

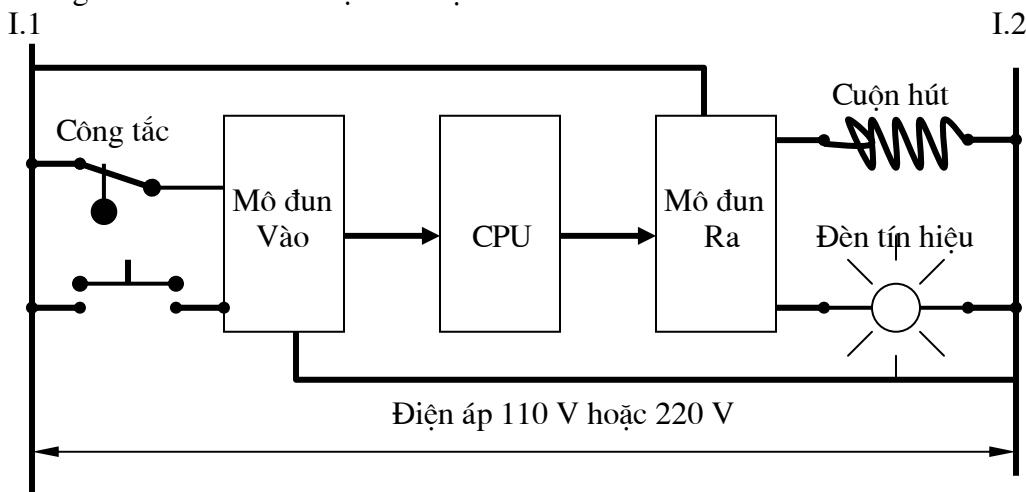
## CHƯƠNG 1 GIỚI THIỆU VỀ PLC

### I.1 MỞ ĐẦU

Trong các hệ thống sản xuất, trong các thiết bị tự động và bán tự động, hệ thống điều khiển đóng vai trò điều phối toàn bộ các hoạt động của máy móc thiết bị. Các hệ thống máy móc và thiết bị sản xuất thường rất phức tạp, có rất nhiều đại lượng vật lý phải điều khiển để có thể hoạt động đồng bộ hoặc theo một trình tự công nghệ nhất định nhằm tạo ra một sản phẩm mong muốn. Từng đại lượng vật lý đơn lẻ có thể được điều khiển bằng một mạch điều khiển cơ sở dạng tương tự hay gián đoạn. Điều khiển nhiều đại lượng vật lý đồng thời chúng ta không thể dùng các mạch điều khiển tương tự mà phải sử dụng hệ thống điều khiển lô gíc. Trước đây các hệ thống điều khiển lô gíc được sử dụng là hệ thống lô gíc rơ le. Nhờ sự phát triển nhanh chóng của kỹ thuật điện tử, các thiết bị điều khiển lô gíc khả lập trình PLC (Programmable Logic Controller) đã xuất hiện vào năm 1969 thay thế các hệ thống điều khiển rơ le. Càng ngày PLC càng trở nên hoàn thiện và đa năng. Các PLC ngày nay không những có khả năng thay thế hoàn toàn các thiết bị điều khiển lo gíc cổ điển, mà còn có khả năng thay thế các thiết bị điều khiển tương tự. Các PLC được sử dụng rộng rãi trong công nghiệp.

Chức năng chính của PLC là kiểm tra trạng thái của các đầu vào và điều khiển các quá trình hoặc các hệ thống máy móc thông qua các tín hiệu trên chính đầu ra của PLC. Tổ hợp lô gíc của các đầu vào để tạo ra một hay nhiều tín hiệu ra được gọi là điều khiển lô gíc. Các tổ hợp lô gíc thường được thực hiện theo trình tự điều khiển hay còn gọi là chương trình điều khiển. Chương trình điều khiển được lưu trong bộ nhớ của PLC có thể bằng cách lập trình bằng thiết bị cầm tay nối trực tiếp với PLC hoặc lập trình trên máy tính cá nhân nhờ các phần mềm chuyên dụng và truyền vào PLC qua mạng hay qua cáp truyền dữ liệu. Bộ xử lý tín hiệu, thường là các bộ vi xử lý tốc độ cao, thực hiện chương trình điều khiển theo chu kỳ. Khoảng thời gian thực hiện một chu trình điều khiển từ lúc kiểm tra các tín hiệu vào, thực hiện các phép tính logic hoặc đại số để có được tín hiệu điều khiển, cho đến khi phát tín hiệu đến đầu ra được gọi là chu kỳ thời gian quét.

PLC trong công nghiệp thường có cấu hình đơn giản nhất, bởi vì các chương trình trình điều khiển quá trình công nghệ hay máy móc thường được hoạt động 24/24 và không cần bất cứ sự can thiệp của con người trong quá trình điều khiển. PLC chỉ dừng quét chương trình điều khiển khi ngắt nguồn hoặc khi công tắc ngừng được kích hoạt. Sơ đồ khối đơn giản hóa của PLC được thể hiện trên hình 1.1.



Hình 1.1 Sơ đồ khối của một bộ PLC đơn giản.

Trên đầu vào của PLC có thể có các kênh tín hiệu tương tự hoặc các kênh tín hiệu số. Các kênh tín hiệu này xuất phát từ các cảm biến, từ các công tắc hành trình, công tắc đóng ngắt mạch điện hoặc từ các biến lô gíc tương ứng với các trạng thái của máy móc, thiết bị. Tín hiệu vào được bộ xử lý trung tâm xử lý nhờ các phép tính lô gíc hay số học và kết quả là các tín hiệu ra. Các tín hiệu tín hiệu ra là các tín hiệu truyền điện năng đến cho các cơ cấu chấp hành như cuộn hút, đèn hiệu, động cơ vv.

Điện áp trên đầu vào của PLC là điện áp công suất thấp, tương ứng với mức từ 0V đến 5V một chiều. Khi ta nối các đầu vào có mức điện áp cao hơn 5V, thường phải dùng các kênh có các mạch chuyển đổi để biến điện áp vào thành điện áp tương đương với mức +/- 5VDC. Điện áp trên đầu ra của PLC có thể có nhiều mức điện áp khác nhau, nhưng đều có mức năng lượng thấp. Nếu cần phải điều khiển cơ cấu chấp hành có mức năng lượng cao hơn, ta phải sử dụng các thiết bị khuyếch đại công suất.

## I.2 LỊCH SỬ PHÁT TRIỂN CỦA PLC

Vào khoảng năm 1968, các nhà sản xuất ô tô đã đưa ra các yêu cầu kỹ thuật đầu tiên cho thiết bị điều khiển lô gíc khả lập trình. Mục đích đầu tiên là thay thế cho các tủ điều khiển công kênh, tiêu thụ nhiều điện năng và thường xuyên phải thay thế các rơ le do hỏng cuộn hút hay gây các thanh lò xo tiếp điểm. Mục đích thứ hai là tạo ra một thiết bị điều khiển có tính linh hoạt trong việc thay đổi chương trình điều khiển. Các yêu cầu kỹ thuật này chính là cơ sở của các máy tính công nghiệp, mà ưu điểm chính của nó là sự lập trình dễ dàng bởi các kỹ thuật viên và các kỹ sư sản xuất. Với thiết bị điều khiển khả lập trình, người ta có thể giảm thời gian dừng trong sản xuất, mở rộng khả năng hoàn thiện hệ thống sản xuất và thích ứng với sự thay đổi trong sản xuất. Một số nhà sản xuất thiết bị điều khiển trên cơ sở máy tính đã sản xuất ra các thiết bị điều khiển khả lập trình còn gọi là PLC.

Những PLC đầu tiên được ứng dụng trong công nghiệp ô tô vào năm 1969 đã đem lại sự ưu việt hơn hẳn các hệ thống điều khiển trên cơ sở rơ le. Các thiết bị này được lập trình dễ dàng, không chiếm nhiều không gian trong các xưởng sản xuất và có độ tin cậy cao hơn các hệ thống rơ le. Các ứng dụng của PLC đã nhanh chóng rộng mở ra tất cả các ngành công nghiệp sản xuất khác.

Hai đặc điểm chính dẫn đến sự thành công của PLC đó chính là độ tin cậy cao và khả năng lập trình dễ dàng. Độ tin cậy của PLC được đảm bảo bởi các mạch bán dẫn được thiết kế thích ứng với môi trường công nghiệp. Các mạch vào ra được thiết kế đảm bảo khả năng chống nhiễu, chịu được ẩm, chịu được dầu, bụi và nhiệt độ cao. Các ngôn ngữ lập trình đầu tiên của PLC tương tự như sơ đồ thang trong các hệ thống điều khiển lô gíc, nên các kỹ sư đã làm quen với sơ đồ thang, dễ dàng thích nghi với việc lập trình mà không cần phải qua một quá trình đào tạo nào. Một số các ứng dụng của máy tính trong sản xuất trong thời gian đầu bị thất bại, cũng chính vì việc học sử dụng các phần mềm máy tính không dễ dàng ngay cả với các kỹ sư.

Khi các vi xử lý được đưa vào sử dụng trong những năm 1974 – 1975, các khả năng cơ bản của PLC được mở rộng và hoàn thiện hơn. Các PLC có trang bị vi xử lý có khả năng thực hiện các tính toán và xử lý số liệu phức tạp, điều này làm tăng khả năng ứng dụng của PLC cho các hệ thống điều khiển phức tạp. Các PLC không chỉ dừng lại ở chỗ là các thiết bị điều khiển lô gíc, mà nó còn có khả năng thay thế cả các thiết bị điều khiển tương tự. Vào cuối những năm bảy mươi việc truyền dữ liệu đã trở nên dễ dàng nhờ sự phát triển nhảy vọt của công nghiệp điện tử. Các PLC có thể điều khiển các thiết bị

cách xa hàng vài trăm mét. Các PLC có thể trao đổi dữ liệu cho nhau và việc điều khiển qua trình sản xuất trở nên dễ dàng hơn.

Thiết bị điều khiển khả lập trình PLC chính là các máy tính công nghiệp dùng cho mục đích điều khiển máy, điều khiển các ứng dụng công nghiệp thay thế cho các thiết bị “cứng” như các rơ le, cuộn hút và các tiếp điểm.

Ngày nay chúng ta có thể thấy PLC trong hàng nghìn ứng dụng công nghiệp. Chúng được sử dụng trong công nghiệp hóa chất, công nghiệp chế biến dầu, công nghiệp thực phẩm, công nghiệp cơ khí, công nghiệp xử lý nước và chất thải, công nghiệp dược phẩm, công nghiệp dệt may, nhà máy điện hạt nhân, trong công nghiệp khai khoáng, trong giao thông vận tải, trong quân sự, trong các hệ thống đảm bảo an toàn, trong các hệ thống vận chuyển tự động, điều khiển rô bốt, điều khiển máy công cụ CNC vv. Các PLC có thể được kết nối với các máy tính để truyền, thu thập và lưu trữ số liệu bao gồm cả quá trình điều khiển bằng thống kê, quá trình đảm bảo chất lượng, chẩn đoán sự cố trực tuyến, thay đổi chương trình điều khiển từ xa. Ngoài ra PLC còn được dùng trong hệ thống quản lý năng lượng nhằm giảm giá thành và cải thiện môi trường điều khiển trong các các hệ thống phục vụ sản xuất, trong các dịch vụ và các văn phòng công sở.

Sự ra đời của máy tính cá nhân PC trong những năm tám mươi đã nâng cao đáng kể tính năng và khả năng sử dụng của PLC trong điều khiển máy và quá trình sản xuất. Các PC giá thành không cao có thể sử dụng như các thiết bị lập trình và là giao diện giữa người vận hành và hệ thống điều khiển. Nhờ sự phát triển của các phần mềm đồ họa cho máy tính cá nhân PC, các PLC cũng được trang bị các giao diện đồ họa để có thể mô phỏng hoặc hiện thị các hoạt động của từng bộ phận trong hệ thống điều khiển. Điều này có ý nghĩa đặc biệt quan trọng đối với các máy CNC, vì nó tạo cho ta khả năng mô phỏng trước quá trình gia công, nhằm tránh các sự cố do lập trình sai. Máy tính cá nhân PC và PLC đều được sử dụng rộng rãi trong các hệ thống điều khiển sản xuất và cả trong các hệ thống dịch vụ.

PLC được sản xuất bởi nhiều hãng khác nhau trên thế giới. Về nguyên lý hoạt động, các PLC này có tính năng tương tự giống nhau, nhưng về lập trình sử dụng thì chúng hoàn toàn khác nhau do thiết kế khác nhau của mỗi nhà sản xuất. PLC khác với các máy tính là không có ngôn ngữ lập trình chung và không có hệ điều hành. Khi được bắt lên thì PLC chỉ chạy chương trình điều khiển ghi trong bộ nhớ của nó, chứ không thể chạy được hoạt động nào khác. Một số hãng sản xuất PLC lớn có tên tuổi như: Siemens, Toshiba, Mitsubishi, Omron, Allan Bradley, Rockwell, Fanuc là các hãng chiếm phần lớn thị phần PLC thế giới. Các PLC của các hãng này được ứng dụng rộng rãi trong công nghiệp sử dụng công nghệ tự động hóa.

Các thiết bị điều khiển PLC tạo thêm sức mạnh, tốc độ và tính linh hoạt cho các hệ thống công nghiệp. Bằng sự thay thế các phân tử cơ điện bằng PLC, quá trình điều khiển trở nên nhanh hơn, rẻ hơn, và quan trọng nhất là hiệu quả hơn. PLC là sự lựa chọn tốt hơn các hệ thống rơ le hay máy tính tiêu chuẩn do một số lý do sau:

- **Tốn ít không gian:** Một PLC cần ít không gian hơn một máy tính tiêu chuẩn hay tủ điều khiển rơ le để thực hiện cùng một chức năng.
- **Tiết kiệm năng lượng:** PLC tiêu thụ năng lượng ở mức rất thấp, ít hơn cả các máy tính thông thường.
- **Giá thành thấp :** Một PLC giá tương đương cỡ 5 đến 10 rơ le, nhưng nó có khả năng thay thế hàng trăm rơ le.
- **Khả năng thích ứng với môi trường công nghiệp:** Các vỏ của PLC được làm từ các vật liệu cứng, có khả năng chống chịu được bụi bẩn, dầu mỡ, độ ẩm, rung động và nhiễu. Các máy tính tiêu chuẩn không có khả năng này.

- **Giao diện trực tiếp:** Các máy tính tiêu chuẩn cần có một hệ thống phức tạp để có thể giao tiếp với môi trường công nghiệp. Trong khi đó các PLC có thể giao diện trực tiếp nhờ các mô đun vào ra I/O.
- **Lập trình dễ dàng:** Phần lớn các PLC sử dụng ngôn ngữ lập trình là sơ đồ thang, tương tự như sơ đồ đấu của các hệ thống điều khiển rơ le thông thường.
- **Tính linh hoạt cao:** Chương trình điều khiển của PLC có thể thay đổi nhanh chóng và dễ dàng bằng cách nạp lại chương trình điều khiển mới vào PLC bằng bộ lập trình, bằng thẻ nhớ, bằng truyền tải qua mạng.

### I.3. PHÂN LOẠI PLC

Căn cứ vào số lượng các đầu vào/ ra, ta có thể phân PLC thành bốn loại sau:

- micro PLC là loại có dưới 32 kênh vào/ ra
- PLC nhỏ có đến 256 kênh vào/ ra
- PLC trung bình có đến 1024 kênh vào/ ra
- PLC cỡ lớn có trên 1024 kênh vào/ra.

**Các micro – PLC** thường có ít hơn 32 đầu vào/ra. Trên hình 1.2 là ví dụ về micro PLC họ T100MD-1616 do hãng Triangle Research International sản xuất. Cấu tạo tương đối đơn giản và toàn bộ các bộ phận được tích hợp trên một bảng mạch có kích thước nhỏ gọn. Micro – PLC có cấu tạo gồm tất cả các bộ phận như bộ xử lý tín hiệu, bộ nguồn, các kênh vào/ra trong một khối. Các micro – PLC có ưu điểm hơn các PLC nhỏ là giá thành rẻ, dễ lắp đặt.



Hình 1.2 Micro PLC họ T100MD-1616

Một loại micro PLC khác là DL05 của hãng Koyo, loại này có 30 kênh vào/ ra



Hình 1.3. Micro PLC họ DL05 của hãng Koyo

Một loại micro-PLC khác là loại xê ri 90 của Fanuc, hình 1.4. Loại này có 8 kênh vào và 8 kênh ra.



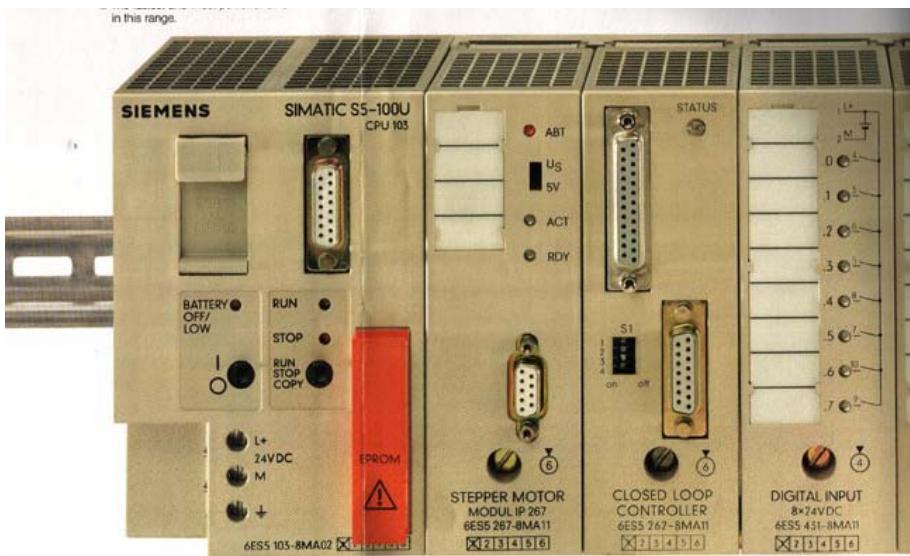
Hình 1.4. Micro-PLC xê ri 90 của Fanuc

**PLC loại nhỏ** có thể có đến 256 đầu vào/ra. Trên hình 1.5 là PLC của hãng OMRON loại ZEN – 10C. Loại PLC này có 34 kênh vào/ ra gồm: 6 kênh vào và 4 kênh ra trên mô đun CPU, còn lại 3 mô đun vào/ ra, với 4 kênh vào và 4 kênh ra cho mỗi mô đun.



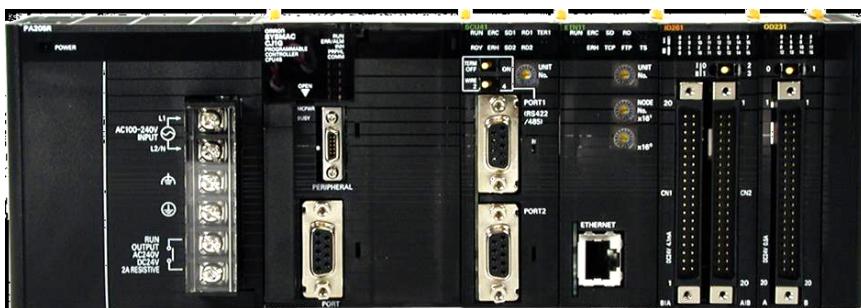
Hình 1.5. PLC loại ZEN-10C của Omron

Hãng Siemens có các PLC loại nhỏ như S5-90U, S5-95U, S5-100U (hình 1.6), S7 – 200 là các loại PLC loại nhỏ, có số lượng kênh vào/ ra nhỏ hơn 256. Cấu tạo của các PLC loại nhỏ cũng tương tự như cấu tạo của các PLC loại trung bình, vì đều là dạng mô đun. Điểm khác biệt là dung lượng bộ nhớ, số lượng kênh vào/ ra của các mô đun khác nhau về độ lớn và tốc độ xử lý thông tin cũng khác nhau. PLC của Siemens được dùng rộng rãi ở trong hầu hết các nước có nền công nghiệp phát triển.



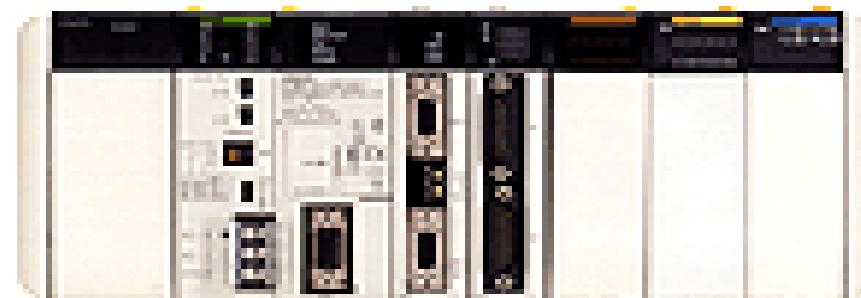
Hình 1.7. PLC S5-100U của Siemens

**Các PLC trung bình** có thể có đến 1024 đầu vào/ra. Loại CJ1M của Omron trên hình 1.8 có 320 kênh vào/ ra.



Hình 1.8. PLC loại CJ1M của Omron

Loại PLC CQM1 hay CQMIH của Omron trên hình 1.9 có 512 kênh vào/ra.



Hình 1.9. PLC loại CQM1 của Omron

Hãng Siemens có một số xê ri S7-200 là các loại PLC hạng trung bình. Số lượng kênh vào/ra của S-300 có thể trong khoảng từ 256 đến 1024.

Các PLC loại lớn có nhiều hơn 1024 đầu vào/ra. Loại này có tốc độ xử lý rất cao, dung lượng bộ nhớ lớn và thường được dùng trong điều khiển các hệ thống thiết bị công nghệ phức tạp. Hãng Omron có PLC loại CJ1 trên hình 1.10, là loại có tới 1280 kênh vào/ ra và loại CJ1H có tới 2560 kênh vào/ra.



Hình 1.10. PLC loại CJ1 của Omron

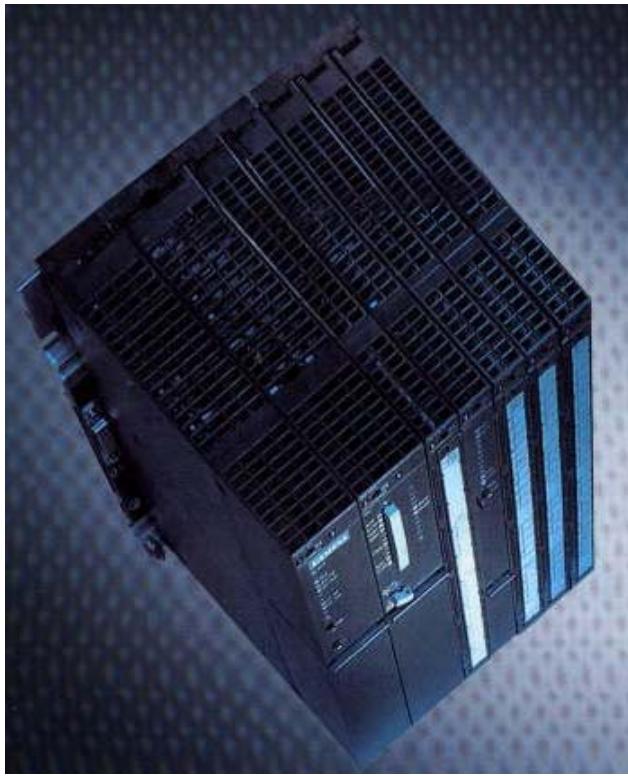
Hãng Omron còn có loại CS1 trên hình 1.11, là loại PLC cỡ lớn với 5120 kênh vào/ ra.



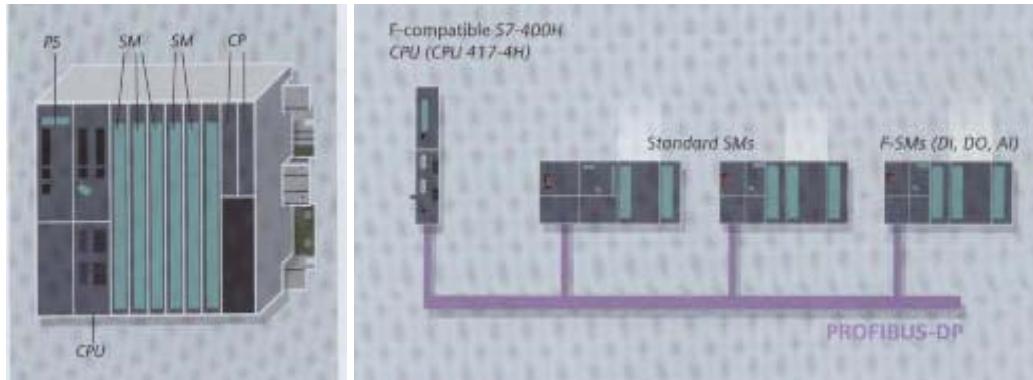
Hình 1.11. PLC loại CS1 của Omron

Các PLC loại lớn của Siemens là các loại xê ri S7-300, S7-400. Các loại này có số lượng kênh vào/ ra rất lớn. Các kênh này không thể đấu trực tiếp lên PLC mà phải thông qua các bộ dồn kênh và tách kênh ( demultiplexeur và multiplexeur). Trên hình 1.12 là PLC S7-400 của Siemens. Đây là loại PLC mạnh nhất của Siemens hiện nay. Cấu hình của PLC này được biểu diễn bằng hình 1.13.a, 1.13.b.

Các PLC trung bình và lớn có các mô đun vào/ra có thể lắp ráp với nhau trên cùng một giá đỡ tiêu chuẩn, cho phép lắp thêm hoặc tháo bớt ra mà không cần tắt nguồn. Các PLC được kết nối với nhau thông qua mạng ETHERNET công nghiệp (hình 1.14).



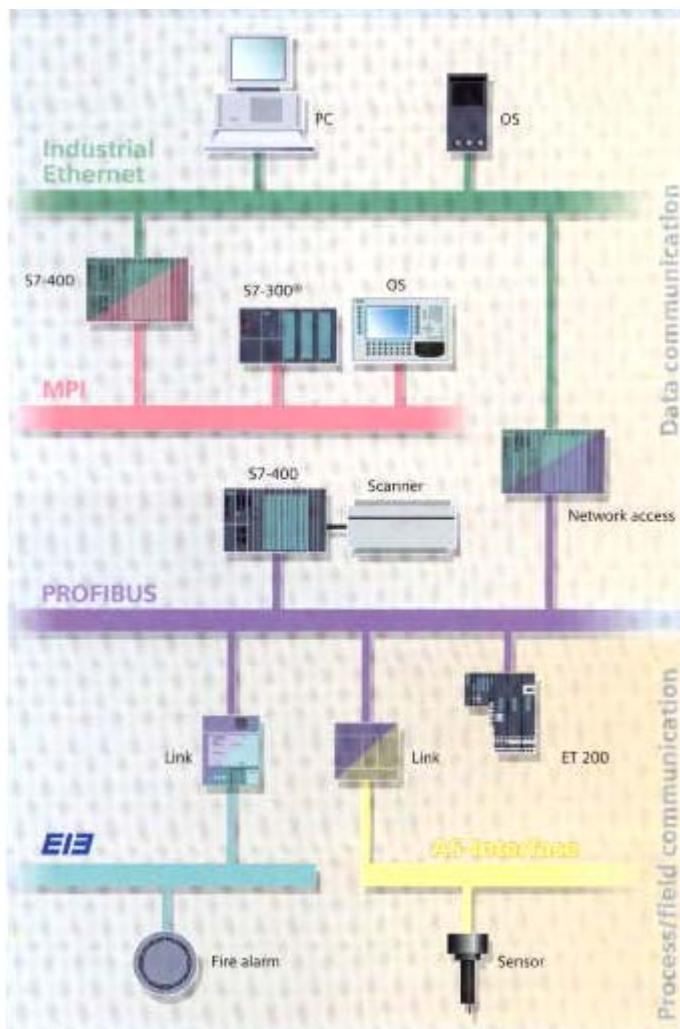
Hình 1.12. PLC S7-400 của Siemens



Hình 1.13. a, Cấu trúc của S7-400; b, Sơ đồ kết nối của S-400

Các PLC loại lớn thường dùng để điều khiển ở mức cao. Ở mức thấp thường là các thiết bị điều khiển tương tự, hay thiết bị điều khiển số với các PLC loại nhỏ, hay loại trung bình. Ở mức thấp, chủ yếu là các thiết bị điều khiển trực tiếp các thiết bị công nghệ, các cơ cấu chấp hành, các động cơ, bơm, van, cuộn hút, đèn hiệu vv. Điều khiển ở mức cao bao gồm các điều khiển liên quan đến phần quản lý hệ thống và quản lý dữ liệu của hệ thống điều khiển. Ở mức này, các dữ liệu có thể được thu thập từ các thiết bị điều khiển mức thấp hoặc từ bên ngoài hệ thống thông qua mạng nội bộ và mạng Internet. Các dữ liệu từ các PLC được truyền về các máy tính trung tâm để lưu trữ và xử lý. Trường hợp các hệ thống sản xuất tự động có điều khiển bằng thống kê, đây chính là điều khiển ở mức cao, tương ứng với cấu trúc quản lý của hệ thống. Hoạt động của hệ thống điều khiển

được điều chỉnh dựa theo kết quả phân tích, đánh giá từ các dữ liệu thống kê, như vậy giúp cho việc sản xuất luôn ở dạng tối ưu nhất và hiệu quả nhất. PLC S7-400 của Siemens là một trong những loại PLC lớn và rất mạnh trong các hệ thống điều khiển sản xuất quy mô như các nhà máy công nghiệp. Loại PLC này có thể kết nối trực tiếp qua mạng Ethernet công nghiệp với các thiết bị điều khiển mức cao hơn để trao đổi dữ liệu hoặc thông qua các kênh giao diện khác như MPI, PROFIBUS, EIB hay giao diện AS để thu thập dữ liệu và điều khiển như hình 1.14.



Hình 1.14. Sơ đồ kết nối mạng của S7-400 trong công nghiệp

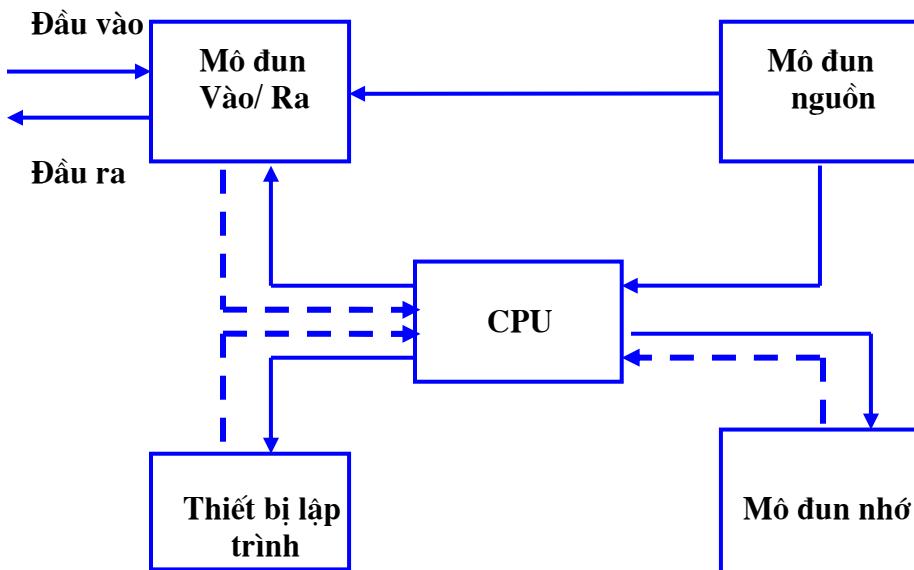
#### I.4. THÀNH PHẦN CƠ BẢN CỦA PLC

Nếu không nhìn về khía cạnh giá thành, kích thước, mức độ phức tạp, tất cả các PLC đều có những thành phần cơ bản và đặc điểm chức năng giống nhau. Một PLC bao giờ cũng gồm có 6 thành phần cơ bản:

- Mô đun xử lý tín hiệu
- Mô đun vào
- Mô đun ra
- Mô đun nhớ
- Mô đun nguồn

- Thiết bị lập trình

Sơ đồ của một bộ PLC cơ bản được biểu diễn trên hình 1.15. Ngoài các mô đun chính này, các PLC còn có các mô đun phụ trợ như mô đun kết nối mạng, các mô đun đặc biệt để xử lý tín hiệu như mô đun kết nối với các can nhiệt, mô đun điều khiển động cơ bước, mô đun kết nối với encoder, mô đun đếm xung vào vv..



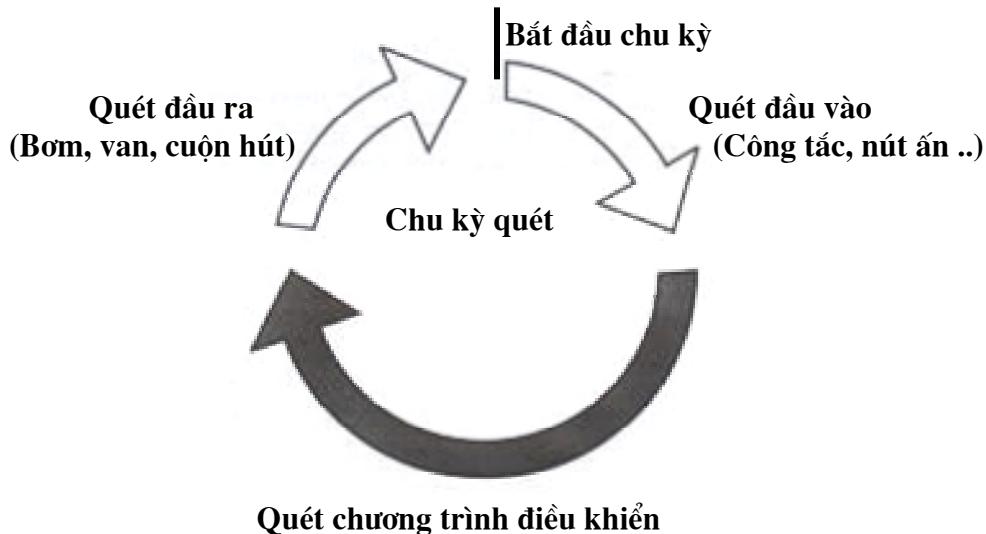
Hình 1.15. Cấu trúc cơ bản của PLC

### BỘ XỬ LÝ TÍN HIỆU

Đây là bộ phận xử lý tín hiệu trung tâm hay CPU của PLC. Bộ xử lý tín hiệu có thể bao gồm một hay nhiều bộ vi xử lý tiêu chuẩn hoặc các bộ vi xử lý hỗ trợ cùng với các mạch tích hợp khác để thực hiện các phép tính lô gíc, điều khiển và ghi nhớ các chức năng của PLC. Bộ xử lý thu thập các tín hiệu vào, thực hiện các phép tính lô gíc theo chương trình, các phép tính đại số và điều khiển các đầu ra số hay tương ứng. Phần lớn các PLC sử dụng các mạch logic chuyên dụng trên cơ sở bộ vi xử lý và các mạch tích hợp tạo nên đơn vị xử lý trung tâm CPU.

Bộ vi xử lý sẽ lần lượt quét các trạng thái của đầu vào và các thiết bị phụ trợ, thực hiện logic điều khiển được đặt ra bởi chương trình ứng dụng, thực hiện các tính toán và điều khiển các đầu ra tương ứng của PLC. Bộ vi xử lý nâng cao khả năng logic và khả năng điều khiển của PLC. Các PLC thế hệ cuối cho phép thực hiện các phép tính số học và các phép tính logic, bộ nhớ lớn hơn, tốc độ xử lý cao hơn và có trang bị giao diện với máy tính, với mạng nội bộ vv.

Bộ vi xử lý điều khiển chu kỳ làm việc của chương trình. Chu kỳ này được gọi là chu kỳ quét của PLC, tức là khoảng thời gian thực hiện xong một vòng các lệnh của chương trình điều khiển. Chu kỳ quét được minh họa trên hình 1.16.



Hình 1.16. Chu kỳ quét của PLC

Khi thực hiện quét các đầu vào, PLC kiểm tra tín hiệu từ các thiết bị vào như các công tắc, cảm biến. Trạng thái của các tín hiệu vào được lưu tạm thời vào bảng ảnh đầu vào hoặc vào một mảng nhớ. Trong thời gian quét chương trình, bộ xử lý quét lần lượt các lệnh của chương trình điều khiển, sử dụng các trạng thái của tín hiệu vào trong mảng nhớ để xác định các đầu ra sẽ được nạp năng lượng hay không. Kết quả là các trạng thái của đầu ra được ghi vào mảng nhớ. Từ dữ liệu của mảng nhớ tín hiệu ra, PLC sẽ cấp hoặc ngắt điện năng cho các mạch ra để điều khiển các thiết bị ngoại vi. Chu kỳ quét của PLC có thể kéo dài từ 1 đến 25 милли giây. Thời gian quét đầu vào và đầu ra thường rất ngắn so với chu kỳ quét của PLC.

### **BỘ NHỚ**

Bộ nhớ của PLC có vai trò rất quan trọng, bởi vì nó được sử dụng để chứa toàn bộ chương trình điều khiển, các trạng thái của các thiết bị phụ trợ. Thông thường các bộ nhớ được bố trí trong cùng một khối với CPU. Thông tin chứa trong bộ nhớ sẽ xác định việc các đầu vào, đầu ra được xử lý như thế nào. Bộ nhớ bao gồm các tế bào nhớ được gọi là bit. Mỗi bit có hai trạng thái 0 hoặc 1. Đơn vị thông dụng của bộ nhớ là K,  $1K = 1024$  từ (word), 1 từ (word) có thể là 8 bit. Các PLC thương có bộ nhớ từ 1K đến 64K, phụ thuộc vào mức độ phức tạp của chương trình điều khiển. Trong các PLC hiện đại có sử dụng một số kiểu bộ nhớ khác nhau. Các kiểu bộ nhớ này có thể xếp vào hai nhóm: bộ nhớ có thể thay đổi và bộ nhớ cố định. Bộ nhớ thay đổi là các bộ nhớ có thể mất các thông tin ghi trên đó khi mất điện. Nếu chương trình điều khiển chứa trong bộ nhớ mà bị mất điện đột xuất do tuột dây, mất điện nguồn thì chương trình phải được nạp lại và lưu vào bộ nhớ. Bộ nhớ cố định ngược lại với bộ nhớ thay đổi là có khả năng lưu giữ thông tin ngay cả khi mất điện. Các loại bộ nhớ hay sử dụng trong PLC gồm :

- a. ROM (*Read Only Memory*)
- b. RAM (*Random Access Memory*)
- c. PROM (*Programmable Read Only Memory*)
- d. EPROM (*Erasable Programmable Read Only Memory*)
- e. EEPROM (*Electrically Alterable Programmable Read Only Memory*)
- f. Bộ nhớ flash

Bộ nhớ ROM dùng để nhớ các lệnh điều khiển cơ bản của PLC, không thay đổi nội dung nhớ ngay cả khi mất điện.

Trong số này chỉ có bộ nhớ RAM là bộ nhớ thay đổi, các bộ nhớ khác lưu thông tin trong bộ nhớ khi mất điện. Bộ nhớ RAM thường hoạt động nhanh và dễ dàng nạp chương trình điều khiển ứng dụng cũng như các dữ liệu. Một số bộ nhớ RAM sử dụng pin để lưu nội dung nhớ khi mất điện. Bộ nhớ RAM được sản xuất từ công nghệ CMOS nên tiêu thụ rất ít năng lượng. Các PLC có thể được mở rộng thêm nên bộ nhớ cũng phải tăng thêm. Chương trình điều khiển đơn giản chỉ cần dung lượng bộ nhớ bé, ngược lại các chương trình phức tạp cần bộ nhớ dung lượng lớn.

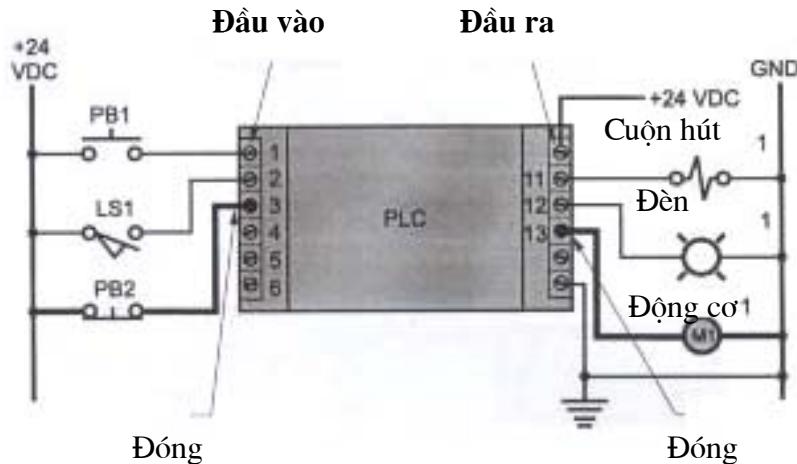
Bộ nhớ động được sử dụng rộng rãi đó là bộ nhớ RAM (*Random Acces Memory*). Bộ nhớ RAM hoạt động nhanh và là tạo ra và lưu các chương trình ứng dụng. Để chống lại khả năng mất dữ liệu khi mất điện, các PLC thường sử dụng pin.

Bộ nhớ tĩnh ROM (*Read Only Memory*) là bộ nhớ không bị thay đổi khi dữ liệu nhớ khi tắt nguồn hoặc mất điện. Bộ nhớ ROM dùng để nhớ các lệnh cơ bản và các hàm toán học của PLC. EEPROM (*Ellectrically Erasable Programmable Read Only Memory*) là bộ nhớ tĩnh có khả năng xoá bằng lập trình lại. EEPROM dùng để ghi chương trình ứng dụng.

Người sử dụng có thể truy cập vào hai vùng nhớ của PLC là vùng nhớ chương trình và vùng nhớ dữ liệu. Vùng nhớ chương trình là nơi chứa chương trình điều khiển ứng dụng, các chương trình con và các lõi của chương trình. Vùng nhớ dữ liệu lưu trữ các dữ liệu liên quan đến chương trình điều khiển như dữ liệu vào/ra; giá trị đầu, giá trị tức thời và giá trị cuối của bộ đếm lệnh hay bộ đếm thời gian; các hằng số và các biến của chương trình điều khiển. Hai vùng nhớ này được gọi là bộ nhớ dành cho người sử dụng. Bộ xử lý tín hiệu còn có bộ nhớ hệ thống dùng để ghi các dữ liệu trung gian trong quá trình thực hiện các phép tính, các lệnh của chương trình và phối hợp giữa chúng; quét các dữ liệu vào và gửi cá dữ liệu ra mới đến mô đun ra. Bộ nhớ hệ thống do nhà sản xuất lập trình từ khi xuất xưởng nên không thể thay đổi được và người sử dụng cũng không thể truy cập được.

## **MÔ ĐUN VÀO/RA**

Hệ thống các mô đun vào/ ra có khả năng kết nối giữa các thiết bị công nghệ với bộ vi xử lý. Hệ thống này dùng các mạch vào khác nhau để ghi nhận hoặc đo lường các đại lượng vật lý của quá trình công nghệ như chuyển động, cao độ, nhiệt độ, áp suất, lưu lượng, vị trí, tốc độ vv. Trên cơ sở các dữ liệu thu được, bộ xử lý tín hiệu tiến hành các phép tính lô gíc hay số học để xác định giá trị mới của tín hiệu ra. Các mô đun ra được nối để điều khiển các van, động cơ, bơm và báo động khi thực hiện quá trình điều khiển máy hoặc điều khiển hệ thống sản xuất. Trên hình 1.17 là sơ đồ kết nối của một bộ micro PLC với các thiết bị của môi trường làm việc. Điện áp 24 VDC không chạy từ bên trái qua bên phải sơ đồ thang như các mạch rơ le “cứng”. Điện áp ở đây chỉ đóng vai trò thể hiện các biến lô gíc đầu vào. Mạch lô gíc của PLC sẽ đảm bảo tính liên tục của lô gíc cho đến đầu ra. Nguồn trên mạch ra được cấp đến các thiết bị bên ngoài nếu lô gíc của các kênh ra được đảm bảo thông suốt từ bên trái qua bên phải của từng bậc trong sơ đồ thang.



Hình 1.17. Sơ đồ kết nối của PLC với các thiết bị vào/ ra

Trường hợp micro – PLC không có mô đun nguồn riêng biệt, thì nguồn điện được lắp trực tiếp trên CPU. Trên hình 1.18 là ví dụ về sơ đồ đấu dây trên micro – 1000 PLC của hãng Allan Bradley.

Nguồn điện áp vào cũng chính là nguồn đi đến các thiết bị đầu ra như bộ khởi động của bơm, bộ công tắc nhiệt. Nguồn điện +24VDC là nguồn lấy ra từ đầu ra của bộ nguồn, lại cấp cho các thiết bị đầu vào như: Công tắc cao độ, công tắc áp lực, công tắc hành trình, công tắc phụ bên ngoài của bơm. Nguồn vào xoay chiều 110VAC được đấu đến các cầu có ký hiệu VAC tương ứng với các kênh ra: kênh đến bộ khởi động bơm, kênh ra công tắc nhiệt.

Đầu vào và đầu ra của PLC thường được gộp vào các mô đun. Các mô đun vào/ra có thể tiếp nhận tín hiệu từ các thiết bị bên ngoài như công tắc, cảm biến quang, công tắc tiềm cận. Các tín hiệu được chuyển đổi từ điện áp 110VAC, 220VAC, +24DC thành tín hiệu  $\pm 5$  VDC. Bộ vi xử lý sẽ lấy tín hiệu này để xác định tín hiệu ra tương ứng. Điện áp 5 VDC được gửi ra mô đun ra, từ đây được khuếch đại lên mức 110VAC, 220VAC hay 24VDC tùy theo yêu cầu. Thông thường một bộ chuyển đổi tín hiệu có giao diện phụ trợ được sử dụng để chuyển trạng thái của các đầu vào từ bên ngoài đến một vùng nhớ đệm xác định. Vùng nhớ đệm này được định nghĩa trong chương trình chính của PLC. Nạp các tín hiệu vào CPU tức là nạp nội dung ghi ở vùng nhớ đệm vào sổ ghi của CPU. Nội dung trong từng vị trí nhớ sẽ được thay đổi kế tiếp nhau. Mô đun Vào/ Ra thường tách khỏi mô đun CPU và được gá trên ray chung. Các đèn báo trên mô đun Vào/ Ra báo hiệu trạng thái làm việc hay sự cố. Các mô đun này được cách điện và có cầu chì để đảm bảo an toàn cho bộ vi xử lý. Trong mô đun Vào/Ra thông thường gồm các mạch sau:

- g. Nguồn AC vào / ra
- h. Nguồn DC vào / ra

- i. Các kênh vào / ra số
- j. Các kênh vào/ ra tương tự
- k. Các môđun chuyên dụng: điều khiển động cơ bước, thiết bị điều khiển PID, bộ đếm thời gian cao tốc, môđun điều khiển servo vv.

Các môđun vào/ ra thường nối với nguồn năng lượng mức cao nên phải cách điện tốt với môđun CPU. Các kênh vào tương tự sử dụng cho việc lấy tín hiệu từ các cảm biến tương tự :

- Cảm biến lưu lượng
- Cảm biến độ ẩm
- Cảm biến áp suất
- Cảm biến nhiệt độ
- Cảm biến áp suất
- Cảm biến vị trí / tốc độ / gia tốc
- Cảm biến lực

Các kênh ra tương tự thường được nối với các cơ cấu chấp hành tương tự:

- Các động cơ DC và AC
- Các van và các động cơ, xi lanh thuỷ khí
- Các thiết bị đo tương tự.

Các kênh vào số thường nối với các cảm biến hai trạng thái dạng đóng/ ngắt (On/Off) như:

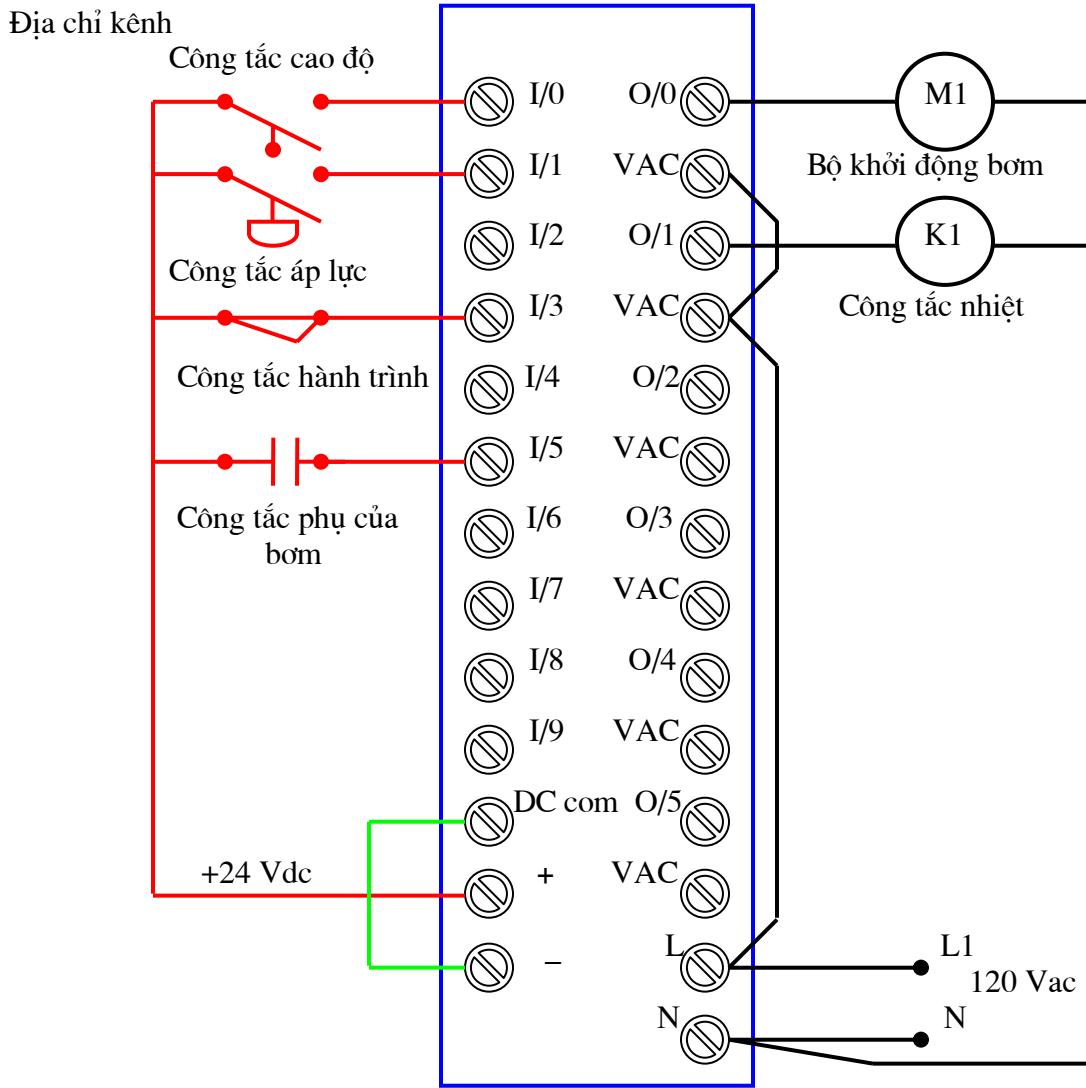
- Cảm biến quang điện,
- Cảm biến tiệm cận
- Cảm biến xung điện
- Các công tắc

Các kênh ra số có thể nối với các thiết bị như:

- Các cuộn hút cho van điện từ
- Các động cơ bước
- Các cơ cấu đóng ngắt vv.

### **Các dạng đầu vào**

Các tín hiệu vào từ các thiết bị hay từ các cảm biến cung cấp các dữ liệu và thông tin cần thiết để bộ xử lý tín hiệu thực hiện các phép tính lô gic yêu cầu quyết định đến việc điều khiển máy hoặc quá trình. Các tín hiệu vào có thể lấy từ các thiết bị khác nhau như nút ấn, công tắc, can nhiệt, ten zô mét, vv. Tín hiệu vào được nối vào các môđun vào để lọc tín hiệu và chuyển đổi tín hiệu về mức năng lượng thấp để bộ xử lý có thể sử dụng được. Đầu vào có hai dạng là đầu vào dạng số và đầu vào dạng tương tự. Đầu vào dạng số được kết nối với các cầu nối kênh trên môđun vào số, các kênh này chỉ có các tín hiệu hai trạng thái 0 hay 1. Đầu vào tương tự có thể là tín hiệu điện áp, dòng điện từ các cảm biến tương tự.



Hình 1.18. Sơ đồ đấu dây của Micro-1000PLC của Allen Bradley

### Các dạng đầu ra

Đầu ra của PLC là các tín hiệu cấp hay ngắn nồng lượng để điều khiển máy hay quá trình. Các tín hiệu này là các tín hiệu điện áp từ các mạch ra và nói chung là chúng có mức nồng lượng thấp. Các tín hiệu này thông thường không truyền trực tiếp đến cơ cấu chấp hành, mà truyền đến các bộ khuếch đại công suất, hoặc các bộ chuyển mạch từ công suất thấp sang công suất cao hơn. Ví dụ tín hiệu điều khiển đóng mở van, tín hiệu này truyền đến cuộn hút của khởi động từ của động cơ, kích hoạt khởi động từ và mạch điện cấp vào động cơ được đóng, động cơ chạy và bắt đầu đóng hay mở van tuỳ theo

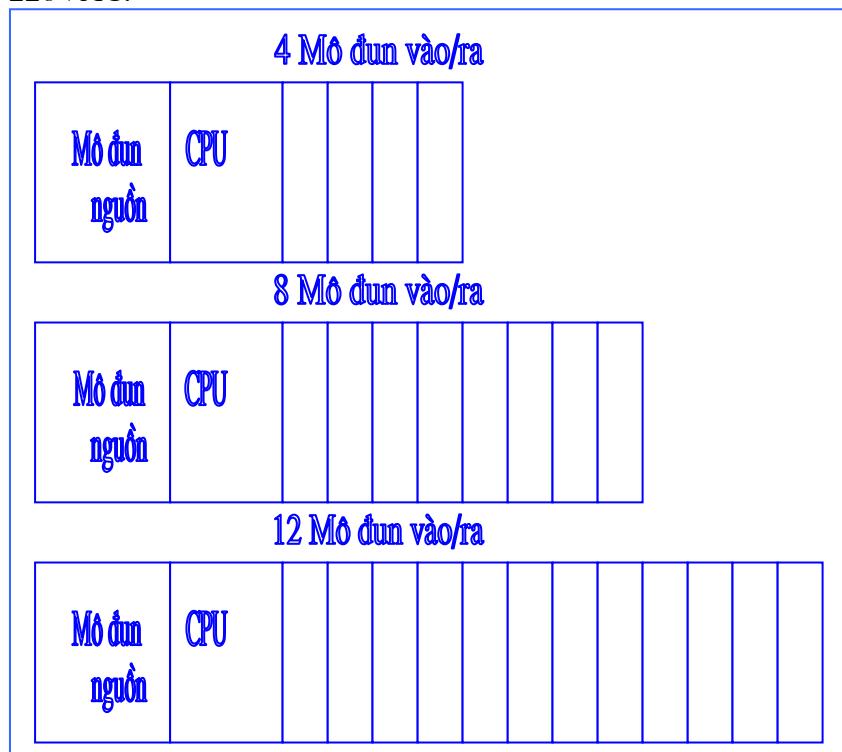
chiều quay của động cơ. Đầu ra cũng có hai dạng tín hiệu là dạng tín hiệu số hoặc tín hiệu tương tự.

### Cấu trúc của các mô đun vào ra

PLC được sản xuất theo dạng mô đun hoá, để thay thế, bổ xung hoặc sửa chữa dễ dàng. Trên hình 1.19 là cấu tạo đặc trưng của các mô đun vào/ ra. Các mô đun này có cấu tạo là các mạch in chứa trong các hộp tiêu chuẩn hoá, có khe truyền dữ liệu song song để có thể kết nối với nhau và truyền dữ liệu vào bộ xử lý. Mặt sau của các hộp này là các cầu nối và nguồn điện áp một chiều để cung cấp cho mạch in hoạt động. Các hộp chứa các mô đun vào/ra có thể lắp lên tủ điều khiển chính hoặc các tủ phụ nhờ các giá đỡ tiêu chuẩn.

### Các mô đun vào/ra số (Digital I/O)

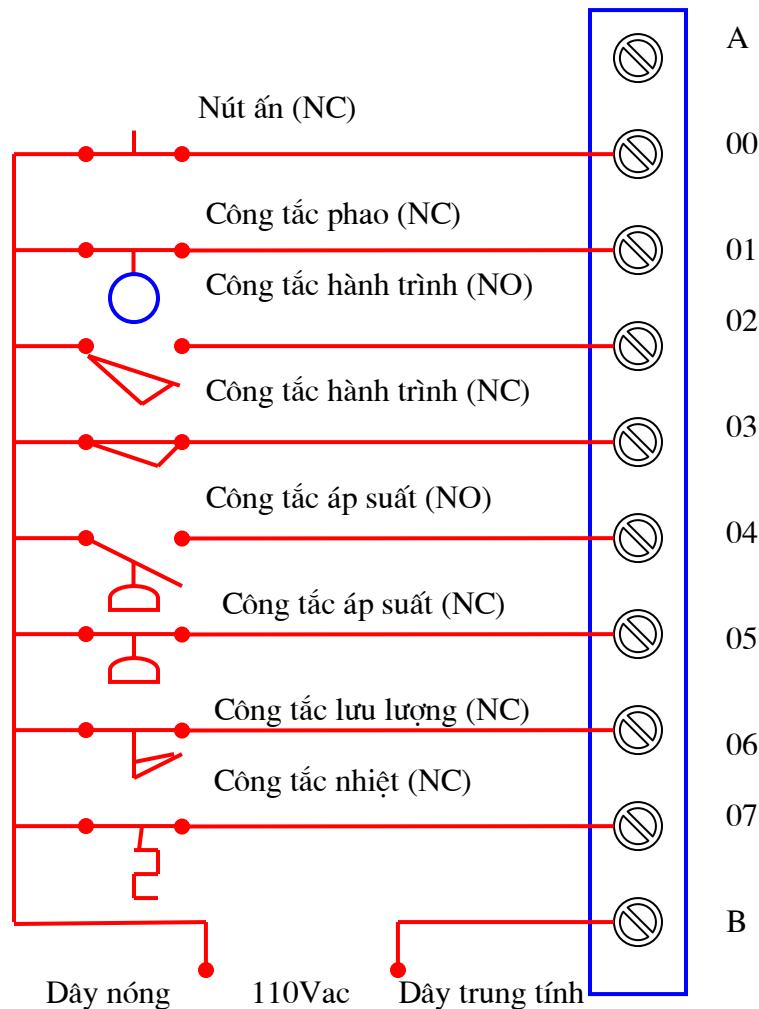
Các kênh vào/ra số là nét chung đặc trưng của phần lớn các hệ thống điều khiển số. Các kênh này đều có hai trạng thái như đóng/ ngắt, mở/ đóng nối qua các giao diện với bộ xử lý tín hiệu. Mỗi mô đun vào/ra số đều được kích hoạt bởi điện áp nguồn do tín hiệu cấp, có thể là điện áp một chiều: +5VDC, +24VDC hay điện áp xoay chiều: 110VAC, 220VAC.



Hình 1.19. Sơ đồ ghép nối các mô đun vào/ ra với CPU

Kênh vào số nếu được nối với công tắc đóng/ngắt thì thông thường nó cấp nguồn điện áp vào trong các mạch in của mô đun. Mô đun vào sẽ chuyển đổi điện áp vào thành mức tương đương với mức tín hiệu lô gíc mà bộ xử lý tín hiệu có thể xử lý được. Giá trị lô gíc 1 tương đương với bật hay đóng, và lô gíc 0 tương đương với ngắt hay mở. Một mạch vào số được biểu diễn trên hình 1.20. Nguồn điện áp cấp đến các thiết bị bên ngoài có thể

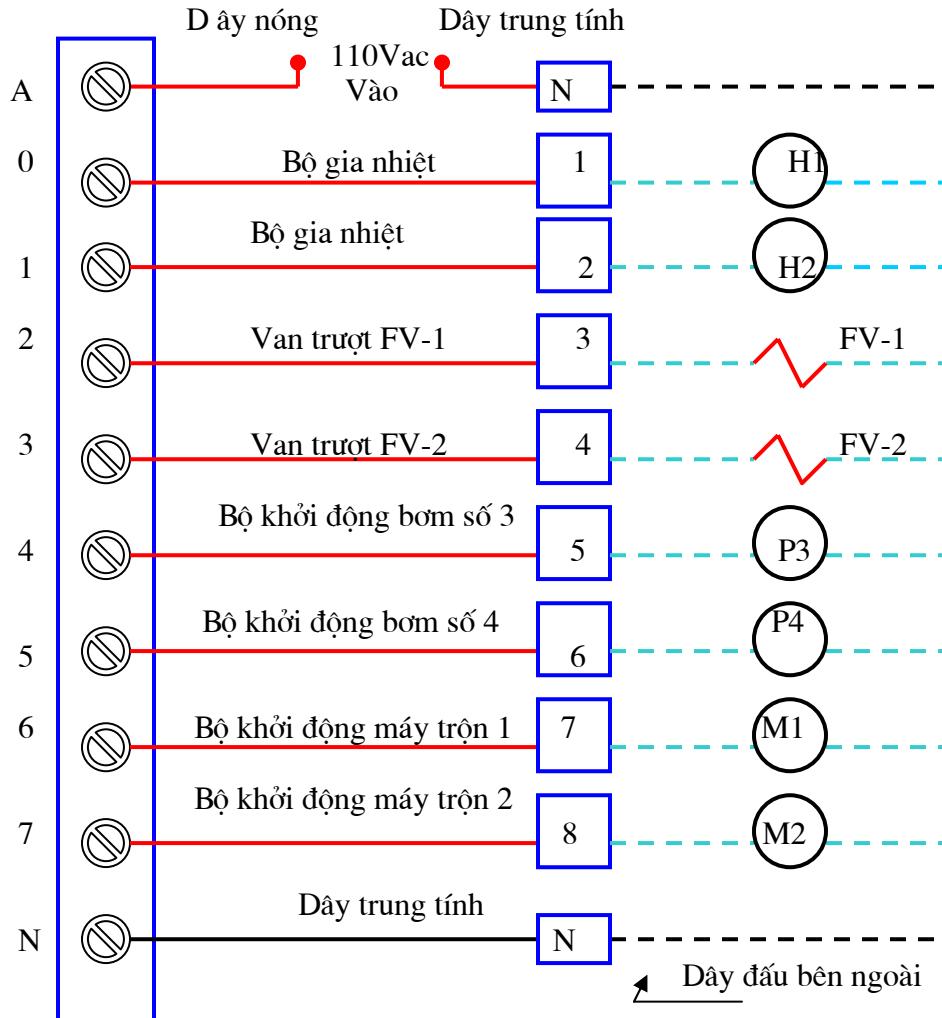
là điện áp 110Vac, 220Vac, +24Vdc, +5Vdc. Các kênh trong mô đun vào này đều có mạnh chuyển đổi điện áp về điện áp +/-5Vdc. Điện áp trên đầu vào khi đi qua các thiết bị như công tắc, nút ấn, nút khởi động đi vào đến PLC thì trở thành tín hiệu lô gíc số chứ không còn là dòng điện chạy qua các tiếp điểm như trong mạch điện rơ le thông thường. Phần lớn các mô đun vào đều có trang bị các đèn LED, để báo hiệu trạng thái của các tín hiệu vào.



