

## **CHƯƠNG 3**

### **KỸ THUẬT LẬP TRÌNH ĐIỀU KHIỂN TUẦN TỰ**

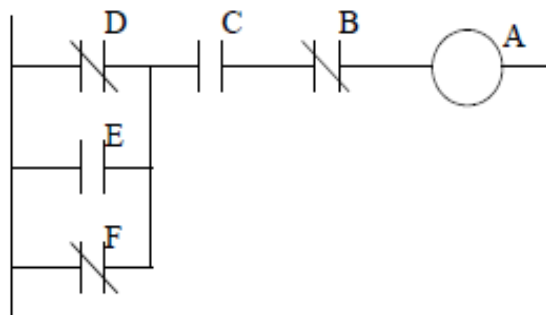
#### **I. Thiết kế mạch điều khiển.**

Các bước thiết kế chương trình trình tự cho PLC như sau:

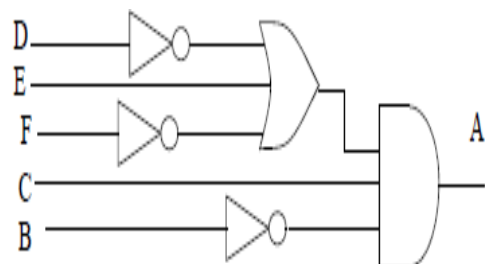
1. Quá trình điều khiển được diễn đạt bằng lời.
2. Sự mô tả đó được chuyển sang dạng lưu đồ hay sơ đồ chức năng.
3. Đến giai đoạn này, các điều kiện logic dễ dàng được xác định, sau đó chuyển sang biểu thức Boolean biểu diễn từng trạng thái của quá trình trình tự.
4. Cuối cùng biểu thức Boolean được chuyển đổi sang chương trình dạng ladder.

Sự diễn đạt bằng lời hay ghi ra giấy mô tả quá trình điều khiển thường dài, khó theo dõi và không chính xác. Để toàn bộ quá trình điều khiển sẽ dễ hơn khi chia nó ra thành những chương trình con. Mỗi chương trình con sau đó có thể được xây dựng theo dạng trình tự và khóa lần để thực hiện một chức năng nào đó theo yêu cầu. Cần phải có các phương pháp mô tả một hệ thống trình tự như trên sao cho rõ ràng và dễ theo dõi quá trình hoạt động.

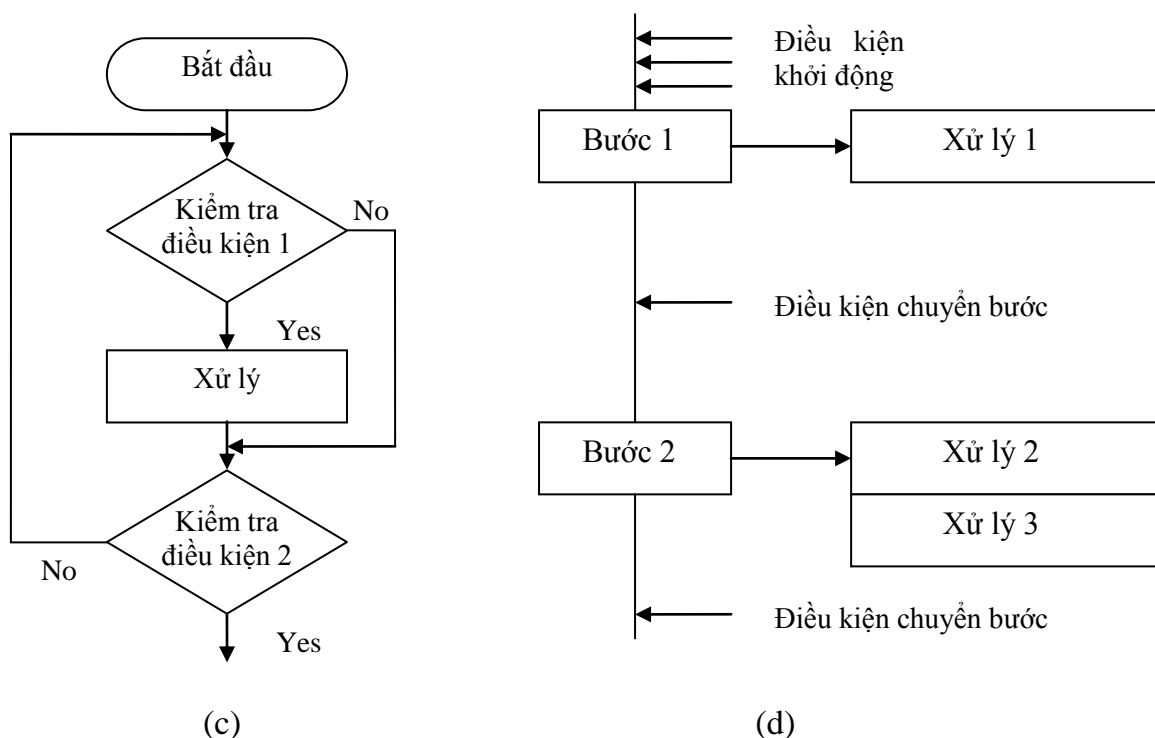
Các phương pháp diễn đạt có thể tùy chọn: logic relay (relay logic diagram), cổng logic (logic schematic), lưu đồ ( flowchart) và sơ đồ chức năng (function chart). Các phương pháp này không thay thế cho bước diễn đạt bằng lời mà nó hỗ trợ rất nhiều cho bước này. Việc áp dụng phương pháp nào tùy thuộc chủ yếu vào kinh nghiệm về phương pháp đó. Người phân tích thiết kế hệ thống có kiến thức tốt về điện tử kỹ thuật số hay về máy tính thì thường dùng ba phương pháp sau, còn phương pháp logic relay được dùng đối với những người quen với thiết kế mạch relay.



(a)



(b)



Hình 3.1: Các phương pháp đặc tả hệ thống điều khiển logic: (a) logic relay; (b) cổng logic; (c) lưu đồ; (d) sơ đồ chức năng.

