

## **CHƯƠNG 1**

### **BỘ ĐIỀU KHIỂN LẬP TRÌNH**

#### **I. Tổng quan về PLC**

##### **1. Giới thiệu PLC**

Thiết bị điều khiển lập trình đầu tiên (programmable controller) đã được những nhà thiết kế cho ra đời năm 1968 (Công ty General Motor - Mỹ). Tuy nhiên, hệ thống này còn khá đơn giản và cồng kềnh, người sử dụng gặp nhiều khó khăn trong việc vận hành hệ thống. Vì vậy các nhà thiết kế từng bước cải tiến hệ thống đơn giản, gọn nhẹ, dễ vận hành, nhưng việc lập trình cho hệ thống còn khó khăn, do lúc này không có các thiết bị lập trình ngoại vi hỗ trợ cho công việc lập trình.

Để đơn giản hóa việc lập trình, hệ thống điều khiển lập trình cầm tay (programmable controller handle) đầu tiên được ra đời vào năm 1969. Điều này đã tạo ra một sự phát triển thật sự cho kỹ thuật điều khiển lập trình. Trong giai đoạn này các hệ thống điều khiển lập trình (PLC) chỉ đơn giản nhằm thay thế hệ thống Relay và dây nối trong hệ thống điều khiển cổ điển. Qua quá trình vận hành, các nhà thiết kế đã từng bước tạo ra được một tiêu chuẩn mới cho hệ thống, tiêu chuẩn đó là: dạng lập trình dùng giản đồ hình thang (The ladder diagram format). Trong những năm đầu thập niên 1970, những hệ thống PLC còn có thêm khả năng vận hành với những thuật toán hỗ trợ (arithmetic), “vận hành với các dữ liệu cập nhật” (data manipulation). Do sự phát triển của loại màn hình dùng cho máy tính (Cathode Ray Tube: CRT), nên việc giao tiếp giữa người điều khiển để lập trình cho hệ thống càng trở nên thuận tiện hơn.

Sự phát triển của hệ thống phần cứng và phần mềm từ năm 1975 cho đến nay đã làm cho hệ thống PLC phát triển mạnh mẽ hơn với các chức năng mở rộng: hệ thống ngõ vào/ra có thể tăng lên đến 8.000 cổng vào/ra, dung lượng bộ nhớ chương trình tăng lên hơn 128.000 từ bộ nhớ (word of memory). Ngoài ra các nhà thiết kế còn tạo ra kỹ thuật kết nối với các hệ thống PLC riêng lẻ thành một hệ thống PLC chung, tăng khả năng của từng hệ thống riêng lẻ. Tốc độ xử lý của hệ thống được cải thiện, chu kỳ quét (scan) nhanh hơn làm cho hệ thống PLC xử lý tốt với những chức năng phức tạp số lượng cổng ra/vào lớn.

Trong tương lai hệ thống PLC không chỉ giao tiếp với các hệ thống khác thông qua CIM Computer Integrated Manufacturing) để điều khiển các hệ thống: Robot, Cad/Cam... ngoài ra các nhà thiết kế còn đang xây dựng các loại PLC với các chức

## ***Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH***

---

năng điều khiển “thông minh” (intelligence) còn gọi là các siêu PLC (super PLCs) cho tương lai.

PLC viết tắt của Programmable Logic Controller, là thiết bị điều khiển lập trình được (khả trình) cho phép thực hiện linh hoạt các thuật toán điều khiển logic thông qua một ngôn ngữ lập trình. Người sử dụng có thể lập trình để thực hiện một loạt trình tự các sự kiện. Các sự kiện này được kích hoạt bởi tác nhân kích thích (ngõ vào) tác động vào PLC hoặc qua các hoạt động có trễ như thời gian định thì hay các sự kiện được đếm. Một khi sự kiện được kích hoạt thật sự, nó bật ON hay OFF thiết bị điều khiển bên ngoài được gọi là thiết bị vật lý. Một bộ điều khiển lập trình sẽ liên tục “lập” trong chương trình do “người sử dụng lập ra” chờ tín hiệu ở ngõ vào và xuất tín hiệu ở ngõ ra tại các thời điểm đã lập trình.

Để khắc phục những nhược điểm của bộ điều khiển dùng dây nối (bộ điều khiển bằng Relay) người ta đã chế tạo ra bộ PLC nhằm thỏa mãn các yêu cầu sau :

- ◆ Lập trình dễ dàng, ngôn ngữ lập trình dễ học.
- ◆ Gọn nhẹ, dễ dàng bảo quản, sửa chữa.
- ◆ Dung lượng bộ nhớ lớn để có thể chứa được những chương trình phức tạp.
- ◆ Hoàn toàn tin cậy trong môi trường công nghiệp.
- ◆ Giao tiếp được với các thiết bị thông minh khác như : máy tính, nối mạng, các module mở rộng.
- ◆ Giá cả cá thể cạnh tranh được.

Trong PLC, phần cứng CPU và chương trình là đơn vị cơ bản cho quá trình điều khiển hoặc xử lý hệ thống. Chức năng mà bộ điều khiển cần thực hiện sẽ được xác định bởi một chương trình. Chương trình này được nạp sẵn vào bộ nhớ của PLC, PLC sẽ thực hiện việc điều khiển dựa vào chương trình này. Như vậy nếu muốn thay đổi hay mở rộng chức năng của qui trình công nghệ, ta chỉ cần thay đổi chương trình bên trong bộ nhớ của PLC. Việc thay đổi hay mở rộng chức năng sẽ được thực hiện một cách dễ dàng mà không cần một sự can thiệp vật lý nào so với các bộ dây nối hay Relay.

## **Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH**

**Những ưu điểm kỹ thuật của bộ điều khiển PLC :**

<b>Chỉ tiêu so sánh</b>	<b>Role</b>	<b>Mạch số</b>	<b>Máy tính</b>	<b>PLC</b>
Giá thành từng chức năng.	Khá thấp	Thấp	Cao	Thấp
Kích thước vật lý	Lớn	Rất gọn	Khá gọn	Rất gọn
Tốc độ điều khiển	Chậm	Rất nhanh	Khá nhanh	Nhanh
Khả năng chống nhiễu	Rất tốt	Tốt	Khá tốt	Tốt
Lắp đặt	Mất thời gian thiết kế và lắp đặt.	Mất thời gian để thiết kế.	Lập trình phức tạp và tốn thời gian.	Lập trình và lắp đặt đơn giản.
Khả năng điều khiển các tác vụ phức tạp.	Không có	Có	Có	Có
Thay đổi, nâng cấp và điều khiển.	Rất khó	Khó	Khá đơn giản	Rất đơn giản
Công tác bảo trì	Kém	Kém	Kém	Tốt

Theo bảng so sánh ta nhận thấy được bộ điều khiển lập trình PLC với những ưu điểm về phần cứng và phần mềm có thể đáp ứng được hầu hết các yêu cầu chỉ tiêu trên. Mặt khác, PLC có khả năng kết nối mạng và kết nối các thiết bị ngoại vi rất cao giúp cho việc điều khiển được dễ dàng.

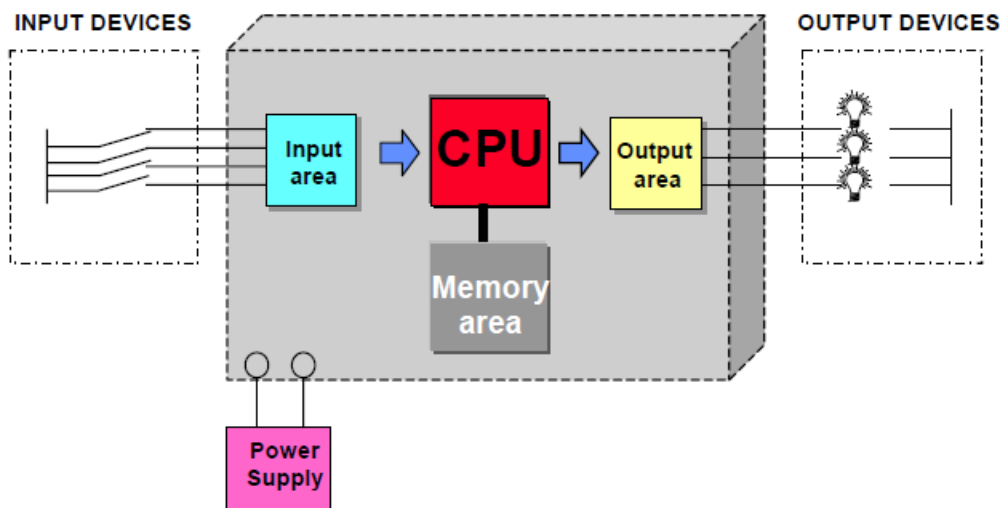
### **2. Cấu trúc của PLC**

Tất cả các PLC đều có thành phần chính là :

Một bộ nhớ chương trình RAM bên trong ( có thể mở rộng thêm một số bộ nhớ ngoài EPROM ).

Một bộ vi xử lý có cổng giao tiếp dùng cho việc ghép nối với PLC .

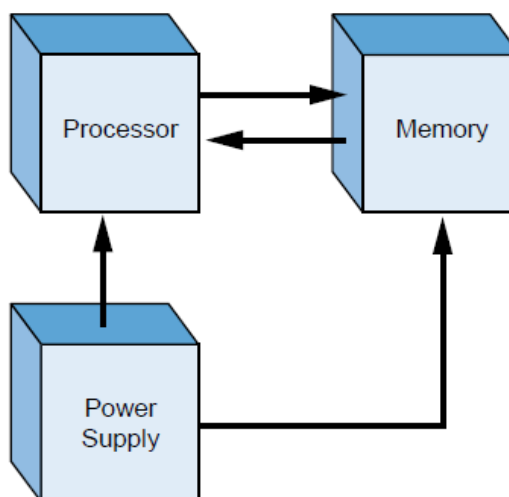
Các Module vào /ra.



Hình 1.1: Sơ đồ khối của hệ thống điều khiển

Bên cạnh đó, một bộ PLC hoàn chỉnh còn đi kèm thêm một đơn vị lập trình bằng tay hay bằng máy tính. Hầu hết các đơn vị lập trình đơn giản đều có đủ RAM để chứa đựng chương trình dưới dạng hoàn thiện hay bổ sung. Nếu đơn vị lập trình là đơn vị xách tay, RAM thường là loại CMOS có pin dự phòng, chỉ khi nào chương trình đã được kiểm tra và sẵn sàng sử dụng thì nó mới truyền sang bộ nhớ PLC. Đối với các PLC lớn thường lập trình trên máy tính nhằm hỗ trợ cho việc viết, đọc và kiểm tra chương trình. Các đơn vị lập trình nối với PLC qua cổng RS232, RS422, RS458, ...

Khối điều khiển trung tâm (CPU) gồm ba phần: bộ xử lý, hệ thống bộ nhớ và hệ thống nguồn cung cấp.



Hình 1.2: Sơ đồ khối tổng quát của CPU

#### **a. Đơn vị xử lý trung tâm**

CPU điều khiển các hoạt động bên trong PLC. Bộ xử lý sẽ đọc và kiểm tra chương trình được chứa trong bộ nhớ, sau đó sẽ thực hiện thứ tự từng lệnh trong

## ***Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH***

---

chương trình, sẽ đóng hay ngắt các đầu ra. Các trạng thái ngõ ra ấy được phát tới các thiết bị liên kết để thực thi. Và toàn bộ các hoạt động thực thi đó đều phụ thuộc vào chương trình điều khiển được giữ trong bộ nhớ.