Hình 1.20. Sơ đồ đấu dây trên mô đun vào số

Các mô đun ra số có các giao diện để cấp điện áp điều khiển cho cơ cấu chấp hành. Nếu kênh ra được bật lên tức là có giá trị lô gíc 1 từ chương trình điều khiển, mạch ra sẽ được cấp điện áp điều khiển để kích hoạt các cơ cấu chấp hành tương ứng với kênh ra này. Sơ đồ ghép nối của mô đun ra được minh họa trên hình 1.21. Trên mạch ra thường được trang bị cầu chì để đề phòng trường hợp dòng quá tải do dây bị chập, có thể làm hỏng cơ cấu chấp hành. Nếu cầu chì không có, thì nó phải được bổ xung vào trong thiết kế của hệ thống. Các tín hiệu ra trên mô đun số cũng là các tín hiệu hoạt động với hai trạng thái đóng và ngắt hay bật (ON) và tắt (OFF). Các cơ cấu chấp hành dạng này có thể là động cơ, bơm, van, đèn hiệu vv. Các động cơ ở đây không điều khiển tốc độ hay vị trí

mà chỉ đơn thuần là chạy với với một tốc độ cố định hay là dừng chạy. Các xi lanh, các van khí nén hay thuỷ lực phần lớn dùng cho hai trạng thái là làm việc và không làm việc. Các trạng thái của đầu ra được duy trì cho đến khi tính liên tục của lô gíc trên các bậc thang không còn đảm bảo.



Hình 1.21 . Sơ đồ đấu dây trên mô đun ra số

#### **Các mô đun vào/ra tương tự (Analog I/O)**

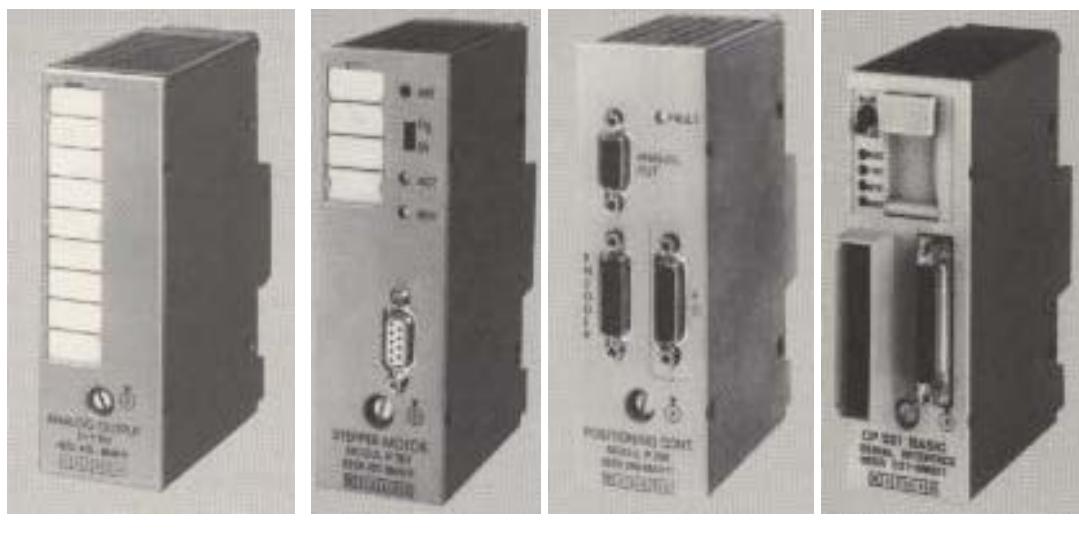
Các mô đun tương tự tạo khả năng theo dõi và điều khiển điện áp hoặc dòng điện tương tự, tương ứng với phần lớn các cảm biến, các động cơ, các thiết bị gia công, xử lý. Bằng việc sử dụng các kênh tương tự phần lớn các đại lượng được điều khiển trong quá trình có thể đo được và điều khiển được nhờ các giao diện tương tự/ số A/D hoặc ngược lại D/A.

Giao diện tương tự/ số thường sử dụng chuẩn đơn cực (unipolar) hoặc lưỡng cực (bipolar). Các kênh vào ra tương tự có thể có mức năng lượng khác nhau, điện áp và dòng điện khác nhau. Các mức năng lượng này có thể chọn thông qua phần cứng hoặc phần mềm. Mô đun vào tương tự được kết nối với các cảm biến tương tự như cảm biến vị trí, tốc độ, áp suất, nhiệt độ vv. Mô đun này có thể dùng trong các thiết bị đo tự động, các hệ thống thu thập dữ liệu, các hệ thống điều khiển tự động. Trên hình 1.22.a là mô đun ra số S5-100U của Siemens.

### **Các mô đun chuyên dụng**

Các mô đun số và tương tự chiếm đến 80% các đại lượng vào/ ra trong các hệ thống điều khiển. Mặc nhiên để xử lý một số dạng tín hiệu hoặc dữ liệu, hệ thống điều khiển cần các mô đun chuyên dụng. Các mô đun này có các giao diện có thể xử lý các tín hiệu vào dang tín hiệu từ can nhiệt, từ các bộ đếm xung, hay các tín hiệu không thể dùng các giao diện vào/ra tiêu chuẩn. Các mô đun chuyên dụng này có thể được trang bị thêm bộ vi xử lý để tạo ra các giao diện thông minh. Các mô đun này có thể thực hiện toàn bộ các chức năng xử lý tín hiệu độc lập với CPU và chu trình quét của chương trình điều khiển. Ví dụ là mô đun điều khiển động cơ bước hình 1.22.b, mô đun điều khiển vị trí hình 1.22.c của S5-100U.

Trong số các mô đun chuyên dụng có cả các mô đun truyền thông hình 1.22.d. Các mô đun này có thể trao đổi với hệ điều khiển phân tán, với mạng PLC khác, các máy tính chủ hoặc các thiết bị thông minh khác.



a, b,

c,

d,

Hình 1.22. Một số mô đun đặc biệt của PLC S5 - Siemens

a, Mô đun ra tương tự S5-100U; b, Mô đun điều khiển động cơ bước;  
c, Mô đun điều khiển vị trí; d, Mô đun kết nối mạng

### **MÔ ĐUN NGUỒN**

Thường nguồn cấp cho PLC là nguồn điện lưới xoay chiều AC để tạo ra nguồn một chiều DC cho các mạch bên trong của PLC. Nguồn điện lưới có thể là 110 VAC, 220 VAC hay điện áp khác tuỳ thuộc theo yêu cầu của người sử dụng. Nguồn này cũng dùng để cấp năng lượng để đóng ngắt động cơ hay các các cơ cấu chấp hành khác nên cần phải được cách điện tốt để tránh gây nhiễu cho mô đun CPU.

### **MÔ ĐUN GHÉP NỐI MẠNG**

Mô đun này cho phép ghép nối các PLC với nhau, với máy tính và các hệ thống điều khiển số khác thông qua mạng nội bộ.

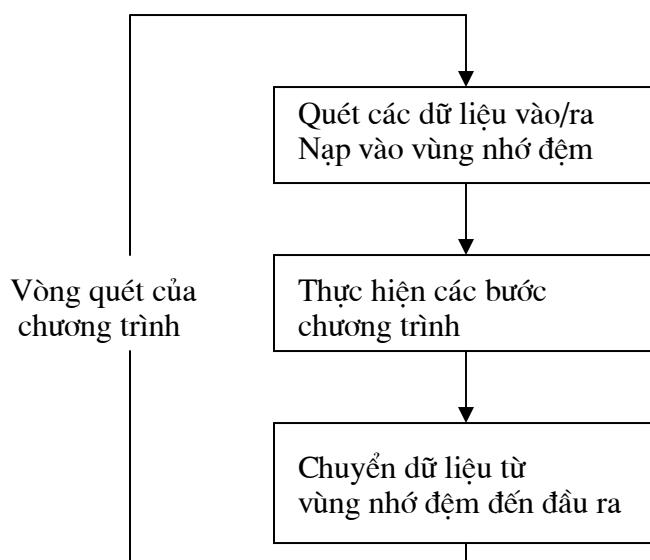
## **MÔ ĐUN PHỤ TRỢ**

Cho phép ghép nối với các thiết bị bên ngoài như màn hình, bàn phím, bộ lập trình cầm tay (hình 1.23), máy in, thiết bị mô phỏng, bộ nạp EPROM, máy tính dạng mô đun hoá, bộ xử lý đồ họa vv. Trong một số hệ thống điều khiển có cần đến màn hình để mô phỏng hay để theo dõi hoạt động của hệ thống, người ta có thể sử dụng một trong hai phương pháp sau:

- Phương pháp thứ nhất là nối các mô đun vào ra của PLC vào một bảng điều khiển với màn hình màu, có trang bị các đèn tín hiệu và các bộ chỉ thị số. Phương pháp này phải kết nối cố định và nếu hệ thống mở rộng thêm sẽ không sử dụng được.
- Phương pháp thứ hai là sử dụng máy tính cá nhân, loại dùng trong môi trường công nghiệp có trang bị phần mềm đồ họa màu. Phương pháp này có ưu điểm là dễ dàng thay đổi màn hình của quá trình điều khiển, có thể thực hiện một số chức năng cảnh báo, lập báo cáo và soạn thảo phần mềm cho PLC.

## **I.5. CHU TRÌNH LÀM VIỆC, LẬP TRÌNH VÀ CẤU TRÚC CHƯƠNG TRÌNH CỦA PLC**

PLC thực hiện chương trình theo một chu trình kín được lặp lại liên tục cho đến khi nào có lệnh dừng. Mỗi vòng lặp hay còn gọi là vòng quét được bắt đầu bằng việc quét các số liệu từ các kênh vào/ra, chuyển các số liệu này đến vùng nhớ đệm đầu vào/ra, tiếp theo là bước thực hiện các lệnh tiếp theo của chương trình như thực hiện các phép tính logic, các phép tính số học để xác định các tác động điều khiển, bước kế tiếp là chuyển dữ liệu từ bộ nhớ đệm đầu ra đến các kênh ra. Khi có một lệnh dừng nào đó xuất hiện thì PLC sẽ dừng các hoạt động xử lý thông tin và truyền tin để kiểm tra khối chương trình tương ứng với lệnh ngắt.



Vòng quét càng ít lệnh dừng thì thực hiện càng nhanh. Nếu chương trình hoạt động bình thường thì chu kỳ của mỗi vòng quét có độ dài như nhau. Tốc độ quét càng cao thì có thể cho phép nhập được nhiều số liệu gần như đồng thời trong thời gian quét, và như vậy khả năng điều khiển được đồng thời nhiều đại lượng là hoàn toàn có thể thực hiện được. Khả năng xử lý tín hiệu trong một chu trình điều khiển không có hiện tượng trễ còn được gọi là điều khiển trong thời gian thực. Các PLC và các PC ngày nay có tốc độ xử lý rất cao nên chất lượng của các hệ thống điều khiển số không kém chất lượng của các hệ thống điều khiển tương tự. Chu kỳ quét của PLC thường vào khoảng từ 1 đến 25 ms/điều khiển. Thời gian quét đầu vào và đầu ra tương đối ngắn so với chu kỳ quét của PLC. Phần lớn thời gian dùng cho việc tính toán các hàm điều khiển.

Thông thường chương trình được nạp vào PLC bởi bộ lập trình cầm tay (hình 1.23), thiết bị lập trình chuyên dụng (hình 1.24) hay máy tính cá nhân (hình 1.25). Bộ lập trình cầm tay thường dùng cho các PLC rẻ tiền, đơn giản. Bộ lập trình chuyên dụng được trang bị màn hình và các phím tương ứng với các phần tử của sơ đồ thang để tiện cho việc lập trình. Các thiết bị này cho phép kiểm tra việc thực hiện các lệnh của chương trình trong thời gian thực. Ngày nay ta thường sử dụng các phần mềm lập trình cho PLC trên máy tính và sau khi chạy thử mô phỏng có thể nạp vào PLC thông qua cổng RS232.

Bộ nạp EPROM cho phép nạp chương trình ghi trên EPROM vào bộ nhớ của PLC. Thiết bị mô phỏng thường gắn với các đĩa quang điện LED hoặc các công tắc để thử nghiệm các bước của chương trình logic.

Bộ xử lý đồ họa thường dùng để làm giao diện giữa hệ thống mô phỏng và hệ thống hiển thị bằng màn hình.

Các PLC hoạt động liên tục từ lúc được bật lên. Khác với máy tính thông thường, PLC không cần có hệ điều hành, không cần có phần mềm nào ngoài phần mềm của người sử dụng và riêng đối với các máy CNC hoặc rô bốt có thể có thêm phần mềm đồ họa dùng cho mô phỏng các quá trình gia công hay các hoạt động của rô bốt. PLC lần lượt đọc các đầu vào, thực hiện tính toán, xác định các tác động điều khiển, truyền các tác động điều khiển đến đầu ra và lặp lại. Kết nối với mô đun vào là các đại lượng vật lý. Các đại lượng vào này có thể là có hai dạng:

- các đại lượng tương tự (analog): là các đại lượng đến từ các cảm biến tương tự.
- các đại lượng lô gíc: là các đại lượng thể hiện các trạng thái hay các điều kiện để thực hiện một hàm lô gíc hay chính là các quyết định lô gíc. Các đại lượng này đến từ các công tắc, cảm biến số.

Các mô đun ra kết nối các đầu ra với các động cơ, các cuộn hút, các đèn tín hiệu vv. Tác động của chương trình điều khiển là các thao tác khởi động động cơ, dừng động cơ, bật/tắt đèn, kích hoạt một cơ cấu nào đó vv.

Tất cả các PLC đều thực hiện các chức năng điều khiển về mặt bản chất là giống nhau. Tuy nhiên về cách thể bằng lập trình có thể khác nhau, phụ thuộc vào nhà sản xuất PLC.

Mỗi đầu vào của PLC được nối với một hay nhiều thiết bị mà qua đó dòng điện bị chặn lại hay được cho đi qua. Nếu có điện áp trên đầu vào thì đầu vào đó được xem như đang ở trạng thái bật. Ngược lại nếu không có điện áp trên đầu vào, có nghĩa là đầu vào đang ở trạng thái tắt.

PLC kiểm tra trạng thái các đầu vào và so sánh với chương trình lô gíc để đóng hay ngắt tín hiệu điện áp trên đầu ra. Các PLC không cần biết đến các thiết bị có được kết nối vào nó qua mô đun vào hay mô đun ra hay không, mà chúng chỉ đơn giản là kiểm

tra các trạng thái của các đầu vào và bật hay tắt các đầu ra tương ứng với lô gíc của chương trình điều khiển.

Mỗi vòng điều khiển hoàn thành được gọi là một chu kỳ quét. Thời gian của một chu kỳ là rất quan trọng, vì nó liên quan đến số lượng các đây ra có thể điều khiển được của PLC. Thời gian chu kỳ càng nhỏ PLC càng hoạt động nhanh, càng có thể điều khiển được nhiều đại lượng vật lý khác nhau. Chính vì vậy PLC trở nên thiết bị điều khiển lý tưởng cho các máy và thiết bị công nghiệp.

Khi chưa có chương trình điều khiển PLC không thể hoạt động được. PLC chỉ hoạt động khi đã có chương trình điều khiển nạp vào bộ nhớ của nó. Chương trình điều khiển có thể nạp vào PLC bằng 3 phương pháp khác nhau:

- Lập trình nhờ các phần mềm lập trình trên máy tính và nạp chương trình lên PLC qua cổng RS232 hay qua cổng kết nối với mạng LAN hay mạng Internet. Máy tính cá nhân là phương tiện lập trình tốt nhất cho PLC, bởi vì chúng ta có thể quan sát được nhiều dòng lệnh trên màn hình, soạn thảo và truy cập vào chương trình dễ dàng. Điều bất tiện là máy tính cá nhân không thích hợp lắp với môi trường công nghiệp và khả năng di chuyển kém.
- Lập trình bằng thiết bị lập trình sách tay: lập trình trực tiếp vào bộ nhớ của PLC. Thiết bị này không dễ sử dụng như máy tính, nhưng lại tiện cho việc mang đi theo người. Lập trình được thực hiện từng dòng lệnh tương ứng với từng bậc của sơ đồ thang.
- Lập trình trên máy tính, nạp lên thẻ nhớ và sau đó nạp từ thẻ nhớ vào PLC qua cổng tiêu chuẩn. Các thẻ nhớ EEPROM là các bộ nhớ ROM có thể xoá và lập trình lại được bằng điện. Ưu điểm của EEPROM là nó có thể thay đổi chương trình của PLC bằng cách cắm vào cổng của PLC.

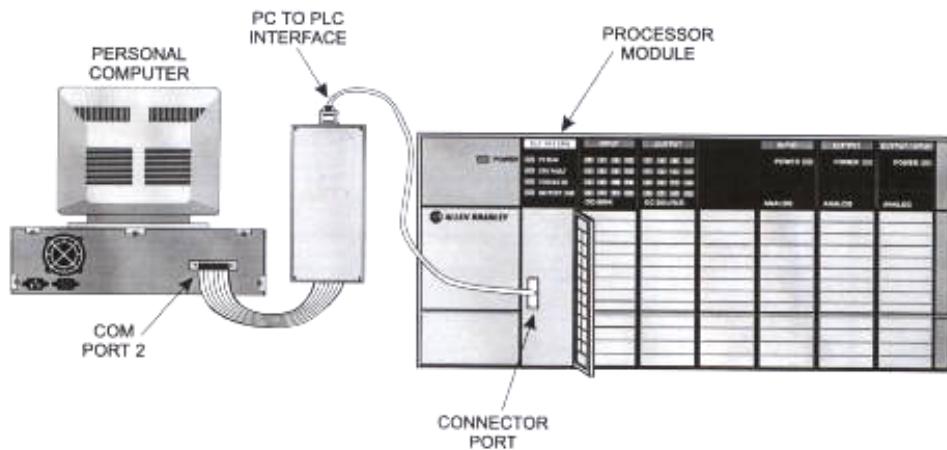


Hình 1.23. Thiết bị lập trình cầm tay PG 605 của Siemens



Hình 1.24. Thiết bị lập trình chuyên dụng PG 730C

Trên hình 1.25 là kết nối máy tính PC để lập trình và nạp vào PLC qua cổng nối tiêu chuẩn.



Hình 1.25. Kết nối máy tính và PLC

Khi nạp chương trình điều khiển từ PC đến PLC, để chương trình có thể chạy được, nó phải được nạp vào bộ nhớ của bộ xử lý. Khi nạp chương trình trực tiếp từ PC cần phải chú ý các thao tác sau:

1. Tất cả các phần tử có liên quan đến PLC phải được ngắt điện.
2. Nối PC với PLC theo đúng như hình 1.25. Như vậy Phần mềm PLC được phép trao đổi với bộ xử lý của PLC.
3. Chuyển công tắc trên bộ xử lý sang chế độ điều khiển từ xa.
4. Bật công tắc nguồn để cấp điện vào PLC và các bộ phận của nó.
5. Thực hiện bước tải chương trình điều khiển từ PC về PLC.
6. Khi việc tải chương trình đã hoàn tất, chuyển sang chế độ gián tiếp, ngừng kết nối với PC (stay offline). Lúc này PLC có thể chạy chương trình mới nạp về.

Phần mềm lập trình cho PLC cũng cho phép PC truy cập trực tiếp vào chương trình đang lưu trong bộ nhớ của PLC. Khi đang ở chế độ truy cập trực tiếp (online), chương trình trong bộ nhớ của PLC sẽ được hiển thị lên màn hình PC. Nếu ta đang có một chương trình nào đó mở sẵn khác với chương trình của PLC, thì phần mềm lập trình sẽ tự động đóng nó lại và chỉ mở chương trình có trong PLC mà thôi. Trên màn hình của phần mềm sẽ có cửa sổ với tín hiệu báo ta đang ở chế độ truy cập trực tiếp. Lúc này ta có thể thay đổi chế độ làm việc của PLC từ chế độ gián tiếp (offline) sang chương trình điều khiển từ xa. Thực hiện việc chạy chương trình điều khiển từ phần mềm lập trình trên PC, ta có thể theo dõi được từng bước thực hiện trên sơ đồ thang.

Phần mềm lập trình còn cho phép dừng chương trình đang chạy trên PLC, khi chuyển sang chế độ chương trình điều khiển từ xa.

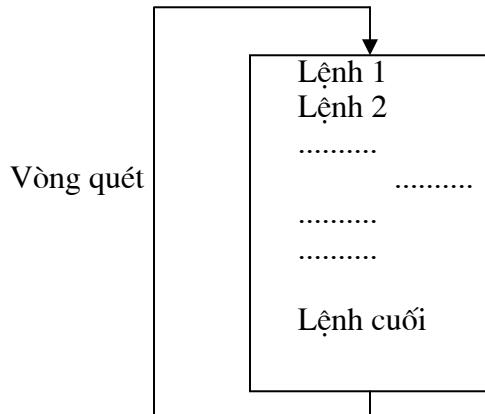
Để nhận biết tính năng của một PLC nào đó ta phải dựa vào đặc tính kỹ thuật của PLC đó. Ví dụ PLC Simatic S5 –100U của Siemens có các đặc tính sau:

- *Dung lượng nhớ: 1024 lệnh*
- *Bộ nhớ tĩnh: EPROM và EEPROM*
- *Thời gian thực hiện một phép tính nhị phân: 70μs*
- *Thời gian chu kỳ: 300ms*
- *Biến trạng thái : 1024, trong đó 512 là biến tĩnh, tức là các biến có thể giữ các dữ liệu ngay cả khi mất điện.*
- *Bộ đếm giờ: 16*
- *Khoảng đếm giờ: 0.01 đến 9990s*
- *Bộ đếm: 16, trong đó 8 là bộ đếm tĩnh.*
- *Khoảng đếm: 0 đến 999 (tăng hoặc giảm)*
- *Kênh Vào/Ra số: 128*
- *Pin: Lithium (3.4V/850mA-h)*
- *Tuổi thọ của pin: 5 năm*
- *Nguồn trung tâm: 24V/0.8A*
- *Cấu trúc của các mô đun vào số:*
  - o 4/8 kênh vào 24V DC/7mA
  - o 4 kênh vào 24 – 60V DC/7.5mA
  - o 4/8 kênh vào 115V AC/10mA
  - o 4/8 kênh vào 230V AC/15mA
- *Cấu trúc của các mô đun ra số:*
  - o 4 kênh ra 24V DC/0.5A
  - o 4 kênh ra 24V DC/2A
  - o 8 kênh ra 24V DC/0.5A
  - o 4 kênh ra 24 – 60V DC/0.5A
  - o 4 kênh ra 115 – 230V AC/1A
  - o 8 kênh ra 150 – 230V AC/0.5A
  - o 4 kênh ra relay: 30V DC/230V AC

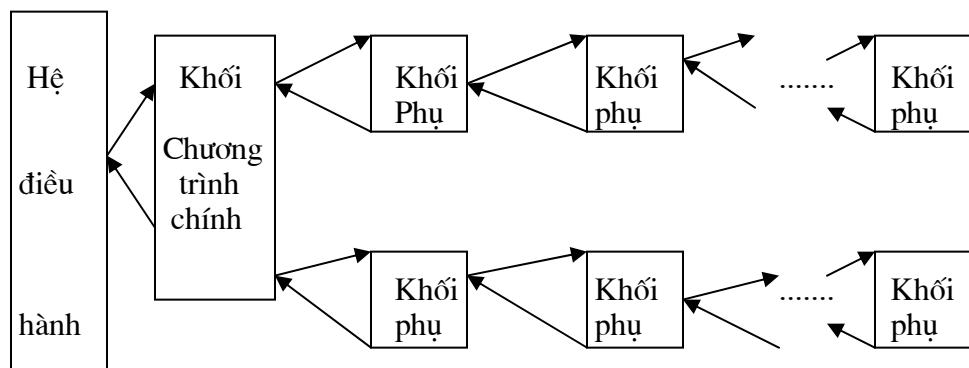
Như vậy ta có thể thấy rằng S5-100U có thể sử dụng được trong điều khiển hệ thống với 128 kênh vào/ra số. Chương trình điều khiển không dài quá 1024 dòng lệnh. Số lượng biến trong chương trình nhiều nhất là 1024. Một chu trình quét của PLC phải nhỏ

hơn hoặc bằng  $300ms$ . Đây là PLC thế hệ của những năm 90 của Siemens. Ngày nay PLC của Hãng này đã phát triển đến thế hệ S7-400, là những PLC rất mạnh và tốc độ rất cao.

Các lệnh của chương trình của PLC thường được gộp vào các khối chương trình con và mỗi chương trình con được liên kết với chương trình chính. Đối với các chương trình đơn giản thì cấu trúc chương trình chỉ gồm một khối.



Đối với các chương trình lớn có nhiều lệnh lặp lại người ta có thể viết chương trình theo dạng có cấu trúc:



## I.6. NGÔN NGỮ LẬP TRÌNH CƠ BẢN

Ngôn ngữ lập trình cho phép người sử dụng trao đổi với thiết bị điều khiển khả lập trình thông qua thiết bị lập trình. Các nhà sản xuất PLC sử dụng các ngôn ngữ lập trình khác nhau, nhưng tất cả các ngôn ngữ này đều sử dụng các lệnh để nạp kế hoạch cơ sở điều khiển vào hệ thống.

Một chương trình điều khiển được định nghĩa như một tập hợp các lệnh sắp đặt theo lô gic điều khiển các hoạt động của một máy hay một quá trình công nghệ. Ví dụ chương trình có thể lệnh cho thiết bị điều khiển bật bộ khởi động động cơ khi nút ấn được

ấn xuống. Chương trình cũng có thể đồng thời lệnh cho thiết bị điều khiển bật đèn trên tủ điều khiển khi tiếp điểm phụ trợ của bộ khởi động động cơ được đóng.

Chương trình được viết bằng sự tổ hợp các lệnh theo một trình tự xác định. Phương thức tổ hợp các lệnh cũng như dạng của các lệnh đều được tuân thủ theo những qui định chung. Các qui định và các lệnh tổ hợp với nhau tạo ra ngôn ngữ lập trình. Có bốn dạng ngôn ngữ hay cơ bản hay sử dụng cho các PLC thế hệ đầu tiên:

1. Bảng lệnh (STT)
2. Sơ đồ thang (LAD)
3. Sơ đồ khối hàm lô gíc (FBD)
4. Grafset

### **NGÔN NGỮ BẢNG LỆNH STT:**

Đây là ngôn ngữ lập trình sử dụng các ký tự thông thường để mã hoá các lệnh của. Cấu trúc của các lệnh tương tự như ngôn ngữ Assembler dùng cho các bộ vi xử lý. Các lệnh này bao gồm các địa chỉ của các bit mà trên đó các lệnh này sẽ tác động lên. Ngôn ngữ bảng lệnh STT bao gồm một dải rộng các lệnh dễ hiểu để lập trình một chương trình điều khiển hoàn chỉnh. Ví dụ PLC Siemens S7 có đến 130 lệnh STT khác nhau và cả một dải rộng các địa chỉ phụ thuộc vào kiểu PLC được sử dụng.

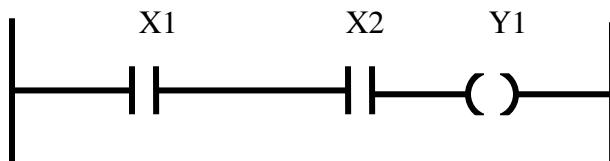
Lệnh STT có hai cấu trúc cơ bản :

- Cấu trúc thứ nhất chỉ có lệnh đơn thuần, ví dụ NOT.
- Cấu trúc thứ hai gồm cả lệnh và địa chỉ.

Địa chỉ của mỗi lệnh chỉ thị một vị trí không thay đổi trong bộ nhớ, nơi mà lệnh đó tìm thấy giá trị và trên đó sẽ thực hiện các phép tính.

Các lệnh lô gíc nhị phân là các lệnh cơ bản nhất của bảng lệnh STT. Các lệnh này thực hiện các phép tính lô gíc trên các bit đơn độc trong bộ nhớ của PLC. Các lệnh lô gíc cơ bản của bit gồm: AND (A), AND NOT (AN), OR (O), EXCLUSIVE OR (OR), EXCLUSIVE OR NOT (XN). Các lệnh này kiểm tra trạng thái tín hiệu của bit địa chỉ để tạo ra hoặc là lô gíc 1 (bit được kích hoạt) hoặc lô gíc 0 (bit không được kích hoạt). Các lệnh lô gíc bit còn được gọi là các lệnh lô gíc rơ le, bởi vì chúng có thể thực hiện các tác động điều khiển thay thế cho các mạch lô gíc rơ le. Trên hình 1.23 là ví dụ về phép tính lô gíc AND. Chương trình bảng lệnh STT ở cột bên trái và chương trình sơ đồ thang ở cột bên phải để tiện so sánh cách diễn đạt lệnh. Lệnh AND lập trình nối hai tiếp điểm nối tiếp NO. Chỉ khi tín hiệu ở các hai bit địa chỉ bằng 1 thì trạng thái của bit đầu ra Q4.0 bằng 1, cuộn hút được kích hoạt.

STR X1  
AND X2  
OUT Y1



a,

b,

Hình 1.23. Lô gíc AND  
a, Bảng lệnh ; b, Sơ đồ thang

Ngôn ngữ này bao gồm một tập hợp các ký hiệu mã hóa tương ứng với một lệnh trong ngôn ngữ máy. Ngôn ngữ PLC của các nhà sản xuất khác nhau cũng rất khác nhau. Ngôn ngữ PLC rất gần với ngôn ngữ máy và rất thích hợp với người sử dụng đã làm quen với kỹ thuật số và máy tính. Một khía cạnh ngôn ngữ PLC là thứ ngôn ngữ duy nhất được sử dụng bởi bộ lập trình đơn giản với khả năng hiển thị chỉ vài dòng lệnh đồng thời. Sau đây là phần giới thiệu ngôn ngữ bảng lệnh cổ điển.

### 1. **Nhận dạng các biến:**

- a. Biến vào Xn. Ký hiệu X chỉ biến vào nhị phân và chỉ số n ký hiệu địa chỉ của kênh nối với đầu vào.
- b. Biến ra Yn. Ký hiệu Y chỉ biến ra nhị phân và n chỉ địa chỉ của kênh nối với đầu ra.
- c. Biến trung gian IRn. Ký hiệu IR chỉ biến nhị phân trung gian (chỉ trong bộ nhớ) và n là chỉ số thứ tự tương ứng.

### 2. **Các lệnh:**

PLC sử dụng ba loại lệnh khác nhau:

- *Lệnh gọi biến vào / ra hoặc lệnh tính toán.*
- *Lệnh đếm giờ hoặc lệnh đếm.*
- *Lệnh điều khiển.*

#### **Lệnh gọi biến vào/ ra hoặc tính toán:**

Các lệnh này thực hiện một trong các thao tác sau:

- + Chọn một biến xác định được sử dụng như một biến gán (Operand), có thể là đầu vào hoặc đầu ra.
- + Thực hiện lệnh quét đầu vào hoặc đầu ra.
- + Thực hiện một số phép tính với một biến đã cho.

Trong nhóm này có các lệnh sau: STR, STR NOT, OUT, OUT NOT, OR, OR NOT, AND, AND NOT, OR STR, AND STR.

**Lệnh STR:** Lệnh này dùng để chọn biến đầu tiên của chuỗi lệnh (lệnh gọi).

Ví dụ : STR X0 - Chọn đầu vào X0

STR Y6 - Chọn đầu ra Y6

STR IR2 - Chọn biến trung gian IR2

**Lệnh STR NOT:** là lệnh phủ định giá trị của biến được chọn. Đây cũng là lệnh gọi và bắt đầu cho một chuỗi lệnh.

Ví dụ : STR NOT X12 - Chọn biến vào X12 và phủ định biến này ( $\overline{X12}$ )

STR NOT Y10 - Chọn biến vào Y10 và phủ định biến này ( $\overline{Y10}$ )

STR NOT IR9 - Chọn biến vào IR9 và phủ định biến này ( $\overline{IR9}$ )

**Lệnh OUT:** Lệnh này chuyển dữ liệu ra kênh ra.

Ví dụ: STR X0

OUT Y0 – chọn X0 và chuyển giá trị này ra kênh ra Y0.

**Lệnh OUT NOT:** Lệnh này phủ định dữ liệu cần chuyển đến kênh ra.

Ví dụ: STR X0

OUT NOT Y0 – Chọn giá trị vào X0, gán nó cho đầu ra Y0 giá trị phủ định của X0.

**Lệnh OR:** Thực hiện phép cộng logic giữa hai hay nhiều biến.

Ví dụ: STR Y5

OR X3

OR IR7

### OUT Y0

Nội dung của chuỗi lệnh này :

- o Chọn biến ra Y5
- o Thực hiện phép tính logic OR giữa Y5 và X3
- o Thực hiện phép tính logic OR giữa kết quả phép tính trước với IR7
- o Chuẩn kết quả tới kênh ra Y0

Đây chính là kết quả của phép tính sau:  $Y0 = Y5 + X3 + IR7$

*Lệnh OR NOT:* Lệnh này thực hiện phép cộng logic với một hay nhiều biến khác, tiếp theo là phủ định kết quả.

Ví dụ: STR IR13  
OR NOT X10  
OR NOT X14  
OUT IR15

Đây chính là thực hiện phép tính logic :  $IR15 = IR13 + \overline{X10} + \overline{X14}$ .

*Lệnh AND :* Đây thực tế chính là phép nhân logic giữa hai hay nhiều biến logic.

Ví dụ: STR NOT X0  
AND X1  
AND IR7  
AND Y3  
OUT Y10

Chuỗi lệnh này thực hiện các thao tác sau:

- Gọi biến vào X0 và phủ định giá trị của nó.
- Thực hiện phép nhân AND giữa X0 và X1.
- Thực hiện phép nhân giữa IR7 và kết quả phép tính trước.
- Thực hiện phép nhân giữa Y3 với kết quả của phép tính trước.
- Chuyển kết quả ra kênh ra Y0.

Đây là kết quả của phép tính:  $Y10 = \overline{X0}.X1.IR7.Y3$ .

Giữa phép cộng và phép nhân logic không có gì khác biệt, lệnh nào theo trình tự đứng trước sẽ thực hiện trước.

Ví dụ: STR X5  
OR X3  
AND Y5  
OUT Y3

Đây là phép tính:  $Y3 = (X5 + X3).Y5$ .

STR X5  
AND X3  
OR Y5  
OUT Y3

Đây là phép tính:  $Y3 = X5.X3 + Y5$

*Lệnh AND NOT:* Lệnh này thực hiện phép nhân logic giữa hai hay nhiều biến và sau đó phủ định kết quả.

Ví dụ: STR Y6  
AND NOT X3  
AND NOT IR9  
AND NOT X9  
OUT IR14

Đây là chuỗi lệnh thực hiện phép tính:  $IR14 = Y6.\overline{X3}.\overline{IR9.X9}$ .

*Lệnh OR STR:* Lệnh này thực hiện phép công logic giữa hai chuỗi trước đó bắt đầu bằng STR hay STR NOT.

Ví dụ: STR X7

```
    OR X9  
    AND NOT Y5  
    STR NOT IR3  
    AND X6  
    OR NOT Y6  
    OR STR  
    OUT Y8
```

Chuỗi lệnh này thực hiện các thao tác sau:

- Gọi biến vào X7
- Thực hiện logic OR giữa X7 và X9
- Thực hiện logic AND giữa  $\overline{Y5}$  và  $X7+X9$
- Bắt đầu chuỗi lệnh mới với lệnh gọi biến trung gian  $\overline{IR3}$ .
- Thực hiện logic AND giữa  $\overline{IR3}$  và X6.
- Thực hiện logic OR giữa  $\overline{Y6}$  và  $\overline{IR3}.X6$ .
- Thực hiện phép tính OR giữa  $(X7 + X9).\overline{Y5}$  và  $\overline{IR3} + X6 + \overline{Y6}$ .

Kết quả của chuỗi này chính là:

$$Y8 = [(X7 + X9).\overline{Y5}] + (\overline{IR3}.X6 + \overline{Y6})$$

*Lệnh AND STR:* là lệnh thực hiện phép nhân logic AND giữa hai chuỗi lệnh bắt đầu bằng lệnh gọi STR hay STR NOT.

Ví dụ: STR X0

```
    AND NOT X1  
    STR X2  
    AND X3  
    OR NOT Y0  
    AND STR  
    OUT Y1
```

Chuỗi lệnh này thực hiện phép tính:  $Y1 = X0.\overline{X1}.(X2.X3 + \overline{Y0})$

#### ***Lệnh đếm thời gian TMR và lệnh đếm CTR:***

Lệnh thời gian và lệnh đếm là các lệnh để tạo ra khả năng đóng ngắt, kéo dài thời gian thực hiện một lệnh hay một chuỗi lệnh ào đó trong chương trình. Các lệnh này là hàm của thời gian hoặc của số lượng xung đếm tác động lên đầu vào của chúng.

**Lệnh TMR:** lệnh này sử dụng hai biến để thực hiện chức năng đếm thời gian là  $X_i$  và  $X_j$ . Đầu ra của lệnh thời gian có thể là một biến trung gian hoặc một đầu ra.

Lập trình cho bộ đếm thời gian cần một chuỗi các lệnh sau:

1. Lệnh khởi động biến  $X_i$

2. Lệnh gọi biến  $X_j$ .

3. Lệnh TMR n chọn bộ đếm thứ n. Lệnh này khởi động bộ đếm thời gian n nếu  $X_i=1$  (Chưa khởi động lại) và  $X_j$  cũng trở thành 1 (đầu vào được bật lên).

4. Một chương trình ghi trong bộ nhớ các giá trị của thời gian được chọn.

Ví dụ: STR X1

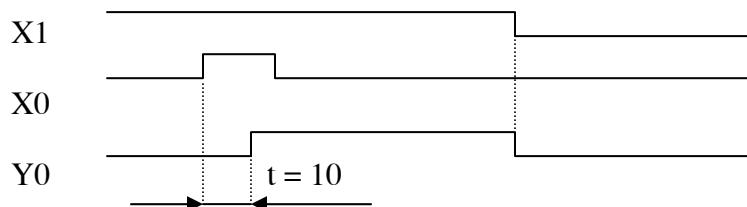
STR X0

TMR 0

10

OUT Y0

Biểu đồ thời gian của các biến như sau:



Lệnh đếm thời gian TMR có thể sử dụng để làm *trẽ thời gian đóng ngắt* của một biến nào đó.

Ví dụ: STR X5

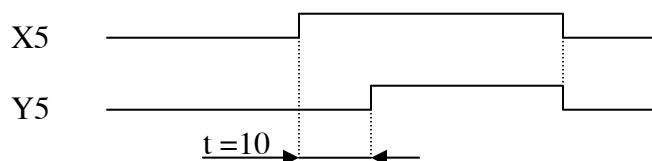
STR X5

TMR 2

10

OUT Y5

Biểu đồ thời gian :



Tương tự như vậy, lệnh TMR có thể sử dụng để *kéo dài thời gian* của biến đang ở trạng thái đóng trước khi ngắt.

Ví dụ: STR NOT X4

STR NOT X4

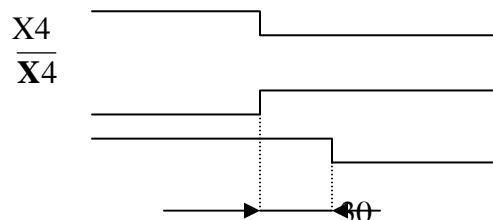
TMR 6

10

OUT NOT Y9

Như vậy biến ra Y9 sẽ được ngắt trễ 10s từ lúc biến vào  $\overline{X4}$  được ngắt.

Biểu đồ thời gian:



Y9

t=10

**Lệnh đếm CTR:** Lệnh này thực hiện chức năng của một bộ đếm với hai hoặc 3 biến. Trường hợp thứ nhất là đếm tăng, biến đầu tiên là biến khởi động và biến thứ hai là biến đếm. Trường hợp thứ hai là bộ đếm tăng/giảm, bộ đếm này sử dụng ba biến. Biến thứ nhất và biến thứ ba tương tự như ở bộ đếm tăng. Biến thứ hai là biến chọn kiểu đếm tăng hay giảm tương ứng với trạng thái 0 hay 1.

Ví dụ: STR X1

STR X0

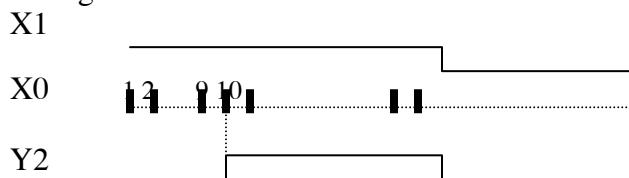
CTR 3

10

OUT Y2

Biến Y2 trở thành 1 sau khi có 10 xung đếm trên đầu vào X0, đồng thời X1 giữ nguyên trạng thái 1 và chuyển về 0 khi X1 chuyển về 0.

Biểu đồ thời gian:



**Chuỗi lệnh của bộ đếm tăng/giảm gồm :**

STR X2

STR X1

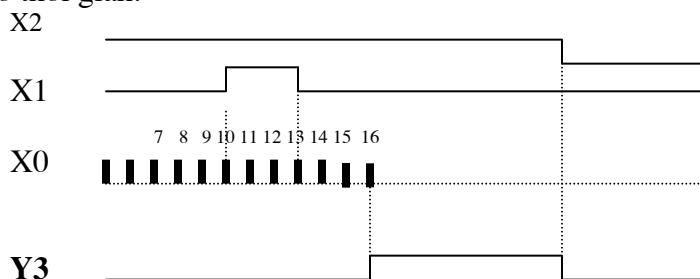
STR X0

CTR 4

10

OUT Y3

Biểu đồ thời gian:



Trong bộ đếm tăng giảm ở ví dụ trên ta thấy bộ đếm đã đếm đến 10, nhưng biến X1 thay đổi trạng thái từ 0 sang 1, giá trị lưu trong bộ đếm được đếm giảm cho đến khi X1 quay về trạng thái 0. Lúc này bộ đếm lại đếm tăng cho đến khi giá trị lưu trong nó đạt giá trị là 10. Đầu ra được kích hoạt khi bộ đếm đạt giá trị cho trước (10). Đầu ra ngắt khi biến X2 chuyển về trạng thái 0.

**Các lệnh điều khiển:**

Các lệnh điều khiển ảnh hưởng trực tiếp đến việc thực hiện các lệnh khác. Tuy nhiên PLC có thể thực hiện chương trình không cần đến các lệnh này, như vậy

chương trình sẽ dài và công kẽm. Các PLC thường sử dụng các cặp lệnh: JMP – JME, IL - ILC.

*Lệnh nhảy JMP – JME:*

Hai lệnh này gây ra việc thực hiện tất cả các lệnh nằm giữa chúng phụ thuộc vào kết quả phép tính logic nằm trước lệnh JMP. Nếu kết quả phép tính lô gíc trước lệnh JMP là 1 thì các lệnh nằm giữa JMP và JME được thực hiện bình thường. Nếu có các biến ra thì các biến này sẽ được gán giá trị mới. Nếu kết quả trước lệnh JMP bằng 0 thì các lệnh này không được thực hiện. Các biến nằm giữa hai lệnh này không thay đổi giá trị.

Ví dụ :  
STR X0  
OUT Y0  
JMP  
STR Y0  
AND X0  
OUT Y1  
STR X1  
OUT IR2  
JME

Như vậy đầu tiên là lệnh gọi biến vào X0, sau đó gán Y0 bằng X0. Bước tiếp theo lệnh nhảy JMP thực hiện phụ thuộc vào giá trị của biến ra Y0 của lệnh trước nó. Lệnh JME kết thúc lệnh nhảy.

*Lệnh IL – ILC:*

Cặp lệnh này gây ra cho tất cả các biến trung gian và biến ra nằm giữa chúng thay đổi giá trị hay giữ nguyên tùy thuộc vào kết quả logic của lệnh nằm trước lệnh IL

Ví dụ: STR X7  
OUT Y10  
IL  
STR X9  
OUT Y9  
STR X15  
OUT Y15  
ILC

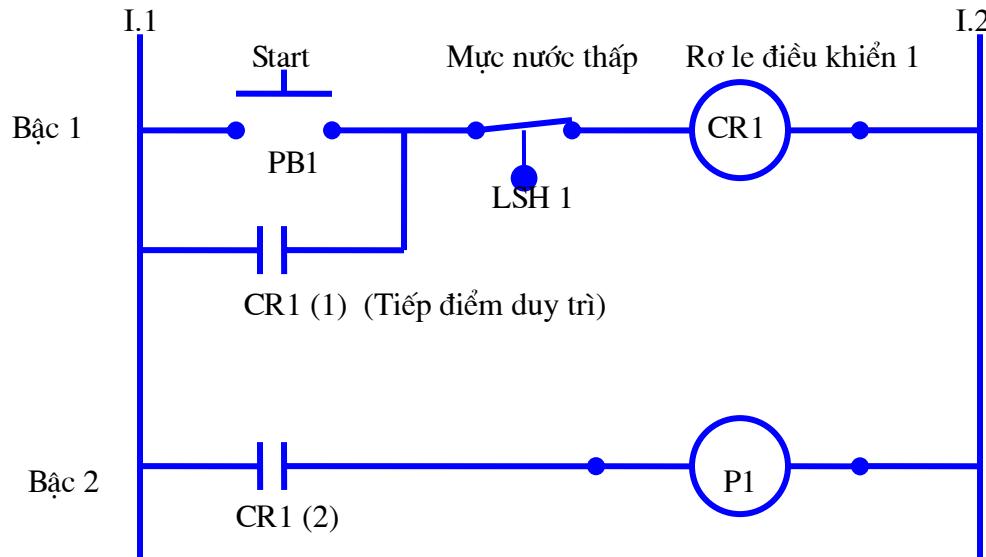
Đầu tiên là lệnh gọi biến vào X7, gán nó cho biến ra Y10. Nếu Y10 = 1, thì các lệnh tiếp theo nằm giữa IL – ILC được thực hiện, ngược lại các lệnh này không được thực hiện, Y9 và Y15 giữ nguyên giá trị cũ.

## NGÔN NGỮ SƠ ĐỒ THANG (LAD):

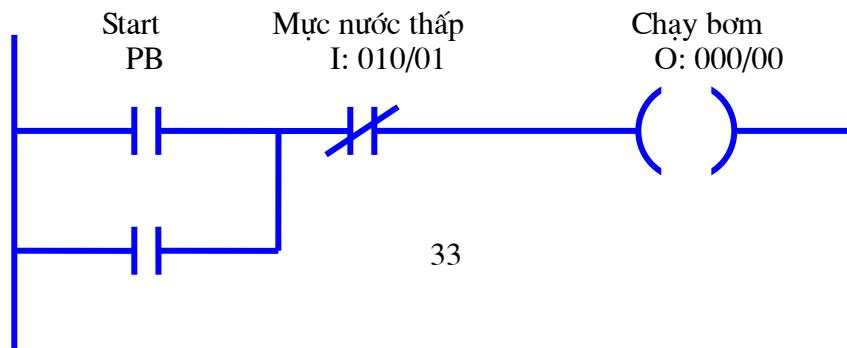
Sơ đồ thang là ngôn ngữ thông dụng nhất, thường được dùng cho các ứng dụng của PLC, bởi vì nó rất đơn giản. Sơ đồ thang tương tự như sơ đồ lô gíc rơ le cho nên các kỹ sư và các kỹ thuật viên đã sử dụng sơ đồ lô gíc rơ le thì không cần phải qua đào tạo cũng có thể sử dụng được ngôn ngữ sơ đồ thang. Đây là dạng ngôn ngữ ký hiệu và các ký hiệu được sử dụng tương tự như các ký hiệu trong các hệ thống lô gíc rơ le. Sơ đồ thang gồm có các chuỗi ký hiệu nối tiếp nhau qua một dây nguồn, cấp dòng điện cho các thiết bị khác nhau. Bản vẽ sơ đồ thang có hai thành phần cơ bản là nguồn năng lượng và các thiết bị lô gíc điều kiện khác nhau tạo thành các bậc thang. Dòng điện lần lượt chạy qua từng bậc thang khi các đầu vào lô gíc hay các điều kiện lô gíc được đáp ứng và kích hoạt

các cuộn hút của rơ le. Trong thiết kế các mạch lô gíc rơ le, người ta cố gắng chỉ ra các mạch điện cần thiết để thực hiện một thao tác của hệ thống điều khiển và sơ đồ như vậy còn được gọi là sơ đồ đấu dây. Sơ đồ này thể hiện lô gíc điều khiển bằng vật chất cụ thể để đảm bảo cho dòng điện đi liên tục qua các phần tử kết nối đầu vào cho đến các phần tử đầu ra như động cơ, cuộn hút vv. Đối với PLC thì điều này khác hẳn về bản chất, bởi vì sơ đồ thang trên PLC chỉ đảm bảo tính liên tục về lô gíc chứ không phải là cho dòng điện chạy qua từ đầu vào đến đầu ra. Đầu ra của PLC được kích hoạt hay được cấp năng lượng khi các biến lô gíc tương ứng với các thiết bị “cứng” đảm bảo tính lô gíc liên tục từ đầu vào đến đầu ra. Mỗi bậc thang của sơ đồ thang trong PLC so với bậc thang tương ứng trong sơ đồ đấu điện, thì chỉ là sơ đồ đấu “ảo” mà thôi.

Trên hình 1.23 là ví dụ về sơ đồ thang của mạch điện và hình 1.24 là sơ đồ thang của PLC cho thao tác đóng bơm khi mức nước giám quá mức tối thiểu. Các địa chỉ của bit lô gíc được ký hiệu bởi các chữ I và O, tiếp theo là 5 chữ số thập phân. Bắt đầu chương trình, bộ xử lý kiểm tra trạng thái của nút ấn khởi động PB, khi nút ấn được ấn xuống, lô gíc của bit này là I:010/00 trở thành 1. Bước tiếp theo là kiểm tra trạng thái của bit I:010/01 – trạng thái của mực nước trong bể chứa. Nếu mực nước thấp, trạng thái của bit này là 0 (OFF), lô gíc của bít khởi động bơm chuyển thành 1 (ON) bơm chạy. Trạng thái của bit khởi động bơm duy trì tính liên tục của mạch lô gíc và bơm tiếp tục chạy cho đến khi trạng thái của bit mực nước chuyển sang trạng thái 1 (ON), tức là mạch lô gíc bị gián đoạn. Trên bậc thang thứ hai là khi lô gíc của bơm đang là 1 thì lô gíc của đèn tín hiệu chỉ bơm đang chạy có giá trị là 1, đèn sáng.



Hình 1.23. Sơ đồ thang của mạch điều khiển bơm



Chạy bơm

O: 000/00

Chạy bơm

Đèn báo bơm chạy

O: 000/00

O: 000/01

Hình 1.24. Sơ đồ thang điều khiển bơm của PLC

Về mặt lô gíc cả hai sơ đồ hình 1.23 và hình 1.24 hoàn toàn tương tự như nhau. Chính vì điều này các kỹ sư đã làm việc với sơ đồ thang của mạch điện không mất nhiều thì giờ trong việc học cách sử dụng sơ đồ thang trên PLC.

Trong chương trình sử dụng sơ đồ thang có ba dạng lệnh được sử dụng để tạo nên chương trình đó là:

- **Lệnh thường mở NO** (Normally Open), tương ứng với tiếp điểm thường mở trong mạch lô gíc rơ le. Lệnh NO trong PLC cũng tương tự, nhưng lệnh này yêu cầu bộ xử lý tín hiệu kiểm tra bit có ký hiệu tương ứng với tiếp điểm này trong bộ nhớ. Nếu bit này là 1 (tương ứng trạng thái bật ON) thì lệnh được thực hiện và tính liên tục của lô gíc lại truyền tiếp tục trên bậc thang. Nếu bit này mang giá trị 0 tức là trạng thái tắt OFF, thì lô gíc bị ngắt quãng, không thể tiếp tục truyền đi tiếp trên bậc thang.
- **Lệnh thường đóng NC** (Normally Closed) tương ứng với tiếp điểm thường đóng trên sơ đồ thang của mạch điện. Lệnh này còn gọi là lệnh ngắt vì khi thực hiện nó sẽ ngắt điện trong mạch điện hay ngắt mạch lô gíc trên PLC. Nếu bit tương ứng với tiếp điểm thường đóng có trạng thái lô gíc là 0, tương đương với lô gíc OFF, lệnh được thực hiện và tính liên tục của lô gíc được truyền đi tiếp trên bậc thang. Nếu bit này có giá trị là 1 tức là lô gíc ON, thì lệnh thường đóng trở thành sai (FALSE), lô gíc bị ngắt quãng ở đây.
- **Lệnh ra cuộn hút** : tương tự như cuộn hút của rơ le trong sơ đồ thang. Lệnh này yêu cầu bộ xử lý chuyển giá trị lô gíc của vị trí xác định trong bộ nhớ tương đương với cuộn hút lên trạng thái 1 hay ON (bật) nếu như tính liên tục của lô gíc trước đó được đảm bảo. Nếu không có sự liên tục của lô gíc trên bậc thang thì bộ xử lý sẽ bật lệnh cuộn hút lên giá trị 0 hay OFF (tắt).

Trong sơ đồ thang tất cả các lệnh được thể hiện bằng sơ đồ tương tự như các mạch điện điều khiển trong các tủ rơ le. Mục đích của ngôn ngữ này là :

- đơn giản hoá việc thay hệ thống điều khiển bằng rơ le bởi PLC,
- đơn giản hoá việc lập trình PLC cho các kỹ sư điều khiển đã quen với thiết kế của các hệ điều khiển rơ le.

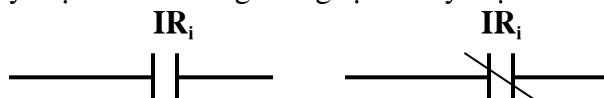
Để lập trình được bằng ngôn ngữ LAD, ta cần phải được trang bị bộ lập trình với màn hình đồ họa để có thể hiển thị được sơ đồ thang.

1. *Nhận dạng biến*: Các biến nhị phân được biểu diễn bằng các công tắc xác định bởi một chữ cái và một chữ số xác định từ danh sách các lệnh Tuy nhiên việc ký hiệu công tắc cũng rất khác nhau, phụ thuộc vào tiêu chuẩn của nước sản xuất.

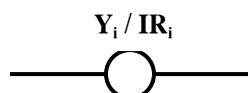
Biến vào  $X_i$  được ký hiệu như sau:



Biến trung gian  $IR_i$  có thể là biến , có thể là đầu ra trung gian. Trường hợp thứ nhất biến  $IR_i$  là biến trung gian thì ký hiệu của nó cũng tương tự như ký hiệu của các biến vào  $X_i$ .

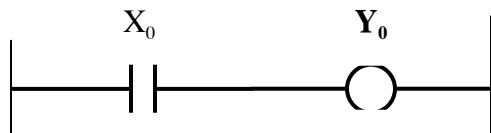


Biến ra  $Y_i$ :



2. *Chuỗi logic* :

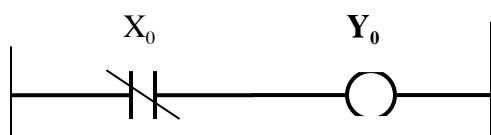
Lệnh gọi biến vào:



STR X<sub>0</sub>

OUT Y<sub>0</sub>

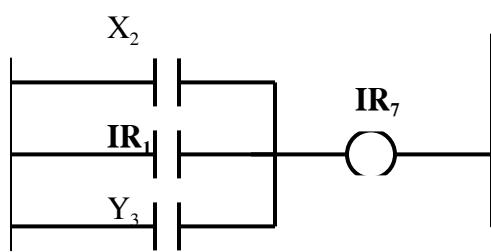
Lệnh gọi và phủ định biến vào:



STR NOT X<sub>0</sub>

OUT Y<sub>0</sub>

Lệnh cộng OR:

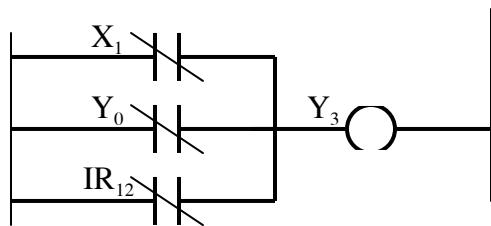


STR X<sub>2</sub>

OR IR<sub>1</sub>

OR Y<sub>3</sub>

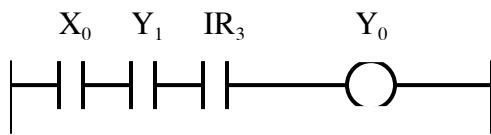
OUT IR<sub>7</sub>



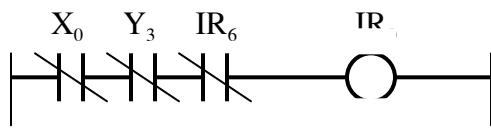
STR NOT X<sub>1</sub>

OR NOT Y<sub>0</sub>  
OR NOT IR<sub>12</sub>  
OUT Y<sub>3</sub>

Lệnh AND :

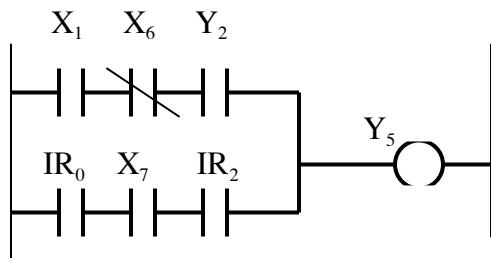


STR X<sub>0</sub>  
AND Y<sub>1</sub>  
AND IR<sub>3</sub>  
OUT Y<sub>0</sub>



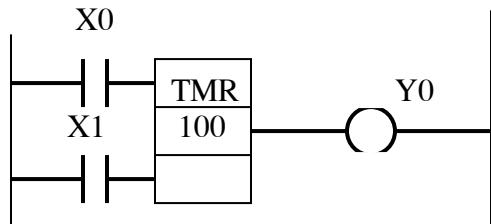
STR NOT X<sub>0</sub>  
AND NOT Y<sub>3</sub>  
AND NOT IR<sub>6</sub>  
OUT IR<sub>0</sub>

Lệnh OR và AND :



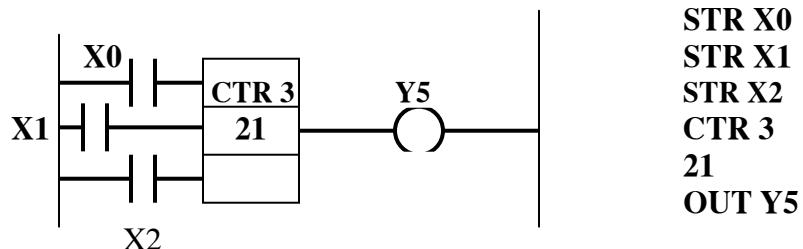
STR X<sub>1</sub>  
AND NOT X<sub>6</sub>  
AND Y<sub>2</sub>  
STR IR<sub>0</sub>  
AND X<sub>7</sub>  
AND IR<sub>2</sub>  
OR STR  
OUT Y<sub>5</sub>

Lệnh đếm thời gian TMR



STR X0  
STRX1  
TMR 0  
100  
OUT Y0

## Lệnh đếm CTR



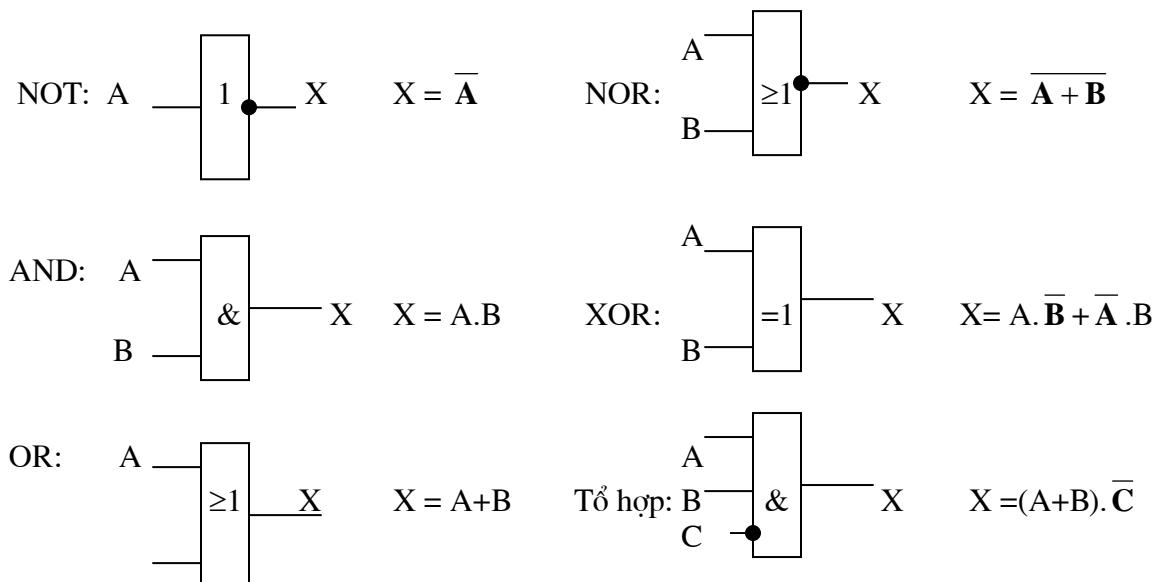
## NGÔN NGỮ SƠ ĐỒ KHỐI CỦA HÀM LÔ GÍC FBD

Sơ đồ khối của hàm lô gíc là ngôn ngữ lập trình đồ họa. Ngôn ngữ này cho phép người lập trình xây dựng các qui trình điều khiển phức tạp bằng cách lấy các hàm từ thư viện FBD và viết chúng vào một diện tích đồ họa. Một khối hàm lô gíc biểu diễn quan hệ hay hàm giữa các biến đầu vào và đầu ra.

Ta có thể xây dựng hàm hoàn chỉnh thao tác bởi chương trình FBD với các sơ đồ khối của các hàm cơ sở từ thư viện FBD. Mỗi khối hàm cơ sở có số lượng đầu vào/ra cố định trên các điểm nối. Đầu vào được nối vào mặt bên trái của khối và đầu ra mặt bên phải. Hàm cơ sở thực hiện các hàm đơn giản giữa các đầu vào và đầu ra của nó. Kết quả của hàm lô gíc được chuyển đến đầu ra. Tên của khối được ký hiệu bằng các ký tự la tinh.

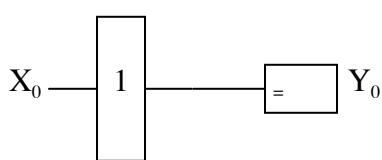
Sơ đồ khối hàm logic là một ngôn ngữ ký hiệu, trong đó các tổ hợp khác nhau của các biến lô gíc được biểu diễn bởi các ký hiệu logic tiêu chuẩn hóa. Tương tự như sơ đồ thang, để lập trình được trong ngôn ngữ này ta cần phải có màn hình để hiển thị sơ đồ.