#### ❖ Phương pháp logic relay và cổng logic.

Cả hai phương pháp có liên hệ trực tiếp đến mạch vật lý, vì vậy nó là các phương pháp lý tưởng cho các ứng dụng trong đó PLC thay thế cho hệ thống dùng relay truyền thống, vì thế các bản vẽ về hệ thống nguyên thủy có thể được dùng làm cơ sở để lập trình cho PLC. Tuy nhiên, các phương pháp này thường dùng cho các hệ thống điều khiển dùng tổ hợp các ngõ vào hay các hệ thống điều khiển trình tự quy mô nhỏ, vì sơ đồ biểu diễn sẽ trở nên phức tạp và rất khó theo dõi đối với các ứng dụng trình tự quy mô lớn.

#### ❖ Phương pháp biểu diễn lưu đồ.

Phương pháp này thường dùng khi thiết kế phần mềm cho máy tính, nhưng là phương pháp phổ biến để biểu diễn trình tự hoạt động của một hệ thống điều khiển. Lưu đồ có quan hệ trực tiếp đến sự mô tả bằng lời hệ thống điều khiển, chỉ ra từng điều kiện cần kiểm tra ở từng bước và các xử lý trong bước đó theo chuỗi trình tự. Các xử lý trong lưu đồ được ghi trong một ô chữ nhật, trong khi các điều kiện được ghi vào các ô hình thoi. Tuy nhiên, phương pháp này chiếm nhiều không gian khi biểu diễn các hệ thống điều khiển lớn và sơ đồ trở nên nặng nề.

#### ❖ Phương pháp sơ đồ chức năng.

Phương pháp này càng trở nên phổ biến để biểu diễn các hoạt động trình tự, cho phép thể hiện chi tiết về các xử lý cũng như trình tự các hoạt động trong quá trình điều khiển. Với cách dùng các ký hiệu gọn và cô đọng, phương pháp này có

được ưu điểm của các phương pháp trên, việc biểu diễn bước tiến trình hoạt động mạch lạc và rõ ràng. Trong từng bước ta có thể ghi ra các điều kiện set và reset, điều kiện chuyển trạng thái và các tín hiệu điều khiển khác. Sơ đồ chức năng còn có thể hỗ trợ đắc lực khi kiểm tra và chạy thử hệ thống.

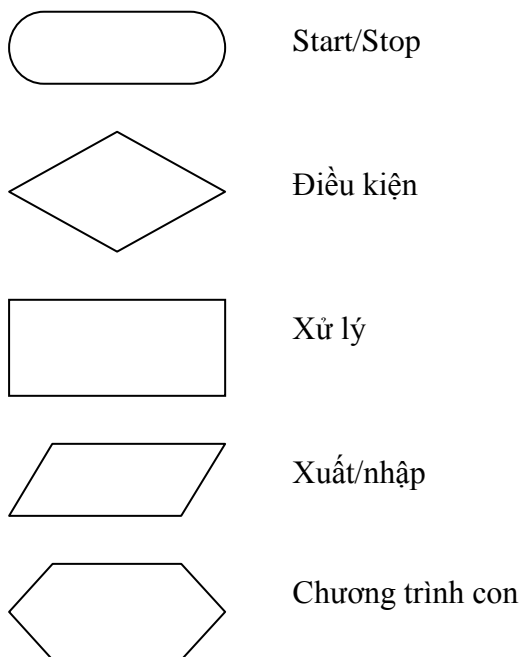
#### **❖ Đại số Boolean.**

Cho dù dùng phương pháp nào, một khi các chức năng được đặc tả rõ ràng thì chúng phải được chuyển sang dạng mà từ đó có thể chuyển thành chương trình PLC. Quá trình này thường được thực hiện bằng cách chuyển đổi các chức năng thành chuỗi liên tiếp các biểu thức Boolean và từ đó chuyển thành ngôn ngữ PLC.

Ta có thể đặc tả toàn bộ hệ thống điều khiển logic chỉ bằng biểu thức Boolean, mặc dù việc dùng biểu thức Boolean thường kém hiệu quả về mặt thời gian thiết kế và không dễ hiểu đối với người chưa có kinh nghiệm về các hệ thống điều khiển. Giải pháp dùng Boolean tiết kiệm được không gian biểu diễn trên giấy khi thiết kế.

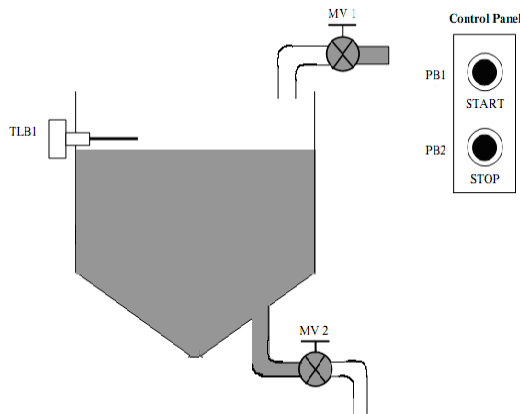
## **II. Phương pháp lưu đồ giải thuật.**

Lưu đồ giải thuật là một phương pháp biểu diễn cho một quá trình điều khiển tuần tự. Các bước ý tưởng sẽ được thực hiện bằng các lệnh một cách đơn giản. Các ký hiệu được sử dụng trong lưu đồ giải thuật:



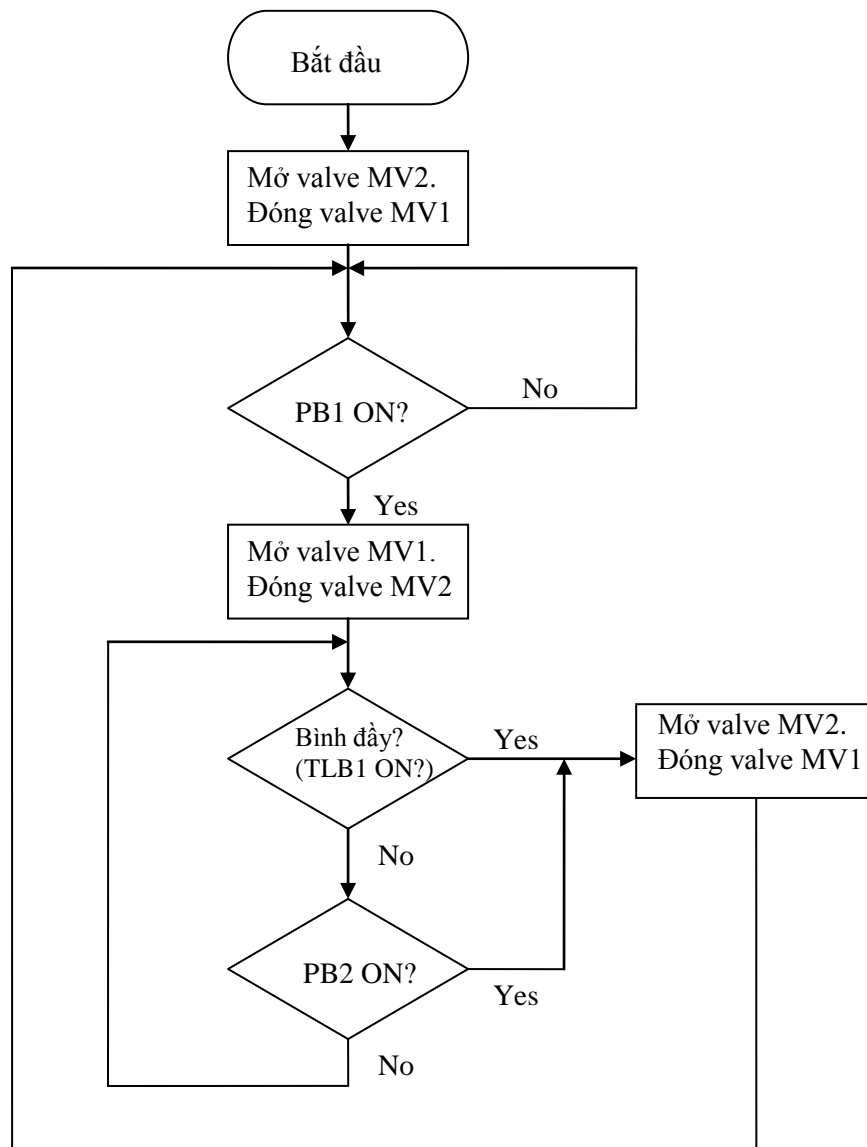
Các khối trên được kết nối bằng mũi tên để chỉ ra các bước liên tiếp của chương trình. Những khối khác nhau thể hiện cho những hoạt động khác nhau của chương trình. Chương trình thì luôn cần khối start, nhưng những chương trình PLC hiếm khi dừng lại vì thế khối stop hiếm khi được sử dụng. Những khối quan trọng khác bao gồm khối xử lý và khối điều kiện. Những chức năng khác có thể sử dụng nhưng không cần thiết cho đa số những ứng dụng PLC.

**Ví dụ 1:**



Hình 3.2: Sơ đồ hệ thống cấp nước

Nhấn nút PB1 (START), bình nước sẽ được đổ đầy qua valve MV1 (valve MV2 sẽ được đóng lại). Khi bình nước đầy (TLB1 ON) hay nút PB2 (STOP) được tác động thì valve MV2 được mở, valve MV1 được đóng lại.



Phương pháp chung của lưu đồ giải thuật:

1. Hiểu rõ quá trình hoạt động.
2. Xác định những hoạt động chính, được thể hiện bằng các khối.
3. Xác định các chuỗi thao tác, kết nối các khối bằng mũi tên.
4. Khi chuỗi thao tác thay đổi thì dùng khối điều kiện để rẽ nhánh.

**Ví dụ 2:**

Dùng lưu đồ thiết kế bộ điều khiển cửa garage. Garage hoạt động như sau:

- Có một nút nhấn đơn trong garage (PB1), và một nút nhấn đơn dùng điều khiển từ xa (PB2) .
- Khi nút nhấn tác động cửa sẽ dịch chuyển (đóng hoặc mở).
- Nếu nút nhấn tác động lần thứ nhất trong khi cửa đang chuyển động, cửa sẽ dừng lại, nút nhấn được tác động lần thứ 2 thì cửa sẽ chuyển động lần nữa theo hướng ngược lại.
- Hai contact hành trình bố trí ở vị trí cao (LS1) và thấp (LS2) nhằm giới hạn đoạn đường dịch chuyển của cửa garage.
- Một chùm tia sáng (S1) được bố trí dưới chân cửa garage, khi chùm tia sáng bị chặn khi cửa đang đóng thì cửa sẽ dừng rồi chạy ngược lại.
- Một bóng đèn trong garage sẽ được bật trong thời gian 5 phút sau khi cửa được đóng hoặc mở.

PB1: nút nhấn trong garage.

