P →ASCII	十六進制值轉換爲 ASCII 碼	FUN64 P →ASCII

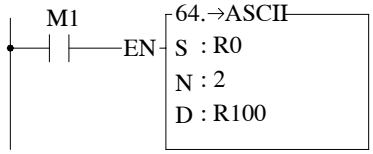
〈範例 1〉M1 由 OFF→ON 時，轉十六進制值爲 ASCII 碼



- 將 R0 之 Nibble 0 轉換爲 ASCII 碼並存入 R100 (高位元組不變)

R0＝0009H ➔ R100＝0039H（9）

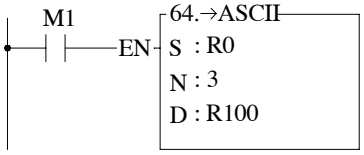
〈範例 2〉M1 ON 時，轉十六進制值爲 ASCII 碼



- 將 R0 之 NB0～NB1 轉換爲 ASCII 碼並存入 R100～R101（高位元組皆維持原值不變）

R0＝009AH ➔ R100＝0039H（9）
R101＝0041H（A）

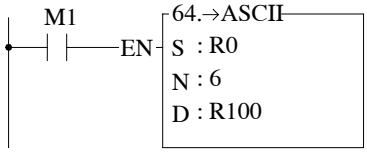
〈範例 3〉M1 ON 時，轉十六進制值爲 ASCII 碼



- 將 R0 之 NB0～NB2 轉換爲 ASCII 碼並存入 R100～R102

R0＝0123H ➔ R100＝0031H（1）
R101＝0032H（2）
R102＝0033H（3）

〈範例 4〉M1 ON 時，轉十六進制值爲 ASCII 碼



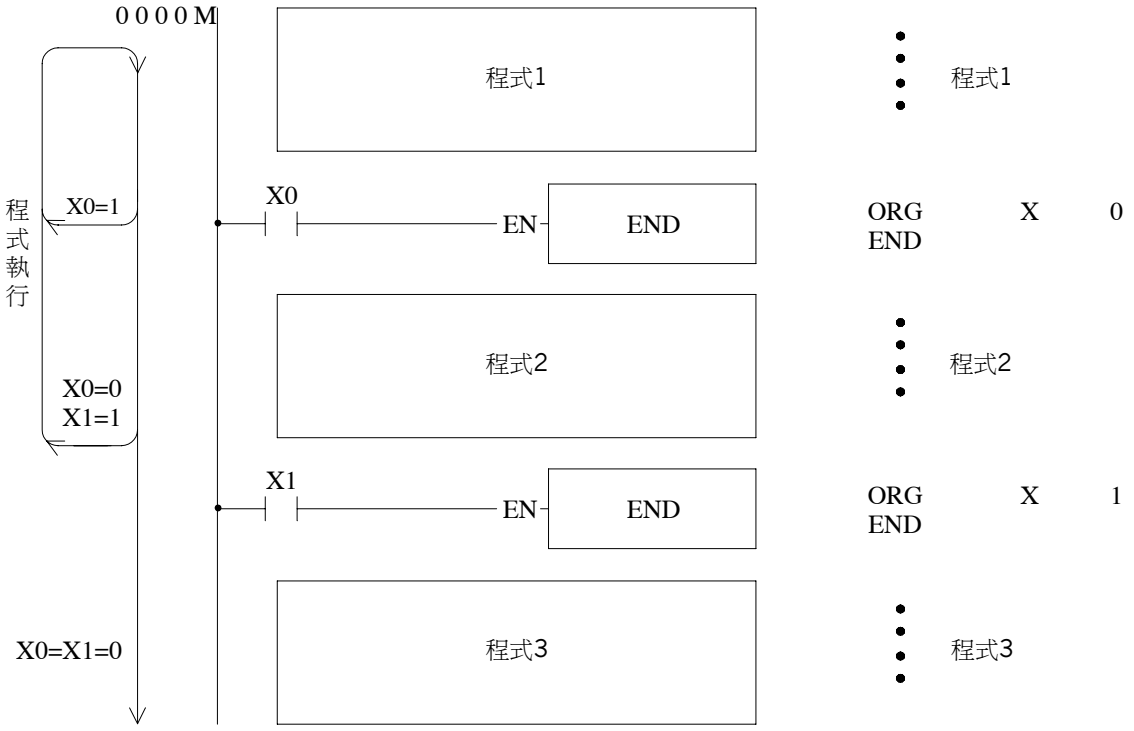
- 將 R0～R1 之 NB0～NB5 轉換爲 ASCII 碼並存入 R100～R105

R0＝3456H ➔ R100＝0031H（1）
R1＝0012H R101＝0032H（2）
R102＝0033H（3）
R103＝0034H（4）
R104＝0035H（5）
R105＝0036H（6）

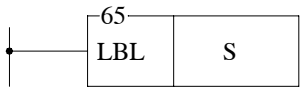
END	程式終止 （PROGRAM END）	END

無運算元

- 當終止控制〝EN〞=1 時，本指令動作，立即結束本次之程式掃描，亦即在 END 指令後之程式雖然存在，但卻不會被執行。當〝EN〞=0 時本指令不執行（當作無此指令），在 END 指令後之程式會繼續被執行。
- 本指令可在程式中多點放置，而以其輸入（終止控制〝EN〞）來控制程式執行之終止處，特別有利於除錯或測試。
- 程式之最後並不一定要有 END 指令，CPU 會自動偵察程式之結束。



FUN65 LBL	標記 （LABEL）	FUN65 LBL



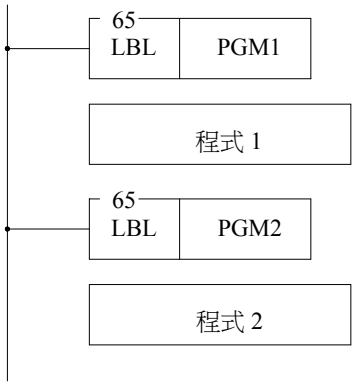
S：英文／數字 1～6 字

- 本指令用於標示程式中某一特定位址，以供程式跳躍（JUMP）到此標記所在之位址來執行，或當作中斷服務程式或副程式之名稱，以供中斷或呼叫（CALL）之用。若不需作跳躍或呼叫等之流程控制，亦可作標記來對程式作註解，以利程式之辨識或提高可讀性。
- 本指令只當程式位址標記以供流程控制或註解用，指令本身不會執行任何動作，程式中有沒有本指令，程式執行結果均不受其影響。
- 標記名稱可以 1～6 個任意英文字母或數字組成，但不得重複，且下列標記名稱是保留給中斷功能使用，稱之爲〝保留字〞，一般程式標記不得使用：

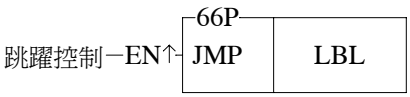
保　　留　　字	中 斷 服 務 程 式 名 稱
X0+I～X15+I（INT0～INT15） X0–I～X15–I（INT0–～INT15–）	外部 X0～X15 之中斷服務程式名稱
HSC0I～HSC7I	HSC0～HSC7 之中斷服務程式名稱
1MSI（1MS）、2MSI（2MS），3MSI（3MS），4MSI（4MS），5MSI（5MS），10MSI（10MS），50MSI（50MS），100MSI（100MS）	PLC 內部 1mS，2mS……100mS 等 8 種定時中斷之服務程式名稱
HSTAI（ATMRI）	高速定時中斷服務程式名稱
PSO0I～PSO3I	脈波輸出結束之中斷服務程式名稱

除非您所標註的程式確實是上述中斷所對應之服務程式才可用上述之名稱，其他地方不能使用，否則當中斷發生時，PLC 會把您標記之一般程式當作中斷程式執行，而造成錯誤或當機。

下圖例爲標記只當作程式註解（未被呼叫或跳躍至此標記）之範例，至於標記在跳躍控制之應用請參閱跳躍（JMP）指令之說明，標記當副程式名稱則請參閱呼叫（CALL）指令之說明。

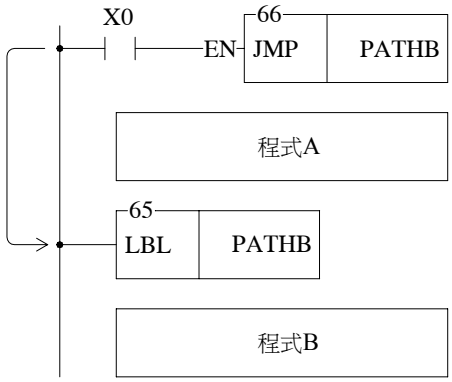


FUN66 P JMP	跳躍 （JUMP）	FUN66 P JMP



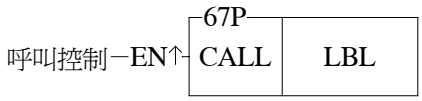
LBL：欲跳躍之程式標記

- 當跳躍控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，PLC 直接跳到其後標記 LBL 所在位置，繼續往下執行程式。
- 本指令之應用特別適合在特定狀況發生才需執行某部分程式的應用，平常不執行以節省時間。以及在線圈多重輸出之應用場合，再以輸入控制選擇執行某一段程式之應用。
- 本指令程式跳躍可往回跳（即跳回之 LBL 位址比該 JMP 指令所在之位址要小），但需注意如往回跳致使掃描時間延長超過 Watchdog Timer 所設定之時間，則 PLC 會發生 WDT 中斷，而停止運轉，並發出錯誤訊號。
- 跳躍指令只限於主程式區跳主程式區，或副程式區跳副程式區，不能跨越主／副程式區作跳躍。



- 左圖中當 X0=1，則程式執行將由 JMP 指令所在處直接跳到 LBL 名稱爲 PATHB 之地方往下執行，故程式 A 被跳過，A 中所有指令均不執行，和程式 A 相關之單點或暫存器狀態均保持不變（如同無 A 這段程式）。

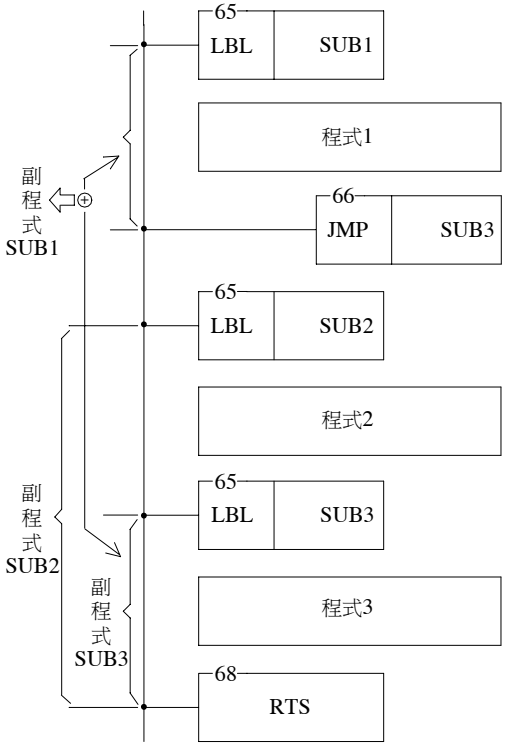
FUN67 P CALL	呼叫 （CALL）	FUN67 P CALL



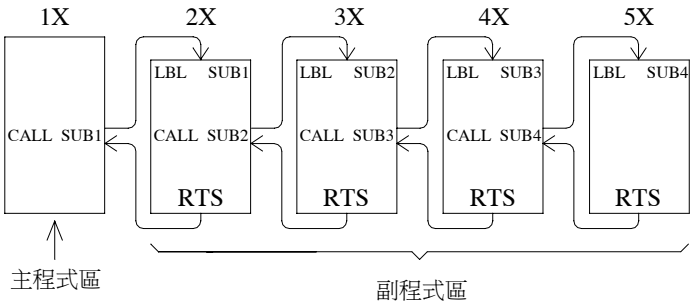
LBL：被呼叫之副程式標記名稱

- 當呼叫控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，PLC 將呼叫（執行）標記名稱與被呼叫之標記名稱相同之副程式，在副程式執行前 PLC 會先將副程式執行完後所需返回之位址（該 CALL 指令之下一個位址）存入 CPU 內部之堆疊（STACK）內，然後再去執行呼叫之副程式，直到遇到副程式中之〝副程式返回指令 RTS〞後才將先前存入堆疊之返回位址取回，而從返回位址處之指令往下繼續執行程式。

- 副程式之最後均要有〝副程式返回指令 RTS〞，否則將造成執行錯誤或當機，但多個副程式可共用一個 RTS 指令（此即所謂之多進入點副程式，此種副程式之進入點不同，返回點卻一致），如右圖例之副程式 SUB1～3。

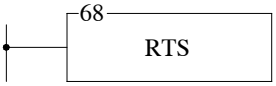


- 主程式呼叫副程式後副程式尚可呼叫其他副程式(即所謂巢式副程式)，最多可達 5 層(中斷＋呼叫)。



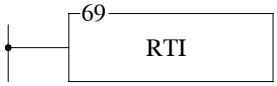
- 中斷服務程式（HSC0I～HSC7I、PSO0I～PSO3I、X0+I～X15+I／INT0～INT15、X0–I～X15–I／INT0–～INT15–、HSTAI／ATMRI、1MSI／1MS、2MSI／2MS、3MSI／3MS、4MSI／4MS、5MSI／5MS、10MSI／10MS、50MSI／50MS、100MSI／100MS）也算是副程式之一種，亦存放在副程式區內，但中斷服務程式之呼叫，是利用硬體觸發信號促使 CPU 去執行對應之中斷服務程式（我們稱之爲中斷服務程式之召用）。中斷服務程式亦能再呼叫副程式或再召用中斷服務程式，但因其本身就是副程式（已佔一層），因此最多只能再呼叫或召用四層副程式或中斷服務程式，請參閱 RTI 指令之說明。

FUN68 RTS	副程式返回 （RETURN FROM SUBROUTINE）	FUN68 RTS



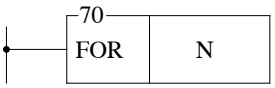
- 本指令用於表示一副程式之終了，因此只能出現在副程式區內，其輸入側無控制信號，故無法串聯任何元件，本指令單獨就是一完整指令，係直接接到母線上。
- 當 PLC 執行到本指令時，表示副程式已執行完畢，因此會將先前存入堆疊中之返回位址取回，以便 PLC 回到先前呼叫副程式之下一個指令，繼續往下執行程式。
- 若在副程式中執行不到 RTS 指令，則程式流程將不再正確，系統堆疊也會被破壞（M1933 ON），並造成系統失控。因此，無論流程如何控制，均需確保所有副程式均會執行到 RTS 指令。
- RTS 指令之應用請參考 CALL 指令之說明。

FUN69 RTI	中斷返回 （RETURN FROM INTERRUPT）	FUN69 RTI



- 本指令之功能和 RTS 類似，只是 RTS 是用於副程式之最後，而 RTI 則用於中斷服務程式之最後，請參閱 RTS 指令之說明。
- 多個中斷服務程式可共用一個 RTI 指令，其用法和多個副程式可共用一個 RTS 指令一樣，請參考呼叫（CALL）指令之說明。
- 中斷和呼叫之差異只有在呼叫係由使用者自行定義副程式之名稱（標記 LBL），然後在主程式或其他副程式中有呼叫指令並指名該副程式之標記，如此當 PLC 執行到該呼叫指令（CALL），且其輸入〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 時，PLC 即會去呼叫（執行）此副程式。而中斷服務程式之執行則是直接以硬體訊號來中斷 CPU，要 CPU 暫停其他較次要之工作，而來執行該硬體信號所對應之中斷服務程式（我們稱爲中斷服務程式召用）。因此較之呼叫必須掃描到該呼叫指令才會執行之作法，中斷則爲更即時（Real Time）之作法。此外因中斷服務程式無法指名呼叫，因此我們以特定之〝保留字〞標記名稱來對應 PLC 所提供之各種中斷（詳見 FUN65 說明），例如保留字 X0+I 指定給輸入點 X0 所發生之中斷，只要副程式中有標記爲 X0+I 之程式，當輸入點 X0 中斷允許發生（X0：↑），PLC 就會立即暫停其他較不優先之程式掃描工作，而馬上跳到副程式中標記爲 X0+I 的位址去執行程式。
- 若中斷發生之時，CPU 正在處理比此中斷優先度更高（如硬體高速計數器中斷）或優先度一樣之中斷（請參考第 10 章之優先等級），則 PLC 會等執行完上述所有中斷服務程式後才會處理此中斷。
- 若在中斷服務程式中執行不到 RTI 指令，則 PLC 之系統堆疊會被破壞、程式流程錯亂，而有可能引起嚴重當機。因此，無論流程如何控制，均需確保任一中斷服務程式均會執行到 RTI 指令。
- 關於中斷之詳細說明與使用方法範例請參閱第 10 章之說明。

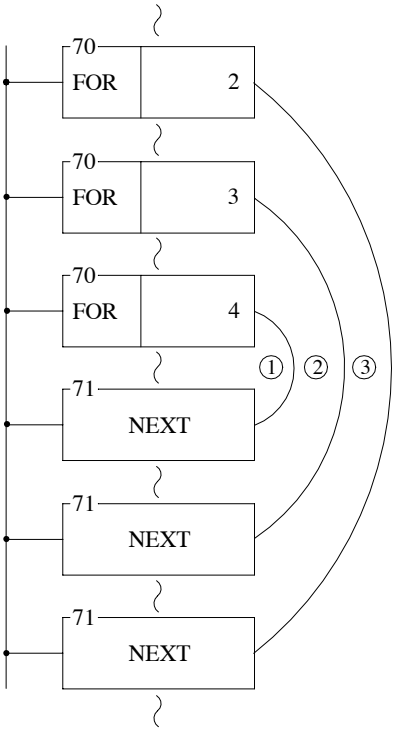
FUN70 FOR	迴圈開始 （FOR）	FUN70 FOR



N：迴圈執行次數

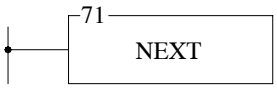
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	1 ｜ 16383
N	○	○	○	○	○	○	○	○	○	○	○	○	○

- 本指令無輸入控制，係直接接於母線，不能串接任何元件。
- FOR 指令和 NEXT 指令所包夾之程式形成一程式迴圈（迴圈程式之開頭爲 FOR 之次一個指令，結尾爲 NEXT 之前一個指令），當 PLC 執行到 FOR 指令時，首先記下該指令後之 N（迴圈執行次數），然後將此迴圈內的程式從頭到尾連續執行 N 次後，跳離該迴圈，繼續往下（NEXT 之次一指令開始）執行。
- 迴圈可爲巢式結構，即迴圈內包含著迴圈，猶如洋蔥一般，一個迴圈稱爲一層最多可達 5 層。FOR 和 NEXT 指令必須成對使用，第一個 FOR 指令和最後一個 NEXT 爲巢式迴圈之最外（第一）層。而第二個 FOR 指令和倒數第二個 NEXT 指令爲第二層，……最後一個 FOR 指令和第一個 NEXT 指令形成最內層之迴圈。



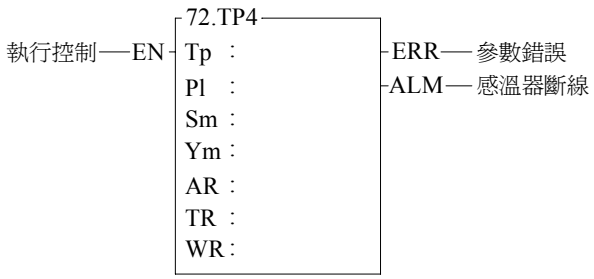
- 左圖例迴圈①將被執行 4×3×2=24 次，迴圈②將被執行 3×2=6 次，而迴圈③則會執行 2 次。
- 若有 FOR 指令而無 NEXT 指令與之對應，或巢式迴圈之 FOR 和 NEXT 指令未配對使用，或 FOR、NEXT 順序顛倒，均將造成語法錯誤，程式無法執行。
- 迴圈中不可使用 JMP 指令跳出迴圈，否則 PLC 之系統堆疊會被破壞、程式流程錯亂，而有可能引起嚴重當機。
- N 之有效範圍爲 1～16383 次，超出此範圍，PLC 均將之視爲 1 次。N 之次數若太大，或迴圈程式太長，可能造成 Watchdog 發生，請注意。

FUN71 NEXT	迴圈結束	FUN71 NEXT



- 本指令和 FOR 指令配合形成一個程式迴圈。指令本身無輸入控制，係直接接於母線，不能串接任何元件。
- 未執行到 FOR 指令，絕不可以執行到 NEXT 指令，否則有可能造成 PLC 當機。
- 其應用請參閱前頁 FOR 指令之說明。

FUN 72 TP4	溫度模組溫度量測便利指令 （功能簡述）	FUN 72 TP4



運算元＼範圍	Y	HR	IR	ROR	DR	K
	Y0 | Y255	R0 | R3839	R3840 | R3903	R5000 | R8071	D0 | D3071	
Tp						0～2
Pl						0～3
Sm						n×4，n=0～7
Ym	○					
AR			○			
TR		○		○	○*	
WR		○		○	○*	

Tp ：感溫器選擇，可選擇 J 或 K Type 即熱電耦或 PT-100 RTD

Pl ：溫度量測模組之電壓範圍與極性設定

Sm ：溫度模組所量測之起始溫度點

Ym ：溫度模組之多工輸出起始號碼

AR ：溫度模組之類比輸入暫存器號碼

TR ：存放溫度量測值之起始暫存器號碼

WR：本指令所需使用之工作暫存器起始號碼

指令功能簡述

●本指令爲 FB-J(K)4 或 FB-RTD4 多工溫度量測模組之專用量測指令，透過本指令，使用者可以用填表方式輕易地取得多點（每一指令可處理一片模組，即 4 點）溫度量測值，以供監視或記錄之用。

●本指令必須配合 FB-J(K)4 或 FB-RTD4 多工溫度量測模組使用，此處僅簡介本指令之功能，詳細之指令功能與說明及用法與範例，請參閱第 20 章〝FB-PLC 之溫度量測及 PID 控制〞之敘述。

FUN 73 TSTC	溫度模組溫度量測 + PID 溫控便利指令 （功能簡述）	FUN 73 TSTC

運算元 \ 範圍	Y	HR	IR	DR	ROR	K
	Y0 | Y255	R0 | R3839	R3840 | R3903	D0 | D3071	R5000 | R8071	
Tp						0～2
Pl						0～3
Sm						n×4 n=0～7
Ym	○					
AR			○			
TR		○		○	○*	
Yh	○					
Sh						0～23
Zh						1～24
Sv		○		○	○*	
Os		○		○	○*	
PR		○		○	○*	
IR		○		○	○*	
DR		○		○	○*	
OR		○		○	○*	
WR		○		○	○*	

Tp ：感溫器選擇，可選擇 J 或 K Type 即熱電耦或 PT-100 RTD

Pl ：溫度量測模組之電壓範圍與極性設定

Sm ：溫度模組所量測之起始溫度點

Ym ：溫度模組之多工輸出起始號碼

AR ：溫度模組之類比輸入暫存器號碼

TR ：存放溫度量測值之起始暫存器號碼

Yh ：PWM 溫控輸出點起始號碼

Sh ：指定從第幾點執行 PID 溫控

Zh ：本指令所控制之 PID 溫控點數

Sv ：存放溫度設定值之起始暫存器號碼

Os ：存放溫度偏差值之起始暫存器號碼

PR ：存放增益設定值之起始暫存器號碼

IR ：存放積分時間常數設定值之起始暫存器號碼

DR ：存放微分時間常數設定值之起始暫存器號碼

OR ：存放溫控數值輸出起始暫存器號碼

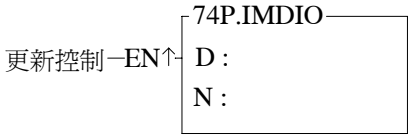
WR ：本指令所需使用之工作暫存器起始號碼

指令功能簡述

●本指令爲 FB-J(K)4 或 FB-RTD4 多工溫度量測模組之專用量測+PID 溫控指令，透過本指令，使用者可以用填表方式輕易地達成多點迴路 PID 溫控。

●本指令必須配合 FB-J(K)4 或 FB-RTD4 多工溫度量測模組使用，此處僅簡介其功能，詳細之指令功能與說明及用法與範例，請參閱第 20 章〝FB-PLC 之溫度量測及 PID 控制〞之敘述。

FUN74 P IMDIO	即時 I／O 更新 （IMMIDIATE I/O REFRESH）	FUN74 P IMDIO



D：欲更新之 I/O 點起頭號碼
N：欲更新之 I/O 點數

運算元＼範圍	X	Y	K
	主機上之 Xn	主機上之 Yn	1 ～ 24
D	○	○	
N			○

- PLC 系統之輸入／輸出信號更新通常在程式執行前先一次抓取全部之輸入信號，然後開始掃描程式，等全部掃描結束才將所有輸出結果一次送到輸出點，如此之輸入動作至輸出反應至少會有一個掃描時間之延遲（最大為 2 個掃描時間）。本指令之作法則為遇到本指令便立即去抓取或送出指令所指定之輸入信號或輸出信號，如此可獲得最即時快速之輸入／輸出反應。

- 當更新控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 D 所指定之 I（輸入點）或 O（輸出點）開始之 N 個輸入點或輸出點（即 D～D+N−1）狀態更新。

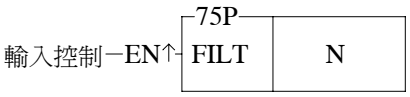
- PLC 之即時 I/O 更新之 I/O 點僅限於在主機上之 I/O 點。下表為 20、28、40 點主機可容許之即時 I/O 號碼：

主機 I/O 點數／容許號碼	20 點	28 點	40 點
輸入點	X0～X11	X0～X15	X0～X23
輸出點	Y0～Y7	Y0～Y11	Y0～Y15

- 如果程式中之即時 I/O 點之範圍超出主機之輸入點或輸出點號碼（例如程式中 D=X7，N=10，則表示要即時抓取 X7～X16 等 10 個輸入點信號，而假設主機為 28 點 I/O 機種，其輸入點最大為 X15，明顯地 X16 已超出該主機之輸入點號碼）則 PLC 將無法運轉（STOP，ERR 燈亮）同時 M1931 錯誤旗號設定為 1。