*1. Các biến logic:* Tương tự như các ngôn ngữ khác, các biến được ký hiệu là các chữ in hoa. Các phần tử lô gíc có thể có nhiều dạng ký hiệu khác nhau, tùy thuộc vào tiêu chuẩn sử dụng: BS, AFNOR, ISO vv.. Các ký hiệu của các phần tử lô gíc theo chuẩn ISO như sau:

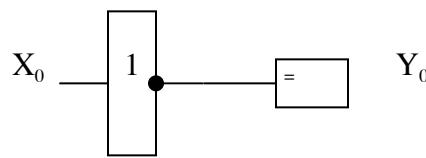


B

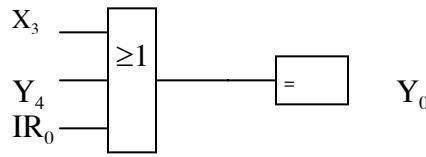
2. Các phép tính logic:



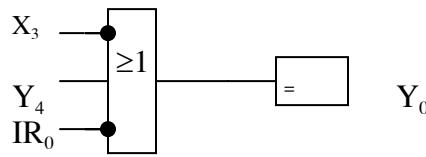
Tương đương với các lệnh sau:  
STR X<sub>0</sub>  
OUT Y<sub>0</sub>



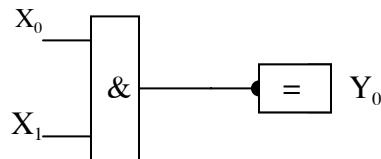
STR NOT X<sub>0</sub>  
OUT Y<sub>0</sub>



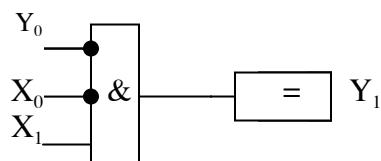
STR X<sub>3</sub>  
OR Y<sub>4</sub>  
OR IR<sub>0</sub>  
OUT Y<sub>0</sub>



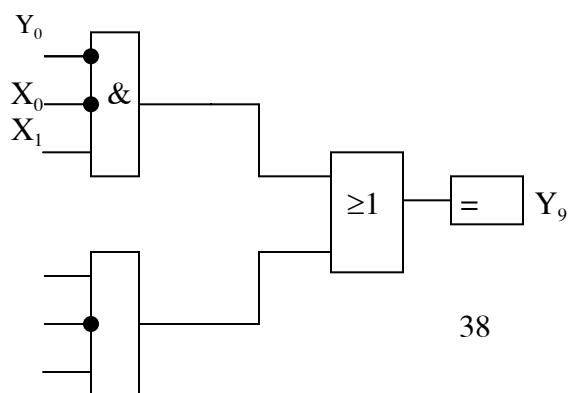
STR NOT X<sub>3</sub>  
OR Y<sub>4</sub>  
OR NOT IR<sub>0</sub>  
OUT Y<sub>0</sub>



STR X<sub>0</sub>  
AND X<sub>1</sub>  
OUT NOT Y<sub>0</sub>



STR NOT Y<sub>0</sub>  
AND NOT X<sub>0</sub>  
AND X<sub>1</sub>  
OUT Y<sub>1</sub>

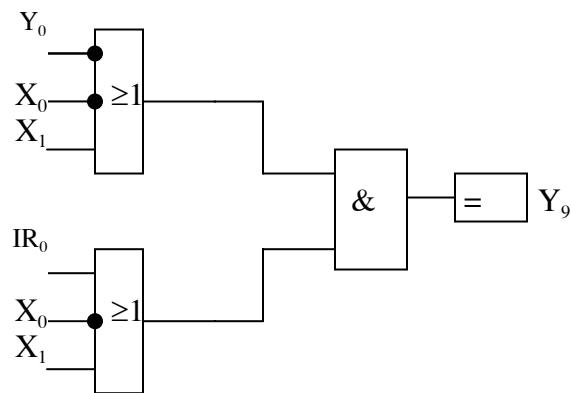


STR NOT Y<sub>0</sub>  
AND NOT X<sub>0</sub>  
AND X<sub>1</sub>

IR<sub>0</sub>

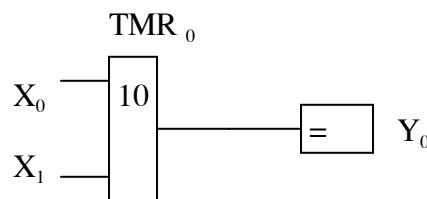
X<sub>0</sub>      &  
X<sub>1</sub>

STR IR<sub>0</sub>  
AND NOT X<sub>0</sub>  
AND X<sub>1</sub>  
OR STR  
OUT Y<sub>9</sub>

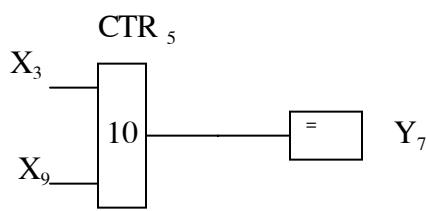


STR NOT Y<sub>0</sub>  
OR NOT X<sub>0</sub>  
OR X<sub>1</sub>

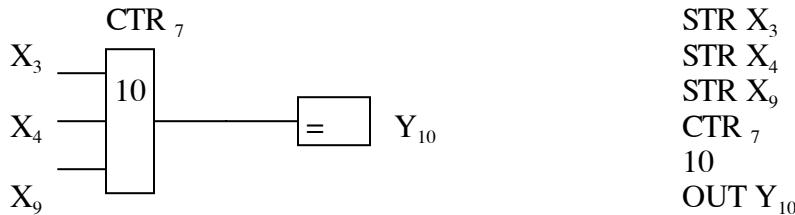
STR IR<sub>0</sub>  
OR NOT X<sub>0</sub>  
OR X<sub>1</sub>  
AND STR  
OUT Y<sub>9</sub>



STR X<sub>0</sub>  
STR X<sub>1</sub>  
TMR<sub>0</sub>  
OUT Y<sub>0</sub>



STR X<sub>3</sub>  
STR X<sub>9</sub>  
CTR<sub>5</sub>  
10  
OUT Y<sub>7</sub>

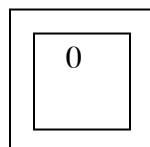


## NGÔN NGỮ GRAFCET

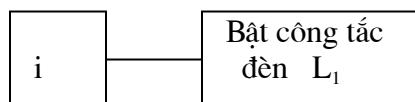
Đây là ngôn ngữ được phát triển bởi các nhà khoa học Pháp vào năm 1982. Ngôn ngữ này được đặt tên từ chữ viết tắt của Graphe Fonctionnel de Commande Etapes – Transition (Đồ họa chức năng điều khiển giai đoạn – chuyển tiếp) do hai cơ quan khoa học của Pháp là AFCET (Liên hiệp Pháp về kinh tế và kỹ thuật) và ADEPA (Hiệp hội về phát triển nền sản xuất tự động) hợp tác soạn thảo. Grafset là công cụ để biểu diễn các hoạt động điều khiển của các hệ thống tự động. Trong các hoạt động của các hệ thống vật lý thì ta có thể gặp hai dạng hoạt động: hoạt động kế tiếp nhau và hoạt động đồng thời. Mỗi hoạt động hay mỗi trạng thái được thể hiện bằng một khối hình vuông hoặc hình chữ nhật. Các hoạt động được liên kết với nhau bằng các đoạn thẳng, giữa các đoạn thẳng này được vạch bằng một gạch phân cách để chỉ trạng thái chuyển tiếp tức là kết thúc của một hoạt động và các điều kiện bắt đầu cho hoạt động tiếp theo. Đầu vào của một khối là điều kiện để bắt đầu của trạng thái hay của hoạt động kế tiếp và chính là kết thúc của một hay nhiều hoạt động trước đó.

Ngôn ngữ Grafset biểu diễn các hàm logic bằng sơ đồ của các chuỗi nhiệm vụ kế tiếp nhau. Ngôn ngữ này rất đơn giản cho người sử dụng, đặc biệt là đối với những người không am hiểu sâu về PLC. Grafset là công cụ được chuẩn hóa theo tiêu chuẩn của Pháp AFNOR. Từ Grafset có thể chuyển sang ngôn ngữ máy, ngôn ngữ bảng lệnh, sơ đồ thang hay sơ đồ hàm logic.

Mỗi trạng thái của hệ thống được ký hiệu bởi một hình vuông có chỉ số của trạng thái đó. Chuyển tiếp từ trạng thái này sang trạng thái kia được xác định bởi tác động được thực hiện tại trạng thái đứng trước.

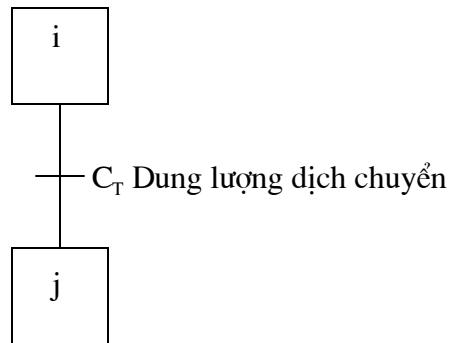


Trạng thái khởi động của Grafset



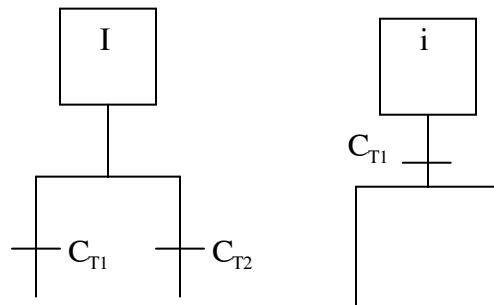
Ký hiệu sử dụng để diễn tả một hoạt động tại trạng thái đã biết.

Sự chuyển tiếp trạng thái của hoạt động thứ i sang hoạt động thứ j được thể hiện bằng điều kiện chuyển tiếp  $C_T$  (hình \*\*)



Sơ đồ dịch chuyển từ trạng thái i sang trạng thái j.

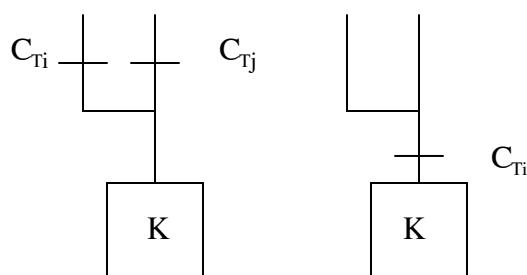
Có thể có khả năng chuyển từ một trạng thái sang hai trạng thái khác nhau:



Trường hợp thứ nhất là trường hợp phân nhánh đơn của chương trình, tương ứng với điều kiện  $C_{T1}$  hay điều kiện  $C_{T2}$ . Một trong hai điều kiện được đáp ứng, chương trình tiếp tục thực hiện các hoạt động tiếp theo. Trường hợp thứ hai là trường hợp phân nhánh song song. Hoạt động của các nhánh diễn ra đồng thời khi trạng thái chuyển tiếp  $C_T$  được đáp ứng.

Ngược lại cũng có khả năng từ hai hay nhiều trạng thái chuyển về một trạng thái:

Trường hợp thứ nhất là một nhánh nào đó đã kết thúc hoạt động cuối cùng của nhánh, thì chương trình sẽ tiếp tục ở bước tiếp theo.



Hình \*\*

## CHƯƠNG 2

### CÁC HỆ THỐNG ĐẦU VÀO VÀ ĐẦU RA

Hệ thống các đầu vào/ra cung cấp các kết nối vật lý giữa các thiết bị bên ngoài và bộ xử lý trung tâm CPU. Các mạch giao diện được sử dụng để chuyển đổi tín hiệu từ cảm biến hay độ lớn của các đại lượng đo được như tốc độ chuyển động, cao độ, nhiệt độ, áp suất và vị trí, thành các tín hiệu logic để PLC có thể sử dụng được. Dựa trên cơ sở của các giá trị cảm nhận được hay đo được, chương trình điều khiển trong PLC sẽ sử dụng các mạch điện ra khác nhau hoặc các mô đun ra để kích hoạt các thiết bị như bơm, van, động cơ, báo động để thực hiện điều khiển máy và quá trình. Các mạch vào I (Input) và các mạch ra O (Output) hoặc các mô đun được lắp ráp trong vỏ của thiết bị, trong trường hợp micro-PLC thì các kênh I/O là một phần của vỏ PLC. Phần lớn vỏ của các PLC có thể cài bất kỳ mô đun I/O vào một giắc cắm I/O nào đó. Vỏ của PLC được thiết kế để tháo các mô đun I/O mà không cần tắt nguồn xoay chiều AC (Alternative Current) hay tháo các dây nối. Đa số các mô đun I/O sử dụng công nghệ mạch in và các bảng mạch đều có giắc nối để cắm vào phích cắm ở bảng mạch chính trên giá đỡ. Bảng mạch chính cũng là mạch in có chứa các cổng giao tiếp song song hoặc các kênh truyền thông tin đến bộ xử lý. Nguồn điện một chiều DC (Direct Current) được cấp đến để kích hoạt mạch lô gíc và các mạch chuyển đổi tín hiệu trong các mô đun I/O.

#### 2.1 CÁC ĐẦU VÀO SỐ

Các kênh vào số thuộc nhóm lớn nhất của tín hiệu bên ngoài trong các hệ thống PLC. Thiết bị ngoại vi cung cấp tín hiệu vào số với hai giá trị khác hẳn nhau về bản chất, đặc trưng cho hai trạng thái đóng/mở, hay bật/tắt.

Các thiết bị gián đoạn thường xuất hiện phần lớn trong các ứng dụng điều khiển quá trình bao gồm:

- + Công tắc bánh gạt,
- + Công tắc nhiệt,
- + Công tắc lưu lượng,
- + Công tắc mức chất lỏng,
- + Công tắc vị trí của van,
- + Công tắc khởi động từ,
- + Công tắc xoay,
- + Nút bấm,
- + Công tắc vị trí,
- + Công tắc áp suất,
- + Công tắc cân gạt,
- + Công tắc tiệm cận,
- + Tiếp điểm rơ le,
- + Công tắc giới hạn,
- + Tiếp điểm khởi động động cơ,
- + Cảm biến quang điện.

Phần lớn các thiết bị này tạo ra dạng tín hiệu là đóng hoặc ngắt (ON hay OFF). Riêng cảm biến quang điện có thể có các tiếp điểm rơ le trên đầu ra hay tín hiệu điện áp ON/OFF tương ứng với mức 0 hay 5 VDC.

Nếu thiết bị gián đoạn được đóng, tức là điện áp được truyền qua thiết bị, trên mạch vào của PLC thu được tín hiệu điện áp cấp đến. Để chỉ thị trạng thái của thiết bị và chuyển đổi thành tín hiệu lô gíc, mạch lô gíc vào biến đổi tín hiệu về mức tương đương với điện áp mà CPU có thể xử lý được. Giá trị lô gíc 1 tương ứng với trạng thái bật (ON) hay đóng (CLOSED), và lô gíc 0 tương ứng trạng thái tắt (OFF) hay ngắt (OPENED).

## 2.2. CÁC ĐẦU RA SỐ

Điều khiển các đại lượng ra gián đoạn chỉ giới hạn trong các thiết bị có yêu cầu một trong hai trạng thái được chọn là ON/OFF, OPEN/CLOSED hay kéo /nén. Các thiết bị gián đoạn thường gặp trong quá trình điều khiển máy và quá trình công nghệ gồm:

- + Thiết bị truyền tín hiệu,
- + Báo động bằng tín hiệu ánh sáng,
- + Rơ le điều khiển bằng điện,
- + Quạt điện,
- + Đèn chỉ thị bằng tín hiệu ánh sáng,
- + Van điện,
- + Còi báo động,
- + Van con trượt,
- + Khởi động từ cho động cơ,
- + Rơ le nhiệt.

Trong lúc hoạt động, mạch giao diện trên đầu ra của PLC bật điện áp điều khiển để truyền đến thiết bị ra. Nếu tín hiệu ra được bật (ON) qua chương trình điều khiển, mạch giao diện sẽ để cho điện áp điều khiển kích hoạt thiết bị đầu ra.

## 2.3. DẠNG TÍN HIỆU VÀO/RA

Mỗi tín hiệu vào và ra đều được cấp năng lượng bởi một bảng nguồn cấp điện áp: +5 VDC, +24VDC,... Các mạch giao diện có thể cho ra các mức điện áp một chiều và xoay chiều khác nhau như:

- + 5 VDC
- + 12 VDC
- + 24 VDC
- + 48 VDC,
- 110 VAC hay DC,
- 220VAC hay DC,
- 100 VDC hoặc các tiếp điểm khô.

Phần lớn điện áp công nghiệp sử dụng là điện áp mức 110 VAC hay 220 VAC, bởi vì các nguồn có sẵn trong các nhà máy công nghiệp. Mặc dù vậy điện áp một chiều +5 VDC, +12 VDC, 24 VDC, + 48 VDC cũng được sử dụng rộng rãi bởi vì an toàn hơn và ít gây tai nạn như điện áp xoay chiều công nghiệp. Trong thiết kế mạch điều khiển cần phân biệt rõ mô đun vào và thiết bị trên đầu vào. Nếu thiết bị cung cấp dòng điện trong trạng thái bật (ON) của nó, thì thiết bị này được gọi là thiết bị cấp dòng. Ngược lại nếu thiết bị tiếp nhận dòng điện trong trạng thái bật (ON) hat trạng thái đúng (TRUE) thì thiết bị này được gọi là thiết bị tiêu thụ dòng. Như vậy chúng ta có thể có thiết bị cung cấp dòng và tiêu thụ dòng trên đầu vào, cũng tương tự như vậy là các mô đun vào cung cấp dòng và tiêu thụ dòng. Mặc dù vậy, phần các cấu hình sử dụng PLC đều dùng các thiết bị cấp nguồn trên đầu vào và các mô đun vào tiêu thụ dòng. Các vấn đề về giao diện tiềm ẩn sẽ tăng lên nếu ta không thiết kế hệ thống vào / ra thích hợp giữa các thao tác cấp nguồn và tiêu thụ trong các thiết bị của hệ thống. Ta phải sử dụng mô đun tiêu thụ dòng nếu thiết bị trên đầu vào

là thiết bị cấp dòng. Ngược lại nếu thiết bị đầu vào là thiết bị tiêu thụ dòng thì ta phải sử dụng mạch nguồn. Vấn đề trực tiếp sẽ xảy ra khi mô đun vào là mô đun tiêu thụ và các thiết bị đầu vào trừ một thiết bị là các đầu vào cấp nguồn. Thiết bị đầu vào tiêu thụ dòng có thể sẽ ở trạng thái bật (ON), nhưng mô đun vào không thể phát hiện tín hiệu ON, mặc dù điện áp có thể đo qua các đầu kết nối của mô đun vào. Đây chính là khả năng tiềm ẩn rằng thiết bị đầu vào không kết nối được và mạch vào của mô đun vào /ra có thể bị hỏng.

### **Mạch điện áp vào xoay chiều gián đoạn**

Một sơ đồ khối đặc trưng của nguồn điện áp vào xoay chiều như trên hình 2...

#### **Hình 2... Sơ đồ khối của mạch điện áp vào xoay chiều**

Các mạch vào của PLC cũng rất đa dạng và phụ thuộc vào các nhà sản xuất, nhưng nhìn chung thì các mạch vào gián đoạn hoạt động tương tự như hình 2..

Mạch điện áp vào cấu tạo bởi hai phần sơ cấp là phần nguồn và phần lô gíc. Phần nguồn và phần lô gíc của mạch thường kết nối với nhau sao cho phần mạch nguồn vào được cách điện với phần mạch lô gíc. Sự cách điện rất quan trọng đặc biệt trong môi trường nhiều công nghiệp. Vấn đề chính của các ứng dụng máy tính trong điều khiển quá trình ở thời kỳ ban đầu là chỗ các mạch vào và mạch ra không được thiết kế cho môi trường tồi tệ như môi trường công nghiệp với độ ẩm cao, rung, ôn bụi, nhiễu điện từ vv.

Trên hình 2..

#### **Hình 2.. Mạch điện áp vào gián đoạn đặc trưng**

Phần nguồn của mạch thực hiện chức năng biến điện áp vào 110 VAC hay 220 VAC từ các thiết bị ngoại vi đến mức tín hiệu lô gíc mà PLC có thể sử dụng được. Một cầu nắn dòng biến đổi tín hiệu vào thành tín hiệu một chiều. Tín hiệu mức một chiều này được truyền qua bộ lọc gồm một tụ điện dung C và các trở R<sub>2</sub>, R<sub>3</sub> để giảm thiểu biên độ sóng của mạch cầu. Mạch RC này có thể gây ra sự trễ 10 đến 25 mil giây. Mạch ra sử dụng di ốt Zener để phát hiện khi nào tín hiệu đến đạt ngưỡng điện áp điện áp vào . Nếu tín hiệu vào vượt quá và duy trì ở trên mức điện áp ngưỡng cho thời gian ít nhất bằng thời gian trễ của bộ lọc, tín hiệu được chấp nhận như tín hiệu vào đúng. Khi tín hiệu vào được phát hiện, nó đi đến mạch cách điện, mạch này chuyển hoàn toàn cách điện tín hiệu từ điện áp AC hay DC thành điện áp ở mức lô gíc. Mạch lô gíc sử dụng tín hiệu ở mức lô gíc từ bộ cách điện và tín hiệu được thực hiện sẵn sàng đưa đến bộ xử lý thông qua kênh dữ liệu trên mặt sau của giá đỡ PLC. Cách điện đạt đến 1500VAC khi không có kết nối nào giữa thiết bị ngoại vi (nguồn) với thiết bị điều khiển (lô gíc). Sự cách điện này giúp ngăn chặn các xung điện áp cao có thể làm hỏng mạch lô gíc của giao diện hay của thiết bị điều khiển. Một mạch cầu quang điện thường dùng để đảm bảo cầu nối giữa phần nguồn và phần lô gíc. Khả năng cách điện chính là một trong các lý do tại sao PLC giành được sự chấp nhận rộng rãi trong các quá trình công nghiệp.

Trong cách hệ thống PLC nhỏ , vừa và lớn, thì các mạch vào gián đoạn được lắp ráp cùng nhau trên một bảng mạch và lắp đặt trong mô đun vào. Các mô đun vào có thể có 4, 8, 16 hay 32 mạch vào trên một mô đun.

### **Các mô đun vào xoay chiều gián đoạn**

Phần lớn các mô đun xoay chiều gián đoạn đều có bộ chỉ thị tín hiệu để báo mức tín hiệu điện áp vào đã có, tức là công tắc được đóng. Bộ chỉ thị dùng di ốt LED thường được sử dụng để chỉ trạng thái của đầu vào. ánh sáng chỉ thị là sự trợ giúp quan trọng trong quá trình khởi động và khắc phục sự cố của hệ thống. Một sơ đồ nối đầu vào xoay chiều gián đoạn được minh họa trên hình 2..

## Hình 2.. Sơ đồ đấu dây đặc trưng của mô đun vào xoay chiều gián đoạn

Trên hình vẽ dây nóng 110VAC (L1) nối đến thiết bị và dây trung tính 110VAC (L2) nối đến đầu nối trung tính của mô đun vào. Ký hiệu ACI110 trên mô đun ở hình 2.. là số ký hiệu kiểu mô đun mà nhà sản xuất.

### Các mô đun vào một chiều (DC)

Các mô đun điện áp một chiều biến đổi trạng thái ON/OFF gián đoạn thành tín hiệu vào một chiều ở mức tín hiệu lô gíc tương thích với thiết bị điều khiển. Các mô đun này thường có ba mức điện áp: 12 VDC, 24 VDC và 48 VDC. Thiết bị tương thích với các mô đun này là công tắc, công tắc hành trình của van, nút ấn, công tắc tiệm cận một chiều, và cảm biến quang điện.

Sơ đồ đấu dây cho mô đun vào DC cũng tương tự như đối với mô đun vào AC, trừ điểm khác biệt là điện áp một chiều DC thay thế cho điện áp xoay chiều AC. Tín hiệu điện áp xoay chiều AC (dây nóng) đến các thiết bị đầu vào được thay thế bằng điện áp một chiều và đầu nối trung tính trên mô đun được thay thế bằng đầu nối mát một chiều chung.

### Các mô đun vào dạng TTL (Transistor – Transistor Logic)

Đây là các mô đun sử dụng mạch lô gíc tạo bởi các bộ transistor. Các mô đun vào TTL cho phép thiết bị điều khiển chấp nhận tín hiệu từ các thiết bị TTL tương thích, kể cả các điều khiển trạng thái cứng và thiết bị cảm biến. Đầu vào TTL được sử dụng để giao tiếp với các thiết bị điều khiển có mức điện áp +5 VDC và một số dạng cảm biến quang điện. Giao diện bởi mạch lô gíc TTL được thiết kế tương tự như các mô đun vào một chiều DC. Mặc dù vậy, thời gian trễ của tín hiệu vào gây ra bởi lọc nhiễu thường ngắn hơn nhiều. Các mô đun vào TTL thường yêu cầu nguồn cấp điện áp ngoài một chiều +5VDC.

### Các mô đun vào gián đoạn cách điện

Các mô đun đầu vào và đầu ra thường có dây trung tính chung nối mỗi nhóm đầu vào hay đầu ra trên mỗi mô đun. Mặc dù đôi khi chúng ta có thể nối thiết bị đầu vào có mức tiếp đất khác đến thiết bị điều khiển. Trong trường hợp như vậy, các mô đun vào cách điện (AC hay DC) với các đường tín hiệu trở về tách biệt khỏi mạch vào sẽ được dùng để nhận các tín hiệu dạng này. Giao diện cách điện và thiết bị vào ra gián đoạn tiêu chuẩn hoạt động giống nhau, trừ tiếp đất chung của các đầu vào được tách khỏi tiếp đất chung trong mô đun. Kết quả là mô đun vào cách điện yêu cầu số lượng đầu vào dây nhiều gấp đôi. Hậu quả là mô đun vào có thể tương thích với một nửa các đầu vào với cùng tính chất vật lý (xem hình 2...)

## Hình 2... Sơ đồ đấu dây của mô đun xoay chiều gián đoạn cách điện.

Số của mô đun cách điện với điện áp 110VAC có ký hiệu “IACI-110” và mô đun cách điện 220 VAC có ký hiệu “ IACI-220”.

### MẠCH RA GIÁN ĐOẠN XOAY CHIỀU AC

Hình 2.. là sơ đồ khôi đặc trưng cho mạch ra gián đoạn xoay chiều. Mạch ra dạng này rất khác nhau giữa các nhà sản xuất PLC. Sơ đồ khôi thể hiện các hoạt động cơ bản của ra xoay chiều . Mạch này gồm phần thứ cấp là phần lô gíc và phần nguồn, kết nối bởi mạch cách điện. Giao diện của đầu ra được thực hiện đơn giản thông qua công tắc, mà nguồn của nó cung cấp để điều khiển thiết bị đầu ra

## Hình 2.. Sơ đồ khối của mạch ra xoay chiều gián đoạn

Đầu tiên, bộ xử lý gửi tín hiệu ra 0 hay 1 đến phần mạch lô gíc. Tín hiệu từ phần lô gíc sâu đó đi qua một mạch cách điện. Tín hiệu lô gíc từ mạch cách điện được cấp tiếp vào mạch công tắc nguồn và bộ lọc. Cuối cùng tín hiệu xoay chiều gián đoạn này điều khiển một thiết bị hoạt động với điện áp xoay chiều AC được kết nối với điểm đầu ra của mô đun.

Phần công tắc nguồn xoay chiều thường sử dụng bộ Trisistor hay mạch nắn đồng bán dẫn SCR (Silicon Controlled Rectifier) để bật nguồn xoay chiều AC giữa hai trạng thái ON hoặc OFF. Công tắc AC thường được bảo vệ bởi mạch RC hay nhiệt điện trở MOV (Metal Oxide Variator), điều này cho phép chống lại sự tăng áp cao quá giá trị điện áp cho phép. Các mạch bảo vệ này hay các thiết bị cũng cho phép ngăn các nhiễu điện từ các hoạt động của các mô đun gây ra. Cầu chì có thể được cung cấp trên mạch ra để bảo vệ dòng quá lớn có thể làm hỏng công tắc xoay chiều AC. Nếu cầu chì không được cấp cho mỗi mạch trong mô đun thì chúng có thể được thêm vào ở bên ngoài của các mạch ra.

## Mô đun xoay chiều gián đoạn AC

Đối với các PLC nhỏ, vừa và lớn thì các mạch xoay chiều gián đoạn AC được lắp chung trên một bảng mạch đơn và cài đặt trong mô đun ra. Các mô đun ra thường có 4, 8, 16 hay 32 mạch ra trên một bo mạch. Cũng như các mô đun vào, các mô đun ra cũng có các diode quang LED để báo hiệu trạng thái của lô gíc đang hoạt động. Một sơ đồ kết nối của mô đun ra được minh họa trên hình 2...

Hình 2... Sơ đồ kết nối của mô đun ra gián đoạn xoay chiều.

Các di ốt LED được đặt ở phía trên của mô đun. Hai đầu ra đầu tiên được nối với hai bộ khởi động cho máy gia nhiệt 1 và 2. Tiếp điểm của rơ le chống quá tải OL được mắc nối tiếp để tắt bộ khởi động khi dòng tăng cao trong mạch khởi động. Hai đầu ra tiếp theo tại điểm ra 2 và 3 được nối với van trượt điện từ 110 VAC là LV-1 và LV-2. Bốn điểm ra còn lại được mắc nối tiếp với 4 bộ khởi động với rơ le chống quá tải cùng với cuộn khởi động. Ta cũng cần chú ý rằng điện áp đóng mạch là điện áp ngoài cấp đến mô đun, hay mô đun cũng là thiết bị cấp nguồn.

## Mô đun ra một chiều DC

Mô đun ra một chiều DC được sử dụng để cấp nguồn một chiều cho thiết bị ra. Chức năng hoạt động của đầu ra DC tương tự như đầu ra AC. Mạch công suất thường được sử dụng các transistor công suất để đóng tải. Giống như tyristor, transistor có khả năng phải chịu trạng thái quá áp và dòng khởi động lớn, điều này gây ra việc nung nóng và đoản mạch. Để tránh hiện tượng này ta phải bảo vệ các transistor công suất bằng các cầu chì.

Sơ đồ đấu dây cho mô đun một chiều cũng tương tự như sơ đồ đấu dây của mô đun xoay chiều, chỉ có khác là điện áp cung cấp là điện áp một chiều thay cho điện áp xoay chiều. Điểm nối của dây nóng xoay chiều được thay bằng điểm nối điện áp dương một chiều. Điểm nối dây AC trung tính được thay bằng tiếp đất hay điểm nối cực điện áp âm.

## Mô đun ra tiếp điểm khô

Mô đun ra các tiếp điểm khô cho phép các thiết bị đầu ra bật lên (ON) hay tắt (OFF) bằng tiếp điểm thường mở NO hay thường đóng NC. Ưu điểm của rơ le hay đầu ra là các công tắc khô là chúng cung cấp khả năng cách điện giữa PLC và thiết bị bên ngoài. Mạch đóng ngắt bằng thiết bị điện trạng thái cứng trong các mô đun ra xoay chiều tiêu chuẩn có sự dò điện với dòng rất nhỏ ngay cả khi mạch đóng được chuyển về trạng thái ngắt. Dòng

điện này có thể gây ra tín hiệu giả trong nhiều trường hợp. Trong các ứng dụng như vậy, ta cần sử dụng mô đun ra với tiếp điểm khô.

Mô đun ra tiếp điểm khô được sử dụng để đóng tải xoay chiều AC hay một chiều DC. Mặc dù vậy, chúng thường được sử dụng trong các ứng dụng với điện áp xoay chiều để cung cấp khả năng cách điện giữa PLC và các thiết bị điện phức tạp khác, như bộ điều tốc VSD (Variable Speed Drives). Hình 2... là mô đun ra tiếp điểm khô với bốn tiếp điểm thường mở NO điều khiển khởi động và tắt hai bộ điều khiển tốc độ động cơ. Trong ứng dụng này, đây là sự cách điện hoàn hảo giữa PLC và VSD.

### **Mô đun ra TTL**

Mô đun ra TTL cho phép thiết bị điều khiển tác động lên thiết bị đầu ra tương thích với TTL như màn hình số 7 đoạn, mạch tích hợp và các thiết bị lò gác cơ sở khác nhau với điện áp +5VDC. Các mô đun này thường yêu cầu nguồn điện áp ngoài +5 VDC với dòng điện yêu cầu đặc biệt.

### **Mô đun ra cách điện xoay chiều**

Giao diện cách điện của đầu ra AC được minh họa trên hình 2..

#### **Hình 2.. Sơ đồ đấu dây của mô đun ra xoay chiều cách điện đặc trưng**

Ta có thể thấy rằng mô đun đầu ra điều khiển ba tải khác nhau ( ba bộ khởi động cho ba bơm khác nhau), chúng được nối tới ba nguồn xoay chiều khác nhau. Ưu điểm của mô đun này là chúng ta có thể không phải bận tâm vì có các nguồn điện áp khác nhau trong nhà máy của chúng ta. Điều bất lợi là số lượng dây đấu tăng lên và giảm số đầu vào có thể của mỗi mô đun bởi cơ số 2. Trong ứng dụng trên hình 2.. ba nguồn điện áp 110 VAC khác nhau được sử dụng để bật ba bộ khởi động động cơ của ba bơm 1, 2 và 3. Đây là ứng dụng đặc trưng cho mô đun ra xoay chiều cách điện AC.

### **Mô đun tương tự vào/ra (I/O)**

Sự hiện diện của các mạch tích hợp giá thành rẻ và các mạch điện tử công nghiệp đã làm tăng các khả năng của các mạch tương tự trong các thiết bị điều khiển PLC. Khả năng mở rộng này đưa đến sự ra đời của các mô đun vào/ra tương tự tinh vi.

Các mô đun tương tự cho phép đo số lượng thu được từ các cảm biến của quá trình và các thiết bị cung cấp dữ liệu tương tự. Các mô đun đầu ra tương tự cho phép điều khiển các thiết bị với tín hiệu tương tự liên tục. Các đầu vào /ra tương tự cho phép theo dõi và điều khiển các điện áp và dòng điện tương tự, tương thích với các cảm biến, các bộ điều khiển động cơ, và các thiết bị quá trình. Sử dụng đầu vào/ra tương tự và chuyên dụng cho phép đo hay điều khiển phần lớn các đại lượng của quá trình công nghiệp dài như giao diện tương ứng sử dụng.

Thiết bị đầu vào tương tự:

- Cảm biến lưu lượng
- Cảm biến áp suất
- Cảm biến nhiệt
- Cảm biến phân tích
- Cảm biến vị trí
- Biến trở
- Cảm biến mực chất lỏng
- Thiết bị đo tốc độ

Thiết bị đầu ra tương tự:

- Thiết bị điều khiển động cơ,
  - Thiết bị đo tương tự,
  - Thiết bị ghi đồ hoạ,
  - Thiết bị điều khiển quá trình
  - Dòng điều khiển bộ chuyển đổi khí nén
  - Van điều khiển bằng điện
  - Bộ điều khiển tốc độ.
- 

### **Mô đun vào tương tự**

Giao diện của mô đun vào tương tự chứa các mạch cần thiết để có thể nhận tín hiệu điện áp hay dòng điện tương tự từ các thiết bị bên ngoài. Đầu vào điện áp hay dòng điện được biến đổi từ tín hiệu tương tự thành các giá trị số tỉ lệ với tín hiệu tương tự nhờ có bộ chuyển đổi tín hiệu ADC (Analog to Digital Converter). Giá trị chuyển đổi đi qua kênh dữ liệu của thiết bị điều khiển và lưu trong bộ nhớ để sử dụng về sau.

Giao diện vào tương tự có đặc trưng là có trở kháng vào rất cao, điều này cho phép chúng giao diện với thiết bị bên ngoài không cần tải tín hiệu. Đường vào từ các thiết bị tương tự thường được bọc chống nhiễu bằng hai lớp dẫn điện. Cáp chống nhiễu giảm ảnh hưởng của nhiễu từ các nguồn bên ngoài đi rất nhiều. Giao diện của tầng đầu vào cung cấp mạch lọc và mạch cách điện để bảo vệ mô đun từ các trường nhiễu phụ. Một sơ đồ kết nối đặc trưng minh họa trên hình 2.... Trong ví dụ này, mô đun vào tương tự cung cấp nguồn điện áp một chiều DC yêu cầu bởi các thiết bị biến đổi dòng bên ngoài.

Phân lớn các mô đun được thiết kế để thu nhận đến 16 tín hiệu đơn cực hay 8 tín hiệu tương tự lưỡng cực, thể hiện lưu lượng, áp suất, mức và tương tự. Chúng sau đó được chuyển đổi thành các từ tỉ lệ với 10 đến 15 bit nhị phân trong bộ nhớ. Đầu vào đến các mô đun riêng biệt nói chung phải là tất cả là đơn cực hoặc lưỡng cực. Chọn dạng tín hiệu có thể thực hiện bằng phần cứng hay phần mềm. Nếu tín hiệu qua chuyển đổi lưu trong bộ nhớ của mô đun và được gửi đến bộ nhớ của vi xử lý trong nhóm hay trong khối các dữ liệu.

Chương trình điều khiển sử dụng cấu trúc dữ liệu để truyền đến mô đun tương tự. Thông tin về cấu trúc bao gồm lựa chọn miền ví dụ +1 đến +5 VDC, 4 đến 20mA vv. Và hệ số tỉ lệ của tín hiệu.

Hình 2..

### **Mô đun ra tương tự**

Mô đun ra tương tự nhận dữ liệu từ bộ xử lý trung tâm của PLC. Dữ liệu được truyền tỉ lệ với điện áp hay dòng điện để điều khiển thiết bị tương tự bên ngoài. Dữ liệu số đi qua bộ chuyển đổi tín hiệu DAC và gửi đi dưới dạng tương tự. Cách điện giữa mạch ra và mạch lõi gác được đảm bảo bởi cầu quang điện. Các mô đun này thường cần nguồn cấp ngoài với dòng điện xác định và điện áp theo yêu cầu.

### **Mô đun chuyên dụng**

Rất nhiều loại mô đun chuyên dụng được sử dụng trong các hệ PLC. Một nhà sản xuất PLC có trên 120 dạng mô đun vào ra. Chúng ta chỉ quan tâm đến hai trong các mô đun đó là: mô đun nối với encoder và bộ đếm và mô đun xung vào tốc độ cao.

### **Mô đun vào nối với encoder và bộ đếm**

Mô đun này cung cấp một bộ đếm tốc độ cao từ bên ngoài đến bộ xử lý, sao cho đáp ứng tới các xung đầu vào ghi nhận được trong giao diện. Bộ đếm này thường hoạt động độc lập ngoài chương trình quét hay quét đầu vào/ra. Lý do có vẻ đơn giản nếu bộ đếm phụ thuộc vào chương trình PLC thì các xung tốc độ cao sẽ không đếm được hay bị mất trong quá trình quét. Ứng dụng tiêu biểu là giao diện encoder/ bộ đếm là các hoạt động yêu cầu trực tiếp các đầu vào từ encoder và có khả năng cung cấp trực tiếp sự so sánh của các đầu ra.

Mô đun này nhận các xung vào từ bộ encoder gia tăng. Các xung này chỉ vị trí khi thiết bị quay. Bộ đếm xung gửi chúng tới bộ xử lý. Bộ encoder tuyệt đối thường sử dụng void giao diện sao cho nhận được dữ liệu dạng mã BCD hay mã Gray, thể hiện vị trí góc của trục cơ đang được đo.

Trong quá trình hoạt động, các mô đun này thu được các xung vào, các xung này được đếm và so sánh với giá trị được người vận hành lựa chọn. Bộ đếm của mô đun vào thường có tín hiệu ra là tín hiệu kích hoạt khi đầu vào và giá trị ngưỡng đếm bằng nhau. Mặc dù vậy, điều này không cần thiết trong phần lớn các PLC. Bởi dữ liệu có trong CPU, chương trình có thể sử dụng hàm so sánh để điều khiển một đầu ra nào đó trong chương trình điều khiển.

Truyền dữ liệu giữa giao diện của encoder /bộ đếm với CPU là hai chiều. Mô đun này chấp nhận đặt giá trị ngưỡng đếm và các dữ liệu điều khiển khác từ CPU và truyền dữ liệu và trạng thái đến bộ nhớ của PLC. Đầu ra điều khiển cho phép từ chương trình điều khiển, sao cho lệnh đến mô đun phải làm hoạt động các đầu ra tương ứng với giá trị đếm nhận được. CPU sử dụng chương trình điều khiển, fcho phép và đặt các hoạt động của bộ đếm.

### **Mô đun đếm xung vào**

Bộ đếm xung vào được dùng để giao tiếp với thiết bị bên ngoài mà chúng tạo ra các xung, như chuyển động theo các bộ đếm lưu lượng chuyển động theo chiều dương và bộ đếm lưu lượng dạng tua bin. Trong ứng dụng đặc trưng, bộ đếm lưu lượng phát ra các xung với biên độ +5VDC phụ thuộc vào thể tích của chất lỏng đi qua. Mỗi xung thể hiện một thể tích cố định, ví dụ một xung có thể tương đương 1 lít chất lỏng. Trong ví dụ trên, bộ đếm của PLC đếm số xung nhận được bằng mô đun xung vào sau đó tính toán thể tích chất lỏng đi qua trong thời gian chu kỳ cố định.

### **Mô đun vào/ra thông minh**

Để xử lý tốt một số dạng tín hiệu hay dữ liệu, cần có các mô đun cấu tạo từ các bộ vi xử lý. Các giao diện thông minh này xử lý các tín hiệu vào giống như các mô đun nối với can nhiệt hay các tín hiệu khác không thể giao diện được bằng các mô đun vào /ra tiêu chuẩn. Mô đun thông minh có thể thực hiện hoàn chỉnh các chức năng xử lý tín hiệu, độc lập với CPU và chu trình quét của chương trình điều khiển. Trong phần này ta sẽ trình bày về hai trong số các mô đun thông minh hay sử dụng nhất: mô đun vào của các can nhiệt và mô đun ra với động cơ bước.

### **Mô đun vào nối với các can nhiệt**

Một mô đun vào của các can nhiệt được thiết kế để nhận trực tiếp các đầu vào từ can nhiệt như trên hình 2..

Hình 2.. Mô đun vào nối với can nhiệt

Mô đun này tạo khả năng hiệu chỉnh nhiệt độ của môi trường để bù những thay đổi của nhiệt độ môi trường xung quanh mô đun can nhiệt. Mô đun này hoạt động như

một mô đun vào tiêu chuẩn, chỉ có khác là nó thu nhận các đầu vào có mức tín hiệu thấp cõi mi li vôn. Các tín hiệu vào sẽ được lọc, khuyếch đại, và số hoá qua bộ chuyển đổi tín hiệu tương tự – số ADC. Các tín hiệu này sau đó được gửi đến bộ vi xử lý có trong mô đun để tuyến tính hoá và chuyển thành giá trị nhiệt độ. Cuối cùng thì giá trị nhiệt độ sẽ được gửi về CPU theo lệnh từ chương trình điều khiển. Dữ liệu nhiệt độ được sử dụng bởi chương trình điều khiển PLC để thực hiện quá trình điều khiển nhiệt hay chỉ thị nhiệt độ.

### Mô đun động cơ bước

Mô đun động cơ bước tạo ra các xung kéo tương thích với bộ điều khiển của động cơ bước. Các xung được gửi đến bộ điều khiển thường thể hiện dưới dạng khoảng cách, tốc độ, và hướng để điều khiển động cơ. Giao diện của động cơ bước nhận các tín hiệu điều khiển từ chương trình điều khiển. Vị trí xác định bởi số lượng định trước các xung ra bằng lệnh điều khiển tiến hay lệnh điều khiển lùi, bằng tăng tốc hay giảm tốc với điều khiển bằng hàm tăng, tức là xác định bởi tốc độ của các xung ra. Các điều khiển này nhìn chung là các điều khiển chuyên dụng trong chương trình điều khiển và một khi giao diện ra được khởi tạo bởi lệnh khởi động, nó sẽ phát ra các xung theo chương trình PLC. Khi chuyển động bắt đầu, mô đun ra sẽ không tiếp nhận một điều khiển nào từ CPU cho đến khi chuyển động được thực hiện xong. Một số mô đun có có thể có các lệnh để huỷ lệnh điều khiển này và đặt lại vị trí tức thời. Lệnh này phải được huỷ bỏ khi tiếp tục thực hiện lệnh điều khiển chuyển động của động cơ. Mô đun này cũng gửi dữ liệu theo trạng thái của bộ xử lý của PLC.

Hình 2.. Sơ đồ nối của mô đun ra điều khiển động cơ bước

## MÔ ĐUN TRUYỀN THÔNG

Có sáu dạng mô đun truyền thông được sử dụng trong hệ thống PLC để trao đổi giữa các phần tử của hệ thống. Đây là các mô đun dạng mã ASCII, mô đun nối vào/ra điều khiển từ xa vạn năng, thẻ giao diện PCMCIA, mô đun giao diện Ethernet, và mô đun biến đổi tín hiệu sợi cáp quang.

### Mô đun truyền thông mã ASCII

Mô đun truyền thông ASCII được sử dụng để truyền và thu dữ liệu dạng ký tự và số giữa các thiết bị ngoại vi và thiết bị điều khiển. Thiết bị ngoại vi đặc trưng là máy in, màn hình số, thiết bị lập trình vv. Mô đun này cũng tuỳ thuộc vào nhà sản xuất. Mạch giao diện truyền thông thường bao gồm bộ nhớ và một vi xử lý dành riêng. Giao diện của các thông tin được trao đổi thường chiếm chỗ qua cổng nối tiếp RS-232C, RS-422 hay RS-485 hay qua đường truyền mạch vòng với dòng 20mA.

Mô đun ASCII sẽ có bộ nhớ RAM riêng, để lưu trữ các khôi thông tin chuẩn bị được truyền đi. Khi dữ liệu vào từ thiết bị ngoại vi được nhận trong mô đun, nó truyền đến bộ nhớ của PLC thông qua lệnh truyền dữ liệu. Tất cả các thông số của khôi thảo truyền tin như bit chẵn (chẵn hay lẻ) hay không chẵn, số của bit dừng (stop bit), tốc độ truyền, có thể được chọn bằng phần mềm hay phần cứng.

### Mô đun kết nối vạn năng I/O

Mô đun kết nối vào/ra từ xa được sử dụng trong các hệ thống PLC cỡ lớn, cho phép các hệ thống vào /ra con đặt cách xa bộ xử lý. Các hệ thống con ở xa được sử dụng để giao diện với đơn vị xử lý qua một giá đỡ vào/ra tiêu chuẩn và các mô đun vào/ra yêu cầu. Các giá đỡ bao gồm cả nguồn một chiều để nuôi các mạch bên trong của các mô đun vào/ra và mô đun kết nối vào/ ra từ xa để cấp thông tin với đơn vị xử lý. Dung lượng của một hệ vào /ra con thường có từ 32 đến 256 đầu vào/ra.

Các hệ vào/ra con thường được nối vào bộ xử lý thông qua đường truyền kênh (bus) hay mạng hình sao. Cự ly từ bộ xử lý đến các hệ thống vào /ra con từ xa thông thường vào khoảng 300m đến vài km, phụ thuộc vào dạng thiết bị điều khiển.

Sự bố trí các hệ thống vào /ra từ xa tạo ra khả năng tiết kiệm khá lớn về dây dẫn, nhân công cho các hệ thống điều khiển cỡ lớn, mà trong đó các thiết bị được gộp vào một số khu vực xử lý từ xa. Nếu bộ xử lý được đặt trong phòng điều khiển chính hay ở một số vị trí trung tâm khác, chỉ cần một cáp thông tin chạy giữa bộ xử lý và các hệ thống từ xa ngoài hiện trường thay vì hàng trăm, thậm chí hàng ngàn đường dây nối từ bộ xử lý đến các thiết bị hiện trường.

Bố trí các hệ thống vào/ra từ xa còn có ưu điểm là cho phép các hệ thống con lắp đặt và thử nghiệm độc lập cũng như cho phép bảo dưỡng và khắc phục sự cố trên từng trạm trong khi các bộ phận khác vẫn tiếp tục hoạt động.

### **Mô đun truyền thông nối tiếp**

Mô đun truyền dữ liệu nối tiếp thường được sử dụng để truyền thông tin giữa thiết bị điều khiển và một thiết bị thông minh với đầu ra nối tiếp như cân khối lượng với cổng nối tiếp. Mô đun truyền thông nối tiếp này thường có hai đến bốn cổng nối tiếp để có thể kết nối với các cổng giao diện nối tiếp tiêu chuẩn như RS-232, RS-422, RS-485.

### **Thẻ giao diện PCMCIA**

Vào năm 1990, hiệp hội thẻ nhớ quốc tế cho máy tính cá nhân (Personal Computer Memory Card International Association) phát triển một tiêu chuẩn cho thẻ tín dụng cỡ thẻ giao diện cho máy tính cá nhân. Tiêu chuẩn này định nghĩa cấu trúc và phương pháp truyền thông cho các thẻ giao diện PC. Các thẻ giao diện này được phát triển theo điều khoản 2.0 của tiêu chuẩn và được sử dụng để lưu trữ dữ liệu và trao đổi thông tin vào/ra. Các nhà sản xuất PLC phát triển thẻ PCMCIA để máy tính sách tay có thể trao đổi được với bộ xử lý của PLC hay đường truyền dữ liệu tốc độ cao để thực hiện phần mềm PLC hay các hàm sự cố.

Các thẻ giao diện PCMCIA giống như với phần mềm chuẩn đoán để kiểm tra rằng thẻ hoạt động tốt và để kết nối nó với mạng truyền thông của PLC.

### **Mô đun truyền thông Ethernet**

Mô đun giao diện Ethernet được thiết kế để cho phép một số PLC và các máy tính điều khiển có thể trao đổi thông tin tốc độ cao trên mạng thông tin tốc độ cao của nhà máy. Mạng nội bộ của nhà máy LAN (Local Area Network) có khả năng truyền dữ liệu và thông tin điều khiển từ một hệ thống đến hệ thống khác với tốc độ truyền tin cao. Như vậy điều khiển hệ thống công nghiệp có thể pân bố thành một ssô lớn các thiết bị điều khiển PLC, máy tính, và các thiết bị thông minh. Trong một hệ thống như vậy, thông tin được trao đổi dễ dàng giữa hệ thống điều khiển, nhưng mỗi hệ thống có thể điều khiển độc lập một phần của nhà máy công nghiệp. Điều này cải thiện rất lớn độ tin cậy của hệ thống điều khiển nhà máy bởi vì từng phần của nhà máy có thể dừng để thay đổi hoặc bảo dưỡng, trong khi các phần khác của nhà máy tiếp tục hoạt động và sản xuất.

### **Mô đun biến đổi tín hiệu từ cáp quang**

Bộ chuyển đổi tín hiệu từ cáp quang biến tín hiệu điện thành tín hiệu ánh sáng và truyền các tín hiệu này qua cáp quang. Trên đầu kia của cáp quang, một sợi cáp quang thứ hai biến tín hiệu ánh sáng thành tín hiệu điện để sử dụng bởi hệ thống PLC.

## **THIẾT KẾ HỆ THỐNG VÀO/RA**

Để thiết kế hệ thống vao/ra đúng, các tiêu chí kỹ thuật của nhà sản xuất phải được quan tâm và tuân theo để tránh thao tác sai hay làm hỏng thiết bị. Các tiêu chí kỹ thuật này đặt các giới hạn không chỉ trên mô đun mà trên cả thiết bị hiện trường mà nó điều khiển. Các tiêu chí này có ba dạng: các tiêu chí về điện, cơ và môi trường.

#### **Tiêu chí kỹ thuật điện:**

Các tiêu chí kỹ thuật điện bao gồm các thông số sau: mức điện áp vào, mức dòng vào, điện áp nguồn, mức điện áp ra, mức dòng ra, mức năng lượng ra và các yêu cầu về dòng cấp vào phía sau để đảm bảo cho mạch của mô đun có thể hoạt động được.

Mức điện áp vào (xoay chiều hay một chiều) cho ta biên độ và dạng tín hiệu vào mà mô đun vào chấp nhận. Trong một số trường hợp, các tiêu chí này xác định miền của điện áp vào thay vì giá trị cố định. Trường hợp như vậy, giá trị max và min chấp nhận được của điện áp để tiếp tục hoạt động được liệt kê. Ví dụ điện áp làm việc 110 VAC cho mô đun vào có thể được chấp nhận từ 95 đến 135 VAC.

Mức dòng vào định nghĩa dòng tối thiểu yêu cầu tại mức điện áp của mô đun mà thiết bị hiện trường phải có khả năng cung cấp để làm hoạt động mạch của mô đun vào.

Nguồn điện áp vào là điện áp mà tín hiệu vào được nhận là đang ở trạng thái bật (ON) hay đúng (TRUE). Một số mô đun vào cũng có giá trị điện áp của trạng thái OFF hay FALSE. Ví dụ tín hiệu ON trên mô đun TTL được xác định là 2.8VDC và mức OFF là điện áp thấp hơn 0.8 VDC.

Mức điện áp ra chỉ biên độ và dạng điện áp nguồn phải được điều khiển mà không có dung sai được công nhận. Ví dụ trên mô đun ra tại mức +24 VDC, thì có thể có miền làm việc từ +20 đến +28 VDC.

Mức dòng ra định nghĩa dòng lớn nhất mà mạch ra trên mô đun ta có thể đảm bảo an toàn khi có tải. Mức dòng này thường được chỉ định như một hàm của mạch ra của các phần tử điện và đặc tính tỏa nhiệt tại môi trường làm việc trong khoảng từ 0oC- 60oC. Nếu nhiệt độ môi trường tăng, dòng điện ra sẽ bị giảm. Dòng lớn quá có thể làm đoản mạch hay gây ra các hư hại khác cho mô đun ra.

Mức năng lượng ra chỉ mức năng lượng lớn nhất mà mô đun ra có thể tiêu thụ với tất cả các kênh ra được kích hoạt. Mức năng lượng cho một đầu ra được tính bằng nhân điện áp ra với mức dòng điện ra.

Các yêu cầu về dòng cấp vào mặt sau là dòng yêu cầu để mạch bên trong mô đun vào/ra hoạt động đảm bảo, đặt sau của giá đỡ nguồn điện. Nhà thiết kế hệ thống phải thêm yêu cầu dòng mặt sau trên tất cả các mô đun được cài đặt vào khung giá đỡ vào/ra và so sánh giá trị tính toán với dòng cực đại mà hệ thống công suất nguồn có thể cấp để xác định rằng công suất được cấp đủ hay không. Nếu mức công suất thấp hơn yêu cầu, hệ thống sẽ hoạt động chập chờn. Dòng khứ hồi yêu cầu thường phải nhỏ hơn dòng cấp bởi nguồn công suất.

#### **Tiêu chí cơ học**

Tiêu chí này xác định số lượng điểm hiện trường được điều khiển hay được theo dõi bởi mô đun. Các mô đun thường có 2,4,8,16 hay 32 điểm vao/ra. Các mô đun mật độ cao, yêu cầu dòng hoạt động cao hơn, và phải kiểm tra rất cẩn thận dòng. Một khía cạnh khác là vấn đề phải tính toán để tránh gây ra đoản mạch.

#### **Tiêu chí môi trường**

Có hai thông số môi trường ảnh hưởng đến hoạt động của hệ thống PLC là nhiệt độ và độ ẩm. Nhiệt độ môi trường là nhiệt độ cao nhất trong không khí xung quanh các mô đun vao/ra đang hoạt động bình thường. Mức nhiệt độ này thường dựa trên cơ sở đặc tính tản nhiệt của các mô đun. Nhiệt độ môi trường thường trong khoảng 0oC đến 40oC.

Nhiệt độ môi trường cao quá có thể gây nguy hiểm vì các mạch bên trong mô đun có thể hoạt động kém tin cậy và giảm tuổi thọ của mô đun.

Độ ẩm thường từ 5% đến 95% là khoảng chưa ngưng tụ nước. Người thiết kế hệ thống phải đảm bảo rằng độ ẩm được điều khiển chuẩn xác trên bàn điều khiển nơi mà hệ thống vào ra được lắp đặt.

## CHƯƠNG 3 HOẠT ĐỘNG CỦA BỘ NHỚ VÀ CÁCH GHI ĐỊA CHỈ

### Mở đầu

Các hệ thống điều khiển PLC lưu trữ thông tin và chương trình điều khiển trong bộ nhớ. Thông tin được lưu trữ ở đây xác định PLC sẽ xử lý các dữ liệu vào ra như thế nào. Trong chương này, chúng ta sẽ trình bày về thành phần và cấu trúc của bộ nhớ, dạng bộ nhớ, tổ chức của bộ nhớ, cách ghi địa chỉ của bộ nhớ và địa chỉ đầu vào/ra. Cuối cùng ta sẽ tập trung vào giao diện từ phần cứng đến phần mềm của PLC.

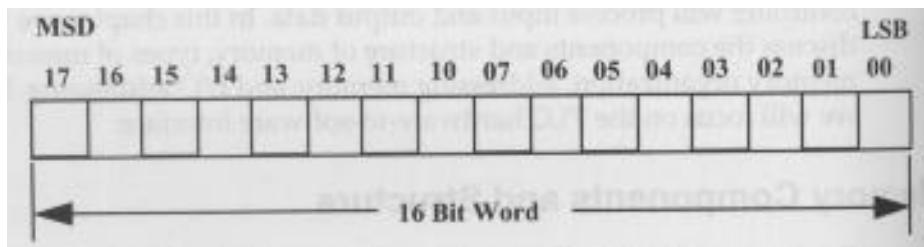
### 3.1 CÁC THÀNH PHẦN VÀ CẤU TRÚC CỦA BỘ NHỚ

Bộ nhớ của PLC có thể được hiển thị bằng một mảng hai chiều của các tế bào nhớ, mà mỗi tế bào có thể chứa một bit đơn của thông tin dưới dạng là giá trị 0 hay 1. Số nhị phân đơn hay “bit” được lấy từ hai chữ đầu của từ *Binary* và chữ cuối của *digit*. Bit là đơn vị nhỏ nhất trong bộ nhớ và chỉ chứa được thông tin dạng 1 và 0. Mỗi tế bào nhớ có điện áp thể hiện trên đầu ra của mạch điện tử chỉ thị giá trị “1” và điện áp 0 chỉ giá trị “0”.

Bit được bật lên trạng thái ON nếu giá trị lưu trên đó là 1 và OFF nếu là lưu giá trị 0. Trong nhiều trường hợp, điều quan trọng đối với bộ xử lý là giữ được nhiều dữ liệu hơn là một bit đơn. Ví dụ khi bộ xử lý truyền dữ liệu đi và về bộ nhớ, lưu trữ các số, các mã chương trình, nó cần một nhóm các bit gọi là “byte” hay từ (*word*). Một byte được định nghĩa là một nhóm nhỏ nhất của các bit mà CPU có thể xử lý được đồng thời trong một lần. Trong các thiết bị điều khiển PLC byte có cỡ thông thường là 8 bit và một từ có cỡ là hai byte hay 16 bit. Mặc dù vậy, một từ có thể có độ lớn lớn hơn hay nhỏ hơn, phụ thuộc vào đặc tính của bộ vi xử lý được sử dụng.

Lượng bộ nhớ được tính bằng hàng 1000 hay K “K=kilo”, với 1 K là 1024 từ (tức là  $2^{10}=1024$ ) của không gian nhớ. Dung lượng bộ nhớ của PLC có thể thay đổi từ nhỏ hơn 1K đến 64K từ. Phụ thuộc vào nhà sản xuất. Tính phức tạp của kế hoạch điều khiển, số lượng của các điểm vào/ra và dạng vào ra là những thông số xác định độ lớn của bộ nhớ PLC.

Độ dài của từ thường là hai byte (16 bit) hay có thể dài hơn. Chiều dài của từ trong PLC có thể là 4, 8, 16 hay 32 bit. Từ có 16 bit thể hiện trên hình3..



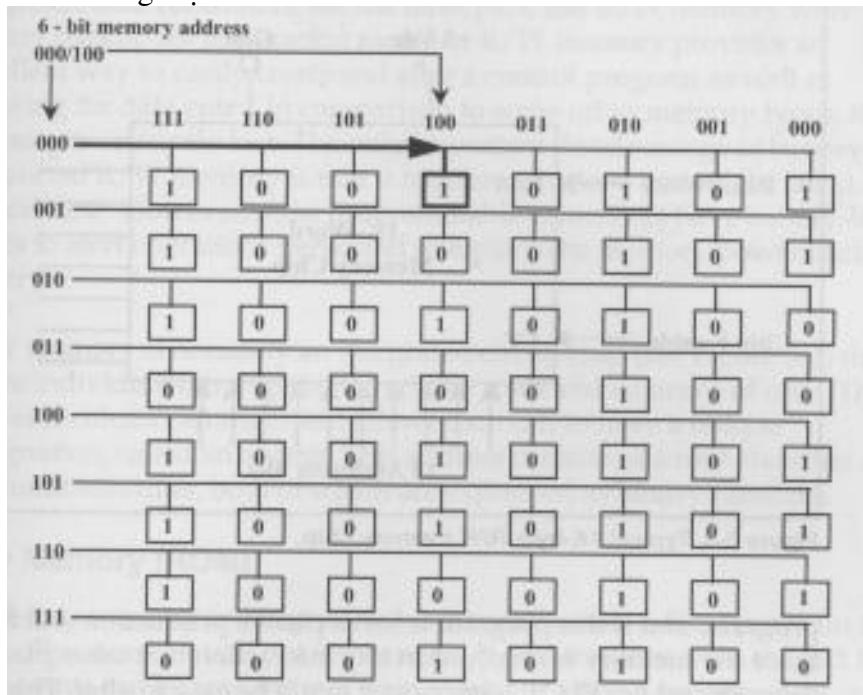
Hình 3.. Từ mươi sáu bit

Một số PLC sử dụng hệ đếm 16 để nhận dạng mỗi bit, như hình 3... Bit có trọng lượng lớn nhất là bit 17 và bit nhẹ nhất là bit 00.

Một mảng nhớ đơn 64 bit được minh họa trên hình 2..

Mảng này gồm 8 hàng và tám cột. Mảng 64 bit cần có 6 bit địa chỉ cấp cho mỗi tế bào. Một tế bào thường là một mạch điện tử được gọi là mạch flip-flop, mạch này có giá trị +5VDC hay 0 VDC. Để lấy dữ liệu từ mảng nhớ, các bộ giải mã địa chỉ dòng và cột sẽ chọn ra tế bào yêu cầu.