### **b. Hệ thống bus**

Hệ thống Bus là tuyến dùng để truyền tín hiệu, hệ thống gồm nhiều đường tín hiệu song song :

**Address Bus** : Bus địa chỉ dùng để truyền địa chỉ đến các Module khác nhau.

**Data Bus** : Bus dùng để truyền dữ liệu.

**Control Bus** : Bus điều khiển dùng để truyền các tín hiệu định thời và điều khiển đồng bộ các hoạt động trong PLC .

Trong PLC các số liệu được trao đổi giữa bộ vi xử lý và các module vào ra thông qua Data Bus. Address Bus và Data Bus gồm 8 đường, ở cùng thời điểm cho phép truyền 8 bit của 1 byte một cách đồng thời hay song song.

Nếu một module đầu vào nhận được địa chỉ của nó trên Address Bus, nó sẽ chuyển tất cả trạng thái đầu vào của nó vào Data Bus. Nếu một địa chỉ byte của 8 đầu ra xuất hiện trên Address Bus, module đầu ra tương ứng sẽ nhận được dữ liệu từ Data bus. Control Bus sẽ chuyển các tín hiệu điều khiển vào theo dõi chu trình hoạt động của PLC .

Các địa chỉ và số liệu được chuyển lên các Bus tương ứng trong một thời gian hạn chế.

Hệ thống Bus sẽ làm nhiệm vụ trao đổi thông tin giữa CPU, bộ nhớ và I/O . Bên cạnh đó, CPU được cung cấp một xung Clock có tần số từ 1÷8 MHZ. Xung này quyết định tốc độ hoạt động của PLC và cung cấp các yếu tố về định thời, đồng hồ của hệ thống.

### **c. Bộ nhớ**

PLC thường yêu cầu bộ nhớ trong các trường hợp :

Làm bộ định thời cho các kênh trạng thái I/O.

Làm bộ đếm trạng thái các chức năng trong PLC như định thời, đếm, ghi các Relay.

Mỗi lệnh của chương trình có một vị trí riêng trong bộ nhớ, tất cả mọi vị trí trong bộ nhớ đều được đánh số, những số này chính là địa chỉ trong bộ nhớ .

Địa chỉ của từng ô nhớ sẽ được trỏ đến bởi một bộ đếm địa chỉ ở bên trong bộ vi xử lý. Bộ vi xử lý sẽ giá trị trong bộ đếm này lên một trước khi xử lý lệnh tiếp

## **Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH**

theo . Với một địa chỉ mới , nội dung của ô nhớ tương ứng sẽ xuất hiện ở đầu ra, quá trình này được gọi là quá trình đọc .

Bộ nhớ bên trong PLC được tạo bởi các vi mạch bán dẫn, mỗi vi mạch này có khả năng chứa  $2000 \div 16000$  dòng lệnh , tùy theo loại vi mạch. Trong PLC các bộ nhớ như RAM, EPROM đều được sử dụng .

RAM (Random Access Memory ) có thể nạp chương trình, thay đổi hay xóa bỏ nội dung bất kỳ lúc nào. Nội dung của RAM sẽ bị mất nếu nguồn điện nuôi bị mất . Để tránh tình trạng này các PLC đều được trang bị một pin khô, có khả năng cung cấp năng lượng dự trữ cho RAM từ vài tháng đến vài năm. Trong thực tế RAM được dùng để khởi tạo và kiểm tra chương trình. Khuynh hướng hiện nay dùng CMOSRAM nhờ khả năng tiêu thụ thấp và tuổi thọ lớn .

EPROM (Electrically Programmable Read Only Memory) là bộ nhớ mà người sử dụng bình thường chỉ có thể đọc chứ không ghi nội dung vào được . Nội dung của EPROM không bị mất khi mất nguồn , nó được gắn sẵn trong máy , đã được nhà sản xuất nạp và chứa hệ điều hành sẵn. Nếu người sử dụng không muốn mở rộng bộ nhớ thì chỉ dùng thêm EPROM gắn bên trong PLC . Trên PG (Programmer) có sẵn chỗ ghi và xóa EPROM.

Môi trường ghi dữ liệu thứ ba là đĩa cứng hoặc đĩa mềm, được sử dụng trong máy lập trình . Đĩa cứng hoặc đĩa mềm có dung lượng lớn nên thường được dùng để lưu những chương trình lớn trong một thời gian dài .

Kích thước bộ nhớ :

- ◆ Các PLC loại nhỏ có thể chứa từ  $300 \div 1000$  dòng lệnh tùy vào công nghệ chế tạo .
- ◆ Các PLC loại lớn có kích thước từ 1K ÷ 16K, có khả năng chứa từ  $2000 \div 16000$  dòng lệnh.

Ngoài ra còn cho phép gắn thêm bộ nhớ mở rộng như RAM , EPROM.

### **d. Các ngõ vào ra I/O**

Các đường tín hiệu từ bộ cảm biến được nối với các module vào (các đầu vào của PLC), các cơ cấu chấp hành được nối với các module ra (các đầu ra của PLC).

Hầu hết các PLC có điện áp hoạt động bên trong là 5V , tín hiệu xử lý là 12/24VDC hoặc 100/240VAC.

Mỗi đơn vị I/O có duy nhất một địa chỉ, các hiển thị trạng thái của các kênh I/O được cung cấp bởi các đèn LED trên PLC, điều này làm cho việc kiểm tra hoạt động nhập xuất trở nên dễ dàng và đơn giản.

## ***Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH***

Bộ xử lý đọc và xác định các trạng thái đầu vào (ON, OFF) để thực hiện việc đóng hay ngắt mạch ở đầu ra.

### **3. Các hoạt động xử lý bên trong PLC**

#### **a. Xử lý chương trình**

Khi một chương trình đã được nạp vào bộ nhớ của PLC, các lệnh sẽ được trong một vùng địa chỉ riêng lẻ trong bộ nhớ.

PLC có bộ đếm địa chỉ ở bên trong vi xử lý, vì vậy chương trình ở bên trong bộ nhớ sẽ được bộ vi xử lý thực hiện một cách tuần tự từng lệnh một, từ đầu cho đến cuối chương trình. Mỗi lần thực hiện chương trình từ đầu đến cuối được gọi là một chu kỳ thực hiện. Thời gian thực hiện một chu kỳ tùy thuộc vào tốc độ xử lý của PLC và độ lớn của chương trình. Một chu kỳ thực hiện bao gồm ba giai đoạn nối tiếp nhau:

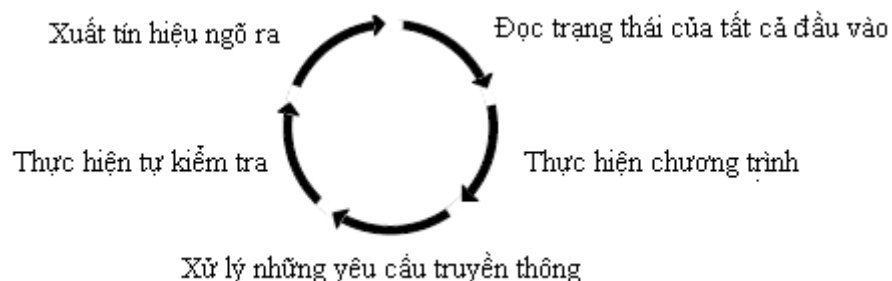
- ♦ **Đọc trạng thái của tất cả đầu vào:** PLC thực hiện lưu các trạng thái vật lý của ngõ vào. Phần chương trình phục vụ công việc này có sẵn trong PLC và được gọi là hệ điều hành.

- ♦ **Thực hiện chương trình:** bộ xử lý sẽ đọc và xử lý tuần tự lệnh một trong chương trình. Trong ghi đọc và xử lý các lệnh, bộ vi xử lý sẽ đọc tín hiệu các đầu vào, thực hiện các phép toán logic và kết quả sau đó sẽ xác định trạng thái của các đầu ra.

- ♦ **Xử lý những yêu cầu truyền thông:** suốt thời gian CPU xử lý thông tin trong chu trình quét. PLC xử lý tất cả thông tin nhận được từ cổng truyền thông hay các module mở rộng.

- ♦ **Thực hiện tự kiểm tra:** trong 1 chu kỳ quét, PLC kiểm tra hoạt động của CPU và trạng thái của modul mở rộng

- ♦ **Xuất tín hiệu ngõ ra:** bộ vi xử lý sẽ gán các trạng thái mới cho các đầu ra tại các module đầu ra.



#### **b. Xử lý xuất nhập**

Gồm hai phương pháp khác nhau dùng cho việc xử lý I/O trong PLC:

##### **❖ Cập nhật liên tục**

Trong phương pháp này, CPU phải mất một khoảng thời gian để đọc trạng thái của các ngõ vào sẽ được xử lý. Khoảng thời gian trên, thường là 3ms, nhằm tránh tác động xung nhiễu gây bởi contact ngõ vào. Các ngõ ra được kích trực tiếp (nếu có) theo sau tác vụ kiểm tra logic. Trạng thái các ngõ ra được chốt trong khối ngõ ra nên trạng thái của chúng được duy trì cho đến lần cập nhật kế tiếp.

### **❖ Lưu ảnh quá trình xuất nhập**

Hầu hết các PLC loại lớn có thể có vài trăm I/O, vì thế CPU chỉ có thể xử lý một lệnh ở một thời điểm. Trong suốt quá trình thực thi, trạng thái mỗi ngõ nhập phải được xét đến riêng lẻ nhằm dò tìm các tác động của nó trong chương trình. Do chúng ta yêu cầu relay 3ms cho mỗi ngõ vào, nên tổng thời gian cho hệ thống lấy mẫu liên tục, gọi là chu kỳ quét hay thời gian quét, trở nên rất dài và tăng theo số ngõ vào.

Để làm tăng tốc độ thực thi chương trình, các ngõ I/O được cập nhật tới một vùng đặc biệt trong chương trình. Ở đây, vùng RAM đặc biệt này được dùng như một bộ đệm lưu trạng thái các logic điều khiển và các đơn vị I/O. Từng ngõ vào và ngõ ra được cấp phát một ô nhớ trong vùng RAM này. Trong khi lưu trạng thái các ngõ vào/ra vào RAM. CPU quét khối ngõ vào và lưu trạng thái chúng vào RAM. Quá trình này xảy ra ở một chu kỳ chương trình.

Khi chương trình được thực hiện, trạng thái của các ngõ vào đã lưu trong RAM được đọc ra. Các tác vụ được thực hiện theo các trạng thái trên và kết quả trạng thái của các ngõ ra được lưu vào RAM ngõ ra. Sau đó vào cuối chu kỳ quét, quá trình cập nhật trạng thái vào/ra chuyển tất cả tín hiệu ngõ ra từ RAM vào khối ngõ ra tương ứng, kích các ngõ ra trên khối vào ra. Khối ngõ ra được chốt nên chúng vẫn duy trì trạng thái cho đến khi chúng được cập nhật ở chu kỳ quét kế tiếp.