- 本指令執行時，雖然 PLC 會立即去抓取或送出即時輸入／輸出信號，但在輸入點上之硬體或軟體積分之延遲或輸出點之動作延遲（如繼電器或電晶體等輸出元件之動作反應時間）仍然存在，請特別注意。

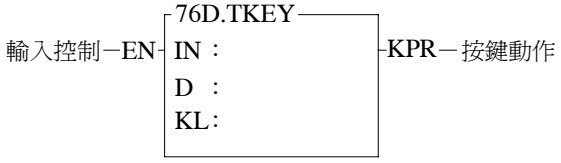
FUN75 P FILT	輸入濾波時間 （FILTER ADJUST）	FUN75 P FILT



N：濾波時間 0～30（mS）

- 本指令專門針對主機上 X0～X15 等 16 個輸入點之輸入作軟體積分（濾波）時間調整，當輸入控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 X0～X15 等 16 點之輸入濾波時間設定爲 NmS。
- 實際上 X0～X15 這 16 個輸入點爲提高雜訊抗性，均加有硬體數位濾波器（Hardware Digital Filter），而硬體數位濾波器可規劃 X0～X15 之最高輸入頻率由 4KHZ～512KHZ，視現場雜訊干擾情況可動態調整。
- 除 X0～X15 這 16 點輸入點外，其餘輸入點均加有大約 4mS 之 RC 濾波電路以提高雜訊抗性，因其延遲時間長不適合作高速之動作。此時可利用本指令將 X0～X15 之濾波時間設定很小，而當作高速輸入。除此之外亦可將其設定大至 30mS，以作爲濾除特定雜波之輸入應用。

FUN76 **D** TKEY	10 進位數字按鍵 （DECIMAL KEY-IN）	FUN76 **D** TKEY

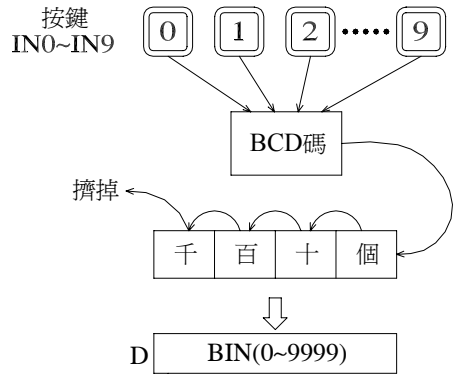


IN ：按鍵輸入點
D ：存放按鍵數字之暫存器號碼
KL：輸入按鍵之對應繼電器起頭號碼
D 可結合 V、Z 作間接定址應用

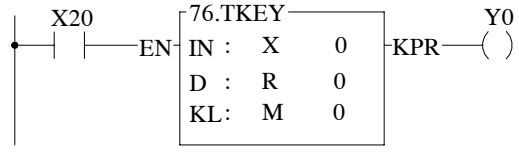
運算元＼範圍	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 | X240	Y0 | Y240	M0 | M1896	S0 | S984	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	V 、 Z
IN	○														
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

- 本指令指定由 IN 開始之連續 10 個輸入點（IN0～IN9）依序代表十進位數字之 0～9（BCD 碼爲 0000～1001），依據這些輸入點被壓下（ON）之先後順序可輸入 4 個或 8 個十進位數字到 D 所指定之暫存器去。

- 當輸入控制〝EN〞=1，本指令會去檢視 IN 開始之 10 個輸入點並將〝ON〞輸入點所代表之十進位數字存入 D 中，俟該輸入點放開後，再檢視下一個〝ON〞的輸入點，再將其所代表之數字擠進 D 中（先存入者爲高位數，後存入者爲低位數）。在 16 位元指令中 D 可存放 4 位數，而 32 位元 **D** 指令可存放 8 位數，超出時則擠掉先存入者（即高位數之數字）。IN 開始之 10 個輸入點按鍵狀況將會被記錄在由 KL 開始之 10 個對應的繼電器上，同時只要有任一輸入點被按下（ON），則按鍵動作旗號〝KPR〞即變爲 1。在同一時間內 IN0～IN9 中只能有一個被按下，若超過一個只取最先按下者，下圖爲 16 位元指令之功能示意圖（32 位元亦同，但數值可達〝千萬〞）。



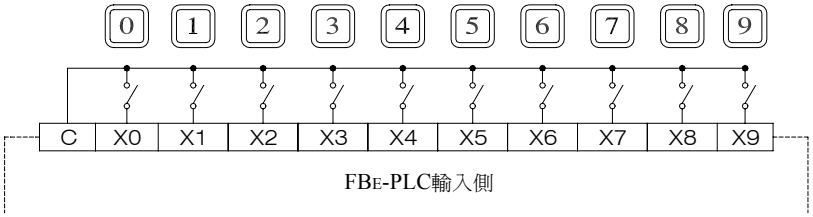
- 當輸入控制〝EN〞=0 時，本指令不執行，同時清除〝KPR〞輸出及 KL 繼電器之狀態爲 0，但暫存器 D 之數值則保持不變。



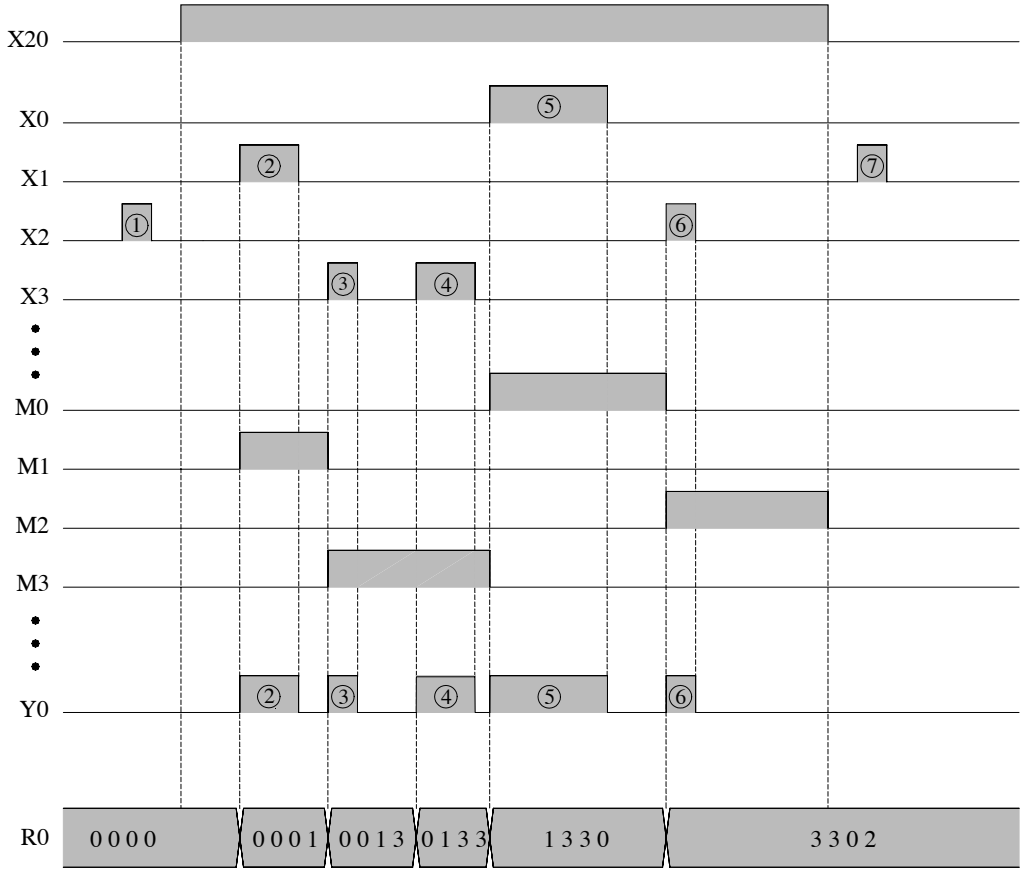
- 左圖程式指定輸入點 X0 代表數字〝0〞，X1 代表〝1〞，……，按鍵狀況則以 M0 記錄 X0 之動作，M1 記錄 X1 之動作……，輸入之數值存於 R0 暫存器中。

FUN76 D TKEY	10 進位數字按鍵 （DECIMAL KEY-IN）	FUN76 D TKEY

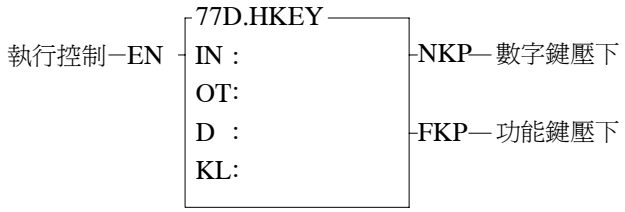
下圖爲本範例之實際輸入配線圖：



- 假設 X0～X3 之按鍵順序如下圖之①②③④⑤⑥⑦之順序，因①和⑦按下時 X20 爲 0，而不發生作用，有效者僅爲②③④⑤⑥，而此 5 個按鍵之第一個按鍵②因已超出 4 位而被擠掉，只剩下③④⑤⑥之按鍵數字 3302 存於暫存器 R0 中。



FUN77 **D** HKEY	16 個鍵多工輸入 （HEX-KEY INPUT）	FUN77 **D** HKEY



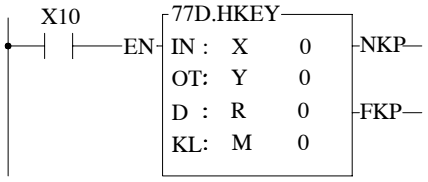
IN ：按鍵掃描輸入點號碼
OT：多工掃描輸出點號碼
D ：存放〝按鍵數字〞之暫存器號碼
KL：記錄〝動作鍵〞之繼電器起頭號碼
D 可結合 V、Z 作間接定址應用

運算元＼範圍	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 ｜ X240	Y0 ｜ Y240	M0 ｜ M1896	S0 ｜ S984	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	V 、 Z
IN	○														
OT		○													
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

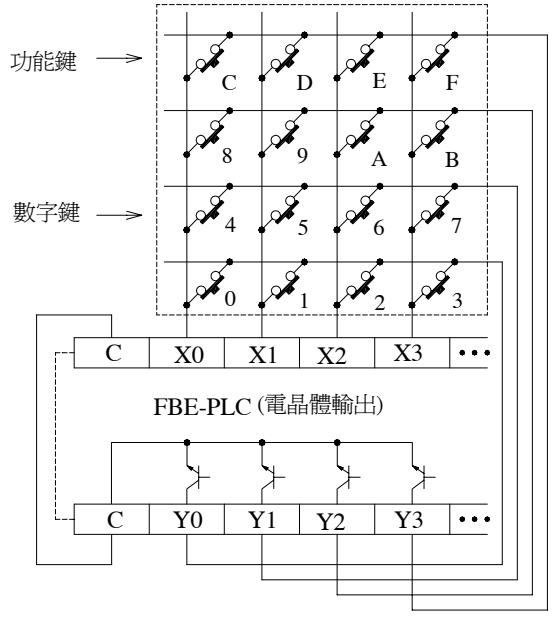
- 本指令之數字鍵（0～9）功能和 TKEY 指令非常類似，只是硬體輸入接線在 TKEY 指令係一個按鍵佔一輸入點，而本指令則以 4 個輸入點配合 4 個輸出點組成多工掃描輸入方式，因 4×4 可有 16 個輸入鍵，除 10 個數字鍵外，尚餘之 6 個則當功能鍵使用（和一般單點輸入相同），數字鍵和功能鍵之動作是獨立而互不影響。

- 當執行控制〝EN〞=1 時，本指令會掃描由 IN 開始之 4 個輸入點和由 OT 開始之 4 個輸出點組成之矩陣回路中之數字鍵和功能鍵兩部分，數字鍵部份請參考 TKEY 指令，而功能鍵則將 A～F 鍵之按鍵狀態保持在 KL 所指 16 個繼電器之後 6 個（前 10 個存數字鍵之按鍵狀態），同時 A～F 有任一鍵壓下，〝FKP〞（FO1）爲 1。本指令之 OT 輸出點必須爲電晶體輸出。

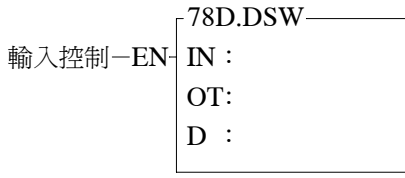
- 16 位元指令最大可輸入 4 位數（9999），**D**指令最大則爲 8 位數（99999999），但功能鍵無論 16 或 32 位元指令均只有 A～F 6 個。



• 上圖程式範例以 X0～X3 和 Y0～Y3 組成多工按鍵輸入，可以輸入 8 位數之數值而將結果存放於 R1R0 中，功能鍵之輸入狀態則存放於 M10（A）～M15（F）中。



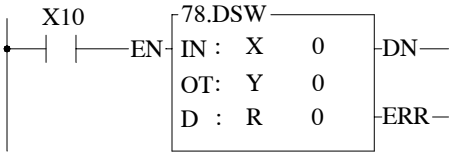
FUN78 **D** DSW	指撥開關輸入 （DIGITAL SWITCH）	FUN78 **D** DSW



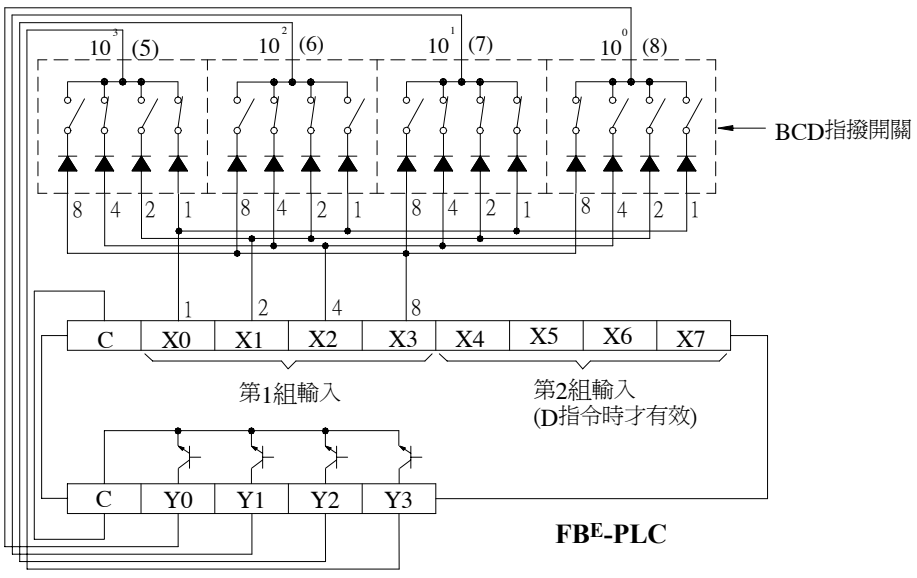
IN ：開關輸入點（4 點，D 指令爲 8 點）
OT：多工掃描輸出點（4 點）
D ：存放讀值之暫存器號碼
D 可結合 V、Z 作間接定址應用

運算元	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 | X240	Y0 | Y240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	V 、 Z
IN	○												
OT		○											
D			○	○	○	○	○	○	○	○*	○*	○	○

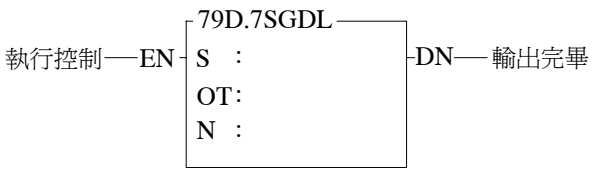
- 當輸入控制〝EN〞=1 時，本指令會以 IN 開始之 4 個輸入點（IN0～IN3）當作一個位數（Nibble），自低（個）位數開始分四次掃描讀取一組 4 個位數之 BCD 數值（0000～9999）再將之存入 D 中，若爲 32 位元（**D**指令）則一次掃描同時讀取兩組之位數（即 IN0～IN3 和 IN4～IN7），而將由 IN4～IN7 讀到之那組 4 個位數值存入 D+1 暫存器中，掃描之順序是將 OT0～OT3 位元依序設爲 1，而分別讀到 10^0（個）、10^1（十）、10^2（百）、10^3（千）4 位數。只要〝EN〞爲 1，則 PLC 會循環不停的掃描讀取，每一循環（10^0～10^3 4 個位數讀取完畢）結束，讀取完畢旗號〝DN〞會設爲 1，但只維持一個掃描時間 t。若有任一數讀值非 0～9（BCD），則讀值錯誤〝ERR〞設爲 1，該組數值設爲 0000。
- 本指令只能使用一次，且其輸出點須爲電晶體輸出。



- 本範例當 X10 爲 1 則指撥開關之數字（本例爲 5678）值會被讀取存入 R0 中。
- 各位數同值之 Bit（8，4，2，1）要並聯在一起且需串二極體，如下圖所示。（市售指撥開關通常已串加二極體）
- **D**指令時再加裝一組同樣之指撥開關到 X4～X7 即可（Y0～Y3 共用）。



FUN79 **D** 7SGDL	7 段顯示器掃描輸出 （7 SEGMENT OUTPUT WITH LATCH）	FUN79 **D** 7SGDL

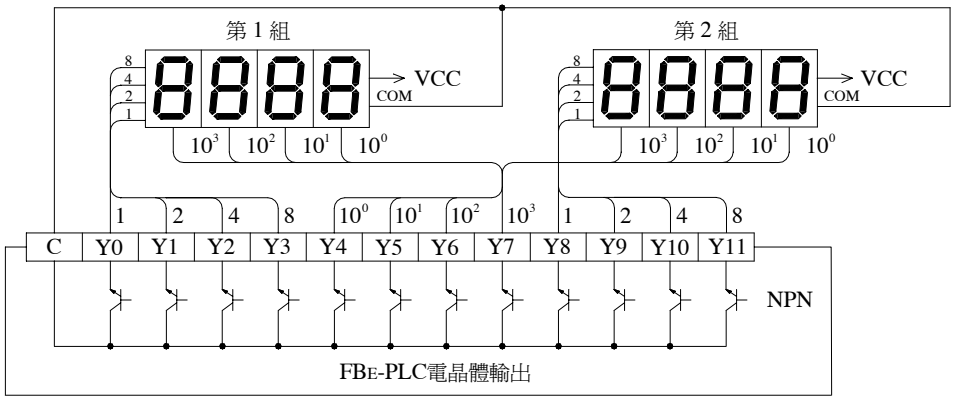


S ：顯示資料（BCD）存放之暫存器號碼
OT：掃描輸出點起頭號碼
N ：指定信號輸出和閂鎖信號之極性
S 可結合 V、Z 作間接定址應用

運算元＼範圍	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	Y0 | Y240	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正負數	V、Z
S			○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○														
N														0～3	

- 當執行控制〝EN〞=1 時，將暫存器 S 之 4 個位數（Nibble）即位數 0～位數 3 依序分四次送至 OT0～OT3 之 4 個輸出點，同時每送出一位數，即送出該位數之閂鎖信號，（OT4 對應到位數 0，OT5 對應到位數 1，……），以便將這些送出之位數值載入並閂鎖在 7 段顯示器內。
- D 指令時則將 S 暫存器之位數 0～3 和 S+1 暫存器之位數 0～3，同時分別送至 OT0～OT3 和 OT8～OT11，因係同時送出，故共用閂鎖信號。16 位元指令沒有使用到 OT8～OT11。
- 只要〝EN〞維持 1，PLC 會循環的執行送出動作，在每次送完整組數值（位數 0～3）後，輸出完畢旗號〝DN〞會變爲 1，但只維持一個掃描時間 t。

- 本程式範例當 X0=1 時，R0 之 4 位數字將被送到下圖第一組 7 段顯示器上，R1 之 4 位數將被送到第 2 組 7 段顯示器上

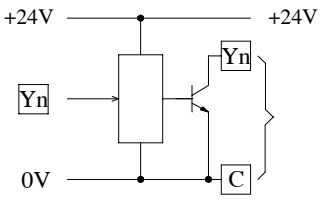


FUN79 **D** 7SGDL	7 段顯示器掃描輸出 （7 SEGMENT OUTPUT WITH LATCH）	FUN79 **D** 7SGDL

● 永宏 PLC 之電晶體輸出有負邏輯電晶體輸出（NPN 電晶體，當該點狀態爲 ON 時，該電晶體輸出端電壓爲 Low）及正邏輯電晶體輸出（PNP 電晶體，當該點狀態爲 ON 時，該電晶體輸出端電壓爲 High）兩種，其結構如下：

SINK

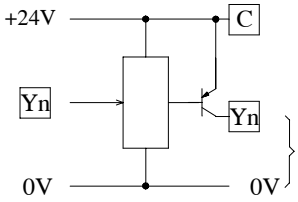
PLC 負邏輯輸出（NPN 電晶體）



當 Yn 爲〝ON〞時此輸出電壓爲 Low

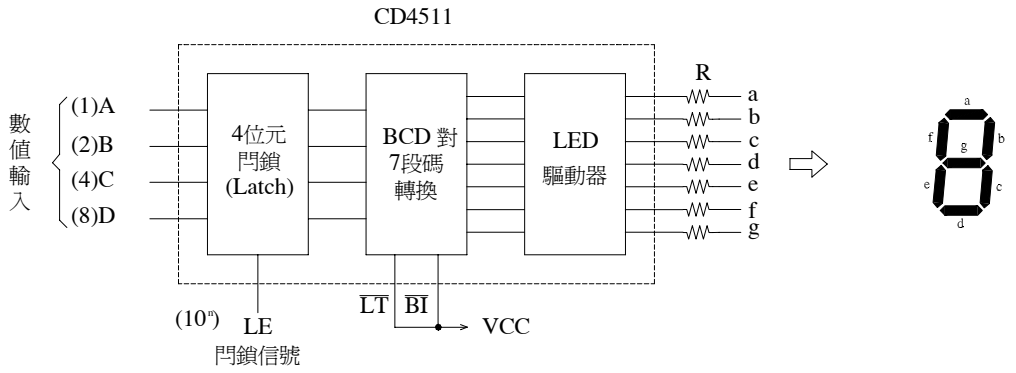
SOURCE

PLC 正邏輯輸出（PNP 電晶體）



當 Yn 爲〝ON〞時 Yn 輸出端電壓爲 High

● 市售 7 段顯示器之數值輸入（8、4、2、1）和閂鎖信號亦有正、負邏輯之分，例如某一位數值爲〝8〞，正邏輯輸入應爲 1000，但負邏輯輸入則爲 0111。相同地，正邏輯閂鎖在該閂鎖信號爲 0 時允許顯示數值進入閂鎖（即載入），而當其爲 1 時將當時閂鎖內之數值鎖住（保持），而負邏輯則反之。下圖 CD-4511 七段顯示 IC 爲正邏輯數值輸入及閂鎖之一例。

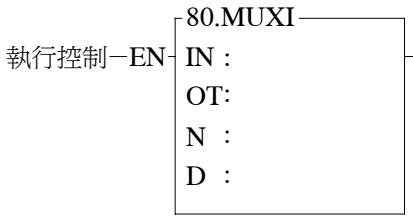


● 因有 PLC 正、負邏輯輸出極性和 7 段顯示器極性之區分，若欲將之連結並得到正確顯示，必須兩者極性要能配合，本指令利用 N 來指定 PLC 電晶體輸出之極性，以配合 7 段顯示器之極性而達成一致，下表爲 PLC 輸出和 7 段顯示極性組合所必須指定 N 值。

數值輸入（8～1）	閂鎖信號（10^0～10^3）	正確之 N 值
一致	一致	0
	不一致	1
不一致	一致	2
	不一致	3

● 以上圖 7 段顯示器 CD4511 爲例，其數值輸入和 PLC 不一致，而閂鎖信號則一致，故 N 值應設定爲 2。

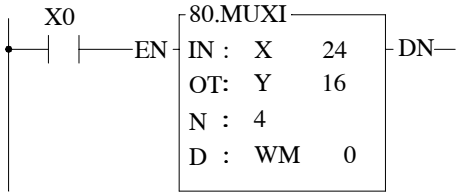
FUN80 MUXI	多工接點輸入 （MULTIPLEX INPUT）	FUN80 MUXI



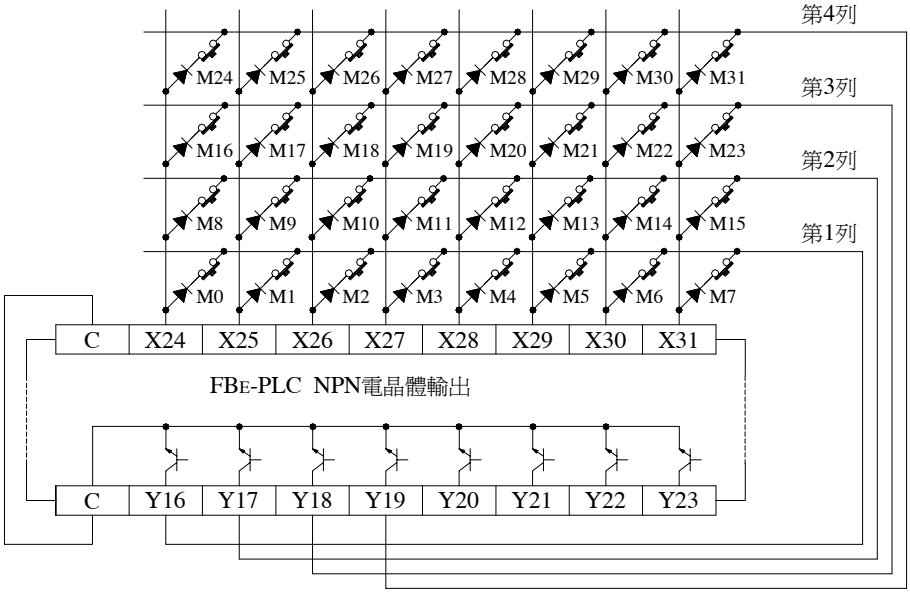
IN ：多工輸入點號碼
OT：多工輸出點號碼
（必須爲電晶體輸出點）
N ：多工輸入之列數（2～8）
D ：存放結果之暫存器號碼
D 可結合 V、Z 作間接定址應用