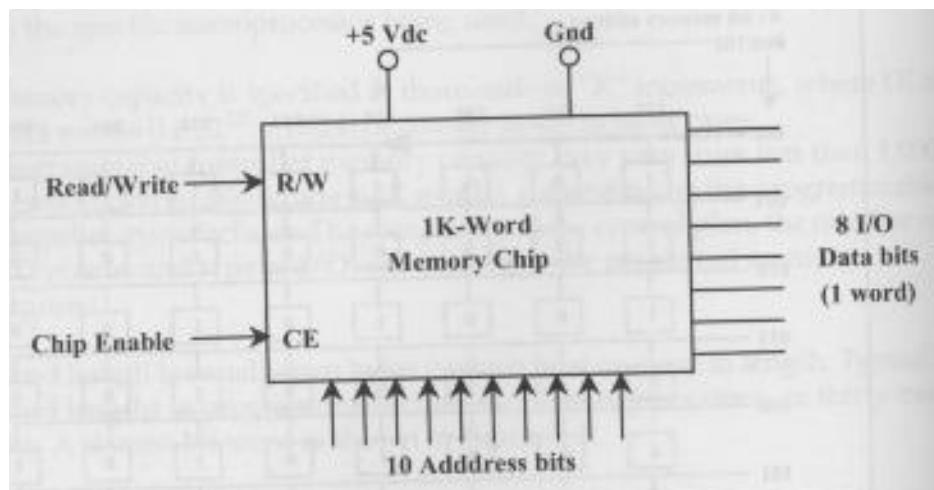
Các mảng nhớ thường được tạo bởi các mạch tích hợp (IC). Một đơn vị đặc trưng của mạch tích hợp chứa nhiều ngàn tế bào nhớ sắp xếp theo cách khác nhau. Một mạch tích hợp 8K-bit (8096 bit) có thể bố trí thành 8K tế bào nhớ với một bit mỗi tế bào, hay 1K byte với 8 tế bào mỗi byte. Số của nhóm (bit, byte hay từ) được ghi địa chỉ là hàm của  $2^n$ , ví dụ  $1K=2^{10}$ ,  $4K=2^{12}$ ,  $8K=2^{13}$  và tương tự. Giá trị n là số bit địa chỉ cần chọn cho mỗi nhóm riêng biệt.



Hình .. Mảng nhớ đặc trưng

Với 1000 từ ta cần sử dụng mươi bit để ghi địa chỉ của mỗi từ trong nhớ, với các từ cỡ tám bit, mươi sáu bit hay ba mươi hai bit. Đối với bộ nhớ 1Kx8, mạch IC cần 10 bit địa chỉ để chọn 1 K từ trong bộ nhớ. Mạch IC có 8 chân cho dữ liệu đầu vào và đầu ra, 10 chân cho chọn địa chỉ, hai chân cho nguồn một chiều DC. Hai chân cấp nguồn được sử dụng để nối với + 5 VDC và tiếp đất. Tín hiệu điều khiển đọc/ghi được sử dụng để xác định khi nào

các bit dữ liệu được ghi vào bộ nhớ hay được đọc từ bộ nhớ. Bit dữ liệu được ghi vào bộ nhớ khi tín hiệu điều khiển (R/W) ở mức điện áp thấp và dữ liệu được truyền đi từ bộ nhớ khi tín hiệu điều khiển (R/W) ở mức điện áp cao. Một chíp phát tín hiệu điều khiển được dùng để chọn hoạt động của mỗi chíp riêng biệt khi một nhóm các mạch tích hợp được sử dụng để tạo ra bộ nhớ lớn hơn khả năng tạo ra bởi một chíp.



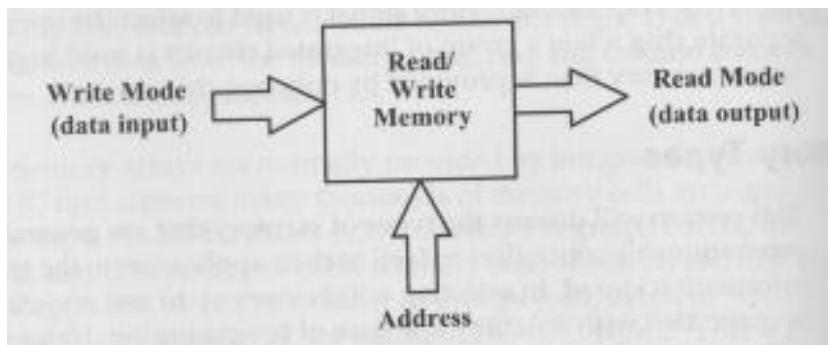
Hình 3.. Một chíp nhớ 1K-byte R/U đặc trưng

### Dạng bộ nhớ

Ta sẽ nói về dạng bộ nhớ hay được sử dụng trong PLC cũng như trong ứng dụng của chúng cho các dạng dữ liệu hay thông tin được lưu trữ. Trong việc lựa chọn bộ nhớ để sử dụng, kỹ sư thiết kế hệ thống phải chú ý khả năng xoá và khả năng lập trình dễ dàng. Kỹ sư thiết chịu trách nhiệm về việc mất thông tin trong bộ nhớ, bởi vì bộ nhớ giữ chương trình điều khiển quá trình, và nếu chương trình này bị xoá thì toàn bộ hệ thống sản xuất bị ngừng. Từ khi bộ nhớ lôi cuốn sự chú ý rằng nó có vị trí quan trọng giữa người sử dụng và PLC, nó có thể dễ dàng thay đổi. Sự tương tác bắt đầu từ khi bắt đầu lập trình cho hệ thống, sửa lỗi và tiếp tục với các thay đổi trực tuyến, như thay đổi các giá trị của các bộ đếm thời gian và bộ đếm thường

### Bộ nhớ đọc/ghi (R/W)

Bộ nhớ ghi và đọc được thiết kế để dữ liệu hay thông tin có thể được ghi vào hay đọc ra từ cùng một vị trí. Trên hình 3.. cho thấy dữ liệu có thể được đưa vào bộ nhớ bằng sử dụng phương thức ghi và có thể lấy ra bằng sử dụng phương thức đọc. Địa chỉ đầu vào bộ nhớ định rõ vị trí và địa chỉ của dữ liệu sẽ được đọc hay được ghi vào.



Hình 3.. Sơ đồ khối của bộ nhớ R/W

Đối với phần lớn PLC, sử dụng bộ nhớ R/W với pin hổ lưu cho các ứng dụng nhớ. Bộ nhớ R/W tạo biện pháp tuyệt vời để có thể tạo ra dễ dàng và thay đổi một chương trình điều khiển tốt như cho phép nhập dữ liệu. So sánh với một số dạng bộ nhớ khác, bộ nhớ R/W hoạt động rất nhanh. Chỉ có một bất lợi là pin hổ trợ bộ nhớ có thể hỏng theo thời gian. Mặc dù vậy, phần lớn PLC có tín hiệu ánh sáng báo pin sắp cạn để báo cho người vận hành thay nguồn pin hổ lưu cho bộ nhớ.

### Bộ nhớ chỉ dùng để đọc ROM

Bộ nhớ ROM được thiết kế để lưu thường xuyên một chương trình cố định mà bình thường không thể hay không bị thay đổi. Tên của bộ nhớ loại này xuất phát từ thực tế nội dung ghi trong nó chỉ được đọc và không thể ghi hay thay đổi được, một khi dữ liệu hay chương trình đã được lưu trong nó. Dữ liệu chỉ có thể sử dụng bằng phương thức đọc. Cũng như bộ nhớ R/W, bộ nhớ ROM cũng có địa chỉ đầu vào, nơi vị trí xác định của dữ liệu sẽ được đọc. Nhờ thiết kế của nó, bộ nhớ ROM không cần bị ảnh hưởng bởi nhiễu điện hay mất nguồn. Chương trình thực hiện hay chương trình của hệ thống điều hành thường được chứa trong bộ nhớ ROM.



PLC rất hiếm khi sử dụng bộ nhớ ROM cho các bộ nhớ chương trình điều khiển ứng dụng. Mặc dù vậy, trong các ứng dụng mà có yêu cầu các dữ liệu cố định, ROM có lợi thế hơn về tốc độ, giá thành và độ tin cậy. Thường các chương trình PLC trên cơ sở ROM được sản xuất tại nhà máy bởi nhà sản xuất thiết bị. Một khi một bộ lệnh gốc được lập trình, người sử dụng không bao giờ có thể thay đổi được nó. Nhà sản xuất sẽ ghi và sửa lỗi chương trình nhờ thiết bị điều khiển đọc/ghi hay máy tính và khi chương trình hoàn thiện sẽ được đưa vào bộ nhớ R/W hay ROM. ROM cũng được tìm thấy trong các bộ nhớ ứng dụng của các hệ thống PLC chuyên dụng như lò vi sóng, máy bán hàng, máy giặt và tương tự.

### Bộ nhớ chỉ dùng để đọc có khả năng lập trình PROM

Bộ nhớ PROM là dạng bộ nhớ đặc biệt của bộ nhớ ROM, rất hiếm được sử dụng trong các ứng dụng của PLC. Mặc dù vậy, khi nó được sử dụng, nó là dạng lưu trữ cố định được yêu thích nhất, hơn hẳn các bộ nhớ RAM (random Access Memory). PROM cũng

như các bộ nhớ ROM và nó cũng không bị mất thông tin khi mất nguồn hay bị ảnh hưởng của nhiễu. Bất lợi là lập trình cho PROM cần có thiết bị đặc biệt. Một khi đã lập trình nó cũng không thể xoá và thay đổi được. Một thay đổi trong chương trình cần phải có một bộ chip PROM mới. Bộ nhớ PROM có thể thích hợp cho lưu trữ chương trình đã được kiểm tra và nhớ trong bộ nhớ RAM và nó không cần thay đổi hoặc có dữ liệu vào trực tuyến.

### Bộ nhớ ROM có khả năng xoá bằng điện EEPROM

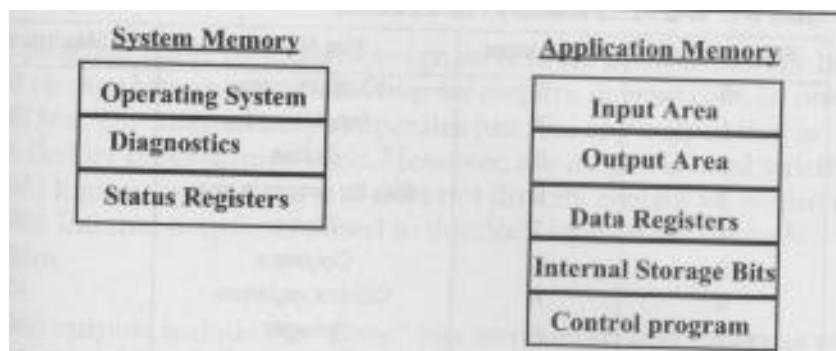
EEPROM là loại PROM đặc biệt có thể lập trình lại sau khi xoá hoàn toàn bằng điện. EEPROM có thể xem như thiết bị nhớ tạm thời, trong đó có thể chứa chương trình đến khi nó sẵn sàng được thay đổi. EEPROM tạo một phương tiện lưu trữ tuyệt vời cho chương trình điều khiển có yêu cầu không bị xoá và không thay đổi chương trình. Phần lớn các nhà sản xuất PLC chế tạo PLC với bộ nhớ EEPROM để tạo bộ nhớ cố định của chương trình máy sau khi đã phát triển, sửa lỗi và hoạt động tốt.

Một chương trình điều khiển gồm chir EEPROM sẽ không thuận tiện nếu có yêu cầu thay đổi trực tuyến hay cần dữ liệu vào. Mặc dù vậy, nhiều PLC đưa ra bộ nhớ chương trình bằng EEPROM như một yêu cầu thêm thay thế cho các bộ nhớ RAM có pin hỗ trợ. EEPROM tạo hệ thống nhớ thích hợp bởi vì nó kết hợp được khả năng lưu trữ và khả năng thay đổi dễ dàng của bộ nhớ R/W.

### Tổ chức của bộ nhớ

Bộ nhớ của PLC đặc trưng có hai phần chính: bộ nhớ hệ thống và bộ nhớ ứng dụng. Bộ nhớ hệ thống là nơi chứa của tập hợp các chương trình và các bộ ghi tạo thành chương trình điều hành hệ thống, phần mềm chẩn đoán sự cố, và các bộ ghi trạng thái của hệ thống. Hệ thống điều hành hướng các hoạt động như thực hiện chương trình điều khiển, trao đổi thông tin với thiết bị ngoại vi, hay các chức năng bảo vệ hệ thống.

Bộ nhớ ứng dụng gồm miền vào, miền ra, các bộ ghi dữ liệu hay thông tin, các miền bit lưu trữ bên trong và chương trình điều khiển. Bộ nhớ hệ thống và bộ nhớ ứng dụng có cách lưu trữ và cách lấy thông tin khác nhau. Bộ nhớ hệ thống chứa các lệnh để làm CPU hoạt động, chứa một bộ các chương trình chẩn đoán và các bộ ghi trạng thái. Bộ nhớ ứng dụng chứa các miền ảnh vào, miền ảnh ra, chương trình điều khiển, và các bộ ghi dữ liệu. Chúng sử dụng các loại bộ nhớ khác nhau. Phần hệ thống điều hành cần có bộ nhớ cố định để lưu các thông tin hay dữ liệu được lưu không bị thay đổi vô tình hay cố ý bởi mất nguồn hay bởi người sử dụng. Ở đây cần đến một số loại bộ nhớ ROM. Một khac người sử dụng cần thay đổi chương trình điều khiển hay dữ liệu vào/ra cho các ứng dụng cho trước, thì dùng bộ nhớ R/W là thích hợp.



### Cấu trúc của tệp

Một tệp nhớ được định nghĩa là một nhóm từ trong bộ nhớ mà chúng có chức năng chuyên dụng. Các tệp vào và tệp ra là các tệp chung giống nhau trong PLC. Các tệp này là các từ nhớ sắp xếp cạnh nhau để cho đầu vào từ bên ngoài và bit đầu ra được gán cho điểm vào/ra trong PLC. Cấu trúc tệp của PLC Allen – Bradley PLC5 được kê trong bảng 3..

Table 5-1. A-B PLC5 Memory File Structure

File Number	File Letter	File Name	Maximum Words
0	O	Output image	32
1	I	Input image	32
2	S	Status	32
3	B	Bits for program use	1000
4	T	Timers	1000
5	C	Counters	1000
6	R	Control registers	1000
7	N	Integer	1000
8	F	Floating point	1000

Bảng 3. Cấu trúc tệp nhớ của PLC S5 Allen Bradley

Để so sánh bảng 3.. giới thiệu một phần cấu trúc tệp nhớ của PLC S7 Siemens Simatic. Trong cả hai loại PLC này, bit ảnh vào được ký hiệu là chữ I trong lập trình. Bit ảnh ra được ký hiệu là O trong PLC S5 Allen-Bradley và là Q trong PLC Siemens S7.

Table 5-2. File Structure for Siemens Simatic S7 PLCs

File Letter	File Type	Description	Max. Address Range
I	Input image	Input bit	0 to 65535.7
IB		Input byte	0 to 65535
IW		Input word	0 to 65534
ID		Input double word	0 to 65532
Q	Output image	Output bit	0 to 65535.7
QB		Output byte	0 to 65535
QW		Output word	0 to 65534
QD		Output double word	0 to 65532
M	Bit memory	Memory bit	0.0 to 255.7
MB		Memory byte	0 to 255
MW		Memory word	0 to 254
MD		Memory double word	0 to 252
PIB	External inputs	Periph. input byte	0 to 65535
PIW		Periph. input word	0 to 65534
PID		Periph. In double word	0 to 65532
PQB	External outputs	Periph. input byte	0 to 65535
PQW		Periph. input word	0 to 65534
PQD		Periph. Out double word	0 to 65532
T	Timers	Timer	0 to 255
C	Counters	Counter	0 to 255
DBX	Data block	Data bit	0.0 to 65535.7
DBB		Data byte	0 to 65535
DBW		Data word	0 to 65534
DBD		Data double word	0 to 65532

Bảng 3.. Cấu trúc tệp nhớ của PLC S7 Siemens Simatic

### Bit nhớ trong

Phần lớn các PLC gán một miền để nhớ các bit trong. Các bit nhớ này còn được gọi là bit đầu ra bên trong , bit ra cuộn hút bên trong hay còn gọi là bit điều khiển trong. Đầu ra trong hoạt động giống như bất kỳ đầu ra nào, chỉ khác nó được điều khiển bằng lô gíc chương trình. Mặc dù vậy, đầu ra này được sử dụng hạn chế cho lập trình lô gíc bên trong và không điều khiển trực tiếp đầu ra nào tới quá trình. Các đầu ra trong được dùng để khoá lô gíc trung gian trong chương trình điều khiển. Đây chính là các biến trung gian, bởi chúng vừa có thể là biến ra trong vừa có thể là biến vào trong.

Đầu ra trong bao gồm cả bit thực hiện trên các bộ đếm và bộ đếm thời gian như một bit lô gíc trong các dạng khác nhau. Mỗi bit ra trong được so sánh bằng địa chỉ trong chương trình điều khiển và có bit lưu trữ trên cùng địa chỉ. Khi lô gic điều khiển là đúng “TRUE”, thì bit đầu ra trong bật lên giá trị 1 hay “ON”.

### Miền nhớ chương trình ứng dụng

Miền nhớ chương trình ứng dụng được sử dụng để chứa chương trình điều khiển lô gíc của quá trình công nghệ. Tất cả các lệnh của thiết bị điều khiển để điều khiển máy hay điều khiển quá trình được lưu trong miền nhớ này. Các địa chỉ của các bit thực và các bit vào/ra trong được xác định trong một phần của miền nhớ. Khi PLC đang ở chế độ hoạt động và chương trình điều khiển đang thực hiện, CPU dịch các vị trí nhớ này và điều

khiển các bit trong bảng dữ liệu, tương ứng với bit thực hay bit vào /ra trong. Chương trình điều khiển được dịch khi bộ xử lý thực hiện chương trình điều khiển.

Giá trị cực đại của bộ nhớ chương trình ứng dụng là khả năng đảm bảo hoạt động bình thường của chương trình điều khiển tùy thuộc cỡ của hệ thống điều khiển. Hệ thống PLC hạng trung hay hạng lớn, độ lớn của chương trình sử dụng cũng thường linh hoạt. Nó có thể được thay đổi bằng thay đổi kích thước bảng dữ liệu sao cho có thể tìm thấy các yêu cầu lưu trữ dữ liệu nhỏ nhất. Trong các PLC nhỏ, độ lớn của chương trình ứng dụng thường cố định.

### Kích cỡ của bộ nhớ ứng dụng

Cỡ của bộ nhớ ứng dụng là một thông số quan trọng trong thiết kế hệ thống điều khiển trên cơ sở PLC. Xác định cỡ của nhớ đúng có thể giảm giá thành và tránh mất nhiều thời gian sau này. Nếu tính toán cỡ của bộ nhớ chính xác ta không cần mua PLC với dung lượng thấp hơn hay vượt quá.

Cỡ của bộ nhớ ứng dụng có thể mở rộng đến điểm cực đại trong một số thiết bị điều khiển, nhưng không thể mở rộng trong một số các PLC nhỏ hơn. Các PLC nhỏ hơn ở đây xác định là các PLC điều khiển từ 10 đến 64 thiết bị vào/ra. Thiết bị điều khiển có sáu mươi tư hay nhiều hơn các thiết bị vào ra thường mở rộng bộ nhớ với dung lượng 1K, 2K, 4K hay tương tự. Mỗi K tương đương với 1024 từ. Trong các PLC cỡ lớn dung lượng bộ nhớ thường từ 64K trở lên.

Cỡ bộ nhớ được thông báo của các PLC chỉ mới đưa ra chỉ số thô của không gian nhớ có sẵn cho người sử dụng, bởi vì một phần của bộ nhớ đã phải dùng cho các chức năng bên trong của các thiết bị điều khiển.

Cần trả chính trong việc xác định cỡ của bộ nhớ cho ứng dụng là tính phức tạp của chương trình điều khiển chưa được xác định cho đến khi đã mua thiết bị. Như vậy, thường ta biết số đầu vào/ra trong hệ thống trước khi phân cứng được mua. Kỹ sư thiết kế hệ thống có thể dự trù cỡ bộ nhớ ứng dụng nhân với số đầu vào/ra với hai mươi tám nhón. ví dụ nếu hệ thống có khoảng một trăm điểm vào/ra, thì chương trình thường cần nói chung cỡ 2K từ. Cần chú ý là cỡ của chương trình bị ảnh hưởng bởi mức độ tinh vi của chương trình điều khiển. Nếu ứng dụng yêu cầu thu thập dữ liệu hay thuật toán điều khiển phức tạp, như điều khiển PID, thì việc bổ sung thêm bộ nhớ là cần thiết.

Sau khi kỹ sư thiết kế xác định bộ nhớ tối thiểu yêu cầu cho ứng dụng, cần phải tính thêm 25% đến 50% cho trường hợp thay đổi chương trình hay mở rộng chương trình.

### Ghi địa chỉ vào/ra

Từ khi một trong các mục tiêu chính là điều khiển các đầu vào và đầu ra của các thiết bị hiện trường, các đầu vào/ra phải chiếm chỗ trong bộ nhớ của bộ xử lý tín hiệu, nơi mà chúng được gán địa chỉ trong chương trình điều khiển của PLC. Mỗi điểm đầu trên mô đun vào/ra có thể được nối dây đến thiết bị hiện trường chiếm 1 bit trong bộ nhớ PLC. Phần của bộ nhớ chứa địa chỉ vào/ra được gọi là bảng ảnh vào và bảng ảnh ra.

### Bảng nhớ ảnh vào

Bảng ảnh vào là mảng các bit trong đó chứa trạng thái của các đầu vào gián đoạn từ quá trình công nghệ, mà chúng được nối đến đầu vào của PLC. Số lượng bit trong bảng bằng số lượng đầu vào cực đại. Mỗi thiết bị điều khiển có dung lượng cực đại là 64 đầu vào sẽ cần một bảng 64 bit đầu vào. Mỗi đầu vào được nối đến có một bit trong bảng đầu vào tương ứng chính xác tới điểm đầu mà đầu vào được kết nối. Nếu đầu vào ở trạng thái “ON” thì bit trong bảng nhớ cũng là ON hay lô gíc 1. Trong phần lớn các PLC, đầu vào ở trạng thái “ON” nếu điện có trên điểm nối vào. Nếu không có điện áp, bit tương ứng sẽ

xoá hay chuyển sang trạng thái “OFF” hay lô gíc 0. Bảng đầu vào tiếp tục được thay đổi phản ánh trạng thái dòng điện của các thiết bị đầu vào được kết nối. Chương trình điều khiển sử dụng các thông tin trạng thái để xác định trạng thái “TRUE” hay “FALSE” của các lệnh trong chương trình.

Bit Number																Word Number
17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:001
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:002
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:003
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:004
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:005
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:006
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1:007

Input I:007/12

The diagram shows a 16-bit register (Word I:007) with bits numbered 17 down to 0. A red arrow points to bit 12, which is labeled '1'. Below the register, the text 'Input I:007/12' is written.

Bảng 3.. là ví dụ đặc trưng của một bit vào đơn trong bảng ảnh vào. Điểm vào I:007/12 được nhận dạng trong bản đồ nhớ.

### Bảng nhớ ảnh ra

Bảng nhớ ảnh ra là mảng bit để điều khiển trạng thái của thiết bị đầu ra gián đoạn, mà thiết bị này được nối tới mạch giao diện của đầu ra. Mỗi đầu ra được kết nối có một bit trong bảng ảnh ra, tương ứng chính xác với điểm đấu mà đầu ra được kết nối. Bit trong bảng ra được điều khiển bởi bộ xử lý của PLC giống như nó dịch chương trình điều khiển và được cập nhật trong quá trình quét đầu vào/ra. Nếu một bit được bật lên trạng thái “ON” hay lô gíc 1, thí sau đó mạch điện nối đầu ra sẽ được kích hoạt, và ở đây sẽ có điện áp trên điểm nối đầu ra. Nếu bit bị xoá hay tắt “OFF” (tức lô gíc 0), đầu ra sẽ bị ngắt.

Hình 3.. minh họa bit ra đơn trong bảng ảnh ra. Trong ví dụ này, vị trí bit ra O:017/16 được chỉ ra trên bản đồ nhớ của đầu ra. Chữ O chỉ một đầu ra, từ địa chỉ 017, bit là bit 16, bit cuối cùng trong từ nhớ.

Bit Number																Word Number O:000
17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00	
0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	O:001
0	1	0	0	0	1	1	0	0	1	1	0	0	0	1	0	O:002
0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	O:003
0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	0	O:004
0	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	O:005
0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	O:006
1	1	0	0	0	1	0	1	0	1	1	0	0	0	1	0	O:007
Output O:007/02																

Hình 3.. Bit ra trong bảng nhớ ảnh ra của PLC s5 Allen – Bradley.

### Giao diện giữa phần cứng và phần mềm

Có lẽ là thứ quan trọng nhất để hiểu về thiết bị điều khiển là bộ xử lý sử dụng các đầu vào của quá trình như thế nào, thứ mà các mạch vào cảm nhận được, để kích hoạt các thiết bị đầu ra theo yêu cầu điều khiển quá trình công nghệ. Đây chính là giao diện giữa phần cứng và phần mềm. Địa chỉ các lệnh là thứ nối chương trình điều khiển của phần mềm với phần cứng của các điểm vào/ra.

Địa chỉ của các đầu vào và đầu ra của PLC nối vị trí vật lý của các mô đun đầu cuối đến vị trí bit nhớ. Cấu trúc hay mật độ của mô đun vào/ra: 8, 16 hay 32 điểm, có quan hệ trực tiếp đến các bit mà mô đun chiếm trong bộ nhớ của PLC. Ví dụ mô đun 8 điểm vào như hình 3.. tám bit 00 đến 07 chiếm 8 vị trí trong bảng nhớ vào trong bộ nhớ của bộ xử lý tín hiệu.

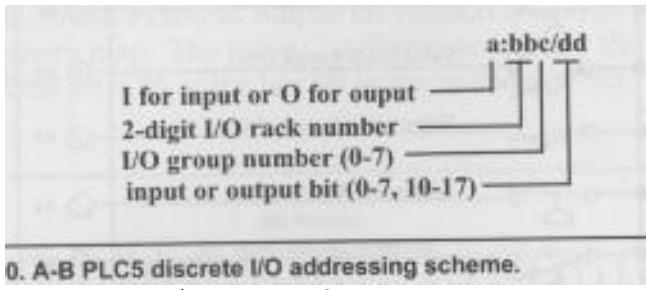
Hình 3.. Mô đun vào một chiều 8 điểm đặc trưng

Ta sẽ giới thiệu một số cách ghi địa chỉ của các hệ thống PLC.

### Cách ghi địa chỉ vào/ra trên Allen Bradley PLC S5

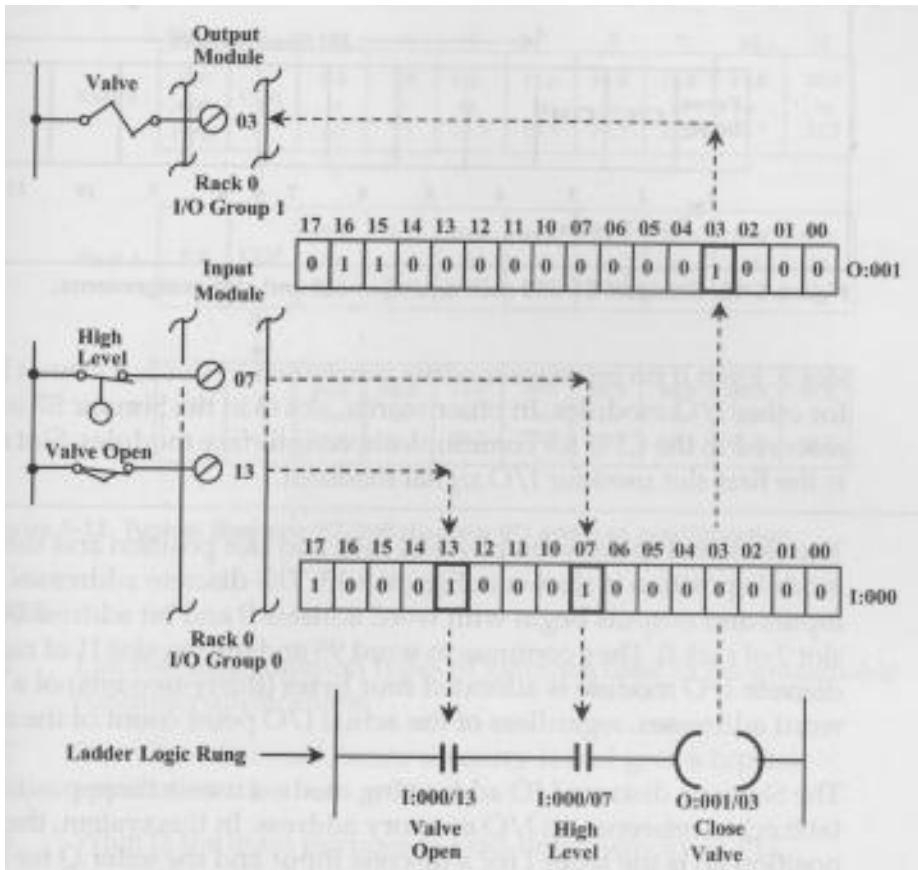
Phương pháp ghi địa chỉ vào ra gián đoạn của Allen – Bradley PLC S5 sử dụng mã 6 vị trí ( $a:bbc/dd$ ) để đổi chiều với địa chỉ nhớ vào/ra và vị trí vật lý của phần cứng. Trong hệ thống này, vị trí phía trái nhất là chữ I cho đầu vào gián đoạn và chữ O cho đầu ra gián đoạn. Hai chữ tiếp theo (bb) là số hiệu của mô đun, chữ số tiếp theo (c) là số của nhóm vào/ra (từ 0 đến 7). Hai chữ số còn lại thể hiện bit vào hay bit ra hay số của điểm cuối: 00 đến 07 hay 10 đến 17.

Ví dụ địa chỉ I:001/07 chỉ thị thiết bị đầu vào được nối đến mô đun số 00 và nhóm vào/ra số 1 tại điểm cuối 07 và bit nhớ 07. Địa chỉ O:074/10 chỉ thị thiết bị đầu ra được nối tới mô đun 07 tại điểm cuối 10 trên nhóm vào/ra số 4.



Hình 3.. Sơ đồ ghi địa chỉ vào/ra gián đoạn của Allen Bradley PLC S5

Một giao diện phần cứng đến phần mềm đối với ứng dụng của Allen Bradley PLC S5 được minh họa trên hình 3..



Hình 3.. Sơ đồ giao diện phần cứng tới phần mềm của PLC S5

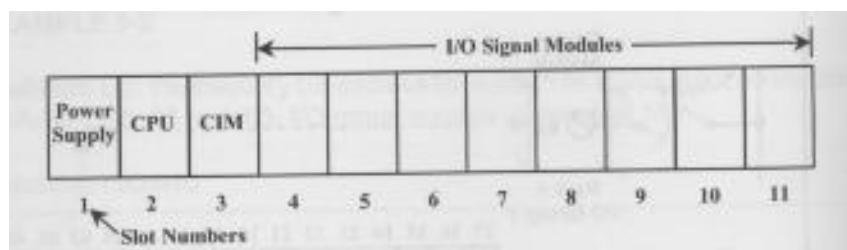
Ứng dụng này chỉ ra quan hệ hoạt động giữa các thiết bị hiện trường, bit nhớ vào/ra gián đoạn và chương trình lô gíc thang ứng dụng.

Trong ví dụ này, nếu công tắc mức cao được nối đến mô đun trong giá số 0, nhóm vào/ra số 0, điểm cuối số 7 được đóng, bit bên trong của phần mềm I:000/07 sẽ được bật lên giá trị 1. Đường gián đoạn từ điểm cuối số 7 đến vị trí của bit nhớ I:000/07 chỉ thi một kết nối trong hệ thống PLC S5. Nếu trong cùng thời gian công tắc van ở vị trí mở được nối đến điểm cuối 13 của cùng mô đun được đóng lại. Trong trường hợp này, bit vào I:000/13 sẽ được bật lên 1, và bậc của lô gíc thang sẽ có tính liên tục của lô gíc, và bit ra O:001/03 sẽ được bật lên giá trị 1. Điều này kích hoạt van con trượt nối với điểm cuối số 3 của giá số 0 và nhóm vào/ra số 1.

Bậc thang lô gíc trên đáy hình 3.. cho thấy ví dụ hai bit đầu vào từ bên ngoài được sử dụng để bật bit ra bên ngoài.

### Cách ghi địa chỉ vào/ra trên PLC Siemens Simatic S7-300

Trên hình 3.. chỉ ra thiết bị có 11 khe cài cho PLC S7 -300 Siemens Simatic. Số khe cài trên giá ảnh hưởng đến số đồ địa chỉ của S7-300. Địa chỉ vào /ra đầu tiên được xác định bởi vị trí của nó trên giá đỡ. Khe cắm đầu tiên được dành cho giá của bộ nguồn, tất nhiên địa chỉ vào/ ra không cần cho nguồn. Mô đun CPU được bố trí tiếp theo trong khe số 2. Không cần địa chỉ vào/ra gán cho mô đun CPU. Khe số 3 chứa mô đun giao diện (CIM: Communication Interface Module). Mô đun này được dùng để nối CPU trong giá đỡ vào/ra chính với các mô đun vào /ra trên giá mở rộng. Không có địa chỉ vào/ra nào được gán cho mô đun giao diện trong khe cắm số 3. Mặc dù ngay cả khi khe số 3 không có mô đun giao diện, thì không một mô đun nào được cắm vào đây. Khe số 4 là khe cắm đầu tiên cho các mô đun vào/ra.



Quan hệ giữa giá đỡ vật lý và vị trí khe cắm và vị trí mô đun vào/ra được chỉ trên hình 3... Địa chỉ gián đoạn cho cả đầu vào và đầu ra bắt đầu bằng địa chỉ từ số 0 và địa chỉ bit 0 (0.0) trong khe số 3 giá đỡ số 0. Chúng tiếp tục tới từ 95 và bit 7 trong khe cắm 11 của giá đỡ số 2. Mỗi mô đun vào/ra gián đoạn được dành 4 byte (32 bit) của địa chỉ từ nhớ, không cần quan tâm đến số lượng điểm vào/ra thực sự của mô đun.

Phương pháp ghi địa chỉ vào/ra gián đoạn của Siemens là dùng mã ba vị trí (abb.cc) để đổi chiều với địa chỉ nhớ vào/ra. Trong hệ thống này, vị trí ngoài cùng phía bên trái (a) là chữ I cho đầu vào gián đoạn và chữ Q cho đầu ra gián đoạn. Hai vị trí tiếp theo (bb) là số byte nhớ được gán cho khe cắm vào/ra, và chữ (c) bên phải dấu chấm là số của bit của bảng nhớ ảnh vào/ra (từ 0 đến 7). Ví dụ mô đun vào/ra trên khe cắm số 5 trên giá đỡ số 0 có 16 đầu vào gián đoạn. Tám đầu vào đầu tiên sẽ được gán địa chỉ từ 4.0 đến 4.7. Tám đầu vào tiếp theo sẽ được gán địa chỉ 5.0 đến 5.7.

	Slot #	3	4	5	6	7	8	9	10	11
Rack 0	P,S and CPU	CIM	0,0 to 3,7	4,0 to 7,7	8,0 to 11,7	12,0 to 15,7	16,0 to 19,7	20,0 to 23,7	24,0 to 27,7	28,0 to 31,7
Rack 1	P,S	CIM	32,0 to 35,7	36,0 to 39,7	40,0 to 43,7	44,0 to 47,7	44,0 to 47,7	48,0 to 51,7	52,0 to 55,7	60,0 to 63,7
Rack 2	P,S	CIM	64,0 to 67,7	68,0 to 71,7	72,0 to 75,7	76,0 to 79,7	80,0 to 83,7	84,0 to 87,7	88,0 to 91,7	92,0 to 95,7

## CHƯƠNG 4 LẬP TRÌNH BẰNG SƠ ĐỒ THANG TIÊU CHUẨN VÀ NÂNG CAO

### 4.1. Mở đầu

Ngôn ngữ lập trình là phương tiện cho phép người sử dụng giao tiếp với thiết bị điều khiển PLC thông qua các thiết bị lập trình. Các nhà sản xuất PLC sử dụng bốn ngôn ngữ lập trình là :

- 1- Sơ đồ thang
- 2- Bảng lệnh
- 3- Sơ đồ khối hàm lô gíc
- 4- Grafset

Chương trình điều khiển là một tổ hợp các lệnh được viết trong một trật tự nhất định. Có các qui tắc điều khiển phương pháp tổ hợp các lệnh và xác định các dạng của lệnh. Các qui tắc này và các lệnh tạo nên ngôn ngữ lập trình.

### 4.2. Các lệnh cơ bản của sơ đồ thang

Sơ đồ thang viết tắt tiếng anh là LAD, là tập hợp các lệnh dạng ký hiệu được sử dụng để tạo ra một chương trình điều khiển cho PLC. Các lệnh này có sáu loại lệnh: lệnh dạng rơ le, lệnh đếm thời gian và lệnh đếm, lệnh trao đổi dữ liệu, lệnh số học, lệnh truyền dữ liệu, lệnh điều khiển chương trình.

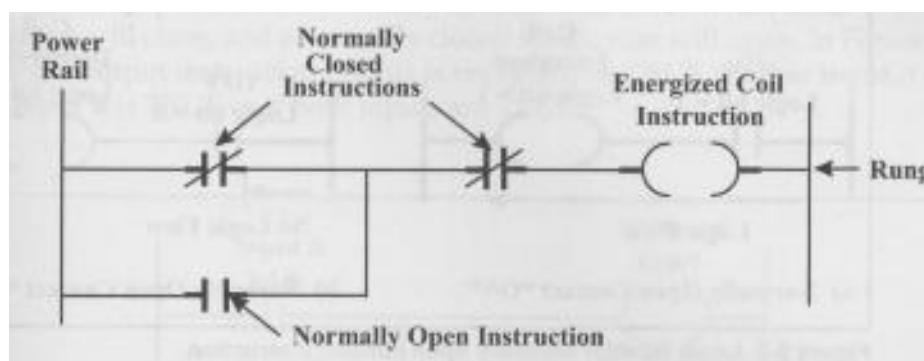
Chức năng chính của chương trình LAD là điều khiển các đầu ra trên cơ sở các điều kiện đầu vào. Sự điều khiển này được hoàn thành thông qua sử dụng tính lô gíc liên tục của các bậc của sơ đồ thang. Một bậc thang lô gíc gồm một tập hợp các điều kiện vào được thể hiện bởi các lệnh dạng tiếp điểm rơ le và trên kết thúc của mỗi bậc là lệnh ra thể hiện bằng ký hiệu của cuộn hút rơ le.

Cuộn hút và tiếp điểm là các ký hiệu cơ bản của tập hợp các lệnh sơ đồ thang. Các ký hiệu tiếp điểm được lập trình trên bậc thang thể hiện điều kiện cần thiết phải được đánh giá để xác định đầu ra sẽ được điều khiển như thế nào. Tất cả các đầu ra gián đoạn được thể hiện bằng ký hiệu cuộn hút.

Khi được lập trình, mỗi tiếp điểm và mỗi cuộn hút (tương ứng với một biến lô gíc) được qui chiếu với một địa chỉ số để nhận dạng và để biết biến nào đang được đánh giá và biến nào đang được điều khiển. Khi gọi đến các địa chỉ số này thì CPU sẽ đối chiếu với vị trí trong bảng dữ liệu để xác định giá trị bit bên trong hay giá trị lô gíc của đầu vào hoặc đầu ra được kết nối.

Hình thức của các tiếp điểm trên mỗi bậc thang phụ thuộc vào lô gíc điều khiển yêu cầu. Các tiếp điểm có thể được bố trí nối tiếp, song song hay vừa nối tiếp vừa song song, tuy theo yêu cầu điều khiển của đâu ra. Để một đầu ra được kích hoạt (tức là cấp năng lượng) thì tiếp điểm trên vị trí cuối cùng phía bên phải phải được đóng (CLOSE). Một đường truyền được đóng hoàn toàn là điều kiện để đảm bảo tính lô gíc liên tục. Khi tính lô gíc được đảm bảo liên tục trên ít nhất một đường truyền, ta có thể nói là điều kiện của bậc thang là đúng (TRUE hay ON), ngược lại là điều kiện không đúng (FALSE hay OFF).

Trong thời gian đầu, bộ lệnh sơ đồ thang tiêu chuẩn có khả năng rất hạn chế, chỉ thực hiện được các hàm lô gíc tương đương với lô gíc rơ le cứng sử dụng ký hiệu cơ bản của tiếp điểm và cuộn hút. Sự cần thiết của tính linh hoạt và cùng với sự phát triển nhanh chóng của công nghệ vi điện tử, đã đưa đến sự mở rộng các lệnh sơ đồ thang, trong đó cho phép thực hiện các phép tính số học, thực hiện di chuyển dữ liệu và điều khiển luồng các lưỡng của chương trình.



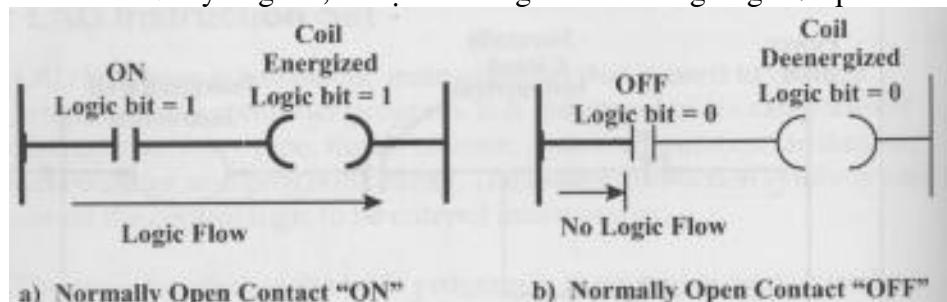
Hình 3... Các lệnh sơ đồ thang cơ bản

### Các lệnh dạng rơ le

Các lệnh dạng rơ le là lệnh cơ bản nhất trong các lệnh của PLC. Các lệnh này tạo cùng một khả năng như lô gíc rơ le ‘cứng’, nhưng có tính linh hoạt cao hơn nhiều. Các lệnh này đầu tiên tạo khả năng kiểm tra trạng thái ON hay OFF của bit có địa chỉ xác định trong bộ nhớ và điều khiển trạng thái của bit ra trong hay bit ra ngoài.

### Lệnh thường mở

Lệnh thường mở NO được lập trình khi sự hiện diện của tín hiệu vào cần để bật đầu ra lên trạng thái ON. Khi được đánh giá, địa chỉ tham chiếu sẽ được xem xét đối với điều khiển ON (lô gíc 1) hay OFF (lô gíc 0). Địa chỉ tham chiếu có thể thể hiện trạng thái của tín hiệu vào ngoài hay tín hiệu ra ngoài. Nếu khi kiểm tra bit của địa chỉ tham chiếu là ON hay lô gíc 1, thì lệnh thường mở cho dòng lô gíc đi qua như hình 4...

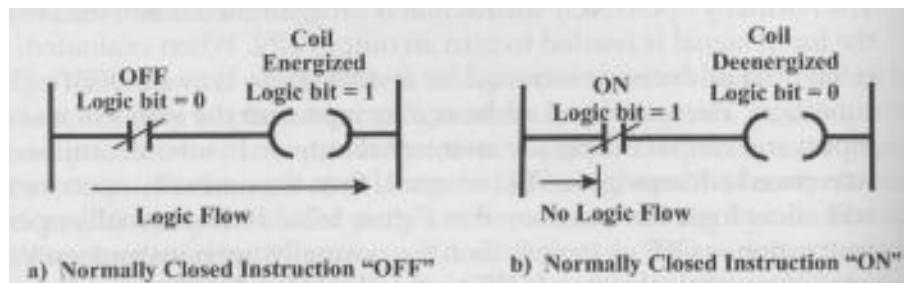


Hình 4.. Dòng lô gíc đối với lệnh tiếp điểm thường mở.

Nếu lệnh thường mở NO là OFF hay lô gíc 0, thì tính liên tục của lô gíc bị đứt và dòng lô gíc bị ngăn lại. Để trợ giúp trong xử lý sự cố của chương trình điều khiển, phân lớn các phần mềm lập trình PLC sẽ sáng bit lô gíc chỉ thị trạng thái ON của bit đầu vào và bit đầu ra. Một số phần mềm sẽ bật sáng cả bậc thang nếu tính liên tục của lô gíc hay dòng lô gíc của toàn bộ bậc thang được kích hoạt hay ở trạng thái ON.

### Lệnh thường đóng

Lệnh thường đóng NC được sử dụng khi không có tín hiệu tham chiếu để bật đầu ra lên trạng thái ON. Khi kiểm tra giá trị của lệnh NC, bit địa chỉ tương ứng của lệnh này được kiểm tra xem là đang có điều kiện lô gíc ON (1) hay OFF (0). Địa chỉ tham chiếu bởi lệnh NC có thể thể hiện trạng thái của tín hiệu vào bên ngoài hay tín hiệu ra bên ngoài. Nếu khi kiểm tra địa chỉ bit tham chiếu có giá trị OFF hay lô gíc 0, thì tiếp điểm thường đóng vẫn tiếp tục giữ trạng thái đóng, cho phép tính lô gíc được liên tục. Nếu bit trên địa chỉ tham chiếu có giá trị ON hay lô gíc 1, thì tiếp điểm thường đóng NC bị ngắt và làm gián đoạn dòng lô gíc.

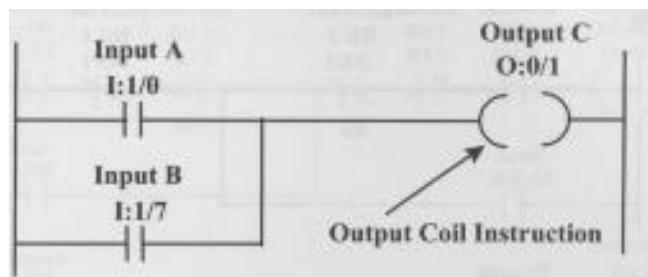


Hình 4.. Dòng lô gíc qua lệnh thường đóng NC

### Lệnh ra cuộn hút

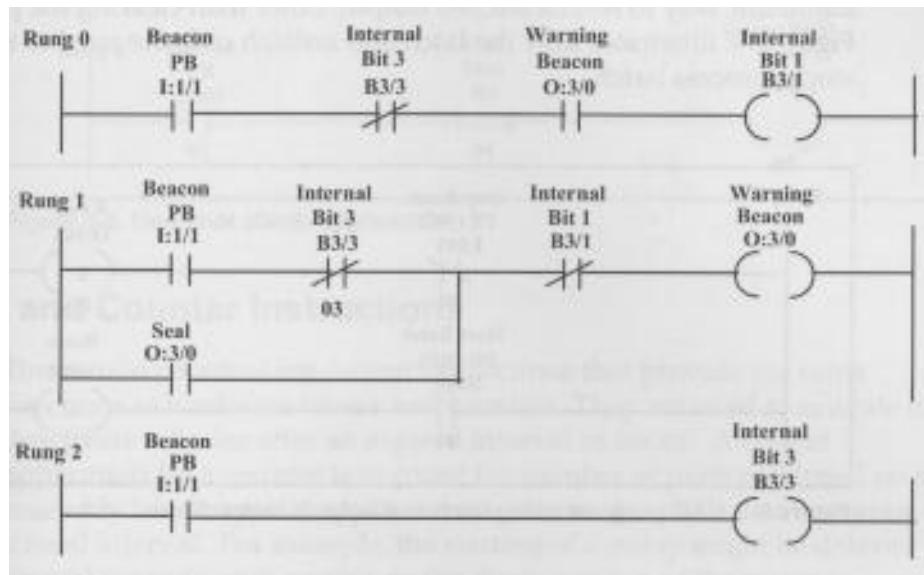
Lệnh ra cuộn hút hay lệnh kích hoạt cuộn hút được lập trình để điều khiển đầu ra kết nối với thiết bị điều khiển hay bit đầu ra bên trong (đầu ra trung gian). Bit lệnh đầu ra cuộn hút thường được ký hiệu là chữ **O** hay chữ **Q** trong hầu hết các hệ thống điều khiển PLC. Nếu một bậc nào đó có tính liên tục lô gíc thì đầu ra tương ứng sẽ được kích hoạt hay bật lên trạng thái TRUE (lô gíc = 1). Bit đầu ra sẽ chuyển về trạng thái OFF nếu dòng lô gíc đến đầu ra cuộn hút bị gián đoạn.

Khi đầu ra được bật ON, lệnh thường mở trên cùng địa chỉ sẽ chuyển sang đóng và các thường đóng sẽ chuyển sang mở. Trên hình 4.. là lệnh ra O:0/01 được kích hoạt hay TRUE nếu đầu vào A hay B là TRUE hoặc cả hai đều là TRUE.



Hình 4.. Lệnh ra cho thực hiện lô gíc OR bằng LAD

Ví dụ : Viết chương trình bằng LAD để khởi động và dừng bơm. Trong ứng dụng này, công tắc thường mở của nút khởi động trên hột điều khiển được nối tới địa chỉ bit vào I: 1/1, và tiếp điểm thường đóng NC của nút dừng được nối đến bit vào địa chỉ I: 1/0. Rơ le khởi động bơm được nối đến đầu ra của PLC O:3/1, và tiếp điểm khởi động phụ NO được nối đến đầu vào của PLC I:1/2.

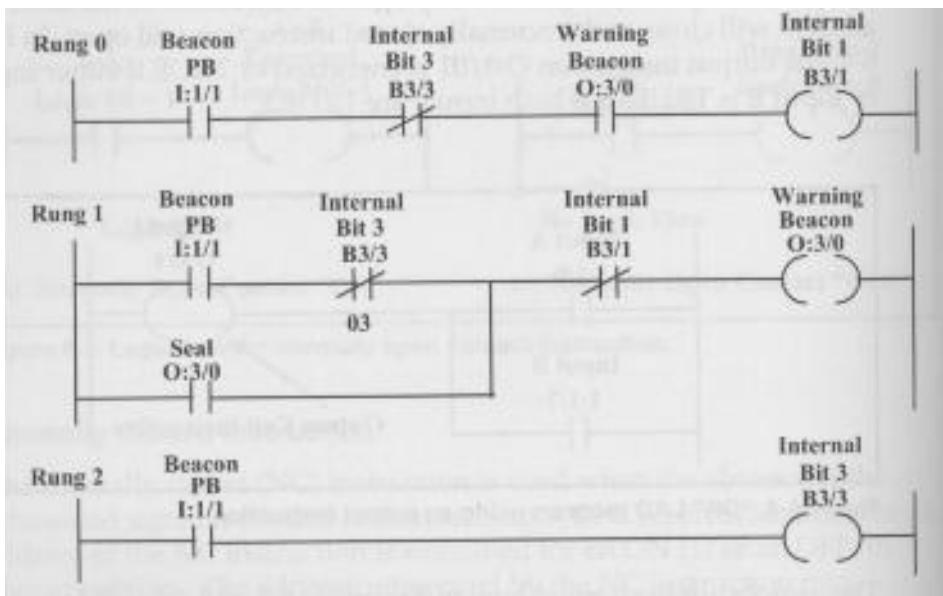


Lời giải như trên hình 4..

Khi nút bấm khởi động NO được ấn, đầu vào I:1/1 có giá trị TRUE. Khi nút bấm dừng NC chưa ấn, đầu vào I:1/0 cũng là TRUE. Kết quả tính liên tục của lô gíc được đảm bảo trên bậc 0, và bit đầu ra O:3/0 kích hoạt hay bật lên giá trị 1. Đầu ra O:3/1 kích hoạt rơ le khởi động bơm, gây ra công tắc phụ đóng lại. Lần này, bit vào I:1/2 duy trì bit khởi động và giữ cho bơm ở trạng thái bật cho đến khi nút bấm dừng được ấn. Khi nút dừng được ấn, bit dừng I:1/0 được bật về không, bit ra để chạy bơm được ngắt năng lượng và cũng bật về 0. Kết quả bơm sẽ tắt và các tiếp điểm phụ trên bộ khởi động bơm sẽ mở ra và bật bit đầu vào I:1/2 về 0.

Điều quan trọng cần ghi nhớ rằng tiếp điểm thường đóng NC của nút bấm dừng luôn được sử dụng để đảm bảo rằng các thiết bị chuyển động hoạt động an toàn. Các tiếp điểm NC được sử dụng trong mạch dừng, như vậy nếu dây từ nút bấm dừng tới PLC bị cắt hay tháo ra, thiết bị chuyển động sẽ dừng và không thể khởi động lại được. Một giả thiết rằng các tiếp điểm thường mở của nút bấm dừng được sử dụng và dây điều khiển từ nút bấm đến PLC được cắt hay tháo ra, khi thiết bị chuyển động dang hoạt động. Trong trường hợp này, ấn nút bấm dừng không thể dừng được thiết bị.

Ví dụ 2: Có một nút bấm đơn để bật đèn nháy báo trạng thái hoạt động hay dừng. Viết chương trình điều khiển bằng sơ đồ thang để điều khiển đèn nháy. Giả thiết tiếp điểm thường mở của nút ấn được nối đến điểm I:1/1 và đèn nháy được nối đến đầu ra O:3/0



Hình 4.. Chương trình sơ đồ thang LAD của đèn nháy báo động

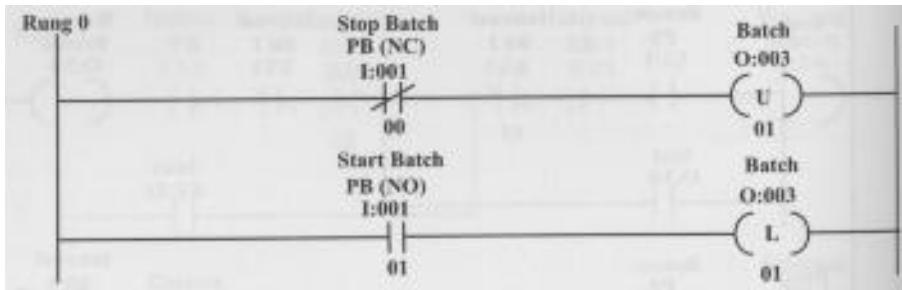
Cũng cung một chương trình trên hình 4., khi nút bấm điều khiển đèn nháy được ấn lần thứ nhất, bit đầu ra O:3/0 được kích hoạt và bật đèn nháy ON. Bit điều khiển đầu ra này cũng có bit duy trì của chính nó. Nếu nút bấm được ấn tiếp, đèn nháy tắt (OFF). Bậc thang thứ hai (bậc số 1) của chương trình phát hiện là nút bấm lần thứ nhất được ấn, khi bậc thứ nhất (bậc số 0) cảm nhận nút ấn được ấn lần thứ hai. Trên bậc cuối cùng (bậc số 2) được dùng để điều khiển bit trong số 3 (b3/3). Tiếp điểm thường đóng NC hay bit trong số 3 được sử dụng trên bậc 0 và 1 để giúp thực hiện chức năng ấn để khởi động và ấn để dừng của chương trình sơ đồ thang.

### Lệnh duy trì - Latch Coil

Lệnh này được lập trình nếu cần để đảm bảo rằng đầu ra vẫn duy trì hoạt động mặc dù trạng thái của bit vào gây ra kích hoạt đã thay đổi. Nếu đường dẫn của một bậc có tính liên tục lô gíc thì đầu ra sẽ bật lên ON và giữ trạng thái ON ngay cả khi tính liên tục lô gíc không còn hay nguồn của hệ thống bị ngắt. Đầu ra duy trì sẽ giữ trạng thái ON cho đến khi được có lệnh ngừng duy trì đầu ra. Lệnh ngừng duy trì đầu ra được lập trình tự động bằng cách bật lại lệnh duy trì. Mặc dù phần lớn các thiết bị điều khiển cho phép đầu ra trong hay đầu ra ngoài được duy trì, một số khác lại hạn chế chỉ duy trì các tín hiệu ra bên trong mà thôi.

### Lệnh ngừng duy trì Unlatch Coil

Lệnh này được lập trình để bật lại trạng thái ban đầu của đầu ra duy trì trên cung một địa chỉ. Nếu một bậc có tính liên tục lô gíc, thì địa chỉ tham chiếu được tắt (OFF). Đầu ra không duy trì chỉ là tự động bật lại trạng thái ban đầu của đầu ra duy trì, hay nói cách khác là xoá chương trình. Hình 4.. minh họa một lệnh duy trì và lệnh ngừng duy trì được sử dụng để khởi động và dừng một đợt xử lý.

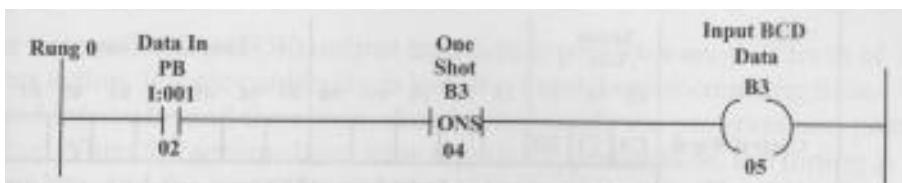


Hình 4.. Chương trình LAD sử dụng lệnh duy trì và lệnh ngừng duy trì.

Ví dụ tiếp điểm thường mở của nút ấn khởi động và tiếp điểm thường đóng của nút bấm dừng được nối vào đầu vào gián đoạn của PLC. Tiếp điểm NC của nút dừng được nối vào bit đầu vào I:001/00, và tiếp điểm NO của nút khởi động được nối đến đầu vào I:001/01. Nếu nút khởi động được ấn, đầu ra O:003/01 là giá trị ON duy trì. Khi bit khởi động I:001/01 chuyển sang FALSE, lệnh đầu ra của bơm vẫn duy trì ON cho đến khi bit dừng I:001/00 được ấn để ngừng duy trì đầu ra. Chú ý rằng lệnh ngừng duy trì đầu ra có cùng một địa chỉ như địa chỉ của bit duy trì. Chương trình LAD hình 4.. là một phương pháp đơn giản hơn để tạo ra chức năng khởi động và dừng mà sau đó chương trình LAD khởi động/dừng chỉ ra trên hình 4..

#### Lệnh nhảy One shot (ONS)

Lệnh nhảy ONS là lệnh đầu vào và nó lấy giá trị TRUE cho một chu trình quét của PLC, nếu có sự chuyển tiếp trạng thái từ FALSE sang TRUE trong những điều kiện trước đây trên bậc. Lệnh này nói chung được sử dụng để khởi động các thao tác được khởi động đồng thời với hoạt động của nút bấm, như khi PLC đạt được giá trị từ công tắc ấn khớp hay hiển thị nhánh dữ liệu LED. Trong Alen Bradley S5 PLC, thì địa chỉ bit phải là một tệp nhị phân (B3) hay tệp nguyên INTERGER (N7). Lệnh đặc trưng minh họa trên hình 4... Trong ứng dụng này, khi dữ liệu trên nút ấn được ấn, nó bật bit đầu vào I:001/02 lên 1, và điều kiện của bit ONS (B3/04) bậc mà đầu ra (B3/05) bật ON cho một lần quét. Đầu ra bật OFF để quét liên tục cho đến khi đầu vào chuyển từ FALSE sang TRUE lần nữa.



Hình 4. Ứng dụng của lệnh ONS.

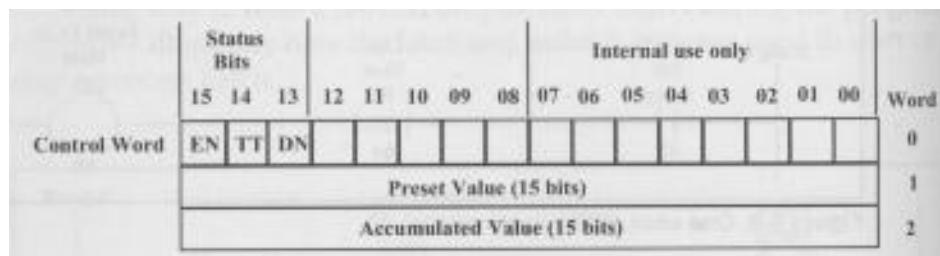
#### Lệnh đếm thời gian và lệnh đếm

Đây là các lệnh ra mà chúng cung cấp cùng một chức năng như bộ đếm giờ hay bộ đếm “cứng”. Khi chúng được sử dụng để kích hoạt hay ngừng một thiết bị sau một khoảng thời gian hay một số lượng. Ứng dụng đặc trưng của bộ đếm là đếm số chi tiết sản xuất trên một dây chuyền lắp ráp. Ứng dụng đặc trưng cho bộ đếm thời gian là sự trễ của một hoạt động cho một chu kỳ cố định. Ví dụ, khởi động bơm có thể được trễ cho vài giây, cho đến khi van trên đường xả của bơm được mở hoàn toàn.

Bộ đếm giờ và bộ đếm hoạt động hoàn toàn tương tự bởi vì bộ đếm giờ cũng chính là bộ đếm. Bộ đếm giờ dùng để đếm các khoảng thời gian cố định. Còn bộ đếm là để đếm các xuất hiện của các sự kiện.

### Cấu trúc từ của bộ nhớ thời gian

Các lệnh đếm thời gian yêu cầu ba bộ ghi nhớ hay ba từ: từ điều khiển hay bộ ghi, từ ác qui để chứa khoảng thời gian trôi qua, và từ bật lại bộ nhớ để chứa giá trị đặt trước của bộ đếm thời gian. Giá trị đặt trước sẽ xác định số khoảng thời gian sẽ được đếm. Khi giá trị tích luỹ bằng giá trị đặt trước, bit trạng thái được bật lên ON và có thể được sử dụng để bật bit đầu ra.

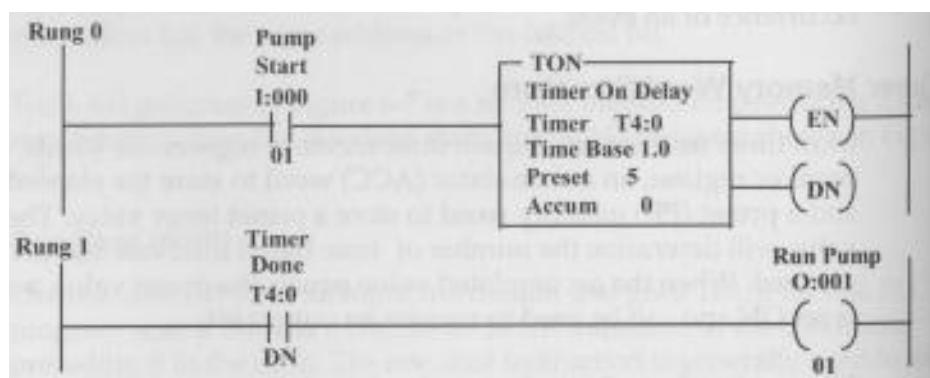


Tren hình 4.. chỉ ra một ví dụ đặc trưng của cấu trúc từ nhớ thời gian của hệ thống điều khiển PLC S5 Allen Bradley. Ba bit bên trái (14,15,16) trong từ điều khiển của bộ đếm thời gian được sử dụng như các bit trạng thái. Bit 15 là bit cho phép bộ đếm thời gian hoạt động (EN), và nó được bật khi lô gíc của bộ đếm thời gian là 1 hay TRUE. Bit 14 là bit thời gian của bộ đếm thời gian (TT), và nó được bật khi bậc của bộ đếm trở thành TRUE. Điều này chỉ thị rằng thời gian hoạt động trong gia tăng. Bit 13 là bit thời gian đã qua (DN), và nó có giá trị TRUE khi giá trị tích luỹ bằng giá trị thời gian đặt trước.

Hình Cấu trúc từ của bộ đếm thời gian allen Bradley PLC s5.

### Bộ đếm thời gian trễ TON (Time On Delay)

Đây là lệnh ra được lập trình để tạo hoạt động trễ hay do sự kéo dài của sự kiện xuất hiện. Nếu một đương dẫn của bậc được nối đến phía đầu vào của bộ đếm thời gian đã có tính liên tục lô gíc, như trên hình 4.., bộ đếm bắt đầu đếm chu kỳ thời gian. Nó đếm đến khi thời gian tích luỹ ACCUM bằng giá trị đặt trước dài như điều kiện bậc duy trì được giá trị TRUE. Khi thời gian tích luỹ bằng thời gian đặt trước thì bit của bộ đếm thời gian hoàn thành (DN) trong từ được bật lên 1. Bất kỳ lúc nào khi điều kiện lô gíc của bậc đổi với lệnh TON chuyển sang FALSE, giá trị tích luỹ sẽ được bật tất cả về không.