Tác vụ cập nhật trạng thái vào/ra trên được tự động thực hiện bởi CPU bằng một đoạn chương trình con được lập trình sẵn bởi nhà sản xuất. Như vậy, chương trình con sẽ được thực hiện tự động vào cuối chu kỳ quét hiện hành và đầu chu kỳ kế tiếp. Do đó, trạng thái của các ngõ vào/ra được cập nhật.

Lưu ý rằng, do chương trình con cập nhật trạng thái được thực hiện tại một thời điểm xác định của chu kỳ quét, trạng thái của các ngõ vào và ngõ ra không thay đổi trong chu kỳ quét hiện hành. Nếu một ngõ vào có trạng thái thay đổi sau sự thực thi chương trình con hệ thống, trạng thái đó sẽ không được nhận biết cho đến quá trình cập nhật kế tiếp xảy ra.

Thời gian cập nhật tất cả các ngõ vào ra phụ thuộc vào tổng số I/O được sử dụng, thường là vài ms. Thời gian thực thi chương trình (chu kỳ quét) phụ thuộc vào độ lớn chương trình điều khiển. Thời gian hi hành một lean cơ bản (một bước) là 0,08  $\mu$ s đến 0.1  $\mu$ s tùy loại PLC, nên chương trình có độ lớn 1K bước (1000



## ***Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH***

---

bước) có chu kỳ quét là 0,8 ms đến 1ms. Tuy nhiên, chương trình điều khiển thường ít hơn 1000 bước, khoảng 500 bước trở lại.

### **4. Ngôn ngữ lập trình.**

Có 5 loại ngôn ngữ dùng để lập trình cho PLC:

#### **a. Ngôn ngữ lập trình ST ( Structure text ) hoặc STL ( Statement List )**

Là một ngôn ngữ lập trình cấp cao gần giống như Pascal, thực hiện các công việc sau:

- Gán giá trị cho các biến
- Gọi hàm và các FunctionBlock
- Tạo và tính toán các biểu thức
- Thực hiện các biểu thức điều kiện

Thí dụ:

```
IF (I:000/00 = 1) THEN  
O:001/00 := 1;  
ELSE  
O:000/01 = 0;  
End_IF;
```

#### **b. Ngôn ngữ lập trình IL ( Instruction List )**

Là ngôn ngữ lập trình cấp thấp, gần giống như ngôn ngữ máy Assembler, thường được dùng để lập trình cho vi xử lý. Cấu trúc của chương trình bao gồm một loạt các câu lệnh, mỗi câu lệnh nằm trên một dòng và được kết thúc bằng ký tự xuống dòng. Mỗi câu lệnh bao gồm một toán tử và nhiều toán hạng. Toán hạng là đối tượng của toán tử và là các biến hoặc các hằng số.

Ngôn ngữ IL phù hợp cho các ứng dụng nhỏ, giải quyết các vấn đề có thứ tự trước sau. Nếu được lập trình tốt, chương trình viết bằng IL sẽ có tốc độ tính toán nhanh nhất.

## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

Thí dụ: Bảng so sánh mã gọi nhớ ( code mnemonics ) của một số hãng

IEC 1131-3 Operators	Mitsubishi	OMRON	Siemens/ Telemecanique	Operation	Ladder diagram
LD	LD	LD	A	Load operand into result register	Start a rung with open contacts
LDN	LDI	LD NOT	AN	Load negative operand into result register	Start a rung with closed contacts
AND	AND	AND	A	Boolean AND	A series element with open contacts
ANDN	ANI	AND NOT	AN	Boolean AND with negative operand	A series element with closed contacts
OR	OR	OR	O	Boolean OR	A parallel element with open contacts
ORN	ORI	OR NOT	ON	Boolean OR with negative operand	A parallel element with closed contacts
ST	OUT	OUT	=	Store result register into operand	An output

Của Mitsubishi:

```
LD    X400
AND   X401
OUT   Y430
```

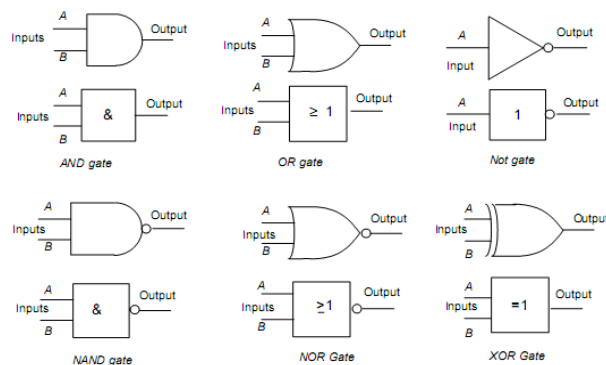
Của Siemens:

```
A    I0.1
A    I0.2
=    Q2.0
```

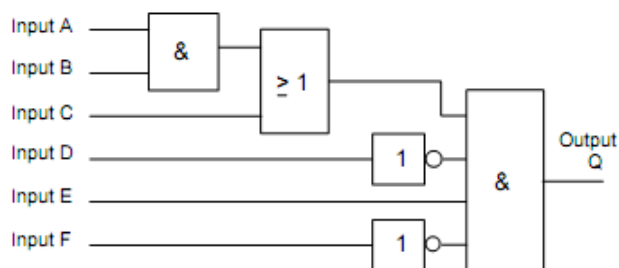
### c. Ngôn ngữ lập trình FBD ( Function Block Diagrams )

Là ngôn ngữ lập trình theo kiểu đồ họa, bằng cách mô tả quá trình dưới các dòng chảy tín hiệu giữa các khối hàm với nhau. Nó giống như việc đi dây trong các mạch điện tử.

Thí dụ: Ký hiệu các công Logic:



Một chương trình hoạt động:



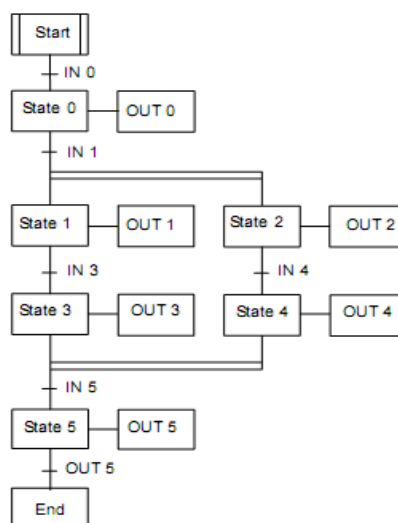
d. Ngôn ngữ lập trình SFC ( Sequence Function Charts )

Là ngôn ngữ lập trình theo kiểu tuần tự, chương trình SFC bao gồm một chuỗi các bước được thể hiện dưới dạng các hình chữ nhật và được nối với nhau.

Mỗi bước đại diện cho một trạng thái cụ thể cần được điều khiển của hệ thống. Mỗi bước có thể thực hiện một hoặc nhiều công việc đồng thời.

Mỗi một mối nối có một hình chữ nhật ở giữa, đại diện cho điều kiện chuyển đổi giữa các trạng thái trong hệ thống. Khi điều kiện chuyển đổi đạt được “ True “ thì cho phép chuyển sang trạng thái tiếp theo.

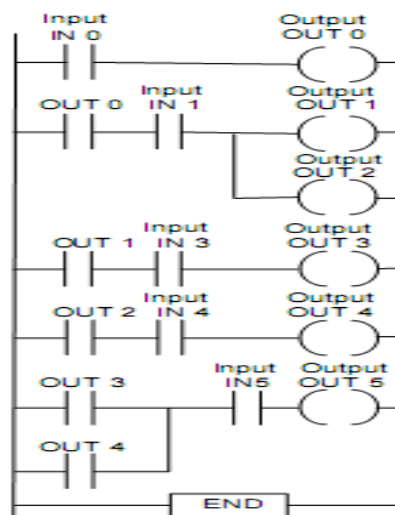
Thí dụ :



e. Ngôn ngữ lập trình LD ( Ladder Diagram )

Còn gọi là ngôn ngữ bậc thang là một kiểu ngôn ngữ lập trình đồ họa. Lập trình theo LD gần giống như khi các kỹ sư điện thiết kế và đi dây các bảng mạch điện điều khiển logic: Rơ-le, công-tắc-tơ, khởi động từ . . .

Thí dụ: Mạch điện tương đương của mạch SFC trên được viết dưới dạng LD:



## II. Giới thiệu một số PLC của hãng MITSUBISHI ELECTRIC

Do nhu cầu sử dụng ngày càng cao PLC trong công nghiệp nên nhà sản xuất đã nghiên cứu chế tạo nhiều họ PLC đáp ứng cho nhu cầu nhiều nhiệm vụ điều khiển với các dạng và qui mô khác nhau. Các PLC được chế tạo được chế tạo dựa trên nhiều đặc trưng như nguồn cấp điện, dạng điện áp ngõ vào, dạng ngõ ra, bộ xử lý, ngôn ngữ lập trình, tập lệnh khả năng xử lý số lệnh, khả năng xử lý tốc độ cao, khả năng mở rộng với module vào/ra và module chức năng chuyên dùng, khả năng nối mạng.

Đặc điểm kỹ thuật	FX1S	FX1N	FX2N	FX3U
Phương pháp xử lý chương trình.	Thực hiện quét chương trình tuần hoàn.			
Phương pháp xử lý vào ra	Cập nhật ở đầu và cuối chu kỳ quét (khi lệnh END được thi hành)			
Thời gian xử lý	Cơ bản: 0,72 $\mu$ s Ứng dụng: 10 $\rightarrow$ 100 $\mu$ s		Cơ bản: 0,08 $\mu$ s Ứng dụng 1,52 $\rightarrow$ 100 $\mu$ s	Cơ bản: 0,065 $\mu$ s Ứng dụng 0,642 $\rightarrow$ 100 $\mu$ s
Ngôn ngữ lập trình	Ngôn ngữ Ladder + Instruction + SFC			
Dung lượng chương trình	2k Steps	8k Steps	8k Steps (16k Steps gắn thêm bộ nhớ ngoài)	8k Steps (64k Steps gắn thêm bộ nhớ ngoài)
Cấu hình vào/ra có thể	30 I/O Max input 16	128 I/O	256 I/O	384 I/O

**Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH**

		Max output 14			
Relay phụ trợ (M)	Chung	M0 ÷ M383	M0 ÷ M383	M0 ÷ M3071	M0 ÷ M7679
	Được chốt	M384 ÷ M511	M384 ÷ M1535	M500 ÷ M3071	M500 ÷ M7679
	Chuyên dùng	M8000 ÷ M8255			M8000 ÷ M8511
Relay trạng thái (S)	Chung	S0 ÷ S127	S0 ÷ S999	S0 ÷ S999	S0 ÷ S4095
	Được chốt	N/A	N/A	S500 ÷ S999	S500 ÷ S4095
	Khởi tạo	S0 ÷ S9	S0 ÷ S9	S0 ÷ S9	S0 ÷ S9
	Cờ hiệu	N/A	N/A	S900 ÷ S999	S900 ÷ S999
Bộ định thì (T)	100 ms	T0 ÷ T55	T0 ÷ T199		
	10 ms	T32 ÷ T62 (M8028 = ON)	T200 ÷ T245		
	1 ms	T63	N/A		T256 ÷ T511
	1ms (được chốt)	N/A	T246 ÷ T249		
	100 ms (được chốt)	N/A	T250 ÷ T255		
Bộ đếm ( C )	Chung (U) 16 bit	C0 ÷ C31	C0 ÷ C199		
	Được chốt (U) 16bit	C16 ÷ C31	C16 ÷ C199	C100 ÷ C199	
	Chung (U/D) 32 bit	N/A	C200 ÷ C234		
	Được chốt (U/D) 32 bit	N/A	C220 ÷ C234		
Bộ đếm tốc độ cao	1 pha (U/D) 32 bit	C235 ÷ C240	C235 ÷ C240		

**Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH**

(HSC)	1 pha khởi động và Reset được gán trước (U/D) 32 bit	C241, C242, C244	C242 ÷ C245		
	2 pha (U/D) 32 bit	C246, C247, C249	C246÷ C250		
	Pha A/B 32 bit	C251, C252, C254	C251 ÷ C255		
Thanh ghi dữ liệu 16 bit (D)	Chung	D0 ÷ D255	D0 ÷D127 và D1000 ÷ D7999	D0 ÷ D7999	
	Được chốt	D128 ÷ D255	D125 ÷ D999	D200 ÷ D7999	
	Thanh ghi tập tin		D1000 ÷ D6999	D1000 ÷ D7999	
	Được điều chỉnh bên ngoài	D8013 hay D8030 và D8031	D8030, D8031		
	Đặc biệt	D8000 ÷ D8255			D8000 ÷ D8511
	Chỉ mục	V, Z	V, Z	V0 ÷ V7 Z0 ÷ Z7	
Con trỏ P và I	Dùng với lệnh CALL/CJ	P0 ÷ P63	P0 ÷ P127		P0 ÷ P4095
	Dùng với ngắt	I00□ ÷ I30□ Cạnh lên: □=1 Cạnh xuống:□=0	I00□ ÷ I30□ Cạnh lên: □=1 Cạnh xuống: □=0	I00□ ÷ I50□ và I6☆☆ I8☆☆ Cạnh lên: □=1 Cạnh xuống: □=0 ☆☆ = thời gian tính bằng ms	
Số mức lồng		8 mức khi dùng với lệnh MC và MCR (N0 ÷ N7)			
Hằng số	Thập	16 bit: -32.768 ÷ +32.767			

## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

	phần K	32 bit: -2.147.483.648 ÷ +2.147.483.647	
	Thập lục phân H	16 bit: 0000 ÷ FFFF 32 bit: 00000000 ÷ FFFFFFFF	
	Dấu chấm động	N/A	32 bit: 0, ±1.175x10 <sup>-38</sup> ÷ 0, ±3.403x10 <sup>+38</sup>
	Số thực R	N/A	32 bit

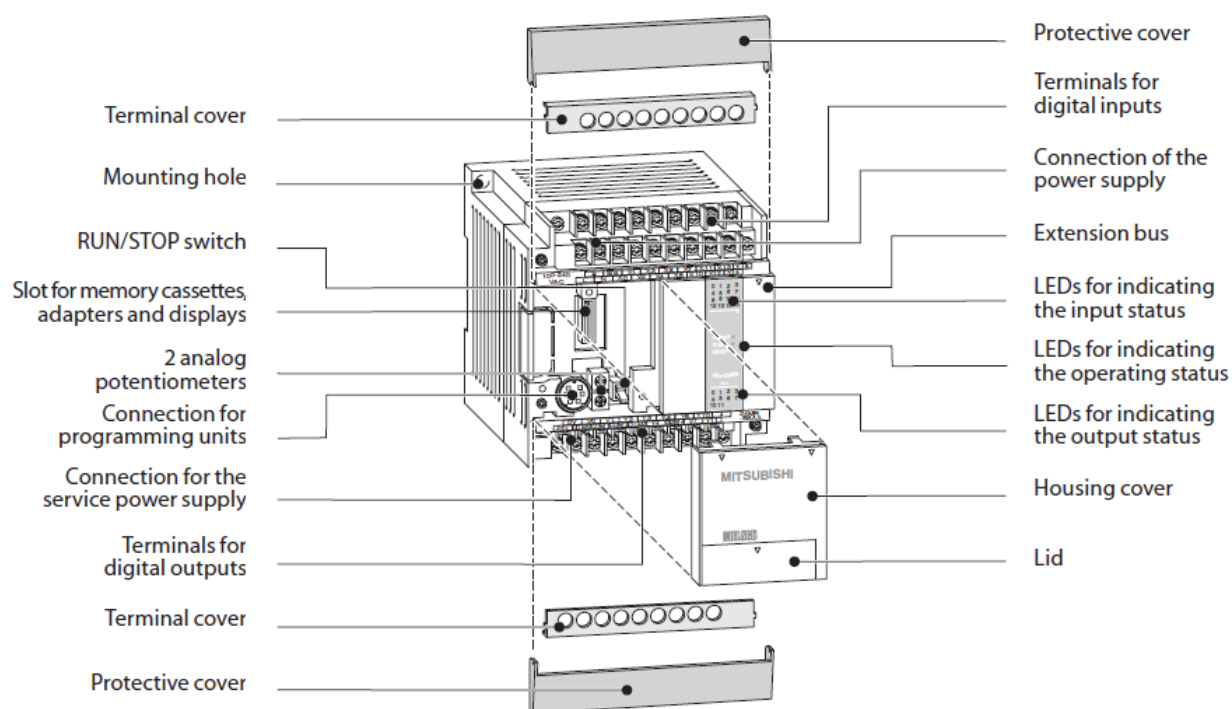
MELSEC FX có nhiều loại phiên bản khác nhau tùy thuộc vào bộ nguồn hay công nghệ của ngõ ra. Ta có thể lựa chọn bộ nguồn cung cấp 100 – 220 V AC, 24 V DC hay 12 – 24 V DC, ngõ ra là relay hoặc transistor.

Series	I/O	Loại	Số ngõ vào	Số ngõ ra	Nguồn	Loại ngõ ra
FX1S	10	FX1S-10M□-□□	6	8	24 VDC hay 100-240 VAC	Transistor hoặc relay
	14	FX1S-14M□-□□	8	6		
	20	FX1S-20M□-□□	12	8		
	30	FX1S-30M□-□□	16	14		
FX1N	14	FX1N-14M□-□□	8	6	12-24V DC hay 100-240 VAC	Transistor hoặc relay
	24	FX1N-24M□-□□	14	10		
	40	FX1N-40M□-□□	24	16		
	60	FX1N-60M□-□□	36	24		
FX2N	16	FX2N-16M□-□□	8	8	24 VDC hay 100-240 V AC	Transistor hoặc relay
	32	FX2N-32M□-□□	16	16		
	48	FX2N-48M□-□□	24	24		
	64	FX2N-64M□-□□	32	32		
	80	FX2N-80M□-□□	40	40		
	128	FX2N-128M□-□□	64	64		
FX2NC	16	FX2NC-16M□-□□	8	8	24 VDC	Transistor hoặc relay
	32	FX2NC-32M□-□□	16	16		
	64	FX2NC-64M□-□□	32	32		

## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

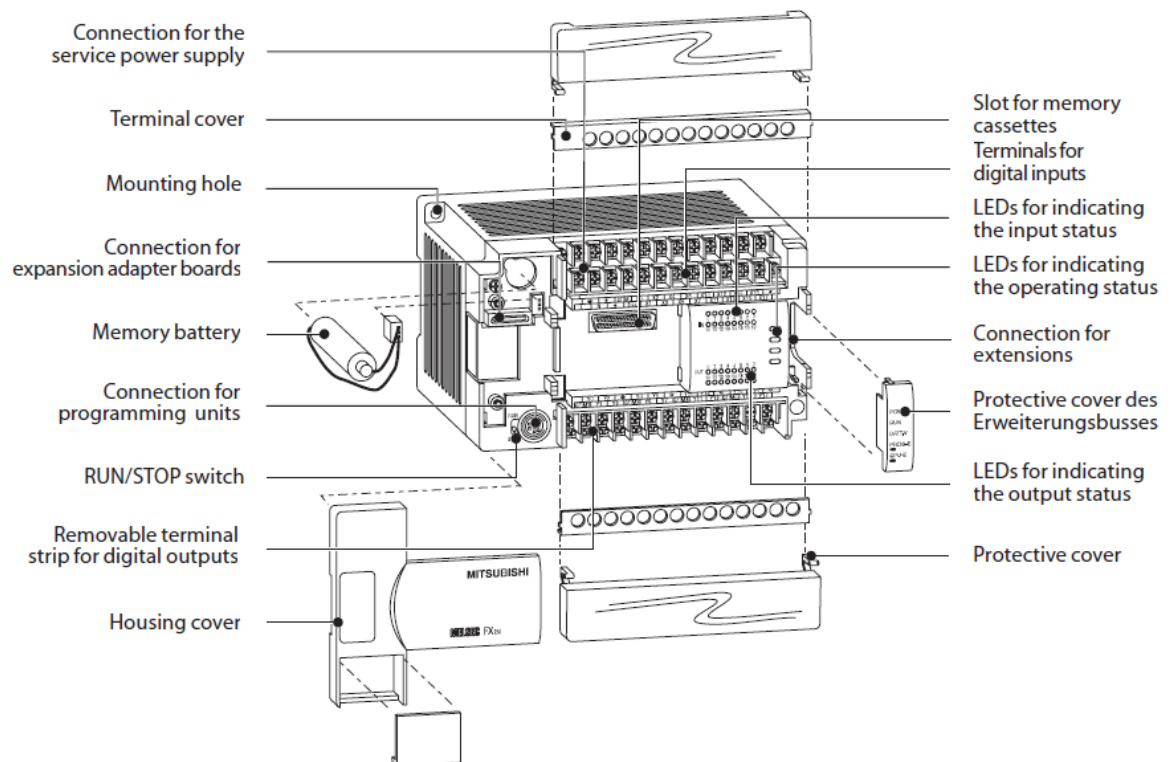
	96	FX2NC-96M□-□□	48	48		
FX3U	16	FX3U-16M□-□□	8	8	24 V DC hay 100- 240 V AC	Transistor hoặc relay
	32	FX3U -32M□-□□	16	16		
	48	FX3U -48M□-□□	24	24		
	64	FX3U -64M□-□□	32	32		
	80	FX3U -80M□-□□	40	40		
	128	FX3U -128M□-□□	64	64	100-240 VAC	Transistor hoặc relay