運算元＼範圍	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	X0 ｜ X240	Y0 ｜ Y240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 8	V 、 Z
IN	○													
OT		○												
N													○	
D			○	○	○	○	○	○	○	○*	○*	○		○

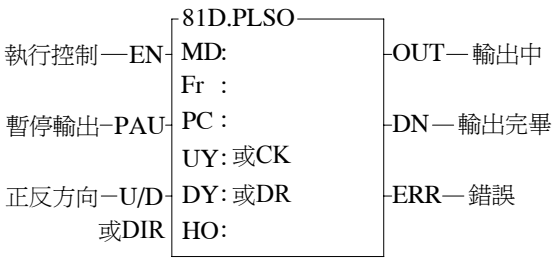
- 本指令以多工方式自 IN 所指定之輸入點開始之連續 8 個輸入點（IN0～IN7），讀取 N 列輸入狀態，而獲得 8×N 個輸入狀態，但卻只須用到 8 個輸入點和 N 個輸出點而已。
- 多工掃描方式是自 OT 輸出點開始之 N 個輸出點中，由 OT0 開始設爲 1，讀取第一列狀態，接著把 OT1 設爲 1，讀取第 2 列狀態，……直到讀完 N 列爲止。再將所讀取到的 8×N 個狀態存入由 D 開始之暫存器中，並將執行完畢旗號〝DN〞設爲 1（但只維持一個掃描時間）。
- 本指令每一次掃描抓取一列 8 個輸入點狀態，故 N 列要 N 個掃描時間才能抓完。



- 本範例抓取 4 列×8 點輸入共 32 點狀態，並將之存放在 DWM0（M0～M31）之 32 位元暫存器中。



FUN81 **D** PLSO	脈波輸出指令 (PULSE OUTPUT)	FUN81 **D** PLSO



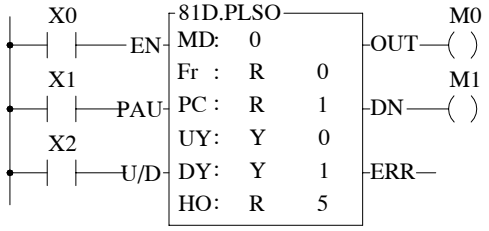
MD ：運轉模式選擇
Fr ：脈波頻率
PC ：輸出脈波數
UY ：正轉脈波之輸出點（MD=0）
DY ：反轉脈波之輸出點（MD=0）
HO ：已送出脈波暫存器（可不指定）
CK ：脈波輸出點（MD=1）
DR ：正／反轉輸出點（MD=1）
DIR：1，正轉；0，反轉

運算元＼範圍	Y	WX	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K
	主機上之 Yn	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正數
MD													0～1
Fr		○	○	○	○	○	○	○	○	○	○	○	8～2000
PC		○	○	○	○	○	○	○	○	○	○	○	○
UY，CK	○												
DY，DR	○												
HO			○	○	○	○	○	○	○	○*	○*	○	

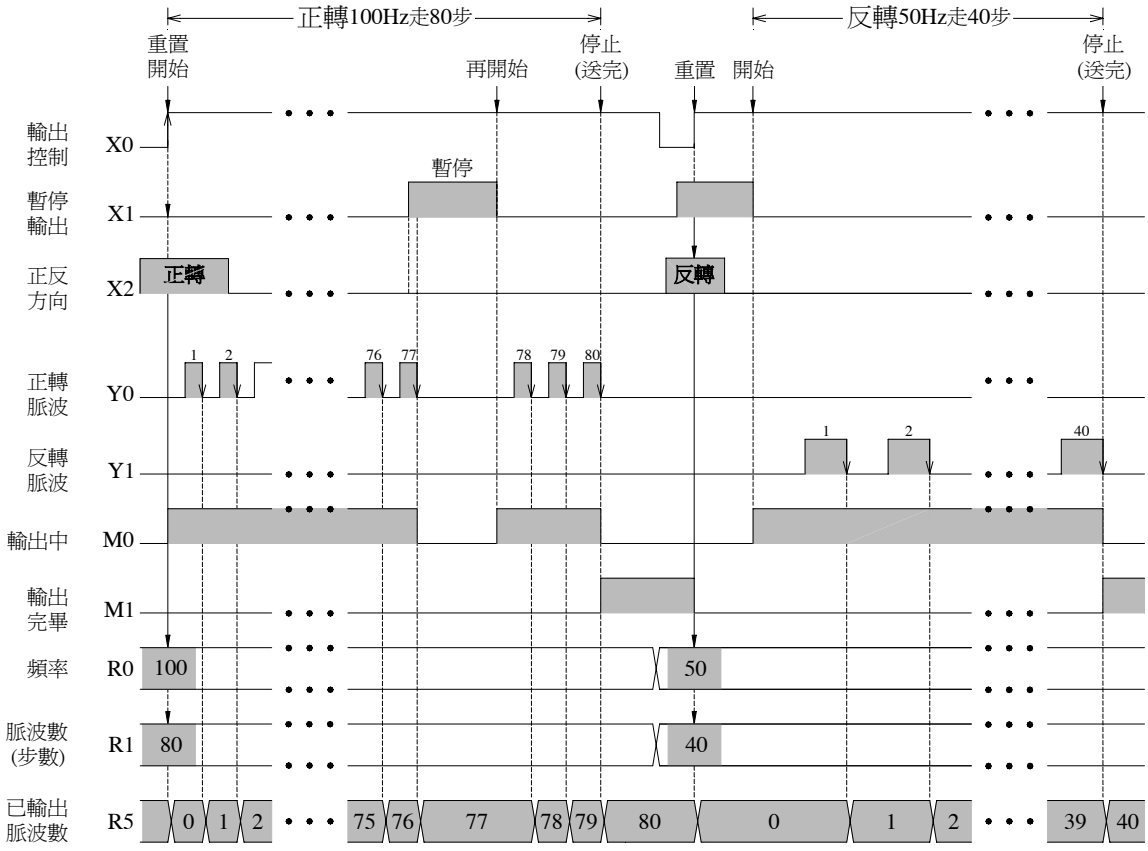
- 當 MD=0 時，本指令如下述方式作脈波輸出控制：
- 當輸出控制〝EN〞由 0→1 瞬間，首先執行重置(RESET)動作，亦即將輸出旗號〝OUT〞和〝DN〞以及已送出脈波暫存器 HO 均清爲 0，並抓取脈波頻率 Fr 及脈波數 PC 之值，再讀取正反方向〝U/D〞之狀態以決定正、反轉方向。完成重置工作後本指令已完成輸出準備，緊接著檢視暫停輸出〝PAU〞之狀態，若其爲 1（暫停輸出）則不作任何動作，若爲 0 則開始以脈波頻率 Fr 所指定之頻率自 UY(U/D=1 時)或 DY(U/D=0 時）輸出點送出 ON/OFF 波寬各爲 50%之方形脈波，每送出一個脈波即將 HO 暫存器加 1，一直到 HO 暫存器內之脈波數等於或大於 PC 暫存器之脈波數始停止脈波之送出，並將輸出完畢旗號〝DN〞設爲 1。任何時刻只要本指令是在脈波輸出當中則輸出中旗號〝OUT〞即設爲 1，否則爲 0。
- 在開始輸出脈波後輸出控制〝EN〞仍應保持爲 1，若其變爲 0，則立刻停止脈波之送出（輸出點變爲 OFF)，〝OUT〞旗號回到 0，其他狀態或資料則保持不變，但當其〝EN〞再度由 0 回到 1 時，卻會造成重置動作而當作一個新的開始，整個程序將重新來過。
- 若您欲暫停脈波輸出而又不被整個重新來過，則可利用暫停輸出〝PAU〞來暫停脈波之輸出。當〝PAU〞=1 時本指令會暫停脈波之送出（輸出點爲 OFF，〝OUT〞旗號回到 0，而其他狀態或資料均保持不變），等〝PAU〞由 1 變回 0 後，本指令會回到暫停前之狀態並由此開始繼續脈波之輸出動作。
- 在脈波輸出當中，本指令每次掃描到時仍會去抓取脈波頻率 Fr 及脈波數 PC 之數值，因此只要脈波尚未送完，均可更改脈波頻率或輸出脈波數。但正反轉方向〝U/D〞之狀態只有在重置動作（〝EN〞由 0→1）時抓取一次便一直保持到送完或下一次重置爲止，也就是除重置瞬間外，〝U/D〞之變化對本指令無任何影響。
- 本指令主要在推動步進馬達，UY（正轉）和 DY（反轉）兩種方向之脈波以方便您控制步進馬達之正反轉動作。若您只需要單方向運轉，您可只指定 UY 或 DY 之其中之一（可省下一個輸出點），另一個空白不指定。此時本指令將不理會正反方向〝U/D〞之輸入狀態，輸出脈波將固定送往您所指定的那個輸出點。

FUN81 **D** PLSO	脈波輸出指令 （PULSE OUTPUT）	FUN81 **D** PLSO

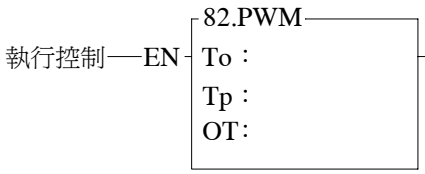
- 當 MD=1 時，控制方式爲正／反轉（DIR=1，正轉；DIR=0，反轉）結合脈波信號（CK）輸出方式控制。控制說明與上述相同。
- 本指令只能使用一次，且 UY（CK）和 DY（DR）必須爲 PLC 主機上之電晶體輸出點。
- 本指令之輸出脈波數 PC 在 16 位元指令時之有效範圍爲 0～32767，在 32 位元（**D**）指令時則爲 0～2147483647。若 PC 值=0 時當作無限脈波數，本指令將無限制地送出脈波而 HO 值和〝DN〞旗號則永遠爲 0。而脈波頻率 Fr 之有效範圍則爲 8～2000。無論 PC 或 Fr，若其值超出上述範圍即爲錯誤，本指令將不執行且將錯誤旗號〝ERR〞設爲 1。



- 本範例控制步進馬達先以 100Hz 之速度向前（正轉）走 80 個脈波（步），然後再以 50Hz 之速度反方向退後 40 個脈波。注意正反方向、頻率 Fr 及脈波數 PC 要在重置（〝EN〞由 0→1）前就要先準備好。



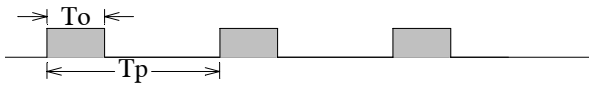
FUN82 PWM	脈波寬度調變 （PULSE WIDTH MODULATION）	FUN82 PWM



To ：指定脈波 ON 的寬度（0～32767mS）
Tp ：指定脈波之週期（1～32676mS）
OT：指定調變脈波之輸出點

運算元＼範圍	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	主機上之 Yn	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	0 | 32767
To		○	○	○	○	○	○	○	○	○	○	○	○	○
Tp		○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○													

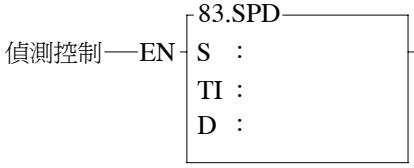
● 當執行控制〝EN〞=1 時，將週期爲 Tp，〝ON〞波寬爲 To 之方波送到輸出點 OT 去，而 OT 必須爲主機上之電晶體輸出點。當〝EN〞=0 時，輸出點爲不動作（OFF）。



● To 和 Tp 之單位均爲 mS，解析度爲 1mS，To 之數值最小可爲 0（此時輸出點 OT 永遠 OFF）最大可等於 Tp（此時輸出點 OT 永遠 ON），若 To＞Tp 則爲錯誤，本指令即不執行，且錯誤旗號〝ERR〞變成 1。

● 本指令只能使用一次。

FUN83 SPD	速度偵測 (SPEED DETECTION)	FUN83 SPD

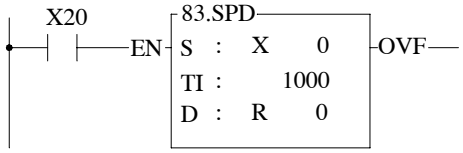


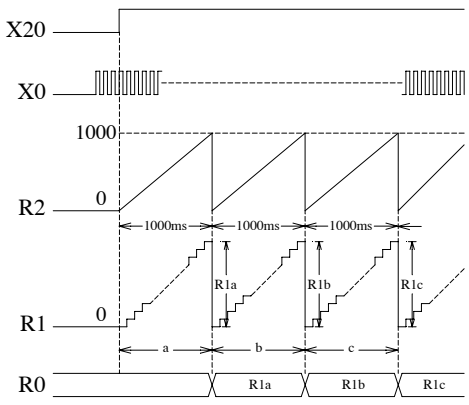
S ：欲偵測速度之脈波輸入點
TI：偵測之取樣時間（單位爲 mS）
D ：存放結果之暫存器號碼

運算元＼範圍	X	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	X0 ｜ X7	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	1 ｜ 32767
S	○													
TI		○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○	○	○	○	○*	○*	○	

- 本指令係利用 PLC 主機上之 X0～X7，8 個高速輸入點之中斷功能，而在一特定之取樣時間 TI 內計算某一輸入點之輸入脈波數，而間接求出接於該輸入點之迴轉裝置（如馬達）之轉速。
- 應用本指令時，齒盤齒數必須大於 60 齒；且輸入頻率總和必須小於 5KHz。
- 本指令之結果暫存器 D 總共使用了由 D 開始之連續 3 個 16 位元暫存器(D0～D2)，除 D0 爲存放計數結果外，D1 和 D2 用以存放計數經過值和累計取樣時間。
- 當執行偵測〝EN〞=1 時，開始計算 S 輸入點之脈波數，並先將之暫存在暫存器 D1 中，同時啓動取樣計時器（D2），等到 D2 值等於偵測取樣時間 TI 後停止計數，並將最後之計數值 D1 存入暫存器 D0 中，然後再重新開始另一次計數，時間到後再將新得到之計數值存入（蓋過）到暫存器 D0 中，如此週而復始地取樣計數，直到〝EN〞=0 方才停止。
- 因 D0 只有 16 位元最多只能計數到 32767，若取樣時間過長，且輸入脈波過快致使計數值超過 32767，則溢位旗號發生，計數動作停止。
- 因取樣時間 TI 已知，若迴轉裝置每轉一圈產生 n 個脈波，則可利用下式求出其轉速。

$$N = \frac{(D0)\times 60}{n\times TI}\times 10^3 \quad (rpm)$$

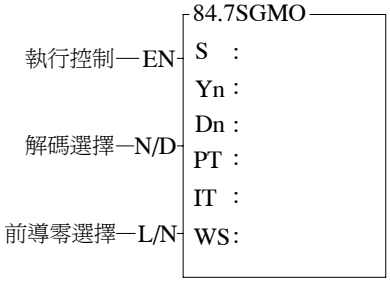




- 如上圖範例，若迴轉裝置每轉一圈產生 60 個脈波（n=60），而 R0 讀值爲 200，則該迴轉裝置每分鐘轉速 N 如下：

$$N = \frac{(200)\times 60}{60\times 1000}\times 10^3 = 200 \ rpm$$

FUN84 7SGMO	7 段顯示器模組便利指令 （功能簡述）	FUN84 7SGMO

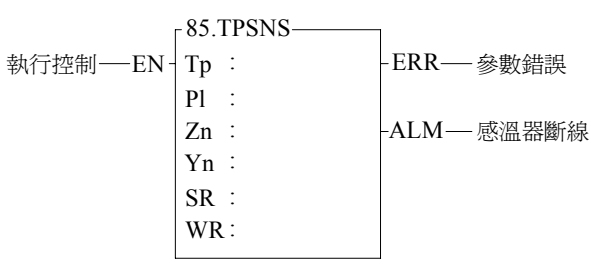


S ：欲作內容顯示之暫存器號碼
Yn：顯示模組之輸出位址號碼
Dn：欲顯示之位數（字數）
PT：小數點亮點指定
（非解碼顯示時無效）
IT ：亮度指定
WS：運算暫存器

運算元＼範圍	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Y0 ｜ Y240	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	正數 16/32 位元
S		○	○	○	○	○	○	○	○	○	○	○	○	○
Yn	○													
Dn		○	○	○	○	○	○	○	○	○	○	○	○	1-8
PT		○	○	○	○	○	○	○	○	○	○	○	○	0-FFH
IT		○	○	○	○	○	○	○	○	○	○	○	○	1-16
WS			○	○	○	○	○	○		○	○*	○*	○	

- 本指令爲 7 段顯示器模組（FB-7SG 1/2/3）之專用輸出指令，本指令採用填表方式用以指定欲顯示之內容位址、顯示之字數、亮度、小數點位置及指定數字顯示或獨立點 LED 顯示，及作數字顯示時是否零前導等指定，可大幅縮減程式設計時間及簡化程式。
- 本指令之詳細說明及範例請參考第 17 章〝FB-7SG 七段 LED 顯示器模組〞之敘述。

FUN85 TPSNS	溫度模組溫度量測便利指令 （功能簡述）	FUN85 TPSNS



Tp ：感溫器選擇，可選擇 J 或 K 熱電耦

Pl ：溫度模組之電壓範圍與極性設定

Zn ：溫度點數選擇

Yn ：溫度模組之多工輸出點起始號碼

SR ：存放溫度量測值之起始暫存器號碼

WR：本指令所需使用之工作暫存器起始號碼

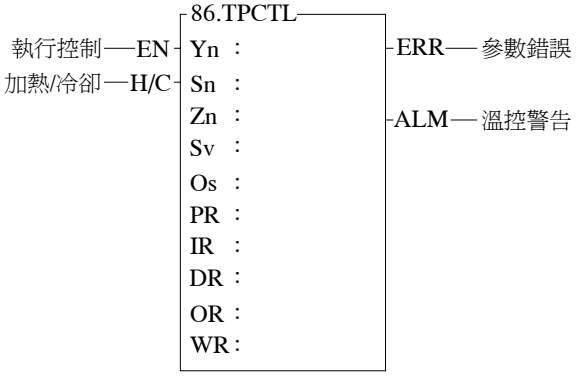
運算元＼範圍	Y	HR	ROR	DR	K
	Y0 ｜ Y255	R0 ｜ R3839	R5000 ｜ R8071	D0 ｜ D3071	
Tp					0～1
Pl					0～3
Zn					6，12，18，24
Yn	○				
SR		○	○*	○	
WR		○	○*	○	

指令功能簡述

- 本指令爲 FB-4AJ(K)××多工溫度量測模組之專用量測指令，透過本指令使用者可以用填表方式，輕易地取得多段之溫度量測值，以供監視或當作 PID 溫控之程控變數（Process Variable 簡稱 PV）。

- 本指令必須配合 FB-4AJ(K)××多工溫度量測模組使用，此處僅簡介本指令之功能，詳細之指令功能與說明及其用法與範例請參考第 20 章〝FB-PLC 之溫度量測及 PID 控制〞之敘述。

FUN86 TPCTL	PID 溫控便利指令 （功能簡述）	FUN86 TPCTL



運算元 \ 範圍	Y	HR	ROR	DR	K
	Y0 | Y255	R0 | R3839	R5000 | R8071	D0 | D3071	
Yn	○				
Sn					0～23
Zn					1～24
Sv		○	○*	○	
Os		○	○*	○	
PR		○	○*	○	
IR		○	○*	○	
DR		○	○*	○	
OR		○	○*	○	
WR		○	○*	○	

Yn ：PWM 溫控輸出起始號碼

Sn ：指定從第幾點開始執行 PID 溫控

Zn ：本指令所控制之 PID 溫控點數

Sv ：存放溫度設定值起始暫存器號碼

Os ：存放溫度偏差值起始暫存器號碼

PR ：存放增益設定值起始暫存器號碼

IR ：存放積分時間常數設定值起始暫存器號碼

DR ：存放微分時間常數設定值起始暫存器號碼

OR ：存放溫控數值輸出起始暫存器號碼

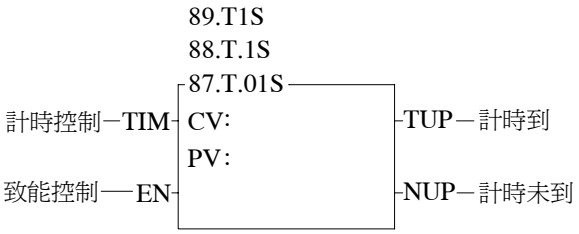
WR：本指令所需使用之工作暫存器起始號碼

指令功能簡述

- 本指令係將 FUN85（TPSNS）指令自多工溫度量測模組 FB-4AJ(K)××量測得到之溫度當作程控變數（Process Variable 簡稱 PV），並將使用者所設定之溫度設定值（Set Point 簡稱 SP），經由軟體 PID 數學式運算後，得到適當之輸出控制值，以控制溫度在使用者所期望之範圍內。

- 本指令必須配合多工溫度量測模組 FB-4AJ(K)××與 FUN85 之溫度量測便利指令來使用，此處僅簡介本指令之功能，詳細之指令功能與說明及其用法與範例請參考第 20 章〝FB-PLC 之溫度量測及 PID 控制〞之敘述。

FUN87 T.01S FUN88 T.1S FUN89 T1S	積算型計時器（0.01 秒，0.1 秒，1 秒） （ACCUMULATIVE TIMER）	FUN87 T.01S FUN88 T.1S FUN89 T1S



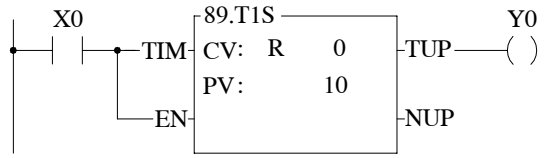
CV：存放計時時間（即現在值）之暫存器號碼

PV：計時器之設定或存放設定值之暫存器號碼

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C199	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	0 | 32767
CV		○	○	○	○	○	○	○	○	○*	○*	○	
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

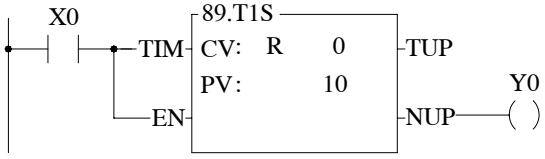
● 本指令之工作原理和一般計時器（T0～T255）相同，只是一般計時器只有一個計時控制〝EN〞輸入，在其輸入爲 1 時計時，爲 0 時則清除，每次輸入變動都會重新計時，無法累計。而本指令之計時需在致能控制〝EN〞=1 之條件下才允許，此時其計時控制〝TIM〞爲 1 時和一般計時器一樣，但爲 0 時，則不清除而保持現值。若需清除則使致能控制〝EN〞變爲 0 即可，若您不清除之，等計時控制〝TIM〞再度爲 1 時，計時器將由上次暫停時之計時值繼續累加。此外本指令尚有計時到〝TUP〞（計時到時爲 1，平時爲 0），及計時未到〝NUP〞（平時爲 1，計時到爲 0）兩個輸出，使用者可利用輸入和輸出組合出各種不同功能需求之計時器，如以下範例：

● ON DELAY ENERGIZING（ON 延遲供電）計時器：



- 係指該計時器輸出（本例爲 Y0）平常不供電，而在該計時器輸入控制（本例爲 X0）動作（ON）後延遲 10 秒後，輸出 Y0 才供應電能（ON）。

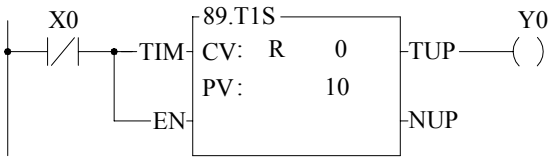
● ON DELAY DE-ENERGIZING（ON 延遲斷電）計時器：



- 係指該計時器輸出 Y0 平常就在供電狀態下，而在該計時器之輸入控制 X0 ON 後延遲 10 秒後，輸出才斷電（OFF）。

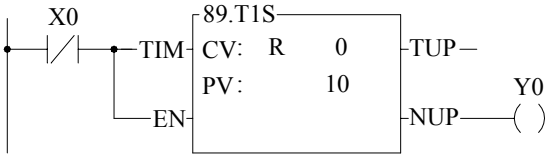
FUN87 T.01S FUN88 T.1S FUN89 T1S	積算型計時器（0.01 秒，0.1 秒，1 秒） （ACCUMULATIVE TIMER）	FUN87 T.01S FUN88 T.1S FUN89 T1S

● OFF DELAY ENERGIZING（OFF 延遲供電）計時器：



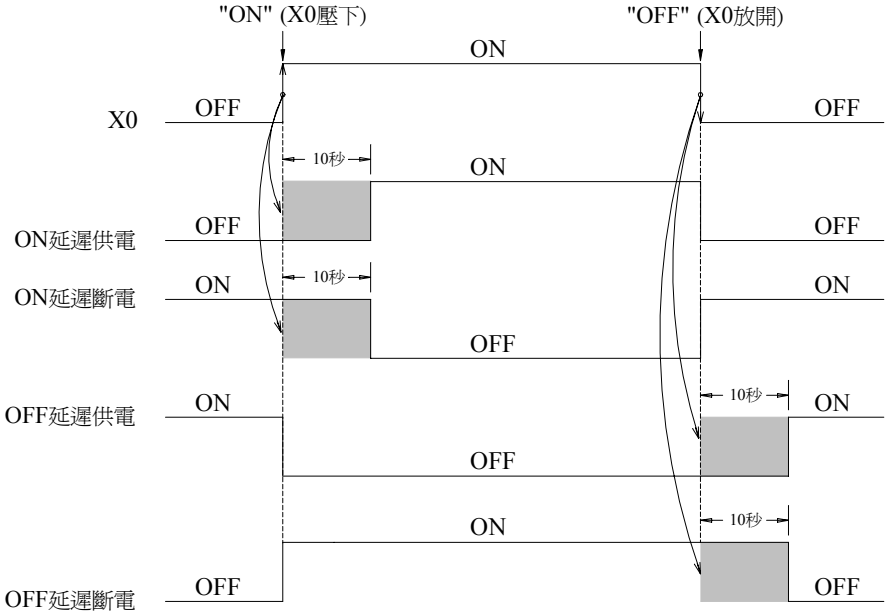
- 係指該計時器之輸出 Y0 平常在斷電狀態下，在該計時器之輸入控制 X0 OFF 後延遲 10 秒後，輸出 Y0 才供電（ON）。