#### Hình 4.. Sơ đồ thang sử dụng lệnh TON

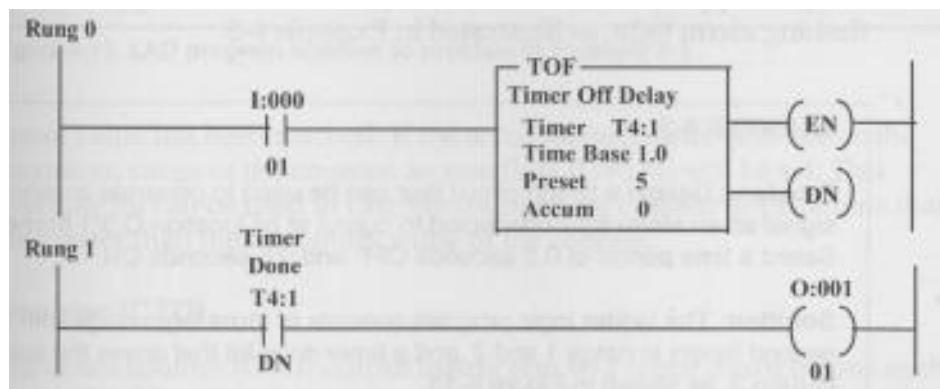
Trong ví dụ ứng dụng trên hình 4.. khi công tắc khởi động bơm ở trạng thái ON, bit I:000/01 được bật lên 1, và bộ đếm thời gian (T4:0) bắt đầu đếm các đơn vị thời gian. Thời gian mà công tắc giữ giá trị CLOSE hay ON, bộ đếm thời gian làm tăng giá trị của từ tích luỹ cho mỗi khoảng thời gian. Khi giá trị tích luỹ bằng giá trị đặt trước là 5 giây, bộ đếm thời gian ngừng đếm và bật bit thời gian đã qua (DN) lên ON. Bit đã thực hiện xong này (T4:0/DN) được sử dụng sau đó trên bậc thang 1 để kích hoạt bit đầu ra của bơm (O:001/01).

#### Bộ đếm sớm TOF (Time Off Delay)

Bộ đếm TOF là lệnh đầu ra, tạo một dạng khác của hoạt động của bộ đếm thời gian. Nếu tính liên tục lô gíc bị mất, bộ đếm thời gian bắt đầu đếm các khoảng thời gian cho đến khi thời gian tích luỹ bằng thời gian được lập trình trước đấy. Khi thời gian tích luỹ bằng thời gian đặt trước, bộ đếm ngừng đếm và bit đếm xong (bit 13) được bật về 0. Bit thời gian hoàn thành (DN) có thể được sử dụng như các lệnh tiếp điểm thường mở NO hay thường đóng NC. Nếu tính liên tục lô gíc lại có trước khi bộ đếm thời gian đếm xong, từ tích luỹ được bật về không và bit hoàn thành được bật lên lô gíc 1.

Trên hình 4.. là ví dụ về chương trình đối với lệnh TOF với giá trị đặt trước là 5 giây.

Trên bậc 0, khi đầu vào I:000/01 là TRUE, bit DN được bật lên 1, bộ đếm thời gian bắt đầu đếm và bit đầu ra O:001/01 lên ON. Nếu công tắc đầu vào I:000/01 là OPEN cho 5 giây hay nhiều hơn thì bộ đếm thời gian sẽ đếm tăng đến 5 giây. Khi giá trị đặt trước bằng giá trị tích luỹ thì bit hoàn thành (T4:1/DN) được bật về 0 và bit đầu ra O:001/01 trên bậc thang số 1 sẽ được bật về OFF.

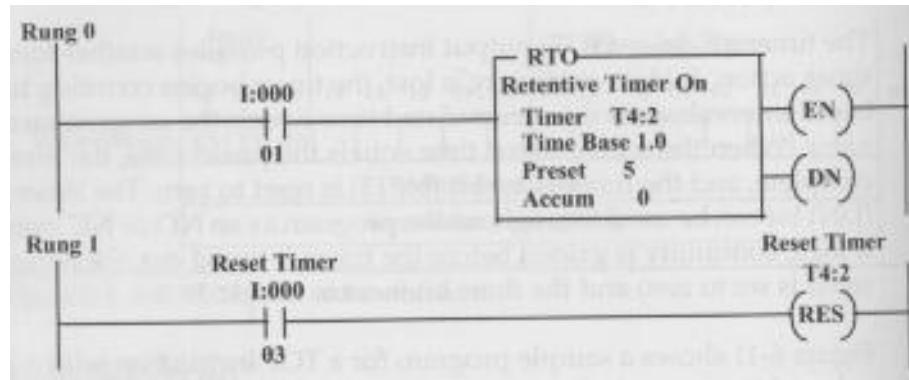


Hình 4.. Lệnh TOF trên sơ đồ thang

#### Bộ đếm thời gian duy trì RTO (Retentive Timer On)

Bộ đếm RTO cũng là lệnh đầu ra. Lệnh này được sử dụng nếu ta cần giữ giá trị tích luỹ của bộ đếm thời gian, mặc dù tính lô gíc liên tục hay nguồn năng lượng bị mất. Nếu đường dẫn trên bậc của bộ đếm thời gian có tính liên tục lô gíc, bộ đếm bắt đầu đếm cho đến khi giá trị tích luỹ bằng giá trị đặt trước. Bộ ghi tích luỹ giữ giá trị này cả khi tính liên tục lô gíc bị mất hay nguồn bị mất trước khi bộ đếm thời gian thực hiện đếm xong. Khi thời gian tích luỹ bằng thời gian đặt trước, bit đếm xong (timer done) được bật lên giá 1. Bit thực hiện đếm thời gian xong có thể được sử dụng trong phần còn lại của chương trình như là lệnh tiếp điểm thường mở NO hay thường đóng NC. Giá trị tích luỹ của bộ đếm thời gian duy trì được bật lại về 0 bởi lệnh RES (reset). Lệnh RES :

(T4:2/RES) trên bậc thang số 1 sẽ bật lên giá trị lô gíc 1, nếu bit I:000/03 được bật lên 1. Điều này sẽ bật lại bộ ăca qui trên bộ đếm T4:2 về 0 và bật lại bit DN.

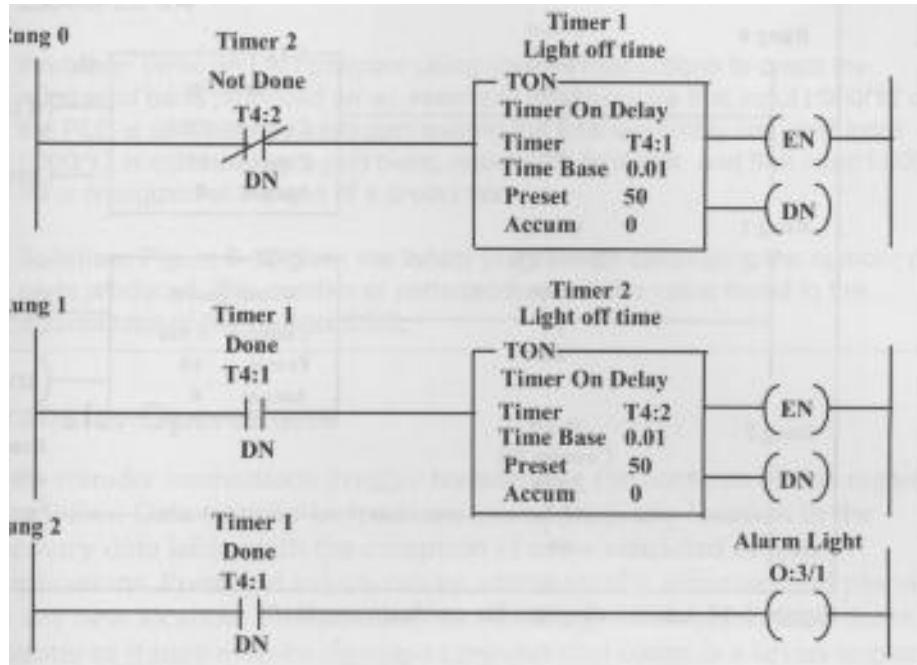


Hình 4.. Lệnh RTO

Đang dụng đặc trưng cho các bộ đếm thời gian là để tạo ra các xung thay đổi cho đèn nháy, đèn báo động.

Ví dụ 3: Thiết kế mạch thời gian sao cho mạch này được sử dụng để tạo ra tín hiệu thay đổi như đèn báo động nối đến đầu ra tại đèn nháy với vị trí bit O:3/1. Chọn chu kỳ thời gian là 0.5 giây tắt (OFF) và 0.5 giây bật (ON).

Giải: chương trình LAD gồm ba bậc thang với bộ đếm 0.5 giây trên bậc thứ nhất và thứ hai, và bit đếm xong điều khiển đèn báo động trên bậc thang thứ 3.



### Bộ đếm tăng CTU

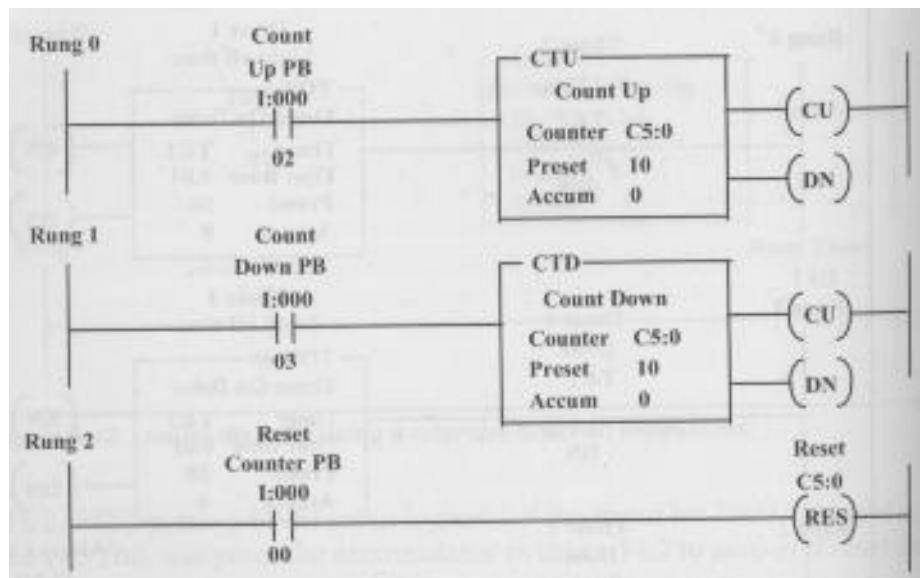
Giá trị tích luỹ trong lệnh đầu ra của bộ đếm tăng (CTU) sẽ tăng lên một trong mỗi lần có trạng thái chuyển tiếp từ 0 lên 1 của lô gíc đầu vào. Ứng dụng điều đặc trưng là đếm để bật thiết bị lên ON hay OFF sau khi một số lượng nhất định được đếm. Từ lúc đếm tăng giá trị tích luỹ chỉ khi đầu vào lô gíc của bộ đếm thực hiện từ 0 lên 1.

Khi điều kiện của bậc phải đi từ TRUE sang FALSE và ngược lại về TRUE trước khi bước đếm tiếp theo được ghi nhận.

Khi giá trị tích luỹ đạt giá trị cho trước, bit đếm xong bật lên 1. Không như lệnh đếm thời gian, lệnh đếm tiếp tục đếm để tăng giá trị tích luỹ sau khi đã đạt giá trị cho trước. Nếu giá trị tích luỹ vượt quá miền đếm cao nhất, một bit tràn OV (overflow) được bật lên

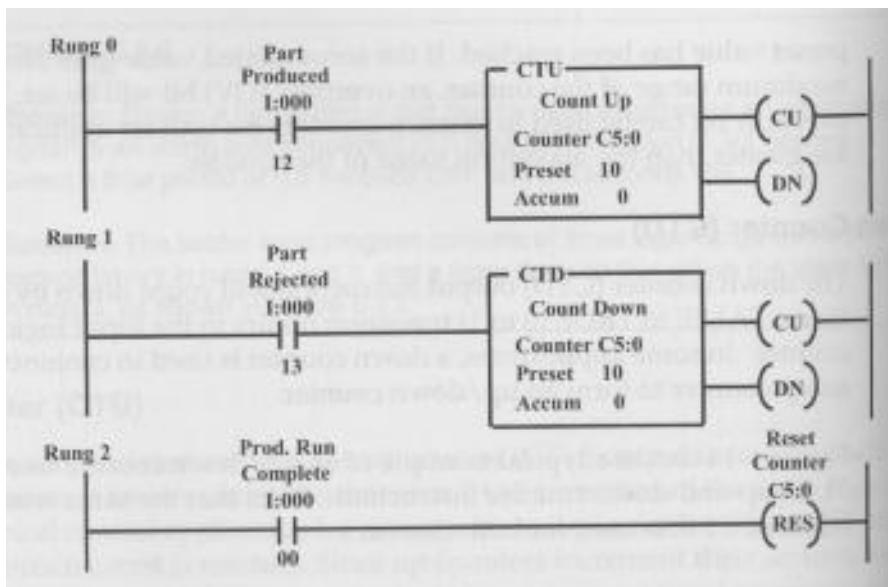
### Bộ đếm giảm CTD (counter down)

Lệnh ra của bộ đếm giảm sẽ tính lùi một đơn vị mỗi lần khi có trạng thái chuyển tiếp từ FALSE sang TRUE của lô gíc đầu vào của bộ đếm. Trong một số ứng dụng, bộ đếm giảm được sử dụng chung với bộ đếm tăng, tạo ra bộ đếm tăng/giảm. Ví dụ trên hình 4.. là bộ đếm tăng /giảm của Allen Bradley PLC S5. Cùng một địa chỉ C5:0 được sử dụng cho cả hai bộ đếm.



Ví dụ 4: Lập trình bằng LAD sử dụng lệnh đếm để đếm số chi tiết sản xuất trên dây chuyền lắp ráp. Giả thiết rằng đầu vào I:000/12 của PLC được kích hoạt bởi mỗi chi tiết rời khỏi dây chuyền lắp ráp, đầu vào I:000/13 được kích hoạt khi chi tiết bị loại do phế phẩm sau kiểm tra công đoạn cuối, và đầu vào I:000/00 được cấp năng lượng tại cuối mỗi hành trình sản xuất.

Giải: Số chi tiết được sản xuất là giá trị tìm thấy trong ác qui của bộ đếm C5:0.



Hình 4..LAD cho đếm sản phẩm.

### Các thao tác truyền dữ liệu

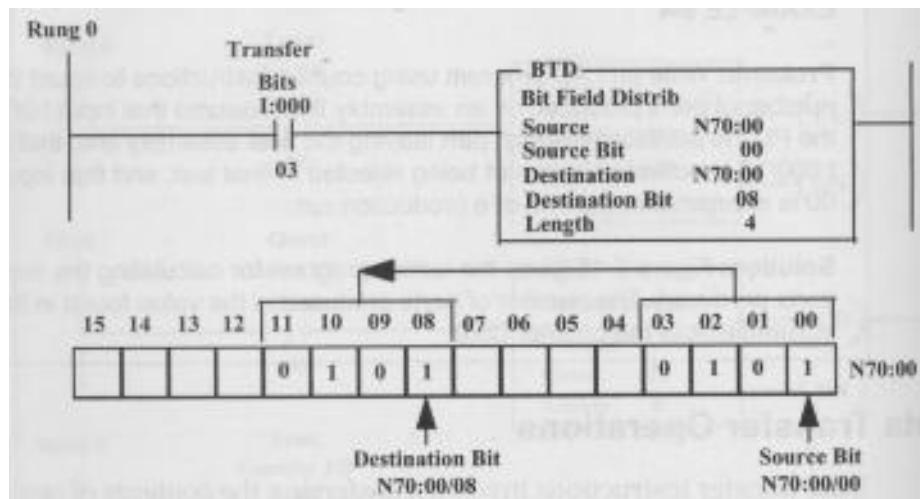
Lệnh truyền dữ liệu gây ra sự truyền các nội dung của một bộ ghi sang bộ ghi khác. Lệnh truyền dữ liệu có thể ghi địa chỉ một vị trí nào đó trong bảng nhớ dữ liệu, với cá biệt các miền hạn chế cho các ứng dụng. Các giá trị lưu trước đó có thể được tự động lấy ra và cho vào một vị trí mới nào đó. Vị trí này có thể là bộ ghi được đặt trước đối với bộ đo thời gian hay bộ đếm hay có thể cả hai là bộ ghi đầu ra điều khiển màn hình số 7 đoạn.

Hệ lập trình Allen Bradley PLC S5 sử dụng 3 lệnh truyền bit dữ liệu và từ : Bit phân phối BTD, chuyển dịch MOV, và chuyển động giấu MVM. Các lệnh truyền dữ liệu được sử dụng bởi phần lớn các nhà sản xuất PLC.

### Bit phân phối BTD

Lệnh BTD là lệnh ra, mà nó dịch chuyển lên 16 bit dữ liệu trong hay giữa các từ. Nguồn của dữ liệu được giữ không thay đổi. Trên hình 4.. là ví dụ lệnh BTD dịch chuyển các bit trong một từ . Lệnh này ghi chồng lên đích với các bit đặc biệt. Nếu chiều dài của trường bit vượt qua từ đích, bộ xử lý sẽ không ghi nhớ phần các bit tràn. Bởi vì chúng không được bọc vào từ kế tiếp và chúng bị mất đi.

Trên mỗi bộ xử lý lập trình quét, khi bậc thang chứa lệnh BTD là TRUE, bộ xử lý chuyển trường bit của từ nguồn đến từ đích. Chuyển dữ liệu trong một từ, người lập trình chọn cùng một địa chỉ từ của hai nguồn và đích, như hình 4... Trong ví dụ này, 4 bit được chuyển từ phía tay trái (bit 00 đến 03) của từ N70:00 đến giữa của từ (bit 08 đến 11).



Hình 4 ...Lệnh BTD

### Lệnh MOVE và MASKED MOVE

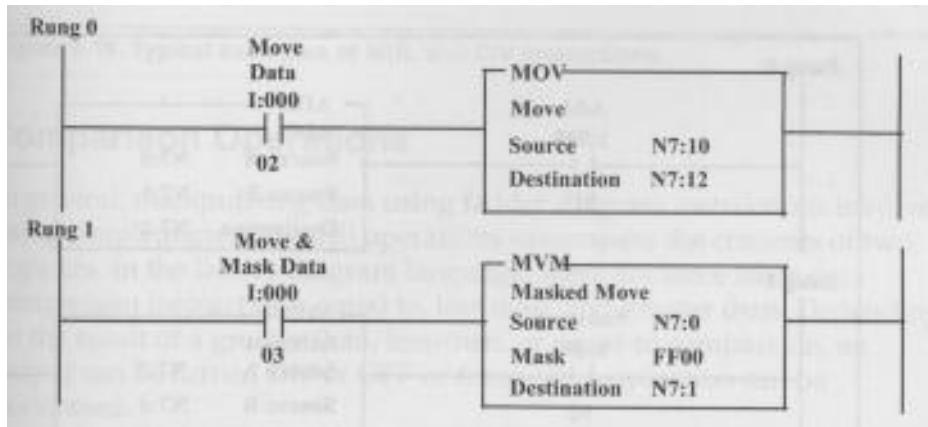
Lệnh chuyển MOV là lệnh ra mà nó cõi pi dữ liệu trên địa chỉ nguồn đến địa chỉ đích. Bậc thang duy trì được giá trị TRUE bao nhiêu thì lệnh MOV dịch chuyển các nội dung của mỗi chu kỳ quét của PLC đến địa chỉ đích. Nguồn có thể lập trình không đổi hay địa chỉ dữ liệu trên đó lệnh này có thể đọc được ảnh của giá trị. Đích của dữ liệu là địa chỉ dữ liệu sẽ được ghi vào như kết quả của phép tính. Lệnh này viết chồng lên một dữ liệu được lưu nào đó tại đích. Ví dụ trên bậc 0, khi đầu vào I:000/02 là TRUE, dữ liệu lưu tại địa chỉ N7:10 được copy và ghi vào vị trí N7:12.

Lệnh MVM là lệnh ra mà nó cõi pi nguồn gửi đến đích và cho phép một phần dữ liệu bị che (cản lại, không cho qua). Bậc thang giữ được TRUE càng lâu bao nhiêu thì lệnh này dịch chuyển dữ liệu mỗi lần quét.

Lệnh MVM có thể được sử dụng để cõi pi bảng ảnh vào/ra, giá trị nhị phân hay số nguyên. Ví dụ nó được sử dụng để tách dữ liệu của bit, như là bit trạng thái hay bit điều khiển, từ một địa chỉ chứa bit dữ liệu hay từ dữ liệu. Nguồn thông tin là chương trình không thay đổi hay địa chỉ dữ liệu từ nơi mà lệnh này sẽ đọc giá trị của ảnh. Nguồn thông tin được giữ không thay đổi.

Mặt nạ hay lá chắn có thể có địa chỉ hay giá trị hệ cơ số 16 để chỉ rõ bit bị chấn lại hay cho đi qua. Người lập trình phải bật bit bị che lên giá trị 1 để cho dữ liệu đi đến đích. Dữ liệu được di chuyển sẽ ghi chồng lên dữ liệu trên đích. Đích là địa chỉ dữ liệu để trên đó lệnh này ghi kết quả của phép tính. Lệnh này ghi chồng lên dữ liệu lưu tại đích đến.

Trên hình 4.. chỉ một ví dụ của hai lệnh MOVE và MVM. Trên bậc 0, khi bit đầu vào I:000/02 là TRUE, nội dung của vị trí nhớ N7:10 được chuyển đến vị trí từ số nguyên N7:12. Trên bậc 1, khi bit vào I:000/03 là TRUE, nội dung của tám bit cao hơn của từ nhớ N7:10 được di chuyển đến địa chỉ từ số nguyên N7:12 từ lúc tám bit cao hơn của mặt nạ (FF00) chứa toàn số 1. Tám bit thấp hơn bị khoá lại, bởi vì tám bit thấp hơn của mặt nạ là các số 0.

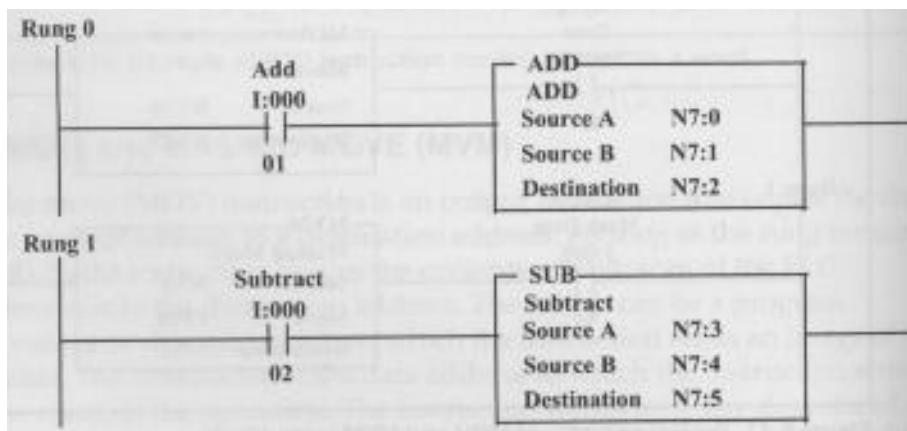


## CÁC PHÉP TÍNH SỐ HỌC

Các phép tính lô gíc gồm bốn phép tính cơ bản là: cộng, trừ, nhân và chia. Các lệnh này sử dụng nội dung trong vị trí của hai từ để thực hiện chức năng yêu cầu. Các lệnh số học là các lệnh ra, mà chúng có thể hay không có thể lệnh vào lô gíc. Các lệnh này sử dụng hoặc một hay hai từ dữ liệu để lưu kết quả. Lệnh cộng hay trừ sử dụng một từ. Lệnh nhân hay chia cần hai từ để tính kết quả. Chúng ta sẽ nói về từng lệnh ở các mục tiếp theo.

### Phép cộng (ADD)

Lệnh cộng thực hiện phép cộng của hai giá trị lưu trong hai vị trí khác nhau của bộ nhớ, nguồn A và nguồn B. Kết quả được lưu trữ trong bộ ghi đích đến. Nguồn A và nguồn B có thể là giá trị hay địa chỉ có chứa các giá trị. Trong ví dụ trên hình 4., nếu đầu vào trên bit I:000/01 là TRUE, lệnh cộng giá trị lưu trong địa chỉ N7:0 vào giá trị lưu trong N7:1 và lưu kết quả lên địa chỉ N7:2.



### Phép trừ SUB

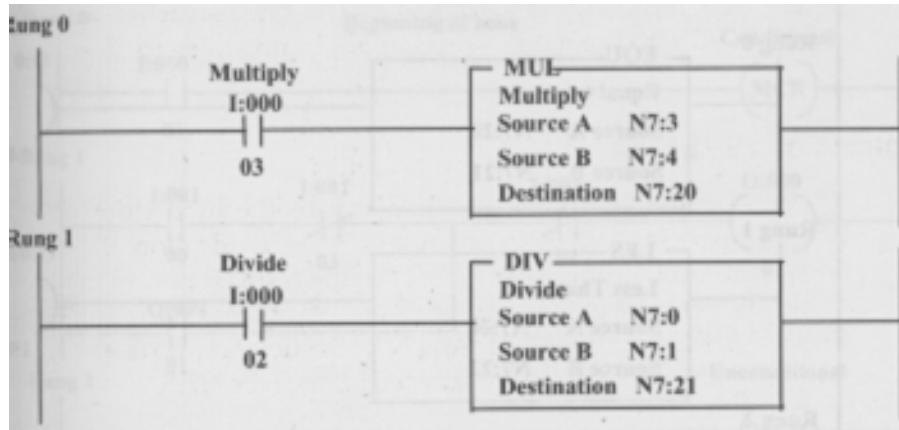
Lệnh trừ thực hiện phép tính trừ của hai số trong hai địa điểm nguồn A và nguồn B. Như trong phép cộng, nếu điều kiện cho phép thực hiện phép trừ bật lên 1, kết quả được đưa đến bộ ghi trên đích đến. Quan sát trên bậc 1 của hình 4.., nếu bit vào tại địa chỉ I:000/02 được bật lên 1, số trên địa chỉ N7:4 được trừ đi số trên địa chỉ N7:3 và kết quả đưa vào bộ ghi N7:5.

### **Phép nhân MUL**

Lệnh nhân được sử dụng để nhân một giá trị (nguồn A) với một giá trị khác (nguồn B) và kết quả đưa vào đích đến. Nguồn A và nguồn B có thể là giá trị hay địa chỉ. Lệnh nhân đặc trưng MUL được minh họa trên hình 4.., nếu bit vào I:000/03 trên bậc 0 là TRUE, bộ xử lý sẽ nhân giá trị trong N7:3 với giá trị trong N7:4 và lưu kết quả trong N7:20.

### **Phép chia DIV**

Lệnh chia được sử dụng để chia một giá trị (nguồn A) bởi một giá trị khác (nguồn B) và lưu kết quả trên đích đến. Nguồn A và nguồn B có thể là giá trị hay địa chỉ. Trong lệnh chia như trên ví dụ 4.., nếu bit vào I:000/02 trên bậc 1 là TRUE, bộ xử lý sẽ chia giá trị trong N7:0 cho giá trị trong N7:1 và kết quả lưu trong N7:21.



Hình 4..

## **CÁC PHÉP SO SÁNH DỮ LIỆU**

Nói chung việc di chuyển dữ liệu sử dụng các lệnh sơ đồ thang gây ra sự sử dụng của một bộ ghi đơn để so sánh nội dung của hai bộ ghi. Trong ngôn ngữ sơ đồ thang, chúng có ba lệnh so sánh là bằng, nhỏ hơn hay lớn hơn. Phụ thuộc vào kết quả của phép so sánh lớn hơn, nhỏ hơn hay bằng, một đầu ra có thể được bật lên (ON) hay tắt đi (OFF) hay một phép tính khác có thể được thực hiện.

### **So sánh bằng EQU**

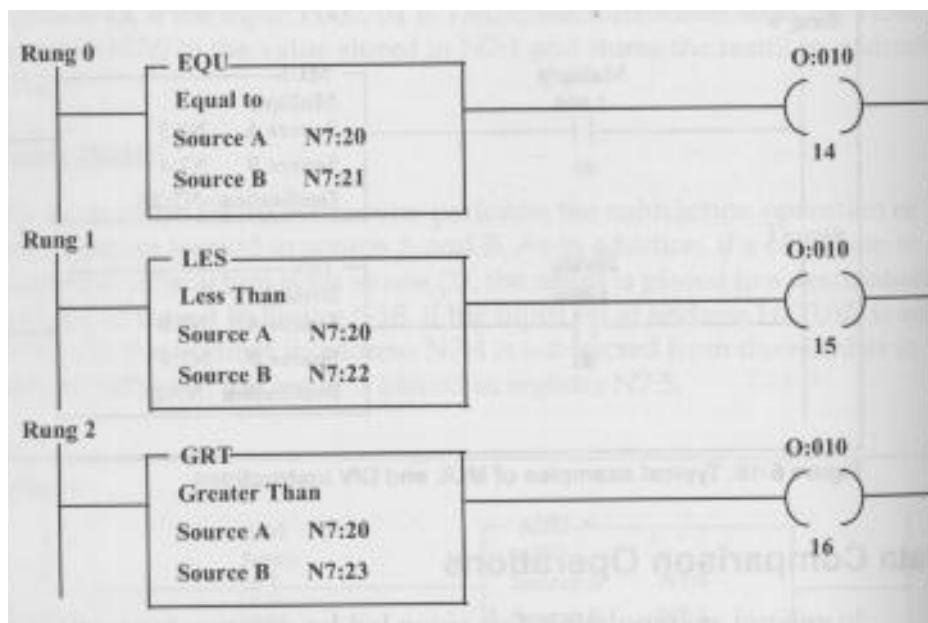
Lệnh này được dùng để kiểm tra hai giá trị bằng nhau hay không. Nguồn A và nguồn B có thể là giá trị hay địa chỉ chứa các giá trị. Ví dụ trên bậc 0 của hình 4.., nếu sự bằng nhau là TRUE, thì đầu ra được kích hoạt.

### **So sánh nhỏ hơn LES**

Giống như lệnh EQU, lệnh LES kiểm tra nội dung của giá trị trong một vị trí (nguồn A) để xem nếu nó nhỏ hơn giá trị lưu trong vị trí thứ hai (nguồn B). Nếu điều kiện kiểm tra là TRUE, đầu ra cuộn hút trên bậc thang 1 được kích hoạt.

### So sánh lớn hơn GRT

Lệnh so sánh lớn hơn hoạt động như lệnh LES, với cá biệt là kiểm tra được thực hiện cho giá trị lớn hơn điều kiện là TRUE, đầu ra cuộn hút trên bậc thang 2 sẽ được kích hoạt.



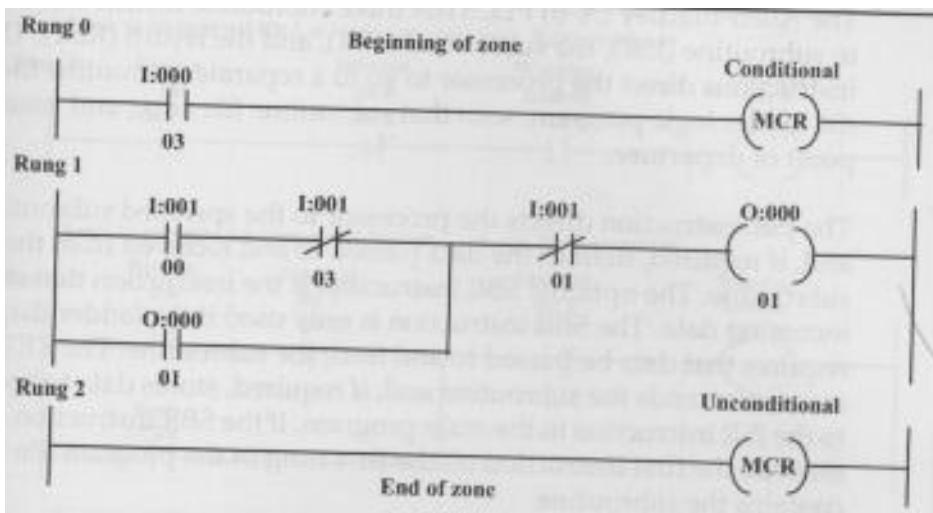
## CÁC LỆNH ĐIỀU KHIỂN

Các hàm của chương trình điều khiển để thực hiện một chuỗi các lệnh nhảy có điều kiện hay không có điều kiện và lệnh khứ hồi. Các lệnh này cho phép chương trình để thực hiện chỉ một số đoạn của lô gíc điều khiển nếu tập hợp cố định các điều kiện được đảm bảo. Các lệnh này diễn tả trêm các phần tiếp theo.

### Lệnh điều khiển rơ le chủ MCR

Lệnh rơ le chủ được dùng theo cặp để kích hoạt hay dừng thực hiện của nhóm hay một vùng của các bậc thang. Lệnh điều kiện MCR được dùng cùng với lệnh cuộn hút MCR không điều kiện để đưa hàng rào xung quanh một nhóm các bậc. Ví dụ, trên hình 4.., nếu đầu vào I:000/03 là đúng (TRUE), cuộn hút MCR có điều kiện trên bậc 0 sẽ được kích hoạt và lô gíc bên trong của vùng này sẽ được thực hiện tương ứng với lô gíc trên từng bậc trong vùng MCR. Nếu lệnh MCR có điều kiện tắt (OFF), tất cả các đầu ra không lưu bên trong vùng này sẽ bị tắt.

Các bậc bên trong vùng MCR còn được quét, nhưng thời gian quét của PLC sẽ giảm, bởi vì trạng thái sai của các đầu ra không lưu. Các đầu ra không lưu được bật lại khi bậc của chúng đi đến sai (FALSE).



### Lệnh nhảy JMP và lệnh nhãn LBL

Lệnh nhảy JMP và lệnh nhãn LBL là các lệnh được sử dụng theo cặp để bỏ qua một đoạn của chương trình lô gíc thang. Lệnh nhảy cho phép sự thực hiện kế tiếp bình thường được thay đổi để CPU nhảy đến vị trí mới trên chương trình thang. Nếu lô gíc của nhảy là TRUE, thì cuộn hút nhảy JMP ra lệnh cho CPU nhảy đến và thực hiện bậc được gán địa chỉ trên cùng như địa chỉ cuộn nhảy. Điều này cho phép chương trình thực hiện các bậc mà đầu ra của dòng kế tiếp bình thường của chương trình sơ đồ thang tiêu chuẩn. Mục đích của của nhãn để nhận dạng bậc thang mà nó sẽ là địa chỉ đích của lệnh nhảy. Nhãn tham chiếu phải khớp với nơi mà lệnh nhảy sẽ được sử dụng. Lệnh nhãn không đóng góp cho tính liên tục lô gíc, và nó luôn là lô gíc TRUE. Lệnh này được bố trí như điều kiện lô gíc đầu tiên trong bậc. Lệnh nhãn tham chiếu bởi một địa chỉ duy nhất, địa chỉ có thể được định nghĩa chỉ một lần trong chương trình.

### Lệnh nhảy đến chương trình con JST

Các chương trình con được sử dụng trong lập trình để sản xuất một chương trình có cấu trúc cao hơn và để giảm số lượng bộ nhớ được dùng cho chương trình. Các chương trình con được sử dụng để lưu các hàm lô gíc được tái xuất hiện mà có thể truy cập vào từ các phần khác nhau của chương trình lô gíc sơ đồ thang chính. Điều này tiết kiệm không gian bộ nhớ bởi vì hàm được lập trình chỉ là một mặc dù nó được sử dụng nhiều lần trong chương trình điều khiển.

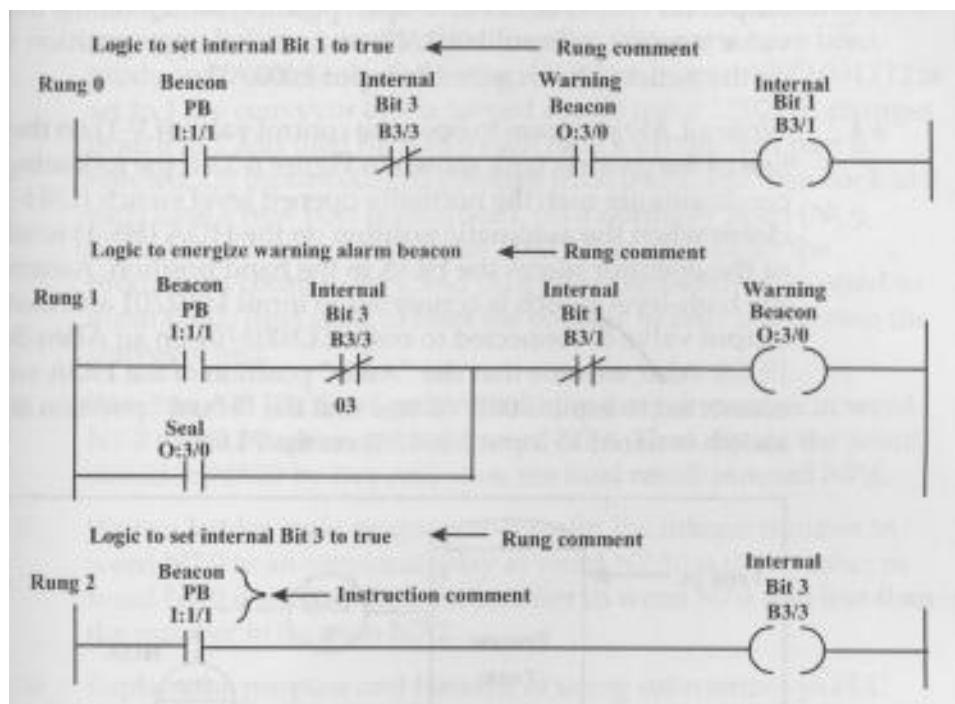
PLC S5 của Allen – Bradley có ba lệnh chương trình con: lệnh nhảy đến chương trình con JSR, chương trình con SBR và khứ hồi RET. Các lệnh này hướng bộ xử lý đi đến một tệp chương trình con riêng biệt trong chương trình lô gíc thang, quét tệp chương trình con này một lần, và quay trở lại điểm xuất phát.

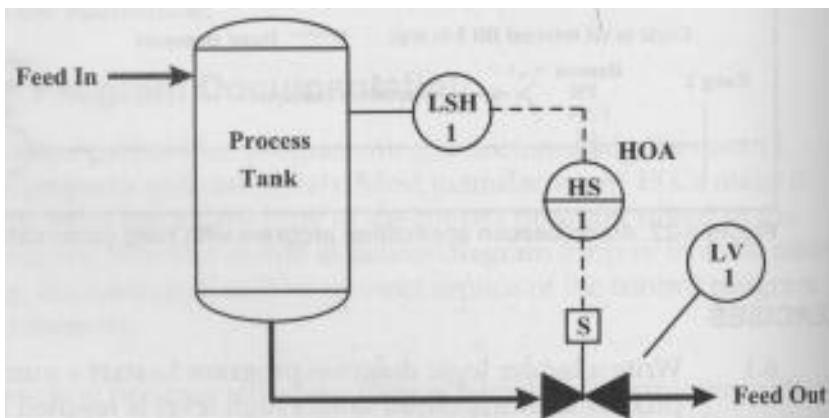
Lệnh JSR hướng bộ xử lý đến tệp chương trình con đặc biệt và nếu yêu cầu, định nghĩa dữ liệu đi qua và nhận được từ chương trình con. Lệnh phụ SBR là lệnh chứa các dữ liệu đến. Lệnh SBR chỉ được sử dụng nếu sơ đồ thang yêu cầu rằng dữ liệu được đi qua đến và đi từ chương trình con. Lệnh RET kết thúc chương trình con và nếu yêu cầu, lưu dữ liệu để được khứ hồi về lệnh JSR trong chương trình chính. Nếu lệnh SBR được dùng, nó phải chứa chương trình con.

## TÀI LIỆU CỦA CHƯƠNG TRÌNH ĐIỀU KHIỂN PLC

Một phần quan trọng của lập trình PLC là tài liệu của chương trình điều khiển phải đúng và hoàn chỉnh. Phần lớn các nhà sản xuất PLC đã làm cho khả năng in chương trình cốt pi “cứng” của chương trình điều khiển ra từ bộ nhớ của PLC. Lưu trong sơ đồ thang hay trong một ngôn ngữ khác, bản cốt pi “cứng” sẽ được xem như bản sao chính xác của chương trình điều khiển lưu trong bộ nhớ.

In ra bản cốt pi cứng sẽ chỉ ra mỗi lệnh được lập trình với các địa chỉ có liên quan của mỗi đầu vào và đầu ra. Mặc dù vậy, thông tin chỉ thị chức năng hay mục đích của mỗi thiết bị hiện trường hay bit điều khiển bên trong hay lệnh la không sẵn sàng xuất hiện. Tài liệu phụ trợ thường được yêu cầu. Phần lớn các nhà sản xuất PLC cấp chương trình tài liệu phần mềm cho phép thiết bị lập trình, nói chung là máy tính cá nhân, có thể vào vị trí nhãn hay chú giải cho từng phân tử hay lệnh của chương trình. Thiết bị điều khiển PLC thường lưu chương trình đã được kiểm xét lần cuối vào trong bộ nhớ. Trước khi kiểm tra chương trình trực tuyến, người sử dụng có thể in chương trình đã xem xét này ra. Trong quá trình khởi động và thử nghiệm, các thay đổi thường xuyên được thực hiện bởi chương trình. Người sử dụng có thể tham khảo chúng ngay lập tức với bậc và chú thích của các lệnh. Đó là thực nghiệm tốt để có được bản cốt pi “cứng” cuối cùng của chương trình PLC trước khi thực hiện bảo trì trên hệ thống.





## LẬP TRÌNH SƠ ĐỒ THANG NÂNG CAO

### Giới thiệu

Các lệnh của sơ đồ thang nâng cao được yêu cầu để thực hiện các chức năng mạnh hơn là điều khiển ON/OFF, đếm thời gian, đếm và di chuyển dữ liệu. Các lệnh nâng cao này được sử dụng để điều khiển tương tự, để tính toán các tệp dữ liệu, các phép tính kế tiếp, báo cáo dữ liệu, các hàm lô gíc phức tạp, và các chức năng khác mà các lệnh LAD cơ bản không thể thực hiện được.

### CÁC LỆNH LAD NÂNG CAO

Các lệnh LAD nâng cao cho phép người sử dụng lập trình những hàm điều khiển PLC phức tạp hơn. Ta sẽ nói về các lệnh nâng cao chung nhất như tệp, bộ ghi cao hơn, sự kế tiếp, sự truyền khối của PLC Allen Bradley S5.

#### Các lệnh tệp (file)

Một tệp là nhóm các từ của bảng dữ liệu liên tiếp được sử dụng để chứa thông tin của PLC. Lệnh tệp được sử dụng để thực hiện các phép tính như số học, lô gíc, tìm kiếm, cối pi, và so sánh. Các lệnh tệp của PLC 5 Allen Bradley gồm tệp số học và lô gíc FAL, tệp tìm kiếm và so sánh FSC, tệp cối pi COP, và tệp điền đầy FLL.

Trên hình 4... minh họa cấu trúc lệnh FAL đặc trưng với các tham số điều khiển, chiều dài, vị trí, chế độ, đích đến, và thể hiện.

Bộ xử lý thông tin này để thực hiện lệnh. Điều khiển là địa chỉ của cấu trúc điều khiển trong một tệp điều khiển kiểu R. Độ dài là số từ (0 đến 999) trong khối dữ liệu mà trên đó lệnh tệp hoạt động. Vị trí là phần tử tức thời với khối dữ liệu mag bộ xử lý đang truy cập. Chế độ là số phần tử của tệp hoạt động trên mỗi lần khi bậc được quét trong chương trình. Có ba chế độ là :

- Chế độ toàn bộ,
- Chế độ số,
- Chế độ gia tăng.

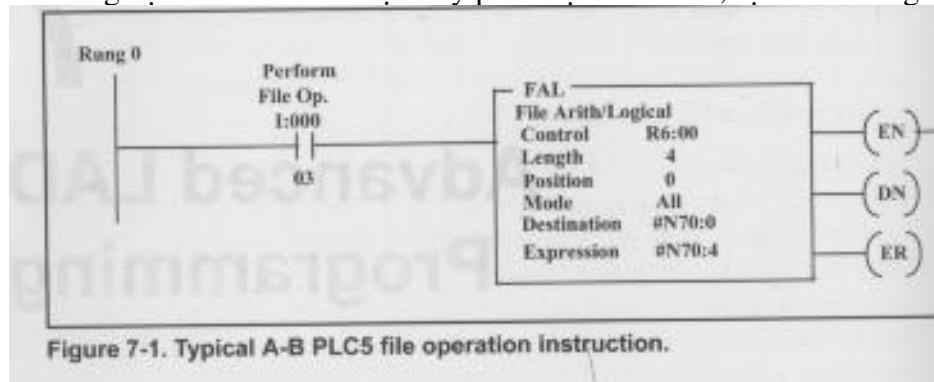
Trong chế độ toàn bộ, cả tệp được hoạt động trước khi bộ xử lý tiếp tục bậc tiếp theo của chương trình. Chế độ số phân phối phép tính của tệp lên trên số lượng vòng quét chương trình. Chế độ gia tăng thao tác với một từ của tệp mỗi khi bậc thang đị từ sai FALSE sang đúng TRUE. Đích đến là địa chỉ, nơi bộ xử lý lưu kết quả của phép tính. Lệnh này chuyển thành dạng dữ liệu đặc biệt bởi địa chỉ đích đến. Sự thể hiện chứa các

địa chỉ, các hằng số của chương trình, các phép tính mà nó xác định nguồn của dữ liệu và các phép tính sẽ được thực hiện.

Lệnh đầu ra cuộn hút bên phải của lệnh tệp là bit cho phép (EN – enable), bit thực hiện xong (DN – done), bit lỗi (ER – error). Các bit này có cùng địa chỉ từ như lệnh điều khiển. Bộ xử lý tự động bật địa chỉ của các bit trạng thái khi người lập trình nạp địa chỉ điều khiển. Bit cho phép EN được bật từ FALSE lên TRUE để chuyển bậc, và nó chỉ thị rằng lệnh này đã được cho phép. Trong chế độ gia tăng, bit EN theo dõi điều kiện của bậc thang. Trong chế độ số và chế độ toàn bộ, bit EN giữ trạng thái đặt cho đến khi lệnh này hoàn tất các phép tính của nó, không cần quan tâm đến điều kiện của bậc. Bit cho phép được đặt lại khi bậc chuyển sang trạng thái FALSE, và lệnh này sau đó hoàn tất phép tính của nó.

Bit thực hiện xong DN được đặt sau khi lệnh đã hoạt động trên nhóm cuối cùng của từ. Trong chế độ gia tăng, nếu lệnh này là FALSE tại lúc hoàn thành, lệnh này có được bộ xử lý bật lại bit DN để quét một chương trình sau khi hoàn thành phép tính của nó. Nếu lệnh này là TRUE tại lúc hoàn thành, bit hoàn thành được đặt lại khi lệnh này chuyển sang trạng thái FALSE.

Bí lỗi ER được đặt khi phép tính gây ra trạng thái tràn. Lệnh dừng đến khi chương trình sơ đồ thang đặt lại bit lỗi. Khi bộ xử lý phát hiện ra các lỗi, vị trí lưu trong số lượng từ bị lỗi.



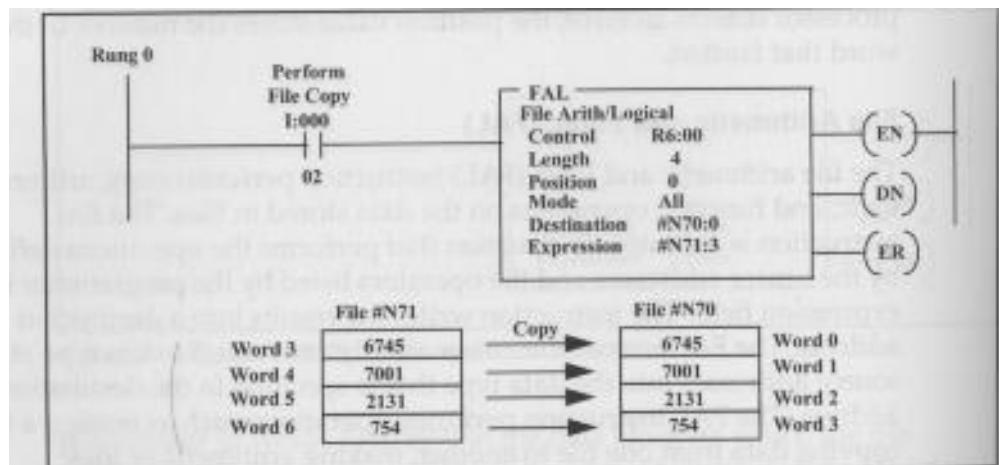
### Tệp số học và lô gíc

Lệnh tệp FAL thực hiện c蘼 pi, các phép tính số học và các hàm trên dữ liệu lưu trên tệp. Lệnh FAL là một lệnh đầu ra, để thực hiện các phép tính định nghĩa bởi địa chỉ nguồn và các phép tính liệt kê bởi người lập trình trong trường thẻ hiện. Lệnh này ghi kết quả đến địa chỉ đích. Lệnh FAL biến đổi dạng dữ liệu trên địa chỉ nguồn thành dạng dữ liệu mà nó xác định trên địa chỉ đích đến. Lệnh FAL thực hiện các phép tính như xoá một tệp, c蘼 pi dữ liệu từ một tệp sang tệp khác, thực hiện tính toán số học và lô gíc trên dữ liệu lưu trong tệp, và tách tệp mã lỗi mỗi lần để hiển thị lên màn hình. Bảng 4.. liệt kê các phép tính thực hiện bởi lệnh FAL trên A-B PLC 5.

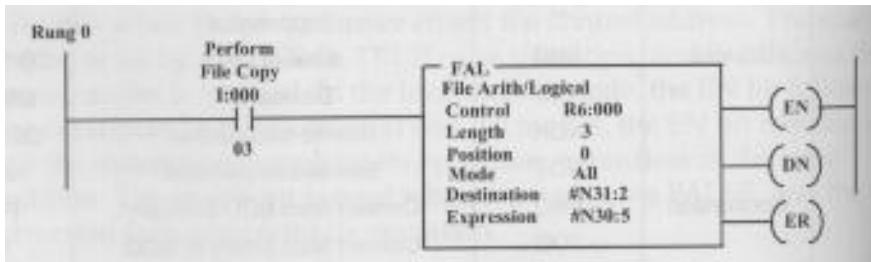
Table 7-1. FAL Operators for A-B PLC5 Family

Type	Operator	Description	Example
Copy	none	Copy from A to B	
Clear	none	Set a value to 0	
Arithmetic	+ - * / -	Add Subtract Multiply Divide Negate	2+2 8-5 3*6 12/4 -N7:0
	SQR **	Square root Exponential	SQR N7:1 10**2
Bitwise	AND OR XOR NOT	Bitwise AND Bitwise OR Bitwise exclusive or Bitwise complement	D9:3 AND D10:4 D9:4 OR D9:5 D9:6 XOR D9:7 NOT D10:11
Conversion	FRD TOD	Convert from BCD to binary Convert from binary to BCD	FRD D10:0 TOD N7:1

Để minh họa các lệnh FAL hoạt động như thế nào, ta thực hiện một lệnh copy pi như hình 4.. Trong ví dụ này, khi bit chuyển sang trạng thái TRUE (bit I:000/02 bật lên 1), bộ xử lý đọc dữ liệu lưu trong 4 từ của tệp số nguyên N71, bắt đầu trên từ số 3. Sau đó nó ghi dữ liệu lên tệp số nguyên N70 bắt đầu trên từ 0. Nó ghi chồng lên dữ liệu trên tệp đích đến.



Ví dụ 1: Viết chương trình PLC 5 LAD để copy pi dữ liệu của một tệp số nguyên N30, từ số 5, từ số 4 và từ số 7 đến tệp N31 bắt đầu với từ số 2, nếu bit vào I:000/03 là TRUE.  
Giải: Chương trình LAD copy pi dữ liệu như trên hình 4..



### Tệp tìm và so sánh FSC

Lệnh tìm kiếm và so sánh là lệnh đầu ra mà nó so sánh giá trị trong tệp nguồn, từ với từ, để thực hiện phép tính xác định trên hình thức thể hiện. Khi bộ xử lý tìm ra sự so sánh đúng TRUE, nó bật bit tìm thấy FD (found) và ghi nhận vị trí, nơi mà sự so sánh đúng được tìm thấy. Bit ngăn IN (inhibit) được tạo ra để ngăn sự tìm kiếm tệp này trong tương lai.

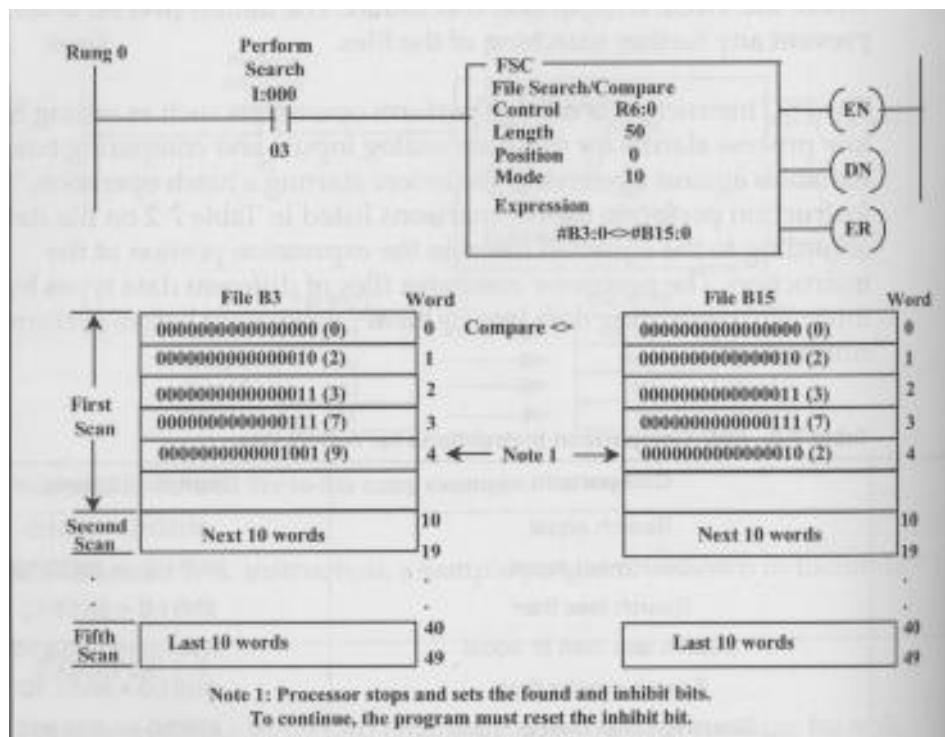
Lệnh FSC được sử dụng để thực hiện các phép tính như đặt chế độ báo động mức cao và mức thấp cho quá trình công nghệ với nhiều đầu vào tương tự và so sánh các khối biến với tệp tham chiếu trước khi bắt đầu phép tính khối. Lệnh FSC thực hiện các phép so sánh liệt kê trên bảng 4.. tóm tắt liệu của tệp tương ứng với phương trình được kê trong phần của hình thức thể hiện của lệnh. Bộ xử lý so sánh các tệp của các dạng dữ liệu khác nhau bởi biến đổi dữ liệu bên trong thành dữ liệu nhị phân tương đương trước khi so sánh.

Comparison	Sample Expression
Search equal	#N50:0 = #N51:0
Search not equal	#N51:0 <> #N53:10
Search less than	#N51:0 < #N53:10
Search less than or equal	#N51:0 <= #N53:10
Search greater than	#N51:0 > #N53:10
Search greater than or equal	#N51:0 >= #N53:10

Trong tìm tệp, khi bậc thang điều kiện là TRUE, sự so sánh yêu cầu được thực hiện trên dữ liệu được địa chỉ hóa trên hình thức thể hiện. Từ được so sánh thứ tự theo chiều tăng tại lúc bắt đầu. Tốc độ được xác định bởi chế độ của phép tính xác định trong lệnh FSC. Bit hoàn thành DN được bật sau khi bộ xử lý đã so sánh cặp cuối cùng. Nếu trên bậc này là TRUE khi hoàn thành, bit hoàn thành được tắt khi bậc thang không ở trạng thái TRUE lâu hơn. Trong chế độ số, mặc dù bậc không phải là TRUE tại lúc hoàn thành, bit DN được giữ trên một vòng quét chương trình sau khi phép tính đã hoàn thành.

Để minh họa lệnh FSC hoạt động như thế nào, ta thực hiện một lệnh FSC “tìm không bằng”, như trên hình 4.. Khi bit I:000/03 trở thành TRUE, bộ xử lý thực hiện phép so sánh “không bằng” giữa các từ, bắt đầu tại B3:0 và B15:0. Số từ được so sánh trên một vòng quét chương trình là 10 trong ví dụ trên, bởi vì chế độ được bật là 10.

Khi bộ xử lý tìm thấy các từ nguồn tương ứng không bằng nhau (B3:4 và B15:4), nó dừng tìm và bật bit tìm thấy FD và bit ngăn IN. Để tiếp tục so sánh, chương trình lô gíc thang phải bật bit ngăn về OFF.



Ví dụ 2 : Viết chương trình LAD của PLC 5 để tìm dữ liệu trong tệp số nguyên N40, từ số 0 đến 99, và so sánh nó cho điều kiện bằng với dữ liệu trong tệp N50 bắt đầu cũng tại từ số 0, nếu bit đầu vào I:000/03 là TRUE.

Giải: Chương trình LAD được thể hiện trên hình 4..

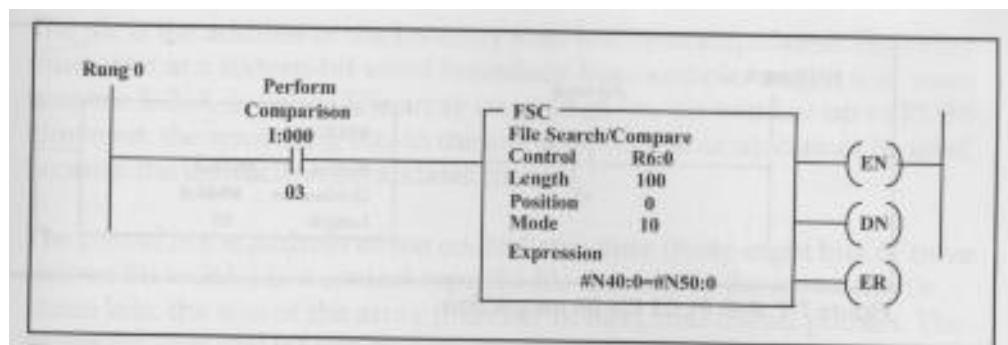
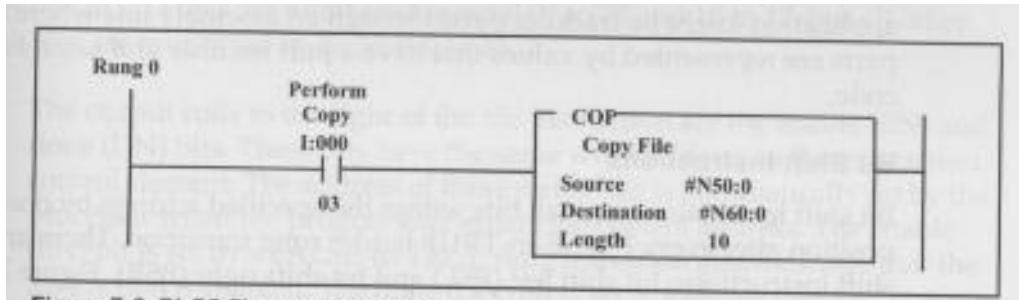


Figure 7-5. LAD program for Example 7-2.

### Lệnh cоп pi tệp COP

Lệnh này là lệnh đầu ra mà nó cоп pi giá trị trên tệp nguồn vào tệp đích. Tệp nguồn giữ không thay đổi. Lệnh COP không sử dụng bit trạng thái. Lệnh COP không ghi chông lên ranh giới của tệp, như vậy dữ liệu tràn sẽ bị mất. Cũng vậy, không có chuyển đổi dữ liệu nén tệp nguồn và tệp đích sử dụng cùng dạng dữ liệu.

Trên hình 4.. chỉ ra ví dụ của chương trình LAD sử dụng lệnh COP. Trong ví dụ này bit vào I:000/03 là TRUE, bộ xử lý sẽ cоп pi 10 từ đầu tiên tại tệp N50:0 đến 10 đầu tiên của tệp N60:0



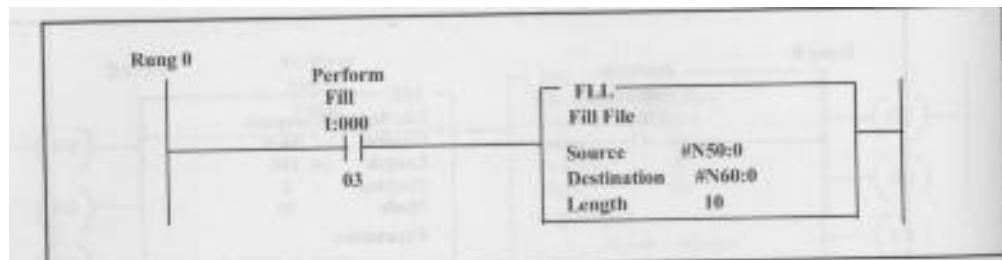
### Lệnh dién tệp FLL

Lệnh FLL là lệnh ra mà nó dién các từ của tệp với giá trị nguồn. Tệp nguồn giữ không thay đổi. Giống như lệnh COP, lệnh FLL không dung bit trạng thái. Lệnh FLL sẽ không ghi lên lề, và như vậy dữ liệu tràn sẽ bị mất. Cũng như vậy, không có sự chuyển đổi dữ liệu, dữ liệu nguồn và dữ liệu đích là cùng một dạng.

Trên hình 4.. lag minh họa của lệnh FLL. Khi bit vào I:000/03 là TRUE, bộ xử lý copy 10 từ bắt đầu từ tại tệp N50:0 đến 10 từ đầu tiên của tệp N60:0.

### CÁC LỆNH CHUYỂN BỘ GHI

Các lệnh chuyển bộ ghi được sử dụng theo dõi chuyển động hay dòng của các phân và thông tin trong các ứng dụng công nghiệp. Các lệnh này gồm: bit chuyển trái BSL, bit chuyển phải BSR, nạp đầu tiên vào đầu tiên ra FFL, xoá đầu tiên vào đầu tiên ra FFU.



Lệnh dịch chuyển bit BSR và BSL được sử dụng để nạp bit vào, bit dịch chuyển qua và bit xoá từ mảng bit với một bit bị xoá mỗi lần. Một ứng dụng là dùng theo dõi các chai trên dây chuyển đóng chai, trong đó mỗi bit thể hiện một chai. Lệnh dịch chuyển ghi và xoá FFL và FFU được dùng để ghi và xoá các giá trị trên cùng một thứ tự. Tương tự như vậy là có thể ứng dụng để theo dõi các chi tiết trên dây chuyển lắp ráp, nơi mà chi tiết được thể hiện bằng giá trị xác định bởi số của chi tiết và mã lắp ráp.