- Bố trí của FX1N

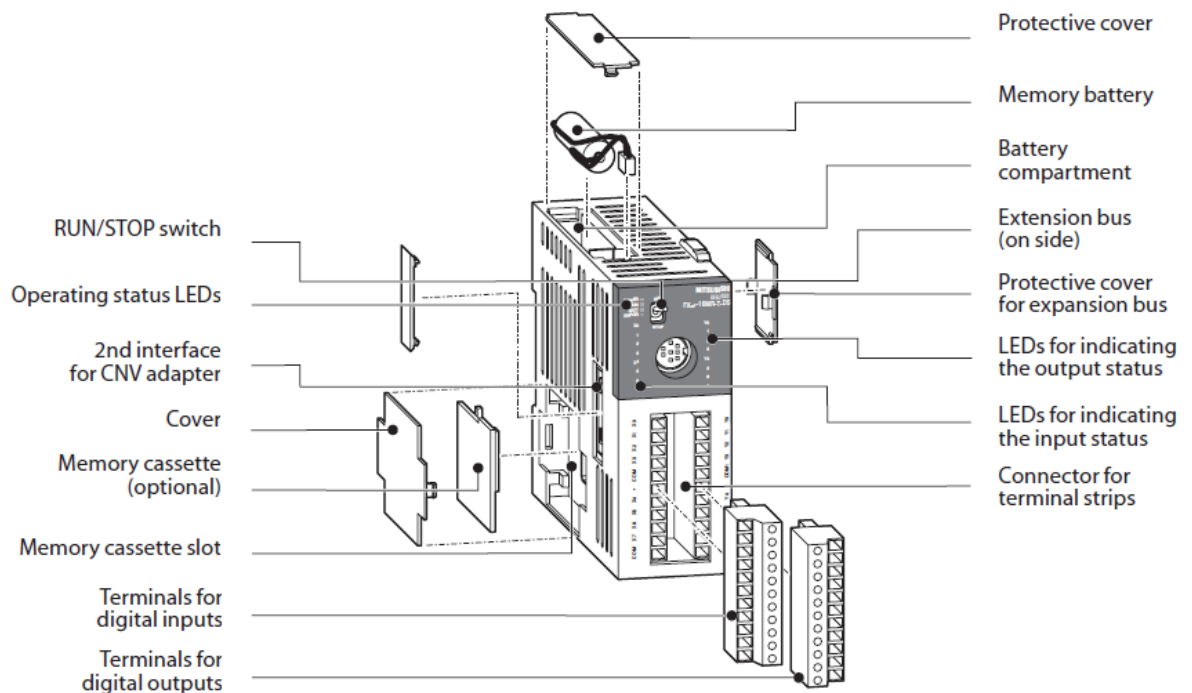


- Bố trí của FX2N

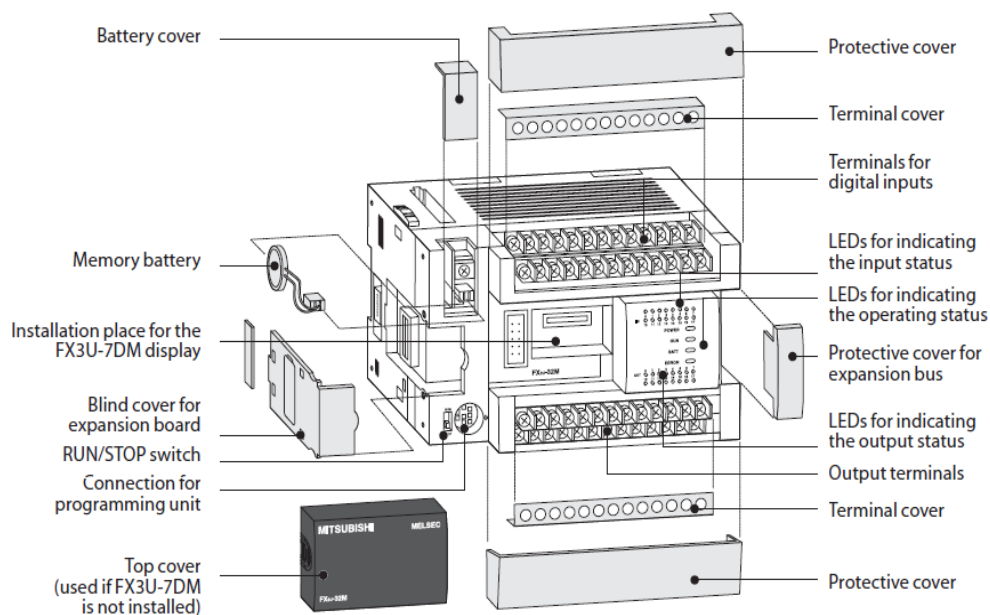




- Bố trí của FX2NC



- Bố trí của FX3U



### III. Kết nối PLC với thiết bị ngoại vi.

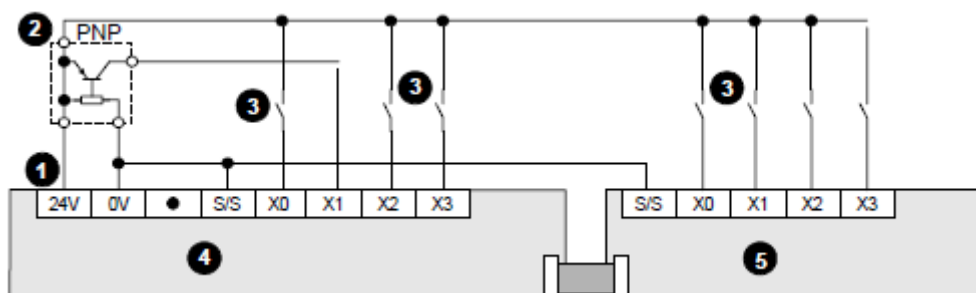
Khối vào ra là mạch giao tiếp giữa mạch vi điện tử của PLC với các mạch công suất bên ngoài kích hoạt các cơ cấu tác động: thực hiện sự chuyển đổi các mức điện áp tín hiệu và cách ly. Tuy nhiên khối vào/ra cho phép PLC kết nối trực tiếp với các cơ cấu tác động có công suất nhỏ, không 2A trở xuống, không cần các mạch trung gian hay relay trung gian.

Tất cả các ngõ vào/ra đều được cách ly với các tín hiệu điều khiển bên ngoài bằng mạch cách ly quang (opto-isolator) trên khối vào ra. Mạch cách ly quang dùng một diode phát quang và một transistor quang gọi là bộ opto-coupler. Mạch này cho phép các tín hiệu nhỏ đi qua và ghim các tín hiệu điện áp cao xuống mức tín hiệu chuẩn. Mạch này có tác dụng chống nhiễu khi chuyển contact và bảo vệ quá áp từ nguồn cấp điện thường lên đến 1500V.

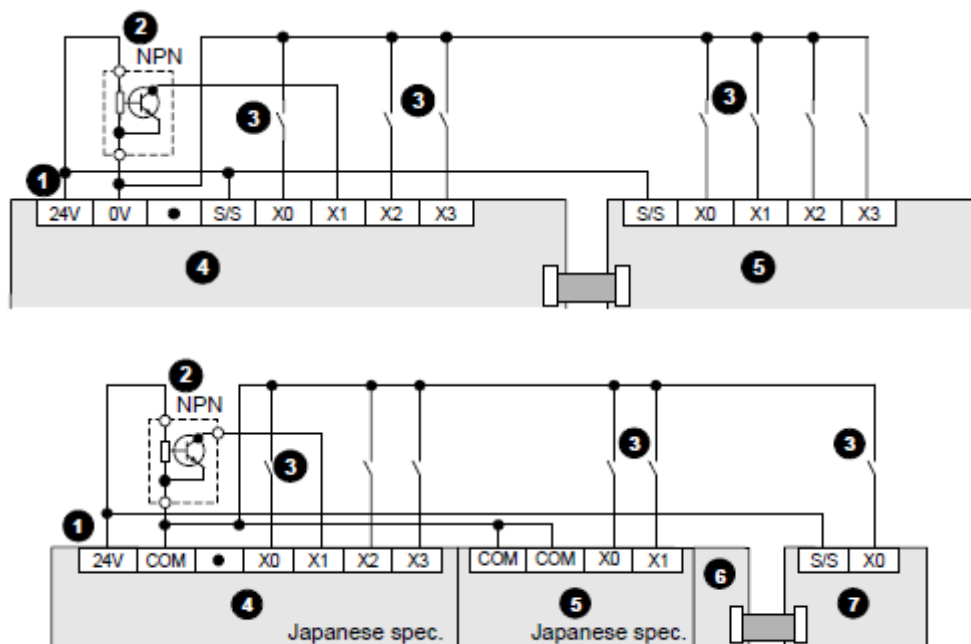
#### 1. Kết nối ngõ vào.

##### a. Ngõ vào V DC.

##### SOURCE

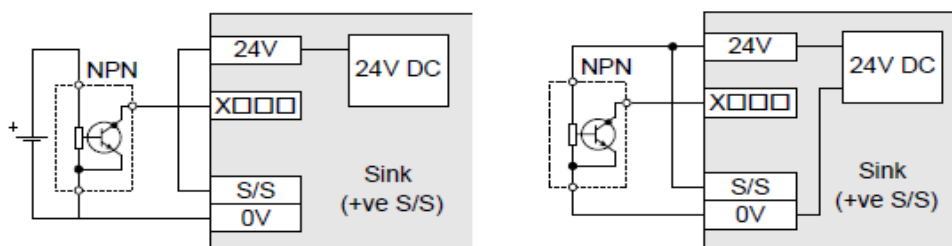


**SINK**



1. Nguồn cung cấp 24VDC
2. Cảm biến PNP (NPN).
3. Contact.
4. MPU (main processing unit).
5. Khối mở rộng.
6. Đầu nối bus mở rộng (trên PLC).
7. Đầu nối bus mở rộng (trên module mở rộng).

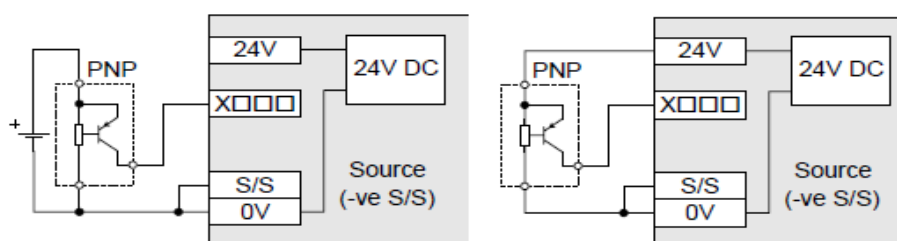
• **Kết nối ngõ vào kiểu transistor NPN.**



Sử dụng nguồn 24VDC của PLC.

Sử dụng nguồn 24 VDC ngoài

• **Kết nối ngõ vào kiểu transistor PNP**



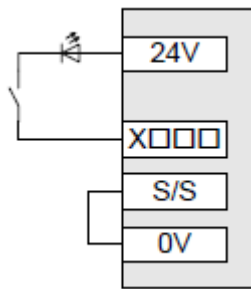
Sử dụng nguồn 24VDC của PLC.

Sử dụng nguồn 24 VDC ngoài.

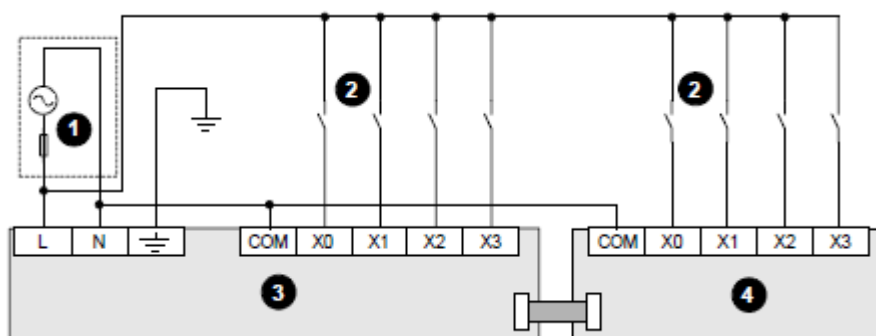
## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

- **Kết nối với diode.**

Không nối hơn 2 LED nối tiếp. Điện áp rơi trên diod tối đa 4V.

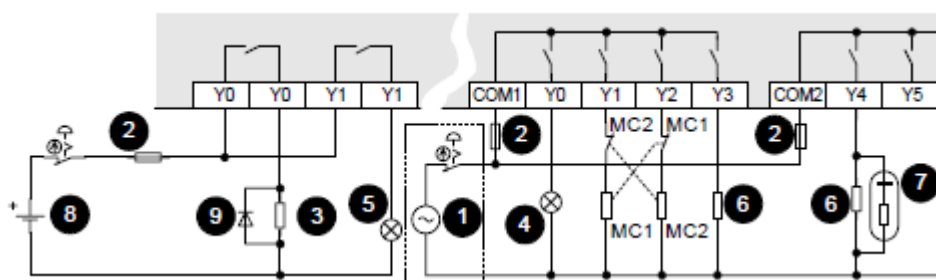


- b. Ngõ vào V AC.**



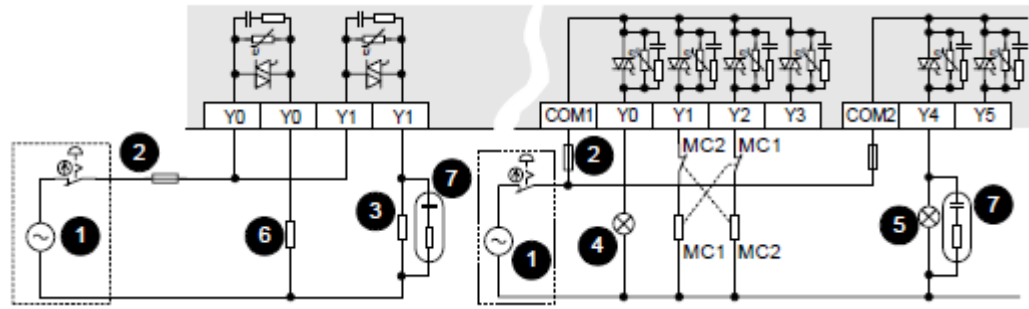
1. Nguồn cung cấp xoay chiều.
2. Contact.
3. MPU (main processing unit).
4. Khởi mở rộng.

## 2. Kết nối ngõ ra.



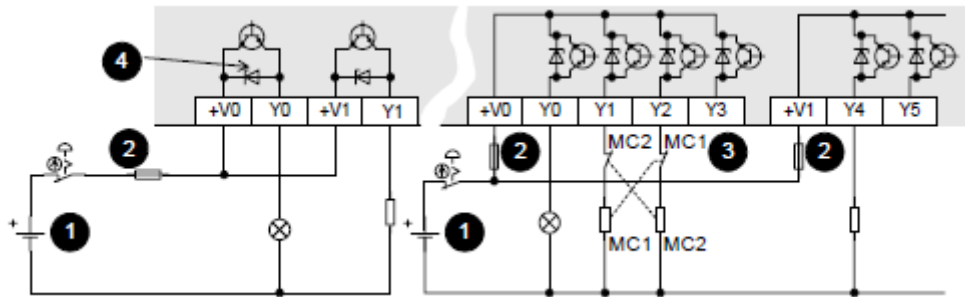
Ngõ ra dùng relay (dùng điện áp xoay chiều – đáp ứng chậm)

- |                      |                    |
|----------------------|--------------------|
| 1. Nguồn xoay chiều. | 6. Contactor.      |
| 2. Cầu chì.          | 7. Bộ chống nhiễu. |
| 3. Van solenoid.     | 8. Nguồn 1 chiều.  |
| 4. Đèn sợi đốt       | 9. Diode ghim áp.  |
| 5. Đèn neon.         |                    |



Loại ngõ ra dùng triac (dùng điện áp xoay chiều – đáp ứng nhanh)

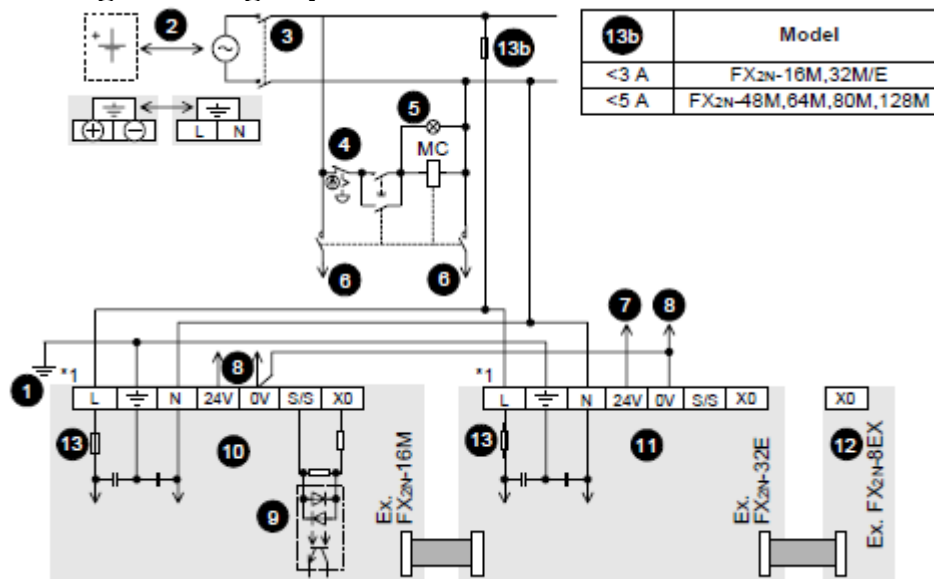
- |                      |                    |
|----------------------|--------------------|
| 1. Nguồn xoay chiều. | 5. Đèn Neon.       |
| 2. Cầu chì.          | 6. Contactor.      |
| 3. Van solenoid.     | 7. Bộ chống nhiễu. |
| 4. Đèn sợi đốt.      |                    |



Loại ngõ ra dùng transistor (dùng điện áp 1 chiều – đáp ứng nhanh)

- |                                   |
|-----------------------------------|
| 1. Nguồn 1 chiều.                 |
| 2. Cầu chì.                       |
| 3. Khóa lẩn cơ khí bên ngoài.     |
| 4. Diode zener bảo vệ transistor. |

### 3. Nguồn cung cấp.

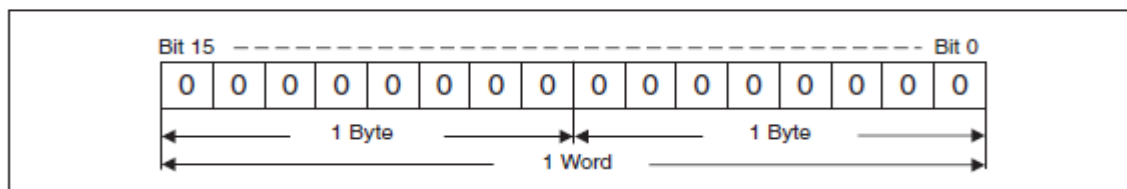


1. Nối đất (100Ω hoặc nhỏ hơn)
2. Nguồn cung cấp.
3. Thiết bị bảo vệ mạch.
4. Nút dừng khẩn cấp
5. Đèn báo
6. Nguồn cấp cho tải
7. Không nối đầu nối “24V” giữa CPU với phần mở rộng.
8. Cung cấp dịch vụ
9. Photocoupler
10. MPU-main processing unit
11. Đơn vị mở rộng
12. Khối mở rộng.
13. Cầu chì

## IV. Hệ thống số.

### 1. Bits, bytes , words

Như mọi thiết bị trong công nghệ số, đơn vị thông tin nhỏ nhất ở một PLC là " bit ". Một bit chỉ có hai trạng thái " 0 " (OFF hay FALSE) và " 1 " ( ON hay TRUE). Các bit có thể nhóm lại với nhau để biểu diễn các dữ liệu lớn hơn. Ví dụ 8 bit liên tiếp tạo thành một byte, 16 bit tạo thành một word và 32 bit tạo thành một double word.



## 2. Hệ thập phân

Trong các hệ thống số thì hệ thập phân gần gũi với chúng ta nhất vì nó được ta sử dụng hằng ngày. Khi hiểu các đặc điểm của nó sẽ giúp chúng ta dễ hiểu hơn những hệ thống số khác.

Hệ thập phân – hay còn gọi là hệ cơ số 10. Bao gồm 10 chữ số (ký hiệu) đó là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Sử dụng những chữ này ta có thể biểu thị được đại lượng bất kỳ.

Hệ thập phân là một hệ thống theo vị trí vì trong đó giá trị của một chữ số phụ thuộc vào vị trí của nó. Để hiểu rõ điều này ta xét ví dụ sau: xét số thập phân 345. Ta biết rằng chữ số 3 biểu thị 3 trăm, 4 biểu thị 4 chục, 5 là 5 đơn vị. Xét về bản chất, 3 mang giá trị lớn nhất trong ba chữ số, được gọi là *chữ số có nghĩa lớn nhất* (MSD). Chữ số 5 mang giá trị nhỏ nhất, gọi là *chữ số có nghĩa nhỏ nhất* (LSD).

Để diễn tả một số thập phân lẻ người ta dùng dấu chấm thập phân để chia phần nguyên và phần phân số.

Ý nghĩa của một số thập phân được mô tả như sau:

$\times 10^3$	$\times 10^2$	$\times 10^1$	$\times 10^0$		$\times 10^{-1}$	$\times 10^{-2}$	$\times 10^{-3}$
= 1000	= 100	= 10	= 1	.	= 0.1	= 0.01	= 0.001
Số có nghĩa lớn nhất ( MSD )				Dấu chấm thập phân			Số có nghĩa nhỏ nhất ( LSD )

Ví dụ 1: Số 435.568

$$435.568 = 4 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2} + 8 \times 10^{-3}$$

Tóm lại, một số thập phân; nhị phân hay thập lục đều là tổng của các tích giữa các giá trị của mỗi chữ số với giá trị vị trí (còn gọi là trọng số) của nó.

$\times 10^2$	$\times 10^1$	$\times 10^0$		$\times 10^{-1}$	$\times 10^{-2}$	$\times 10^{-3}$
4	3	5	.	5	6	8
MSD			Dấu chấm thập phân			LSD

## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

Với N chữ số ta có thể đếm qua  $10^N$  số khác nhau, từ 0 đến  $10^N - 1$ .

### 3. Hệ nhị phân

Trong hệ thống nhị phân (binary system) chỉ có hai giá trị số là 0 và 1. Nhưng có thể biểu diễn bất kỳ đại lượng nào mà hệ thập phân và hệ các hệ thống số khác có thể biểu diễn được, tuy nhiên phải dùng nhiều số nhị phân để biểu diễn đại lượng nhất định.

Tất cả các phát biểu về hệ thập phân đều có thể áp dụng được cho hệ nhị phân. Hệ nhị phân cũng là hệ thống số theo vị trí. Mỗi nhị phân đều có giá trị riêng, tức trọng số, là lũy thừa của 2. Để biểu diễn một số nhị phân lẻ ta cũng dùng dấu chấm thập phân để phân cách phần nguyên và phần lẻ.

Ý nghĩa của một số nhị phân được mô tả như sau:

$2^3$	$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	$2^{-3}$	← Giá trị vị trí
= 8	= 4	= 2	= 1	.	= 1/2	= 1/4	= 1/8	
1	1	0	0	.	1	0	1	← Số nhị phân
MSB				Dấu chấm thập phân			LSB	

Để tìm giá trị thập phân tương đương ta chỉ việc tính tổng các tích giữa mỗi số (0 hay 1) với giá trị vị trí của nó.