● OFF DELAY DE-ENERGIZING（OFF 延遲斷電）計時器：

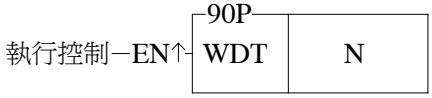


- 係指該計時器之輸出 Y0 平常在供電狀態下，在該計時器之計時控制 X0 OFF 後延遲 10 秒後輸出 Y0 才斷電（OFF）。

● 下圖爲以上四種計時器之輸入與輸出之對應結果。



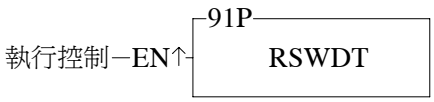
FUN90 P WDT	監控定時器（WATCHDOG TIMER）設定時間設定	FUN90 P WDT



N：監控定時器之設定時間。
其值祇能爲 50、60、70……120，
單位爲 10mS，即設定之時間範圍爲
(50～120)×10mS，即 0.5 秒～1.2 秒。

- 當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將監控定時器之設定時間改爲 Nx10ms。一旦設定後，Watchdog Timer (WDT) 即以此爲計時時間，若掃描時間超過此設定時間，PLC 將會停機不執行。
- WDT 設定時間主要是以系統應用上之安全考量而特別設計的，例如 PLC CPU 若突然損壞，無法執行程式或 I/O 更新時，經過 WDT 所設定之時間後，WDT 會自動從硬體上將 I/O 完全關閉以確保安全。在某些應用上若掃描時間太長亦可能造成某些安全上或不符控制要求之問題，亦可利用本指令設定您所要求之掃描時間極限值，超過您的設定值，PLC 會立刻停機，以確保安全。
- 設定值一旦設定後，即永遠保存，無需每次掃描均設定一次，因此本指令實用上應使用 P 指令。
- WDT 時間內定爲 0.25 秒。
- WDT 之工作原理請參考 FUN91（RSWDT）指令。

FUN91 P RSWDT	清除監控計時器 （RESET WATCHDOG TIMER）	FUN91 P RSWDT



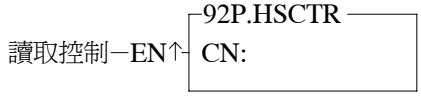
本指令無運算元

- 當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 WDT 計時器清除（亦即使 WDT 重新由 0 開始計時）。

- Watchdog Timer 之功能已在 FUN90（WDT 指令）中敘述，其原理如下：
  監控計時器一般均爲硬體單擊（One-Shot）計時器（不能用軟體作，否則 CPU 若當機，該計時器便失效，當然談不上能保護了），所謂單擊，意思就是只要您觸發計時器一下，該計時器之計時值便立刻清爲 0 再重新計時。若您在 WDT 開始計時後都未去觸發它，則 WDT 計時時間繼續累增至設定值 N 後 WDT 即動作，而將 PLC 停機。若您每次在 WDT 計時時間 N 尚未到達前就觸發 WDT 一次，則 WDT 永遠都不會發生，PLC 即利用此原理來確保系統安全，因爲 PLC 一般會在程式掃描和 I/O 更新後，進入系統服務（Housekeeping）時觸發 WDT 一次，若系統正常且掃描時間未超出 WDT 之設定時間 N，就一定來得及在 WDT 未動作前清掉 WDT 而使之不動作，但若 CPU 損壞而無法觸發 WDT 或掃描時間過長致來不及在 N 時間內觸發 WDT，WDT 即會動作而關掉 PLC。

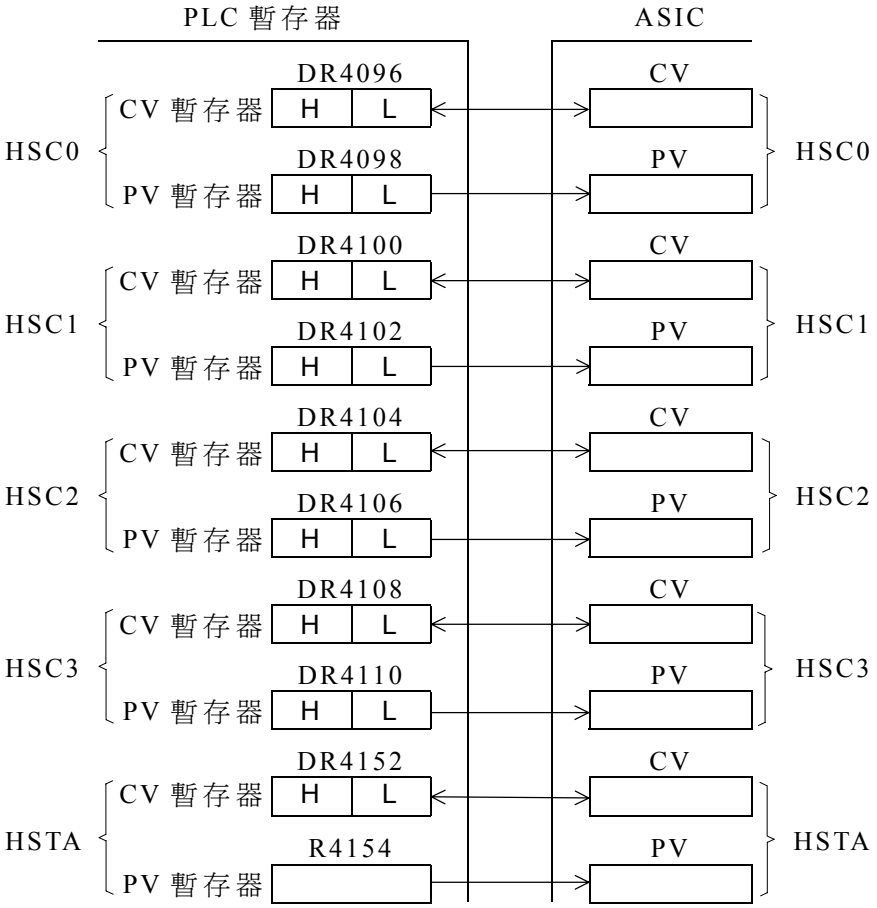
- 在某些應用下，您已設好了 WDT 時間（FUN90），而您的程式在某些情況下掃描時間可能會暫時超出 WDT 之設定時間，這情況是您所能預期且允許的，您當然不希望因此而 PLC 停機，此時您可用本指令觸發 WDT 一下即可避免 WDT 發生，此即本指令之主要目的。

FUN92 P HSCTR	硬體高速計數器現在值（CV）讀取	FUN92 P HSCTR



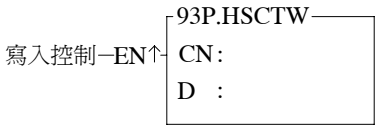
CN：硬體高速計數器號碼
0：HSC0 或 HST0
1：HSC1 或 HST1
2：HSC2 或 HST2
3：HSC3 或 HST3
4：HSTA

- FB-PLC 之硬體高速計數器 HSC0～HSC3 是四組 32 位元可作雙相或單相上下數之高速計數器，是由 ASIC 內部硬體電路所組成，能獨立進行計數、比較，發出中斷，而不會佔用 CPU 的時間。〔軟體高速計數器 HSC4～HSC7 是用中斷方式由 CPU 來處理，因此當使用個數多或計數頻率高時，整個 PLC 之性能（掃描速度）將嚴重劣化〕。因爲 HSC0～HSC3 之現在值 CV 是在 ASIC 內部之硬體電路中，使用者之控制（階梯圖）程式無法直接到 ASIC 去抓取，因此需利用本指令將硬體 HSC 之 CV 值讀出並放至控制程式能抓取到的暫存器去，以下爲 ASIC 中 CV、PV 和 PLC 內部相對應之 CV、PV 暫存器之安排。



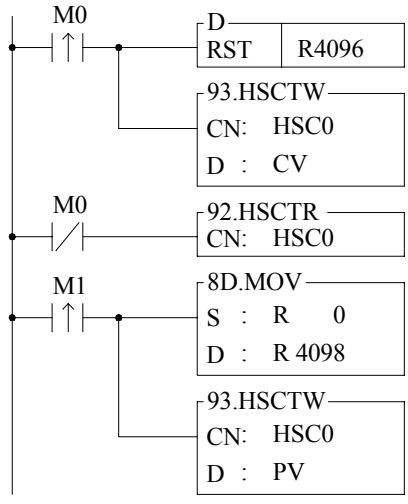
- 當讀取控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 ASIC 中，CN 所指定的 HSC 之 CV 值讀到 PLC 內部該 HSC 所對應之 CV 暫存器去。（即將 HSC0 之 CV 讀到 DR4096 或將 HSC1 之 CV 讀到 DR4100）。
- 雖然 ASIC 中之 PV 在 PLC 內部亦有 PV 暫存器與之對應，但卻不需讀取，因 ASIC 中之 PV 值是來自 PLC 內部之 PV 暫存器。
- HSTA 係以 0.1ms 爲時基之計時器，CV 之內容，代表經過多少個 0.1mS 時間。
- 詳細之應用請參考第 11 章〝FB-PLC 之高速計數器與高速計時器〞。

FUN93 P HSCTW	硬體高速計數器 CV 或 PV 值寫入	FUN93 P HSCTW



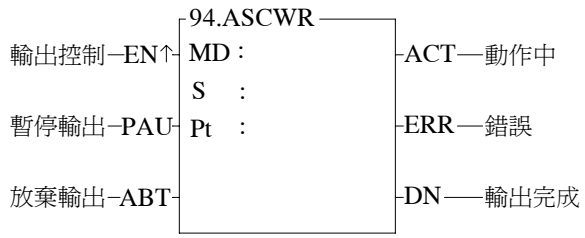
CN：硬體高速計數器號碼
0：HSC0 或 HST1
1：HSC1 或 HST2
2：HSC2 或 HST3
3：HSC3 或 HST4
4：HSTA
D ：寫入對象（0：表示 CV，1：表示 PV）

- 請先參閱 FUN92 有關 ASIC 中 HSC0～HSC3 與 HSTA 之 CV 或 PV 值和 PLC 內部相對應之 CV 暫存器和 PV 暫存器之關係。
- 當寫入控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 PLC 內部 CN 所指定的那個高速計數器之 CV 暫存器或 PV 暫存器之內容值寫到 ASIC 內部相對應 HSC 的 CV 或 PV 去。
- 一般應用常需寫入 PV，亦即將您所預設之設定值寫到 ASIC 中之 PV 去，當計數值到達您的設定值時，該計數器立即發出中斷，透過中斷服務程式您可作各種精密之計數或定位控制。
- FB-PLC 在電源斷電時會自動將當時 ASIC 內部 HSC0～HSC3 之現在值暫存器 CV 之值讀出再將之寫入 PLC 內部 HSC0～HSC3 之 CV 暫存器（具斷電保持功能）中，而在 PLC 復電時則會反向地將 PLC 內部之 CV 暫存器寫回 ASIC 內部之 CV 暫存器，因此每次 PLC 斷電再復電，ASIC 內部 HSC0～HSC3 之 CV 暫存器內容值將會自動回復到上次斷電前之數值，但若您的控制應用在復電時需清爲 0 或自某一特定值開始計數，您就必須利用本指令來作 ASIC 內部 HSC 之 CV 值寫入。
- HSTA 寫入不爲 0 之 PV 值，代表每 PV×0.1ms 會定時發出中斷；HSTAI 中斷副程式即爲定時中斷處理程式。
- 詳細之應用請參閱第 11 章〝FB-PLC 之高速計數器與高速計時器〞。



- 左圖程式，M0 由 0→1 時，將 HSC0 目前值清除爲 0，並透過 FUN93 寫入硬體 ASIC 中
- M0 爲 0 時，隨時讀出目前之計數值
- M1 由 0→1 時，將 DR0 之計數設定值搬至 DR4098，並透過 FUN93 寫入硬體 ASIC 中
- 當計數值等於 DR0 中之設定值時，立即執行 HSC0I 中斷處理副程式

FUN94 ASCWR	ASCII 檔案資料輸出 （ASCII FILE WRITE）	FUN94 ASCWR



MD：輸出模式選擇
=0，週邊為 Printer 時，利用 ASCWR 將 ASCII Editer所編之訊息經由通訊埠 1 傳送給 Printer。
=1 ，保留特定用途。
S ：檔案資料之起始暫存器號碼
Pt ：指令運作起始暫存器共佔用 8 個暫存器，其它地方不可重複使用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3967 | R4167	R5000 | R8071	D0 | D3071	0 | 1
MD													○
S	○	○	○	○	○	○	○	○	○	○	○	○	
Pt		○	○	○	○	○	○		○	○*	○*	○	

- 當 MD＝0 時，輸出控制〝EN↑〞由 0→1 時，將 S 開始之 ASCII 檔案資料送到通訊埠 1（Port1）去，直到送完整個檔案（遇到檔案結尾字元 END）為止。
- S 檔案資料可透過階梯圖大師（PROLADDER）套裝軟體上之 ASCII 編輯器來編輯（請參閱第 15 章〝ASCII 功能應用〞之說明），使用者亦可自行依接於 Portl 之 ASCII 週邊之特性，自己編輯所希望之報表或畫面的檔案資料，但檔案資料必須符合本指令規定之格式（詳述於第 15 章），否則本指令將中止輸出動作，並將錯誤旗號〝ERR〞設為 1。若整個檔案均正確且成功送出則輸出完成〝DN〞設為 1。
- 本指令之輸入為正緣觸發，一旦〝EN↑〞由 0→1 時，本指令開始執行後，就一直要等到整個檔案送完才算執行完成，這期間動作中旗號〝ACT〞均將一直維持 1，除非遇到暫停輸出、錯誤或放棄輸出才會變回 0。
- 本指令可重覆使用，但任一時間內只能有一個被執行（作輸出），使用者必須自行控制其執行之先後順序。
- 若本指令執行中，暫停輸出〝PAU〞若變為 1，則本指令暫停檔案資料之輸出，待暫停輸出〝PAU〞變為 0，本指令才又繼續先前檔案資料之輸出。
- 若本指令執行中，放棄輸出〝ABT〞若變為 1，則本指令中斷該輸出中檔案之執行，此時可接受下一個指令之執行。
- 使用 FUN94 (ASCWR）指令，必須將 CPU 主機上之 DIP 開關設定成 SW–1 OFF & SW–2 ON。
- 詳細之應用請參閱第 15 章〝ASCII 檔案功能之應用〞之說明

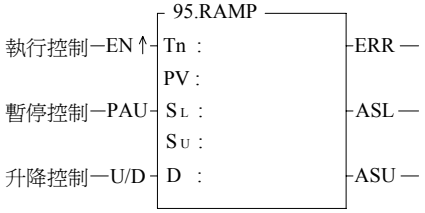
FUN94 ASCWR	ASCII 檔案資料輸出 （ASCII FILE WRITE）	FUN94 ASCWR

● 介面處理信號：

M1927：此信號由 CPU 產生，適用於 ASCWR MD：0
：ON，代表 Printer 之 RTS (接至 PLC 之 CTS)〝False〞，
亦即 Printer Not Ready 或異常。
：OFF，代表 Printer 之 RTS〝True〞, Printer Ready。
※利用 M1927 結合計時器可計時偵測 Printer 是否異常。

R4158：通訊參數設定（參考 12.6.2 節之說明）。

FUN95 RAMP	D／A 輸出緩升／緩降指令	FUN95 RAMP



Tn：緩升／緩降計時器號碼
PV：緩升／緩降計時器設定值（單位爲 0.01 秒）
　　或每 10mS 之增／減量設定值
SL：下限值（緩升初始值或緩降最終值）
SU：上限值（緩降初始值或緩升最終值）
D ：緩升／緩降值存放暫存器
D+1：工作暫存器
SU，SL 配合 AO 模組應用可爲正、負值

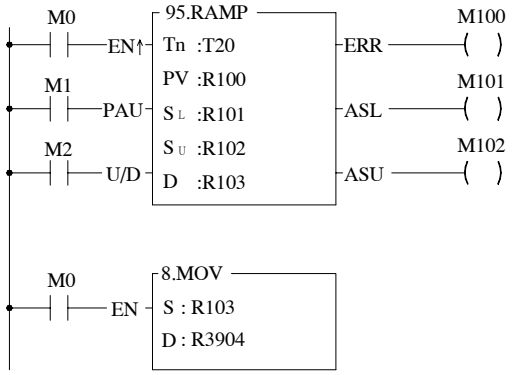
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	正負數 16 位元
Tn					○								
PV	○	○	○	○	○	○	○	○	○	○	○	○	○
SL	○	○	○	○	○	○	○	○	○	○	○	○	○
SU	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○	○*	○	

指令說明

- Tn 務必使用時基爲 0.01 秒之計時器，且在程式裏不得重複使用。
- 當 M1974=0 時，PV 爲緩升／緩降計時器設定值，單位爲 10ms（0.01 秒）。
- 當 M1974=1 時，PV 爲每 10mS 之增／減量設定值。
- 當輸入控制〝EN〞由 0→1 時，首先將計時器 Tn 復歸爲 0；
  如此時〝U/D〞=1，則表示緩升而將 SL 之值載入緩升／緩降值存放暫存器 D，以後每 0.01 秒等比例（SU−SL / PV，M1974=0）或以 PV 爲設定值（M1974=1）增加輸出量，並存放至暫存器 D，達計時器設定值時（M1974=0）或達上限設定值時（M1974=1），輸出值等於 SU，輸出〝ASU〞=1（達上限值）；
  如此時〝U/D〞=0，則表示緩降而將 SU 之值載入緩升／緩降值存放暫存器 D，以後每 0.01 秒等比例（SU−SL / PV，M1974=0）或以 PV 爲設定值（M1974=1））減少輸出量，並存放至暫存器 D，達計時器設定值時（M1974=0）或達下限設定值時，輸出值等於 SL，輸出〝ASL〞=1（達下限值）。
- 緩升／緩降（U/D）之決定是在輸入控制〝EN〞由 0→1 時，其他時間無效；祇要輸入控制〝EN〞由 0→1 即自動完成一次緩升／緩降控制。
- 如需暫停緩升／緩降動作，則必須使輸入控制〝PAU〞=1；當〝PAU〞=0 時，如果緩升／緩降動作未完成，則繼續完成未完成之動作。
- SU 之值必須大於 SL，否則緩升／緩降動作不執行，輸出〝ERR〞=1。
- 本指令使用緩升／緩降值存放暫存器 D 來存放輸出變化值；如使用 AO 模組來作速度控制時，可將緩升／緩降值存放暫存器 D 之值搬至 AO 輸出暫存器（R3904～R3967），而使啓動／結束之控制較爲平穩。
- 本指令除了使用緩升／緩降值存放暫存器 D 來存放輸出變化值外，另外使用暫存器 D+1 來作爲工作暫存器，所以程式裏不得再使用 D+1 這個暫存器。

FUN95 RAMP	D／A 輸出緩升／緩降指令	FUN95 RAMP

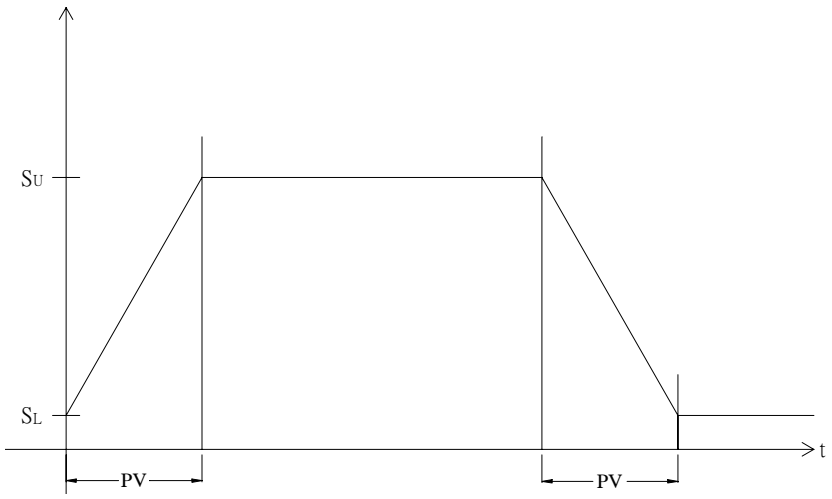
程式範例



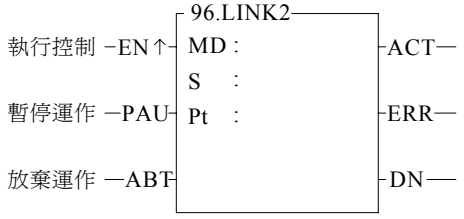
將緩升／緩降值搬至 AO 輸出暫存器 R3904 輸出

T20 ：緩升／緩降計時器號碼（0.01 秒時基計時器）
R100：緩升／緩降計時器設定值（M1974=0 時，單位爲 0.01 秒）
每 10mS 之增／減量設定值（M1974=1 時，無單位）
R101：下限值（緩升初始值或緩降最終值）
R102：上限值（緩降初始值或緩升最終值）
R103：緩升／緩降值存放暫存器
R104：工作暫存器

- 若 M1974=0，當輸入控制 M0 由 0→1 時，首先將計時器 T20 復歸爲 0，如此時 M2=1，則表示緩升而將 R101（下限）之值載入 R103，以後每 0.01 秒等比例（R102−R101 / R100）增加輸出量，並存放至暫存器 R103，達計時器設定值 R100 時，輸出值等於 R102，輸出 M102=1（達上限值）；如此時 M2=0，則表示緩降而將 R102（上限）之值載入 R103，以後每 0.01 秒等比例（R102−R101 / R100）減少輸出量，並存放至暫存器 R103，達計時器設定值 R100 時，輸出值等於 R101，輸出 M101=1（達下限值）。
- M1=1，暫停緩升／緩降動作。
- R102 之值必須大於 R101，否則緩升／緩降動作不執行，輸出 M100=1。



FUN96 LINK2	通訊埠 2（RS-485）通訊連線便利指令 （功能簡述）	FUN96 LINK2



MD：指令模式有 MD0～MD3 四種

S ：通訊程式起始暫存器

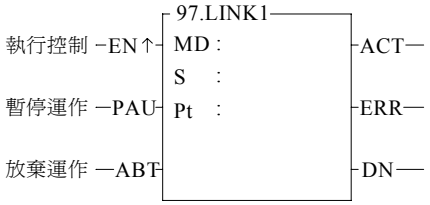
Pt ：指令運作起始暫存器

運算元＼範圍	HR	ROR	DR	K
	R0 ｜ R3839	R5000 ｜ R8071	D0 ｜ D3071	
MD				0～3
S	○	○	○	
Pt	○	○*	○	

● 當使用者之階梯圖程式中有使用到本指令，在 PLC 啓動（RUN）後，PLC 會自動將通訊埠 2（Port2）設爲〝階梯圖指令掌控界面〞並交給本指令來掌控（PLC Stop 時，Port2 又將回到〝標準界面〞不受本指令支配）。本指令共有四種指令模式 MD0～MD3，其中 MD0～MD2 三種指令模式均爲〝一般 LINK 網路〞，而 MD3 則爲〝高速 LINK 網路〞，以下爲各指令模式之功能簡述，各模式之詳細功能應用請參考第 13.1.2 小節之說明。

- MD0：永宏一般 CPU LINK 網路之主站（Master）指令模式
  任一 PLC 之使用者程式中有 FUN96：MD0 指令，則該 PLC 將變成永宏之一般 CPU LINK 網路之主站。主站 PLC 會依 FUN96 之通訊程式暫存器所預存之通訊對象與資料類別、號碼、長度…等指定，以〝永宏 FB-PLC 通訊協定〞之格式，寫入或讀取僕站 PLC 之資料，可作多達 254 台 PLC 間之資源共享。
- MD1：主動傳送 ASCII 資料指令模式
  在本指令模式下，FUN96 將主動地將預存於通訊程式暫存器內之二進制數據或 ASCII 碼資料（使用者須自行依外界 ASCII 週邊之通訊協定之格式存入通訊程式暫存器中），依序地傳送給 ASCII 週邊（諸如電腦、它牌 PLC、變頻器、字幕機…等以 ASCII 碼來通訊之智慧型週邊）。其運作方式可指定爲①僅傳送而不理會外界週邊之回應②傳送後需接收外界週邊之回應；當指定爲第②項〝傳然後收〞方式，則使用者必須依據接收之通訊協定，在 Ladder 程式裡處理接收到之 ASCII 回應訊息。
- MD2：隨時接收 ASCII 資料指令模式
  在本指令模式下，FUN96 將隨時等待接收外界 ASCII 週邊（諸如電腦、他牌 PLC、刷卡機、條碼機、電子磅秤…等以 ASCII 碼來通訊之智慧型週邊）所傳送過來之 ASCII 訊息，使用者再依所連接對象之通訊協定來解析該訊息，並將之還原爲 CPU 所能處理之二進位資料。其運作方式可指定爲①僅接收不回應，或②接收後再回應訊息兩種方式。回應訊息之方式，使用者只需將欲回應之二進制數據或 ASCII 資料依外界 ASCII 週邊之通訊協定格式存入通訊程式暫存器中，FUN96 便會將之回應給外界週邊。
- MD3：永宏高速 CPU LINK 網路之主站指令模式
  本模式和 MD0 模式最大之差異乃在於 MD3 之通訊反應時間遠低於 MD0，使用 MD3 模式之 CPU LINK 可達到①分散式控制②及時資料監控。