### Các lệnh dịch chuyển bit

Lệnh dịch chuyển dịch tất cả các bit trong một địa chỉ xác định bởi một vị trí bit sau mỗi lần bậc thang chuyển trạng thái từ FALSE sang TRUE. Có hai lệnh dịch chuyển là dịch chuyển một bit sang trái BSL và dịch chuyển một bit sang phải BSR. Trên hình 4.. chỉ ra cấu trúc của lệnh dịch chuyển bit sang trái BSL với các tham số của nó trong tệp, điều khiển, bit địa chỉ và độ dài.

*Tên tệp* là địa chỉ của mảng bit sẽ được dịch chuyển. Mảng này bắt đầu từ lề của của từ mười sáu bit. Ví dụ bit 0 của từ số 1, 2, 3 và v.v. Mảng có thể kết thúc tại một bit số đến 15,999. Mặc dù vậy, các bit còn lại trong từ cuối cùng của mảng không thể được sử dụng bởi vì lệnh này không có giá trị đối với chúng.

Tham số **điều khiển** là địa chỉ của cấu trúc điều khiển (bốn mươi tám bit hay ba từ mười sáu bit) trong điều khiển của dạng tệp (R) được chứa các bit trạng thái của lệnh, kích cỡ của mảng (số lượng bit), và bit con trỏ. Bộ xử lý sử dụng thông tin này để thực hiện lệnh. **Bit địa chỉ** là địa chỉ của bit nguồn. Lệnh dịch chuyển bit chèn giá trị 0 hay 1 của một bit vào bit khác bắt đầu từ vị trí bit thấp nhất cho BSL hay từ bit có vị trí cao nhất của mảng cho lệnh BSR.

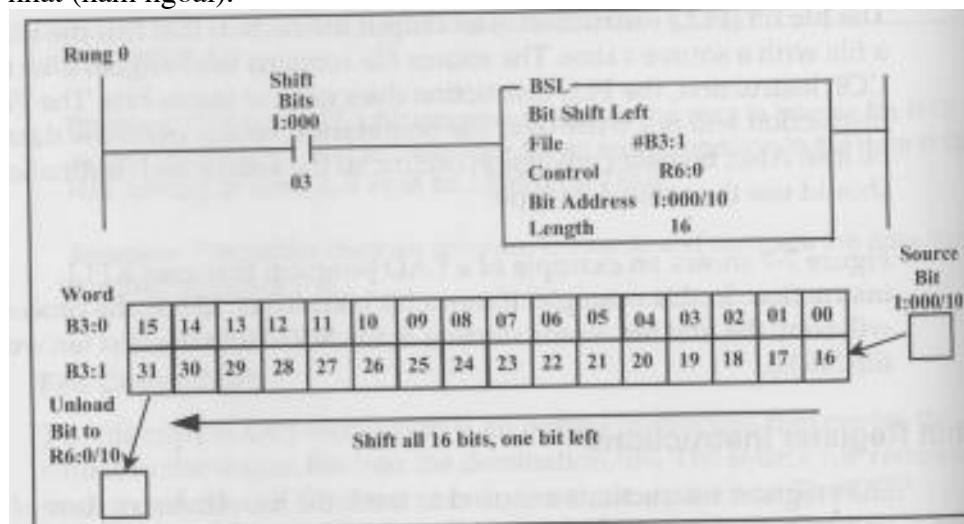
Tham số **độ dài** là số thập phân chỉ số lượng của các bit sẽ được dịch chuyển. Trong Allen Bradley PLC5, các bit trong tệp vào/ra được đánh số cơ số 8 từ 00 đến 07 và 10 đến 17, nhưng trong tất cả các tệp khác được đánh số thập phân từ 0 đến 15.

**Các đầu ra cuộn hút** của các lệnh tệp này là bit cho phép EN và bit hoàn thành DN. Các bit này có cùng một địa chỉ từ như là các phần tử điều khiển của lệnh. Các địa chỉ của các bit trạng thái được bật tự động bởi bộ xử lý khi người lập trình nhập vào các địa chỉ điều khiển. Bit cho phép EN được bật khi trạng thái của bậc thang chuyển từ FALSE sang TRUE và chỉ thị rằng lệnh được cho phép. Bit hoàn thành DN được bật để chỉ thị rằng mảng bit được dịch chuyển sang phải hay sang trái của một bit.

**Phân tử điều khiển** cũng chứa hai bit trạng thái: bit báo lỗi ER tại bit 11 và xoá tại UL tại bit 12. Bit báo lỗi ER được bật để chỉ thị rằng lệnh phát hiện ra lỗi, như lỗi vào với độ dài tệp âm. Bit chuyển UL là đầu ra của lệnh. Bit UL lưu trạng thái của bit đã được di chuyển từ mảng mỗi khi lệnh dịch chuyển được cho phép.

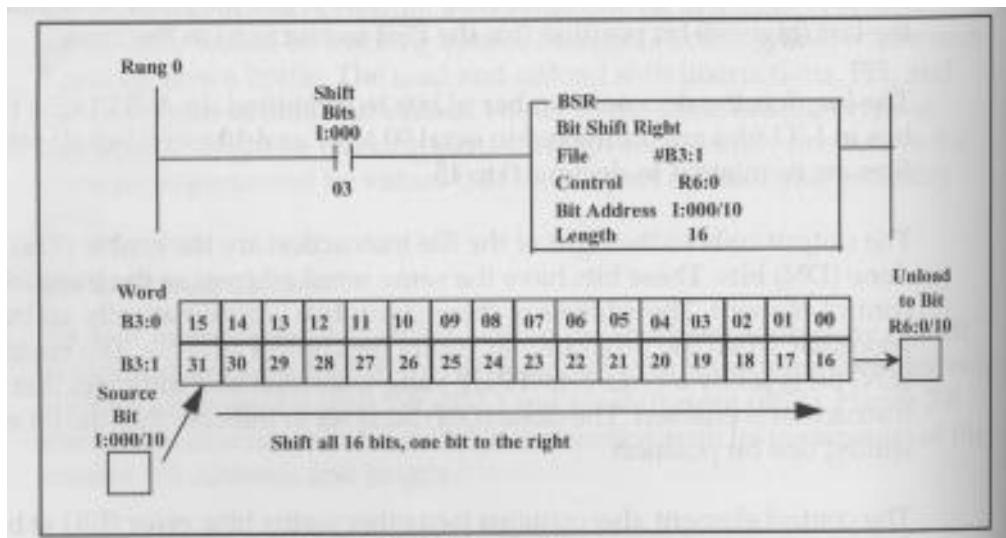
Trong chương trình LAD trên hình 4.. khi bậc thang có chứa lệnh BSL đi từ trạng thái FALSE sang TRUE, bộ xử lý bật bit EN. Bộ xử lý sau đó dịch chuyển 16 bit trong tệp bit B3, bắt đầu với bit 16, sang bên trái( bit có trọng lượng cao hơn) một vị trí. Bit cuối cùng dịch chuyển ra là bit trên vị trí bit 31, bit này được chuyển đến bộ ghi điều khiển UL, bit UL (R6:0/10). Bit nguồn đến từ địa chỉ I:000/10 và dịch chuyển đến vị trí đầu tiên của B3:1. Sau khi bộ xử lý hoàn tất việc dịch chuyển trong một chu kỳ quét, khi bit điều khiển của bậc thang I:000/03 trở thành FALSE, lệnh này bật lại các bit trạng thái EN, ER (nếu đã được bật) và DN.

Để đóng gói phép tính, thì địa chỉ nguồn được gán cùng một địa chỉ như là địa chỉ bit cao nhất (nằm ngoài).



Cấu trúc của phép tính dịch chuyển phải BSR được minh họa như trên hình 4.. với các tham số của tệp, điều khiển, địa chỉ của bit và độ dài. Lệnh này hoạt động tương tự như lệnh dịch chuyển bit sang trái., chỉ có khác là các bit dịch sang phải. Khi bậc thang

chứa lệnh BSR đi từ FALSE sang TRUE, bộ xử lý bật bit EN. Sau nó đó dịch chuyển mươi sáu bit trong tệp bit B3 sang phải (đến số bit thấp hơn) một vị trí, bắt đầu từ bit 31. Bit cuối cùng là bit 16 bị dịch ra ngoài vào bộ ghi điều khiển UL, bit (R6:0/10). Bit nguồn đến từ I:000/10 và dịch chuyển vào vị trí bit cao nhất, bit 31 của tệp B3.



Sau khi bộ xử lý hoàn tất phép dịch chuyển trong một vòng quét chương trình, khi bậc thang của bit điều khiển I:000/03 trở thành FALSE, lệnh này bật lại tất cả các bit trạng thái: bit EN, bit ER (nếu đã được bật) và bit DN. Để đóng gói phép tính, địa chỉ của nguồn được gán cùng địa chỉ như địa chỉ của bit cao nhất (nằm ra ngoài).

### Các lệnh vào đầu tiên – ra đầu tiên FIFO

Có hai lệnh thuộc nhóm này là lệnh nạp FIFO dùng nhớ FFL và lệnh di chuyển FIFO có sử dụng nhớ FFL. Hai lệnh này được sử dụng theo cặp để lưu và lấy dữ liệu theo một trật tự cho trước. Khi sử dụng theo cặp, chúng tạo ra một bộ ghi dịch chuyển không đồng bộ hay bộ ghi dồn.

Hai lệnh này phải sử dụng cùng một tệp và cùng một địa chỉ điều khiển, độ dài, và giá trị vị trí như trên hình 4..

Dữ liệu được nạp bắt đầu trên từ số 0 và tăng dần cho đến từ số n. Dữ liệu được bỏ đi từ từ số 0, và tất cả các từ được dịch chuyển lên một vị trí hướng vào từ số 0.

Tham số **nguồn** là địa chỉ mà nó sẽ lưu các giá trị vào tiếp theo của bộ ghi dồn. Lệnh FIFO-FFL lấy giá trị từ địa chỉ nguồn và nạp vào từ tiếp theo của bộ ghi dồn. Tham số **dịch đến** là địa chỉ mà nó lưu giá trị có trong bộ ghi dồn.

FIFO là địa chỉ được đánh chỉ số của bộ ghi dồn. Cùng một địa chỉ FIFO này được sử dụng cho cả lệnh FFL và FFU. Tham số **điều khiển** là địa chỉ của cấu trúc điều khiển (bốn mươi tám bit hay ba tám mươi sáu bit) trong miền điều khiển của bộ nhớ. Cấu trúc điều khiển lưu các bit trạng thái của lệnh, độ dài của bộ ghi dồn, và vị trí sẵn sàng kế tiếp (con trỏ) trong bộ ghi dồn.

Tham số **độ dài** xác định số lượng từ lớn nhất chứa trong bộ ghi dồn. Tham số độ dài được ghi địa chỉ bằng cách thêm từ “.LEN” trong bộ nhớ vào trong địa chỉ điều khiển (bộ ghi R).

Tham số **vị trí** chỉ thị vị trí kế tiếp sẵn sàng khi lệnh nạp dữ liệu vào bộ ghi dồn. Vị trí được ghi địa chỉ bằng cách thêm từ POS vào trong địa chỉ điều khiển (bộ ghi R). Thông thường giá trị vị trí là 0, nếu người lập trình không muốn lệnh bắt đầu từ một điểm mốc nào đó khi khởi động.

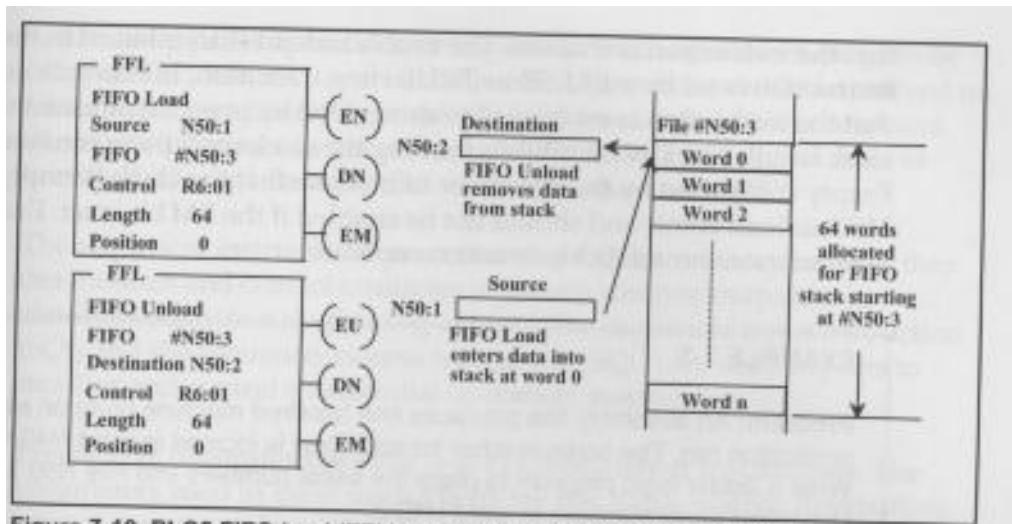
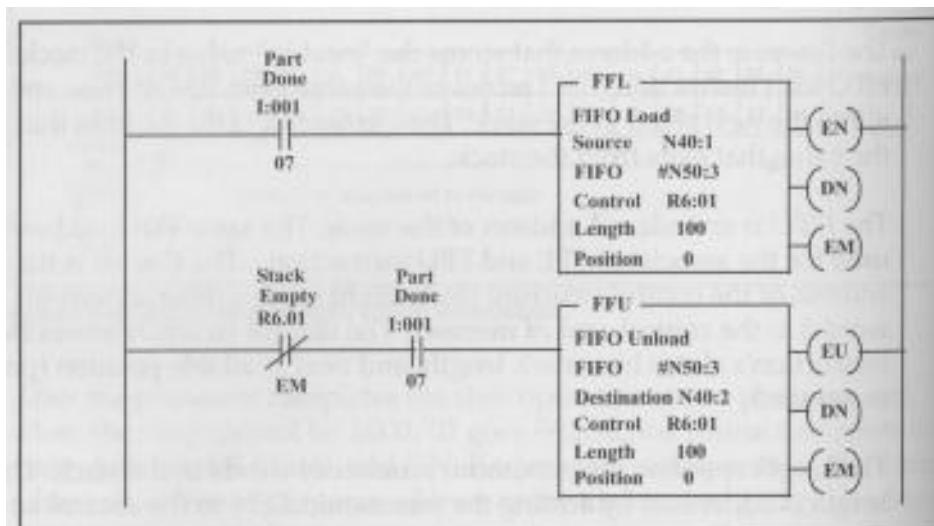


Figure 7.10. PLCS FIFO Load (FFL) and FIFO Unload (FFU)

Các đầu ra cuộn hút của các lệnh FIFO là bit cho phép nạp EN, bit cho phép chuyển dữ liệu EU, bit xoá EM (empty) và bit hoàn thành DN. Các bit này có cùng một địa chỉ từ như *phân tử điều khiển* của lệnh (R). Bộ xử lý tự động bật địa chỉ của các bit trạng thái khi người lập trình nhập địa chỉ điều khiển. Bit cho phép nạp EN được dùng trong lệnh FFL được bật khi có trạng thái chuyển tiếp của bậc thang từ FALSE sang TRUE. Điều này chỉ thi rằng lệnh này được phép thực hiện. Bit cho phép dịch chuyển EU được dùng trong lệnh dịch chuyển FFU khi bậc thang chuyển trạng thái từ FALSE sang TRUE. Bit này chỉ thi rằng lệnh đã được phép thực hiện. Bit hoàn thành DN được bật để chỉ thi rằng bộ ghi dồn đã đầy. Bit DN ngăn nạp tiếp vào bộ ghi dồn cho đến khi nó có chỗ. Bit xoá EM được bật bởi bộ xử lý để chỉ thi rằng bộ ghi dồn được xoá. Lệnh điều khiển dịch chuyển FIFO sẽ không cho phép thực hiện nếu bit EM được bật. Ví dụ minh họa trên hình 4..



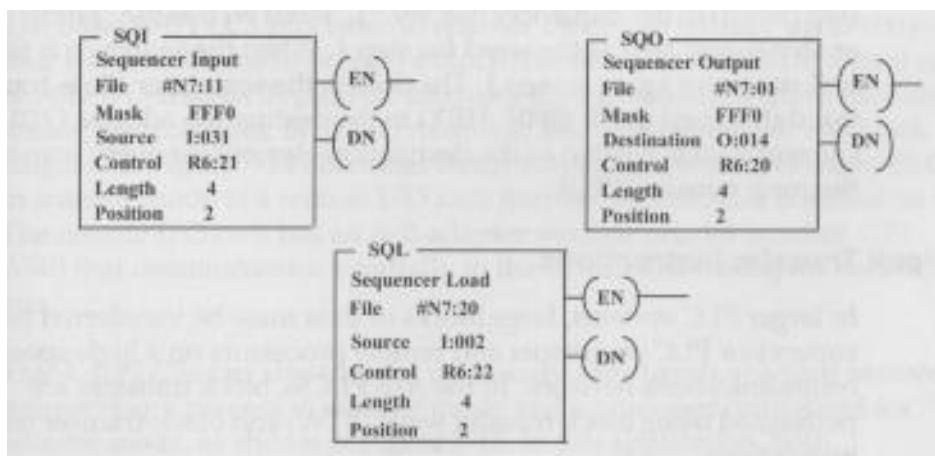
Ví dụ 3: Một dây chuyền lắp ráp có một trặc chi tiết máy trên mỗi vòng chạy. Số xê ri của mỗi chi tiết được lưu trong từ N40:1. Viết chương trình LAD để đưa số xê ri này vào tệp N50:3.. Hãy tải số xê ri này đến từ N40:2.

### Các lệnh kế tiếp

Các lệnh kế tiếp là các lệnh được sử dụng để điều khiển tự động các máy lắp ráp, mà các máy này có các thao tác không đổi và lặp lại. Có ba lệnh kế tiếp chung là: đầu vào kế tiếp, đầu ra kế tiếp và nạp kế tiếp.

Các lệnh kế tiếp nói chung được sử dụng để chuyển dữ liệu từ bộ nhớ đến mô đun ra gián đoạn để điều khiển các thao tác của quá trình gián đoạn hay các thao tác lô kế tiếp (đầu ra gián đoạn). Các lệnh này cũng được sử dụng để so sánh các dữ liệu của các từ vào/ra với dữ liệu lưu trong bảng nhớ mà các điều kiện hoạt động của quá trình có thể được kiểm tra để điều khiển hay để chẩn đoán (đầu vào kế tiếp). Lệnh này cũng được sử dụng để chuyển dữ liệu của từ vào/ra vào bộ nhớ (nạp kế tiếp).

Các lệnh kế tiếp có thể chiếm bộ nhớ chương trình bởi vì chúng có thể hiển thị và điều khiển nhiều đầu ra trong số mười sáu đầu ra gián đoạn, trên một bậc thang lô gíc đơn. Lệnh vào kế tiếp SQI và lệnh ra kế tiếp SQO được sử dụng theo cặp, để hiển thị và điều khiển một thao tác kế tiếp tương ứng. Trên hình 4.. là minh họa của ba lệnh kế tiếp của Alen Bradley PLC5. Các tham số dùng cho các lệnh này gồm *tệp, mặt nạ, nguồn, đích đến, điều khiển, độ dài và vị trí*. *Tệp* là tên địa chỉ được đánh số của cuat tệp kế tiếp, nơi mà lệnh này chuyển dữ liệu vào và ra từ bộ nhớ. Mục *đích* nó phụ thuộc vào bản thân mỗi lệnh. *Mặt nạ* đối với SQI và SQO là mã mười sáu hay địa chỉ của phần tử mặt nạ hay tệp qua đó lệnh này dịch chuyển dữ liệu. Bật bit mặt nạ lên 1 là cho dữ liệu đi qua; bật bit mặt nạ về 0 để ngăn lệnh điều hành tương ứng trên các bit đích. Người lập trình có thể xác định giá trị hệ cơ số mười sáu để có giá trị mặt nạ không đổi. Người lập trình cũng có thể lưu mặt nạ trên phần tử hay tệp nếu muốn thay đổi mặt nạ phù hợp cho các yêu cầu ứng dụng điều khiển. *Nguồn* đối với SQI và SQO là địa chỉ của phần tử đầu vào hay tệp từ đó lệnh này có được dữ liệu cho tệp kế tiếp của nó.

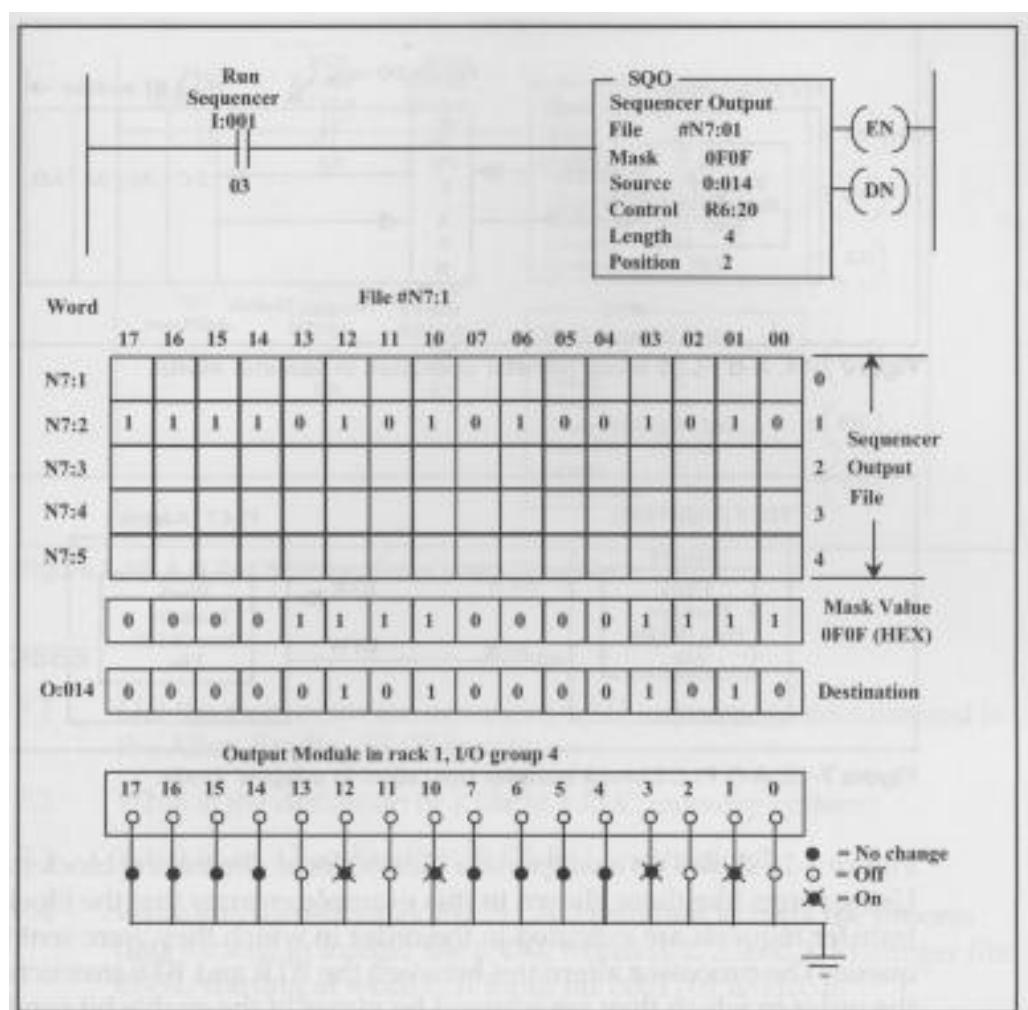


*Dích đến* chỉ cho SQO là địa chỉ đích của từ đầu ra hay tệp mà đến đó lệnh này sẽ chuyển dữ liệu tới từ tệp kế tiếp. *Điều khiển* là địa chỉ của cấu trúc điều khiển trong mảng điều khiển của bộ nhớ (bốn mươi tám bit hay ba từ mười sáu bit), để chứa các bit trạng thái của lệnh, độ dài của tệp kế tiếp, và vị trí tức thời trong tệp.

Người lập trình có thể sử dụng địa chỉ điều khiển với khả năng nhớ để ghi địa chỉ các thông số độ dài (LEN) và vị trí POS. Độ dài của tệp kế tiếp, và vị trí là vị trí tức thời của từ trong tệp kế tiếp để bộ xử lý sử dụng. Độ dài là số bước trong tệp kế tiếp bắt đầu từ vị trí 1. Vị trí 0 là vị trí khởi động. Lệnh này bật lại vị trí 1 tại mỗi lần thực hiện xong. Chú ý địa chỉ gán cho tệp kế tiếp là bước 0. Các lệnh kế tiếp sử dụng từ dữ liệu (độ dài +1) cho mỗi tệp tham chiếu trong lệnh này. Điều này cũng áp dụng cho nguồn, mặt nạ, và giá trị đích nếu chúng được ghi địa chỉ như các tệp.

Vị trí là vị trí của từ trong tệp kế tiếp. Giá trị của nó được tăng nội hàm bởi lệnh SQO và SQI. Để minh họa các lệnh kế tiếp hoạt động như thế nào, ta xem ví dụ lệnh ra kế tiếp SQO như trên hình 4..

Lệnh SQO chuyển dữ liệu của bước đang thực hiện (1) qua mặt nạ đến từ đầu ra nối đến



mô đun ra của Alen – Bradley PLC5, trên giá đỡ số 1, nhóm ra O số 4. Lệnh SQO bước qua tệp kế tiếp với các từ mười sáu đầu ra mà các bit đã được bật để điều khiển các thiết bị đầu ra khác nhau nối trên mô đun ra gián đoạn. Khi bậc thang đi từ FALSE sang TRUE, lệnh

này tăng lên bước tiếp theo (tù tiếp theo) trong tệp kế tiếp: #N7:1. Từ 7:1 là vị trí “home” hay bước 0, và N7:2 là từ cho bước 1. Khi chuỗi kết thúc, nó lại bắt đầu lặp lại từ bước 1. Dữ liệu trong tệp kế tiếp được dịch chuyển qua mặt nạ cố định (F0F0,HEX) để tới địa chỉ đích O:014. Dữ liệu tức thời được ghi đến phần tử đích cho mỗi chu kỳ quét trong lúc bậc thang duy trì giá trị TRUE.

### Lệnh chuyển khối

Trong các hệ PLC cỡ lớn, các khối dữ liệu lớn phải được truyền giữa Bộ xử lý của PLC giám sát và bộ xử lý từ xa trên mạng truyền thông cao tốc. Trong PLC5 của Allen Bradley, sự di chuyển các khối dữ liệu được thực hiện bằng sử dụng lệnh khối ghi dịch chuyển BTW (Block Transfer Write) và lệnh khối BTR (Block Transfer Read).

Bộ xử lý cơ sở trong PLC 5 của Allen Bradley ở chế độ quét có thể truyền tối sáu mươi tư từ mỗi lần đến hay đi từ mô đun truyền khối (BT) trong các giá đỡ tại chỗ hay hay từ xa. Mô đun BT đặc trưng là mô đun đầu vào nối với các can nhiệt, các mô đun vào/ra, các mô đun vào ra BCD, các bộ đếm xung. Số đồ khối của các lệnh BT được minh họa trên hình 4.. Các khối dữ liệu được truyền từ mô đun BT gá giá đỡ từ xa về PLC giám sát trong chế độ quét. Các mô đun từ xa của Allen Bradley PLC5 có một mô đun kết nối tương thích ABS -17, mà mô đun này có thể trao đổi với các mô đun bằng kênh truyền dữ liệu trên giá đỡ.

PLC của Allen Bradley có thể truyền tối sáu mươi tư từ trong mỗi lần truyền giữa quá trình được giám sát với bộ xử lý thông qua chế độ quét và mô đun ABS. Trong ứng dụng, cả hai bộ xử lý thực hiện đồng thời các lệnh truyền khối nhưng ngược nhau.

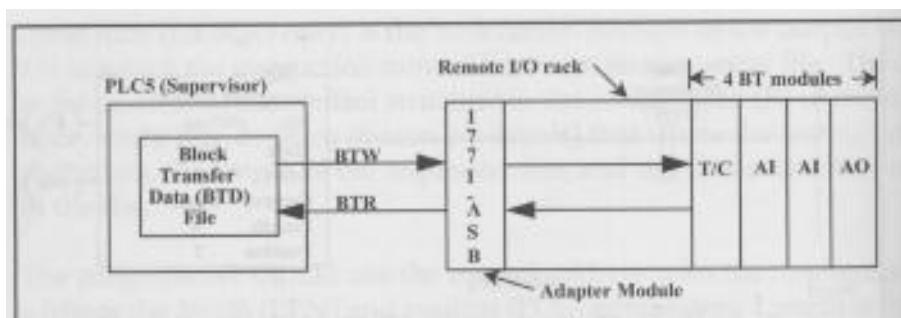
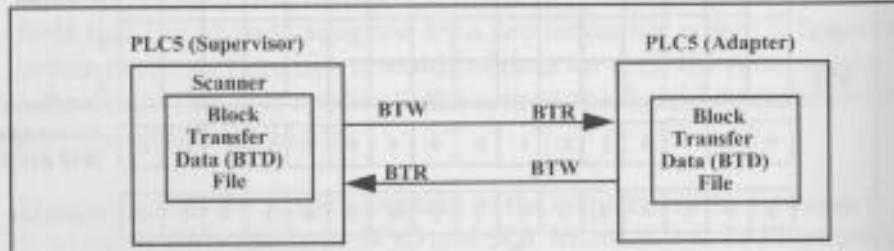
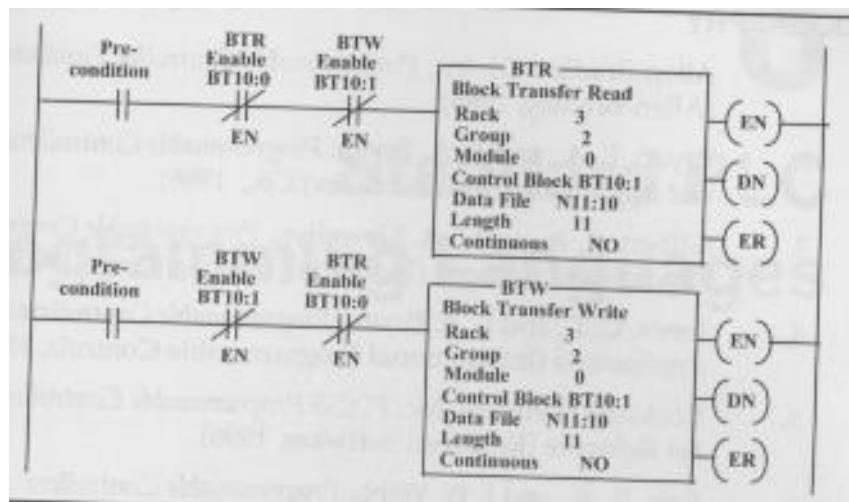


Figure 7-14. A-B PLC5 block transfer operation in scanner mode.



Trên hình là ví dụ về truyền khối hai chiều. Các bậc thang được sử dụng để đảm bảo các khối thông tin được truyền theo thứ tự mà chúng được gửi yêu cầu đến hàng chờ. Bộ xử lý đảo chiều giữa lệnh BTR và BTE trong thứ tự trên đó chúng được quét bởi khả năng tiềm ẩn của bit điều kiện cho phép trên bậc thang vào. Sử dụng bit thường đóng NC thông thường để cho phép thực hiện truyền khối giữa hai khối có thể ngăn các lệnh đọc và ghi của các lệnh truyền khối trong hàng chờ đồng thời. Trong trường hợp này lệnh điều

kiện cho trước thường mở NO được dùng bên trái của hai tiếp điểm của bit cho phép. Các bit điều kiện cho trước này cho phép các lệnh truyền được thực hiện theo thời gian hay theo sự kiện.



## CHƯƠNG 5 LẬP TRÌNH BẰNG NGÔN NGỮ TIÊU CHUẨN

### Giới thiệu

Theo tiêu chuẩn quốc tế, ngôn ngữ lập trình của PLC có năm ngôn ngữ đó là sơ đồ thang LAD, sơ đồ hàm lô gíc FBD, Sơ đồ hàm nhiệm vụ gián đoạn GRAFCET, bảng lệnh STL và ngôn ngữ lập trình bậc cao: soạn thảo cấu trúc ST (Structured Text). Bốn ngôn ngữ cơ bản ta đã giới thiệu ở phần đầu.

### Tiêu chuẩn quốc tế cho các ngôn ngữ PLC

Có rất nhiều tiêu chuẩn lập trình cho PLC được đưa ra bởi các Uỷ ban quốc gia và quốc tế để phát triển giao diện chung cho thiết bị điều khiển khả lập trình. Từ 1979 một nhóm các chuyên gia lập trình quốc tế được tập hợp từ một số các Uỷ ban quốc gia để soạn thảo những sơ họa đầu tiên về tiêu chuẩn dễ hiểu của PLC. Uỷ ban phác họa tiêu chuẩn này ra đời năm 1982. Sau khi Uỷ ban này đệ trình tài liệu phác thảo đầu tiên và hiệu chỉnh ban đầu, họ cho rằng tiêu chuẩn này khá phức tạp để có thể xử lý như một tài liệu đơn giản. Kết quả, nhóm làm việc phải tách ra làm năm nhóm với nhiệm vụ khác nhau tương ứng với mỗi phần của tiêu chuẩn. Phần 1 thực hiện về các thông tin chung, phần hai về thiết bị và thử nghiệm, phần ba về các ngôn ngữ lập trình, phần bốn là hướng dẫn sử dụng và phần năm là truyền thông.

Mỗi nhóm có vài chuyên gia quốc tế có sự trợ giúp của các nhóm chuyên gia quốc gia. Tiêu chuẩn IEC 61131-3 về các ngôn ngữ lập trình cho PLC được ra đời năm 1993 bởi Uỷ ban kỹ thuật điện tử quốc tế (International Electrotechnical Commission). Theo tiêu chuẩn này có ba ngôn ngữ lập trình đồ họa cho PLC là sơ đồ thang LAD, Sơ đồ hàm lô gíc FBD và GRAFCET và hai ngôn ngữ trên cơ sở bảng lệnh STL và cấu trúc soạn thảo ST. Tiêu chuẩn các ngôn ngữ lập trình của PLC cho phép lập trình các phân khai nhau của ứng dụng bằng một ngôn ngữ khác nhau và tổ hợp lại thành một chương trình thực hiện đơn giản.

Ngôn ngữ LAD là ngôn ngữ sử dụng rộng rãi nhất của các PLC. Ngôn ngữ này bao gồm một tập hợp các lệnh để thực hiện phần lớn các chức năng điều khiển cơ bản như: lô gíc dạng rơ le, lệnh đếm thời gian và đếm, và các phép toán cơ bản. Mặc dù vậy, tùy thuộc vào mô đun của PLC, người lập trình có thể mở rộng hay nâng cao các tập hợp lệnh để thực hiện các phép tính khác. Các chức năng tăng cường được sử dụng cho điều khiển tương tự, thao tác với dữ liệu, báo cáo, điều khiển lô gíc phức tạp, nhờ thư viện sơ đồ hàm chức năng có kết nối với các sơ đồ khối.

Ngôn ngữ sơ đồ hàm chức năng FDB là ngôn ngữ đồ họa. Ngôn ngữ này sử dụng các sơ đồ của các phân tử lô gíc tương tự như trong đại số Boolean để thể hiện các hàm lô gíc. Người ta cũng sử dụng các sơ đồ khối phức hợp để thực hiện các phép tính như đếm thời gian, đếm, các phép tính số học, nạp, so sánh và truyền dữ liệu. Người lập trình có thể sử dụng ngôn ngữ này để lập chương trình điều khiển phức tạp, nhờ thư viện sơ đồ hàm chức năng có kết nối với các sơ đồ khối.

Ngôn ngữ bảng lệnh là ngôn ngữ lập trình cấp thấp. Ngôn ngữ này rất đơn giản và dễ lập trình cho các ứng dụng nhỏ hay để tối ưu hóa các phân của một ứng dụng. Các lệnh luôn gắn với kết quả tức thời. Kết quả của phép tính thực hiện giữ các bộ ghi được lưu vào bộ ghi kết quả, thay cho giá trị trước đó.

Ngôn ngữ lập trình bằng soạn thảo theo cấu trúc ST là ngôn ngữ lập trình bậc cao, thiết kế cho tự động hóa quá trình. Ngôn ngữ này được sử dụng để thực hiện các các quá trình phức tạp mà các ngôn ngữ đồ họa không thể biểu diễn được dễ dàng. Ngôn ngữ ST là

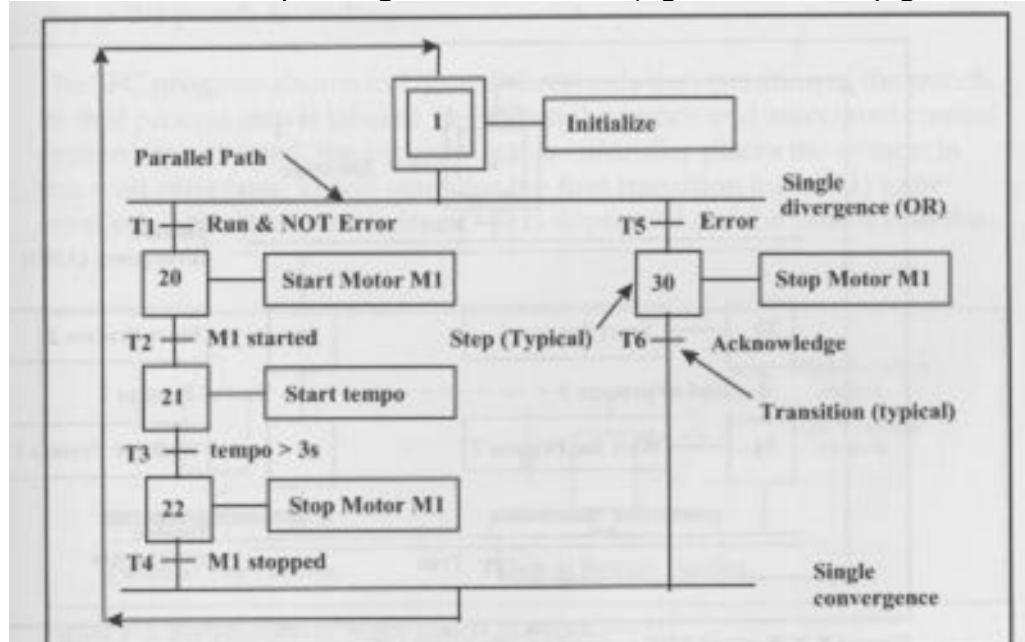
ngôn ngữ diễn tả các hoạt động bên trong các bước và các điều kiện gắn với trạng thái quá độ tương tự như ngôn ngữ GRAFCET.

GRAFCET là ngôn ngữ đồ họa dùng để diễn tả các hoạt động kế tiếp. Quá trình điều khiển được thể hiện như một tập hợp các bước hoạt động liên kết bằng các trạng thái chuyển tiếp quá độ. Điều kiện lô gíc của mỗi hoạt động trong mỗi bước là trạng thái chuyển tiếp gắn với bước này. Các hoạt động trong một bước và các điều kiện chuyển tiếp trạng thái của chúng có thể thực hiện bằng các lệnh từ các ngôn ngữ tiêu chuẩn khác.

## NGÔN NGỮ GRAFCET

Ngôn ngữ này diễn tả các hoạt động giống như trong thực tế của các hệ thống. Các bước hoạt động được thể hiện bằng các sơ đồ khối, với các liên kết là các điều kiện bắt đầu và kết thúc của mỗi bước. Các điều kiện này chính là các trạng thái chuyển tiếp của giữa các bước. Chương trình bắt đầu bao giờ cũng bằng bước khởi động, là nơi mà PLC bắt đầu thực hiện các chức năng điều khiển của mình và cũng đồng thời là nơi quay trở về của chương trình sau khi thực hiện đến lệnh cuối cùng hay sau khi thực hiện một lệnh điều kiện nào đó buộc hệ thống phải hoạt động lại từ đầu. Mỗi sơ đồ khối của mỗi bước hoạt động là một đơn vị cơ bản của ngôn ngữ GRAFCET, chứa đựng bên trong nó lô gíc hoạt động cho mỗi bước độc lập của quá trình công nghệ hay của máy và thiết bị. Trong mỗi khối có thể được đánh số qui ước thứ tự thực hiện trong chương trình và có thể được ghi chú sang bên phải của mỗi khối. Các điều kiện chuyển tiếp của các trạng thái quá độ giữa các bước cũng được ghi chú sang bên phải để tiện theo dõi hoạt động của chương trình. Điều kiện chuyển tiếp có giá trị lô gíc là 1 hay TRUE, của bước hay các bước hoạt động trước bước này đã kết thúc, và đây là điều kiện để bắt đầu của hoạt động kế sau. Kết thúc của lệnh cuối cùng hay của lệnh điều kiện nào đó thường là điều kiện để bắt đầu lại bước đầu tiên của chương trình.

Trên hình 4... là ví dụ chương trình điều khiển động cơ với nhiều trạng thái chuyển tiếp.



Hình ... Chương trình Grafcet của ví dụ ..

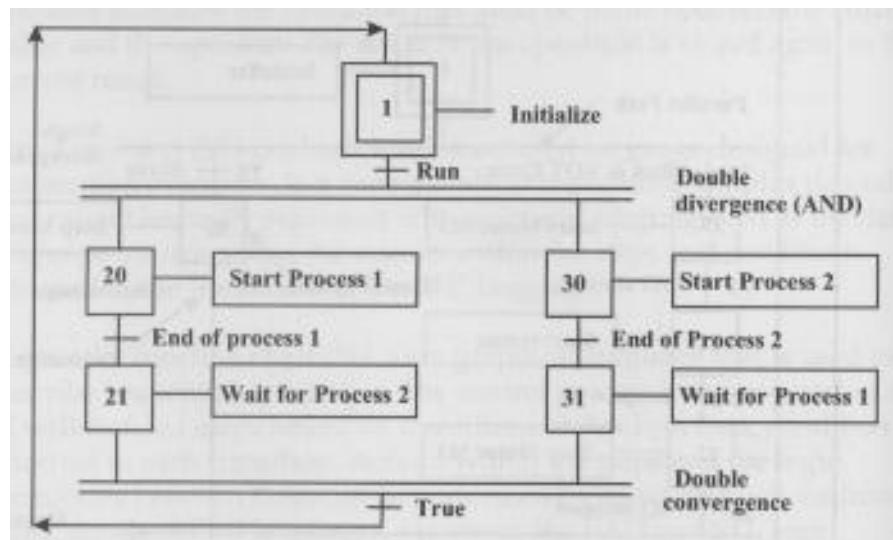
Trong Grafcet có các điều kiện chuyển tiếp đơn và các điều kiện chuyển tiếp kép để diễn tả trường hợp một trong các hoạt động có thể diễn ra hay trường hợp tất cả các hoạt động diễn ra đồng thời (song song). Điều kiện chuyển tiếp đơn có thể là điều kiện phân nhánh của các hoạt động, tương ứng với điều kiện lô gíc OR, hay là điều kiện hội tụ đơn, là bước hoạt động kế tiếp sẽ diễn ra khi một trong các nhánh đã kết thúc các hoạt động bên trong nó.

Để diễn tả điều kiện bắt đầu hay kết thúc của các hoạt động diễn ra đồng thời trên các nhánh song song, người ta sử dụng ký hiệu chuyển tiếp kép, tương ứng với lô gíc AND.

Điều kiện chuyển tiếp kép là điều kiện bắt đầu là điều kiện để nhiều hoạt động đồng thời có thể xảy ra trong cùng một thời điểm. Điều kiện chuyển tiếp kép là điều kiện kết thúc hay điều kiện bắt đầu của một bước hay một hoạt động kế tiếp khi tất cả các bước hoạt động kế trước của các nhánh song song cùng kết thúc tại một thời điểm. Nếu có một nhánh nào đó trong các nhánh song song, chưa kết thúc xong hoạt động trong bước của nó thì hoạt động của bước kế tiếp chưa thể xảy ra.

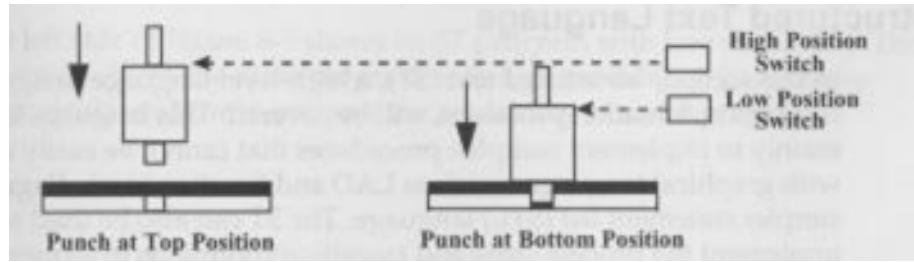
Trong GRAFCET có ba nguyên tắc cơ bản. Nguyên tắc thứ nhất là bước khởi động luôn được kích hoạt ở đầu chương trình. Chương trình kết thúc lệnh cuối cùng là điều kiện để bắt đầu lại bằng lệnh khởi động cho chu kỳ mới. Nguyên tắc thứ hai là trạng thái chuyển tiếp được kiểm tra sau khi nó kết hợp với bước tiếp theo và như vậy các phép tính đi qua từ bước này sang bước tiếp theo khi trạng thái chuyển tiếp có giá trị lô gíc là 1 (TRUE). Nguyên tắc thứ ba là sau khi trạng thái chuyển tiếp có lô gíc là 1, bộ xử lý quét bước này một lần nữa để bật lại toàn bộ các lệnh đếm thời gian à sau đó thực hiện bước tiếp theo. Lần quét này được gọi là quét hậu. Nhưng quét hậu chỉ dùng cho các bước thông thường. Trong các tệp chuyển tiếp, không cần có các bộ đếm thời gian, cho nên bộ xử lý không cần thực hiện quét hậu.

Khi có các nhánh chương trình song song với điều kiện đơn OR thì bộ xử lý sẽ quét chương trình lần lượt từ trái qua phải và từ trên xuống dưới của mỗi nhánh. Nếu các nhánh này là các điều kiện kép AND thì bộ xử lý sẽ tiến hành quét từ bên trái qua bên phải và từ trên xuống dưới đồng thời của các nhánh.

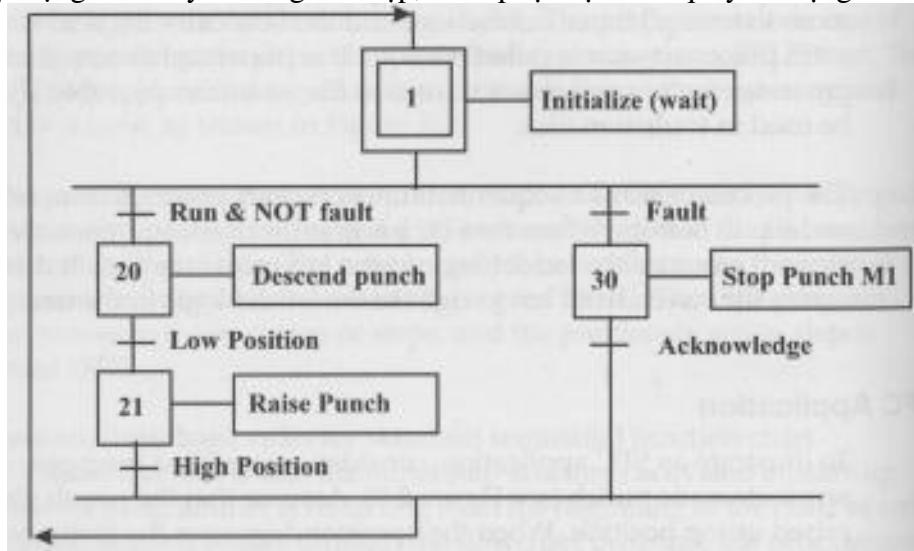


## Ứng dụng của Grafcet

Ứng dụng của Grafcet thuận tiện hơn các ngôn ngữ khác, ở chỗ nó thể hiện từng bước hoạt động của máy hay quá trình công nghệ cùng với các điều kiện bắt đầu và kết thúc của mỗi bước. Ngôn ngữ này rất đơn giản và tương tự như thuật toán dùng trong lập trình, nên rất dễ sử dụng. Ví dụ ta có một máy dập bán tự động điều khiển quá trình dập các sản phẩm cơ khí. Khi người thao tác ấn nút điều khiển, máy đi từ vị trí bắt đầu (vị trí cao) đi xuống dập vào tấm tôn. Khi đầu dập đạt đến vị trí thấp nhất tức là chi tiết đã dập xong, đầu dập chuyển động ngược lại. Các hành trình lên xuống được giới hạn bởi các công tắc hành trình. Trong hành trình dập, nếu có sự cố thì đầu dập sẽ dừng lại, khắc phục xong sự cố nó tự động quay về vị trí xuất phát.



Bước 1 là bước khởi động, đầu dập ở vị trí chờ trên cao. Khi nút ấn được ấn và máy không bị trục trặc gì, đầu dập thực hiện bước thứ hai là dập xuống. Khi đầu dập chạm công tắc hành trình dưới, tức là kết thúc hành trình dập, đầu dập thực hiện bước tiếp theo là quay trở về vị trí xuất phát. Hoạt động của bước này kết thúc khi đầu dập chạm công tắc hành trình trên cao. Máy lại về trạng thái chờ. Trường hợp đầu dập di xuống và có sự cố, thì bước hoạt động của máy là dừng đầu dập, khắc phục sự cố và quay về trạng thái chờ.



## CHƯƠNG 6

### NGÔN NGỮ DẠNG SOẠN THẢO CẤU TRÚC ST VÀ NGÔN NGỮ BẢNG LỆNH STL

Đây là ngôn ngữ lập trình bậc cao thiết kế cho tự động hóa các quá trình phức tạp. Ngôn ngữ này chủ yếu dùng để thực hiện các quá trình phức tạp mà các ngôn ngữ đồ họa như LAD, FBD, hay ngôn ngữ đơn giản kiểu bảng lệnh STT khó diễn đạt được. Ngôn ngữ soạn thảo cấu trúc ST cũng được sử dụng để thực hiện các bước của quá trình và các điều kiện chuyển tiếp trong các hàm nhiệm vụ kế tiếp. Một chương trình ST là danh các lệnh lập trình. Mỗi lệnh kết thúc bằng dấu cách (;). Các tên được dùng trong mã nguồn, như các biến nhận dạng, các hằng số, và các từ khoá của ngôn ngữ, được tách ra bởi các dấu cách thụ động như dấu cách trống, dấu hết dòng, dấu chấm. Các lệnh thường được tách riêng bằng các dấu cách tác động ví dụ dấu “<” hay dấu “>”. Người lập trình có thể chèn các ghi chú để cho người đọc có thể hiểu được. Phân chú thích phải bắt đầu bằng hai ký tự (\* và kết thúc bằng hai ký tự \*). Mỗi lệnh kết thúc bằng dấu ; (dấu cách).

Các lệnh cơ bản của chương trình ST là:

1. Lệnh gán tên biến (*Assignment*)  
*variable:=expression (biến:=hàm diễn tả)*
2. Lệnh gọi các hàm hay chương trình con : *Subprogram or functions call*
3. Lệnh gọi hàm khối: “C” *function block call*
4. Lệnh chọn điều kiện: *Selection statement (IF, THEN, ELSE, CASE, ...)*
5. Lệnh lặp : *Iteration statement (FOR, WHILE, REPEAT,...)*
6. Lệnh điều khiển: *Control statement (RETURN, EXIT,...)*
7. Các lệnh đặc biệt để có thể liên kết với các ngôn ngữ khác.

Người lập trình có thể chèn tuỳ ý các dấu cách thụ động giữa các dấu cách tác động, giữa các diễn tả hằng số, các biến nhận dạng để giúp cải thiện khả năng đọc cho chương trình. Các dấu cách thụ động của ngôn ngữ ST là dấu trắng, dấu *tabs*, và dấu hết dòng. Không giống như các ngôn ngữ hình thức như bảng lệnh STL, kết thúc dòng có được nhập vào bất kỳ của chương trình. Điều này nâng rất cao khả năng đọc của chương trình khi sử dụng các dấu cách thụ động. Một số qui tắc cần nhớ khi lập trình là:

1. Không viết lệnh dài quá một dòng.
2. Sử dụng dấu cách *tabs* để làm các lệnh phức hợp lùi vào.
3. Chèn các chú thích để đọc được các dòng lệnh.

Ví dụ về một đoạn chương trình trong ngôn ngữ ST:

(\*imax: số vòng lặp\*)  
(\*i: chỉ số của lệnh vòng FOR\*)  
(\*cond: biến điều khiển từ quá trình công nghệ\*)

```
imax:=max_ite;
cond:=X12
if not(cond(*báo động*))
then return;
end_if
(* Vòng lặp của quá trình công nghệ*)
for i (*chỉ số*):=1 to max_ite
do if 1<>2 then
SPcall();
```

```
End_if;  
End_for;
```

Ngôn ngữ soạn thảo cấu trúc ST hay ngôn ngữ lập trình bậc cao của PLC không khác nhiều so với các ngôn ngữ lập trình cho máy tính.

## NGÔN NGỮ BẢNG LỆNH TIÊU CHUẨN

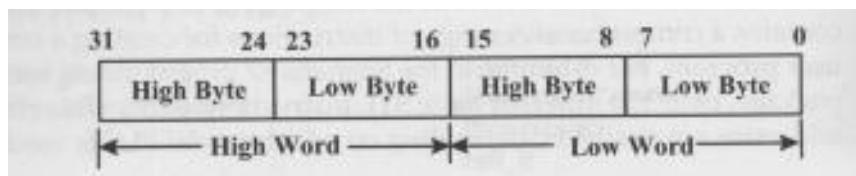
Đối với các PLC của hãng Siemens thì người ta hay dùng ngôn ngữ bảng lệnh STL cho cả các thế hệ PLC mới nhất PLC S7-300 và S7-400. Ngôn ngữ này là ngôn ngữ lập trình bằng các câu lệnh có cấu trúc tương tự như ngôn ngữ máy ASSEMBLER. Các lệnh thường đi với địa chỉ, để chỉ ra nơi mà các lệnh sẽ thực hiện. Ngôn ngữ bảng lệnh STL chứa một mảng các lệnh dễ hiểu để có thể tạo ra một chương trình điều khiển hoàn chỉnh. Các PLC S7 của Siemens sử dụng hơn 130 lệnh STL cơ bản cùng với một miền rộng các địa chỉ sẵn có phụ thuộc vào kiểu PLC. Phần lớn các nhà sản xuất PLC cũng sử dụng thể hệ tương tự của ngôn ngữ STL.

### Cấu trúc của lệnh STL

Lệnh STL có hai cấu trúc cơ bản. Cấu trúc thứ nhất là lệnh không cần địa chỉ, ví dụ như lệnh phủ định NOT. Cấu trúc thứ hai là lệnh có kèm theo địa chỉ. Đây là loại lệnh dùng nhiều nhất.

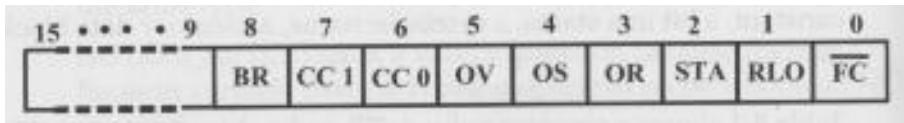
Địa chỉ của lệnh chỉ thị một hằng số hay một vị trí nhớ mà trên đó lệnh tìm được giá trị để thực hiện một phép tính. Địa chỉ có thể có tên bằng ký hiệu hay một vị trí nhớ cố định. Địa chỉ có thể chỉ đến số của phần tử nào đó như một hằng số, bit trạng thái, tên bằng ký hiệu, khối dữ liệu nhớ vv.

Các PLC S7 của Siemens sử dụng từ 32 bit, ký hiệu từ 0 đến 31.



Ví dụ lệnh nạp hằng số +77 và chuỗi ký tự “END” vào ắc qui 1.

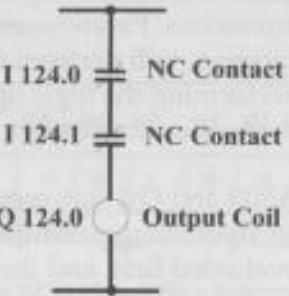
STL Instruction	Description
L +77	Load the integer 77 into accumulator 1.
L "END"	Load the ASCII character "END" into accumulator 1.

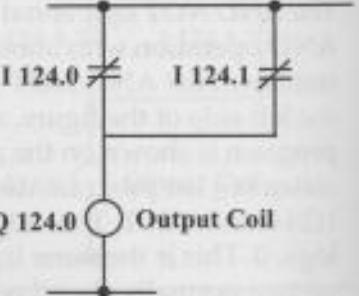


Statement List Program	Ladder Diagram Program
A 1124.0 A 1124.1 = Q 124.0	<p>I 1124.0    NO Contact  I 1124.1    NO Contact  Q 124.0    Output Coil</p>

Statement List Program	Ladder Diagram Program
O 1124.0 O 1124.1 = Q 124.0	<p>I 1124.0    I 1124.1  Q 124.0    Output Coil</p>

Statement List Program	Ladder Diagram Program
X 1124.0 X 1124.1 = Q 124.0	<p>I 1124.0    I 1124.0  I 1124.1    I 1124.1  Q 124.0    Output Coil</p>

Statement List Program	Ladder Diagram Program
AN I 124.0 AN I 124.1 = Q 124.0	

Statement List Program	Ladder Diagram Program
ON I 124.0 ON I 124.1 = Q 124.0	

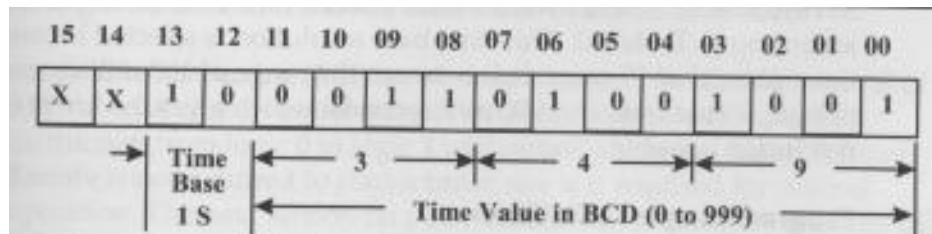


Table 8-2. Time Base and Its Binary Code

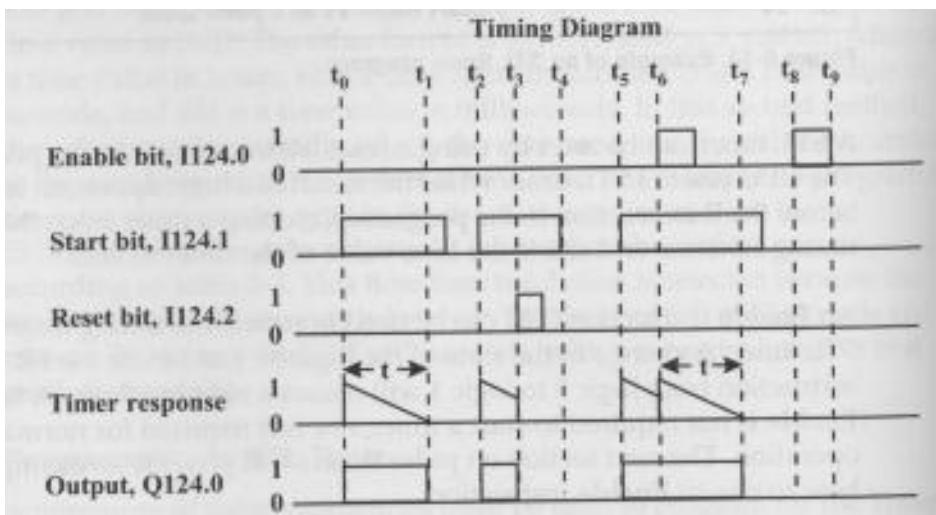
Time Base	Binary Code for Time Base
10 ms	00
100 ms	01
1 second	10
10 seconds	11

Table 8-3. Time Base Resolution and Ranges

Resolution	Range
0.01 second	10MS to 9S_990MS
0.1 second	100MS to 1M_39S_900MS
1 second	1S to 16M_39S
10 seconds	10S to 2HR_46M_30S

STL Program	Description
A I124.1	Check the logic state of "Start" bit I124.1
L S5T#1M10S	Load starting time of 1 minute & 10 seconds
SP T1	Start timer T1 as a pulse timer

STL Program	Description
A I124.0	Check the logic state of "Enable" bit I124.0
FR T1	Restart timer
A I124.1	Check the logic state of "Start" bit I124.1
L S5T#1M10S	Load starting time of 1 minute & 10 seconds
SP T1	Start timer T1 as a pulse timer
A I124.2	Reset timer bit, I124.2
R T1	Reset timer T1
A T1	
- Q124.0	Check the logic state of Timer T1

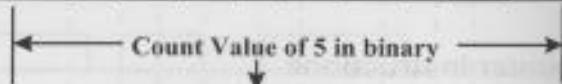


**STL Instruction: LC C3**

**Counter word for counter 3**

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

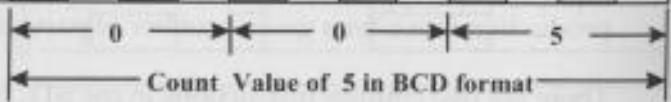
X	X	X	X	X	X	0	0	0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



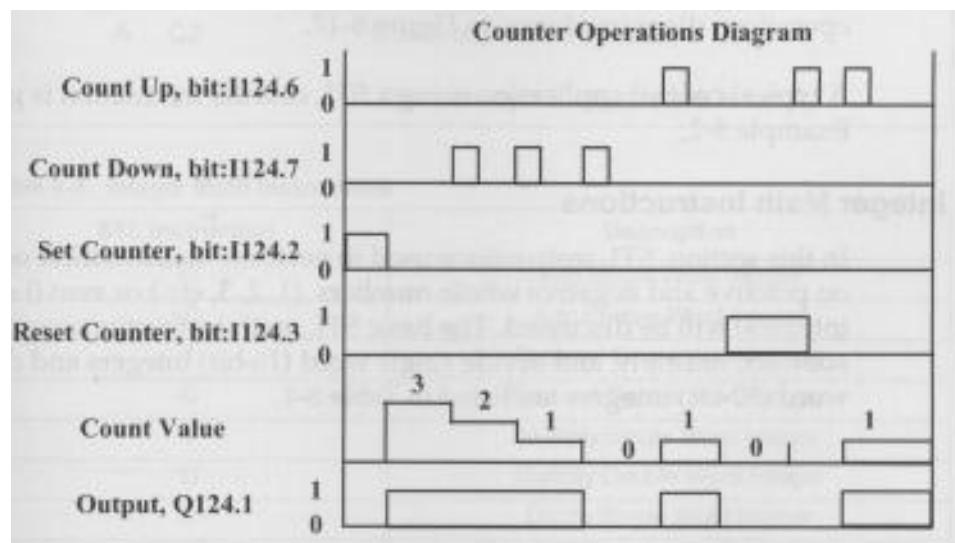
**Accumulator 1 low**

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

X	X	X	X	0	0	0	0	0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



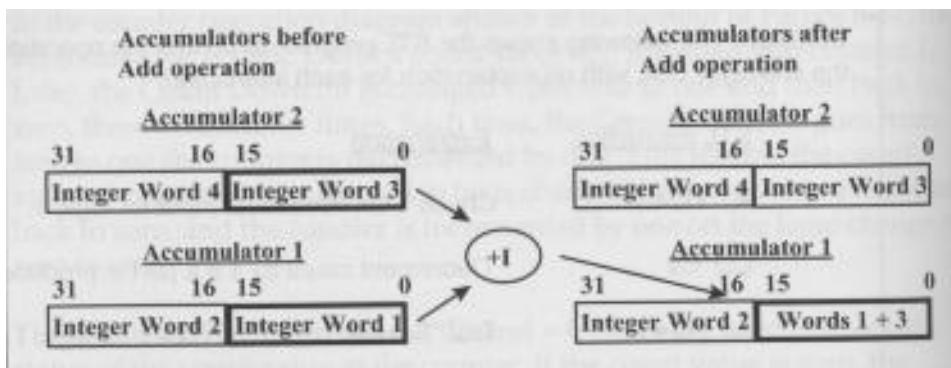
STL Program	Description
A I124.6	Check the logic state of "Count Up" bit I124.6
CU C1	Increment count value by 1
A I124.7	Check the logic state of "Count Down" bit I124.7
CD C1	Decrement count value by 1
A I124.2	Check the logic state of "Set Counter" bit I124.2
L C#3	Load count value of 3
S C1	Set count value to 3 in counter number 1.
A I124.3	Check logic state of "Reset Counter" bit I124.3
R C1	Reset count value to 0
A C1	Check the logic state of Counter C1
= Q124.1	Output the logic state of Counter C1



<u>STL Instruction</u>	<u>Explanation</u>
A I124.7	Check "Part Produced" input for logic 1.
CD C2	Decrement count by 1 if a part is produced.
A I124.2	Test "Set part count" bit I124.2 for logic 1.
L C#150	Load count of 150.
S C2	Set count value to 150 for counter number 2.
A I124.3	Check logic state of "Reset Counter" input.
R C2	Reset count to 0, if I124.3 = 1.
A C2	Check logic state of counter C2.
= Q124.3	Run conveyor if count value is not 0.