PB2: nút nhấn điều khiển từ xa

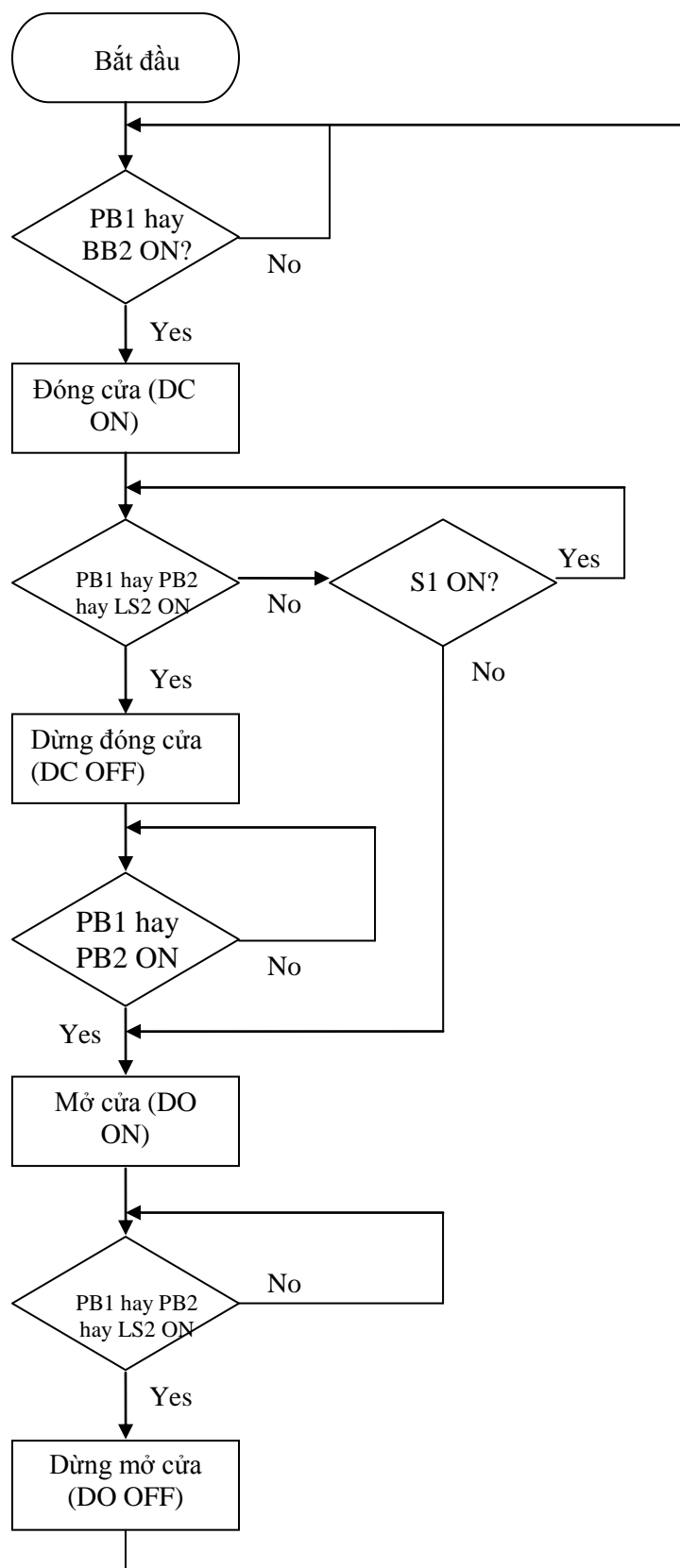
LS1: contact hành trình giới hạn trên.

LS2: contact hành trình giới hạn dưới.

S1: chùm tia sáng ở dưới chân cửa garage.

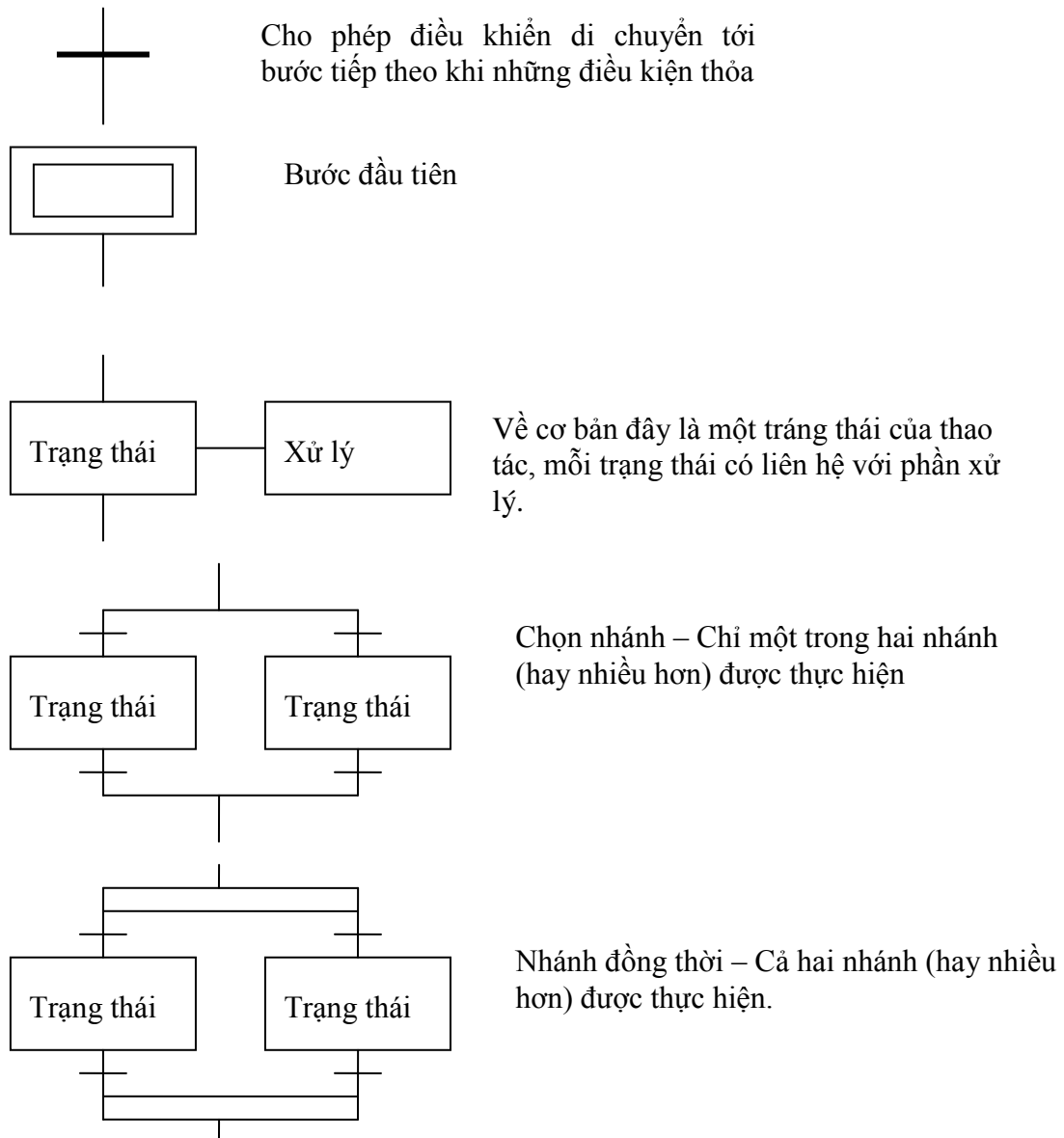
DC: đóng cửa.