FUN97 LINK1	通訊埠 1（RS-232）通訊連線便利指令 （功能簡述）	FUN97 LINK1



MD：指令模式有 MD0～MD2 三種

S ：通訊程式起始暫存器

Pt ：指令運作起始暫存器

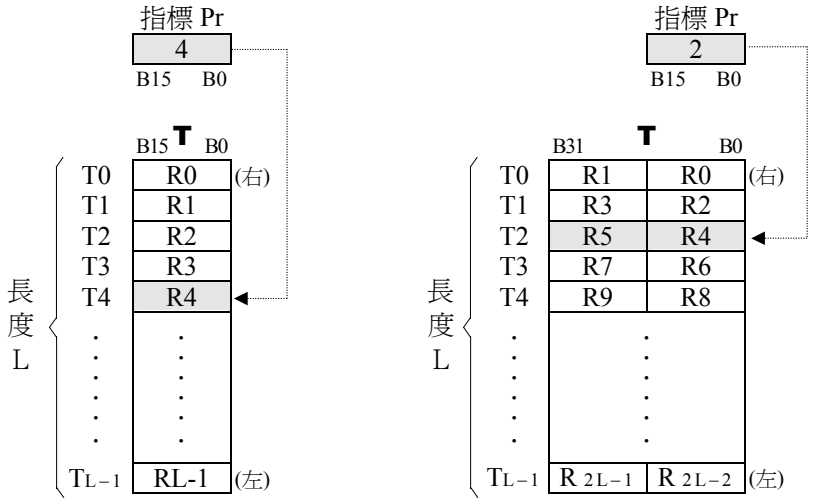
運算元＼範圍	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D3071	
MD				0～2
S	○	○	○	
Pt	○	○*	○	

● 本指令之 MD0～MD2 三種指令模式之運作方式與用法和 FUN96 之 MD0～MD2 完全相同，請參考 FUN96（LINK2）與第 13 章之說明。

列　表　(T A B L E)　指　令

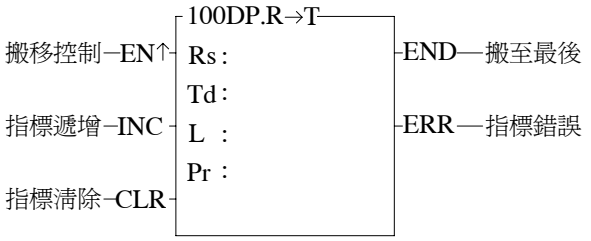
100. R→T	107. T_FIL
101. T→R	108. T_SHF
102. T→T	109. T_ROT
103. BT_M	110. QUEUE
104. T_SWP	111. STACK
105. R-T_S	112. BKCMP
106. T-T_C	

- 列表是 2 個以上連續之暫存器（16 或 32 位元）所組成，組成列表之暫存器個數稱爲列表的長度 L（Length），列表指令運算每次均以列表之一個暫存器爲單位（即 16 或 32 位元資料）。
- 列表指令主要在處理列表和暫存器或列表和列表間之資料處理，諸如搬移、拷貝、比較、搜尋等，是極爲方便和重要之應用指令。
- 在列表指令運作中通常需要有一指引器來指定列表中之某一暫存器當作運算對象，此指引器我們稱之爲指標 Pr（Pointer）。無論 16 或 32 位元列表指令其指標都只是一個 16 位元之暫存器。指標之有效範圍爲 0～L–1，分別用以對應（指引）至列表之暫存器 T0～T_{L-1}（共 L 個），以下爲 16 位元及 32 位元列表之示意圖。
- 在列表運算中有左、右移或旋轉，我們定義高序號者爲左，低序號者爲右，如下圖示。



16 位元列表　　32 位元列表

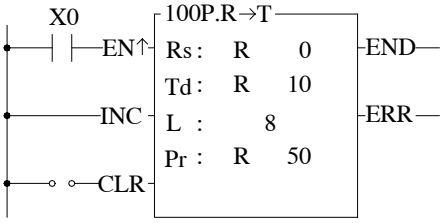
FUN100 DP R→T	暫存器→列表搬移 (REGISTER TO TABLE MOVE)	FUN100 DP R→T



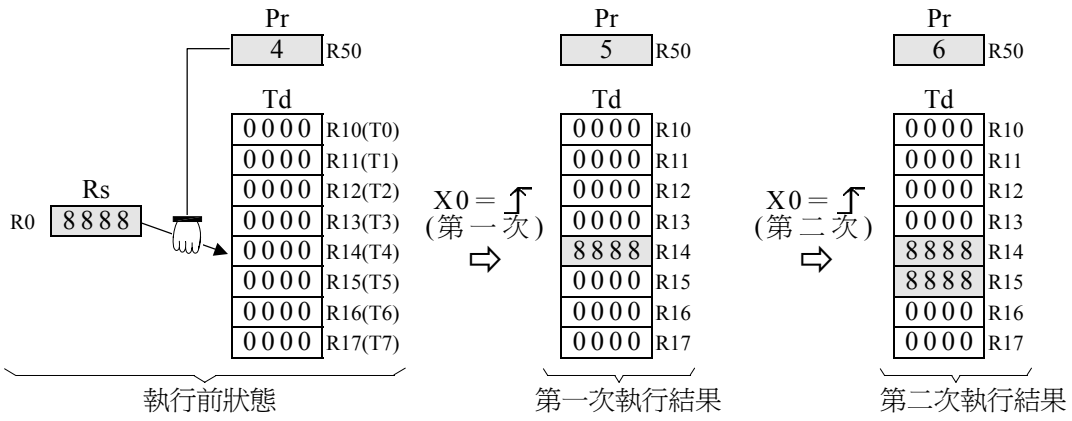
Rs：來源資料或其暫存器號碼
Td：目的列表之起頭暫存器號碼
L ：目的列表之長度
Pr：指標暫存器號碼
Rs，Td 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正、負數	V、Z
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~2048	
Pr		○	○	○	○	○	○		○	○*	○*	○		

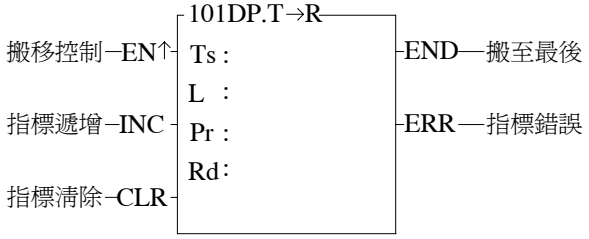
- 當搬移控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將來源暫存器 Rs 的內容寫到目的列表 Td（長度爲 L）中指標 Pr 所指之暫存器 Tdpr 去。在執行搬移前本指令會先檢視指標清除〝CLR〞之輸入信號，若〝CLR〞爲 1，則會先將指標 Pr 之內容清除爲 0 後再作搬移。在作完搬移動作後接著檢視 Pr 值，若 Pr 值已達 L–1（已指在列表之最後一個暫存器）則將搬至最後旗號〝END〞設爲 1後結束本指令之執行；若 Pr 小於 L–1，則再檢視指標遞增〝INC〞之狀態，若〝INC〞爲 1，則再將 Pr 加 1 後才結束執行。此外，指標清除〝CLR〞可單獨執行，不受其他輸入影響。
- 指標之有效範圍爲 0～L–1，超出此範圍則指標錯誤〝ERR〞設爲 1，且本指令不執行。



- 左圖程式例假設剛開始指標 Pr=4，列表 Td 內容全部爲 0，而 Rs 值爲 8888，下圖爲當 X0 連續由 0→1 變換 2 次所得之運算結果。
- 因 INC 爲 1，故每執行一次 Pr 即加 1 一次。



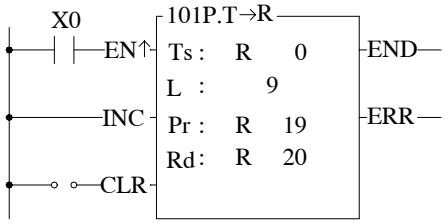
FUN101 **D** **P** T→R	列表→暫存器搬移 （TABLE TO REGISTER MOVE）	FUN101 **D** **P** T→R



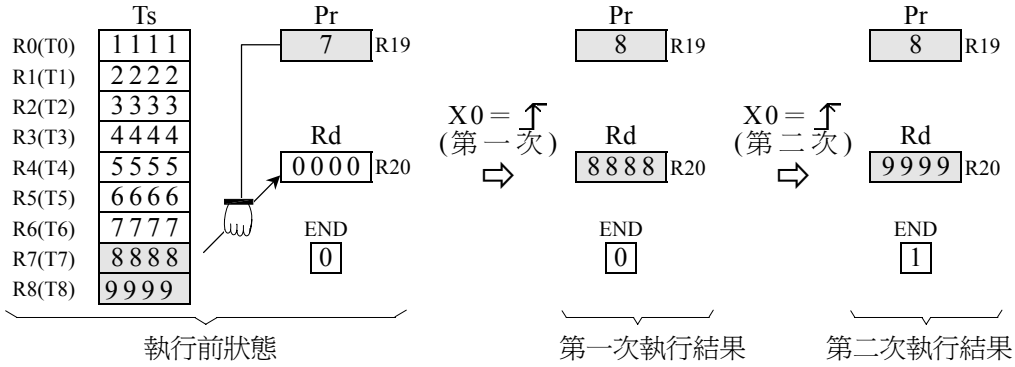
Ts ：來源列表之起頭暫存器號碼
L ：來源列表之長度
Pr ：指標暫存器號碼
Rd：目的暫存器之起頭號碼
Ts，Rd 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正、負數	V、Z
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○		
Pr		○	○	○	○	○	○		○	○*	○*	○	2~2048	
Rd		○	○	○	○	○	○		○	○*	○*	○		○

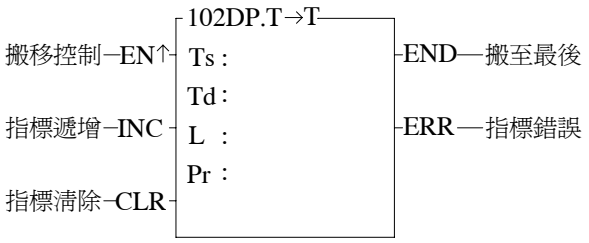
- 當搬移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將來源列表 Ts（長度爲 L）中指標 Pr 所指之暫存器 Tspr 之內容值寫到目的暫存器 Rd 去。在執行搬移前本指令會先檢視指標清除〝CLR〞之輸入信號，若〝CLR〞爲 1，則會先將指標 Pr 之內容清除爲 0 後再作搬移。在作完搬移動作後接著檢視 Pr 值，若 Pr 值已達 L–1（已指在列表之最後一個暫存器）則將搬至最後旗號〝END〞設爲 1 後結束本指令之執行；若 Pr 小於 L–1，則再檢視指標遞增〝INC〞之狀態，若〝INC〞爲 1，則再將 Pr 加 1 後才結束執行。此外，指標清除〝CLR〞可單獨執行，不受其他輸入影響。
- 指標之有效範圍爲 0～L–1，超出此範圍則指標錯誤〝ERR〞設爲 1，且本指令不執行。



- 左圖程式例假設剛開始指標 Pr=7，而 Ts 和 Rd 內容如下圖左所示，當 X0 連續由 0→1 變換 2 次後，可得到下圖右方之兩個結果。
- 第二次執行時指標已至最後，故不再增加。



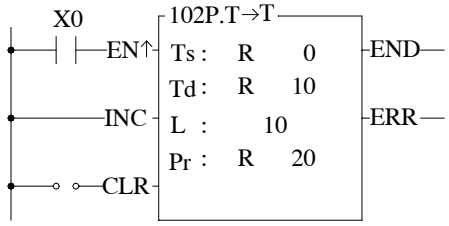
FUN102 DP T→T	列表→列表搬移 （TABLE TO TABLE MOVE）	FUN102 DP T→T



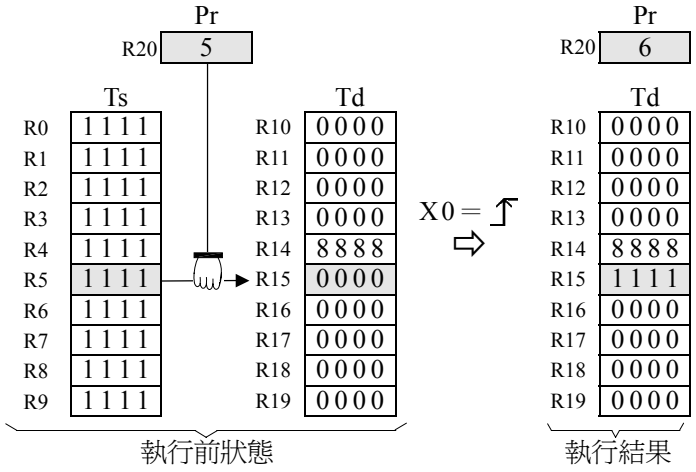
Ts：來源列表之暫存器起頭號碼
Td：目的列表之暫存器起頭號碼
L ：列表（Ts 和 Td）之長度
Pr：指標暫存器號碼
Ts，Td 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 2048	V、Z
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

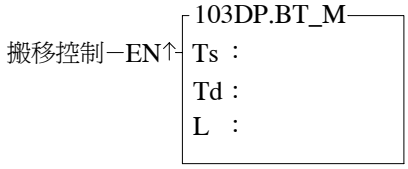
- 當搬移控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將來源列表中指標 Pr 所指的那一個暫存器資料 Tspr 搬到目的列表中同樣是 Pr 所指之暫存器 Tdpr 去。在執行搬移前本指令會先檢視指標清除輸入〝CLR〞信號，若其爲 1，會先清除 Pr 爲 0 後再搬移（此時爲 Ts0→Td0）。在作完搬移動作後接著檢視指標 Pr 之值，若 Pr 值已達 L–1（已指到列表之最後一對暫存器），則將搬至最後旗號〝END〞設爲 1 後結束指令之執行；若 Pr 值小於 L–1，將再檢視指標遞增〝INC〞之狀態，若〝INC〞爲 1，則再將 Pr 值加 1 後才結束指令之執行。此外，指標清除〝CLR〞可單獨執行，不受其他輸入影響。
- 指標之有效範圍爲 0～L–1，超出此範圍則指標錯誤〝ERR〞設爲 1，且本指令不執行。



- 左圖程式範例起始狀態如下圖左之執行前狀態，當 X0 由 0→1 時，將得到下圖右之結果，Ts 爲來源列表不受指令執行之影響。



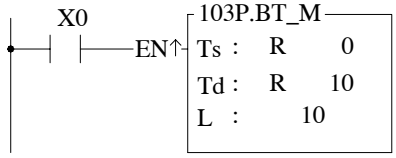
FUN103 **D** **P** BT_M	整個列表搬移 （BLOCK TABLE MOVE）	FUN103 **D** **P** BT_M



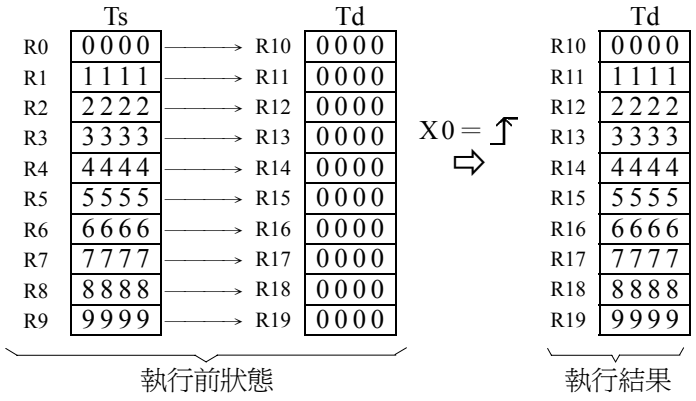
Ts：來源列表之起頭暫存器號碼
Td：目的列表之起頭暫存器號碼
L ：來源和目的列表之長度
Ts，Td 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 本指令來源和目的列表長度相同，且在一次指令執行中即將整個 Ts 列表資料全部搬至 Td 中，故無需指標來指定列表中之某一暫存器。
- 當搬移控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將來源列表 Ts（長度爲 L）的所有資料整個一次搬至相同長度之目的列表 Td 去。
- 因本指令在每次執行中是整個列表一次搬完，若列表長度較長時，將會耗費較多的時間，實用上應使用**P**指令，以免每次掃描均重複同樣之搬移動作而浪費時間。



• 左圖程式範例，假設 Ts 和 Td 列表之狀態如下圖左之執行前狀態，當 X0 由 0→1 時，可得到下圖右之執行結果。



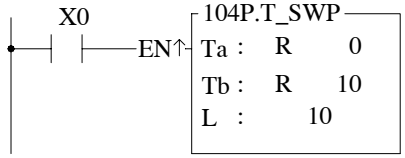
FUN104 DP T_SWP	整個列表互換 （BLOCK TABLE SWAP）	FUN104 DP T_SWP

搬移控制─EN↑ 104DP.T_SWP
Ta：
Tb：
L ：

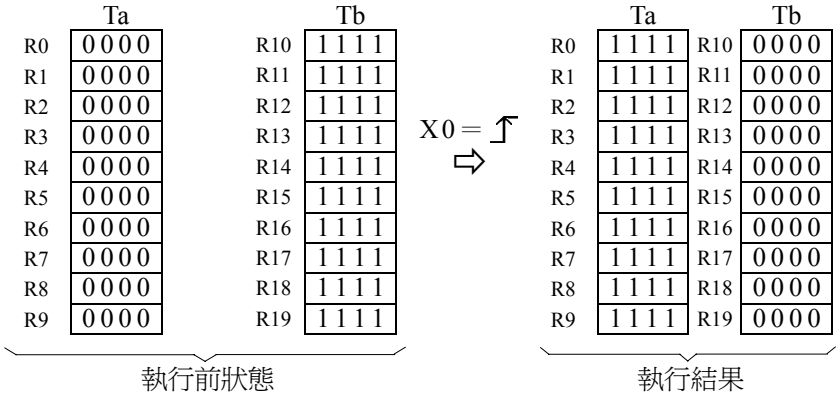
Ta ：列表 a 之暫存器起頭號碼
Tb ：列表 b 之暫存器起頭號碼
L ：列表 a 和 b 之長度
Ta，Tb 可結合 V、Z 作間接定址應用

運算元＼範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ta	○	○	○	○	○	○	○	○*	○*	○		○
Tb	○	○	○	○	○	○	○	○*	○*	○		○
L						○			○*	○	○	

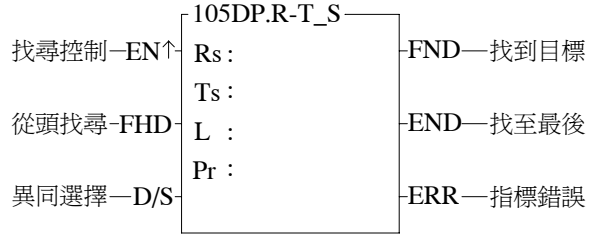
- 本指令列表 a，b 內容要對換（SWAP），故列表長度必相同，且列表本身必須是可寫入之暫存器。因其爲一次執行即全部互換完畢，故無需指標。
- 當搬移控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將列表 Ta 和列表 Tb 的內容整個互換。
- 本指令因一次互換完畢，若列表長度過長，將會耗費較多的時間，實用上應使用 P 指令。



• 左圖程式範例假設 Ta 和 Tb 之起始狀態如下圖左執行前之情況，當 X0 由 0→1 時，可得到下圖右執行之結果。



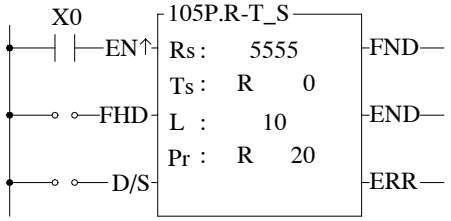
FUN105 DP R-T_S	暫存器對列表找尋異同 （REGISTER TO TABLE SEARCH）	FUN105 DP R-T_S



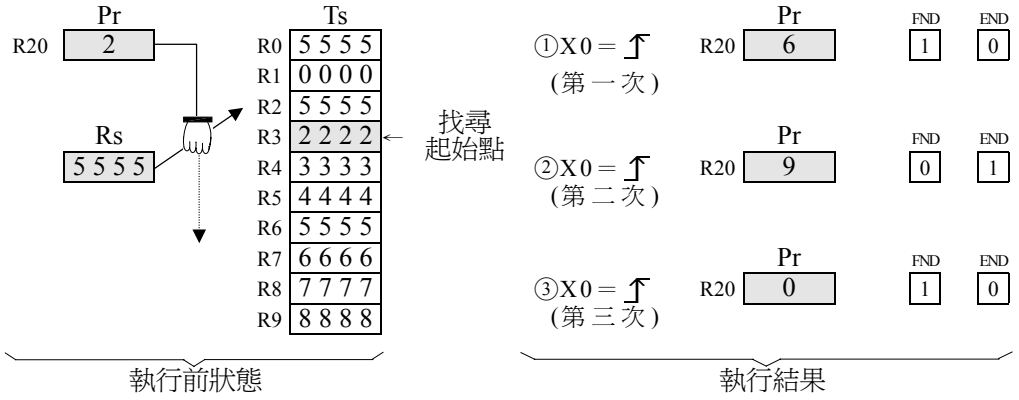
Rs：找尋之來源（樣本）資料或其暫存器號碼
Ts：被找尋之列表起頭暫存器號碼
L ：列表長度
Pr：指標，用以記錄目標所在之位置值
Rs，Ts 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正、負數	V、Z
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		

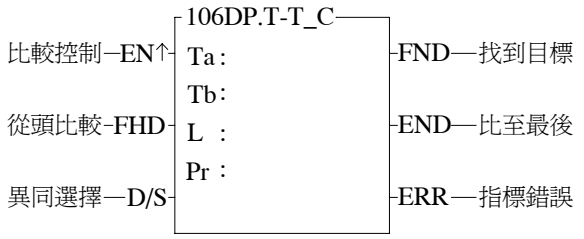
- 當找尋控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，自列表 Ts 之開頭第一個暫存器開始（〝FHD〞=1 或 Pr 值已達 L–1 時）或自列表中當時指標所指那個暫存器之下一個暫存器 Tspr+1 開始（〝FHD〞=0 同時 Pr 值小於 L–1）往下找尋和樣本資料 Rs 不同（D/S=1 時）或相同（D/S=0 時）之暫存器。若找到目標（不同或相同者），則立即停止找尋動作，並將該目標在列表之位置序號值存到指標 Pr 去，同時將找到目標旗號〝FND〞設爲 1 後結束本指令之執行。當找到列表之最後一個暫存器時，無論是否找到目標均將結束該次指令執行，並將找至最後旗號〝END〞設爲 1，而 Pr 值則停在 L–1。當本指令下次再度被執行時，Pr 將會自動循環至列表之最開頭（Pr=0）開始往下找尋。
- 指標值之有效範圍爲 0～L–1，若其值超出此範圍，則指標錯誤旗號〝ERR〞變爲 1，且本指令不執行。



- 左圖程式範例在列表中找尋數值同爲 5555 之暫存器（因 D/S=0，爲找相同），執行前指標指在 R2，但開始找尋點是 Pr+1（即 R3 開始），在 X0 連續 3 次由 0→1 動作後，可得到如下圖①、②、③三個結果。



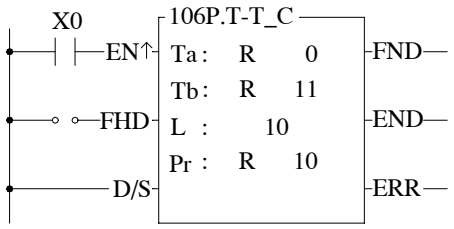
FUN106 DP T-T_C	列表對列表比較異同 (TABLE TO TABLE COMPARE)	FUN106 DP T-T_C



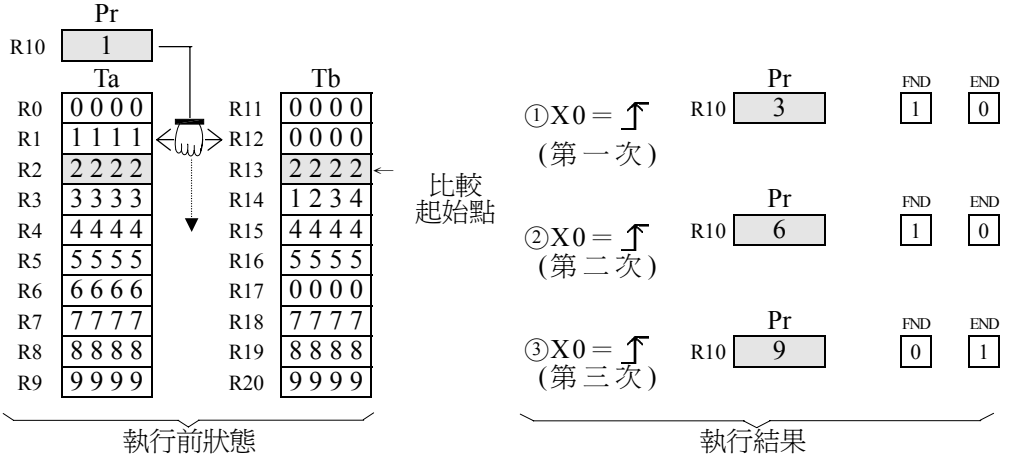
Ta ：列表 a 之暫存器起頭號碼
Tb ：列表 b 之暫存器起頭號碼
L ：列表 a 和 b 之長度
Pr ：指標，用以記錄目標所在之位置值
Ta，Tb 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ta	○	○	○	○	○	○	○	○	○	○	○	○		○
Tb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

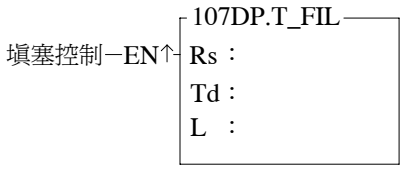
- 當比較控制〝EN〞=1 或〝EN↑〞(P 指令）由 0→1 時，自 Ta 和 Tb 兩列表中之最開頭那對暫存器（Ta0 和 Tb0）開始（〝FHD〞=1 或 Pr 值已達 L−1 時）或自當時 Pr 所指那對暫存器的下一對暫存器（Tapr+1 和 Tbpr+1）開始（〝FHD〞=0 同時 Pr 值小於 L−1）往下雙雙成對比較找尋內容值不同（〝D/S〞=1 時）或相同（〝D/S〞=0 時）之暫存器對（Pair），當找到目標（不同或相同者）後立即停止比較找尋，同時將該對目標在列表中之位置序號值存到指標 Pr 去，並將找到目標旗號〝FND〞設爲 1，然後結束本指令之執行。當找到列表之最後一對暫存器，無論其是否爲所要找尋之目標，均將結束本指令之執行，同時將比至最後旗號〝END〞設爲 1，而指標值則停在 L−1。而當本指令下一次再度被執行時，Pr 將會自動循環至列表之最開頭開始往下找起。
- 指標 Pr 之範圍爲 0～L−1，在運作中應避免更動到 Pr 值，以免影響其正確之比較找尋，若 Pr 值超出此範圍，則指標錯誤旗號〝ERR〞變爲 1，且本指令不執行。