Table 8-4. Integer Math Instructions

<u>STL Instruction</u>	<u>Description</u>
+I	Add Single Word Integer
+D	Add Double Word Integer
-I	Subtract Single Word Integer
-D	Subtract Double Word Integer
*I	Multiply Single Word Integer
*D	Multiply Double Word Integer
/I	Divide Single Word Integer
/D	Divide Double Word Integer

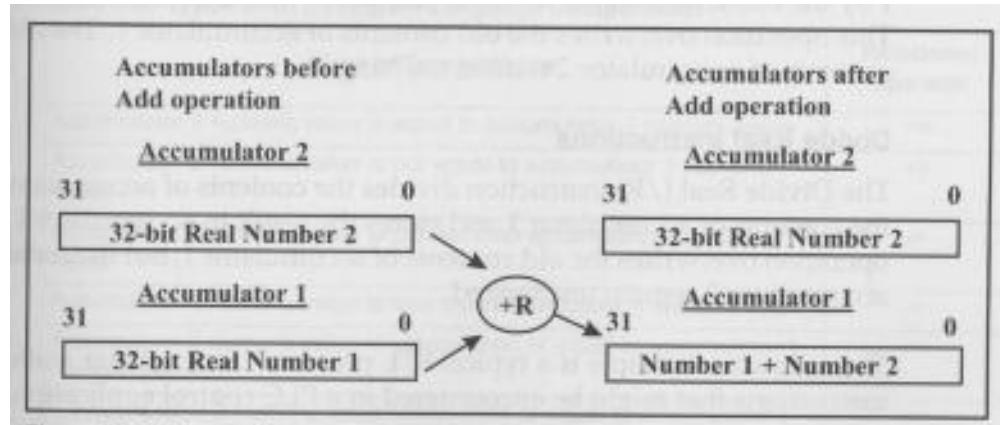


<u>STL Instructions</u>	<u>Description</u>
L MW10	Load the integer value in MW10 into accumulator 1
L MW12	Load the integer value in MW12 into accumulator 1
+I	Add single word MW12 to single word MW10
T MW14	Transfer integer result to data word MW14

<u>STL Instruction</u>	<u>Explanation</u>
L MW22	Load the integer value in MW22 into accumulator 1.
L MW40	Load the integer value in MW40 into accumulator 1.
+I	Add value in MW22 to value in MW40.
L 5	Load integer 5 into accumulator 1.
*I	Multiply result of addition (MW22 + MW40) by 5.
T MW50	Store result in MW50.

Table 8-5. STL Real Math Instructions

<u>STL Instruction</u>	<u>Description</u>
+R	Add 32-bit floating-point numbers
-R	Subtract 32-bit floating-point numbers
*R	Multiply 32-bit floating-point numbers
/R	Divide 32-bit floating-point numbers



<u>STL Instruction</u>	<u>Description</u>
L MD100	Load the real number in MD100 into accumulator 1
L MD104	Load the real number in MD104 into accumulator 1
+R	Add real number in MD100 to real number in MD104
T MD108	Transfer the result to double data word MD108

<u>STL Instruction</u>	<u>Description of Instruction</u>
L MD50	Load value in MD50 into accumulator 1.
L MD54	Load value in MD54 into accumulator 1.
	CPU transfers MD50 value to accumulator 2.
+R	Add MD50 to MD54.
L 1.000E+02	Load 100 into accumulator 1.
/R	Divide 100 into sum of MD50 and MD54.
T MD58	Transfer result to MD58.

Table 8-6. Types of STL Program Language Comparisons

Type of Comparisons	Relational Operator
Accumulator 2 numeric value is equal to accumulator 1 numeric value	==
Accumulator 2 numeric value is not equal to accumulator 1 numeric value	<>
Accumulator 2 numeric value is greater than accumulator 1 numeric value	>
Accumulator 2 numeric value is less than accumulator 1 numeric value	<
Accumulator 2 numeric value is greater than or equal to accumulator 1 numeric value	>=
Accumulator 2 numeric value is less than or equal to accumulator 1 numeric value	<=

STL Instructions	Description
L 5	Load the number 5 into accumualtor 1
L MW10	Load the integer value in MW10 into accumualtor 1
=I	Determine if the integer value of MW10 is equal to 5.
= Q124.0	Set Q124.0 to 1, if MW10 = 5
>I	Determine if MW10 is greater than 5
= Q124.1	Set Q124.1 to 1, if MW10 > 5
<I	Determine if MW10 is less than 5
= Q124.2	Set Q124.2 to 1, if MW10 < 5

Cách biểu diễn lệnh và dấu ngoặc

Lập trình bằng ngôn ngữ bảng lệnh

Cấu trúc của bảng lệnh

Lệnh bit lô gíc

Lệnh AND

Lệnh OR

Lệnh EXCLUSIVE OR

Lệnh AND NOT

Lệnh OR NOT

Lệnh thời gian

Cấu trúc từ của bộ đếm thời gian

Lập trình ngôn ngữ lệnh cho một bộ đếm giờ

Ứng dụng của lập trình bộ đếm giờ

Lệnh đếm

Cấu trúc từ của bộ đếm

Ứng dụng của lập trình bộ đếm

Các lệnh tính toán với số nguyên

Phép cộng số nguyên

**Phép trừ số nguyên**

**Phép nhân số nguyên**

**Phép chia số nguyên**

**Các lệnh toán có dấu phẩy**

**Lệnh cộng số thực**

**Lệnh trừ**

**Lệnh nhân**

**Lệnh chia**

**Các lệnh so sánh**

**So sánh số nguyên đơn**

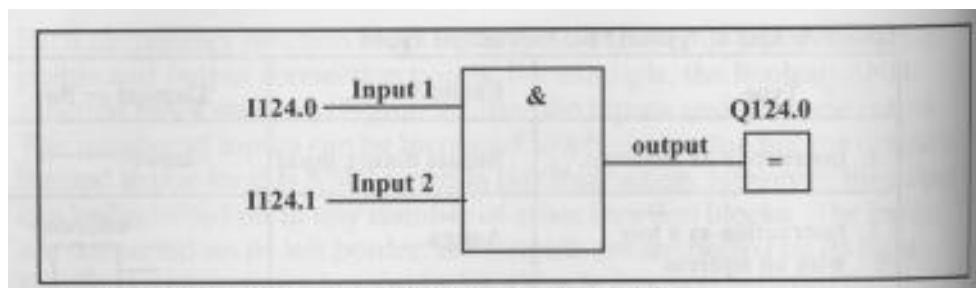
**So sánh số nguyên kép**

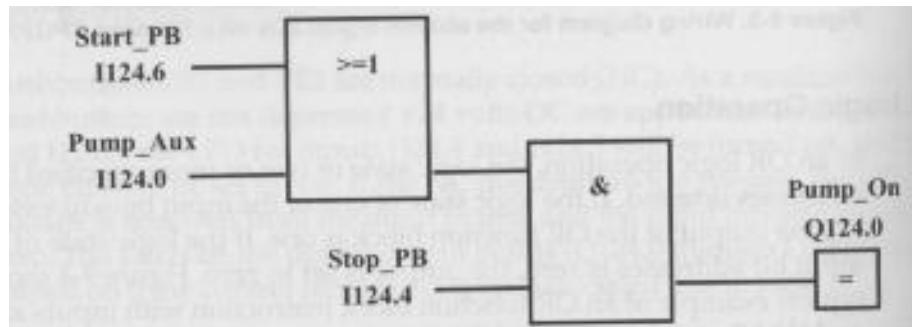
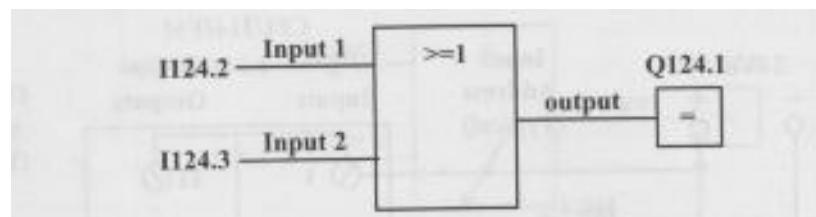
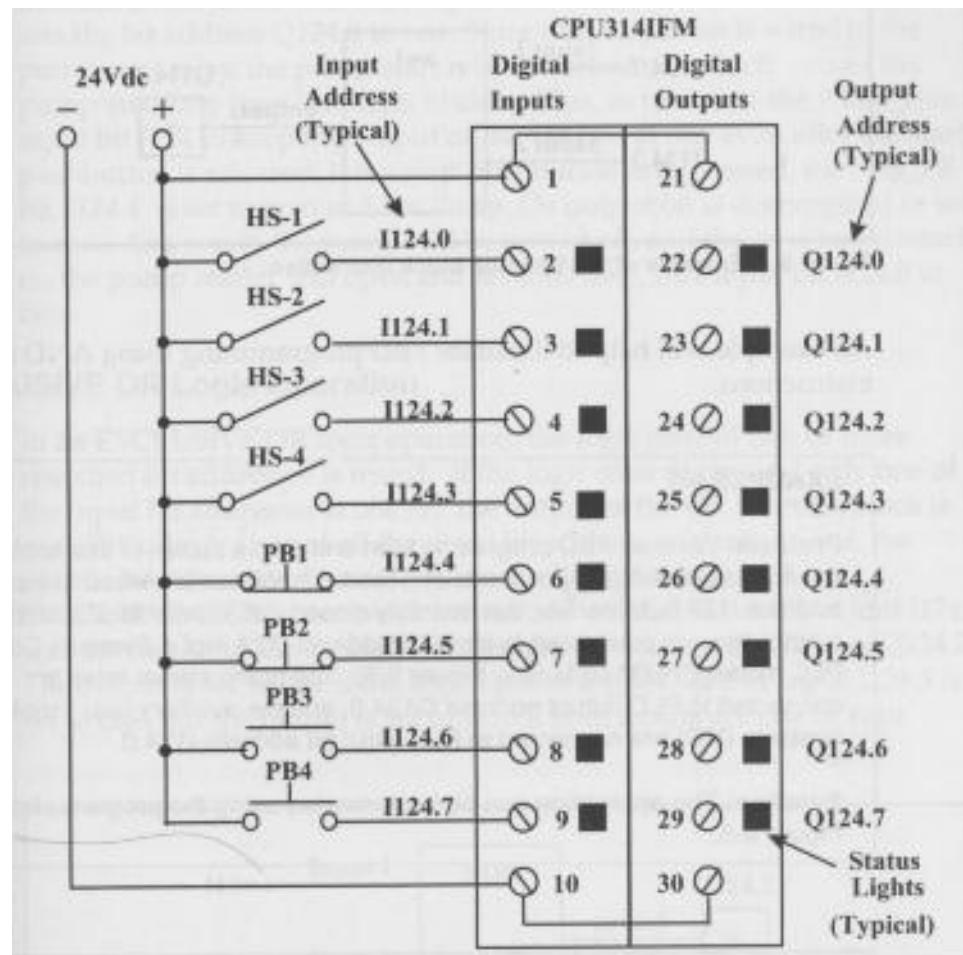
**So sánh số thực**

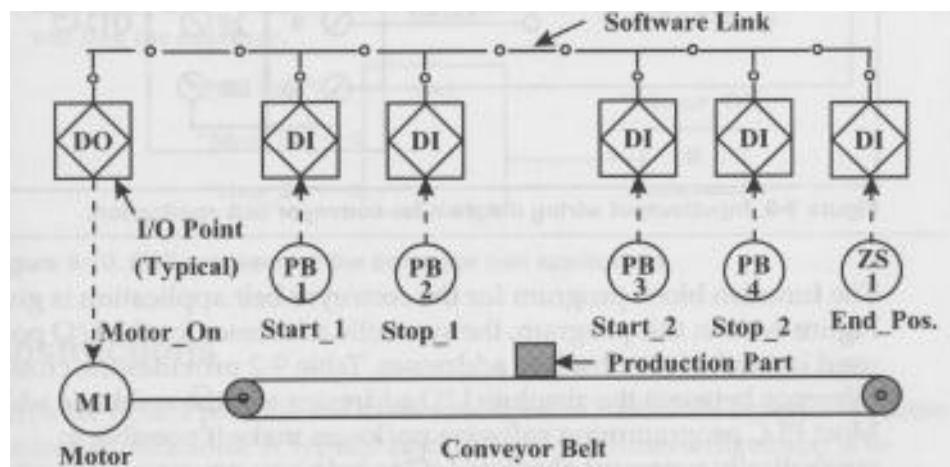
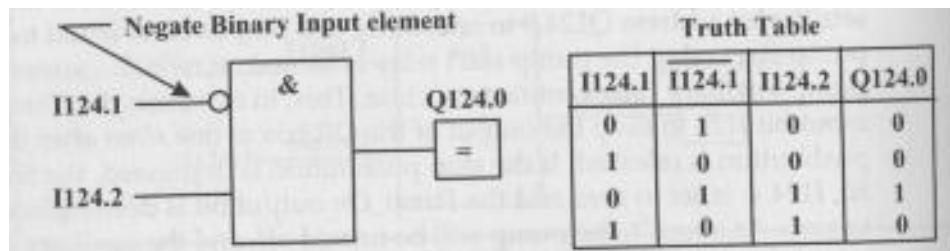
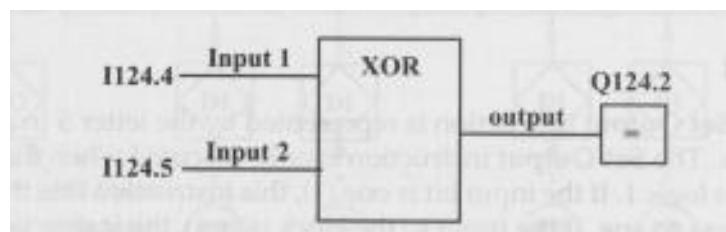
## CHƯƠNG 7

### NGÔN NGỮ LẬP TRÌNH BẰNG SỐ ĐỒ HÀM LÔ GIC TIÊU CHUẨN

#### Giới thiệu







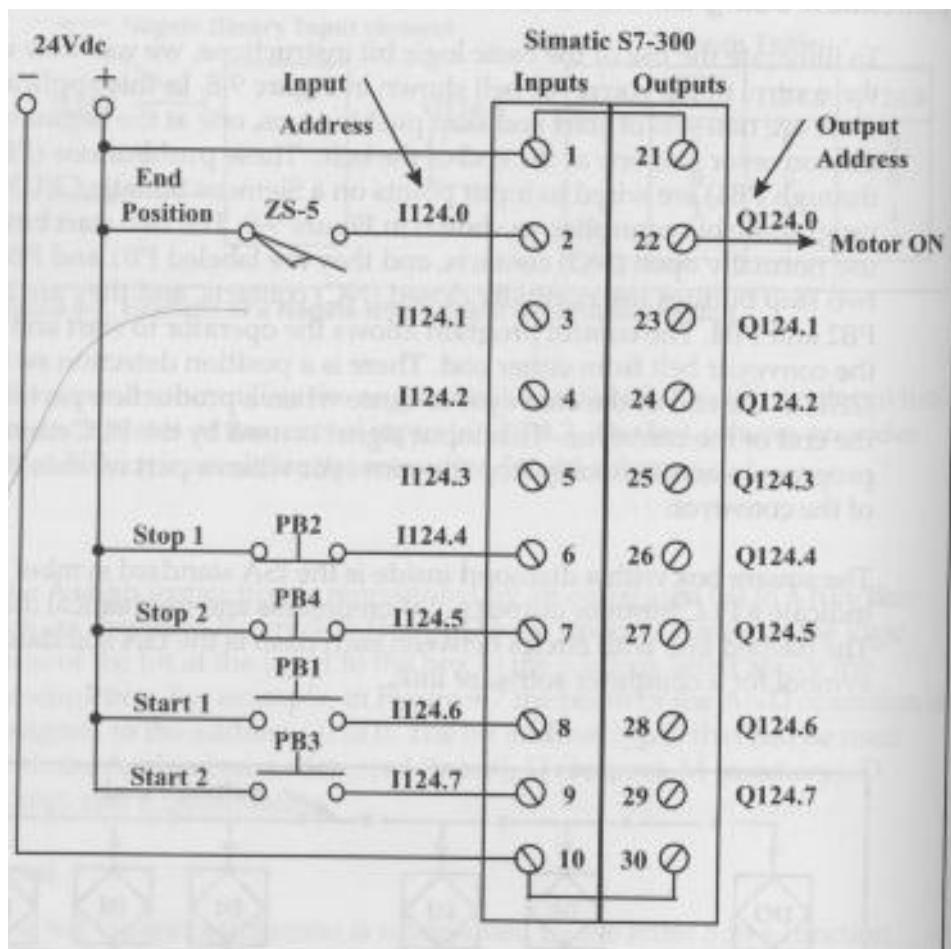
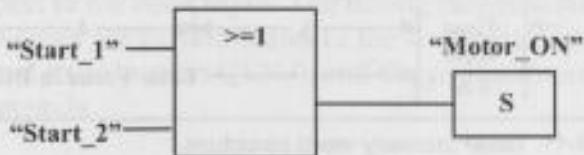


Table 9-2. Elements for Symbolic Programming of Conveyor Belt System

System Component	Absolute Address	Symbol	Symbol Table
Stop 1 NC PB	I124.4	Stop_1	I124.4 Stop_1
Stop 2 NC PB	I124.5	Stop_2	I124.5 Stop_2
Start 1 NO PB	I124.6	Start_1	I124.6 Start_1
Start 2 NO PB	I124.7	Start_2	I124.7 Start_2
Pos. Sw. ZS1	I124.0	End_Pos	I124.0 End_Pos
Motor Starter	Q124.0	Motor_On	Q124.0 Motor_On

### Network: 1 Start Conveyor

Depressing a start pushbutton will turn on the conveyor.



### Network: 2 Stop Conveyor

Depressing a stop pushbutton or the position sensor detecting a part will stop the conveyor.

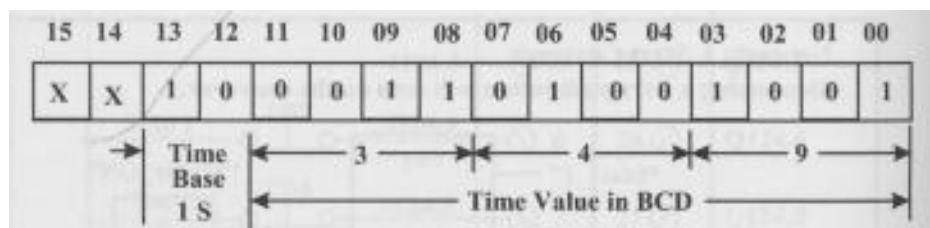
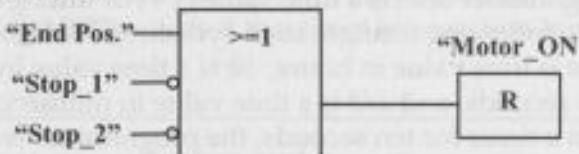
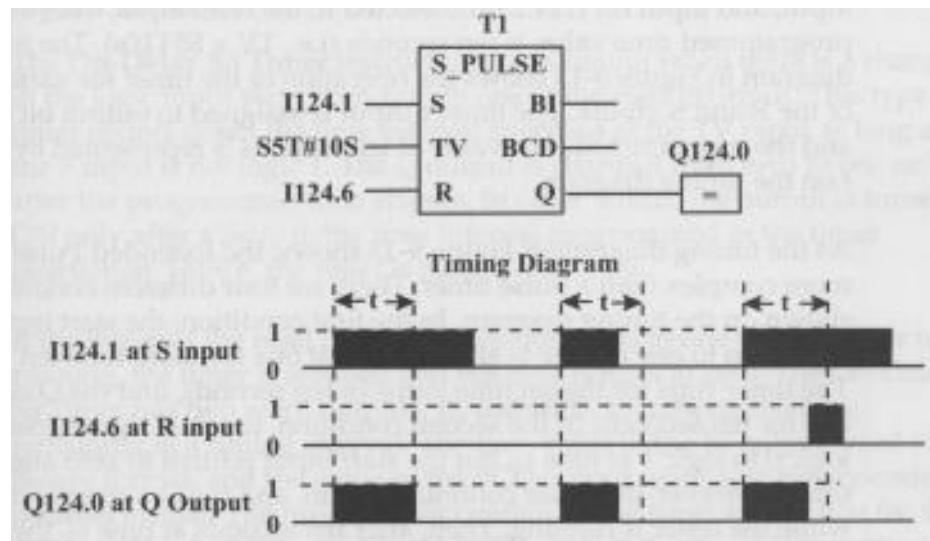
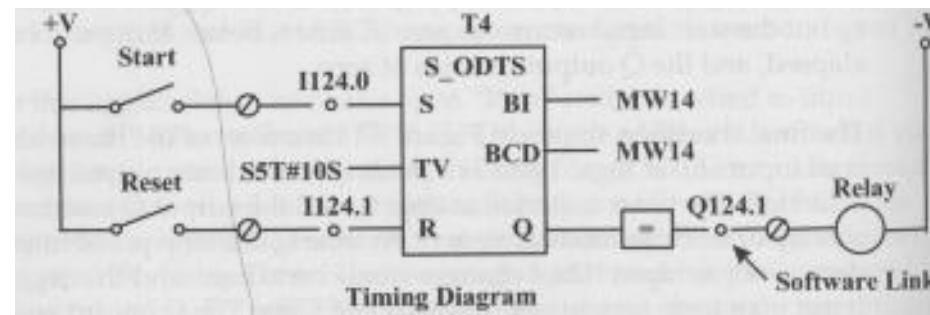
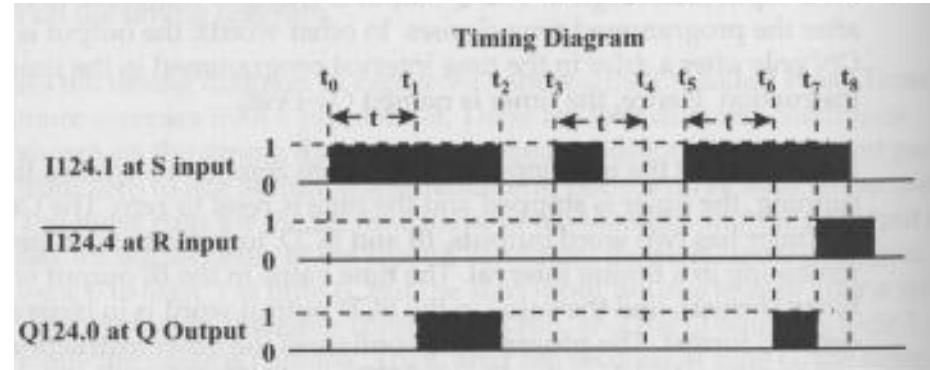
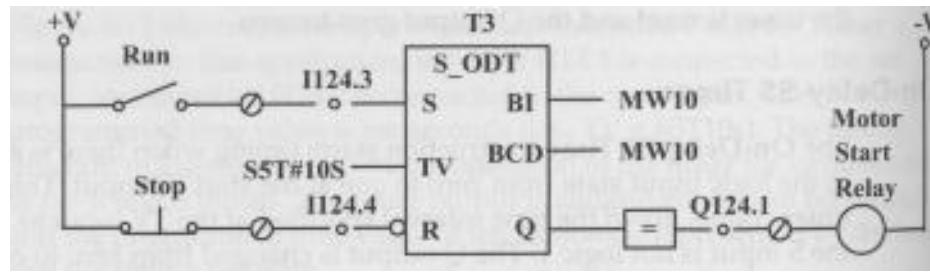
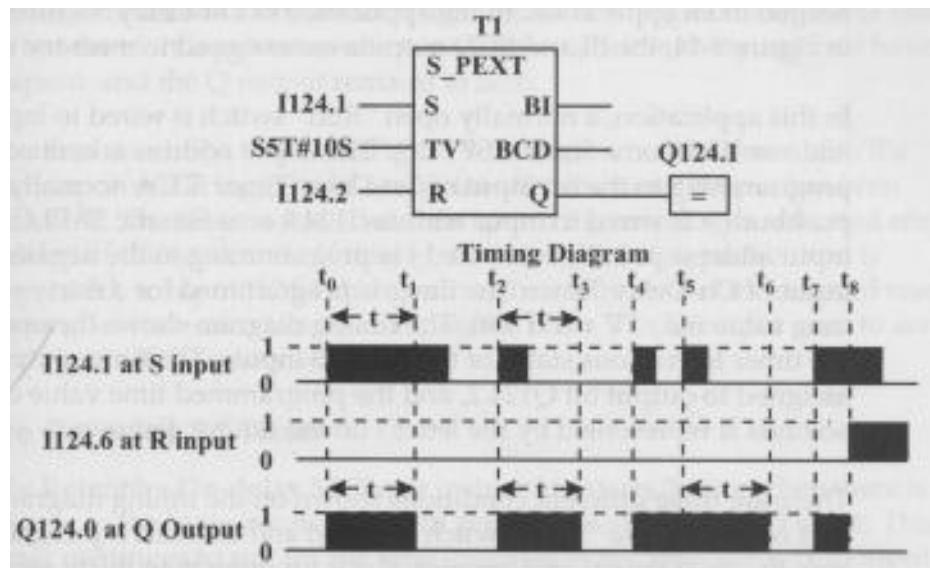
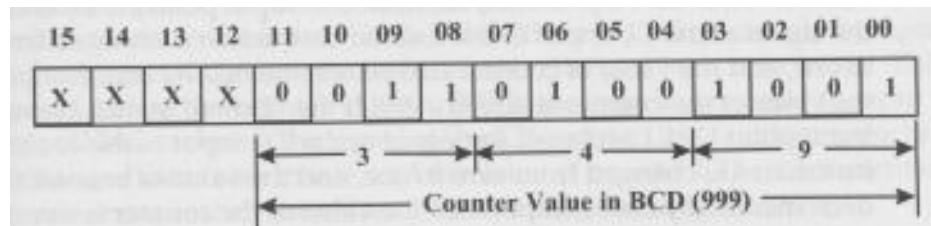
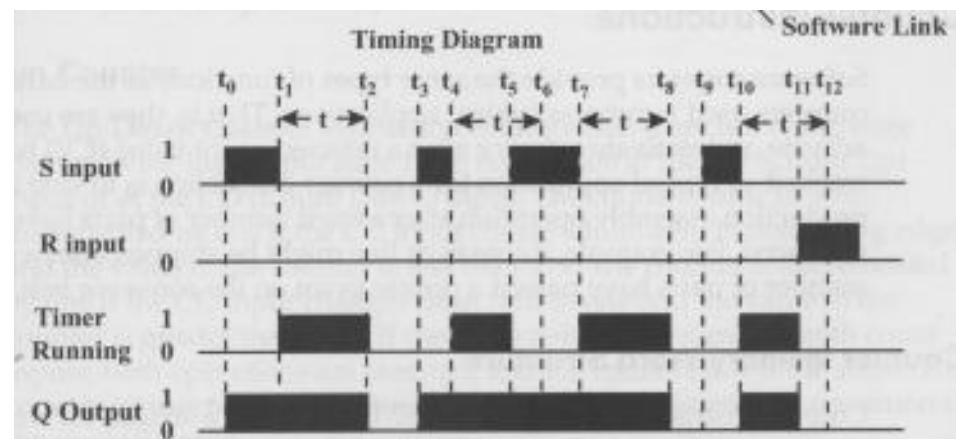
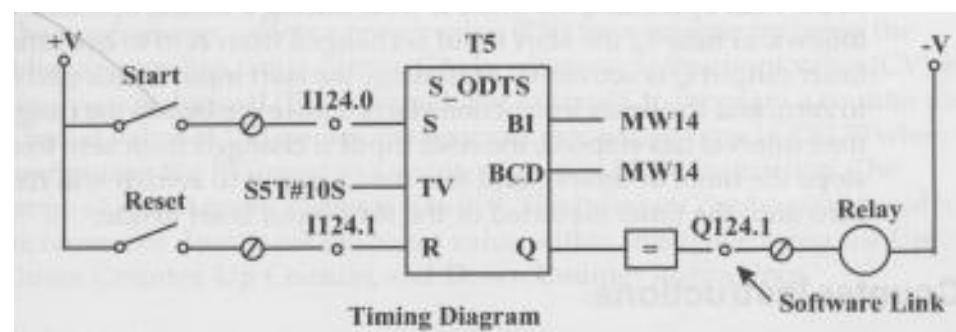
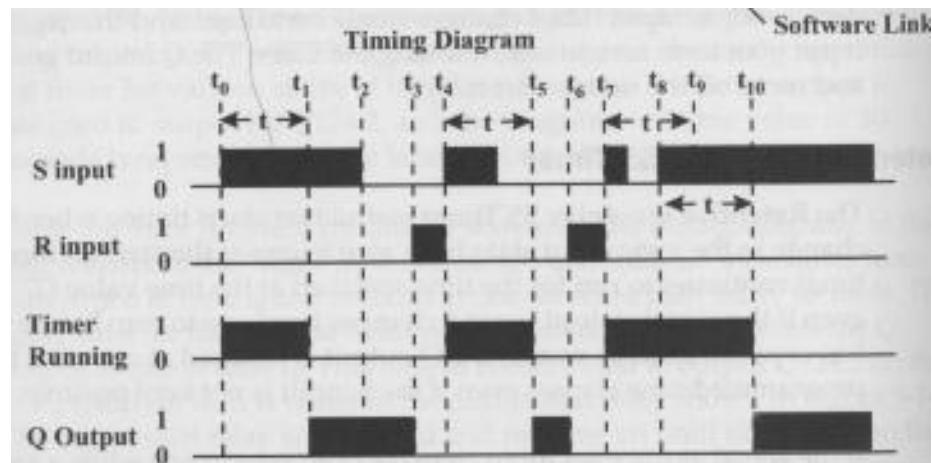


Table 9-3. Time Base and Its Binary Code

Time Base	Binary Code for Time Base
10 ms	00
100 ms	01
1 second	10
10 seconds	11







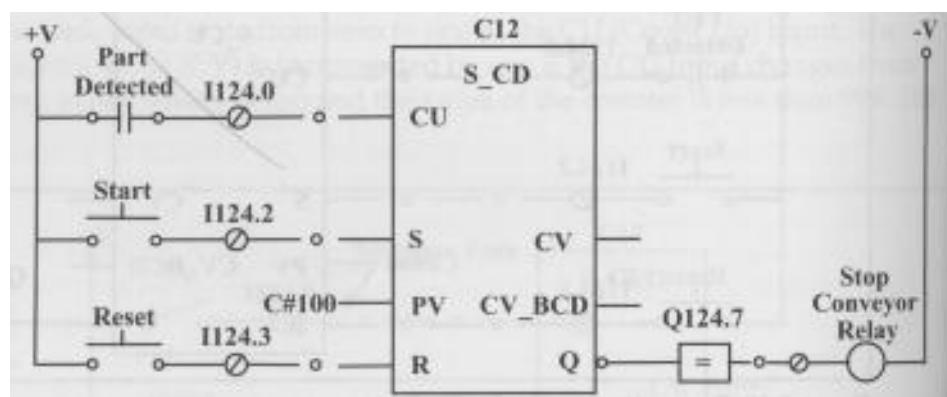
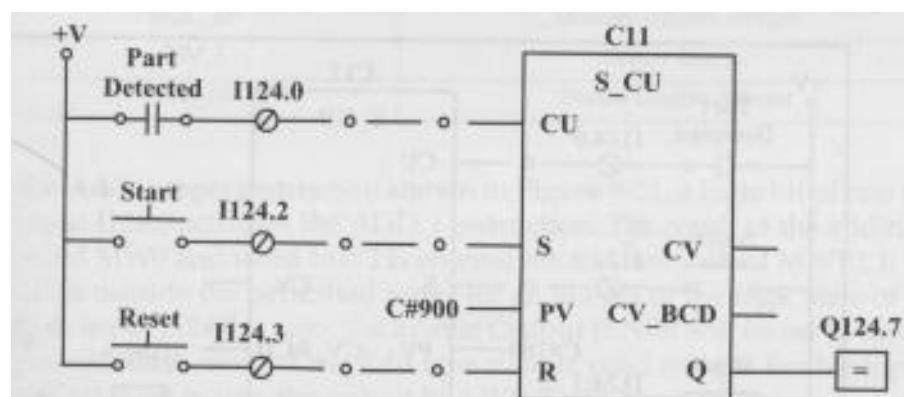
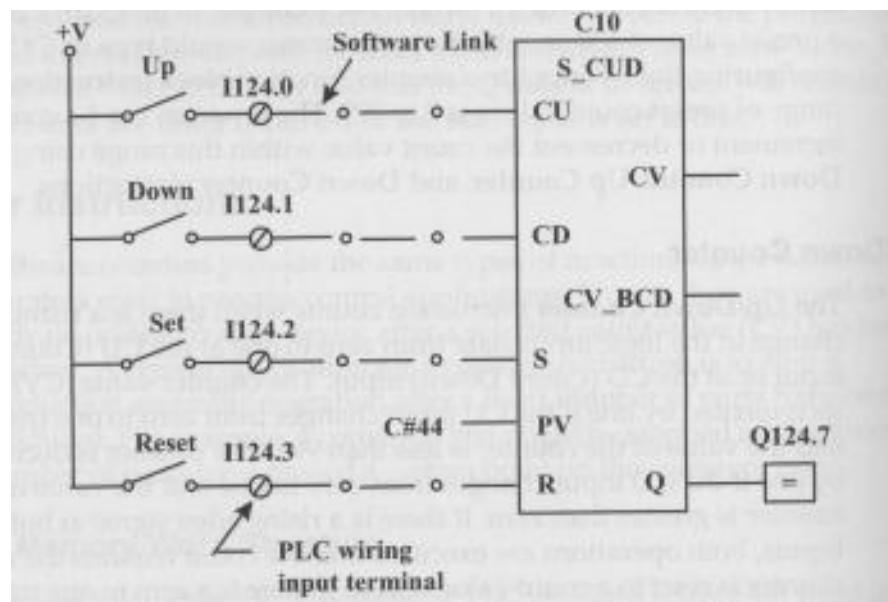
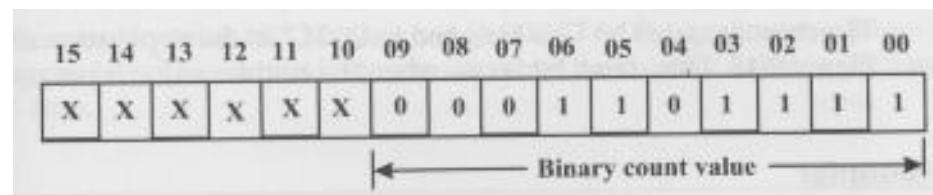


Table 9-4. Integer Math Instructions

Instruction Symbol	Description
ADD_I	Add Integer
ADD_DI	Add Double Integer
SUB_I	Subtract Integer
SUB_DI	Subtract Double Integer
MUL_I	Multiply Integer
MUL_DI	Multiply Double Integer
DIV_I	Divide Integer
DIV_DI	Divide Double Integer

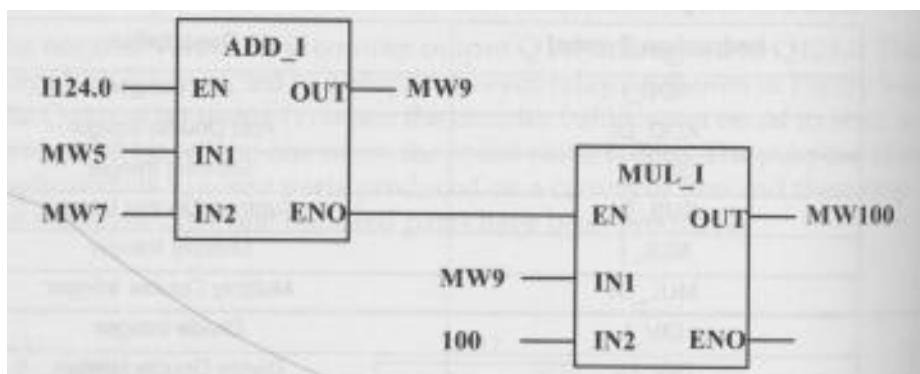
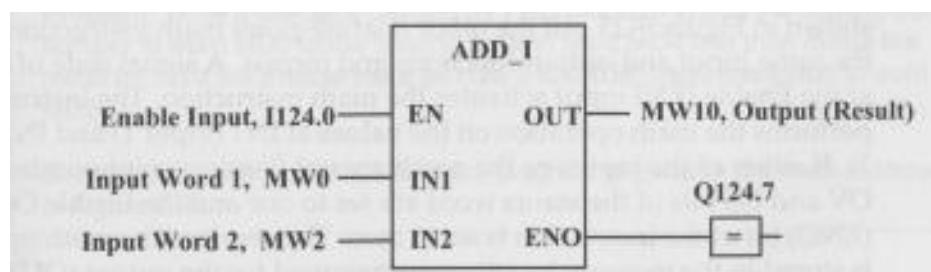


Table 9-5. Floating-Point Math Instructions

Instruction Symbol	Description
ADD_R	Add Real
SUB_R	Subtract Real
MUL_R	Multiply Real
DIV_I	Divide Integer

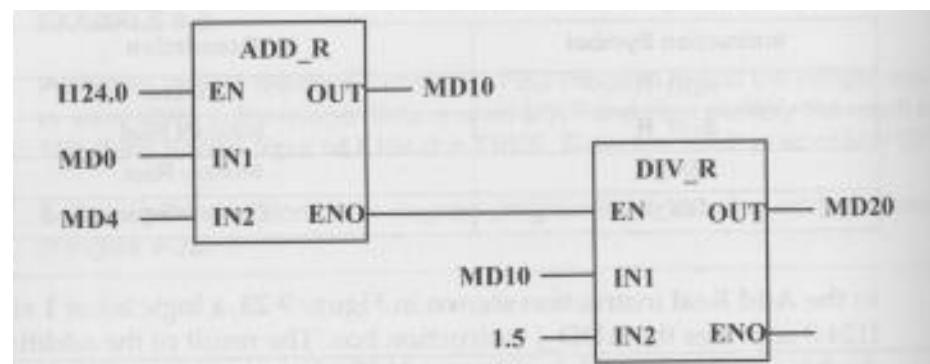
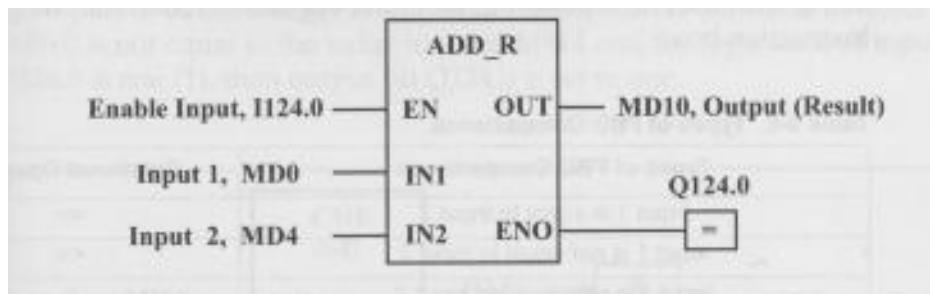
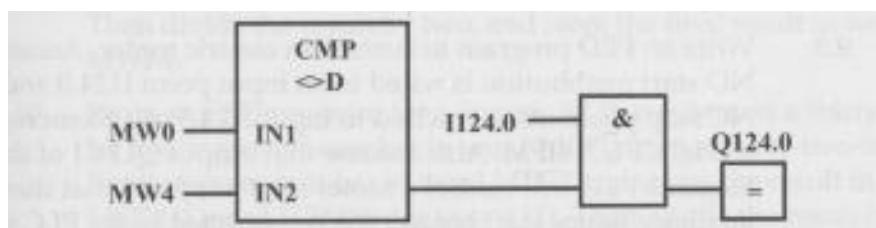
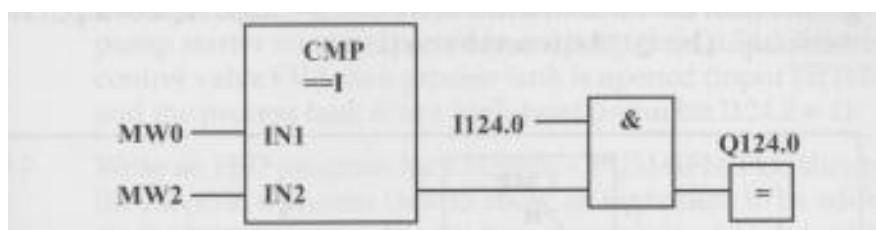
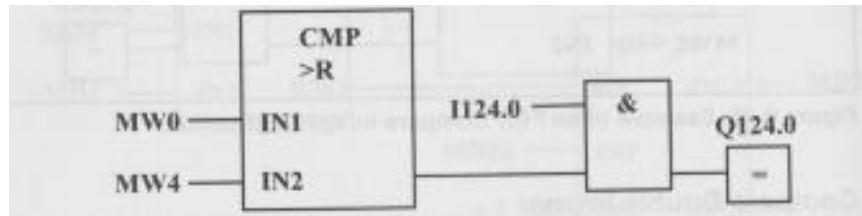


Table 9-6. Types of FBD Comparisons

Types of FBD Comparisons	Relational Operator
Input 1 is equal to Input 2	$=$
Input 1 is not equal to Input 2	$\neq$
Input 1 is greater than Input 2	$>$
Input 1 is less than Input 2	$<$
Input 1 is greater than or equal to Input 2	$\geq$
Input 1 is less than or equal to Input 2	$\leq$





## Phân tử và cấu trúc hộp

### Các lệnh bit lô gic

Lệnh AND

Lệnh OR

Lệnh EXCLUSIVE OR

Phủ định bit vào

Lệnh gán

Lệnh ra

Lệnh duy trì lệnh ra

Ứng dụng của các lệnh bit

### Các lệnh thời gian

Cấu trúc từ của bộ nhớ thời gian

Cấu hình của Sơ đồ khối của bộ đếm thời gian

Bộ đếm xung thời gian S5

Bộ đếm xung mở rộng S5

Bộ đếm thời gian trễ S5

Bộ đếm trễ có nhớ

Bộ đếm sớm S5

### Các lệnh đếm

Cấu trúc từ của bộ đếm

Cấu hình của Sơ đồ khối của bộ đếm

Bộ đếm tăng/giảm

Bộ đếm tăng

Bộ đếm giảm

### Các lệnh tính số nguyên

### Các lệnh tính với số thực

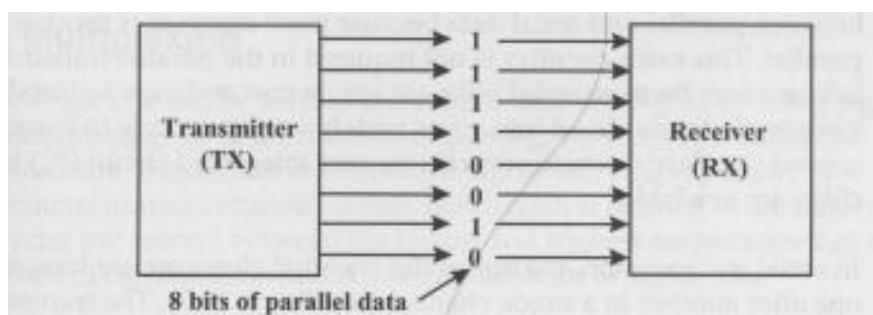
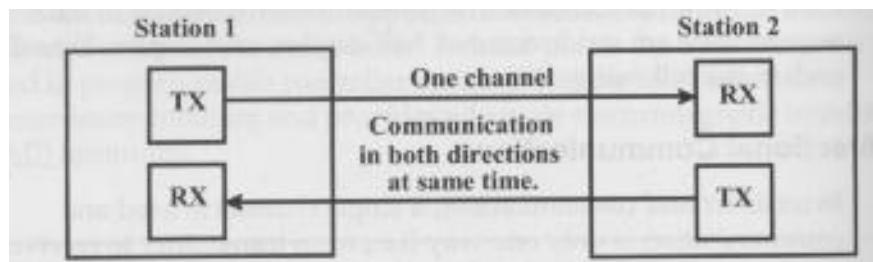
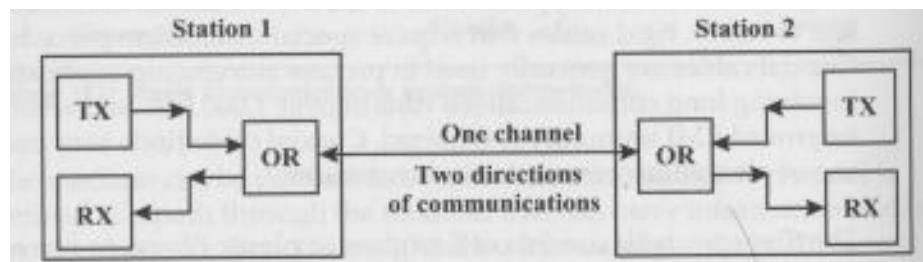
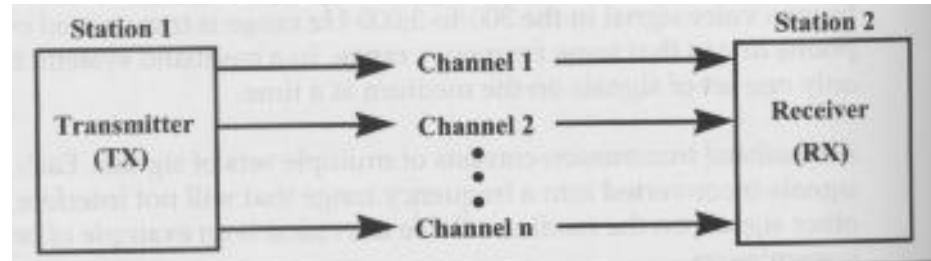
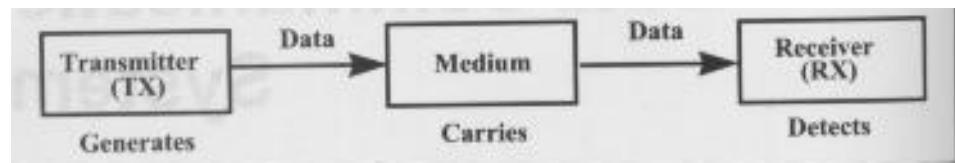
### Các lệnh so sánh

So sánh số nguyên

So sánh số thực

## CHƯƠNG 8 MẠNG VÀ HỆ THỐNG TRUYỀN DỮ LIỆU

### Cơ sở về truyền thông



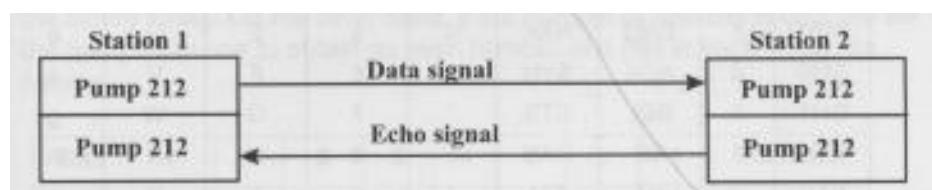
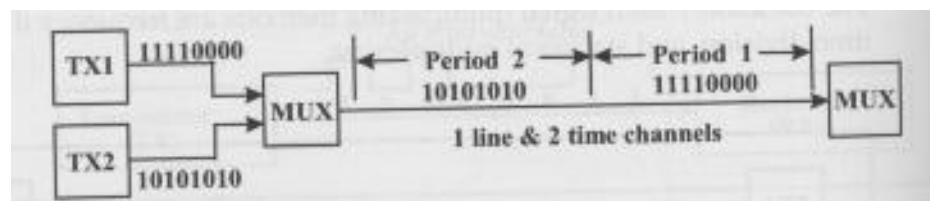
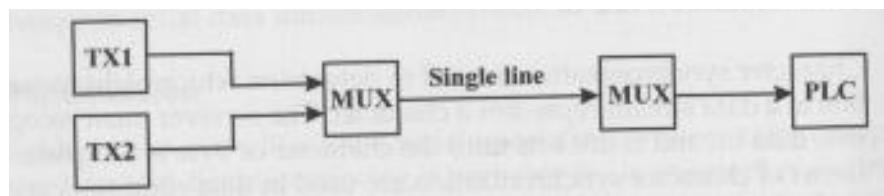
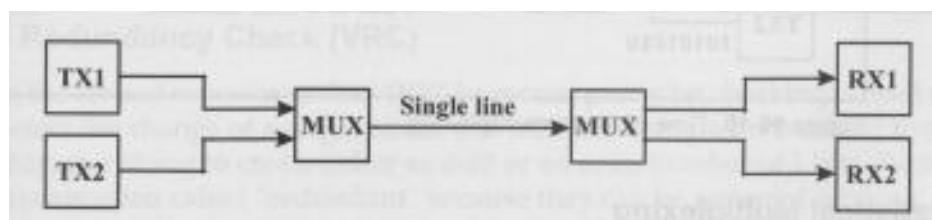
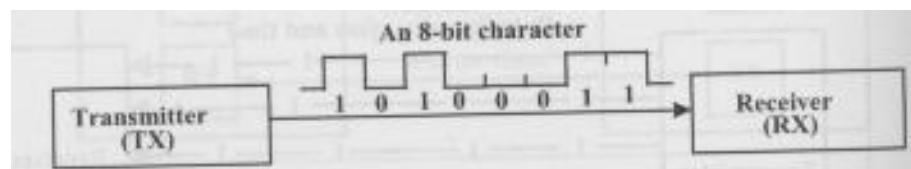
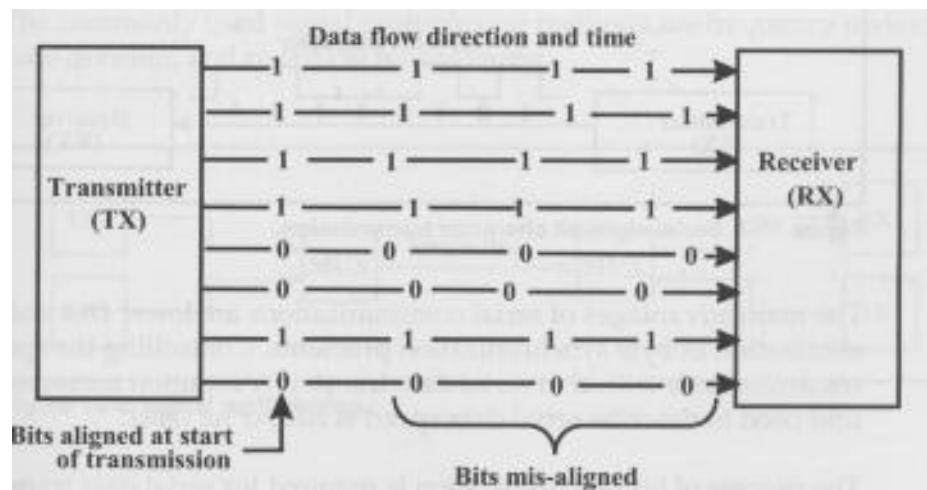


Table 10-1. ASCII Code

Bits	7	0	0	0	0	1	1	1	1
	6	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1
4321	HEX	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	@	P	.	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	:	K	[	k	{
1100	C	FF	FS	'	<	L	\	l	
1101	D	CR	GS	-	=	M	]	m	)
1110	E	SO	RS	,	>	N	^	n	-
1111	F	SI	US	/	?	O	-	o	DEL

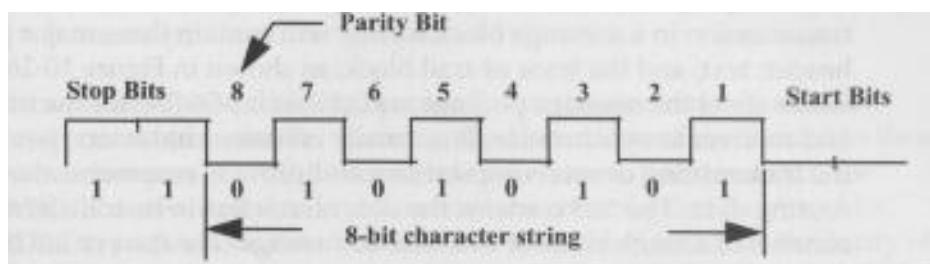
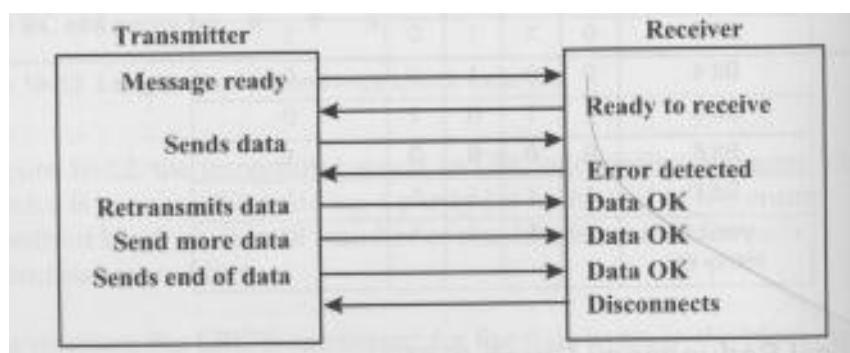
Bits	7	6	5	4	3	2	1	Character
	1	0	1	0	0	1	0	S
	1	0	1	0	1	0	0	T
	1	0	0	1	1	1	1	O
	1	0	1	0	0	0	0	P

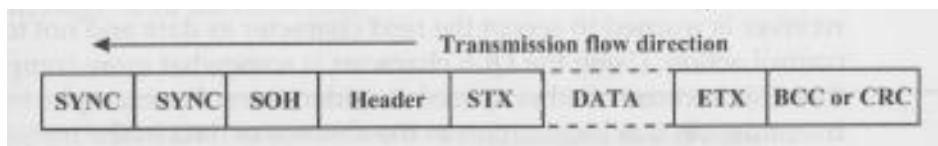
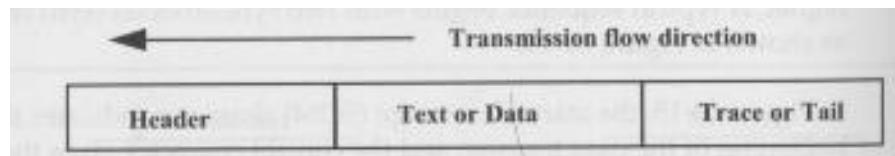
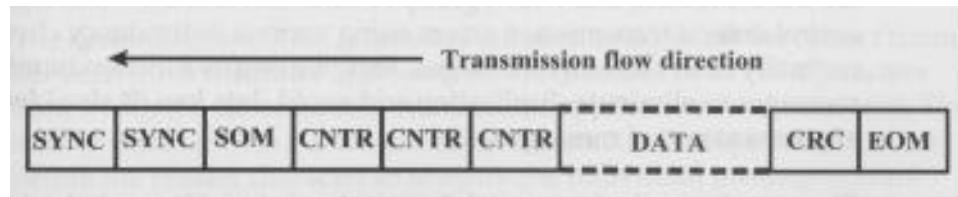
Bits	P	7	6	5	4	3	2	1	Character
	1	1	0	1	0	0	1	0	S
	1	1	0	1	0	1	0	0	T
	1	1	0	0	1	1	1	1	O
	0	1	0	1	0	0	0	0	P

Data flow			
Longitudinal check			LRC odd parity bit
R	U	N	
Bit 1:	0	1	0
Bit 2:	1	0	1
Bit 3:	0	1	1
Bit 4:	0	0	1
Bit 5:	1	1	0
Bit 6:	0	0	1
Bit 7:	1	1	0
VRC odd parity bit:	0	1	1

Table 10-2. VRC/LRC Odd-parity Example

Word	P	U	M	P	LRC Parity
Bit 1	0	1	1	0	1
Bit 2	0	0	0	0	1
Bit 3	0	1	1	0	1
Bit 4	0	0	1	0	0
Bit 5	1	1	0	1	0
Bit 6	0	0	0	0	1
Bit 7	1	1	1	1	1
Vertical Parity Bit	1	1	1	1	0

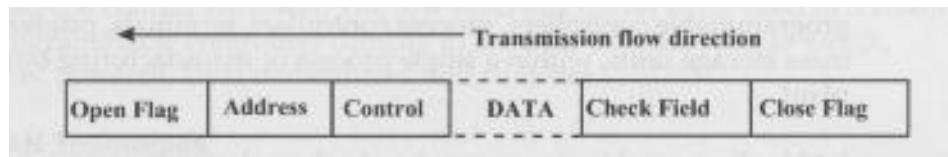


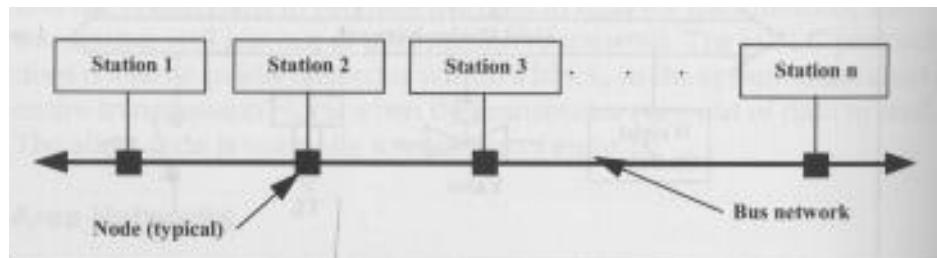
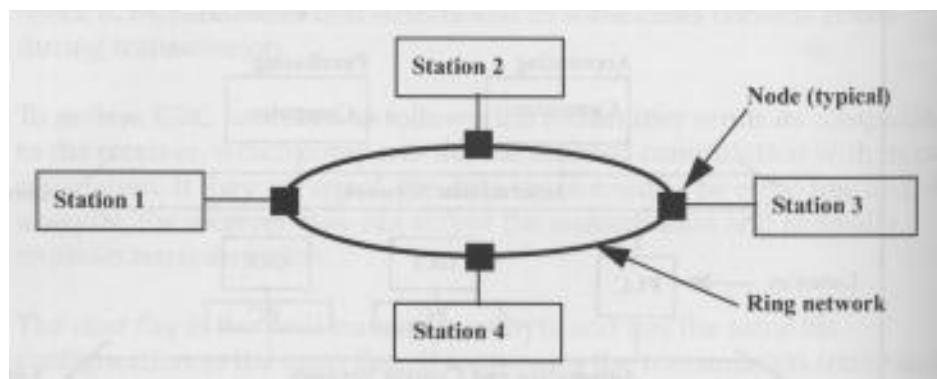
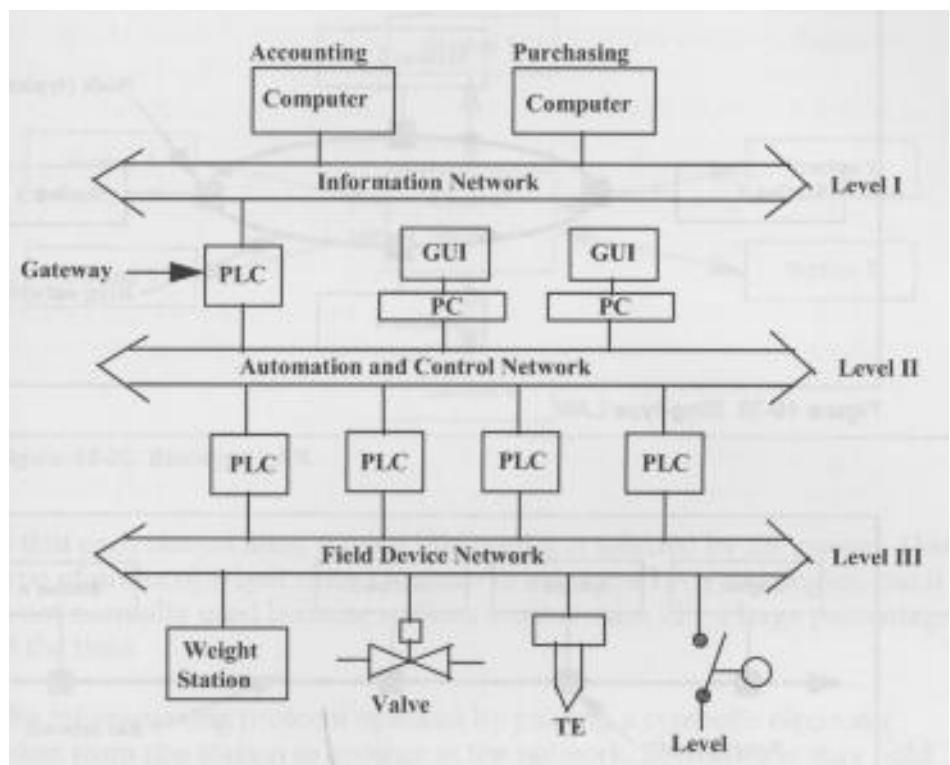


### EXAMPLE 10-3

**Problem:** Assemble the bit streams that are required to send the message M1 ON using eight-bit odd-parity ASCII code and BISYNC protocol. Assume that the header is given by 00000001 for transmitting station 1, and omit the check character (BCC or CRC) at the end of the transmission.

**Solution:** Use the ASCII code in Table 10-1 to find the bits required as follows: SYNC = 00010110, SOH = 00000001, STX = 00000010, M = 11001101, 1 = 00110001, space = 00100000, 0 = 01001111, N = 11001110, ETX = 10000011.





