

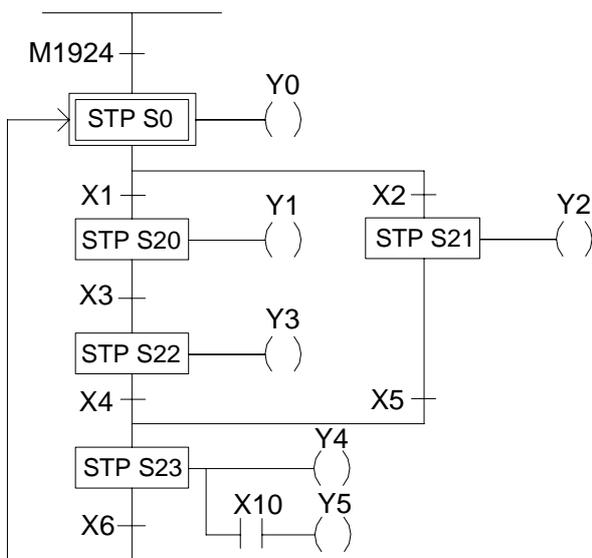
第 8 章：步進指令說明

結構化程式設計是軟體設計的一大趨勢，它的特點是可讀性高、維護、更新容易，軟體品質可靠性大大提昇。尤其當控制偏向機械流程控制時，使用傳統的設計方式來設計程式，往往令第三者難以接手，也就是程式可讀性低、更新、維護風險較高。因此，專門針對機械動作流程之順序控制，如能結合現有廣泛之階梯圖語言，再加上步進執行指令輔助，將使這方面的設計工作更省時，更省力，且軟體掌握度更高。我們將這種結合流程控制與階梯圖語言的設計方式稱為步進階梯（STEP LADDER）語言。

步進階梯圖是以一個步進點（STEP）為最小單元。一個步進點相當於機械運件中的一個步序（站），每個步序皆有動作輸出，整台機械或是整個順序控制的流程，便是一個一個步進點邏輯串聯或並聯組成，其一步接一步循序執行之環境，使人對機械之運作一目了然，在設計、操作、維護上皆相當便捷容易。

8.1 步進階梯圖工作原理

【範例】

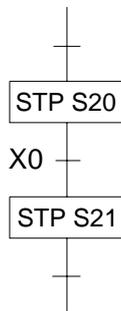


【說明】

1. **STP Sxxx** 是步進點（STEP）Sxxx 的表示符號，其中 Sxxx 可從 S0 ~ S999。當執行到此步進點時（此點 ON），便會執行連在右邊的階梯圖，而前一個步進點及輸出皆會變 OFF。
2. M1924 為一開機 ON 一個掃描時間的接點，因此，一開機即進入初始步進點 S0（S0 ON）這一站，而其他步進點皆不動作，Y1~Y5 皆 OFF。即 M1924 ON → S0 ON → Y0 ON，Y0 會維持到 X1 或 X2 其中一接點先 ON 為止。
3. 假設 X2 先 ON，就會執行 S21 這條路徑，即 X2 ON ⇒ { S21 ON ⇒ { Y2 ON
S0 OFF ⇒ { Y0 OFF
，而 Y2 會維持到 X5 ON 為止。
4. 假設 X5 ON，就會前進至步進點 S23 這一站，即 X5 ON ⇒ { S23 ON ⇒ { Y4 ON
S21 OFF ⇒ { Y2 OFF
，Y4 和 Y5 會維持到 X6 ON 為止。
※如 X10 ON，則 Y5 也會 ON。
5. 假設 X6 ON，就會前進至步進點 S0 這一站，即 X6 ON ⇒ { S0 ON ⇒ { Y0 ON
S23 OFF ⇒ { Y4、Y5 OFF
，如此便完成一個循環的控制流程，而進入下一循環的控制流程。

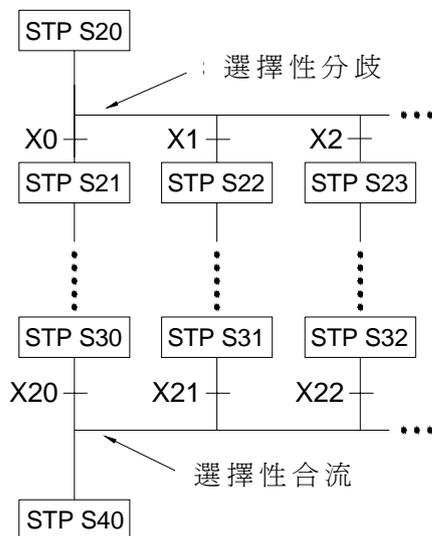
8.2 步進階梯圖基本組成

① 單一回路



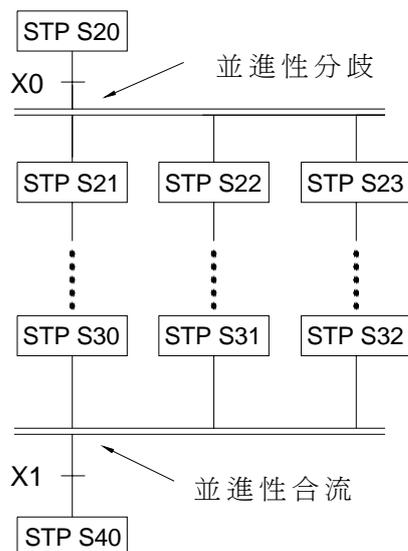
- 步進點 S20 單獨經 X0 前進至步進點 S21。
- X0 可改為其他接點的串、並聯組合。

② 選擇性分歧/合流



- 步進點 S20 選擇其下先 ON 之接點，當作唯一執行的回路，例如 X2 先 ON，則只執行步進點 S23 這條回路。
- 一個分歧最多有 8 條回路。
- X1、X2……X22 皆可改為其他接點的串、並聯組合。

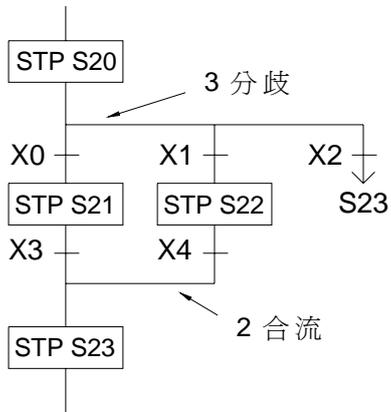
③ 並進式分歧/合流



- 步進點 S20 在 X0 ON 後，同時執行其下的所有回路，即 S21、S22、S23……皆動作。
- 在合流處上之所有分歧回路皆執行至最後一個步進點（如 S30、S31、S32），在等到 X1 ON，即可轉至步進點 S40 執行。
- 並進式分歧的分歧回路數和合流回路數需一致，且最多是 8 條回路。

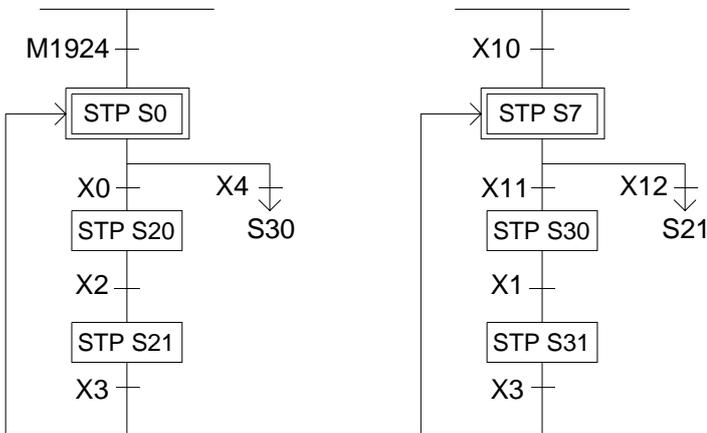
④ 跳躍

a. 同一步進流程



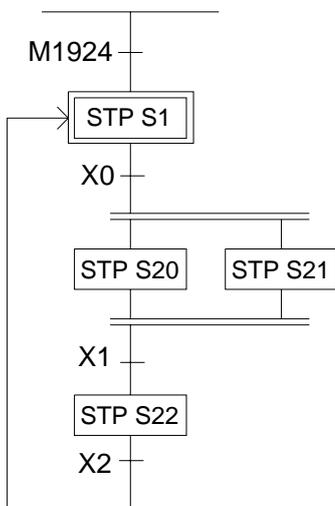
- 如左邊步進點 S20 下有三條路徑，假設 X2 ON，則直接跳躍至步進點 S23 執行，不需經選擇性合流的過程。
- 但並進式分歧的路徑不能跳躍執行。

b. 不同步進流程

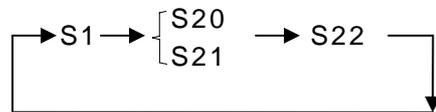


⑤ 閉回路型和單循環型

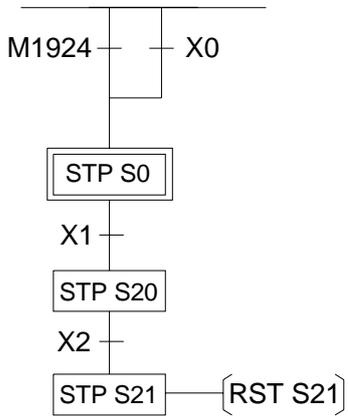
a. 閉回路型



- 一開機初始步進點 S1 ON，往後只會如下做無限次循環。

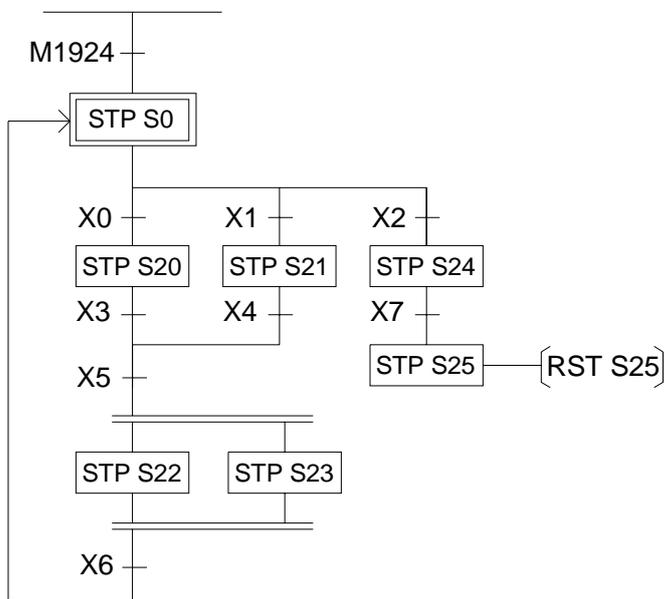


b.單循環型

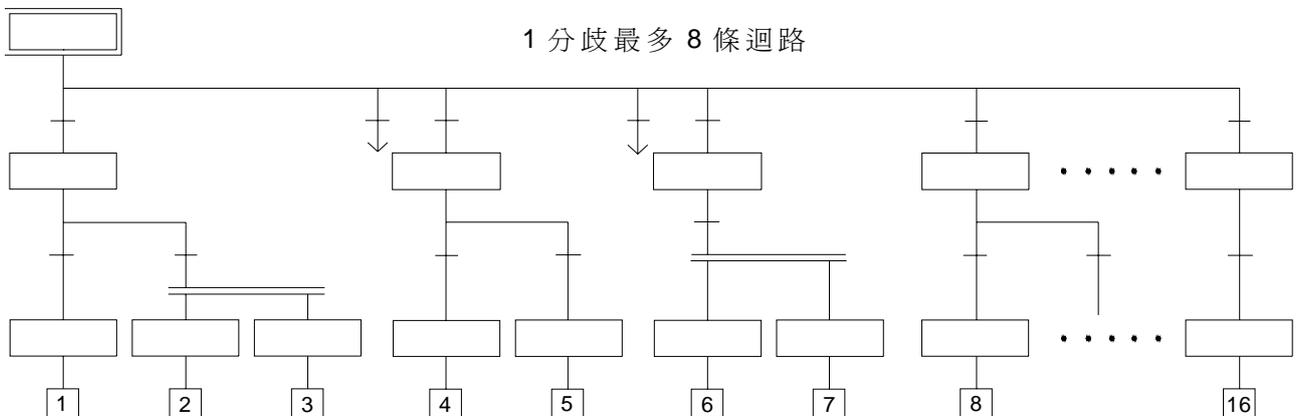


- 在步進點 S20 ON 時，如 X2 ON，S21 本應 ON，但被“RST S21”給 OFF，而結束此步進流程。

c.混合型流程



⑥ 綜合應用



1 分歧最多 8 條迴路

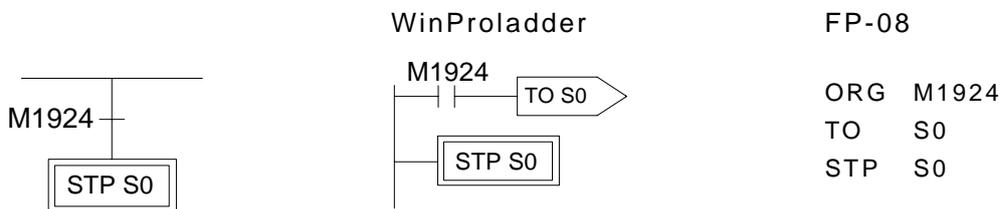
一個初始步進點下橫向的最大分歧回路數為 16

8.3 步進指令介紹：STP、FROM、TO、STPEND

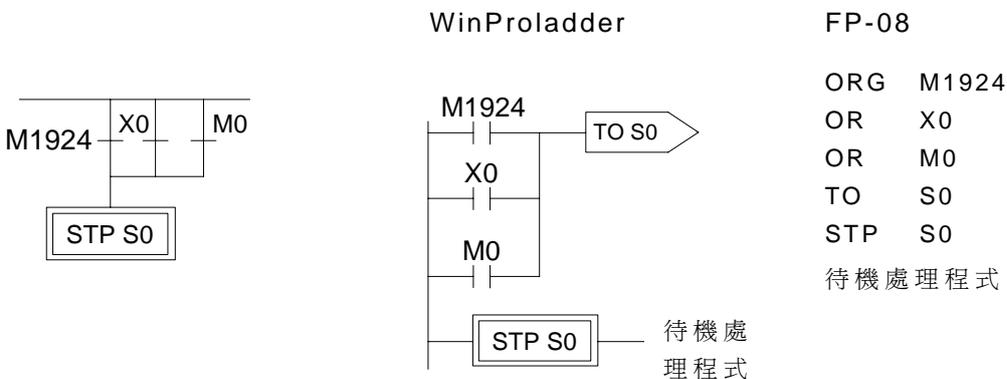
- **STP Sx** : $S0 \leq Sx \leq S7$ (WinProladder 輸入/顯示格式)
 或
STP Sx : $S0 \leq Sx \leq S7$ (FP-08 輸入按 "STP" Key)

此指令為初始步進點 (Initial Step) 指令，由此指令才可往下衍生出各個機械流程之步進控制。FBs 系列最多可提供 8 個初始步進點，也就是說一台 PLC 最多可同時作 8 個流程控制。每一步進流程可獨立運作或是產生運作結果供其他流程參考使用。

【範例一】每次開機啓動初始步進點 S0



【範例二】每次開機或按手動鈕或自動生產異常發生而在某特定時間內無人員處置自動進入初始步進點 S0 待機



【說明】X0：手動鈕；M0：異常之接點

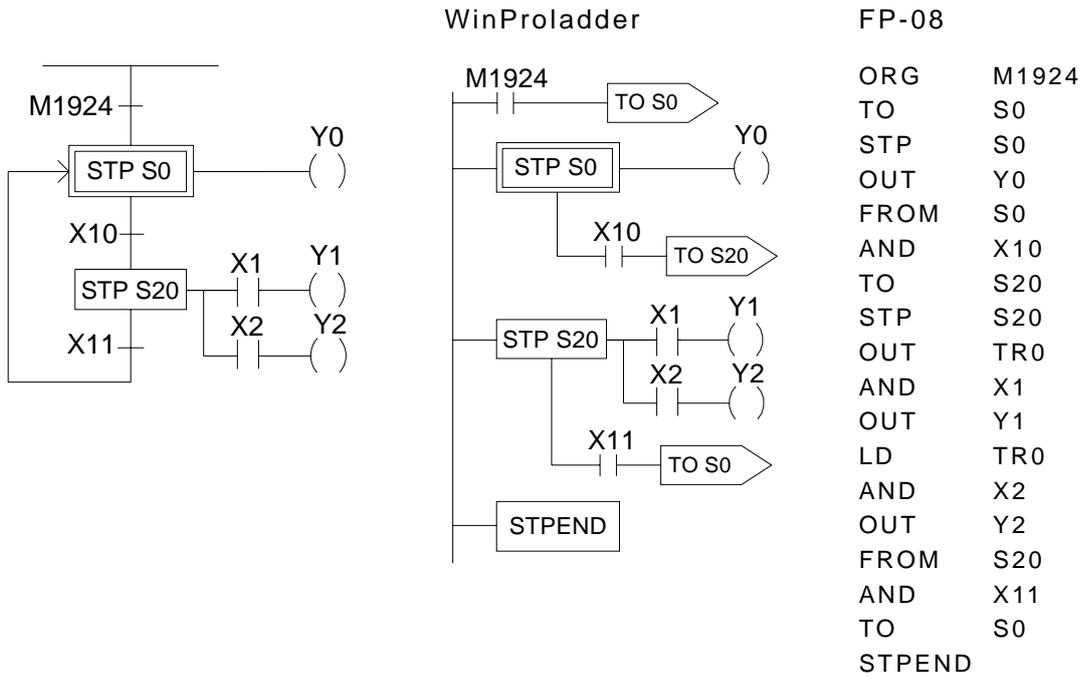
● **STP Sxxx** : $S20 \leq Sxxx \leq S999$ (WinProladder 輸入/顯示格式)

或

STP Sxxx : $S20 \leq Sxxx \leq S999$ (FP-08 輸入按 “STP” Key)

此指令為流程中之步進點指令，每一步進點代表一個步序（站），ON 代表該步序作動，並會執行該步序下的階梯程式。

【範例】



【說明】 1.開機時，初始步進點 S0 ON、Y0 ON。

2.當轉進條件 X10（實際使用時，轉進條件可由 X、Y、M、T、C 各接點之串、並聯組合而成）ON 時，則步進點 S20 作動，當次掃描時間內系統會自動將 S0 OFF，且 Y0 自動清除為 OFF。

$$\text{即 } X10 \text{ ON} \Rightarrow \begin{cases} S20 \text{ ON} \\ S0 \text{ OFF} \end{cases} \Rightarrow \begin{cases} X1 \text{ ON} \rightarrow Y1 \text{ ON} \\ X2 \text{ ON} \rightarrow Y2 \text{ ON} \\ Y0 \text{ OFF} \end{cases}$$

3.當轉進條件 X11 ON 時，則步進點 S0 ON，Y0 也 ON，同時 S20、Y1 和 Y2 變 OFF。

$$\text{即 } X11 \text{ ON} \Rightarrow \begin{cases} S0 \text{ ON} \\ S20 \text{ OFF} \end{cases} \Rightarrow \begin{cases} Y0 \text{ ON} \\ Y1 \text{ OFF} \\ Y2 \text{ OFF} \end{cases}$$

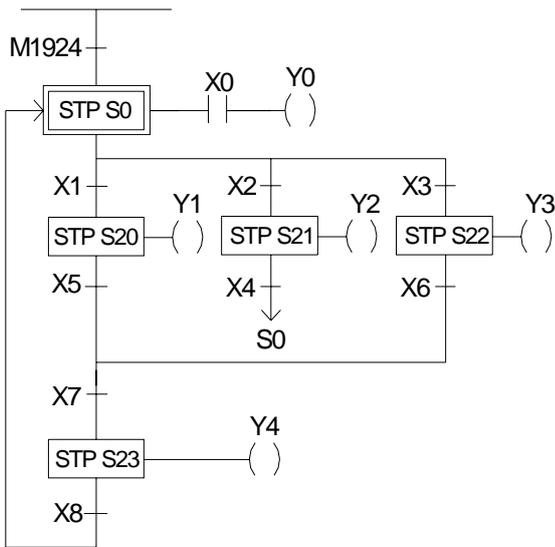
● FROM Sxxx : $S0 \leq Sxxx \leq S999$ (WinProladder 輸入/顯示格式)

或

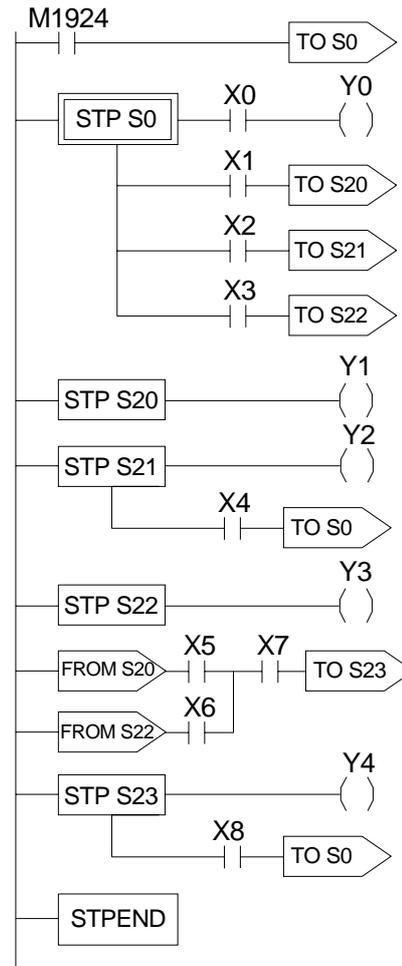
FROM Sxxx : $S0 \leq Sxxx \leq S999$ (FP-08 輸入按 "FROM" Key)

此指令描述轉進之來源步進點，亦即要由步進點 Sxxx 配合轉進條件前進至下一個步進點。

【範例】



WinProladder



FP-08

```

ORG    M1924
TO     S0
STP    S0
AND    X0
OUT    Y0
FROM   S0
OUT TR 0
AND    X1
TO     S20
LD TR 0
AND    X2
TO     S21
LD TR 0
AND    X3
TO     S22
STP    S20
OUT    Y1
STP    S21
OUT    Y2
FROM   S21
AND    X4
TO     S0
STP    S22
OUT    Y3
FROM   S20
AND    X5
FROM   S22
AND    X6
ORLD
AND    X7
TO     S23
STP    S23
OUT    Y4
FROM   S23
AND    X8
TO     S0
STPEND

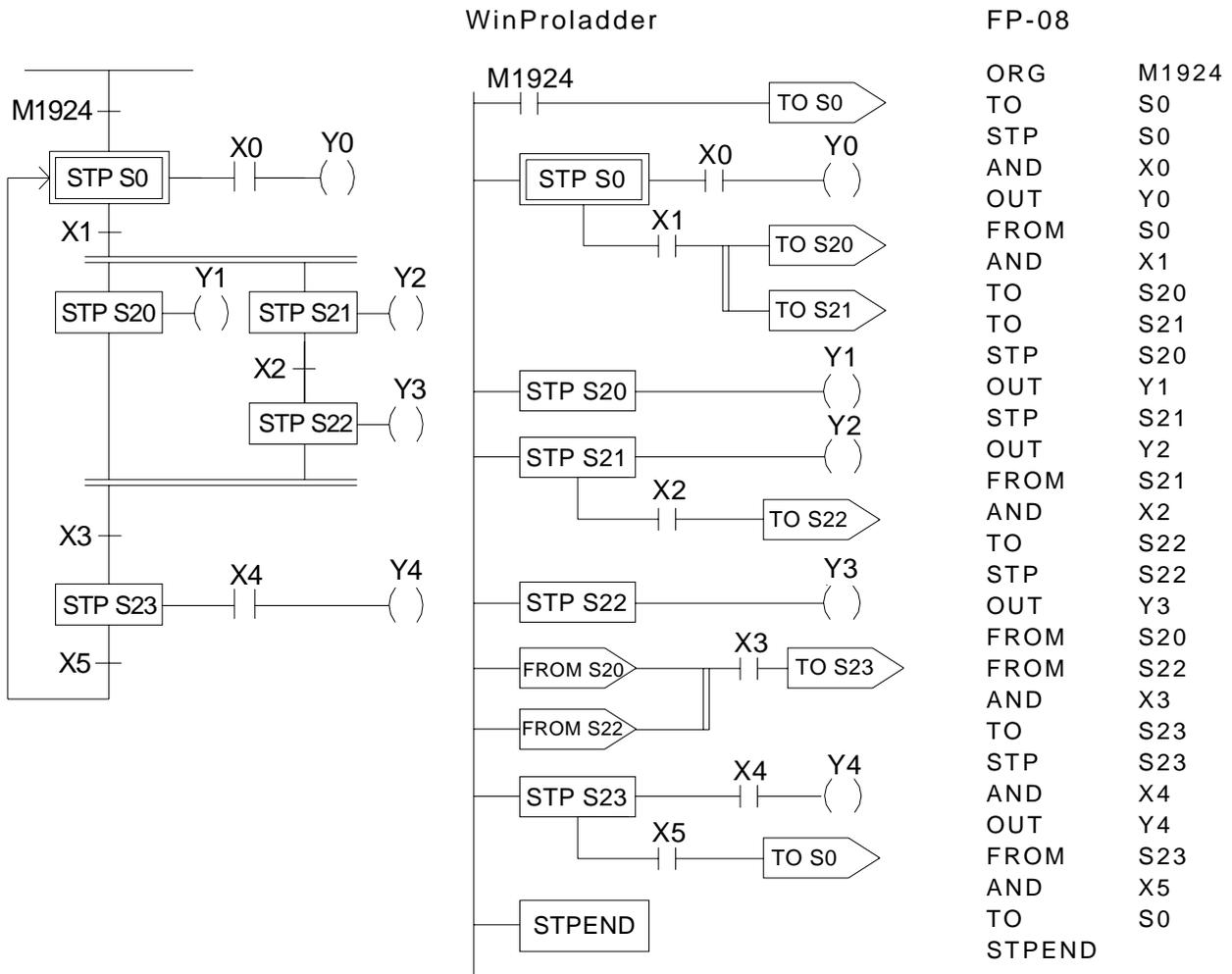
```

- 【說明】：1.開機時進入初始步進點 S0 ON；X0 ON 則 Y0 ON。
- 2.S0 ON 時，
- a.當 X1 ON 時，則步進點 S20 ON、Y1 ON。
 - b.當 X2 ON 時，則步進點 S21 ON、Y2 ON。
 - c.當 X3 ON 時，則步進點 S22 ON、Y3 ON。
 - d.如果 X1、X2 和 X3 同時 ON，則步進點 S20 優先 ON，S21 或 S22 不會 ON。
 - e.如果 X2 與 X3 同時 ON，則步進點 S21 優先 ON，S22 不會 ON。
- 3.S20 ON，當 X5 和 X7 同時 ON 時，則步進點 S23 ON、Y4 ON、S20 OFF、Y1 OFF。
- 4.S21 ON，當 X4 ON 時，則步進點 S0 ON、S21 OFF、Y2 OFF。
- 5.S22 ON，當 X6 和 X7 同時 ON 時，則步進點 S23 ON、Y4 ON、S22 OFF、Y3 OFF。
- 6.S23 ON，當 X8 ON 時，則步進點 S0 ON、S23 OFF、Y4 OFF。

- **TO Sxxx** : $S0 \leq Sxxx \leq S999$ (Winproladder 輸入/顯示格式)
或
TO Sxxx : $S0 \leq Sxxx \leq S999$ (FP-08 輸入按 "TO" Key)

此指令描述欲轉往之步進點。

【範例】



- 【說明】**：
- 1.開機時進入初始步進點 S0 ON；X0 ON 則 Y0 ON。
 - 2.S0 ON，當 X1 ON 時，則同時步進點 S20 ON、S21 ON，兩路並進；Y1 ON，Y2 ON。
 - 3.S21 ON，當 X2 ON 時，步進點 S22 ON、Y3 ON、S21 OFF、Y2 OFF。
 - 4.S20 和 S22 同時 ON 且轉進條件 X3 ON 時，則步進點 S23 ON (X4 ON 時 Y4 ON)；而 S20 和 S22 自動 OFF，Y1 和 Y3 變為 OFF。
 - 5.S23 ON，當 X5 ON 時，則轉進回復至初始步進點，即 S0 ON、S23 OFF、Y4 OFF。

● **STPEND** : (WinProladder 輸入/顯示格式)

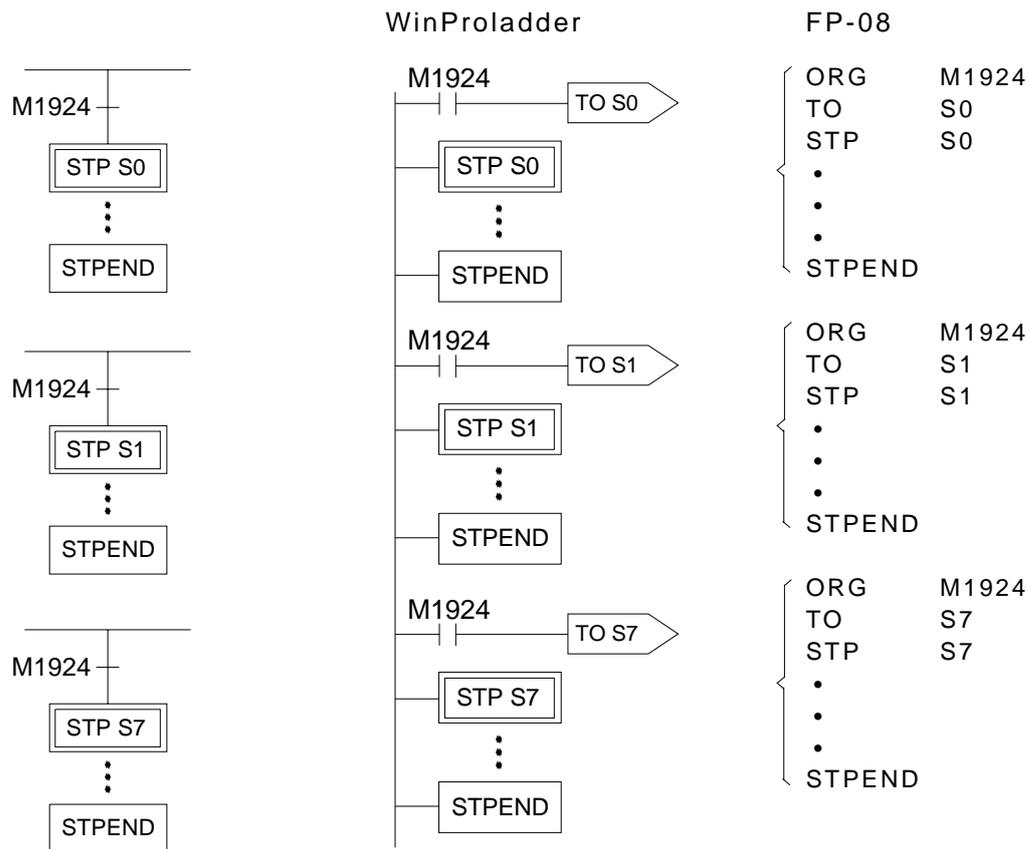
或

STPEND : (FP-08 輸入按 “STP” 和 “END” Key)

此指令代表一流程指令之結束，必須有此指令，所有流程才會正確運作。

PLC 最多有 8 個步進流程 (S0~S7) 可同時控制，所以最多有 8 個 **STPEND** 指令。

【範例】

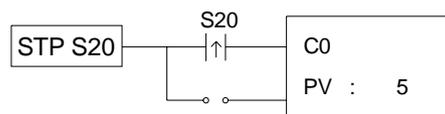


【說明】 開機時 8 個步進流程同時作動。

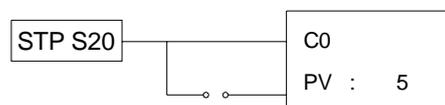
8.4 步進階梯圖寫法

【注意事項】

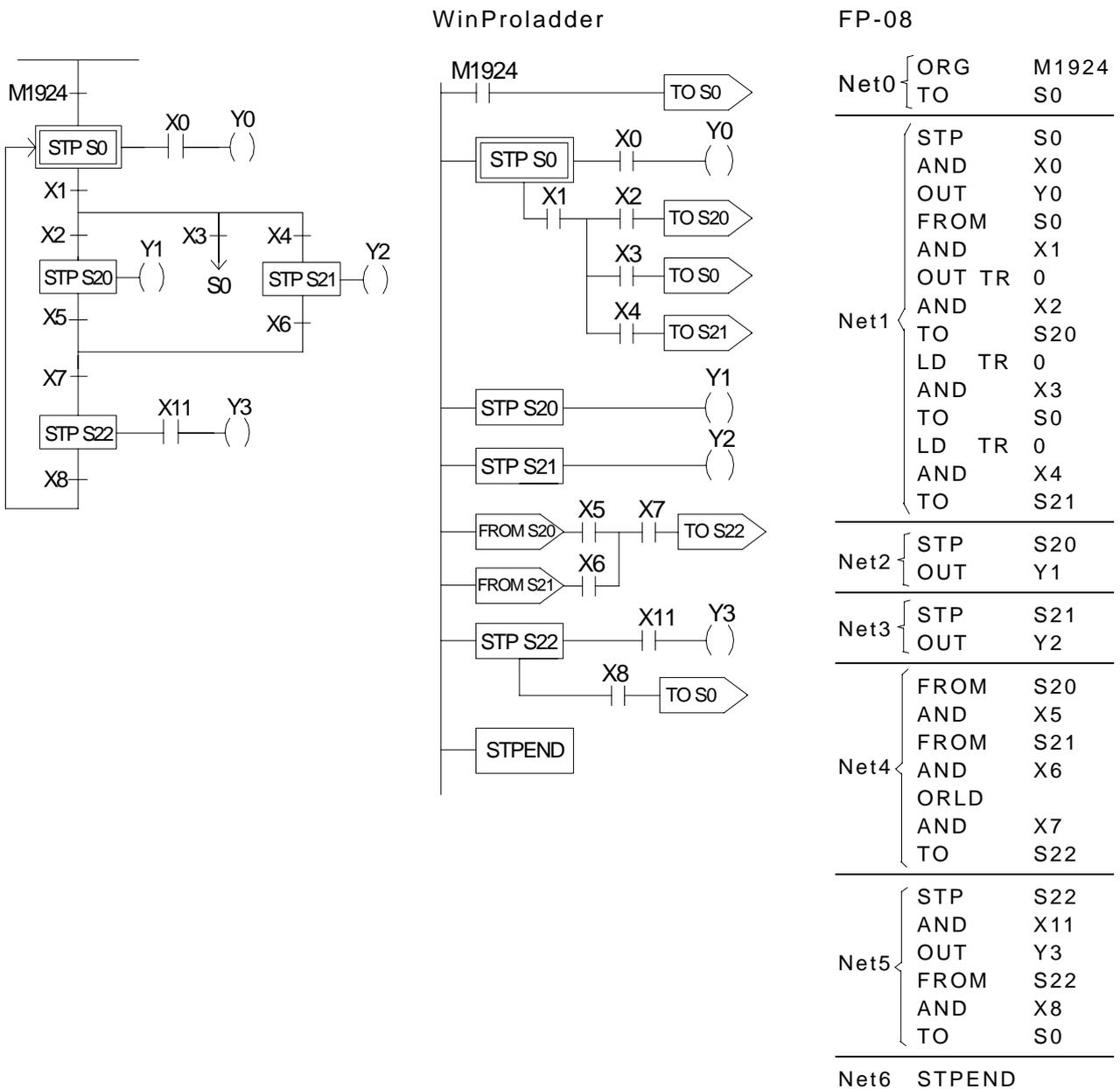
- 在實際的應用上，可將單純的階梯圖和步進階梯圖組合使用。
- 作為開頭的步進點我們稱為初始步進點，共有 8 點，即 S0~S7。
- 要讓初始步進點作動當然可以由任何一個步進點來加以觸發，但 PLC 開始運轉時，必須讓初始步進點 ON；我們可利用系統提供之 M1924（第一次掃描 ON 信號）來觸發初始步進點 ON。
- 除了初始步進點用上述方法觸發啟動外，其他的步進點的觸發必須由另外一個步進點來驅動。
- 在步進階梯圖程式當中必須有開頭之初始步進點，及最後之 STPEND 指令，才算一完整之步進流程程式。
- 一般步進點共有 980 點，由 S20~S999，可任意使用，無須按順序，但號碼絕不可重覆使用。系統內定 S500~S999 為停機保持型（當然可由 USER 修改），機械流程在斷電後如想繼續斷電前之動作，則可使用這些步進點。
- 一個步進點在基本上必須具備驅動步進點內輸出負載、指定轉進條件及轉進目的地等三個功能。
- 在步進程式中不可使用 MC，SKP 指令；副程式區不能輸入步進程式。當然 JMP 指令儘量少用。
- 若步進轉進後，輸出點仍需保持 ON 則需用 SET 指令推動該輸出點；要清除該輸出點為 OFF，則需用 RST 指令。
- 從一個初始步進點往下看，橫向分歧步進點最大 16 點，但一個分歧點最多只可作 8 個分歧回路。
- M1918=0（內定值）時，在 MC（FUN 0）或步進點程式中如需使用 PULSE 型功能指令，則必須在該功能指令前串接一該步進點之 TU 指令，例如



M1918=1 時，則不需加該步進點之 TU 指令，例如：

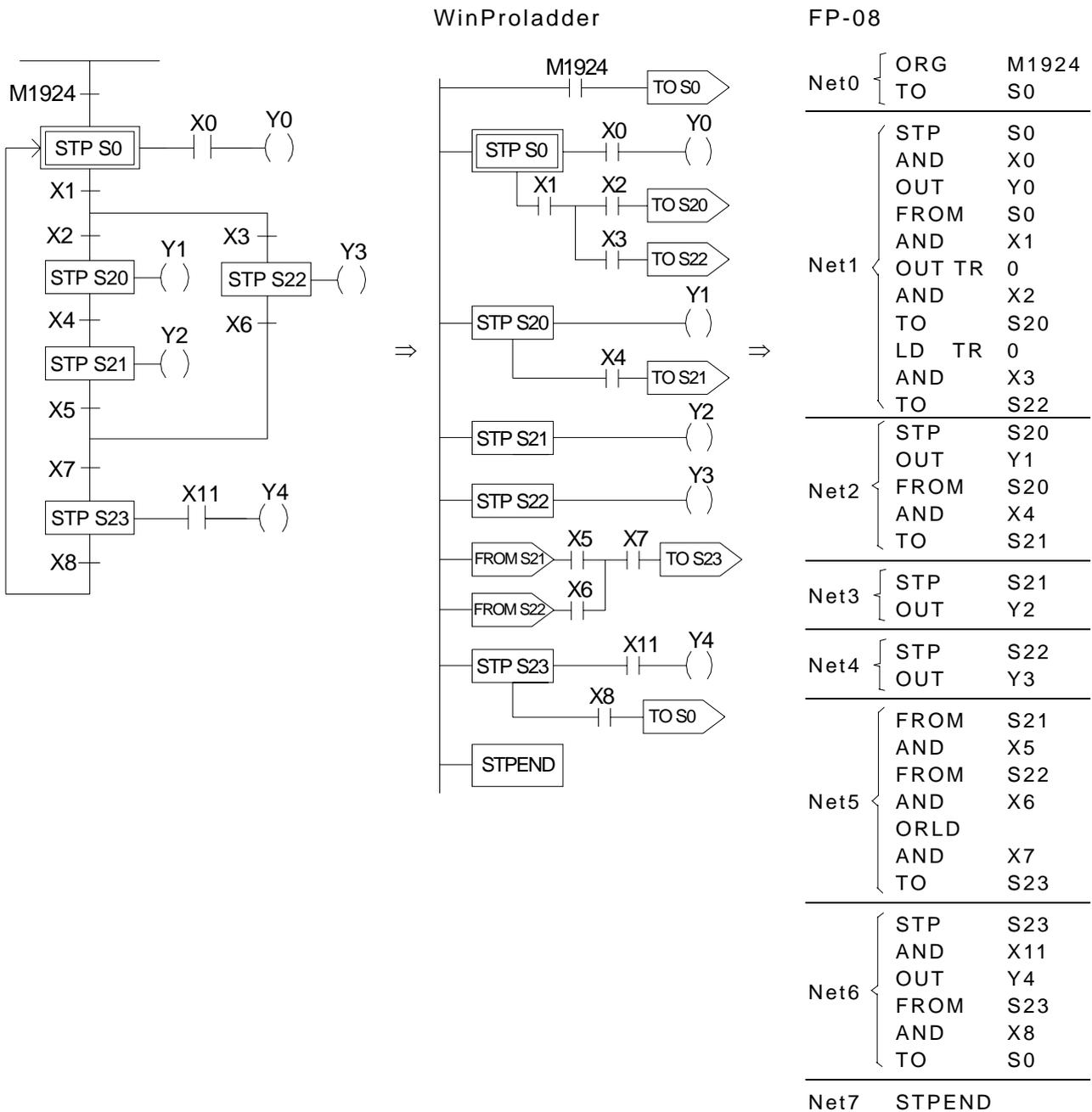


【範例 1】



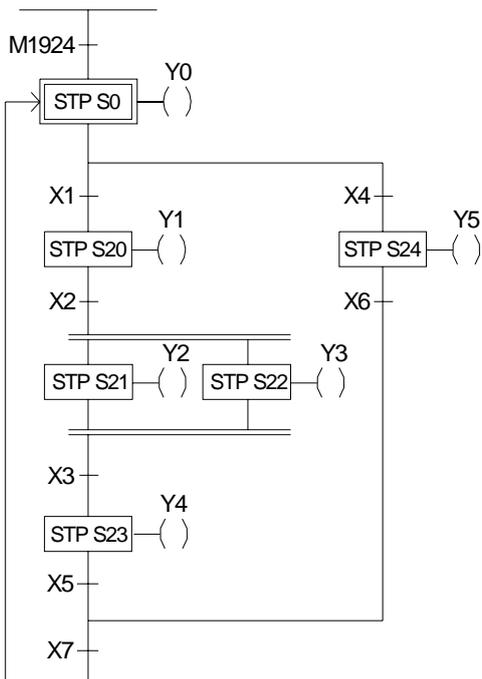
- 【說明】：
1. 編輯初始步進點 S0
 2. 編輯 S20、S0、S21 之分歧
 3. 編輯 S20
 4. 折返編輯 S21
 5. 編輯 S20、S21 之合流
 6. 往下編輯 S22

【範例 2】

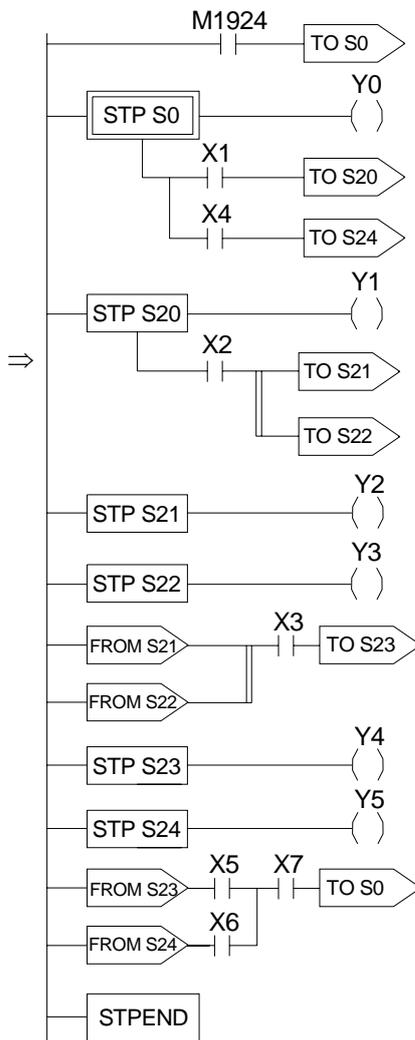


- 【說明】：
1. 編輯初始步進點 S0
 2. 編輯 S20、S22 之分歧
 3. 編輯 S20
 4. 編輯 S21
 5. 折返編輯 S22
 6. 編輯 S21、S22 之合流
 7. 往下編輯 S23

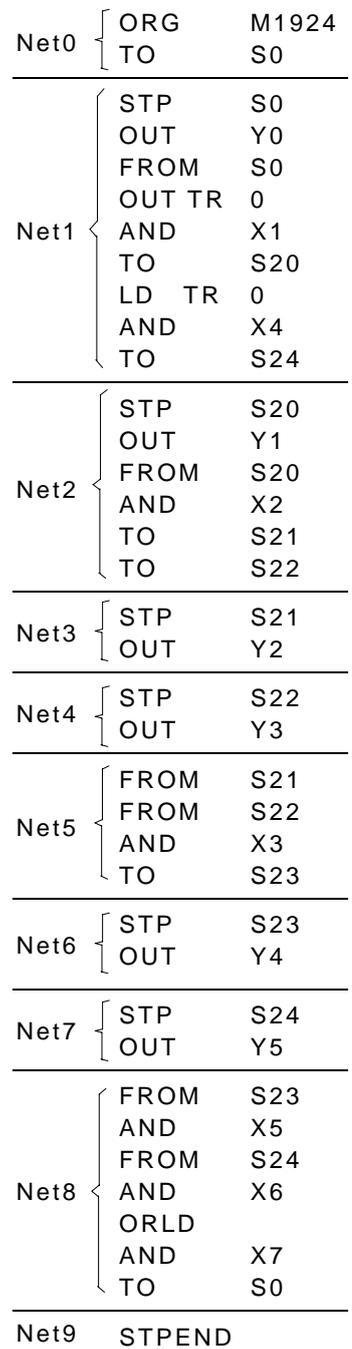
【範例 3】



WinProLadder



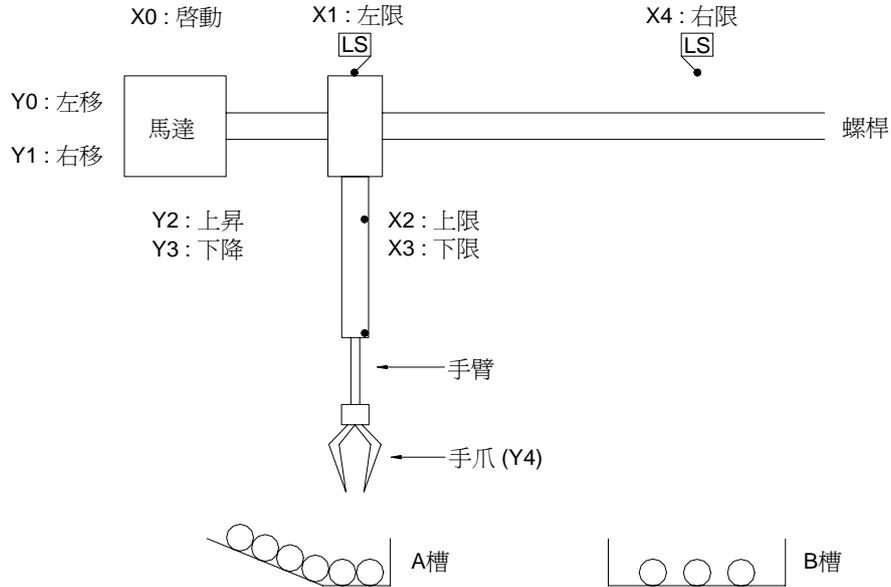
FP-08

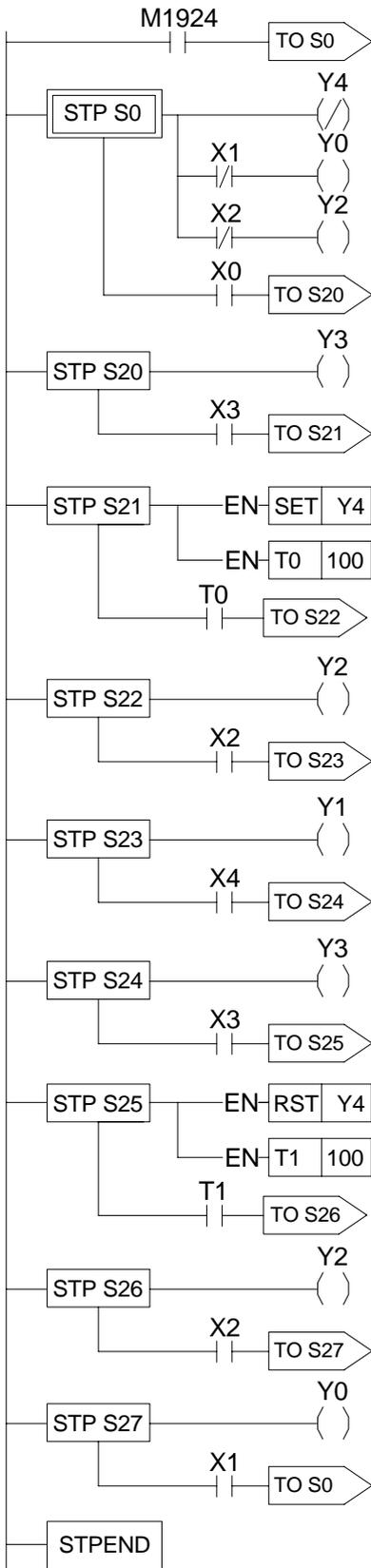


- 【說明】：
1. 編輯初始步進點 S0
 2. 編輯 S20、S24 之分歧
 3. 編輯 S20
 4. 編輯 S21、S22 之分歧
 5. 編輯 S21
 6. 折返編輯 S22
 7. 編輯 S21、S22 之合流
 8. 編輯 S23
 9. 返回上層編輯 S24
 10. 編輯 S23、S24 之合流

8.5 實際應用範例

【範例 1】自 A 槽抓取物體放至 B 槽內



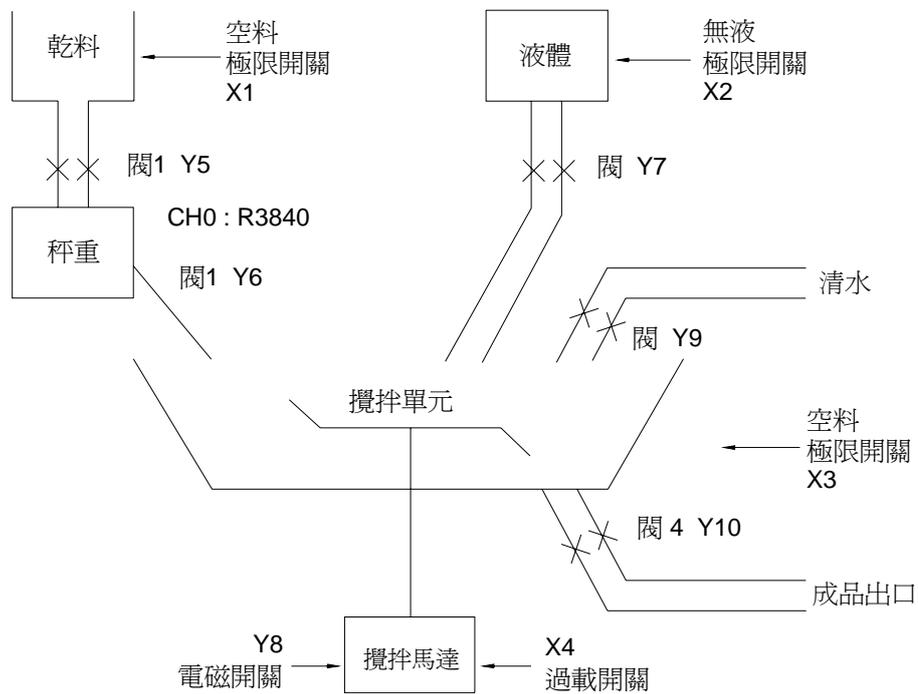


釋放手爪
回左限
回上限
啓動開關 ON 後
移行至 S20
手臂下伸
伸至下限後
移行至 S21
手爪抓取(因用 SET 指令
故 STP S21 離開後, Y4
仍保持 ON)
1 秒後轉進 S22
手臂上昇
到上限後轉進 S23
手臂右移
移至右限後轉進 S24
手臂下伸
伸至下限後轉進 S25
手爪鬆開
1 秒鐘延遲
1 秒鐘後轉進 S26
手臂上昇
昇至上限後轉進 S27
手臂左移
待至左限後, 轉進
S0(一完整 CYCLE)

```

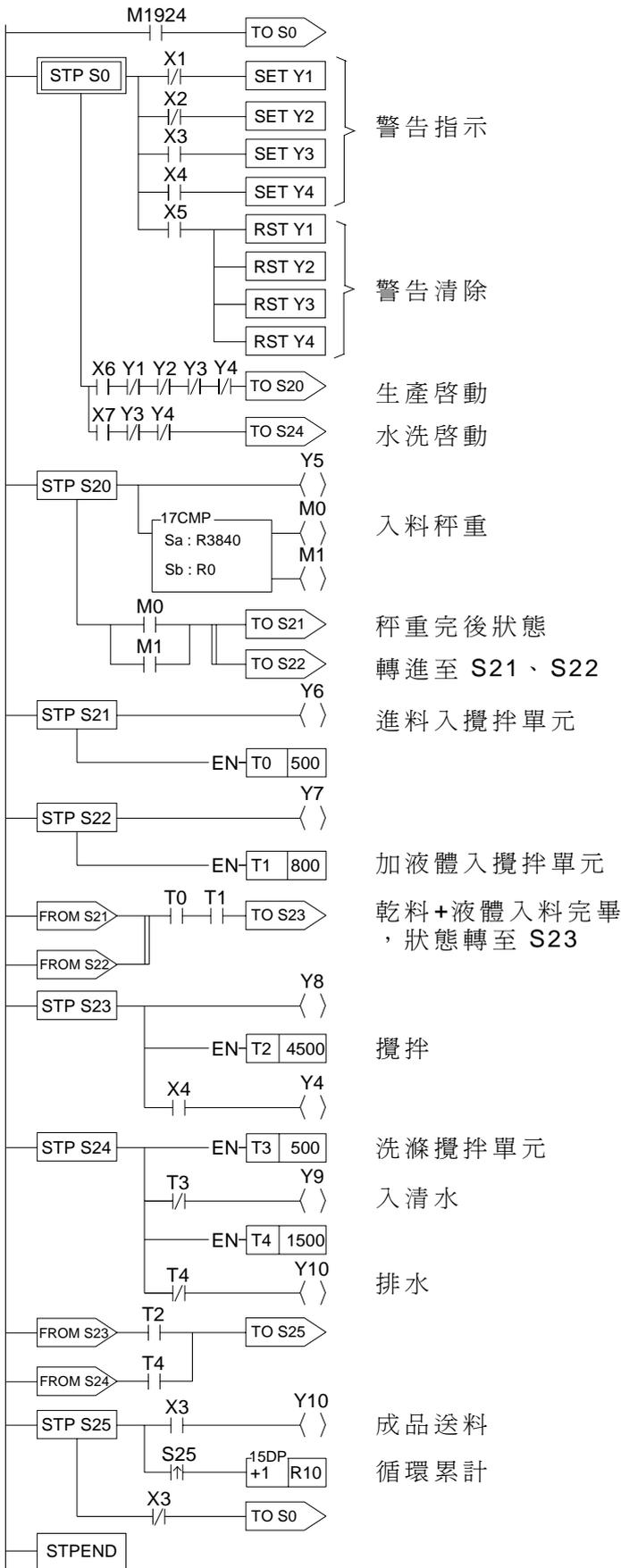
ORG      M1924
TO       S0
STP      S0
OUT TR   0
OUT NOT  Y4
AND NOT  X1
OUT      Y0
LD TR    0
AND NOT  X2
OUT      Y2
FROM     S0
AND      X0
TO       S20
STP      S20
OUT      Y3
FROM     S20
AND      X3
TO       S21
STP      S21
SET      Y4
T0       PV: 100
FROM     S21
AND      T0
TO       S22
STP      S22
OUT      Y2
FROM     S22
AND      X2
TO       S23
STP      S23
OUT      Y1
FROM     S23
AND      X4
TO       S24
STP      S24
OUT      Y3
FROM     S24
AND      X3
TO       S25
STP      S25
RST      Y4
T1       PV: 100
FROM     S25
AND      T1
TO       S26
STP      S26
OUT      Y2
FROM     S26
AND      X2
TO       S27
STP      S27
OUT      Y0
FROM     S27
AND      X1
TO       S0
STPEND
    
```

【範例 2】液體攪拌處理



- 輸 入 點：空料極限開關 X1
無液極限開關 X2
空料極限開關 X3
過載開關 X4
警告清除鈕 X5
啓動鈕 X6
水洗鈕 X7
- 警告指示燈：乾料空料 Y1
液體缺液 Y2
攪拌單元空料 Y3
馬達過載 Y4
- 輸 出 點：乾料入料閥 Y5
乾料入料閥 Y6
液體入料閥 Y7
啓動馬達電磁閥 Y8
清水入水閥 Y9
成品送料閥 Y10
- 秤 重 輸 入：CH0 (R3840)
- M1918=0

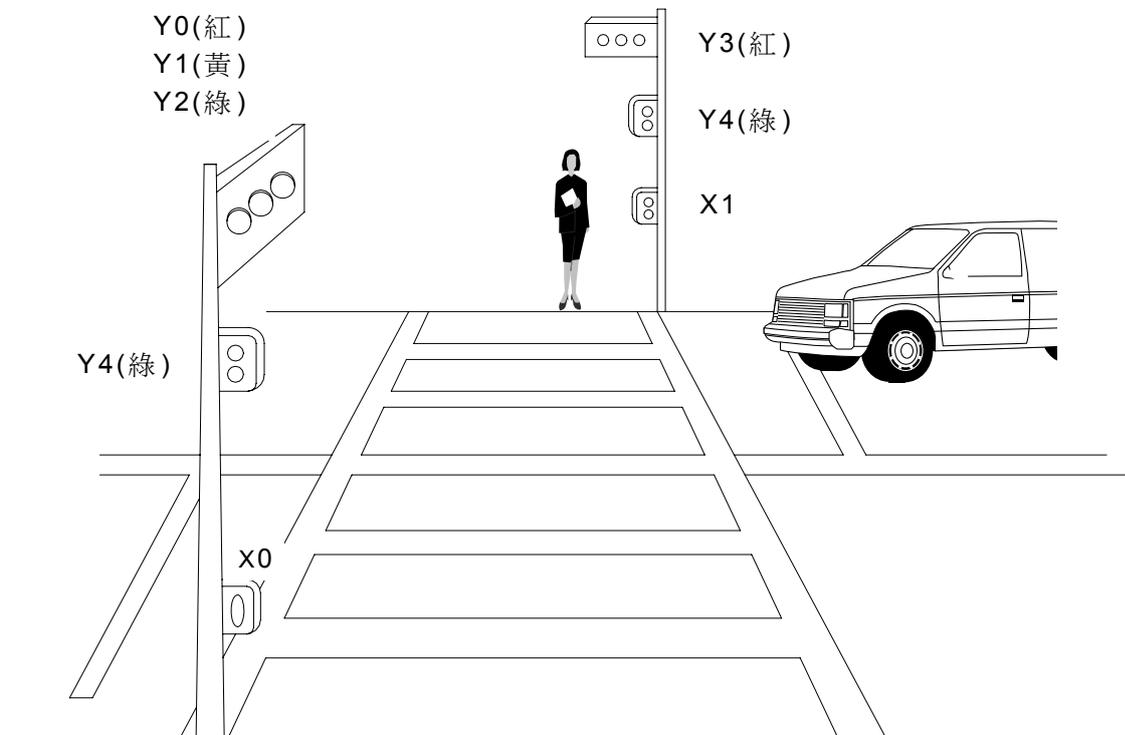
WinProLadder



FP-08

ORG	M1924	STP	S22
TO	S0	OUT	Y7
STP	S0	T1	PV: 800
OUT TR	0	FROM	S21
AND NOT	X1	FROM	S22
SET	Y1	AND	T0
LD TR	0	AND	T1
AND NOT	X2	TO	S23
SET	Y2	STP	S23
LD TR	0	OUT TR	0
AND	X3	OUT	Y8
SET	Y3	LD TR	0
LD TR	0	T2	PV: 4500
AND	X4	LD TR	0
SET	Y4	AND	X4
LD TR	0	OUT	Y4
AND	X5	STP	S24
RST	Y1	OUT TR	0
RST	Y2	T3	PV: 500
RST	Y3	LD TR	0
RST	Y4	AND NOT	T3
FROM	S0	OUT	Y9
OUT TR	1	LD TR	0
AND	X6	T4	PV: 1500
AND NOT	Y1	LD TR	0
AND NOT	Y2	AND NOT	T4
AND NOT	Y3	OUT	Y10
AND NOT	Y4	FROM	S23
TO	S20	AND	T2
LD TR	1	FROM	S24
AND	X7	AND	T4
AND NOT	Y3	ORLD	
AND NOT	Y4	TO	S25
TO	S24	STP	S25
STP	S20	OUT TR	0
OUT	Y5	AND	X3
FUN	17	OUT	Y10
	Sa:R3840	LD TR	0
	Sb:R0	AND TU	S25
FO	0	FUN	15DP
OUT	M0		D:R10
FO	1	FROM	S25
OUT	M1	AND NOT	X3
FROM	S20	TO	S0
LD	M0	STPEND	
OR	M1		
ANDLD			
TO	S21		
TO	S22		
STP	S21		
OUT	Y6		
T0	PV: 500		

【範例 3】人行道紅綠燈

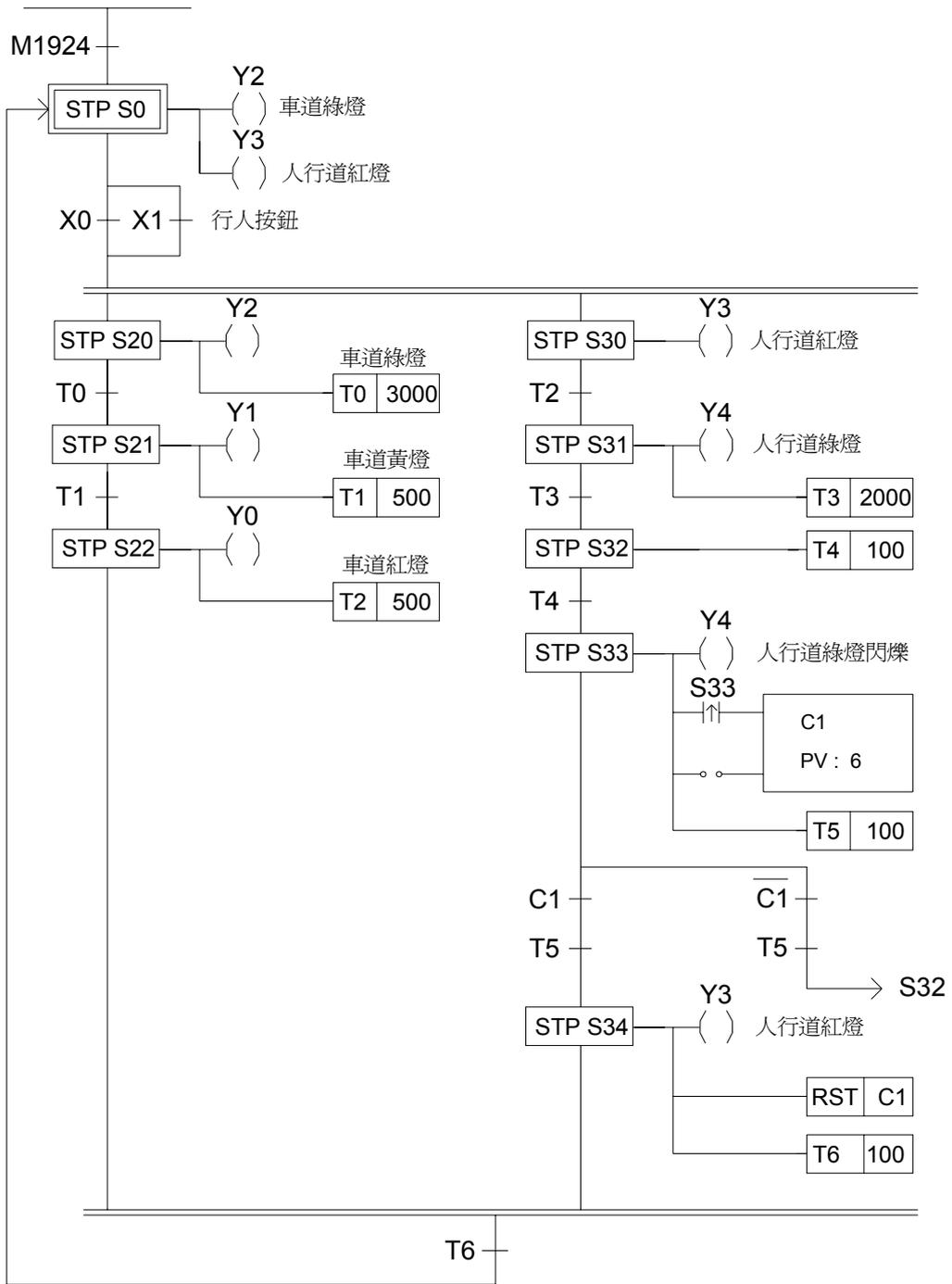


- ◆ 輸入點：行人按鈕 X0
行人按鈕 X1

- ◆ 輸出點：車道紅燈 Y0
車道黃燈 Y1
車道綠燈 Y2
人行道紅燈 Y3
人行道綠燈 Y4

- ◆ M1918=0

人行道紅綠燈控制流程圖



8.6 步進程式語法檢查錯誤碼說明

步進語言程式、語法檢查錯誤之編號如下：

- E51 : TO(S0~S20)必需以 ORG 為起始指令
- E52 : TO(S20~S999)不得以 ORG 為起始指令
- E53 : 同一網路中，TO(S20~S999)前，必需在有 FROM
- E54 : TO 之前一指令，必需為 TO、AND、OR、ANDLD、ORLD
- E56 : 此時 FROM 之前一指令，必需為 FROM 或 AND、OR、ANDLD、ORLD
- E57 : OUT、TMR、CTR、FUN 不與 TO(S0~S19)並存於同一網路中
- E58 : OUT、TMR、CTR、FUN 前一 STEP 指令必需為 STP
- E59 : 同一網路中，TO 超過 8 個
- E60 : 同一網路中，FROM 超過 8 個
- E61 : TO(S0~S19) 必需為網路第一列
- E62 : 接點佔據 TO 位置
- E71 : 連繫不完整(理應不會發生)
- E72 : TO Sxx 重複
- E73 : STP Sxx 重複
- E74 : FROM Sxx 重複
- E76 : 上一個 STP(S0~S19)缺乏 STPEND 或 STPEND 往前找不到相對應之 STP(S0~S19)
- E77 : STP(S0~S19)的前一個網路並非是以 ORG 為起始唯一之 TO(S0~S19)
- E78 : 尚未使用 STP(S0~S19)就使用 TO(S20~S999)、STP (S20~S999)、FROM
- E79 : 尚未使用 TO Sxx 就使用STP Sxx 或 FROM Sxx
- E80 : 尚未使用 STP Sxx 就使用 FROM Sxx
- E81 : 同一時間，尚未處理之分歧層數不得大於 16
- E82 : 同一時間，分歧中尚未處理之分枝不得大於 16
- E83 : 單一步進點，未依照 TO Sxx→STP Sxx→FROM Sxx 之順序且緊密連繫
- E84 : 進入分歧後，需依照由左至右之順序來處理分枝
- E85 : 合流時，與先前之分歧情況不對應
- E86 : 尚未利用 TO 來完成上一個合流，就使用 STP 或 FROM
- E87 : 尚未利用 FROM+TO 來轉移上一個 STP，就使用 STP 或 FROM
- E88 : 分歧中，STP Sxx 或 FROM Sxx，在此分歧內，往前找不到相對應之 TO Sxx
- E89 : 尚未利用 STP 來承接 TO 之處理，就使用 FROM
- E90 : 並進式分歧之轉接不合法
- E91 : 上一個 STP(S0~S19)尚未處理完全，就使用 ORG、LBL、RTS、RTI、MCE、SKPE、FOR、NEXT、END