- 左圖程式範例爲自指標所指之下一個暫存器開始往下比較找尋（因〝FHD〞爲 0）兩列表中資料不相同之暫存器對（因〝D/S〞=1）。剛開始 Pr 指在 Ta1 和 Tb1，雖然兩列表中分別在列表位置 1、3、6 三處有資料不同，但因其非從頭比較，故本指令將先找到位置 3，下圖右顯示 X0 由 0→1 變換 3 次之結果。



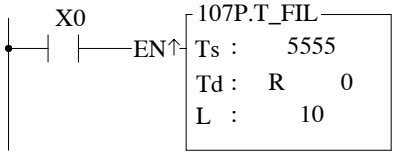
FUN107 DP T_FIL	列表填塞 （TABLE FILL）	FUN107 DP T_FIL



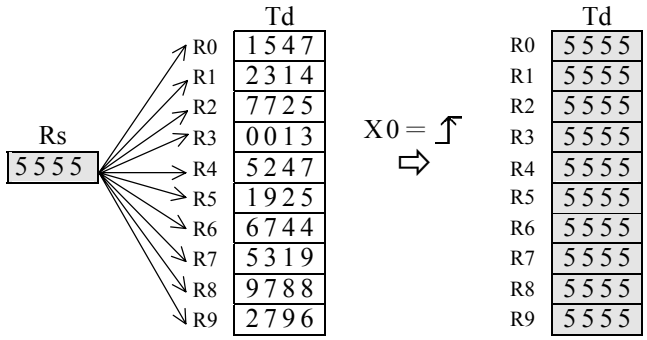
Rs：欲填入列表之來源資料或其暫存器號碼
Td：列表之起頭暫存器號碼
L ：列表之長度
Rs，Td 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正、負數	V、Z
Ts	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	

- 當填塞控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將 Rs 之資料填寫到列表 Td 中所有的暫存器中。
- 本指令主要用於列表之清除（填 0）或一致化（全部填相同值）用，實用上應使用 P指令。

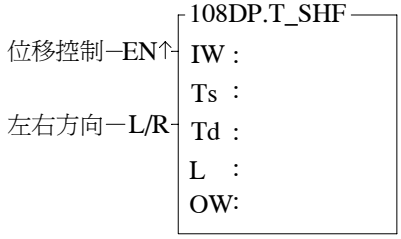


• 左圖程式範例，將列表 Td 全部填入 5555，如下圖結果。



執行前狀態　　執行結果

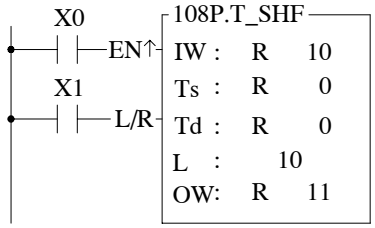
FUN108 DP T_SHF	列表位移 （TABLE SHIFT）	FUN108 DP T_SHF



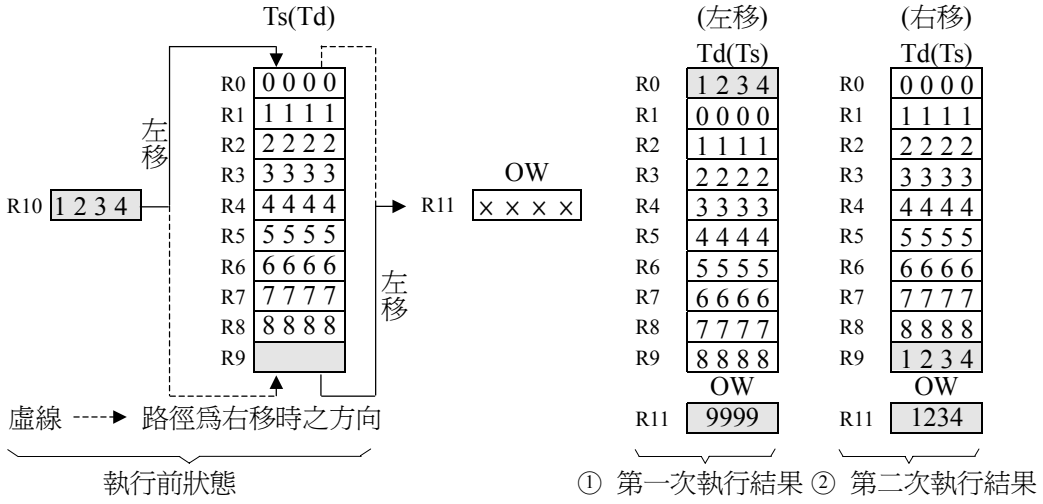
IW ：填補位移空位之資料或其暫存器號碼
Ts ：被位移之來源列表
Td ：存放結果之目的列表
L ：列表 Ts 和 Td 長度
OW：存放自列表移出資料之暫存器號碼
Ts，Td 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16 或 32 位元 正、負數	V 、 Z
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
OW		○	○	○	○	○	○		○	○*	○*	○		

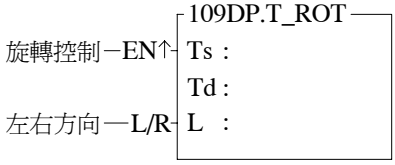
● 當位移控制〝EN〞=1 或〝EN↑〞（ P指令）由 0→1 時，將列表 Ts 之資料全部取出，整個向左（〝L/R〞=1 時）或向右（〝L/R〞=0 時）位移一個位置，因位移造成之空位，以 IW 填補之，再將此位移過並填補之結果，寫到列表 Td 去，而因位移而擠出之資料則寫到 OW 去。



• 左圖程式範例 Ts 和 Td 相同，因此就是把列表自己位移後再寫回自己（列表必須爲可寫入之暫存器），如下圖執行前之狀態，先執行一次左移（使 X1=1，再使 X0 由 0→1），然後再作一次右移（使 X1=0，再使 X0 由 0→1），將可得到下圖右方之兩個結果。



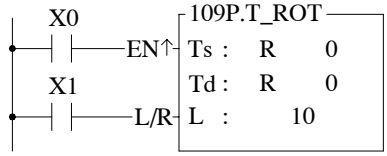
FUN109 DP T_ROT	列表旋轉 （TABLE ROTATE）	FUN109 DP T_ROT



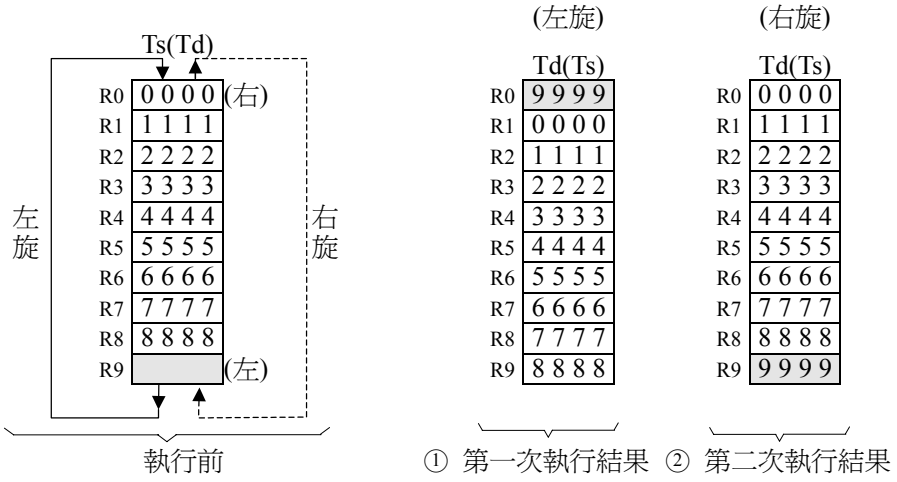
Ts ：被旋轉之來源列表
Td：存放旋轉結果之列表
L ：列表 Ts 和 Td 之長度
Ts，Td 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

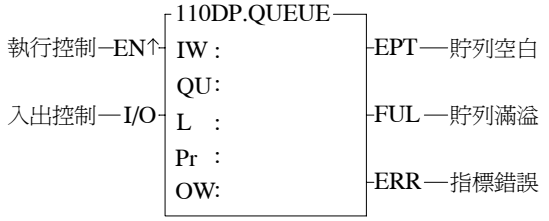
● 當旋轉控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將列表 Ts 之資料取出後向左（〝L/R〞=1 時）或向右（〝L/R〞=0 時）旋轉一個位置後，將此旋轉過之結果寫到 Td 去。



• 左圖程式例，Ts 和 Td 相同，故是將列表自己取出旋轉後再寫回自己，如下圖執行前之狀態，先執行一次左旋（使 X1=1，再使 X0 由 0→1），然後再作一次右旋（使 X1=0，再使 X0 由 0→1）後，可得到下圖右方之①、②兩個結果。



FUN110 DP QUEUE	貯列 （QUEUE）	FUN110 DP QUEUE

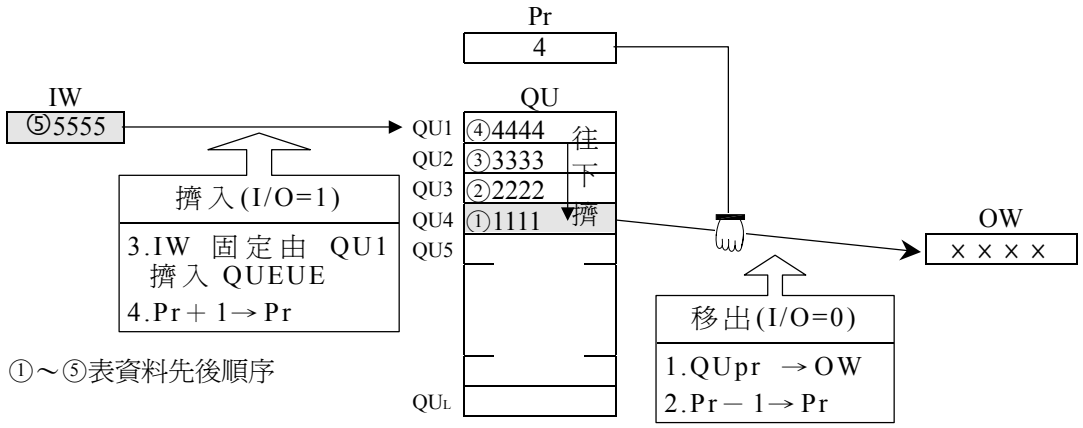


IW ：擠入貯列之資料或其暫存器號碼
QU ：貯列之起頭暫存器號碼
L ：貯列之長度
Pr ：指標暫存器號碼
OW：接收自貯存器移出資料之暫存器號碼
QU 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16 或 32 位元 正、負數	V 、 Z
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
QU		○	○	○	○	○	○		○	○	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

● 貯列（QUEUE）亦屬列表之一種，其有別於一般列表的是其貯列暫存器序號是由 1～L 而非 0～L–1，亦即 QU_1～QU_L，分別以指標 Pr=1～L 來對應，而指標 Pr=0 則用以表示該貯列爲空白。

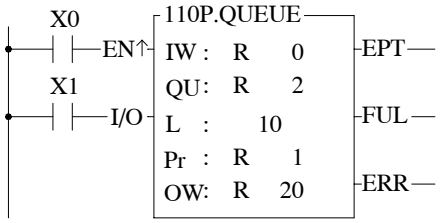
● 貯列（QUEUE）是一種先進先出裝置，即最先擠入（PUSH）貯列之資料，在移出（POP）時要最先移出。本指令之貯列是由 QU 暫存器開始之連續 L 個 16 位元或 32 位元（D指令）暫存器所組成。



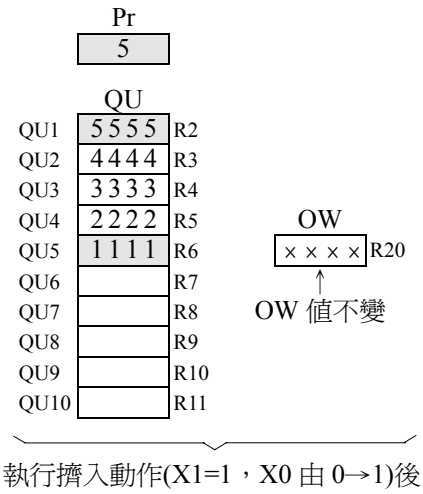
● 貯列指令之動作係當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，由入出控制〝I/O〞之狀態判斷是將擠入資料 IW 擠入貯列（〝I/O〞=1 時）或將貯列中最早擠入的那資料移出送到 OW 去（〝I/O〞=0 時），如上示意圖所示，擠入資料 IW 永遠往貯列之第一個暫存器 QU_1 擠，擠入後 Pr 立刻加 1，使指標能永遠指在貯列中最早擠入之資料。在移出時則直接將 Pr 所指之資料送至 OW，再將 Pr 減 1，使之仍然保持指在剩餘資料中最先擠入的那個資料。

FUN110 DP QUEUE	貯列 (QUEUE)	FUN110 DP QUEUE

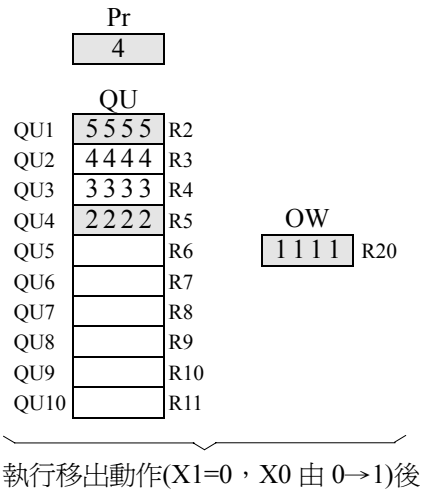
● 在貯列未擠入任何資料或填入者均已被移出時（Pr=0），貯列空白旗號〝EPT〞將變爲 1，此時即使再有移出動作，本指令亦不執行。而若資料僅擠入不移出或擠入多移出少，最終造成貯列已被擠滿（指標 Pr 已指在 QUL 處），則貯列滿溢旗號〝FUL〞變爲 1，此時若再有擠入動作本指令亦不再執行。本指令之指標爲供貯列於存取時永遠保持指在最先擠入之資料，應避免其他程式去更動到它，否則將造成運作錯誤。若有特定之應用需強制設定指標值，則其容許範圍爲 0～L（0 表空白，1～L 則分別對應至 QU1～QUL），超出此範圍，指標錯誤旗號〞ERR〞設爲 1，且本指令不執行。



• 左圖範例程式，假設其起始狀態如上頁之貯列示意圖範例所示，先將之作一次貯列擠入動作，再作一次移出動作，將可得到如下兩個執行結果。無論如何 Pr 總是指在 QUEUE 中最先擠入之資料。

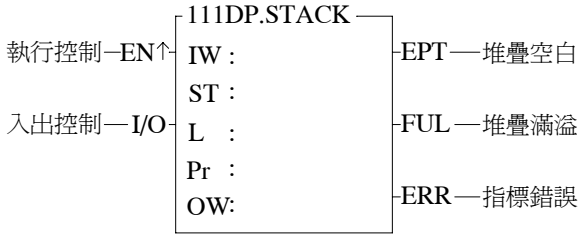


執行擠入動作(X1=1，X0 由 0→1)後



執行移出動作(X1=0，X0 由 0→1)後

FUN111 DP STACK	堆疊 （STACK）	FUN111 DP STACK

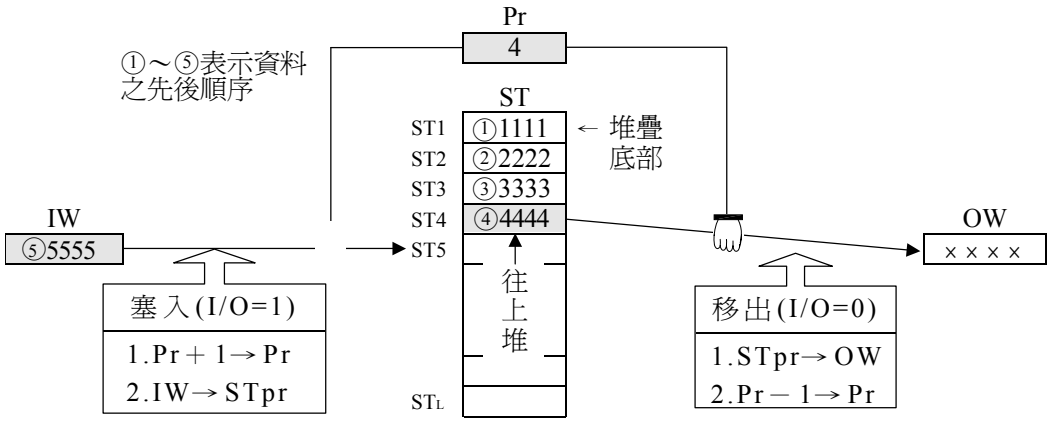


IW ：塞入堆疊之資料或其暫存器號碼
ST ：堆疊之起頭暫存器號碼
L ：堆疊之長度
Pr ：指標暫存器號碼
OW：接收堆疊移出資料之暫存器號碼
ST 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	16 或 32 位元 正、負數	V 、 Z
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
ST		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

- 堆疊和貯列一樣同屬列表之一種，其指標序號性質和貯列完全相同，以 Pr=1～L 來對應 ST_1～ST_L，而 Pr=0 則用以表示該堆疊爲空白。

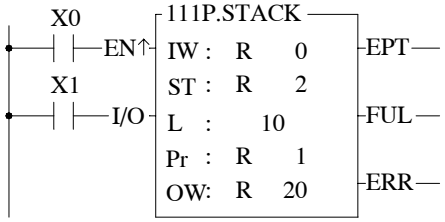
- 堆疊和貯列正好相反，是一種後進先出之裝置，即最後塞入（PUSH）堆疊之資料，在移出（POP）時要最先移出，堆疊是由 ST 開始之連續 L 個 16 位元或 32 位元（D 指令）暫存器所組成，如下之示意圖所示：



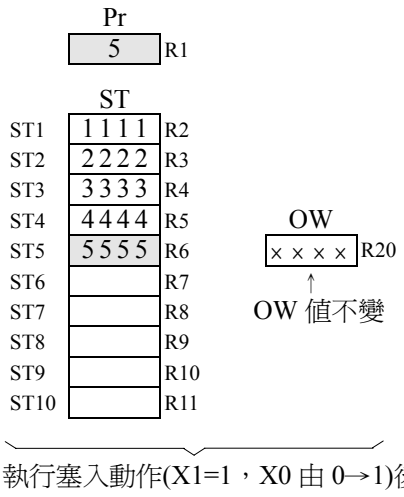
- 堆疊指令之動作係當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，由入出控制〝I/O〞之狀態判斷是將塞入資料 IW 塞入堆疊（〝I/O〞=1 時）或將堆疊中指標 Pr 所指之資料（現存資料中最後塞入者）搬出送到 OW 去（〝I/O〞=0 時），注意塞入之資料是堆疊（STACKING）上去的，故在塞入前要先將 Pr 加 1 使之指到堆疊之最上面（倒著看），再將資料塞入，而在移出時只須將指標 Pr 所指之資料（最後塞入之資料）送到 OW，然後再將 Pr 減 1，使之無論如何動作指標 Pr 均能永遠指在堆疊中最後塞入之那個資料。

FUN111 D P STACK	堆疊 （STACK）	FUN111 D P STACK

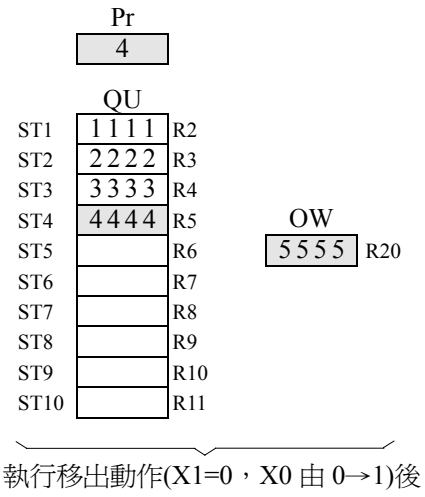
● 在堆疊未塞入任何資料或塞入者均已被移出時（Pr=0），堆疊空白旗號〝EPT〞變爲1，此時若再有移出動作本指令亦不執行，而若資料僅塞入不移出或塞入多移出少，最終造成整個堆疊被塞滿（指標 Pr 已指在 ST 處），則堆疊滿溢旗號〝FUL〞變爲 1，此時若再有塞入動作本指令亦不再執行。同貯列一樣，堆疊之指標 Pr 應避免去更動它，若有特殊應用需強制設定 Pr 值，其有效範圍爲 0～L（0 表空白，1～L 則分別對應至 ST_1～ST_L），超出此範圍，指標錯誤〝ERR〞設爲 1，且本指令不執行。



• 左圖範例程式，假設堆疊狀況正好如上頁之堆疊示意圖所示，先將之作一次塞入動作，再執行一次移出動作，可得到如下兩個結果，無論如何動作 Pr 始終指在堆疊中最後塞入的那個資料。

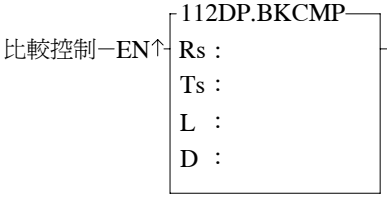


執行塞入動作(X1=1，X0 由 0→1)後



執行移出動作(X1=0，X0 由 0→1)後

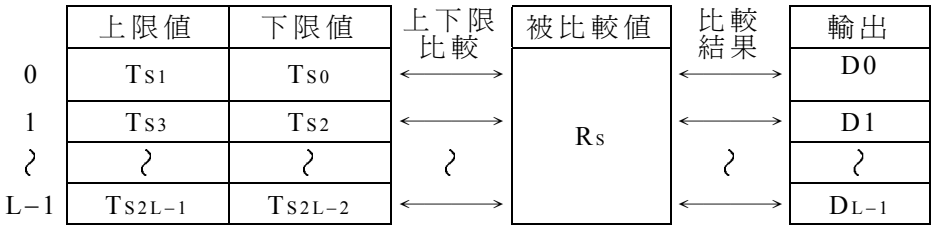
FUN112 **D** **P** BKCMP	區塊比較（凸輪開關 DRUM）（BLOCK COMPARE）	FUN112 **D** **P** BKCMP



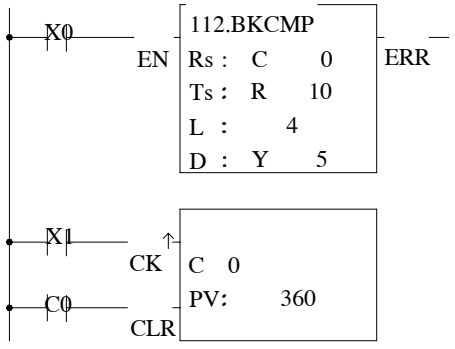
Rs：被比較之資料或其暫存器號碼
Ts：上下限值暫存器區塊之起頭號碼
L ：上下限值之組數
D ：存放比較結果之繼電器起頭號碼

運算元＼範圍	Y	M	S	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Y0 | Y255	M0 | M999	S0 | S999	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	16 或 32 位元正、負數
Rs				○	○	○	○	○	○	○	○	○	○	○	○	○
Ts				○	○	○	○	○	○	○	○	○	○	○	○	
L										○				○*	○	1~256
D	○	○	○													

- 當比較控制〝EN〞=1 或〝EN↑〞（**P**指令）由 0→1 時，將 Rs 之內容值逐一和由暫存器 Ts 開始之 L 組 16 或 32（**D**指令）位元之上下限值（由 T0 開始每相鄰之兩暫存器形成一組上下限設定值）作比較，若 Rs 值落於該組設定值範圍內，則將比較結果之繼電器 D 中對應於該組之位元設爲 1，否則爲 0，直到比完所有 L 組上下限設定值爲止。
- 當 M1975=0 時，若上下限設定值中有任一組之上限值小於下限值，則限值錯誤旗號〝ERR〞設爲 1，且該組比較輸出爲 0。
- 當 M1975=1 時，下限值可大於上限值之設定，而適用於 360°圓周運動，當有跨 0°之電子凸輪角度之應用。



- 本指令實質上爲一機械式之絕對式凸輪開關（DRUM）指令；其亦可置放於定時中斷程式裡，配合 FUN74(IMDIO)可得到較準確之電子凸輪角度輸出。

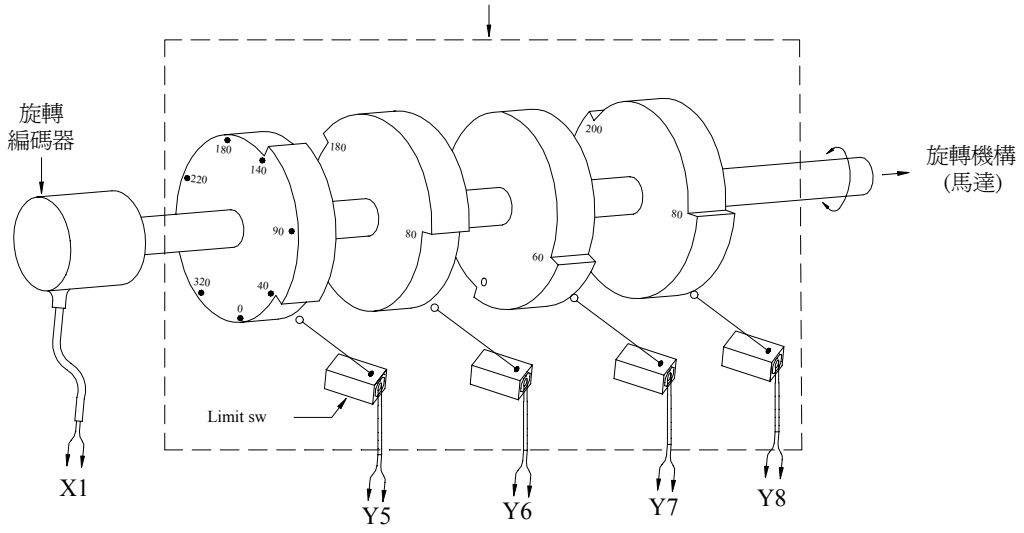


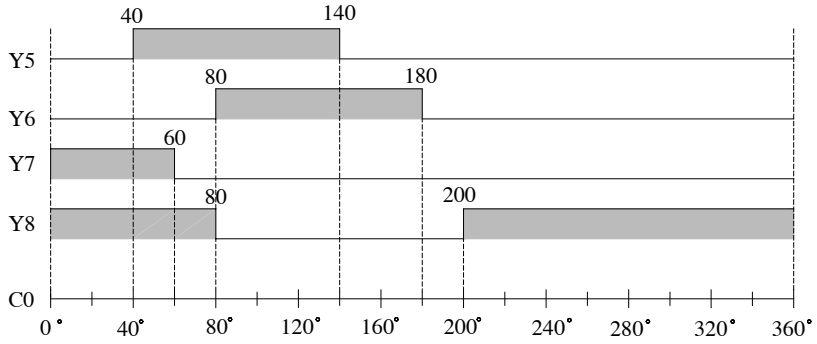
- 本程式範例指定 C0 值作爲凸輪軸之旋轉角度（Rs），並藉由本區塊比較指令，將 Rs 和 R10R11、R12R13、R14R15 和 R16R17 等 4 組（因 L=4）上下限設定值作比較，而得到 Y5～Y8 四個凸輪輸出點之輸出結果。
- 程式中輸入接點 X1 爲一旋轉角度檢知器，係安裝於凸輪軸上，凸輪軸角度每轉動 1 度，X1 就產生一個脈波，凸輪軸轉一周，X1 會產 360 個脈波。

FUN112 DP BKCMP	區塊比較（凸輪開關 DRUM） （BLOCK COMPARE）	FUN112 DP BKCMP

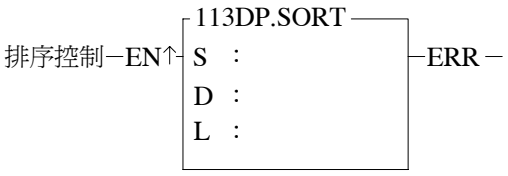
● 上圖程式配合一旋轉編碼器或其他轉動角度檢知裝置（直接連結至旋轉機構）即可組成和實際凸輪機械結構等效之機構裝置（如下圖虛線標示之機構）。同時透過上下限值之修改，您更可隨時改變凸輪觸動之角度範圍，爲傳統凸輪機構所無法達成。

本指令範例所取代之
實際凸輪機械結構





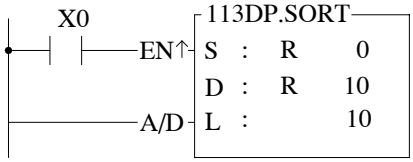
FUN113 DP SORT	大小排序便利指令 （SORTING）	FUN113 DP SORT



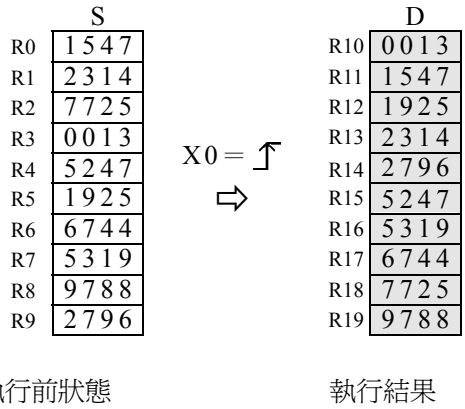
S ：欲排序之來源資料起頭暫存器號碼
D ：排序後之資料起頭暫存器號碼
L ：排序之資料長度

運算元＼範圍	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 127
S	○	○	○	○	○	○	○	○	
D			○				○*	○	
L			○				○	○	○

- 當排序控制〝EN〞=1 或〝EN↑〞（ P 指令）由 0→1 時，將以 S 爲起始之 L 個資料由小而大排序（A/D=1）或由大而小排序（A/D=0），並將排序結果存放到以 D 爲起始之暫存器中。
- 當排序之資料長度錯誤（127 < L 或 L < 2）時，本指令不執行，輸出〝ERR〞=1。



• 左圖程式範例，將 R0 爲起始之暫存器列表由小而大排序，並將排序結果存放到以 R10 爲起始之暫存器列表中，如下圖結果。

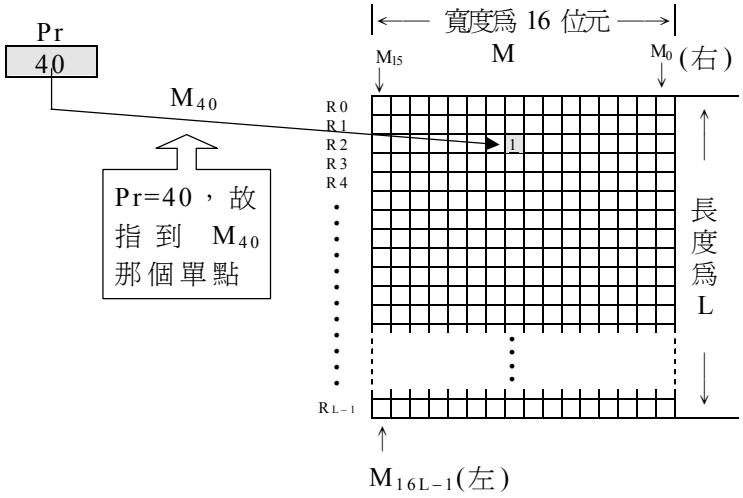


執行前狀態　　執行結果

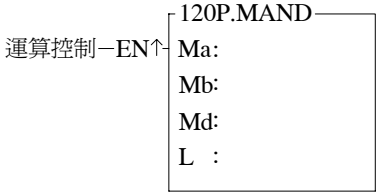
矩　陣　（ＭＡＴＲＩＸ）　指　令

120. MAND
121. MOR
122. MXOR
123. MXNR
124. MINV
125. MCMP
126. MBRD
127. MBWR
128. MBSHF
129. MBROT
130. MBCNT

- 矩陣是 2 個以上連續之 16 位元暫存器所組成，組成矩陣之暫存器個數稱爲矩陣之長度 L，一個矩陣共有 L×16 個位元（點），其運算單位一次只有一個位元（點）。
- 矩陣指令是將 16×L 個矩陣位元（序號由 M_0～M_{16L-1}）當作一連串單點之集合，而自此集合中指定某一單點作運作，而不將之當作數值看待。
- 矩陣指令主要在處理單點對多點（矩陣）或多點對多點之狀態處理，如搬移、拷貝、比較、搜尋等，爲極爲方便和重要之應用指令。
- 在矩陣指令運作中，通常需要有一個 16 位元暫存器來指定矩陣中 16L 個單點之某個單點當作運算對象，此暫存器稱爲矩陣之指標 Pr（Pointer），其有效範圍爲 0～16L−1，分別對應至矩陣中之位元 M_0～M_{16L-1}。
- 矩陣運作中有左、右位移或旋轉，我們定義高序號者爲左，低序號者爲右，如下圖示。



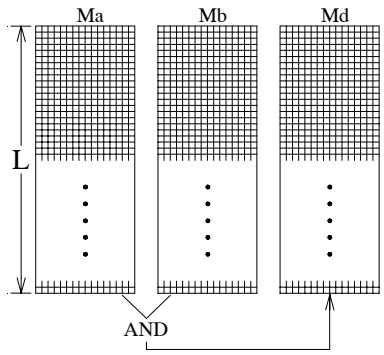
FUN120 P MAND	矩陣邏輯及（AND）運算 （MATRIX AND）	FUN120 P MAND

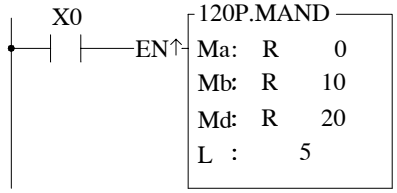


Ma：來源矩陣 a 之起頭暫存器號碼
Mb：來源矩陣 b 之起頭暫存器號碼
Md：存放結果之目的矩陣起頭暫存器號碼
L ：矩陣（Ma、Mb 和 Md）之長度
Ma，Mb，Md 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

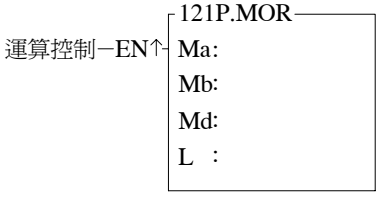
● 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將長度爲 L 之兩來源矩陣 Ma 和 Mb 整個作邏輯 AND 運算（兩位元均爲 1 結果始爲 1，否則爲 0）後，再將結果存到長度同爲 L 之目的矩陣 Md 去（相同序號之位元作 AND，例如 Ma_0=0，Mb_0=1，則 Md_0=0；Ma_1=1，Mb_1=1 則 Md_1=1；……一直 AND 至 Ma_{16L-1} 和 Mb_{16L-1} 止）。





• 左圖程式範例，當 X0 由 0→1 時將 R0～R4 構成之 Ma 和 R10～R14 構成之 Mb 作 AND 後，將結果存至由 R20～R24 所構成之 Md 去，其執行結果如下圖右。

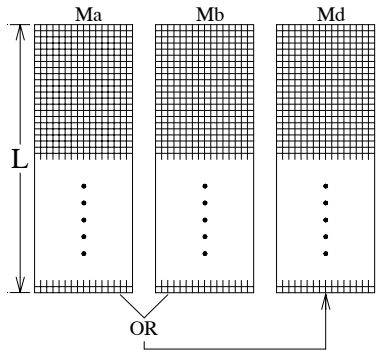
FUN121 P MOR	矩陣邏輯或（OR）運算 （MATRIX OR）	FUN121 P MOR

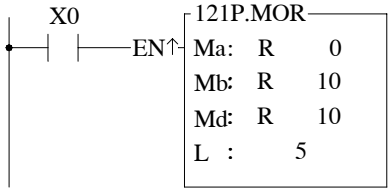


Ma：來源矩陣 a 之起頭暫存器號碼
Mb：來源矩陣 b 之起頭暫存器號碼
Md：存放結果之目的矩陣起頭暫存器號碼
L ：矩陣（Ma、Mb 和 Md）之長度
Ma，Mb，Md 可結合 V、Z 作間接定址應用

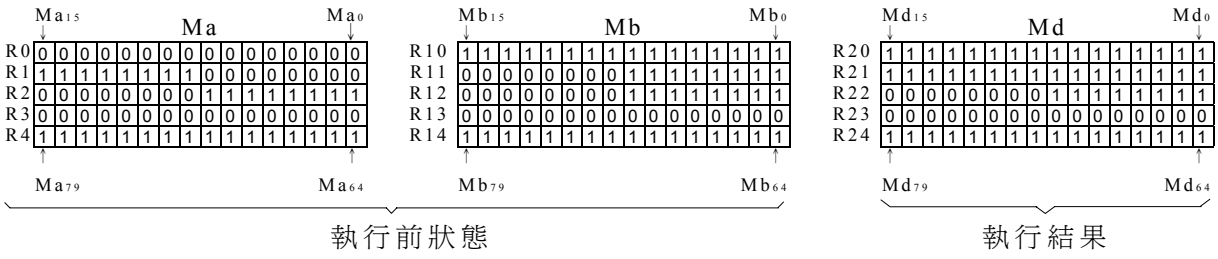
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

● 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將長度爲 L 之兩來源矩陣 Ma 和 Mb 整個作邏輯 OR 運算（兩位元有任一爲 1 則結果爲 1，兩者均爲 0 結果才爲 0）後，再將結果存回長度同爲 L 之目的矩陣 Md 去。（相同序號之位元作 OR，例如 M_0=0，Mb_0=1，則 Md_0=1；Ma_1=0，Mb_1=0 則 Md_1=0；……一直 OR 至 Ma_{16L-1} 和 Mb_{16L-1} 止）。

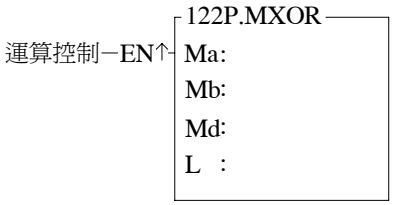




• 左圖程式範例，當 X0 由 0→1 時，將 R0～R4 構成之 Ma 和由 R10～R14 構成之 Mb 作 OR 運算後，將結果存回由 R10～R14 構成之目的矩陣 Md 去，本例因 Mb 和 Md 爲同一個矩陣，故運算後來源矩陣 Mb 已被新值覆蓋，如下右圖之結果。



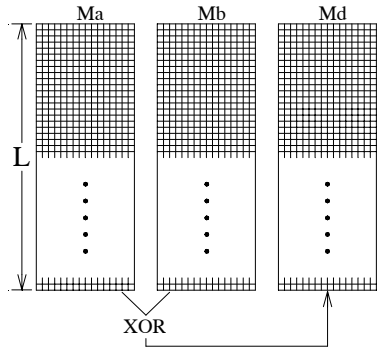
FUN122 P MXOR	矩陣邏輯互斥或（XOR）運算 （MATRIX EXCLUSIVE OR）	FUN122 P MXOR

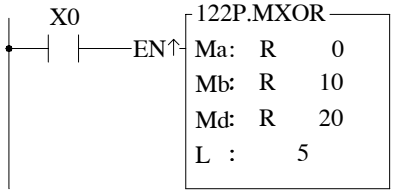


Ma：來源矩陣 a 之起頭暫存器號碼
Mb：來源矩陣 b 之起頭暫存器號碼
Md：存放結果之目的矩陣起頭暫存器號碼
L ：矩陣（Ma，Mb，Md）之長度
Ma，Mb，Md 可結合 V、Z 作間接定址應用

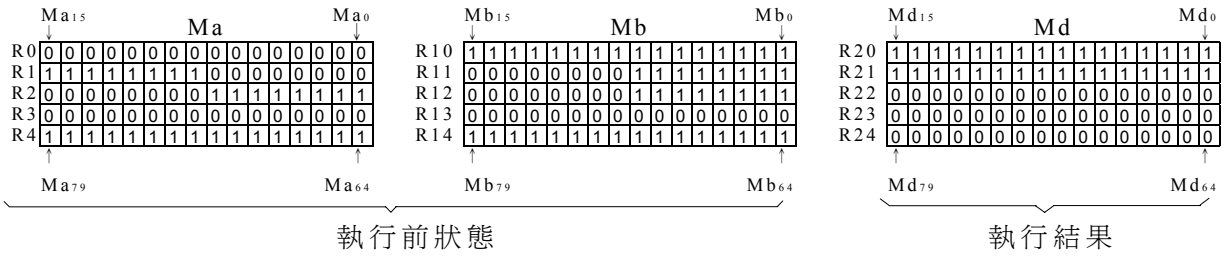
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

● 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將長度爲 L 之兩來源矩陣 Ma 和 Mb 整個作邏輯 XOR 運算（兩位元不同結果爲 1，否則爲 0）後，再將結果存回長度同爲 L 之目的矩陣 Md 去。（相同序號之位元作 XOR，例如 Ma_0=0，Mb_0=1，則 Md_0=1；Ma_1=1，Mb_1=1 則 Md_1=0；……一直 XOR 至 Ma_{16L-1} 和 Mb_{16L-1} 止）。

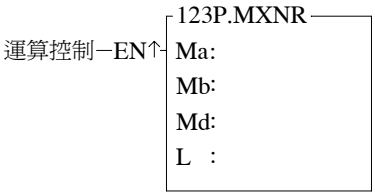




• 左圖程式範例，當 X0 由 0→1 時，將 R0～R4 構成之 Ma 和由 R10～R14 構成之 Mb 作 XOR 運算後，將結果存到由 R20～R24 構成之目的矩陣 Md 去，其運算結果如下圖右。



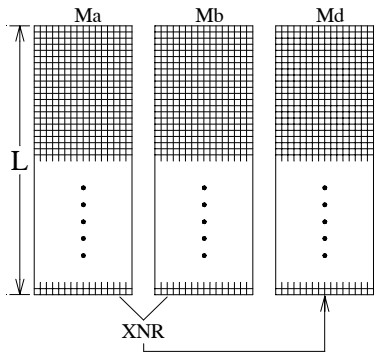
FUN123 P MXNR	矩陣互容或（XNR）運算 （MATRIX ENCLUSIVE OR）	FUN123 P MXNR

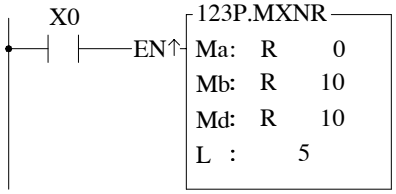


Ma：來源矩陣 a 之起頭暫存器號碼
Mb：來源矩陣 b 之起頭暫存器號碼
Md：存放結果之目的矩陣起頭暫存器號碼
L ：矩陣（Ma，Mb，Md）之長度
Ma，Mb，Md 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

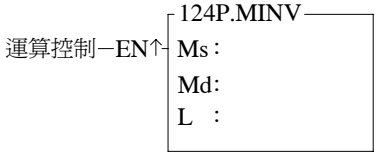
● 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將長度爲 L 之兩來源矩陣 Ma 與 Mb 整個作邏輯 XNR 運算（兩位元相同則結果爲 1，否則爲 0）後，再將結果存到長度同爲 L 之目的矩陣 Md 去（相同序號之位元作 XNR，例如 Ma_0=0，Mb_0=1，則 Md_0=0；Ma_1=0，Mb_1=0 則 Md_1=1；……一直 XNR 至 Ma_{16L-1} 和 Mb_{16L-1} 止）。





• 左圖程式範例，當 X0 由 0→1 時，將 R0～R4 所構成之來源矩陣 Ma 和由 R10～R14 構成之來源矩陣 Mb 作 XNR 運算後，將結果存回由 R10～R14 所構成之 Md 去，本例因 Mb 和 Md 爲同一矩陣，故運算後來源矩陣 Mb 已被新值所覆蓋，如下右圖之結果。

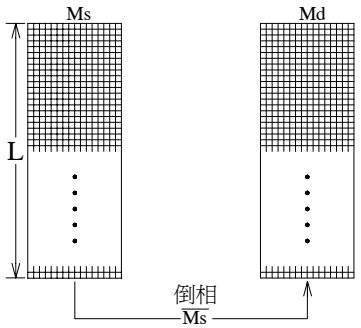
FUN124 P MINV	矩陣倒相 （MATRIX INVERSE）	FUN124 P MINV

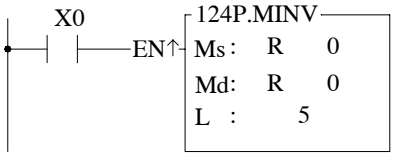


Ms ：來源矩陣之起頭暫存器號碼
Md：存放結果之目的矩陣起頭暫存器號碼
L ：矩陣（Ms 和 Md）之長度
Ms，Md 可結合 V、Z 作間接定址應用

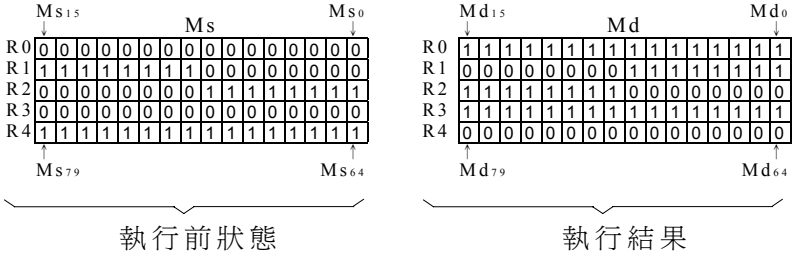
運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

● 當運算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將長度爲 L 之來源矩陣 Ms 整個反相（所有狀態爲 1 之位元變成 0，而狀態爲 0 者則變爲 1）後，再存到目的矩陣 Md 去。

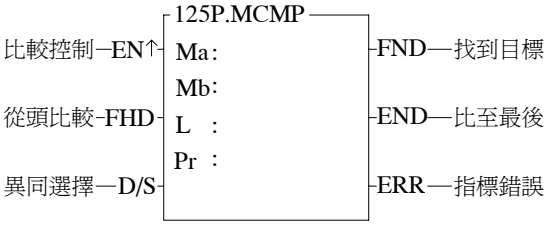




• 左圖程式範例，當 X0 由 0→1 時，將 R0～R4 所組成之矩陣反相後存回自己（因本例 Ms 和 Md 爲同一矩陣）。所獲得之結果如下圖右。



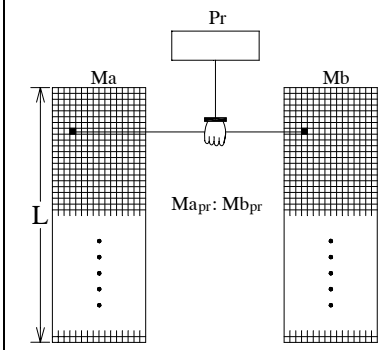
FUN125 P MCMP	矩陣對矩陣比較異同 （MATRIX COMPARE）	FUN125 P MCMP



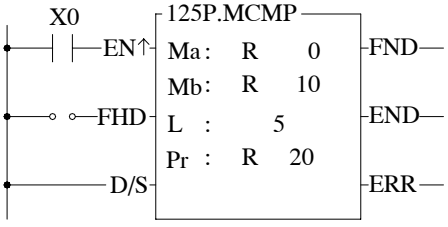
Ma：矩陣 a 之起頭暫存器號碼
Mb：矩陣 b 之起頭暫存器號碼
L ：矩陣（Ma，Mb）之長度
Pr ：指標，用以存放目標之位置值
Ma，Mb 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

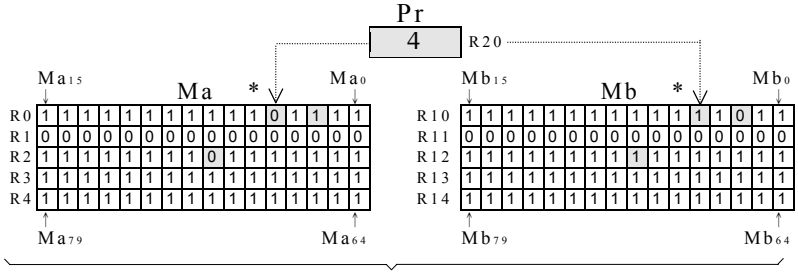
● 當比較控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，自 Ma 和 Mb 兩矩陣中之最開頭那對位元（Ma_0 和 Mb_0）開始（〝FHD〞=1 或 Pr 值已等於 16L-1 時）或自當時指標 Pr 所指那對位元之下一對位元（Ma_{pr+1} 和 Mb_{pr+1}）開始（〝FHD〞=0 同時 Pr 值小於 16L−1）往下雙雙成對比較找尋狀態不同（D/S=1 時）或相同（D/S=0 時）之位元對（Pair），當找到目標（狀態不同或相同之位元對）後立即停止比較動作，同時將該對目標在矩陣中之位置序號值存到指標 Pr 去，並將找到目標旗號〝FND〞設爲 1 後結束本指令之執行。當找到矩陣之最後一對位元（Ma_{16L-1}，Mb_{16L-1}）時，無論其是否爲要找尋的目標均將結束該次之比較找尋動作，並將比至最後旗號〝END〞設爲 1，而 Pr 值則停在 16L−1。當下次本指令再度被執行時，Pr 將會自動循環至矩陣之最開頭（Pr=0）處開始往下比較。

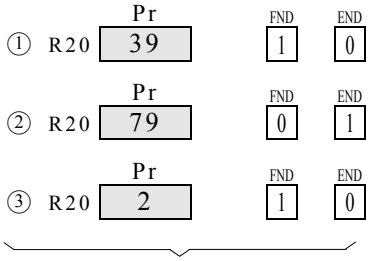


● 指標值之範圍爲 0～16L-1，在運作中應避免更動到 Pr 值，以免影響其正確之比較找尋，若 Pr 值超出此範圍則指標錯誤旗號〝ERR〞設爲 1，且本指令不執行。

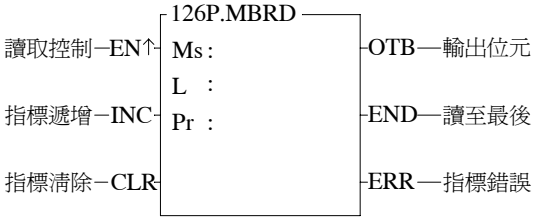


- 左圖程式範例，因〝FHD〞輸入爲 0，故由指標當時值加 1 處（標註*處）開始往下比較找尋位元狀態不同（因 D/S=1 爲找不同）者。當 X0 由 0→1 動作 3 次，可得到如下圖右①，②，③三個執行結果。





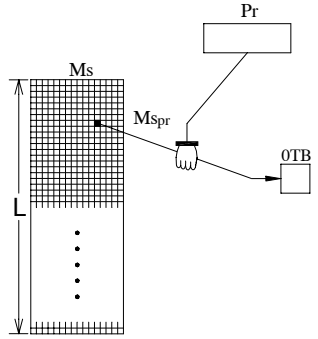
FUN126 P MBRD	矩陣位元讀取 (MATRIX BIT READ)	FUN126 P MBRD



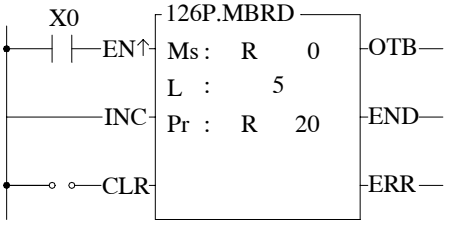
Ms：矩陣之起頭暫存器號碼
L ：矩陣之長度
Pr ：指標暫存器號碼
Ms 可結合 V、Z 作間接定址應用

運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C199	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

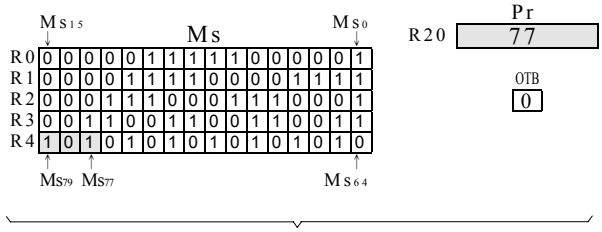
● 當讀取控制〝EN〞=1 或〝EN↑〞(P指令）由 0→1 時，讀取矩陣 Ms 中指標 Pr 所指之那個位元 Ms_{pr} 之狀態並將之送到輸出位元〝OTB〞去。在讀取前會先去檢視指標清除〝CLR〞之狀態，若〝CLR〞爲 1，則會先將 Pr 清爲 0 後再作讀取動作。在讀取完畢後接著檢視 Pr 值，若 Pr 值已達 16L–1（指到最後一個位元），則將讀至最後旗號〞END〞設爲 1，然後結束本指令之執行，若 Pr 小於 16L–1，則再查看指標遞增〝INC〞之狀態，若〝INC〞爲 1 則將指標 Pr 值加 1 後才結束本指令之執行。此外，指標清除〝CLR〞可單獨執行，不受其他輸入影響。



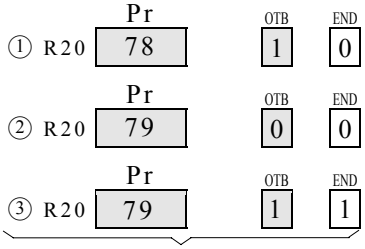
● 指標之有效範圍爲 0～16L–1，超出此範圍則將指標錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例，因 INC=1，故每讀取一次，指標即加 1，如此可連續讀取 Ms 中之每一位元，如下圖左之狀態，當 X0 由 0→1 動作 3 次後，可得到下圖右①，②，③三個執行結果。

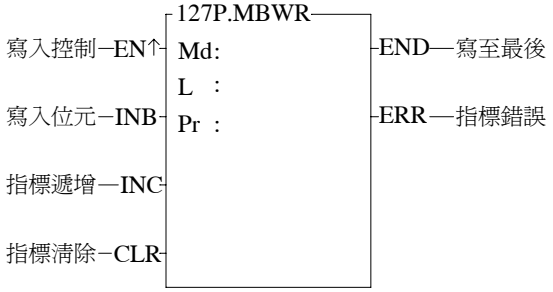


執行前狀態



執行結果

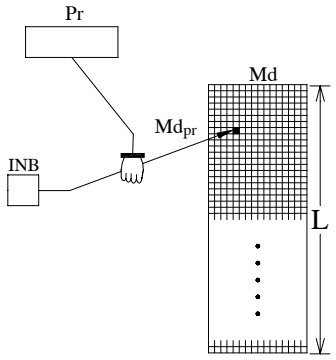
FUN127 P MBWR	矩陣位元寫入 （MATRIX BIT WRITE）	FUN127 P MBWR



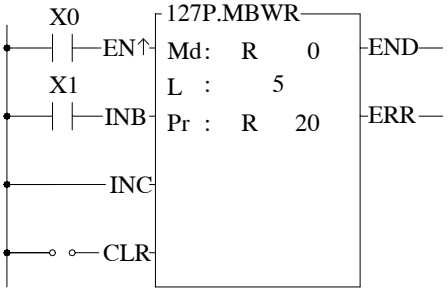
Md：矩陣之起頭暫存器號碼
L ：矩陣之長度
Pr ：指標暫存器之號碼
Md 可結合 V、Z 作間接定址應用

運算元＼範圍	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D3071	2 ｜ 256	V 、 Z
Md	○	○	○	○	○	○	○	○	○	○		○
L						○			○*	○	○	
Pr	○	○	○	○	○	○	○	○*	○*	○		

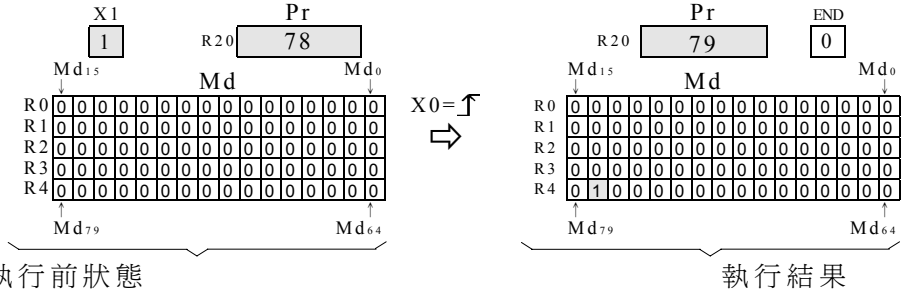
● 當寫入控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將寫入位元〝INB〞之狀態寫到矩陣 Md 中指標 Pr 所指的那個位元 Md_{pr} 去。在寫入之前會先去檢視指標清除〝CLR〞之狀態，若〝CLR〞爲 1，則會先將 Pr 清爲 0 後再作寫入動作。在執行完寫入動作後接著檢視指標 Pr 之值，若 Pr 值已達 16L–1（指到最後位元），則將寫至最後旗號〝END〞設爲 1 後結束該次執行。若 Pr 值小於 16L–1，則再檢視指標遞增輸入〝INC〞之狀態，若〝INC〞爲 1 則將 Pr 值加 1 後才結束執行。此外，指標清除〝CLR〞能單獨執行，不受其他輸入影響。



● Pr 之有效範圍爲 0～16L–1，超出此範圍則指標錯誤旗號〝ERR〞設爲 1，且本指令不執行。



• 左圖程式範例，指標每次執行後均會遞增（因〝INC〞爲 1）。如下圖示，當 X0 由 0→1 動作一次後，INB 之狀態（X1）即被寫到 Md_{pr}（即 Md_{78}）處，且指標 Pr 加 1（變爲 79）。此時雖 Pr 已指到最後但尚未寫入 Md_{79}，故〝END〞仍爲 0，須等到下次動作真正寫入 Md_{79} 後，〝END〞才會變爲 1。



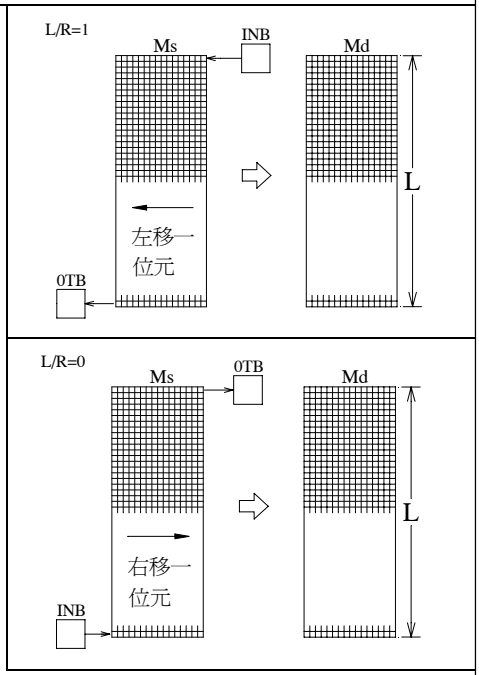
執行前狀態　　執行結果

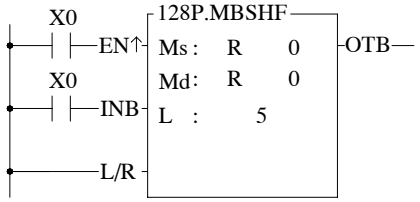
FUN128 P MBSHF	矩陣位元位移 （MATRIX BIT SHIFT）	FUN128 P MBSHF

Ms：來源矩陣之起頭暫存器號碼
Md：目的矩陣之起頭暫存器號碼
L ：矩陣（Ms 和 Md）長度
Ms，Md 可結合 V、Z 作間接定址應用

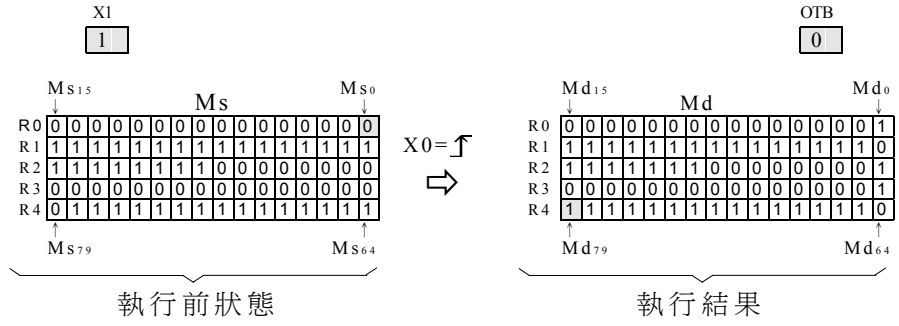
運算元 \ 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

● 當位移控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將矩陣 Ms 整個取出向左（L/R=1 時）或向右（L/R=0 時）位移一個位置，因位移而騰出之空位（左移時爲 M_0，右移時爲 M_{16L-1}）則以填補位元〝INB〞之狀態填補。而因位移而擠出之位元（左移時爲 M_{16L-1}，右移時爲 M_0）狀態則送到移出位元〝OTB〞去，然後再將此位移過之矩陣結果填入目的矩陣 Md 去。

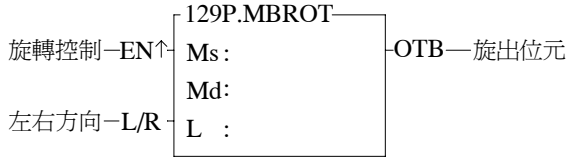




• 左圖程式範例之 Ms 和 Md 爲同一矩陣之範例，當 X0 由 0→1 動作時，將 Ms 整個取出作左移（因 L/R=1）一位元後，再存回 Md 而得到如下圖右之結果。



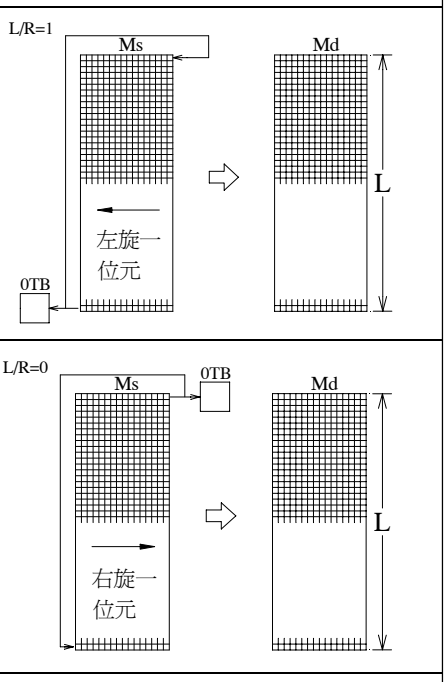
FUN129 P MBROT	矩陣位元旋轉 （MATRIX BIT ROTATE）	FUN129 P MBROT

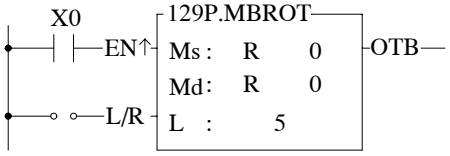


Ms：來源矩陣之起頭暫存器號碼
Md：目的矩陣之起頭暫存器號碼
L ：矩陣（Ms 和 Md）長度
MS，Md 可結合 V、Z 作間接定址應用

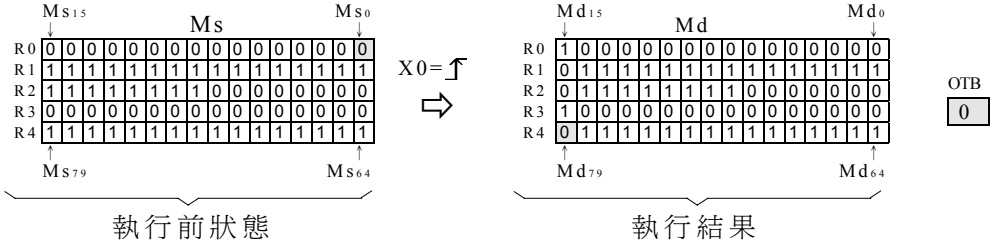
運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

● 當旋轉控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，將矩陣 Ms 整個取出向左（L/R=1 時）或向右（L/R=0 時）旋轉一位元，因旋轉造成之空位（左旋時爲 M_0，右旋時爲 M_{16L-1}）由旋出位元（左旋時爲 M_{16L-1} 右旋時爲 M_0）狀態填補之。再將此經旋轉後之結果填入 Md 去。旋出位元不但用以填補前述之空位，同時並將之送到旋出位元〝OTB〞去。

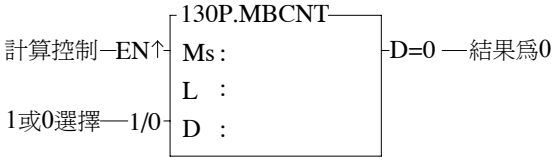




• 左圖程式範例之 Ms 和 Md 爲同一矩陣，當 X0 由 0→1 動作時，將整個 Ms 取出向右旋轉（因 L/R=0）一位元後再存回 Ms 自己（因本例 Ms 和 Md 爲同一矩陣），而得到如下圖右之結果。



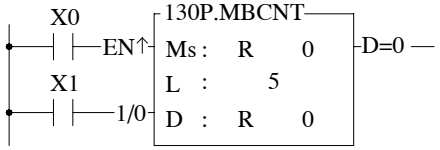
FUN130 P MBCNT	矩陣位元狀態數量計算 （MATRIX BIT STATUS COUNT）	FUN130 P MBCNT



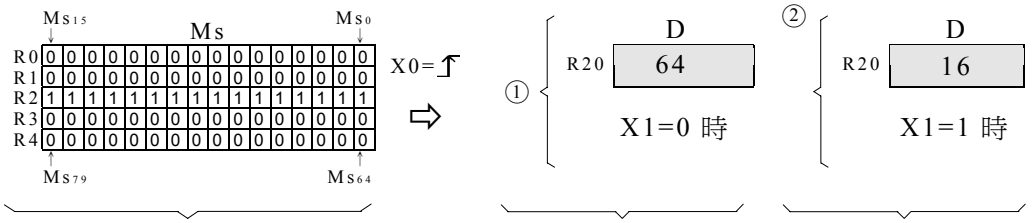
Ms：矩陣之起頭暫存器號碼
L ：矩陣之長度
D ：存放數量結果之暫存器號碼
Ms 可結合 V、Z 作間接定址應用

運算元＼範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D3071	2 | 256	V、Z
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
D		○	○	○	○	○	○		○	○*	○*	○		

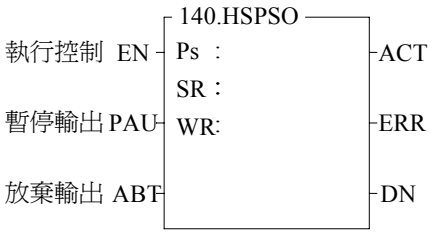
● 當計算控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，統計矩陣 Ms 之 16L 個位元中，所有狀態爲〝1〞（ON）之位元總數目（1 或 0 選擇輸入〝1/0〞=1 時）或所有狀態爲〝0〞（OFF）之位元總數目（1 或 0 選擇輸入〝1/0〞=0 時）。再將統計所得之數目結果存到 D 所指定之暫存器去，若此數目值爲 0，則將結果爲 0 旗號〝D=0〞設爲 1。



• 左圖程式範例分別將 X1 設爲 0（統計狀態爲 0 之位元數）及設爲 1（統計狀態爲 1 者）兩種條件，再分別於此兩條件下使 X0 由 0→1 各一次，可獲得如下圖右①，②兩個執行結果。



FUN140 HSPSO	高速脈波輸出（HSPSO）指令 （功能簡述）	FUN140 HSPSO



Ps ：第幾組 Pulse Output （0～3）
0：Y0 & Y1
1：Y2 & Y3
2：Y4 & Y5
3：Y6 & Y7

SR ：定位程式起始暫存器

WR：指令運作起始暫存器，共佔用 7 個暫存器，其它程式不可重覆使用

運算元 \ 範圍	HR	DR	ROR	K
	R0 ～ R3839	D0 ～ D3071	R5000 ～ R8071	2 ～ 256
Ps				0～3
SR	○	○	○	
WR	○	○	○*	

指令功能簡述

- HSPSO（FUN140）指令之 NC 定位程式是以文字之程式書寫方式來編輯；每一定位點我們稱一步（含輸出頻率、動作行程、轉移條件），一個 FUN140 最多可編 250 步定位點，每一步定位點需佔用 9 個暫存器。 (詳細之應用請參考第 14 章〝FB-PLC 之 NC 定位控制)。
- 將定位程式存在暫存器最大好處是，如果結合人機作機台操控設定，則可將定位程式存入人機，更換模具時，可直接由人機操作存取該副模具之定位程式。
- 本指令之 NC 定位無直線補間功能。
- 當執行控制輸入〝EN〞=1 時，如 Ps0～3 沒有被其它 FUN140 指令佔用（Ps0=M1992，Ps1=M1993，Ps2=M1994，Ps3=M1995 之狀態爲 ON），則由下一步定位點開始執行（如已至最後一步，則重新由第 1 步開始執行）；如 Ps0～3 被其它 FUN140 指令佔用（Ps0=M1992，Ps1=M1993，Ps2=M1994，Ps3=M1995 之狀態爲 OFF），則等佔用之 FUN140 釋出控制權，本指令取得定位控制之脈波（Pulse）輸出權。
- 當執行控制輸入〝EN〞= 0 時，馬上停止脈波輸出。
- 當暫停輸出〝PAU〞= 1 ，且執行控制〝EN〞原先爲 1 時，則暫停脈波輸出；當暫停輸出〝PAU〞= 0，而執行控制〝EN〞仍爲 1 時，繼續輸出未完成之脈波數。
- 當放棄輸出〝ABT〞= 1 時，馬上停止脈波輸出。（下一次當執行控制輸入〝EN〞= 1 時，重新由第一步定位點開始執行）。
- 當脈波輸出中，輸出指示〝ACT〞ON。.
- 當指令執行錯誤時，輸出指示〝ERR〞ON 。（錯誤代碼存放在錯誤碼暫存器）
- 當每一步定位點完成時，輸出指示〝DN〞ON。

*** 務必設定 Pulse Output 之工作模式（不設定時，Y0～Y7 當作一般輸出）爲 U/D，K/R 或 A/B 等三種模式之一，Pulse Output 才能正常輸出。

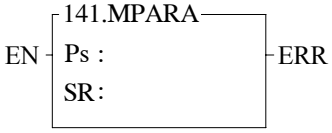
U/D 模式：Y0（Y2，Y4，Y6）送出上數脈波
Y1（Y3，Y5，Y7）送出下數脈波

K/R 模式：Y0（Y2，Y4，Y6）送出脈波
Y1（Y3，Y5，Y7）送出方向信號；ON=上數，OFF=下數

A/B 模式：Y0（Y2，Y4，Y6）送出 A 向脈波
Y1（Y3，Y5，Y7）送出 B 向脈波

- Pulse Output 輸出極性可選擇 Normal ON 或 Normal OFF
- 在 PROLADDER〝HSC〞設定頁可設定 Pulse Output 之工作模式。

FUN141 MPARA	NC 定位參數值設定指令 （功能簡述）	FUN141 MPARA



Ps：第幾組 Pulse Output （0～3）

SR：參數表起始暫存器，共 18 個參數，
佔用 24 個暫存器

運算元＼範圍	HR	DR	ROR	K
	R0 ｜ R3839	D0 ｜ D3071	R5000 ｜ R8071	2 ｜ 256
Ps				0 ～3
SR	○	○	○	

指令功能簡述

- 本指令並不一定要使用；如果系統內定之參數值已符合使用者需求，則可不必有此指令；如果需開放參數值作動態修改，則需有此指令。
- 本指令配合 FUN140 作定位控制使用。
- 不管執行控制輸入〝EN〞=0 或 1 時，本指令皆會被執行。
- 當參數值有錯誤時，輸出指示〝ERR〞ON。（錯誤代碼存放在錯誤碼暫存器）
- 詳細功能敘述與使用方法請參考第 14 章〝FB-PLC 之 NC 定位控制〞之說明。

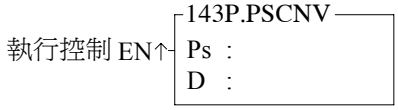
FUN142 P PSOFF	強制停止 HSPSO 脈波輸出指令 （功能簡述）	FUN142 P PSOFF

Ps：0～3
強制第幾組 Pulse Output 停止輸出

指令功能簡述

- 當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，本指令將強制所指定之第幾組 HSPSO（High Speed Pulse Output）停止脈波輸出。
- 在執行機械原點復歸之應用時，當原點條件滿足時，利用本指令可快速停止脈波輸出，讓每次作機械原點復歸時，都停在同一個位置。
- 詳細功能敘述與使用方法請參考第 14 章〝FB-PLC 之 NC 定位控制〞之說明。

FUN143 P PSCNV	目前脈波值轉換爲顯示值（mm，Deg，lnch，PS）指令 （功能簡述）	FUN143 P PSCNV



Ps ：0～3；將第幾組脈波位置（PS）轉換爲與設定值同單位之 mm（Deg，Inch，PS），以作爲目前位置顯示.

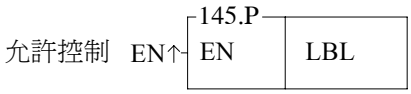
D ：儲存轉換後目前位置之暫存器,共需使用二個暫存器；例如 D10，即代表 D10（Low Word）與 D11（High Word）二個暫存器。

運算元 \ 範圍	HR	DR	ROR	K
	R0 ｜ R3839	D0 ｜ D3071	R5000 ｜ R8071	2 ｜ 256
Ps				0 ～3
D	○	○	○	

指令功能簡述

- 當執行控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，本指令將所指定之目前脈波位置（PS）轉換爲與設定值同單位之 mm（或 Deg 或 Inch 或 PS），以作爲目前位置顯示。
- FUN140 指令執行後，本指令執行時，才會得到正確之轉換值。
- 詳細功能敘述與使用方法請參考第 14 章〝FB-PLC 之 NC 定位控制〞之說明。

FUN145 P EN	允許外界輸入或週邊中斷作動指令	FUN145 P EN



LBL：允許中斷作動之外界輸入或週邊標記名稱。

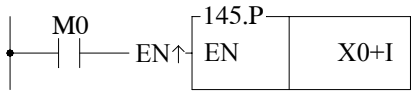
- 當允許控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，允許 LBL 所指定之外界輸入或週邊中斷作動。

- 可允許之中斷標記名稱如下：（詳細請參考 10.3 節之說明）

LBL 名稱	敘　述	LBL 名稱	中斷控制點	LBL 名稱	中斷控制點
HSTAI	HSTA 高速計時器中斷	X4+I	X4 正緣中斷	X10+I	X10 正緣中斷
HSC0I	HSC0 高速計數器中斷	X4–I	X5 負緣中斷	X10–I	X10 負緣中斷
HSC1I	HSC1 高速計數器中斷	X5+I	X5 正緣中斷	X11+I	X11 正緣中斷
HSC2I	HSC2 高速計數器中斷	X5–I	X5 負緣中斷	X11–I	X11 負緣中斷
HSC3I	HSC3 高速計數器中斷	X6+I	X6 正緣中斷	X12+I	X12 正緣中斷
X0+I	X0 正緣中斷	X6–I	X6 負緣中斷	X12–I	X12 負緣中斷
X0–I	X0 負緣中斷	X7+I	X7 正緣中斷	X13+I	X13 正緣中斷
X1+I	X1 正緣中斷	X7–I	X7 負緣中斷	X13–I	X13 負緣中斷
X1–I	X1 負緣中斷	X8+I	X8 正緣中斷	X14+I	X14 正緣中斷
X2+I	X2 正緣中斷	X8–I	X8 負緣中斷	X14–I	X14 負緣中斷
X2–I	X2 負緣中斷	X9+I	X9 正緣中斷	X15+I	X15 正緣中斷
X3+I	X3 正緣中斷	X9–I	X9 負緣中斷	X15–I	X15 負緣中斷
X3–I	X3 負緣中斷				

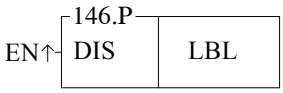
- 在實際應用上，有些中斷信號在有些時候不可讓它發生作用，而在有些時候卻必須讓它發生作用；利用 FUN146（DIS）和 FUN 145（EN）指令即可達成上述需求。

程式範例



- 當 M0 由 0→1 時，允許 X0 由 0→1 時發出中斷；CPU 可立即快速處理 X0+I 之中斷服務程式。

FUN146 P DIS	禁止外界輸入或週邊中斷作動指令	FUN146 P DIS



LBL：禁止作動之外界輸入或週邊之中斷標記名稱。

- 當禁止控制〝EN〞=1 或〝EN↑〞（P指令）由 0→1 時，禁止 LBL 所指定之外界輸入或週邊中斷或週邊功能作動。

- 可禁止之中斷標記名稱如下：(同可允許者)

LBL 名稱	敘　述	LBL 名稱	中斷控制點	LBL 名稱	中斷控制點
HSTAI	HSTA 高速計時器中斷	X4+I	X4 正緣中斷	X10+I	X10 正緣中斷
HSC0I	HSC0 高速計數器中斷	X4–I	X5 負緣中斷	X10–I	X10 負緣中斷
HSC1I	HSC1 高速計數器中斷	X5+I	X5 正緣中斷	X11+I	X11 正緣中斷
HSC2I	HSC2 高速計數器中斷	X5–I	X5 負緣中斷	X11–I	X11 負緣中斷
HSC3I	HSC3 高速計數器中斷	X6+I	X6 正緣中斷	X12+I	X12 正緣中斷
X0+I	X0 正緣中斷	X6–I	X6 負緣中斷	X12–I	X12 負緣中斷
X0–I	X0 負緣中斷	X7+I	X7 正緣中斷	X13+I	X13 正緣中斷
X1+I	X1 正緣中斷	X7–I	X7 負緣中斷	X13–I	X13 負緣中斷
X1–I	X1 負緣中斷	X8+I	X8 正緣中斷	X14+I	X14 正緣中斷
X2+I	X2 正緣中斷	X8–I	X8 負緣中斷	X14–I	X14 負緣中斷
X2–I	X2 負緣中斷	X9+I	X9 正緣中斷	X15+I	X15 正緣中斷
X3+I	X3 正緣中斷	X9–I	X9 負緣中斷	X15–I	X15 負緣中斷
X3–I	X3 負緣中斷				

- 在實際應用上，有些中斷信號在有些時候不可讓它發生作用，即可利用此指令將該中斷信號禁止而不會產生中斷處理。

程式範例

• 當 M0 由 0→1 時，禁止 X2 由 0→1 時發出中斷處理。