Ví dụ 2:

$$\begin{aligned} 1100.101_2 &= (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) \\ &= 8 + 4 + 0 + 0 + 0.5 + 0 + 0.125 \\ &= 12.175 \end{aligned}$$

$2^3$	$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	$2^{-3}$	← Giá trị vị trí
= 8	= 4	= 2	= 1	.	= 1/2	= 1/4	= 1/8	
1	1	0	0	.	1	0	1	← Số nhị phân
MSB				Dấu chấm thập phân			LSB	

### CÁCH GỌI NHỊ PHÂN



## ***Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH***

---

Một con số trong số nhị phân được gọi 1 bit (Binary Digital). Bit đầu (hàng tận cùng bên trái) có giá trị cao nhất được gọi là MSB (Most Significant Bit – bit có nghĩa lớn nhất), bit cuối (hàng tận cùng bên phải) có giá trị nhỏ nhất và được gọi LSB (Least Significant Bit – bit có nghĩa nhỏ nhất). Với số thập phân phải nói MSD và LSD.

Số nhị phân có 8 bit được gọi là 1 byte, số nhị phân có 4 bit gọi là nibble. Một nhóm các bit nhị phân nói chung được gọi một word (từ) nhưng thường dùng để chỉ số có 16 bit, số 32 bit gọi là doubleword, 64 bit gọi là quadword.

Để thuận tiện cho việc chuyển đổi số ta cần phải biết một số lũy thừa nguyên của. Lũy thừa của  $2^{10} = 1024$  được gọi tắt là 1K (đọc K hay kilo), trong ngôn ngữ nhị phân 1k là 1024 chứ không phải là 1000. Những giá trị lớn hơn tiếp theo như:

$$2^{11} = 2^1 \cdot 2^{10} = 2K$$

$$2^{12} = 2^2 \cdot 2^{10} = 4K$$

$$2^{20} = 2^{10} \cdot 2^{10} = 1K \cdot 1K = 1M \text{ (Mega)}$$

$$2^{24} = 2^4 \cdot 2^{20} = 4 \cdot 1M = 4M$$

$$2^{30} = 2^{10} \cdot 2^{20} = 1K \cdot 1M = 1G \text{ (Giga)}$$

$$2^{32} = 2^2 \cdot 2^{30} = 4 \cdot 1G = 4G$$

Bảng trị giá của  $2^n$

n	$2^n$	Viết tắt
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
9	512	
10	1024	1K
11	2048	2K
12	4096	4K
13	8192	8K
14	16384	16K
15	32768	32K
16	65536	64K
20	1048576	1M
24	16777216	16M
30	1073741824	1G
32	4294967296	4G

### CÁCH ĐẾM NHỊ PHÂN

Cách đếm một số nhị phân được trình bày theo bảng sau

Trọng số	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	Thập phân tương đương
	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
	1	0	0	1	9
	1	0	1	0	10
	1	0	1	1	11
	1	1	0	0	12
	1	1	0	1	13
	1	1	1	0	14
	1	1	1	1	15

## Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH

Nếu sử dụng N bit hoặc N chữ số thì ta có thể đếm được  $2^N$  số độc lập nhau

Ví dụ 3:

2 bit ta đếm được  $2^2 = 4$  số (  $00_2$  đến  $11_2$  )

4 bit ta đếm được  $2^4 = 16$  số (  $0000_2$  đến  $1111_2$  )

Ở bước đếm cuối cùng, tất cả các bit đều ở trạng thái 1 và bằng  $2^N - 1$  trong hệ thập phân.

Ví dụ: sử dụng 4 bit, bước đếm cuối cùng là  $1111_2 = 2^4 - 1 = 15_{10}$

### 4. Hệ bát phân

Hệ bát phân có cơ số 8 nghĩa là có 8 ký số : 0, 1, 2, 3, 4, 5, 6, 7, mỗi ký số của số bát phân có giá trị bất kỳ từ 0 đến 7. Mỗi vị trí ký số của hệ bát phân có trọng số như sau:

$8^3$	$8^2$	$8^1$	$8^0$		$8^{-1}$	$8^{-2}$	$8^{-3}$
= 512	= 64	= 8	= 1	.	= 1/8	= 1/64	= 1/512
MSD				Dấu chấm bát phân			LSD

### 5. Hệ thập lục phân.

Hệ thống số thập lục phân sử dụng cơ số 16, nghĩa là có 16 ký số. Hệ thập lục phân dùng các ký số từ 0 đến 9 cộng thêm 6 chữ A, B, C, D, E, F. Mỗi một ký số thập lục phân biểu diễn một nhóm 4 ký số nhị phân.

Ý nghĩa của hệ thống số thập lục phân được mô tả bằng bảng sau:

$16^3$	$16^2$	$16^1$	$16^0$		$16^{-1}$	$16^{-2}$	$16^{-3}$
= 4096	= 256	= 16	= 1	.	= 1/16	= 1/256	= 1/4096
MSD				Dấu chấm thập lục phân			LSD

Mối quan hệ giữa các hệ thống thập lục phân, thập phân, bát phân và nhị phân được trình bày bằng bảng sau:

Decimal notation	Octal notation	Hexadecimal notation	Binary notation
0	0	0	0000 0000 0000 0000
1	1	1	0000 0000 0000 0001
2	2	2	0000 0000 0000 0010
3	3	3	0000 0000 0000 0011
4	4	4	0000 0000 0000 0100
5	5	5	0000 0000 0000 0101
6	6	6	0000 0000 0000 0110
7	7	7	0000 0000 0000 0111
8	10	8	0000 0000 0000 1000
9	11	9	0000 0000 0000 1001
10	12	A	0000 0000 0000 1010
11	13	B	0000 0000 0000 1011
12	14	C	0000 0000 0000 1100
13	15	D	0000 0000 0000 1101
14	16	E	0000 0000 0000 1110
15	17	F	0000 0000 0000 1111
16	20	10	0000 0000 0001 0000
:	:	:	:
99	143	63	0000 0000 0110 0011
:	:	:	:

**CÁCH ĐẾM SỐ THẬP LỤC PHÂN:** khi đếm số thập lục phân, mỗi vị trí được tăng dần 1 đơn vị từ 0 cho đến F. khi đếm đến giá trị F, vòng đếm lại trở về 0 và vị trí ký số kế tiếp tăng lên 1. Trình tự đếm được minh họa như dưới đây: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13,..., 1A, 1B,..., 20, 21,..., 26, 27, 28, 29, 2A, 2B, 2D, 2E, 2F,..., 40, 41, 42 ....., 6F8, 6F9, 6FA, 6FB, 6FC, 6FD, 6FE, 6FF, 700,....

## 6. Mã BCD.

Trực tiếp liên quan đến mạch số (bao gồm các hệ thống sử dụng số) là các số nhị phân nên mọi thông tin dữ liệu dù là số lượng, các chữ, các dấu, các mệnh lệnh sau cùng cũng phải ở dạng nhị phân thì mạch số mới hiểu ra và xử lý được. Do đó phải có quy định cách thức mà các số nhị phân được dùng để biểu thị các dữ liệu khác nhau, kết quả là có nhiều mã số (gọi tắt là mã) được dùng. Trước tiên mã số thập phân thông dụng nhất là mã BCD ( Binary Coded Decimal: mã số thập phân được mã hóa theo nhị phân ). Sự chuyển đổi thập phân sang BCD và ngược lại gọi là mã hoá và sự lập mã.

Người ta biểu thị các số thập phân từ 0 đến 9 bởi số nhị phân 4 bit có giá trị như bảng dưới đây.

## **Chương 1: BỘ ĐIỀU KHIỂN LẬP TRÌNH**

Chúng ta nên chú ý rằng: mã BCD phải được viết đủ 4 bit và sự tương ứng chỉ được áp dụng cho số thập phân từ 0 đến 9, nên số nhị phân từ 1010 ( $= 10_{10}$ ) đến 1111 ( $= 15_{10}$ ) của số nhị phân 4 bit không phải là mã BCD.

Thập phân	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Khi chuyển đổi qua lại giữa thập phân và BCD ta làm như ví dụ minh họa sau đây:

Ví dụ 1: Đổi  $489_{10}$  sang mã BCD

4	8	9	( thập phân )
↓	↓	↓	
0100	1000	1001	( BCD )

Ví dụ 2: Đổi  $537_{10}$  sang mã BCD

<b>5</b>	<b>3</b>	<b>7</b>	<b>( thập phân )</b>
⇓	⇓	⇓	
<b>0101</b>	<b>0011</b>	<b>0111</b>	<b>( BCD )</b>

Ví dụ 23: Đổi  $0011010010010101_2$  (BCD) sang số thập phân

<b>0011</b>	<b>0100</b>	<b>1001</b>	<b>0101</b>	<b>( BCD )</b>
⇓	⇓	⇓	⇓	
<b>3</b>	<b>4</b>	<b>9</b>	<b>5</b>	<b>(Thập phân)</b>

## **7. So sánh BCD và số nhị phân.**

Điều quan trọng là phải nhận ra rằng BCD không phải là hệ thống số như hệ thống số thập phân, nhị phân, bát phân và thập lục phân. Thật ra, BCD là hệ thập phân với từng ký số được mã hóa thành giá trị nhị phân tương đương. Cũng phải hiểu rằng một số BCD không phải là số nhị phân quy ước. Mã nhị phân quy ước biểu diễn số thập phân hoàn chỉnh ở dạng nhị phân; Còn mã BCD chỉ chuyển đổi từng ký số thập phân sang số nhị phân tương ứng.

Mã BCD cần nhiều bit hơn để biểu diễn các số thập phân nhiều ký số (2 ký số trở lên). Điều này là do mã BCD không sử dụng tất cả các nhóm 4 bit có thể có, vì vậy có phần kém hiệu quả hơn.

Ưu điểm của mã BCD là dễ dàng chuyển đổi từ thập phân sang nhị phân và ngược lại. Chỉ cần nhớ các nhóm mã 4 bit ứng với các ký số thập phân từ 0 đến 9.

### **Phối hợp các hệ thống số**

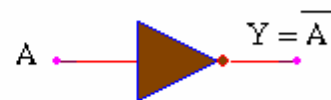
Các hệ thống số đã trình bày có mối tương quan như bảng sau đây:

Thập phân	Nhị phân	Bát phân	Thập lục phân	BCD
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101

## V. Các phép toán logic

### 1. Đảo - NOT

Biến (Ngõ vào)	Hàm (Ngõ ra)
A	Y
0	1
1	0



### 2. Và – AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



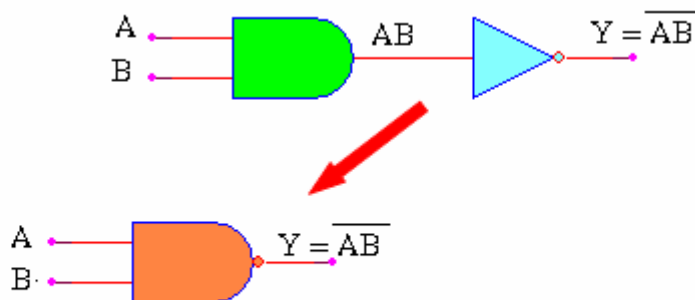
### 3. Hoặc – OR

Biến (ngõ vào)		Hàm (ngõ ra)
A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1



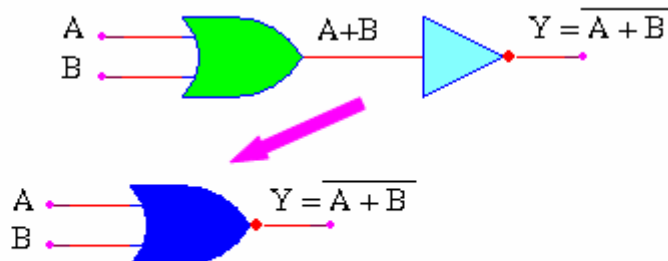
### 4. NAND

Vào		Ra
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



### 5. NOR

Vào		Ra
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



### 6. EXCLUSIVE (EX – OR)

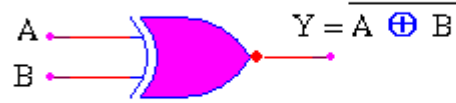
Biến (Ngõ vào)		Hàm (Ngõ ra)
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0





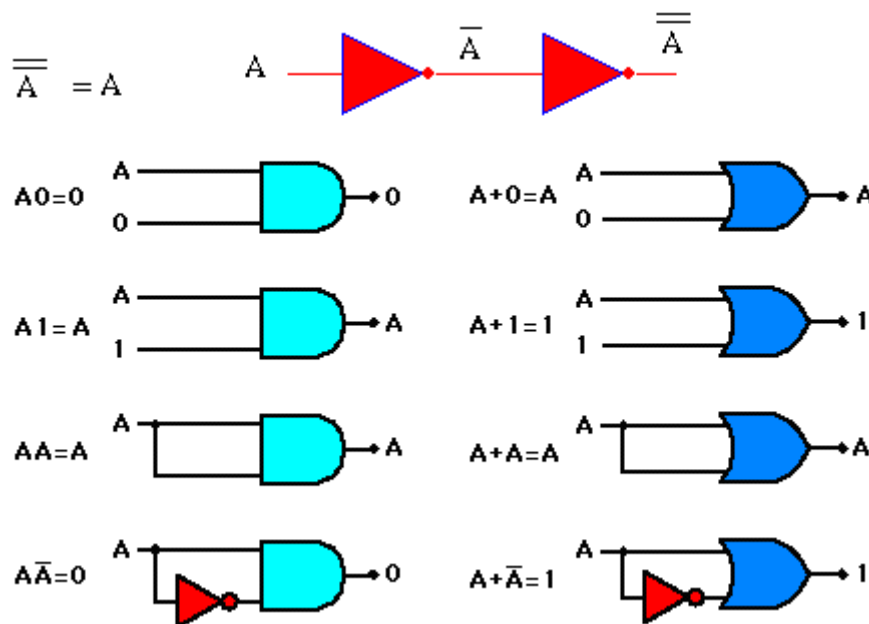
## 7. EXCLUSIVE NOR (EX – NOR)

Biến (Ngõ vào)		Hàm (Ngõ ra)
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



## 8. Các định lý đại số BOOLE

- Một biến số



- Giao hoán

$$AB = BA$$

$$A + B = B + A$$

- Kết hợp.

$$ABC = (AB)C = A(BC)$$

$$A + B + C = A + (B + C) = (A + B) + C$$

- Phân phối.

$$A(B + C) = AB + AC$$

$$(A + B)(C + D) = AC + AD + BC + BD$$

- Một số đẳng thức hữu dụng

$$A(A+B) = A$$

$$A+AB = A$$

$$AB+A\bar{B} = A$$

$$A+\bar{A}B = A+B$$

$$A(\bar{A}+B) = AB$$

$$(A+B)(A+\bar{B}) = A$$

$$(A+B)(A+C) = A+BC$$

$$AB+\bar{A}C+BC = AB+\bar{A}C$$

$$(A+B)(\bar{A}+C)(B+C) = (A+B)(\bar{A}+C)$$

- Định lý Morgan

$$\overline{ABC\dots} = \bar{A} + \bar{B} + \bar{C} + \dots$$

$$\overline{A+B+C+\dots} = \bar{A}\bar{B}\bar{C}\dots$$

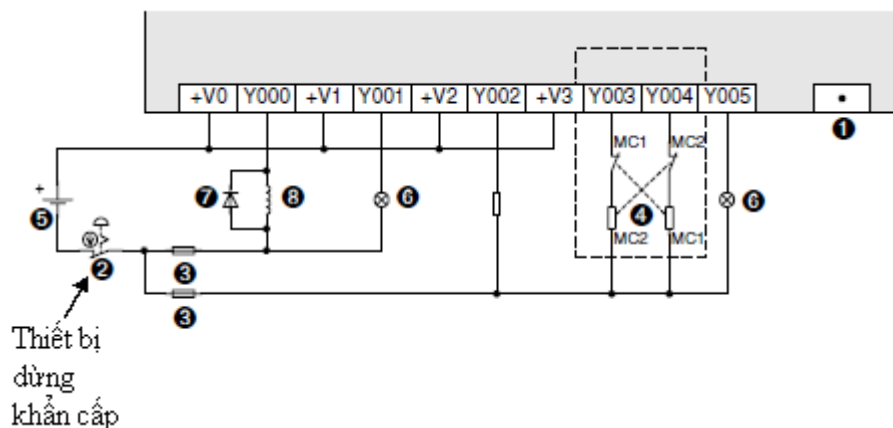
## VI. Quy tắc an toàn khi sử dụng PLC.

PLC có nhiều thuận lợi hơn so với điều khiển bằng cơ khí khi điều khiển các dây chuyền sản xuất. Tuy nhiên, ta không nên phó thác sự an toàn của người sử dụng, vận hành cho PLC.

### 1. Thiết bị dừng khẩn cấp.

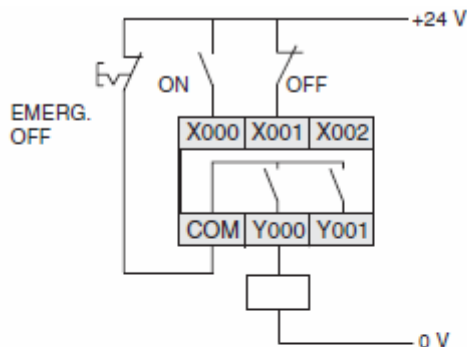
Đó là yêu cầu cần thiết để đảm bảo những lỗi trong hệ thống điều khiển không gây ra những mối nguy hiểm cho người vận hành. Thiết bị dừng khẩn phải dừng toàn bộ các thiết bị kể cả khi PLC không hoạt động.

Không bao giờ kết nối thiết bị dừng khẩn cấp một cách đơn độc như là ngõ vào của PLC, bị sự điều khiển của PLC. Điều này rất nguy hiểm.

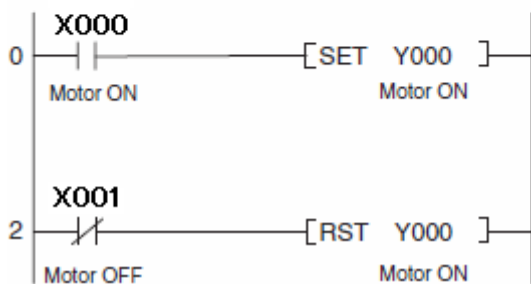


### 2. Đề phòng trong trường hợp đứt cable tín hiệu ngõ vào

Ta phải đảm bảo sự an toàn cho người vận hành kể cả khi bị đứt cable tín hiệu điều khiển ngõ vào dùng để ngắt thiết bị ở ngõ ra.



Ở ví dụ bên, contactor dùng để điều khiển cho hệ thống có thể được tắt bằng tay thông qua thiết bị dừng khẩn cấp (Emergency Off).

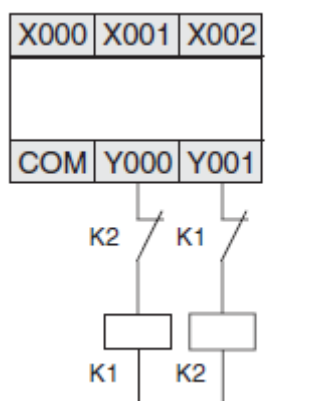


hoặc kết nối giữa contact Off và X002 bị ngắt.

Trong chương trình, để tạo contact ON ta sử dụng lệnh LD, tạo contact ngắt OFF sử dụng lệnh LDI. Ngõ ra cũng được điều khiển tương tự, ngõ ra ở trạng thái OFF khi X001 ở trạng thái “0”. Trạng thái này xảy ra khi contact Off hoạt động

### 3. Khóa chéo.

Nếu có hai ngõ ra mà cả hai không được hoạt động cùng một lúc, chẳng hạn như một động cơ hoạt động 2 chiều quay, sự khóa chéo phải được thực hiện bằng cách khóa chéo vật lý ở ngõ ra bằng các tiếp điểm của contactor (contactor được điều khiển bởi PLC). Điều này rất cần thiết, nếu chỉ khóa chéo trong chương trình có thể xảy ra sự cố nếu PLC bị lỗi thì cả 2 ngõ ra đều được tác động.



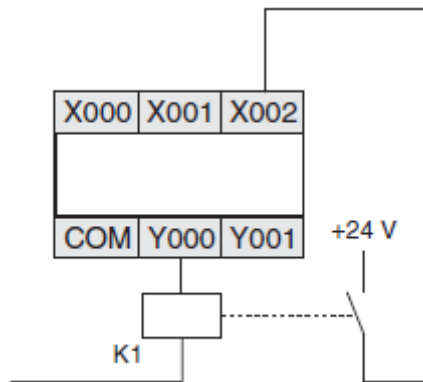
Ở ví dụ trên, cả 2 contactor đều không thể hoạt động cùng một thời điểm do bị khóa chéo bởi tiếp điểm của 2 contactor.

### 4. Tự động tắt máy

Khi PLC được dùng để điều khiển một hoạt động tuần tự mà trong đó có những mối nguy hiểm có thể nảy sinh. Khi thiết bị chuyển động qua những điểm nhất định, contact hành trình được cài đặt sẽ tự động ngắt sự chuyển động của thiết bị.

## **5. Phản hồi tín hiệu ngõ ra**

Thông thường ngõ ra của PLC không được theo dõi. Khi một đầu ra được kích hoạt, chương trình giả thuyết rằng sự đáp ứng đúng xảy ra bên ngoài PLC. Trong đa số trường hợp không có thiết bị bổ sung bên ngoài. Tuy nhiên, trong một số ứng dụng giới hạn cần phải theo dõi đầu ra những tín hiệu của PLC, ví dụ như lỗi mạch điện ở ngõ ra (đứt dây, dính tiếp điểm) có thể gây ra hậu quả nghiêm trọng cho sự an toàn hay hệ thống vận hành.



Ở ví dụ trên, tiếp điểm của contactor K1 dùng để cấp tín hiệu ngõ vào X002 khi ngõ ra Y000 được kích hoạt. Điều này dùng để cho phép chương trình theo dõi đầu ra và contactor có hoạt động đúng theo yêu cầu hay không. Chú ý, giải pháp đơn giản này không phải dùng để kiểm tra thiết bị có hoạt động đúng chức năng hay không. Những thiết bị cần thiết để kiểm tra điều này, chẳng hạn như cảm biến tốc độ.