DO: mở cửa.



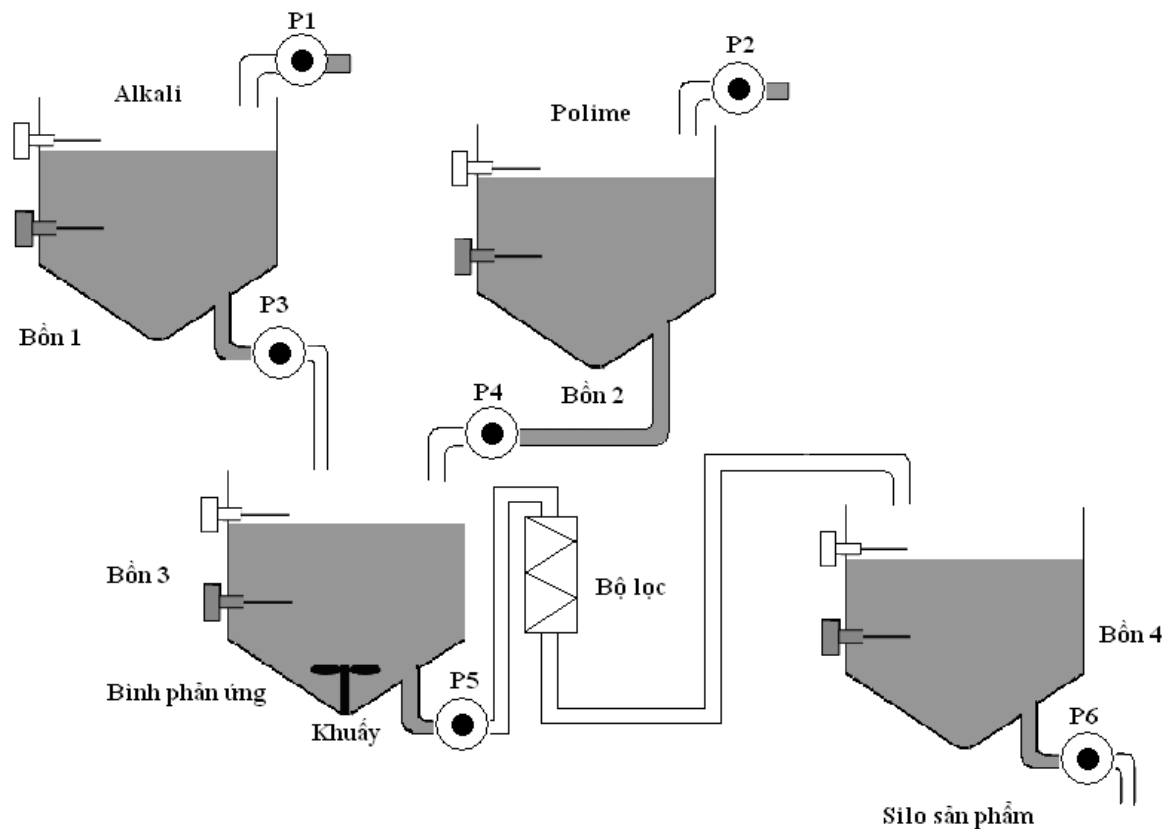
**III. Phương pháp sơ đồ chức năng.**

Phương pháp này thể hiện chi tiết về các xử lý cũng như trình tự các hoạt động trong quá trình điều khiển. Trong từng bước ta có thể ghi ra các lệnh set và reset, điều kiện chuyển trạng thái và các tín hiệu điều khiển khác.



**❖ Ví dụ điều khiển máy xử lý hóa chất.**

Hệ thống gồm 4 bồn chứa có các bơm để chuyển chất lỏng qua hệ thống. Mỗi bồn có gắn các cảm biến để phát hiện bồn cạn hay đầy và bồn 2 có phản tử nung nóng với cảm biến nhiệt độ. Bồn 3 được gắn 1 cần khuấy để trộn hai thành phần chất lỏng khi chúng được bơm vào bồn 1 và bồn 2. Các bồn phía dưới, bồn 3 và bồn 4, có dung tích lớn gấp đôi bồn 1 bồn 2, và được đổ đầy bởi bồn 1 và bồn 2 (chất kiềm với chất polime).

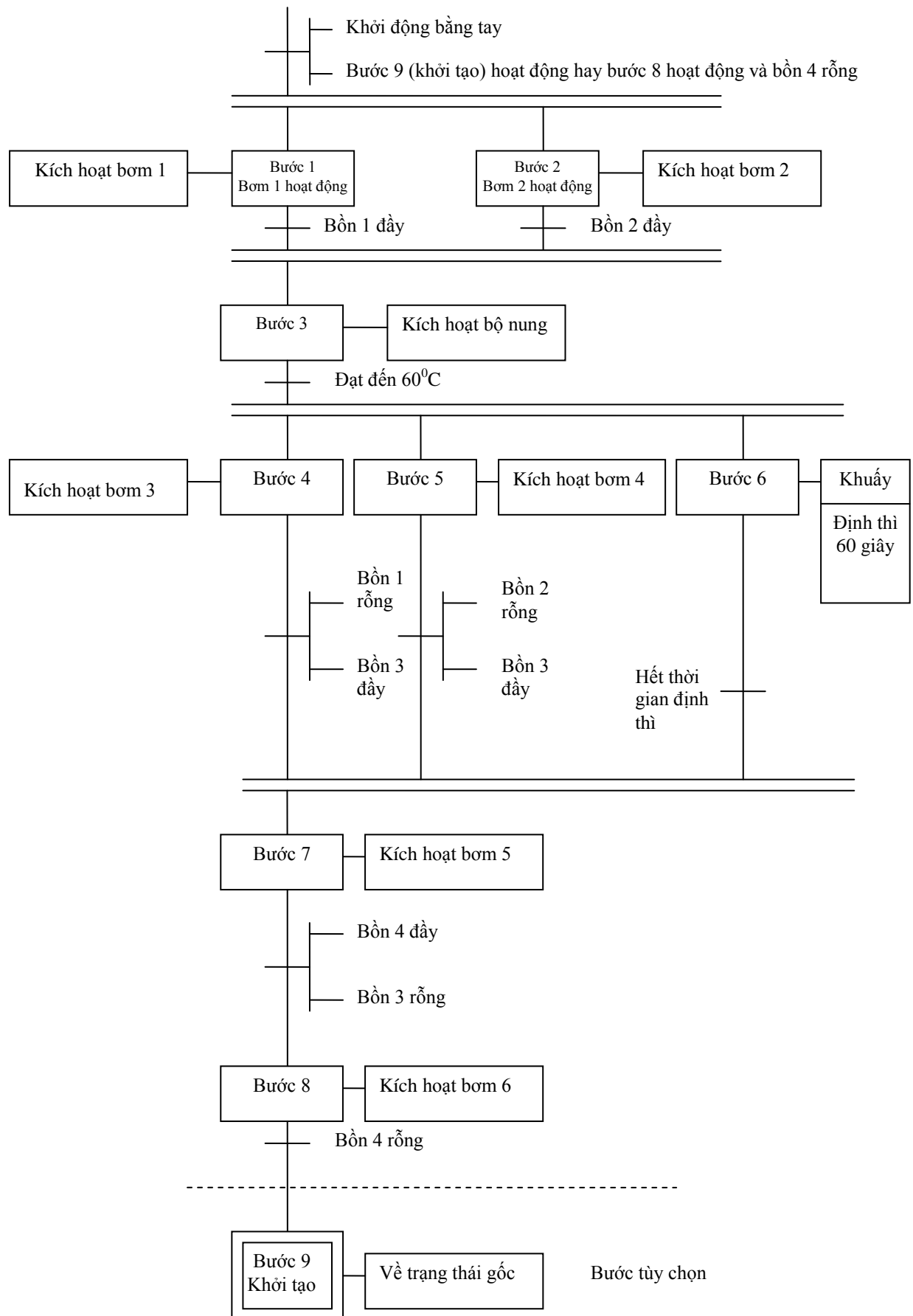


Hình 3.3: Sơ đồ hệ thống xử lý hóa chất.

### Hoạt động

Bồn 1 và 2 được đổ đầy từ các bồn chứa kiềm và polime riêng biệt, thông qua bơm 1 và bơm 2. Bơm 1 và 2 ngưng hoạt động khi có tín hiệu từ cảm biến báo đầy bồn. Phần tử nung nóng trong bồn 2 được kích hoạt, nâng nhiệt độ lên  $60^{\circ}\text{C}$ . Khi cảm biến nhiệt độ nóng, tín hiệu này sẽ tắt bộ điều khiển nung và kích hoạt bơm 3 và bơm 4 để chuyển hỗn hợp dung dịch vào bồn phản ứng, bồn 3. Cần khuấy cũng được kích hoạt khi bồn này có hỗn hợp dung dịch và trong khoảng thời gian tối thiểu 60 giây. Khi bồn 3 đầy, bơm 3 và bơm 4 ngưng hoạt động. Nếu thời gian khuấy lớn hơn 60 giây, bơm 5 sẽ chuyển hỗn hợp đã trộn vào bồn 4, bồn silo sản phẩm, thông qua bộ lọc. Bơm 5 dừng hoạt động khi bồn 4 đầy hay bồn 3 cạn. Cuối cùng, sản phẩm dung dịch được đưa vào bồn chứa lưu trữ nhờ bơm 6. Đến đây, quá trình xử lý kết thúc một chu kỳ hoạt động, và quá trình xử lý có thể bắt đầu chu kỳ mới.





Hình 3.4: Lưu đồ chức năng xử lý hóa chất.

**IV. Thiết kế bằng chuỗi logic.**

Các ý tưởng thiết kế có thể được chuyển thành các phương trình của đại số Boolean. Các phương trình của đại số Boolean có thể được sắp xếp lại hay được rút gọn, sau đó được chuyển đổi thành ngôn ngữ ladder hay sơ đồ mạch điện.

Nếu chúng ta có thể diễn tả quá trình điều khiển hoạt động, thì chúng ta có thể biến đổi trực tiếp thành phương trình đại số Boolean.

**❖ Ví dụ:**

**Mô tả quá trình.**

Một lò đốt nóng có hai ngăn, dùng để đốt nóng một thỏi kim loại trong mỗi ngăn. Khi lò được bật, nó sẽ cung cấp nhiệt lượng cho hai ngăn. Nếu chỉ có một thỏi kim loại trong lò mà lò quá nóng thì hệ thống quạt sẽ hoạt động nhằm làm giảm nhiệt độ của lò.

**Mô tả quá trình điều khiển.**

Nếu nhiệt độ quá cao và chỉ có một thỏi kim loại trong lò thì hệ thống quạt được bật.

**Định nghĩa các ngõ vào và ngõ ra**

X1: thỏi kim loại ở ngăn 1

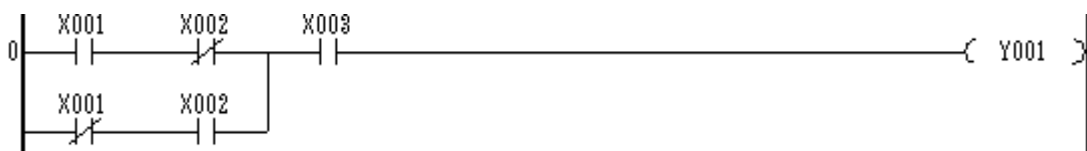
X2: thỏi kim loại ở ngăn 2

X3: cảm biến nhiệt độ

Y1: quạt

X3	X1	X2	Y1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$Y1 = X3\overline{X1}X2 + X3X1\overline{X2}$$



**❖ Thiết kế hệ thống chống trộm.**

Hệ thống sẽ hoạt động nếu có người xâm nhập bất hợp pháp. Hệ thống được kích hoạt nhờ các cảm biến phát hiện chuyển động và cảm biến ở hệ thống cửa sổ và cửa ra vào. Hệ thống cảm biến ở cửa sổ và cửa ra vào sẽ bị ngắt khi cửa sổ bị phá vỡ hay cửa ra vào bị mở. Cảm biến chuyển động sẽ tác động khi có vật thể dịch chuyển.

A: hệ thống báo động. (1 = on)

S: contact nguồn cho hệ thống báo động. (1 = on)

W: cảm biến cửa sổ và cửa ra vào. (1 = OK)

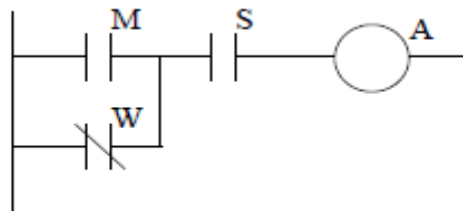
M: cảm biến phát hiện sự dịch chuyển. (0 = OK)

S	M	W	A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$$A = \overline{S}\overline{M}\overline{W} + \overline{S}M\overline{W} + S\overline{M}\overline{W} + SMW$$

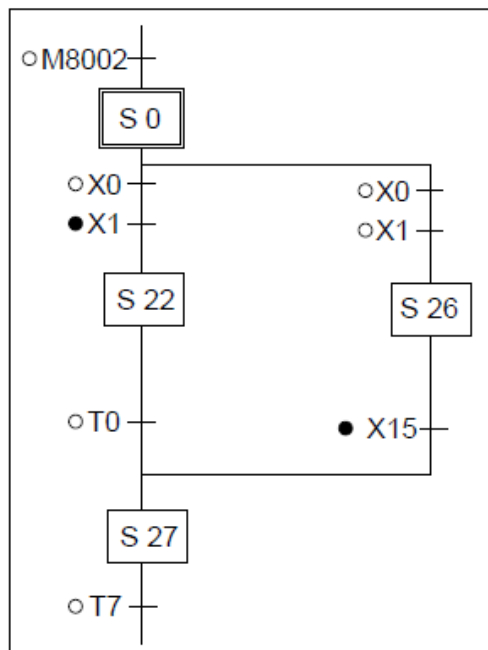
$$A = \overline{S}\overline{W}(\overline{M} + M) + SM(\overline{W} + W)$$

$$A = S(M + \overline{W})$$



**V. Điều khiển trình tự dùng Step Ladder (STL)**

Step ladder, gọi tắt là STL, là công cụ điều khiển trình tự rất hiệu quả. Kỹ thuật lập trình này tương tự với sự biểu diễn sơ đồ chức năng của hoạt động trình tự, thay thế phương pháp dùng tổ hợp logic truyền thống. STL thể hiện mặt mạnh của nó thông qua việc tổ chức một chương trình lớn thành những phần tử nhỏ hơn. Mỗi phần tử này gọi là một trạng thái hoặc một bước. Để nhận biết các trạng thái, mỗi trạng thái được gán một ký hiệu xác định duy nhất. Các ký hiệu này chính là các thiết bị relay trạng thái (state relay devices).



Mỗi trạng thái hoàn toàn tách biệt với các trạng thái khác trong chương trình. Cách tốt nhất để hình dung điều này là mỗi trạng thái xem như là một chương trình tách biệt và người sử dụng đặt từng phần chương trình này lại với nhau theo thứ tự hoạt động của nó. Ngoài ra các trạng thái này có thể sử dụng lại nhiều lần và theo thứ tự khác nhau. Điều này tiết kiệm được thời gian lập trình và hạn chế các lỗi lập trình.

Lập trình Step ladder có đặc điểm làm cho việc điều khiển trình tự được đơn giản như sau:

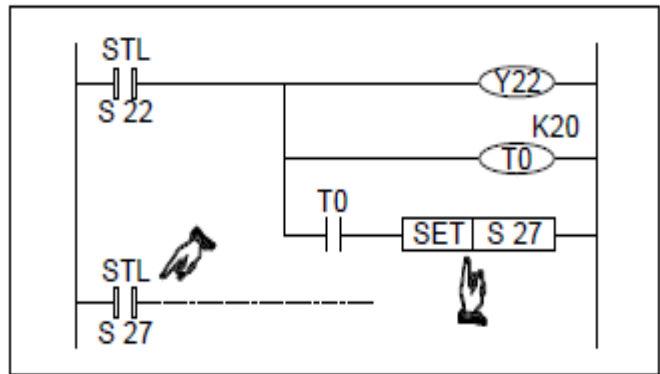
- Khả năng giữ được trạng thái hiện hành nhờ dùng cờ có khả năng chốt.
- Tự động vô hiệu (reset) trạng thái trước đó khi chuyển vào trạng thái hiện hành.
- Dễ dàng phân nhánh song song cả dạng OR và AND.

**VI. Lệnh STL và lập trình STL**

Chương trình STL không hoàn toàn thay thế chương trình Ladder chuẩn. Từng phần chương trình STL lớn hoặc nhỏ có thể được kích hoạt bất cứ lúc nào. Khi mỗi tác vụ STL hoàn tất, chương trình có thể trở về và tiếp tục xử lý chương trình ladder chuẩn cho đến khi chương trình STL kế tiếp cần thực thi. Do đó phần điều khiển khởi động và dừng một chương trình STL rất quan trọng.

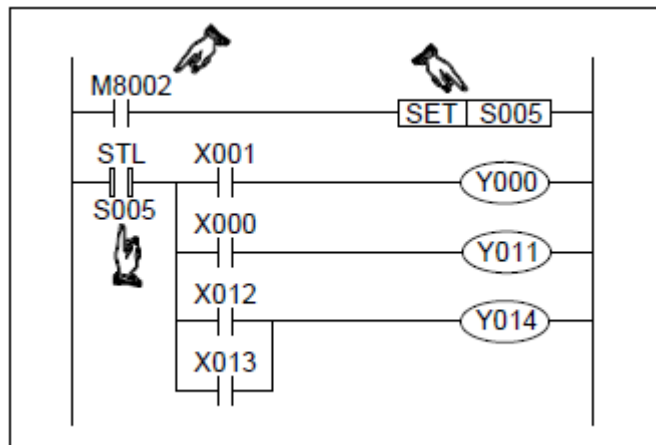
### Chương 3: KỸ THUẬT LẬP TRÌNH ĐIỀU KHIỂN TUẦN TỰ

Ở dạng mẫu chương trình STL của Mitsubishi, có thể dùng thanh ghi trạng thái (State registers) để khởi tạo một đoạn chương trình STL. Tuy nhiên, có 10 thanh ghi trạng thái chuyên dùng (S0 tới S9) được gán một chức năng khởi tạo đặc biệt. Vì vậy, khi khởi động một lưu trình STL nên được khởi tạo bằng một trong các relay trạng thái từ S0 tới S9.



Để bắt đầu một trình tự STL ta phải bật trạng thái khởi tạo lên ON thích hợp. Có nhiều phương pháp điều khiển trạng thái khởi tạo. Ví dụ, các cuộn dây khởi tạo STL có thể được kích bằng xung, dùng lệnh SET hoặc sử dụng với lệnh OUT. Tuy nhiên trong ngôn ngữ lập trình STL của Mitsubishi cuộn dây STL được kích hoạt bằng lệnh SET hoạt động khác khi dùng với lệnh OUT. Đối với tác vụ STL thông thường nên dùng lệnh SET để chọn các trạng thái. Khi một trạng thái được kích hoạt thì có nghĩa là nó được đặt lên ON.

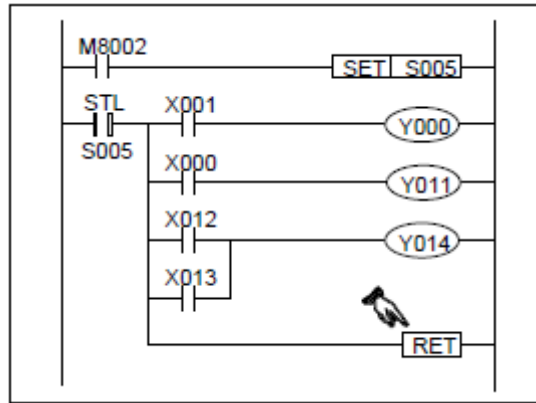
Đối với một chương trình STL được kích hoạt vừa khi cấp điện cho bộ điều khiển, có thể dùng mạch như hình bên, trong đó M8002 dùng để điều khiển việc thiết lập trạng thái khởi tạo.



Bây giờ chương trình STL đã được khởi tạo và CPU sẽ xử lý tất cả các lệnh bên trong chương trình STL. Có nghĩa là khi bắt đầu quét chương trình lần thứ hai thì

chương trình này dùng để khởi tạo chương trình STL được xét đến ở trong chương trình STL. Điều này hiển nhiên không đúng và CPU sẽ tiến xác định lỗi sai chương trình và vô hiệu hóa hoạt động của bộ lập trình điều khiển.

Lệnh RET (RETurn) là câu lệnh cuối cùng của trạng thái STL, khi đó lệnh này sẽ trả sự điều khiển về cho Ladder. Sau đó, mạch khởi tạo đã lập trình sẽ được xử lý đúng như phần chương trình Ladder và không còn nằm trong phạm vi hoạt động trạng thái STL cuối cùng.

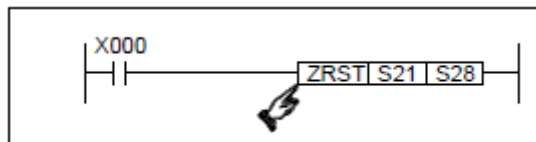


## **VII. Lập bước giữa các trạng thái STL**

Để kích hoạt một trạng thái STL, trước tiên người sử dụng phải tác động cuộn dây STL. Việc tác động lên cuộn dây đó giống như cách khởi động một chương trình STL, hay gọi là kích trạng thái khởi tạo. Lưu ý rằng dùng lệnh OUT (để kích cuộn dây STL) có sự khác biệt về cách hoạt động so với dùng lệnh SET. Sự khác nhau này được giải thích như sau:

### ***Dùng lệnh SET để kích cuộn dây STL***

- SET được dùng để kích cuộn dây STL làm cho nó hoạt động. Một khi trạng thái STL hiện hành kích hoạt tiếp trạng thái STL thứ hai, thì cuộn dây STL được reset. Do vậy, dù lệnh SET được dùng để kích hoạt một trạng thái còn việc đặt lại được tự động.
- Để reset một trạng thái STL, ta dùng lệnh RST hoặc ZRST (Zone Reset). Tuy nhiên lệnh ZRST chỉ reset được trạng thái STL sau khi trạng thái STL đã được thực hiện xong hết một chu kỳ của nó. Đây cũng chính là nhược điểm của lệnh ZRST.



### ***Dùng lệnh OUT để kích cuộn dây STL***

- Lệnh OUT có tính năng giống như lệnh SET. Tuy nhiên, lệnh OUT khác lệnh SET là lệnh OUT có thể cho phép người lập trình được nhảy cách khoảng, bỏ qua nhiều trạng thái STL.
- Nếu người sử dụng muốn ‘nhảy’ (jump) ngược lên một chương trình, nghĩa là chạy ngược về trạng thái đã được xử lý thì lệnh OUT sẽ được dùng với ký hiệu trạng thái STL tương ứng. Ngoài ra người sử dụng có thể tạo ra bước nhảy lớn bỏ qua nhiều trạng thái STL.