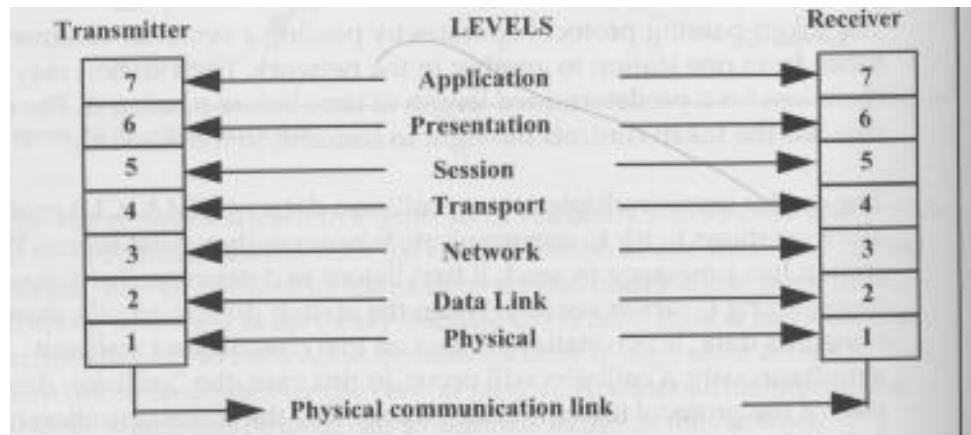
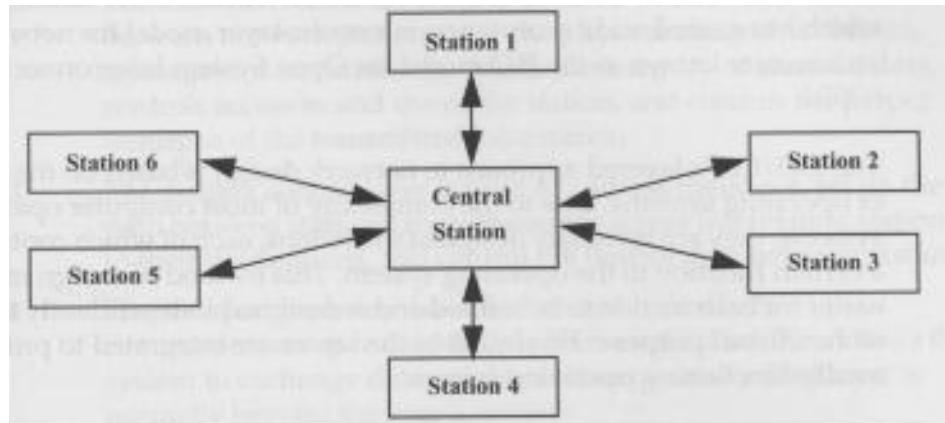
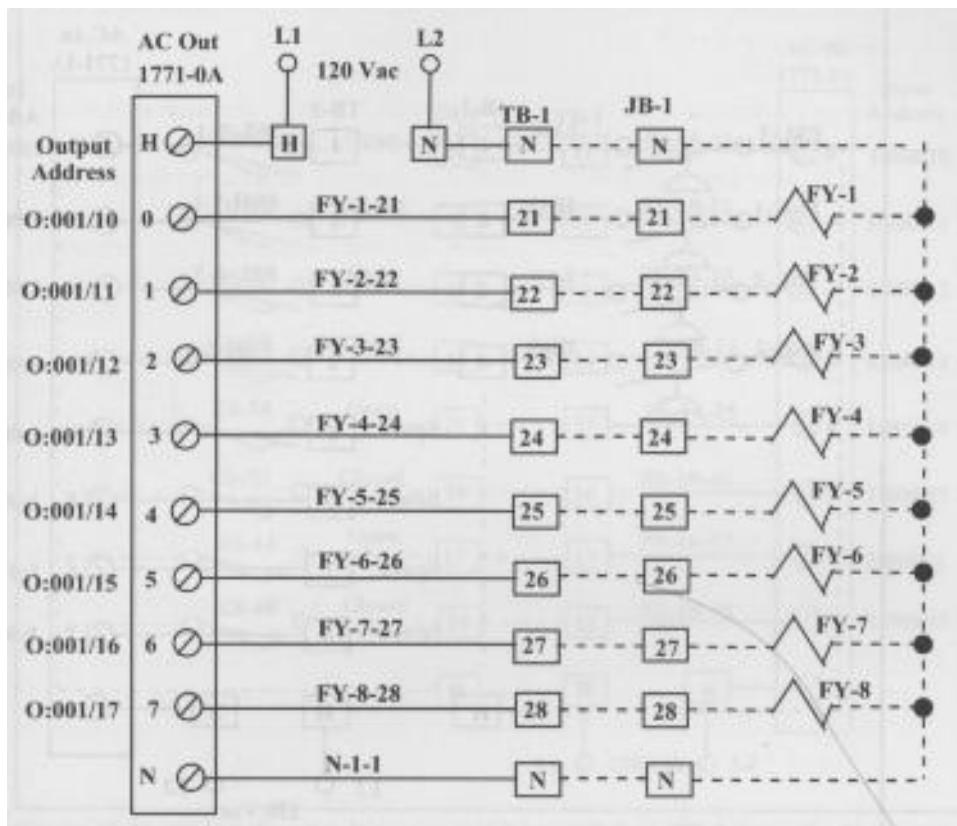
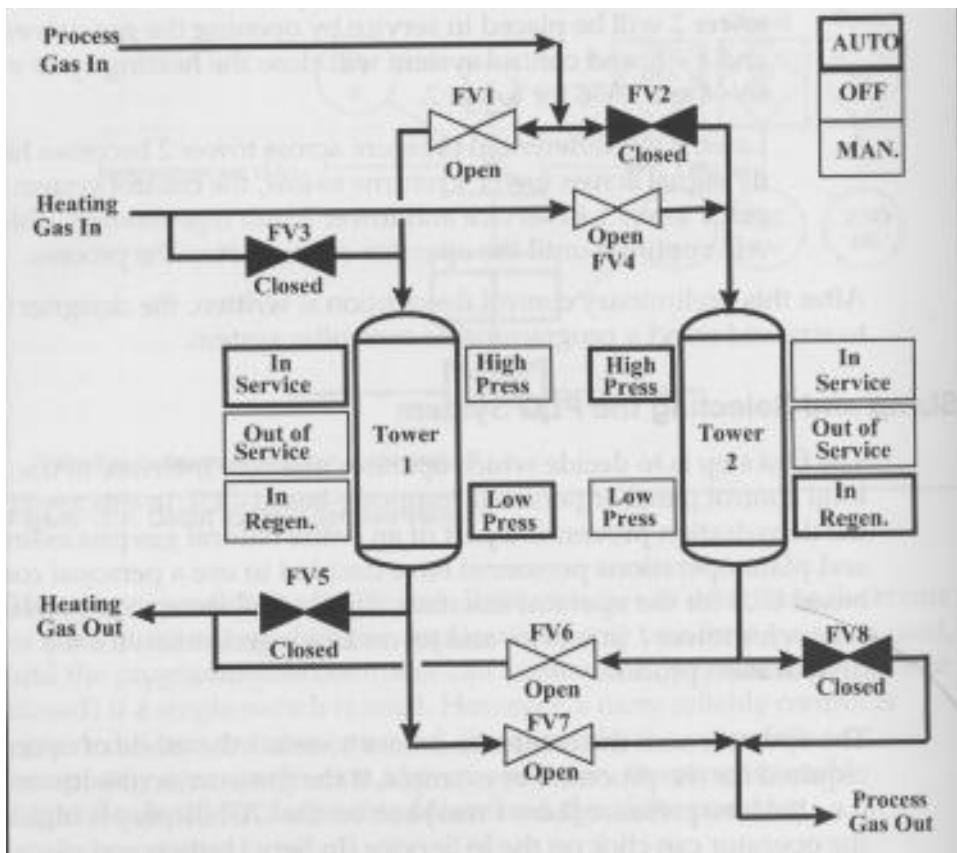
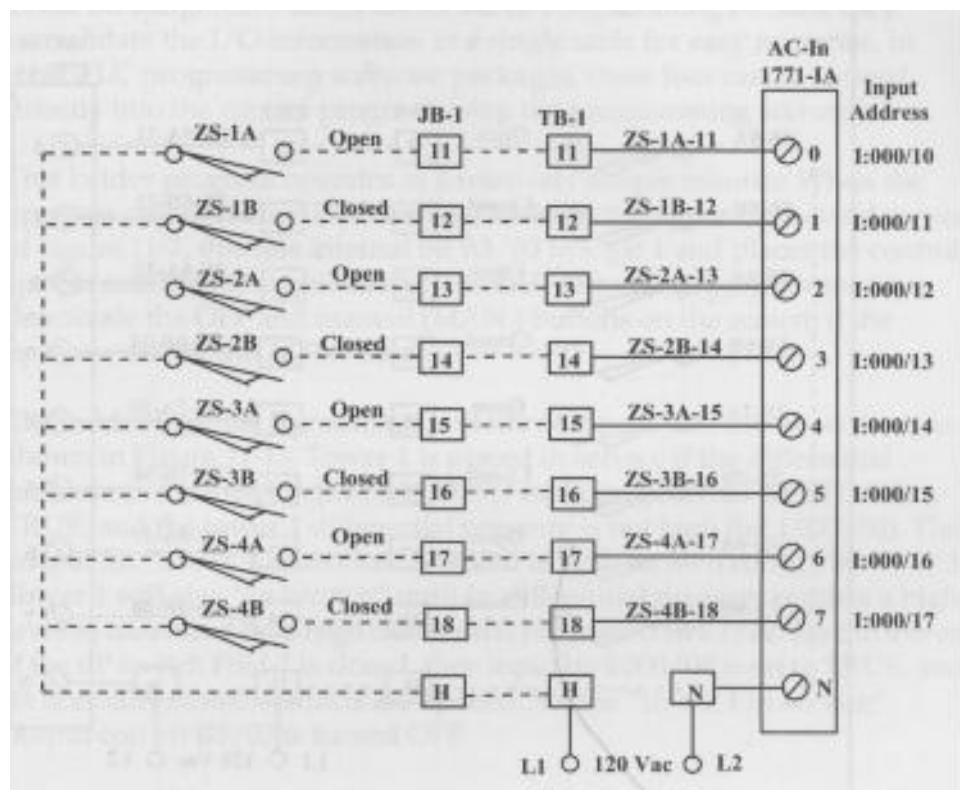
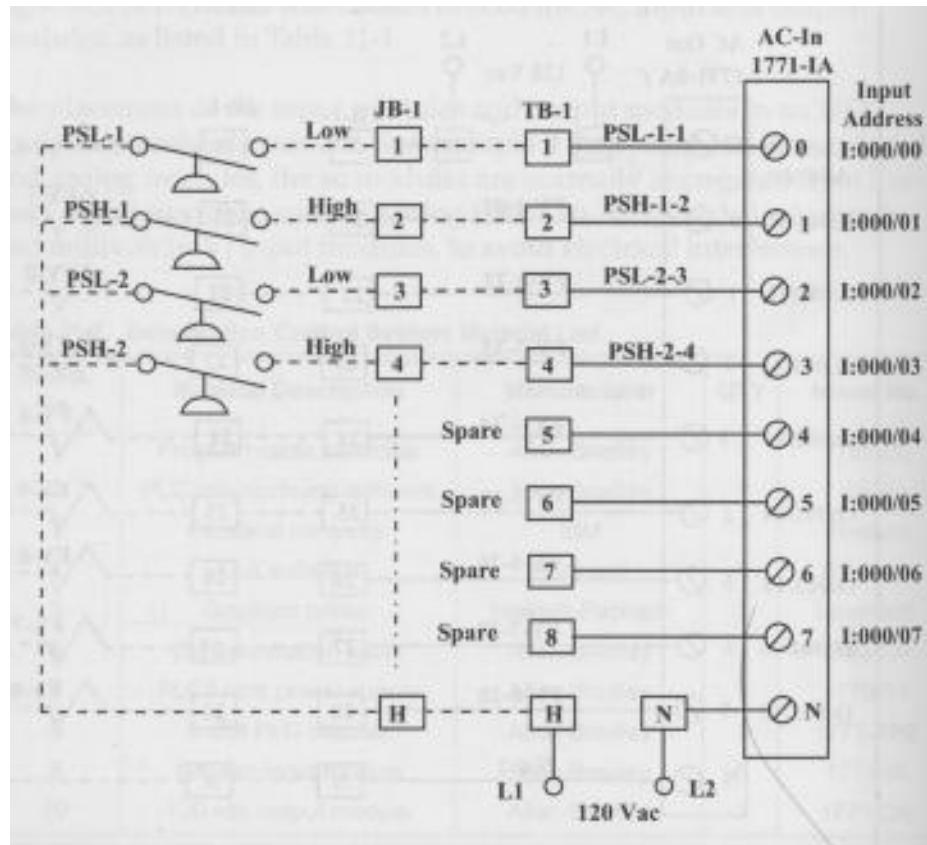


Figure 10-23. ISO/OSI seven-layer communications standard.





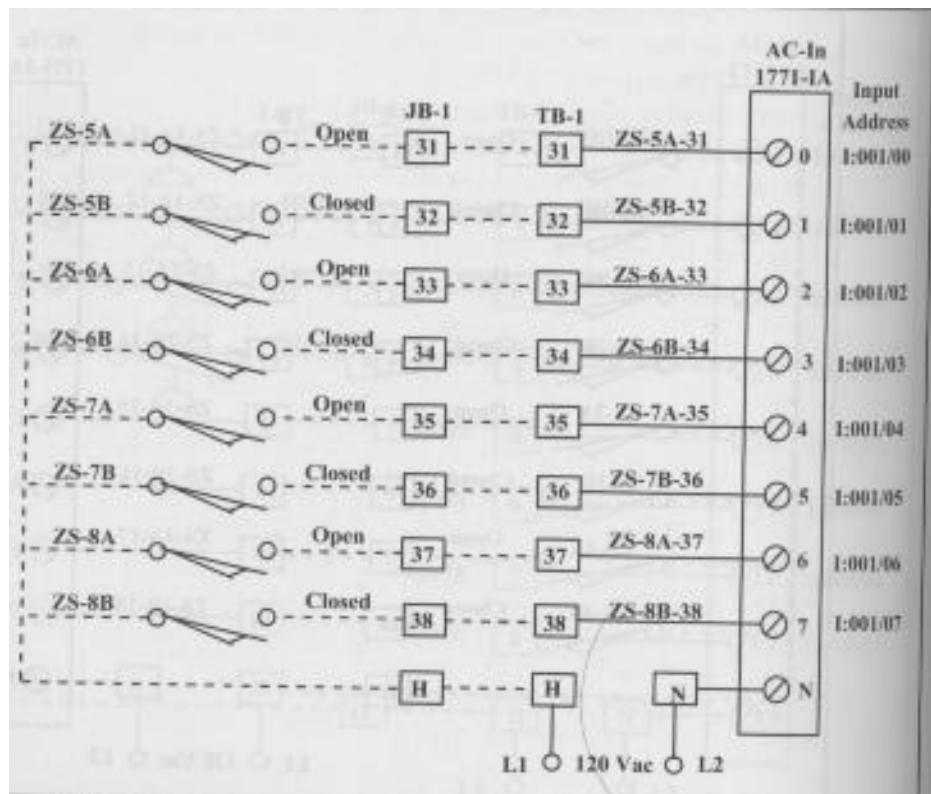
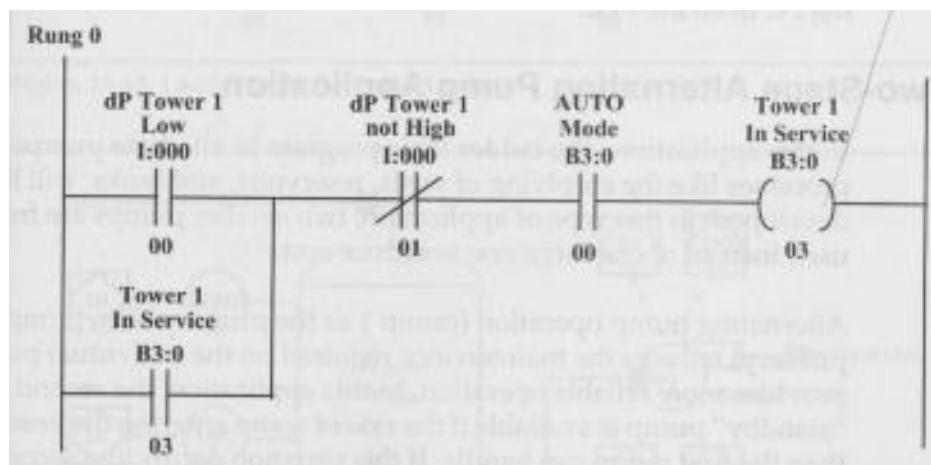
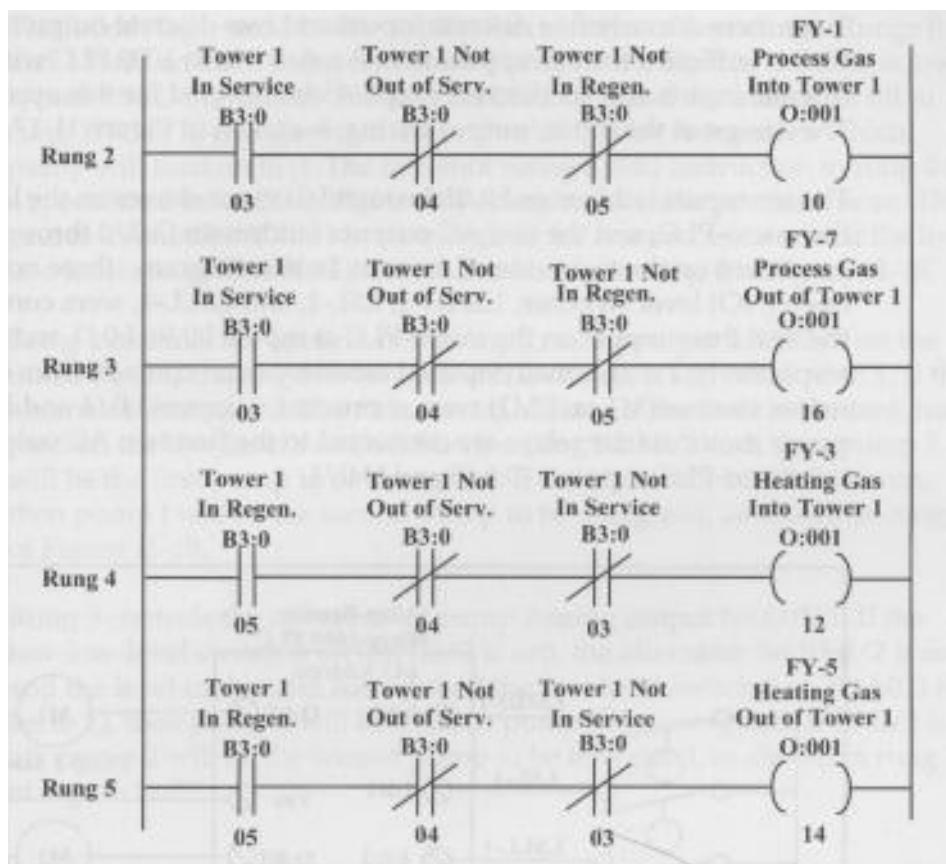
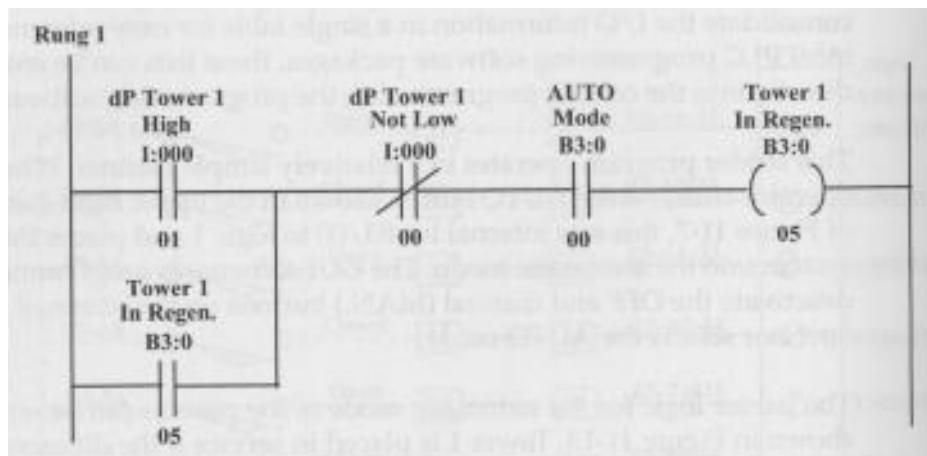
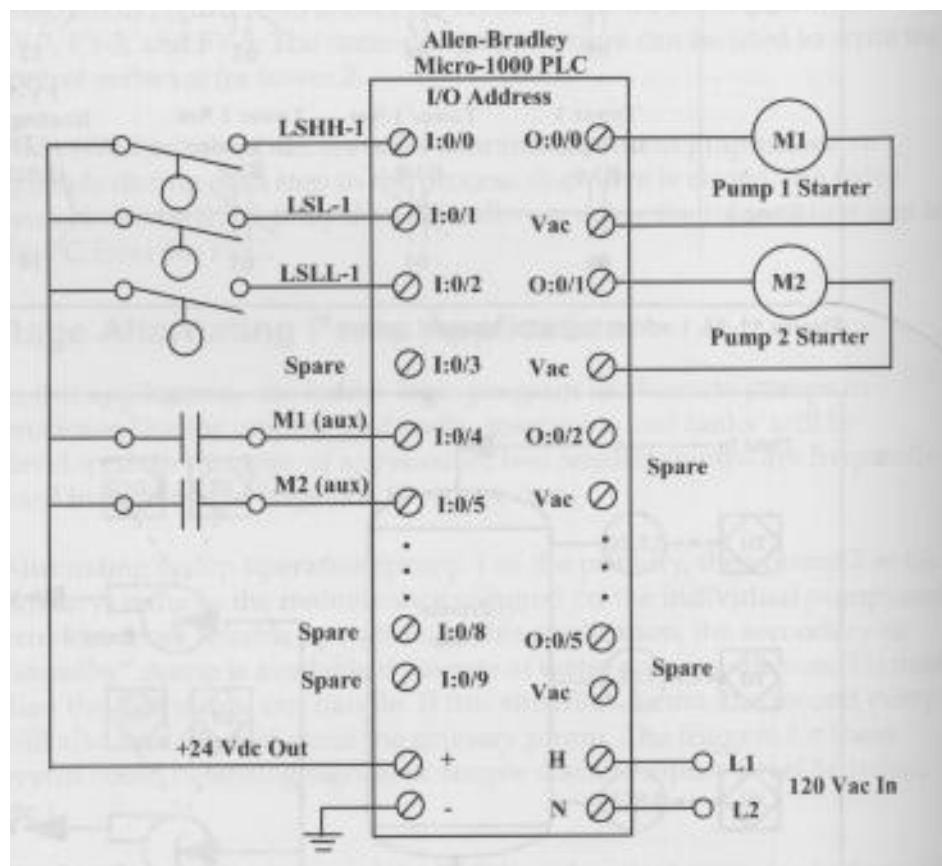
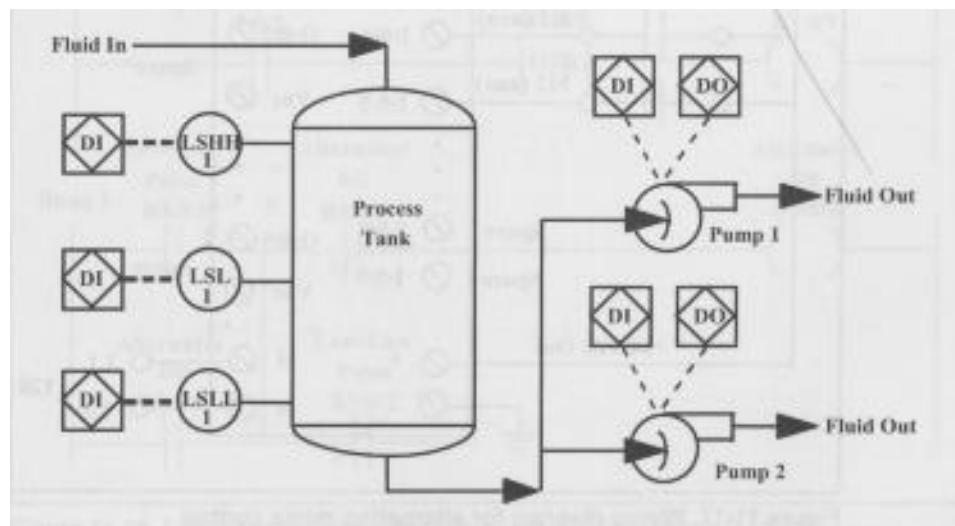


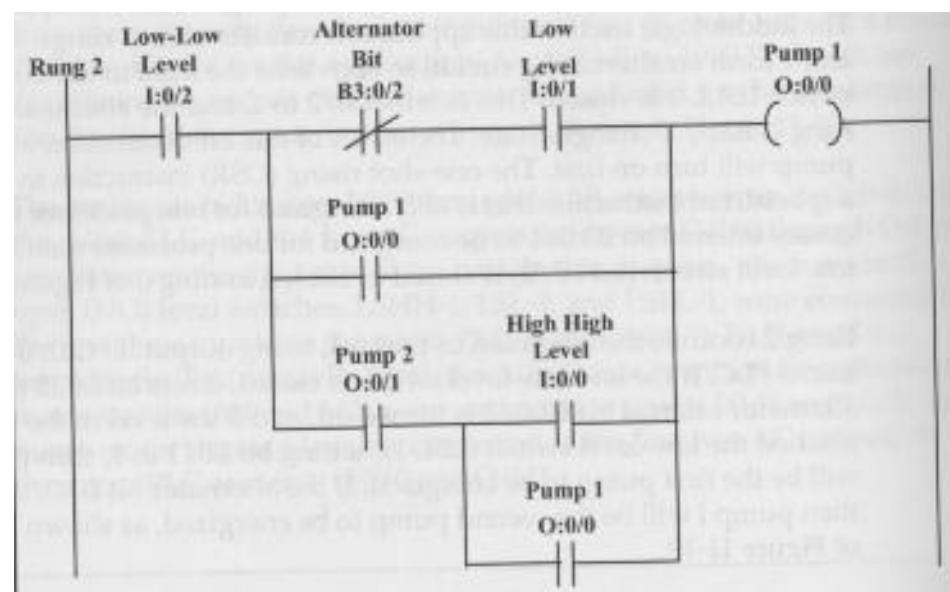
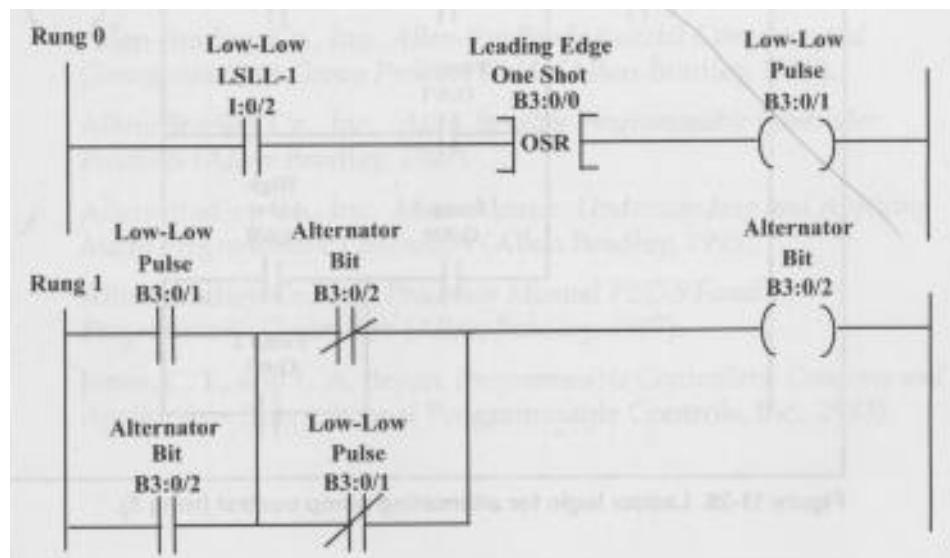
Table 11-2. GUI Bit Assignments

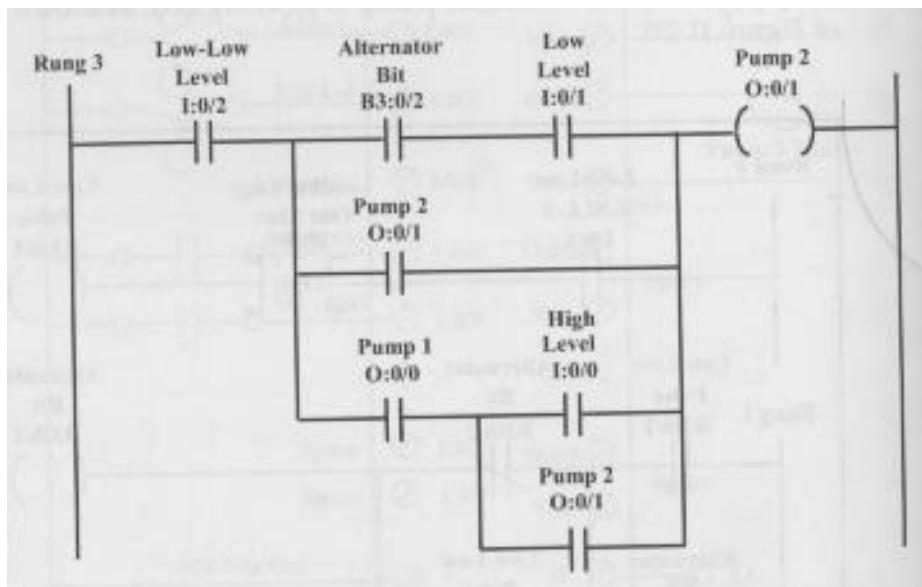
Bit Address	Description	Bit Address	Description
B3:0/00	Automatic mode	B3:0/05	T1In regeneration
B3:0/01	Off	B3:0/06	T2In service
B3:0/02	Manual mode	B3:0/07	T2Out of service
B3:0/03	T1in service	B3:0/08	T1In regeneration
B3:0/04	T1Out of service		











**Truyền thông đơn hướng**

**Truyền thông bán hai chiều**

**Truyền thông hai chiều**

**Phương pháp truyền tin**

**Truyền song song**

**Truyền nối tiếp**

**Bộ dòn tín hiệu**

**Bộ dòn kênh tần số**

**Dòn kênh thời gian**

**Dòn kênh bằng thống kê**

**Lõi điều khiển và kiểm tra**

**Kiểm tra ôn**

**Kiểm tra mức lượng dư đọc**

**Kiểm tra lượng dư ngang**

**Kiểm tra lượng dư trù**

**Các qui chuẩn truyền thông**

**Truyền nối tiếp đồng bộ**

**Qui phạm truyền nối tiếp đồng bộ**

**Qui phạm truyền nhị phân đồng bộ**

**Qui phạm HDLC**

**Mạng nội bộ LAN**

**Cấu hình mạng LAN**

**Qui phạm mạng LAN**

**Cấu trúc mạng tiêu chuẩn**

**Tiêu chuẩn phần cứng nối tiếp**

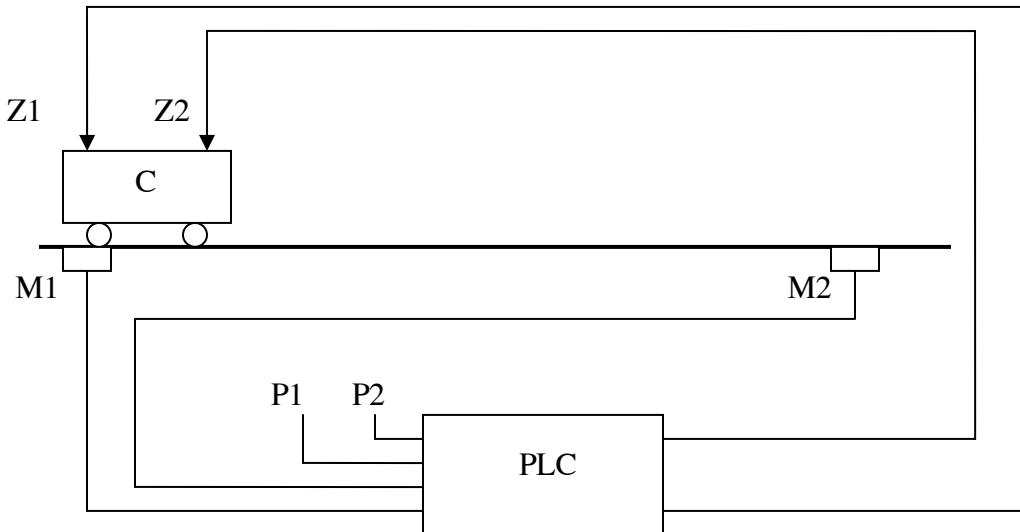
**Tiêu chuẩn EIA232**

**Tiêu chuẩn RS-422**  
**Tiêu chuẩn RS-485**  
**Tiêu chuẩn mạch vòng dong 20-mA**

## CHƯƠNG 9 CÁC ÚNG DỤNG CÔNG NGHIỆP

### 1. Điều khiển bằng tải:

**Ví dụ 9.1** Trong công nghiệp ta hay gặp vấn đề cần điều khiển xe cát phoi điều khiển tự động AGV (Automated Guided Vehicle) trong các phân xưởng. Một ví dụ đơn giản là ta cần điều khiển Xe C giữa hai điểm A và B. Các điểm này được chỉ thị bởi các công tắc mini M1 và M2. Xe được điều khiển bằng hai nút ấn P1 và P2. Ban đầu xe dừng tại điểm A, công tắc mini M1 được bật và xe sẽ đứng yên cho tới khi nào nút P1 được ấn. Khi đó đầu ra Z1 được kích hoạt và động cơ của xe được bật lên, xe bắt đầu chuyển động hướng đến điểm B. Chuyển động này sẽ tiếp tục ngay cả khi P1 hay P2 được ấn. Khi xe đạt đến điểm B, nó làm bật công tắc mini M2. Công tắc M2 bật làm cho đầu ra Z2 được kích hoạt và ngắt đầu ra Z1. Xe bắt đầu chuyển động theo chiều ngược lại từ B đến A. Trong quá trình chuyển động nếu ta ấn nút P2, xe sẽ đảo chiều chuyển động và lại chuyển động theo hướng đến B. Như vậy là đầu ra Z2 được ngắt và đầu ra Z1 được kích hoạt. Nếu P2 không được bấm, xe tiếp tục đi về đến điểm A và sẽ dừng khi công tắc mini M1 được bật.



Hình 1\*\*\*

Hệ thống điều khiển hoạt động khi bắt đầu công tắc nguồn trên PLC được bật và trạng thái khởi động được kích hoạt. Đầu ra Z1 và Z2 đang ở trạng thái tắt. Nếu nút ấn P1 chưa được ấn, xe ở nguyên ở trạng thái khởi động. Khi nút ấn P1 chuyển từ trạng thái 0 sang 1, thiết bị điều khiển thay đổi trạng thái bên trong nó và đầu ra Z1 được kích hoạt. Tại B, khi xe đến thì công tắc mini M2 chuyển từ 0 sang 1, Z1 được tắt và Z2 được kích hoạt. Xe chuyển động về A. Nếu nút P2 không được ấn, xe chạy về đến A, công tắc mini M1 chuyển trạng thái từ 0 sang 1, xe dừng tại vị trí khởi động. Nếu nút P2 được ấn trước khi

xe quay về điểm A, Z2 tắt và Z1 được kích hoạt, xe quay lại điểm B và lặp lại hành trình từ B về A.

Nếu thể hiện chức năng của hệ thống điều khiển bằng các hàm lô gíc thì ta sẽ có:

$$Z_1 = M_1 \cdot P_1 + \overline{M}_1 \cdot M_2 + \overline{Z}_2 \cdot P_2$$

$$Z_2 = (M_2 + \overline{M}_1) \cdot \overline{P}_2 \cdot \overline{Z}_1$$

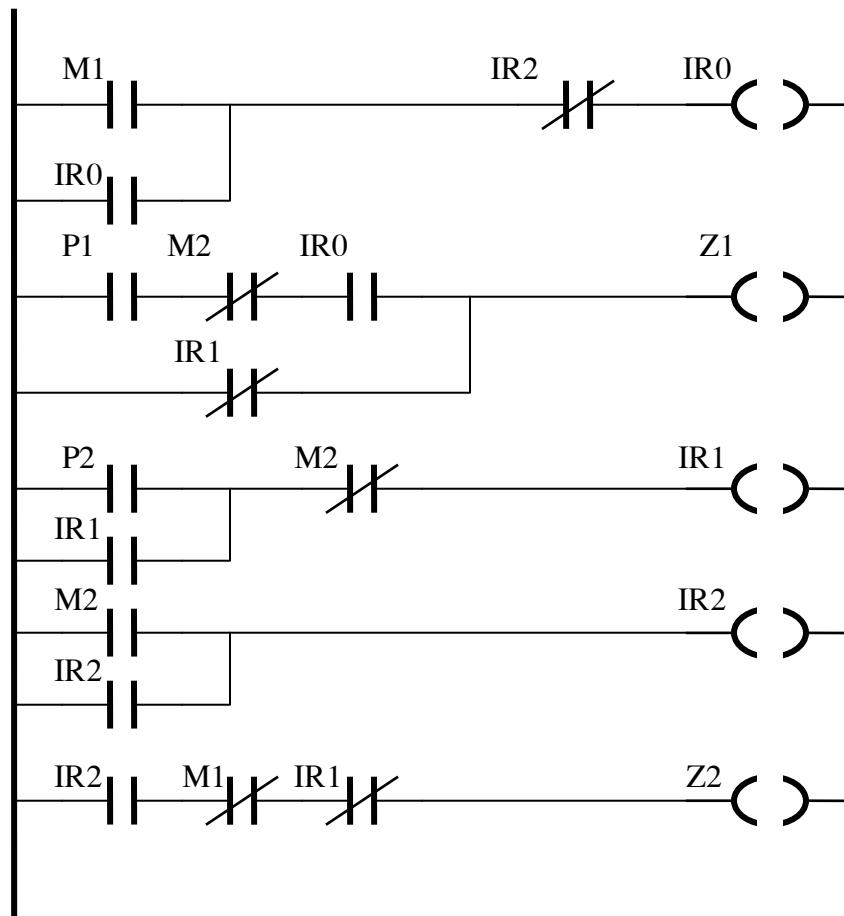
Chương trình điều khiển của hệ thống viết bằng ngôn ngữ bảng lệnh cơ bản sẽ có dạng:

```

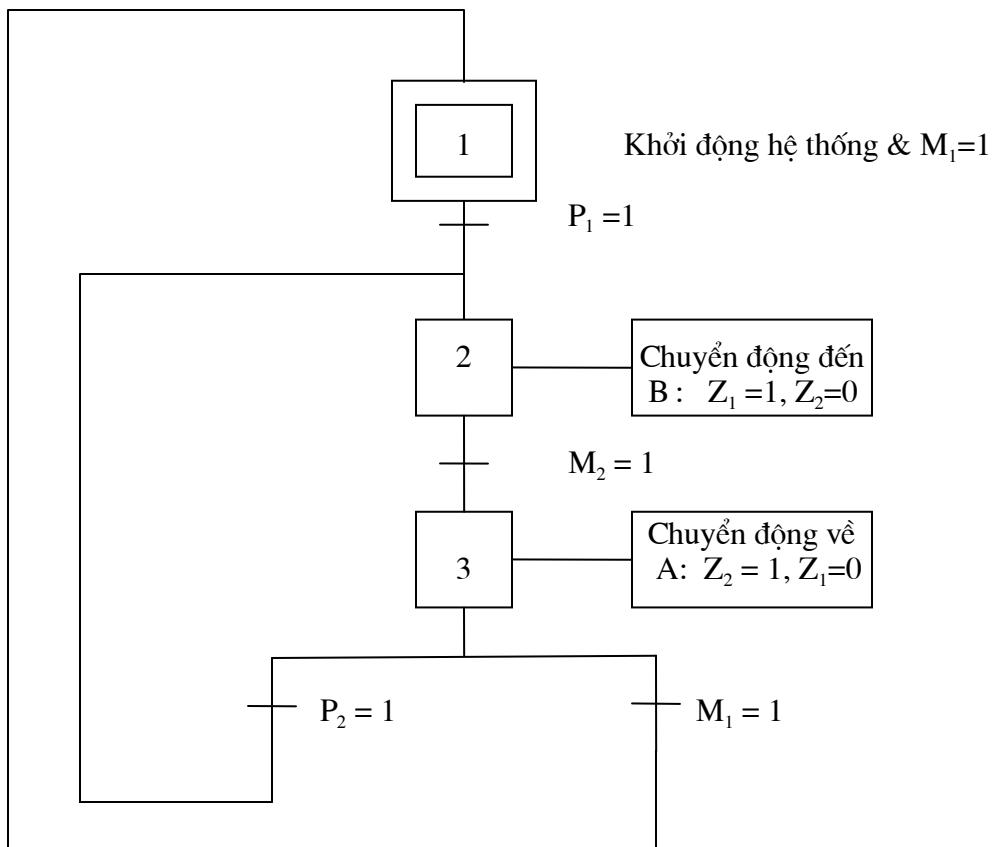
STR M1
AND P1
STR NOT M1
AND M2
OR STR
STR NOT Z2
AND P2
OR STR
OUT Z1
STR M2
OR NOT M1
AND NOT P2
AND NOT Z1
OUT Z2

```

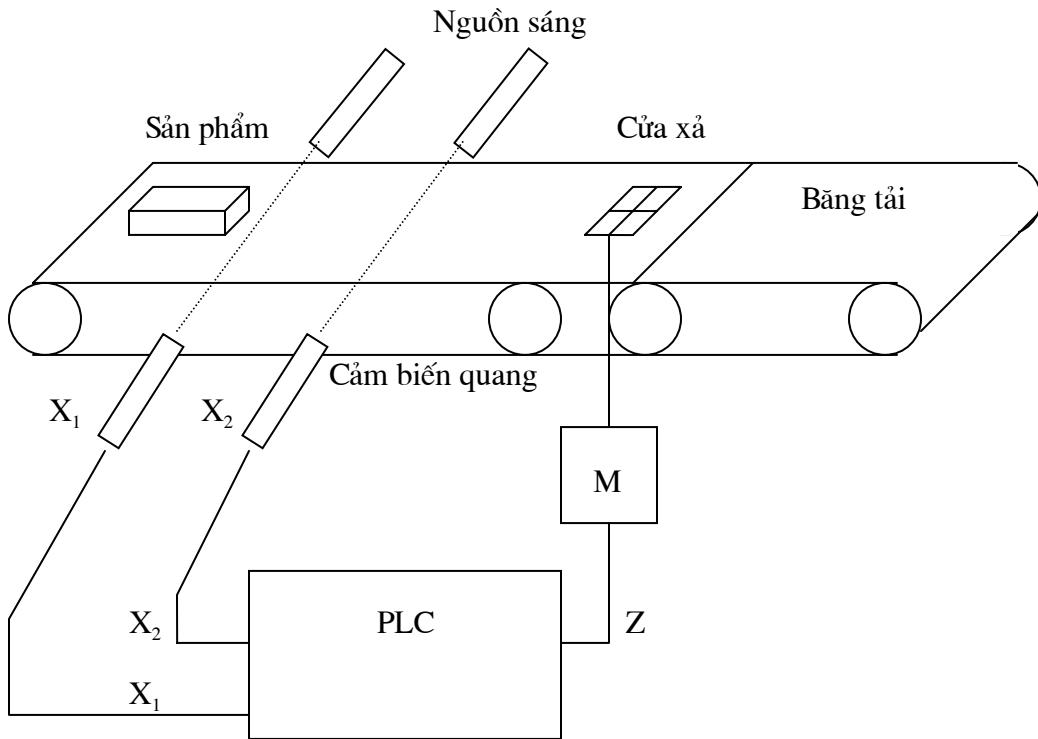
Chương trình điều khiển viết trong ngôn ngữ LAD:



Chương trình thể hiện bằng ngôn ngữ Grafset:



Ví dụ 9.2. Một sản phẩm cơ khí dạng thanh sản xuất trên dây chuyền tự động được chuyển ra trên băng tải. Chiều dài của sản phẩm không được vượt qua kích thước L. Hệ thống phân loại sản phẩm (hình \*\*.) bao gồm băng tải, hai cảm biến quang điện đặt cách nhau một khoảng đúng bằng L. Các cảm biến này gồm một đầu phát sáng và một đầu thu. Đầu ra của các cảm biến quang điện này là tín hiệu điện áp tương ứng với mức lô gíc 0 hay 1, khi có sản phẩm đi qua giữa chúng và các đầu phát. Phía sau cảm biến thứ hai là một cửa xả, điều khiển băng động cơ M. Nếu thanh sản phẩm nào dài hơn L, động cơ phải được kích hoạt để thanh sản phẩm đó rơi xuống. Nếu sản phẩm không dài hơn L, động cơ M không hoạt động và để thanh sản phẩm đi ra. Một khi thanh sản phẩm đã rơi, động cơ M được tắt và cửa xả được đóng lại và hệ thống lại sẵn sàng cho chọn sản phẩm tiếp theo. Khoảng cách giữa các thanh sản phẩm để lựa chọn, phải đảm bảo được điều kiện là chỉ có 1 thanh đi vào vùng kiểm tra mỗi lần. Vấn đề ở đây là thiết kế hệ thống điều khiển lô gíc, mà đầu vào của nó là các đầu ra từ cảm biến x1 và x2 và đầu ra Z kích hoạt động cơ M.



Hình \*\*. Hệ thống phân loại sản phẩm

Như vậy ta thấy rằng khi khởi động hệ thống thì  $X_1=0$  và  $X_2=0$ . Nếu sản phẩm đi qua các cảm biến thì có thể  $X_1=1$ , sau đó  $X_2=1$ . Nếu  $X_1=1$  khi  $X_2$  chuyển từ 0 thành 1, thì sản phẩm này đã dài hơn yêu cầu. Động cơ dừng khi  $X_2=0$

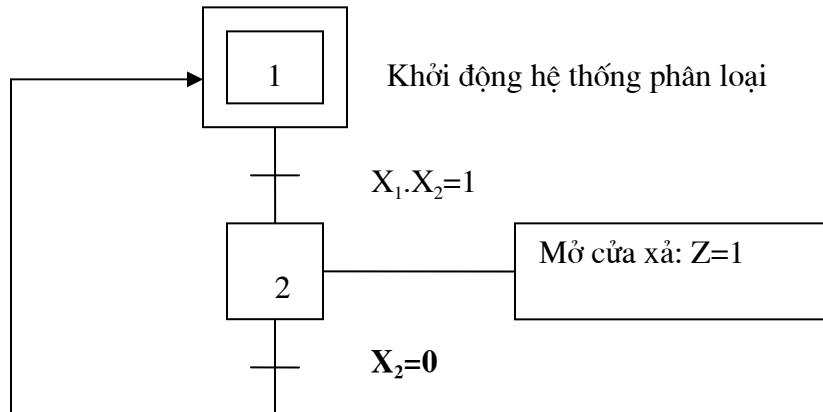
Chương trình trong ngôn ngữ bảng lệnh cơ bản sẽ có dạng:

```

STR X1
AND X2
OUT Z

```

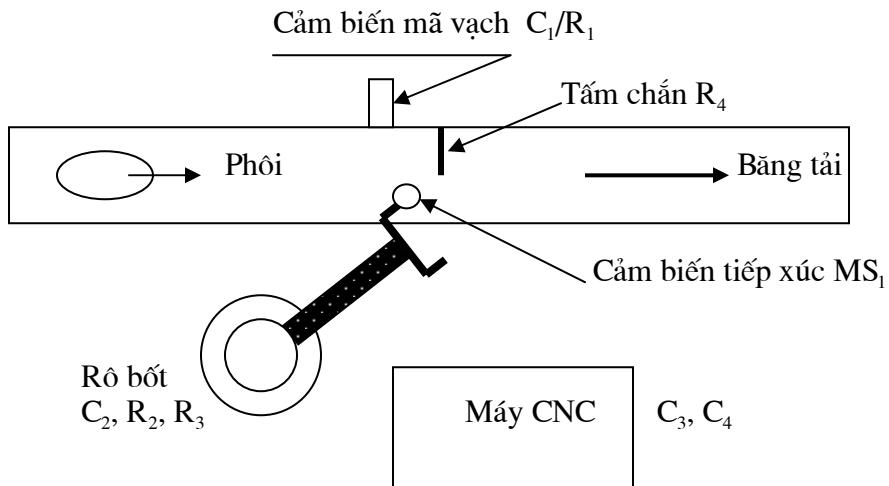
Chương trình bằng ngôn ngữ Grafcet:



## 1. Điều khiển cấp phôi tự động cho máy CNC

### 4.1 ÚNG DỤNG PLC TRONG ĐIỀU KHIỂN HỆ THỐNG CẤP PHÔI CHO MÁY CÔNG CỤ

Ta có hệ thống tự động cấp phôi cho máy công cụ phục vụ bởi một rô bốt công nghiệp như hình vẽ.



Rô bốt cấp phôi vào máy CNC và lấy chi tiết từ máy CNC ra băng tải. Khi phôi chạy trên băng tải, chạm vào cảm biến tiếp xúc MS<sub>1</sub>, cảm biến mã vạch sẽ quét để nhận dạng phôi. Nếu đúng chi tiết yêu cầu, tấm chắn sẽ được đẩy ra để dừng phôi lại. Rô bốt sẽ kẹp phôi và gá vào máy nếu máy đang ở trạng thái không làm việc. Ngược lại, rô bốt phải chờ để tháo chi tiết đã gia công xong từ máy CNC ra băng tải. Các thành phần của hệ thống điều khiển gồm:

TT	Ký hiệu	Mô tả	Trạng thái	Chú thích
1	MS <sub>1</sub>	Cảm biến tiếp xúc	1	Phôi đến
2	R <sub>1</sub>	Đầu ra của cảm biến mã vạch	1	Quét mã vạch
3	C <sub>1</sub>	Đầu vào của cảm biến mã vạch	1	Đúng loại phôi
4	R <sub>2</sub>	Đầu ra của rô bốt	1	Gá lắp phôi
5	R <sub>3</sub>	Đầu ra của rô bốt	1	Tháo chi tiết
6	C <sub>2</sub>	Đầu vào của rô bốt	1	Rô bốt đang bận
7	R <sub>4</sub>	Đầu ra của cơ cấu đẩy tấm chắn	1	Đẩy tấm chắn ra
8	C <sub>3</sub>	Đầu vào từ máy CNC	1	Máy CNC đang bận
9	C <sub>4</sub>	Đầu vào từ máy CNC	1	Gia công xong

2. Điều khiển thang máy
3. Điều khiển máy rửa xe
4. Điều khiển lò hơi

Hệ thống cấp dầu cho lò hơi trong các nhà máy nhiệt điện hoạt động theo nguyên tắc sau:

1. Dầu trong thùng chứa được sấy đến một nhiệt độ nhất định bằng bộ sấy bằng điện (hoặc bằng chính hơi nước). Cảm biến nhiệt TSL sẽ chỉ thị mức nhiệt độ cần sấy (TSL=1).
2. Trong thùng chứa dầu còn có cảm biến mức dầu LSL. Nếu cảm biến này báo mức dầu thấp hơn mức cho phép, phải dừng bơm P (LSL=1, PP=1), van điện từ XV3 mở để dầu trở về thùng chứa và đồng thời đèn đỏ RL được bật lên để báo hiệu hệ thống đang bị dừng (XV3=1, RL=1).
3. Nếu áp suất của dầu tồn thất qua bộ lọc F, làm tăng áp trên đầu ra của bơm (cảm biến cao áp DPSH = 1) hay áp suất dầu trên đầu ra của bộ lọc giảm (cảm biến áp thấp PSL=1), như vậy hoạt động thứ hai được lặp lại.
4. Nếu áp suất dầu trên đầu ra của bộ lọc vượt quá một mức cho phép (cảm biến PSH=1), van điện từ XV3 phải được mở (XV3=1).
5. Nếu một trong các súng phun nào hoạt động (BS1=1 hay BS2=1) thì các van điện từ tương ứng XV1 hay XV2 phải được đóng (XV1=0 hay XV2=0) và XV3 phải được mở (XV3=1).
6. Nếu không có súng phun nào hoạt động (BS1=1 và BS2=1), hệ thống phải dừng hoạt động như ở bước 2.
7. Trong điều kiện hoạt động bình thường đèn xanh GL được bật (GL=1).

Với hoạt động như vậy, điều kiện để bơm ngừng hoạt động có thể biểu diễn bằng hàm lô gíc sau:

$$PP = LSL + TSL + DPSH + PSL + BS1.BS2$$

$$RL = PP$$

$$GL = \overline{PP}$$

$$XV1 = BS1$$

$$XV2 = BS2$$

$$XV3 = LSL + TSL + DPSH + PSL + BS1 + BS2$$

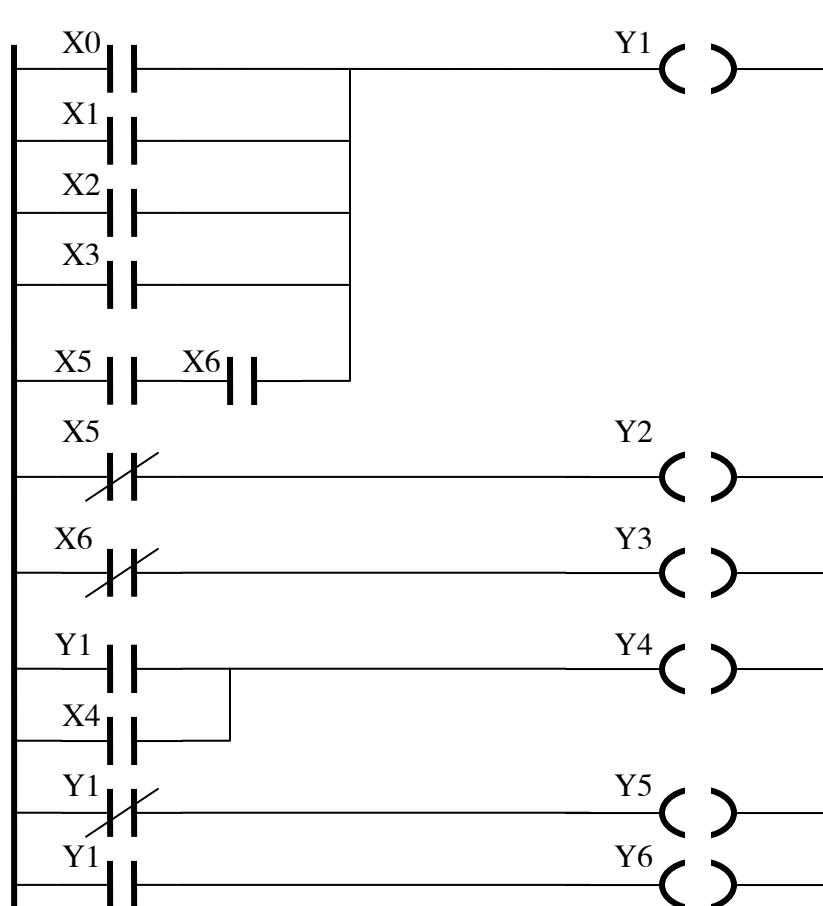
Qui ước các biến vào/ra được thể hiện trên bảng sau:

<b>STT</b>	<b>Tên biến</b>	<b>Ký hiệu biến cho PLC</b>
1	LSL	X0
2	TSL	X1
3	DPSH	X2
4	PSL	X3
5	PSH	X4
6	BS1	X5
7	BS2	X6
8	PP	Y1
9	XV1	Y2
10	XV2	Y3
11	XV3	Y4
12	GL	Y5
13	RL	Y6

Chương trình thể hiện bằng ngôn ngữ bảng lệnh:

```
STR  X0
OR   X1
OR   X2
OR   X3
STR  X5
AND  X6
OR STR
OUT  Y1
STR NOT  X5
OUT   Y2
STR NOT  X6
OUT   Y3
STR   Y1
OR    X4
OUT   Y4
STR NOT  Y1
OUT   Y5
STR   Y1
OUT   Y6
```

Ngôn ngữ LAD:



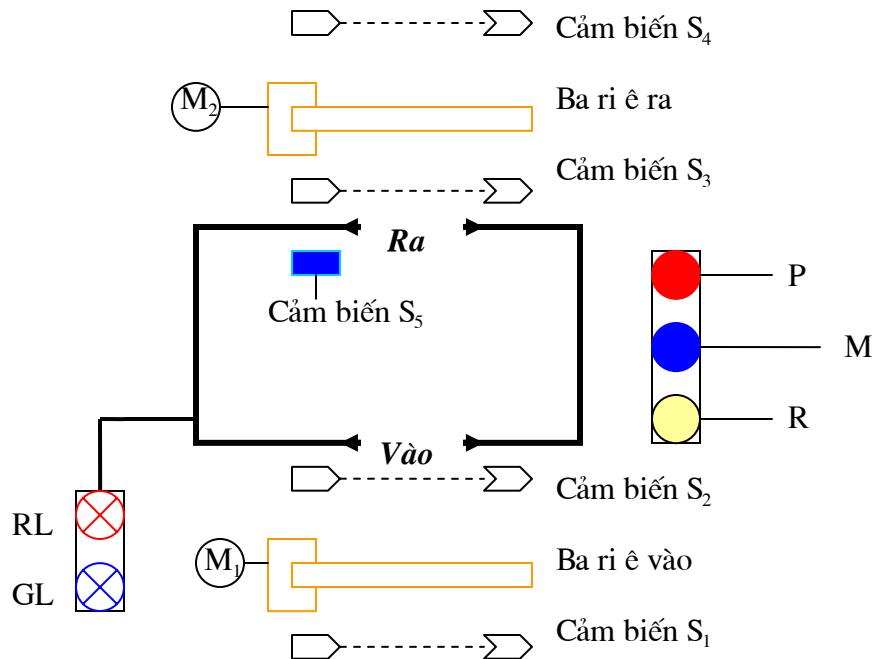
## 5. Điều khiển ga ra tự động

Có một ga ra tự động điều khiển bằng PLC. Ga ra này có dung lượng chứa là 10 xe. Trên đầu vào và đầu ra của ga ra có hai barie điều khiển bằng hai động cơ M1 và M2. Mỗi barie điều khiển có hai cảm biến đặt phía trước và phía sau của barie để báo có xe phía trước hay xe đã đi qua barie (S1, S2, S3 và S4). Không bao giờ đồng thời cả S1 và S2 hoặc S3 và S4 đều được bật đồng thời. Trên đầu ra có một cảm biến S5 để kiểm soát vé ra. Hệ thống điều khiển của ga ra sẽ điều khiển các hoạt động sau:

1. Đóng và mở barie tự động. Barie vào sẽ mở nếu trong ga ra có ít hơn 10 xe và cảm biến S1 = 1, báo có xe vào trước barie. Khi xe đi qua barie sẽ đóng lại nếu cảm biến S2 chuyển trạng thái từ 1 sang 0. Barie ra sẽ mở nếu S5 bật và S3 chuyển trạng thái từ 0 sang 1. Và barie ra đóng lại khi S4 chuyển từ trạng thái 1 sang 0, báo xe đã đi qua.
2. Trên đầu vào của ga ra nếu đèn xanh bật, báo hiệu ga ra đang còn chỗ.
3. Nếu trên đầu vào có đèn đỏ, báo hiệu ga ra đã đầy.

Hệ thống điều khiển có các đầu vào sau:

- Nút khởi động M: từ khi nguồn được cấp vào PLC, không có một xe nào được vào hay ra khi chưa ấn nút này.
- Nút dừng P: nếu nút này được bật, không xe nào được vào hay ra khi chưa ấn lại nút M. Trường hợp cả hai nút P và M ấn đồng thời, thì nút P được ưu tiên.
- Nút ấn R: để đặt lại số của bộ đếm xe khi PLC được bật lên.



Hình \*\*..

Để xác định được chương trình điều khiển của hệ thống này, trước hết ta phải qui ước các biến vào/ra:

<i>Biến ngoài</i>	<i>Biến trong PLC</i>
S <sub>1</sub>	X <sub>0</sub>
S <sub>2</sub>	X <sub>1</sub>
S <sub>3</sub>	X <sub>2</sub>
S <sub>4</sub>	X <sub>3</sub>
S <sub>5</sub>	X <sub>4</sub>
M	X <sub>5</sub>
P	X <sub>6</sub>
R	X <sub>7</sub>
M <sub>1</sub>	Y <sub>0</sub>
M <sub>2</sub>	Y <sub>1</sub>
GL	Y <sub>2</sub>
RL	Y <sub>3</sub>

Các hàm lô gíc tương ứng với các chức năng của hệ thống điều khiển gồm:

$$Y_0 = X_5 \cdot X_0 \cdot Y_2$$

$$Y_1 = X_5 \cdot X_4 \cdot X_2$$

$$Y_2 = X_5 \cdot \overline{CTR}$$

$$Y_3 = X_5 \cdot CTR$$

Do cần sử dụng bộ đếm, cho nên ta phải sử dụng các biến trung gian để tạo xung và tạo các mạch duy trì các trạng thái của các biến vào. Trong chương trình này sử dụng 6 biến trung gian từ IR<sub>0</sub> đến IR<sub>5</sub>.

Chương trình PLC bằng ngôn ngữ bảng lệnh cơ bản của hệ thống điều khiển được thể hiện như sau:

STR	X <sub>5</sub>	
OR	IR <sub>0</sub>	( Tạo mạch duy trì)
AND NOT	X <sub>6</sub>	
OUT	IR <sub>0</sub>	

STR	X <sub>0</sub>	
OR	Y <sub>0</sub>	
AND NOT	IR <sub>2</sub>	(Gán mạch duy trì cho đầu ra Y <sub>0</sub> )
AND	IR <sub>0</sub>	
AND NOT	Y <sub>3</sub>	
OUT	Y <sub>0</sub>	

STR	X <sub>1</sub>	
OR	IR <sub>1</sub>	
AND NOT	IR <sub>2</sub>	(Tạo xung đếm xe vào ga ra cho bộ đếm và
AND	Y <sub>0</sub>	hẹn giờ)
OUT	IR <sub>1</sub>	
STR NOT	X <sub>1</sub>	

AND	IR <sub>1</sub>	
OUT	IR <sub>2</sub>	
STR	X <sub>4</sub>	
OR	IR <sub>3</sub>	
AND NOT	IR <sub>5</sub>	(Tạo mạch duy trì cho trạng thái soát vé ra)
AND	IR <sub>0</sub>	
OUT	IR <sub>3</sub>	
STR	IR <sub>3</sub>	
AND	X <sub>2</sub>	
OR	Y <sub>1</sub>	( Tạo mạch duy trì cho đầu ra Y <sub>1</sub> )
AND NOT	IR <sub>5</sub>	
OUT	Y <sub>1</sub>	
STR	X3	
AND	Y1	
OR	IR4	
AND NOT	IR5	(Tạo xung đếm cho xe ra )
AND	IR0	
OUT	IR4	
STR NOT	X3	
AND	IR4	
OUT	IR5	
STR	X7	
AND	Y1	
STR	IR5	
STR	IR2	( Đếm xe và bật đèn đỏ khi số xe bằng 10)
OR	IR5	
CTR	45	
	10	
OUT	Y3	
STR	IR0	
AND NOT	Y3	( Tạo tín hiệu đèn xanh)
OUT	Y2	
END		

Chương trình viết bằng ngôn ngữ LAD:

