

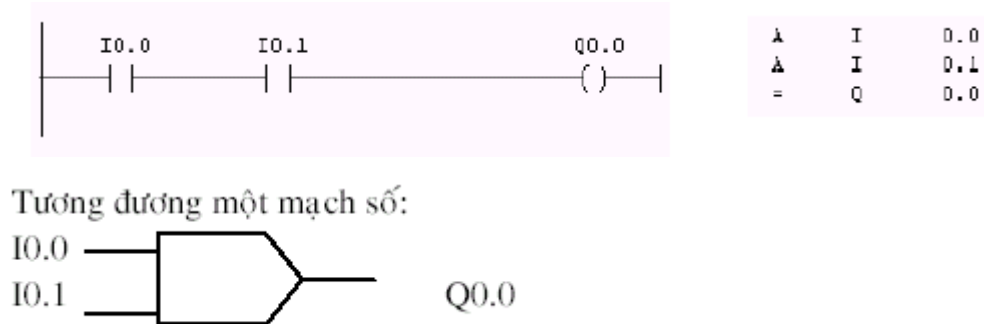
## Chương 1: TỔNG QUAN VỀ PLC

### 1.1 GIỚI THIỆU VỀ PLC (Programmable Logic Control) (Bộ điều khiển logic khả trình)

Hình thành từ nhóm các kỹ sư hãng General Motors năm 1968 với ý tưởng ban đầu là thiết kế một bộ điều khiển thỏa mãn các yêu cầu sau:

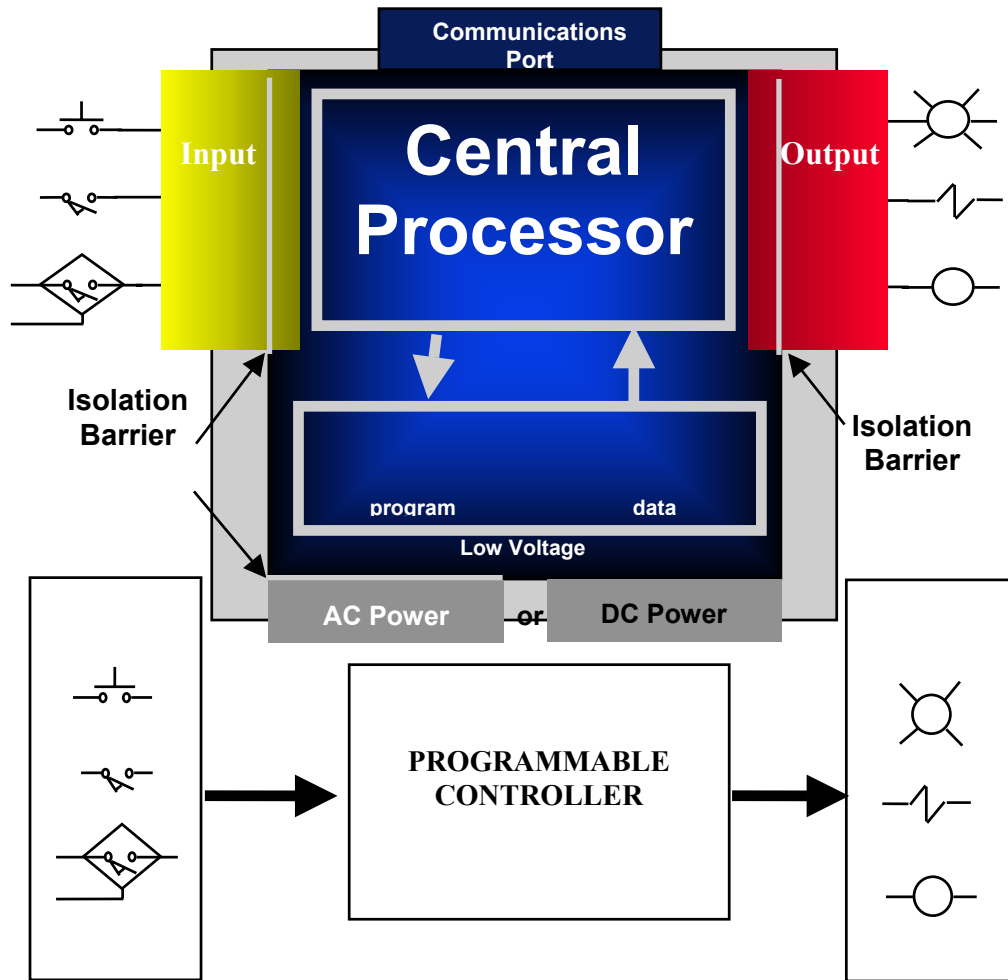
- Lập trình dễ dàng, ngôn ngữ lập trình dễ hiểu.
- Dễ dàng sửa chữa thay thế.
- Ổn định trong môi trường công nghiệp.
- Giá cả cạnh tranh.

Thiết bị điều khiển logic khả trình (PLC: Programmable Logic Control) (hình 1.1) là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển số thông qua một ngôn ngữ lập trình, thay cho việc thể hiện thuật toán đó bằng mạch số.



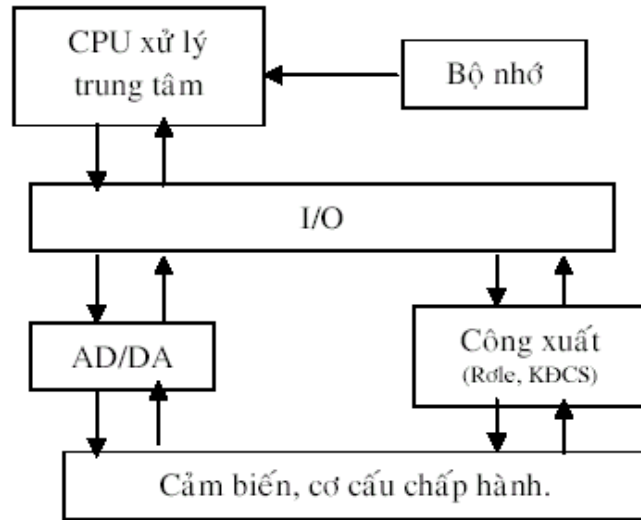
Hình 1.1

Như vậy, với chương trình điều khiển trong mình, PLC trở thành bộ điều khiển số nhỏ gọn, dễ thay đổi thuật toán và đặc biệt dễ trao đổi thông tin với môi trường xung quanh (với các PLC khác hoặc với máy tính). Toàn bộ chương trình điều khiển được lưu nhớ trong bộ nhớ PLC dưới dạng các khối chương trình (khối OB, FC hoặc FB) và thực hiện lặp theo chu kỳ của vòng quét.



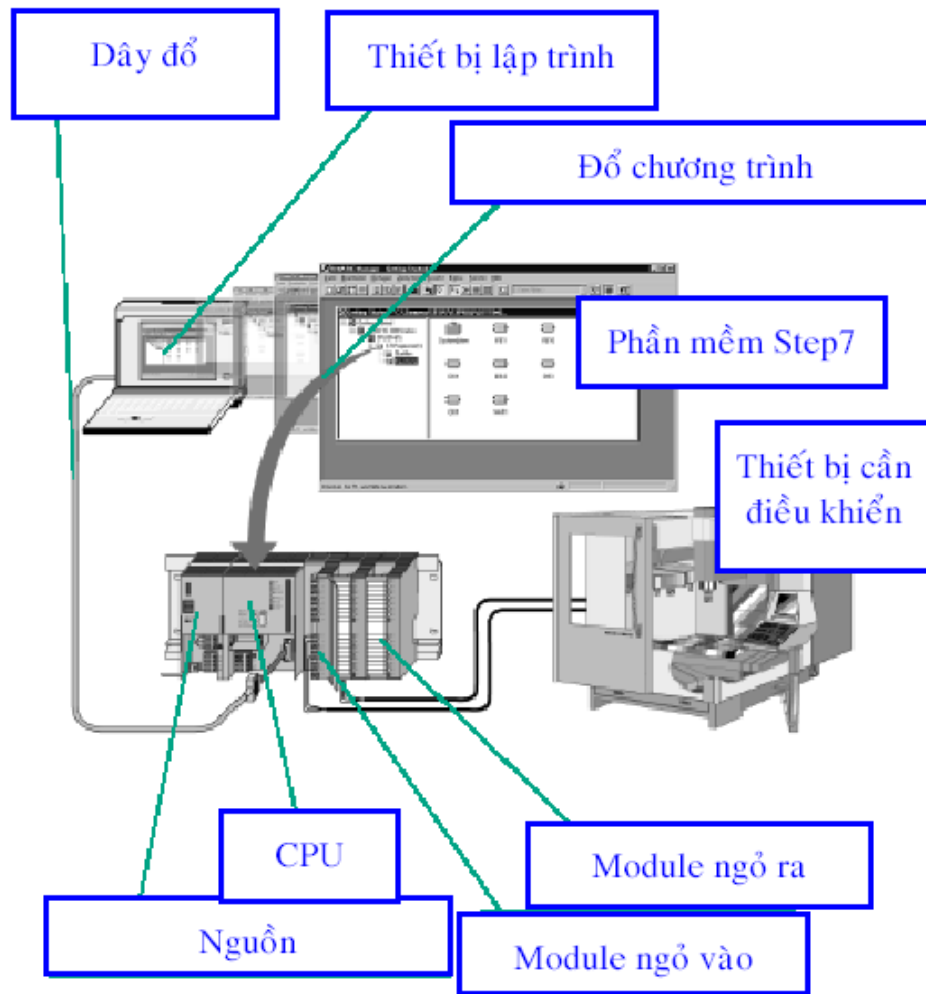
Hình 1.2

Để có thể thực hiện được một chương trình điều khiển, tất nhiên PLC phải có tính năng như một máy tính, nghĩa là phải có một bộ vi xử lý (CPU), một hệ điều hành, bộ nhớ để lưu chương trình điều khiển, dữ liệu và các cổng vào/ra để giao tiếp với đối tượng điều khiển và trao đổi thông tin với môi trường xung quanh. Bên cạnh đó, nhằm phục vụ bài toán điều khiển số, PLC còn cần phải có thêm các khối chức năng đặc biệt khác như bộ đếm (Counter), bộ định thì (Timer) ... và những khối hàm chuyên dụng.



Hình 1.3

**Hệ thống điều khiển sử dụng PLC**



Hình 1.4 Hệ thống điều khiển dùng PLC

## 1.2 PHÂN LOẠI

PLC được phân loại theo 2 cách:

➤ Hãng sản xuất: Gồm các nhãn hiệu như Siemen, Omron, Misubishi, Alenbratlay...

➤ Version:

Ví dụ: PLC Siemen có các họ: S7-200, S7-300, S7-400, Logo.

PLC Misubishi có các họ: Fx, Fx<sub>0</sub>, Fx<sub>ON</sub>

## 1.3 CÁC BỘ ĐIỀU KHIỂN VÀ PHẠM VI ỨNG DỤNG

### 1.3.1 Các bộ điều khiển

Ta có các bộ điều khiển: Vi xử lý, PLC và Máy tính.

### 1.3.2 Phạm vi ứng dụng

#### 1.3.2.1 Máy tính

- Dùng trong những chương trình phức tạp đòi hỏi độ chính xác cao.
- Có giao diện thân thiện
- Tốc độ xử lý cao
- Có thể lưu trữ với dung lượng lớn

#### 1.3.2.2 Vi xử lý

- Dùng trong những chương trình có độ phức tạp không cao (vì chỉ xử lý 8 bit).
- Giao diện không thân thiện với người sử dụng
- Tốc độ tính toán không cao.
- Không lưu trữ hoặc lưu trữ với dung lượng rất ít

#### 1.3.2.3 PLC

- Độ phức tạp và tốc độ xử lý không cao
- Giao diện không thân thiện với người sử dụng
- Không lưu trữ hoặc lưu trữ với dung lượng rất ít
- Môi trường làm việc khắc nghiệt

## 1.4 CÁC LĨNH VỰC ỨNG DỤNG PLC

PLC được sử dụng khá rộng rãi trong các ngành: Công nghiệp, Máy nông nghiệp, Thiết bị y tế, Ô tô (xe hơi, cần cẩu...)...

## 1.5 CÁC ƯU ĐIỂM KHI SỬ DỤNG HỆ THỐNG ĐIỀU KHIỂN VỚI PLC:

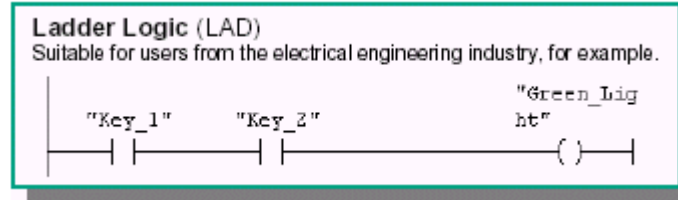
- Không cần đấu dây cho sơ đồ điều khiển logic như kiểu dùng rơ le.
- Có độ mềm dẻo sử dụng rất cao, khi chỉ cần thay đổi chương trình (phần mềm) điều khiển.
- Chiếm vị trí không gian nhỏ trong hệ thống.
- Nhiều chức năng điều khiển.
- Tốc độ cao.
- Công suất tiêu thụ nhỏ.
- Không cần quan tâm nhiều về vấn đề lắp đặt.
- Có khả năng mở rộng số lượng đầu vào/ra khi nối thêm các khối vào/ra chức năng.
- Tạo khả năng mở ra các lĩnh vực áp dụng mới.
- Giá thành không cao.

Chính nhờ những ưu thế đó, PLC hiện nay được sử dụng rộng rãi trong các hệ thống điều khiển tự động, cho phép nâng cao năng suất sản xuất, chất lượng và sự đồng nhất sản phẩm, tăng hiệu suất, giảm năng lượng tiêu tốn, tăng mức an toàn, tiện nghi và thoải mái trong lao động. Đồng thời cho phép nâng cao tính thị trường của sản phẩm.

## 1.6 GIỚI THIỆU CÁC NGÔN NGỮ LẬP TRÌNH

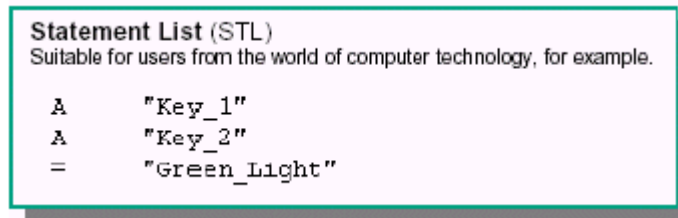
Các loại PLC nói chung thường có nhiều ngôn ngữ lập trình nhằm phục vụ các đối tượng sử dụng khác nhau. PLC S7-300 có 5 ngôn ngữ lập trình cơ bản. Đó là:

- Ngôn ngữ “hình thang”, ký hiệu là LAD (Ladder logic).



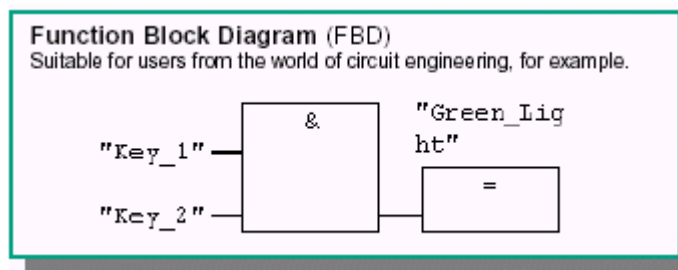
Đây là ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch logic.

- Ngôn ngữ “liệt kê lệnh”, ký hiệu là STL (Statement list).



Đây là dạng ngôn ngữ lập trình thông thường của máy tính. Một chương trình được ghép gởi nhiều câu lệnh theo một thuật toán nhất định, mỗi lệnh chiếm một hàng và đều có cấu trúc chung là “tên lệnh” + “toán hạng”.

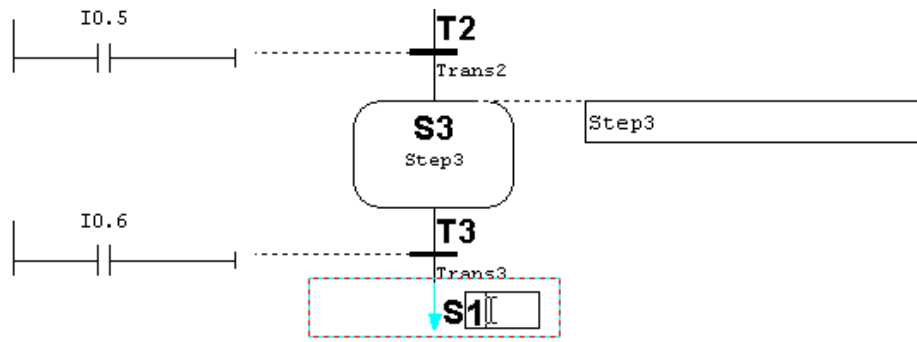
- Ngôn ngữ “hình khối”, ký hiệu là FBD (Function Block Diagram).



Đây cũng là ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch điều khiển số.

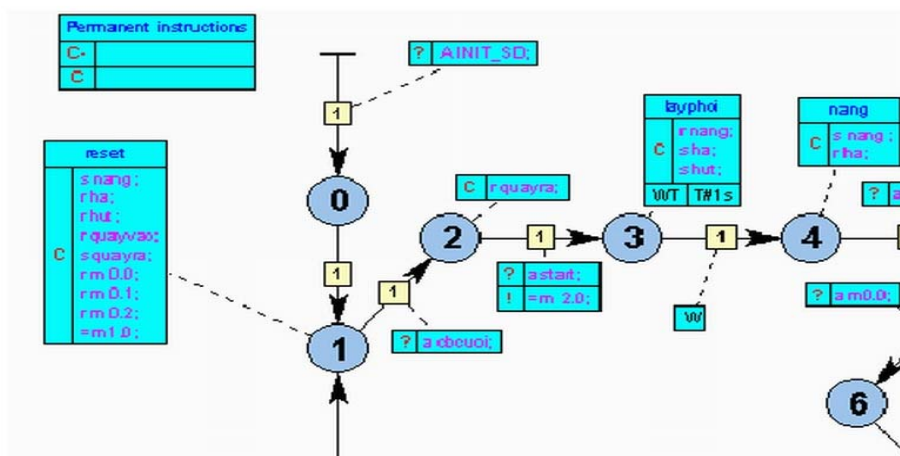
- Ngôn ngữ GRAPH.

Đây là ngôn ngữ lập trình cấp cao dạng đồ họa. Cấu trúc chương trình rõ ràng, chương trình ngắn gọn. Thích hợp cho người trong ngành cơ khí vốn quen với giản đồ Grafacet của khí nén.



Hình 1.5

➤ Ngôn ngữ High GRAPH.



Hình 1.6

Là dạng ngôn ngữ lập trình phát triển từ ngôn ngữ lập trình GRAPH.

## Chương 2: CẤU TRÚC PHẦN CỨNG PLC HỌ S7

### 2.1 CÁC TIÊU CHUẨN VÀ THÔNG SỐ KỸ THUẬT HỌ S7-200, S7-300

Xem phụ lục 1

### 2.2 CÁC TÍNH NĂNG CỦA PLC S7-300, S7-200

#### 2.2.1 S7-300

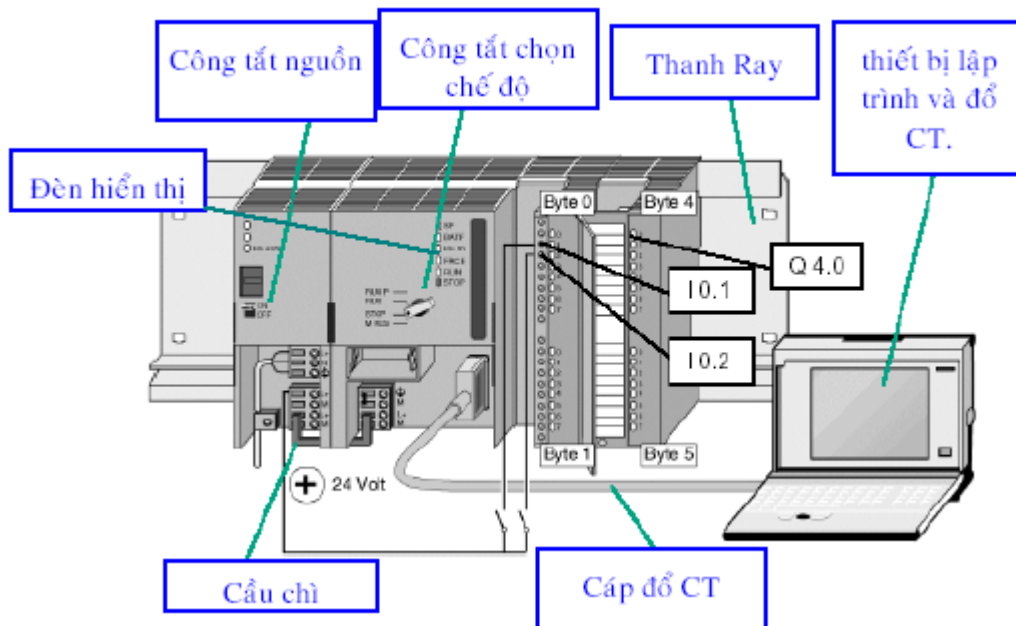
- Hệ thống điều khiển kiểu Module nhỏ gọn cho các ứng dụng trong phạm vi trung bình
- Có nhiều loại CPU
- Có nhiều Module mở rộng
- Có thể mở rộng đến 32 Module
- Các Bus nối tích hợp phía sau các Module
- Có thể nối mạng Multipoint Interface (MPI), Profibus hoặc Industrial Ethernet
- Thiết bị lập trình trung tâm có thể truy cập đến các Module
- Không hạn chế rãnh
- Cài đặt cấu hình và thông số với công cụ trợ giúp “HW-Config.”

#### 2.2.2 S7-200

- Hệ thống điều khiển kiểu Module nhỏ gọn cho các ứng dụng trong phạm vi hẹp
- Có nhiều loại CPU
- Có nhiều Module mở rộng
- Có thể mở rộng đến 7 Module
- Bus nối tích hợp trong Module ở mặt sau
- Có thể nối mạng với cổng giao tiếp RS 485 hay Profibus
- Máy tính trung tâm có thể truy cập đến các Module
- Không qui định rãnh cắm
- Phần mềm điều khiển riêng
- Tích hợp CPU, I/O nguồn cung cấp vào một Module
- “Micro PLC với nhiều chức năng tích hợp.”

## 2.3 CÁC MODULE CỦA PLC S7-300, S7-200

### 2.3.1 S7-300



Hình 2.1

#### ➤ Module CPU






Module CPU là module chứa bộ vi xử lý, hệ điều hành, bộ nhớ, các bộ định thì, bộ đếm, cổng truyền thông (RS 485) ... và có thể còn có một vài cổng vào/ra số. Các cổng vào/ra số có trên module CPU được gọi là cổng vào/ra *onboard* như CPU 314IFM.

Trong họ PLC S7-300 có nhiều loại module CPU khác nhau. Nói chung chúng được đặt tên theo bộ vi xử lý có trong nó như module CPU312, module CPU314, module CPU315...

Những module cùng sử dụng một loại bộ vi xử lý, nhưng khác nhau về cổng vào/ra onboard cũng như các khối hàm đặc biệt được tích hợp sẵn trong thư viện của hệ điều hành phục vụ việc sử dụng các cổng vào/ra onboard này sẽ phân biệt với nhau trong tên gọi bằng cách thêm cụm chữ cái IFM (Intergrated Function Module). Ví dụ module CPU313IFM, module CPU314IFM...

Ngoài ra, còn có các loại module CPU với hai cổng truyền thông, trong đó cổng truyền thông thứ hai có chức năng chính là phục vụ việc nối mạng phân tán như mạng PROFIBUS (PROcess FIEld BUS). Tất nhiên kèm theo cổng truyền thông thứ hai này là những phần mềm tiện dụng thích hợp cũng đã được cài sẵn trong hệ điều hành. Các loại module CPU này được phân biệt với các loại module CPU khác bằng cách thêm cụm từ DP (Distributed Port). Ví dụ như module CPU315-2DP. Tham khảo hình dưới:

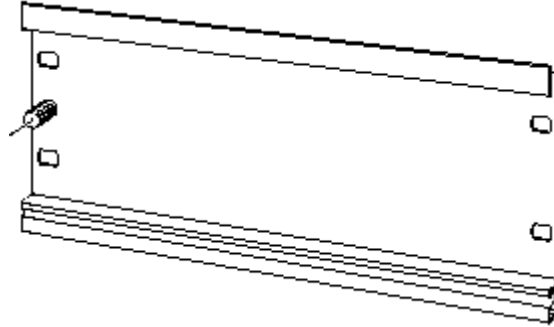


CPU 312 IFM CPU 313 CPU 314 IFM CPU 314	CPU 315-2 DP CPU 316-2 DP		CPU 318-2	
MPI interface	MPI interface	PROFIBUS-DP interface	MPI/DP Interface	PROFIBUS-DP interface
				
-	-	-	Reconfiguration as a PROFIBUS-DP interface is possible	-

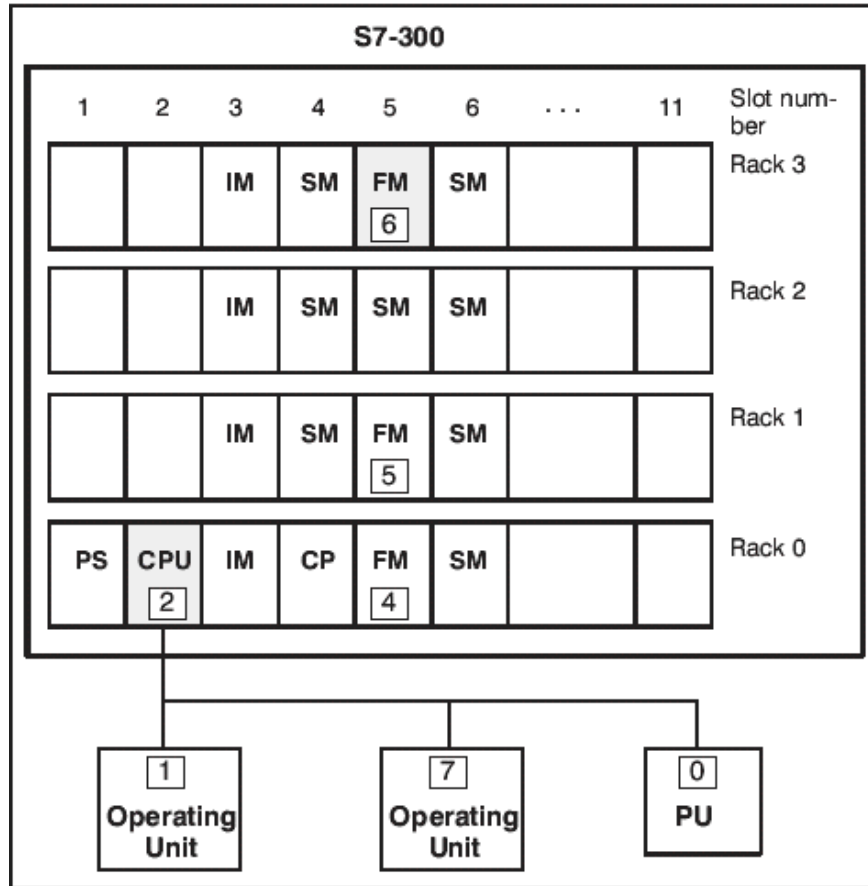
Hình 2.2 Cổng giao tiếp của các PLC

**Các loại module mở rộng:**

- **PS (Power Supply):** Module nguồn nuôi, có 3 loại 2A, 5A và 10A.
- **SM (Signal Module):** Module mở rộng cổng tín hiệu vào/ra, gồm có:
  - **DI (Digital Input):** Module mở rộng các cổng vào số với số lượng cổng có thể là 8, 16 hoặc 32 tùy theo từng loại module. Gồm 24VDC và 120/230V AC.
  - **DO (Digital Output):** Module mở rộng các cổng ra số với số lượng cổng có thể là 8, 16 hoặc 32 tùy theo từng loại module. Gồm 24VDC và ngắt điện từ.
  - **DI/DO (Digital Input/Digital Out):** Module mở rộng các cổng vào/ra số với số lượng cổng có thể là 8 vào/8 ra hoặc 16 vào/16 ra tùy theo từng loại module.
  - **AI (Analog Input):** Module mở rộng các cổng vào tương tự. Về bản chất chúng là những bộ chuyển đổi tương tự số 12 bits (AD), tức là mỗi tín hiệu tương tự được chuyển đổi thành một tín hiệu số (nguyên) có độ dài 12 bits. Số các cổng vào tương tự có thể là 2, 4 hoặc 8 tùy theo loại module. Tín hiệu vào có thể là áp, dòng, điện trở.
  - **AO (Analog Output):** Module mở rộng các cổng ra tương tự. Chúng là những bộ chuyển đổi số tương tự 12 bits (DA). Số các cổng ra tương tự có thể là 2, 4 hoặc 8 tùy theo loại module. Tín hiệu ra có thể là áp hoặc dòng.
  - **AI/AO (Analog Input/Analog Output):** Module mở rộng các cổng vào/ra tương tự. Số các cổng tương tự có thể là 4 vào/2 ra hoặc 4 vào/4 ra tùy theo từng loại module.
- **IM (Interface Module):** Module ghép nối. Đây là loại module chuyên dụng có nhiệm vụ nối từng nhóm các module mở rộng lại với nhau thành một khối và được quản lý chung bởi một module CPU. Thông thường các module mở rộng được gá liền với nhau trên một thanh đỡ gọi là *rack* (hình 2.3). Trên mỗi thanh rack chỉ có thể gá tối đa 8 module mở rộng (không kể module CPU, nguồn nuôi). Một module CPU S7-300 có thể làm việc trực tiếp với nhiều nhất 4 racks và các racks này phải được nối với nhau bằng module IM (xem hình 2.4).



Hình 2.3 Thanh rack



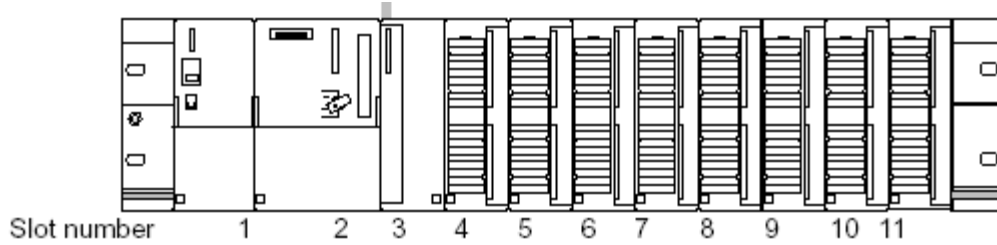
Hình 2.4 Sơ đồ phân bố các racks

➤ **FM (Function Module):** Module có chức năng điều khiển riêng, ví dụ như module điều khiển động cơ servo, module điều khiển động cơ bước, module PID, module điều khiển vòng kín, Module đếm, định vị, điều khiển hồi tiếp ...

➤ **CP (Communication Module):** Module phục vụ truyền thông trong mạng (MPI, PROFIBUS, Industrial Ethernet) giữa các PLC với nhau hoặc giữa PLC với máy tính.

Hình sau là cấu hình đầy đủ của một thanh Rack và sơ đồ kết nối nhiều Rack:

PS CPU SM/FM/CP



Hình 2.5 Cấu hình một thanh rack của PLC S7-300



Hình 2.6 Cấu hình tổng quát của một PLC S7-300 với 4 thanh rack nối với nhau nhờ module IM và cáp nối 368

- Phụ kiện  
Bus nối dữ liệu (Bus connector).
- Kiểm tra phần cứng

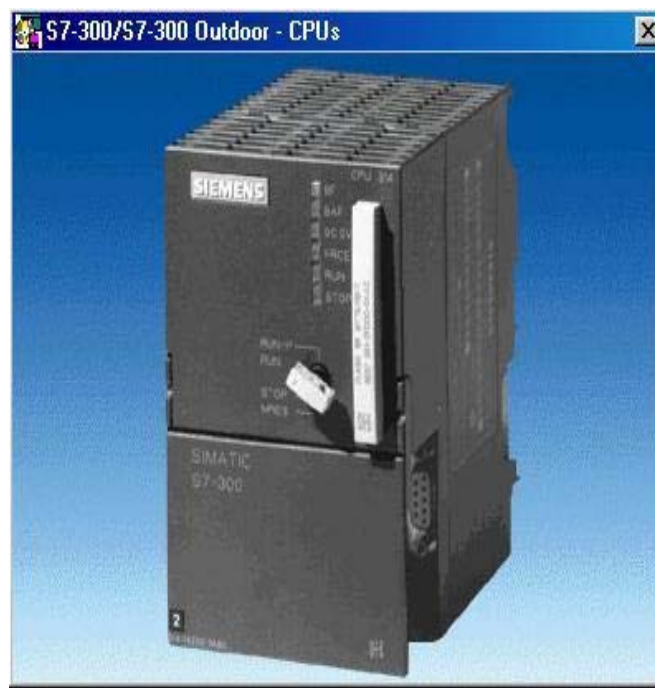
**Kiểm tra bằng cách nhìn LED ở bộ nguồn**

**Bảng 2.1**

LED “DC 24V	TRẠNG THÁI	PHẢN ỨNG CỦA NGUỒN
Sáng liên tục	Có điện áp 24V	Có điện áp 24V
Chớp	Mạch ra quá tải:	

LED “DC 24V	TRẠNG THÁI	PHẢN ỨNG CỦA NGUỒN
	Cao đến 130% (động) Cao đến 130% (tĩnh)	Bị sụt áp Điện áp được hồi phục khi không còn quá tải Cao đến 130% tĩnh Điện áp suy giảm, giảm tuổi thọ
Off	Ngắn mạch lối ra	Mất điện áp, tự động hồi phục khi ngắn mạch được loại bỏ
Off	Quá áp hay thấp áp phần sơ cấp	Quá áp có thể gây thiệt hại. Khi bị thấp áp tự động ngắt

### Kiểm tra bằng cách nhìn LED ở CPU của S7 – 300

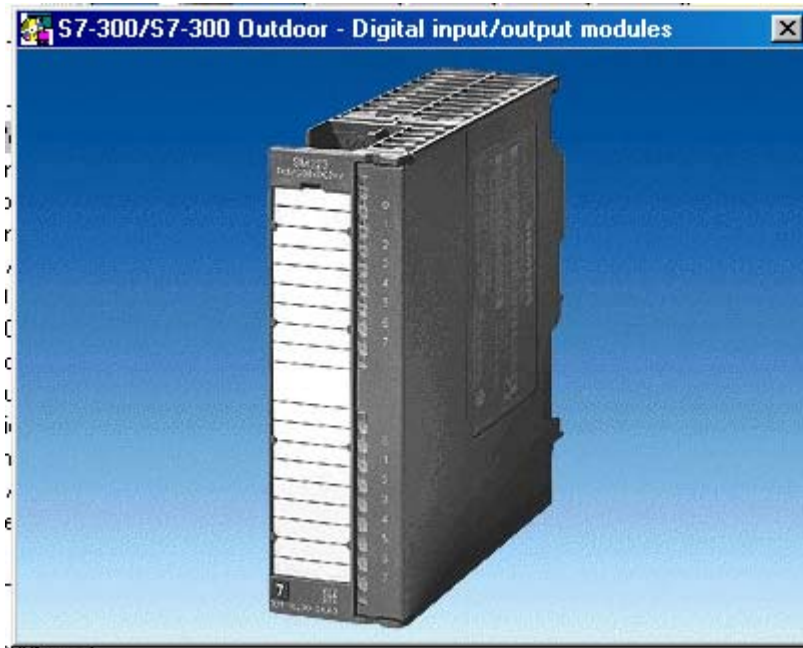


Hình 2.7

- Trạng thái hiển thị LED:
  - SF = Lỗi nhóm, chương trình sai hay lỗi từ khối chuẩn đoán
  - BATF = Lỗi Pin, Pin hết hay không có pin
  - DC5V = Báo có 5 VDC
  - FRCE = Sáng lên khi biến cường bức tác động
  - RUN = Nhấp nháy khi CPU khởi động, ổn định ở chế độ RUN
  - STOP = Ổn định ở chế độ STOP  
Chớp chậm khi có yêu cầu RESET bộ nhớ  
Chớp nhanh khi đang RESET bộ nhớ
- Chìa khóa công tắc: Để đặt bằng tay các trạng thái hoạt động của CPU
- MRES = Reset bộ nhớ (Reset khối)

- STOP = Trạng thái dừng STOP, chương trình không thực hiện
- RUN-P = Trạng thái chạy RUN, CPU thực hiện chương trình
- RUN = Chương trình được thực hiện, hoặc có thể, tuy nhiên, chỉ đọc thôi không sửa được chương trình.

**Kiểm tra bằng cách nhìn LED ở khối Digital**



Hình 2.8

Mỗi kênh vào/ra của các modul đều được hiển thị bằng LED. Chúng có công dụng có thể cho biết vị trí lỗi của chương trình. Chúng chỉ thị trạng thái quy trình hoặc trạng thái bên trong trước bộ giao tiếp quang điện.

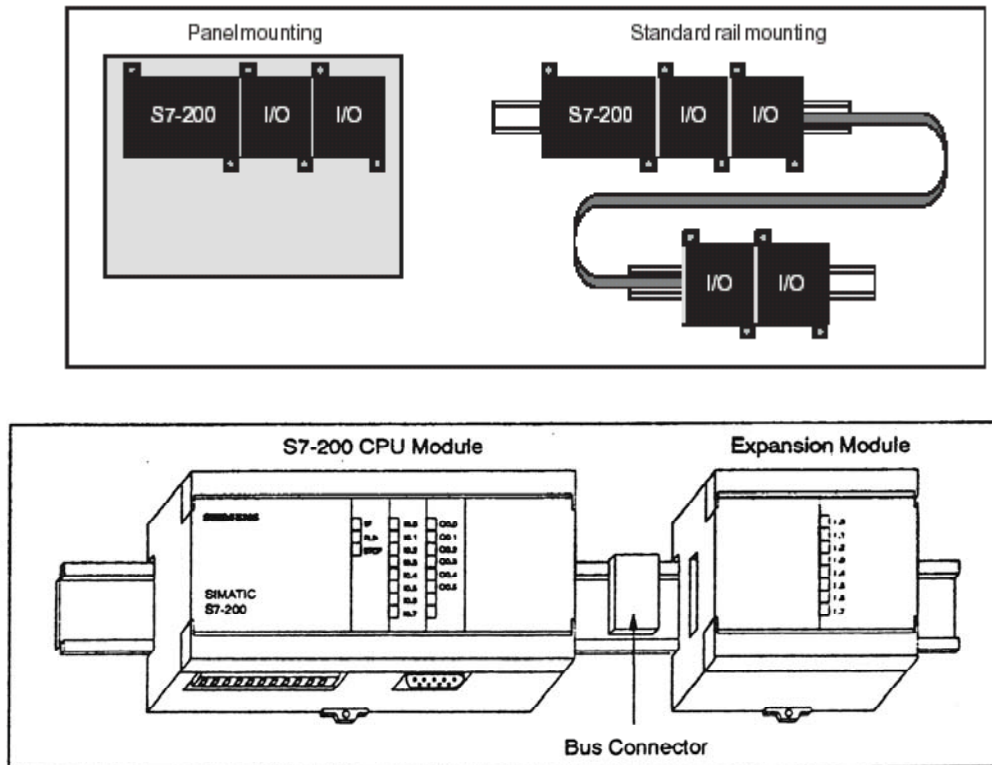
**2.3.1 S7-200**

DC24V OUTPUTS										STOP RUN		VR1	
1M 1L+0.0 0.1 0.2 0.3 0.4 2M2L+0.5 0.6 0.7 1.0 1.1										TERM		0 1	
<b>SIEMENS</b>          <b>SIMATIC S7-200</b>				<input type="checkbox"/> SF <input type="checkbox"/> RUN <input type="checkbox"/> STOP		<input type="checkbox"/> I0.0 <input type="checkbox"/> I0.1 <input type="checkbox"/> I0.2 <input type="checkbox"/> I0.3 <input type="checkbox"/> I0.4 <input type="checkbox"/> I0.5 <input type="checkbox"/> I0.6 <input type="checkbox"/> I0.7		<input type="checkbox"/> I1.0 <input type="checkbox"/> I1.1 <input type="checkbox"/> I1.2 <input type="checkbox"/> I1.3 <input type="checkbox"/> I1.4 <input type="checkbox"/> I1.5		<input type="checkbox"/> Q0.0 <input type="checkbox"/> Q0.1 <input type="checkbox"/> Q0.2 <input type="checkbox"/> Q0.3 <input type="checkbox"/> Q0.4 <input type="checkbox"/> Q0.5 <input type="checkbox"/> Q0.6 <input type="checkbox"/> Q0.7		<b>CPU-214</b>          <b>6ES7 214-1AC00</b>	
				DC 24V INPUTS									

Hình 2.9

❖ Tích hợp CPU, I/O nguồn cung cấp vào một Module. Có nhiều loại CPU: CPU212, CPU 214, CPU 215, CPU 216... Hình dáng CPU 214 thông dụng nhất được mô tả trên hình 2.1.

- ❖ Các Module mở rộng (EM) (External Modules):
  - ✓ Module ngõ vào Digital: 24V DC, 120/230V AC
  - ✓ Module ngõ ra Digital: 24V DC, ngắt điện từ
  - ✓ Module ngõ vào Analog: Áp, dòng, điện trở, cặp nhiệt
  - ✓ Module ngõ ra Analog: Áp, dòng



Hình 2.10 Ráp nối Module mở rộng

- ❖ Module liên lạc xử lý (CP) (Communication Processor)
 

Module CP 242-2 có thể dùng để nối S7-200 làm chủ module giao tiếp AS. Kết quả là, có đến 248 phần tử nhị phân được điều khiển bằng 31 Module giao tiếp AS. Gia tăng đáng kể số ngõ vào và ngõ ra của S7-200.

- ❖ Phụ kiện
  - Bus nối dữ liệu (Bus connector).

- ❖ Các đèn báo trên CPU

Các đèn báo trên mặt PLC cho phép xác định trạng thái làm việc hiện hành của PLC:

SF (đèn đỏ) : khi sáng sẽ thông báo hệ thống PLC bị hỏng.

RUN (đèn xanh) : khi sáng sẽ thông báo PLC đang làm việc và thực hiện chương trình được nạp vào máy.

STOP (đèn vàng) : khi sáng thông báo PLC đang ở chế độ dừng. Dừng chương trình đang thực hiện lại.

Ix.x (đèn xanh) : Thông báo trạng thái tức thời của cổng vào PLC: Ix.x (x.x = 0.0 ÷ 1.5). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

Qy.y (đèn xanh) : Thông báo trạng thái tức thời của cổng ra PLC: Qy.y (y.y = 0.0 ÷ 1.1). Đèn này báo hiệu trạng thái của tín hiệu theo giá trị logic của cổng.

- ❖ Công tắc chọn chế độ làm việc của CPU:

Công tắc này có 3 vị trí : RUN – TERM - STOP, cho phép xác lập chế độ làm việc của PLC.

-**RUN** : Cho phép PLC vận hành theo chương trình trong bộ nhớ. Khi trong PLC đang ở RUN, nếu có sự cố hoặc gặp lệnh STOP, PLC sẽ rời khỏi chế độ RUN và chuyển sang chế độ STOP.

- **STOP:** Cường bức CPU dừng chương trình đang chạy và chuyển sang chế độ STOP. Ở chế độ STOP, PLC cho phép hiệu chỉnh lại chương trình hoặc nạp chương trình mới.

- **TERM :** Cho phép máy lập trình tự quyết định chế độ làm việc của CPU hoặc ở chế độ RUN hoặc STOP.

## 2.4 GIỚI THIỆU CẤU TẠO PHẦN CỨNG CÁC KIT THÍ NGHIỆM S7-300, S7-200

### 2.4.1 S7-300

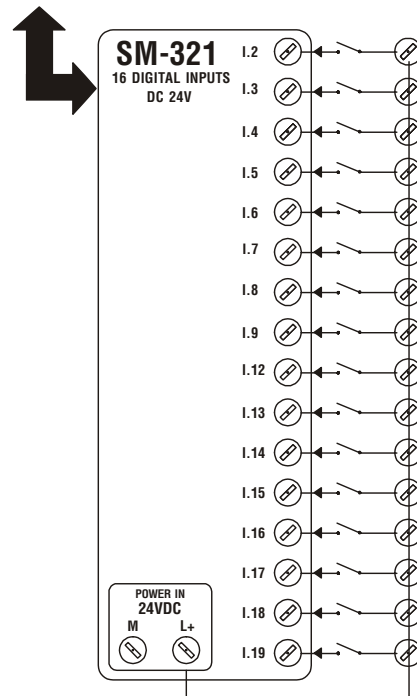
Bộ điều khiển PLC S7-300 Station 1200 bao gồm các module sau:

#### 2.4.1.1 Khối CPU-312 :

CPU-312 IMF sử dụng cho hệ thống thí nghiệm với thế nuôi 24VDC, có bổ sung 10 lối vào số /24VDC, và 6 lối ra /24VDC.

#### 2.4.1.2 Digital Input Module (DI) SM 321 DI 16 x DC24V (hình 2.11)

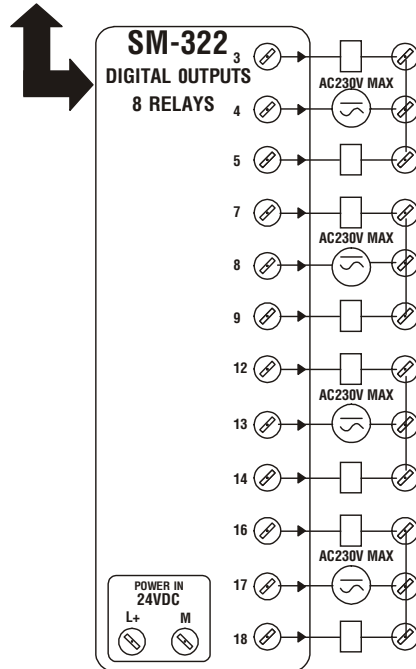
- 16 lối vào số (digital) độc lập và cách ly, điện thế cho lối vào 24V/10mA, chỉ thị LED trạng thái vào
- Điện thế nuôi cho khối (qua chân L+ & M): 24V.
- Chức năng mở rộng lối vào điều khiển cho CPU S7-300.



Hình 2.11: Sơ đồ khối SM 321 DI 16 x DC 24V / 321 – 1BH02-0AA0

#### 2.4.1.3 Digital Output Module (DO) SM 322 (hình 2.12)

- 8 lối ra relay độc lập, dòng giới hạn ở tiếp điểm relay: 2A. Chỉ thị LED trạng thái ra.
- Điện thế nuôi cho khối (qua chân L+ & M): 24V.

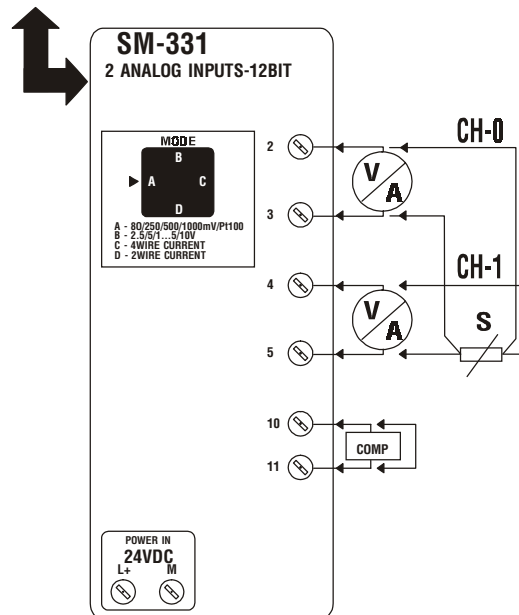


Hình 2.12 Sơ đồ khối SM 322 DO 8 x RELAY AC 230V / 322-1HF01-0AA0

- Chức năng mở rộng lối ra điều khiển cho CPU S7-300.

#### 2.4.1.3 Analog Input Module SM 331 (hình 2.13)

- 2 lối vào analog độc lập, phân giải 12 bit,
- Điện thế nuôi cho khối (qua chân L+ & M): 24V.
- Chức năng mở rộng lối vào điều khiển cho CPU S7-300.

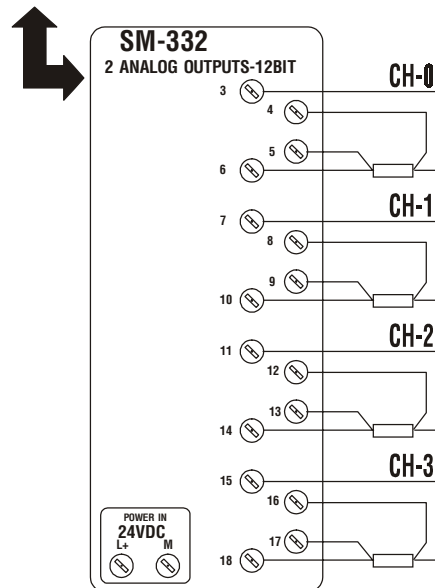


Hình 2.13 Sơ đồ khối SM 331 AI 2 x 12 bit / 331-7KB02-0AB0



#### 2.4.1.4 Analog Output Module SM 332 (Hình 2.14)

- 2 lối ra analog độc lập, phân giải 12 bit,
- Điện thế nuôi cho khối (qua chân L+ & M): 24V.
- Chức năng mở rộng lối ra điều khiển cho CPU S7-300.



Hình 2.14 Sơ đồ khối SM 332 AO 4 x 12 bit / 332-5HD01-0AB0

#### 2.4.1.5 Các khối phụ trợ cho thí nghiệm

Các khối phụ trợ cho thí nghiệm gồm các module chứa công tắc, relay, đèn báo, có cấu trúc như trên hình 1.9.

➤ **Khối Contact LSW-16**

Chứa 16 công tắc đơn, phục vụ cho việc tạo các trạng thái lối vào cho PLC.

➤ **Khối Relay RL-16**

Chứa 16 relay 24V, sử dụng với tác động điều khiển từ lối ra PLC.

➤ **Khối Đèn LL-16**

Chứa 16 đèn 24V, sử dụng để chỉ thị trạng thái điều khiển.

➤ **Khối AM-1 Simulator**

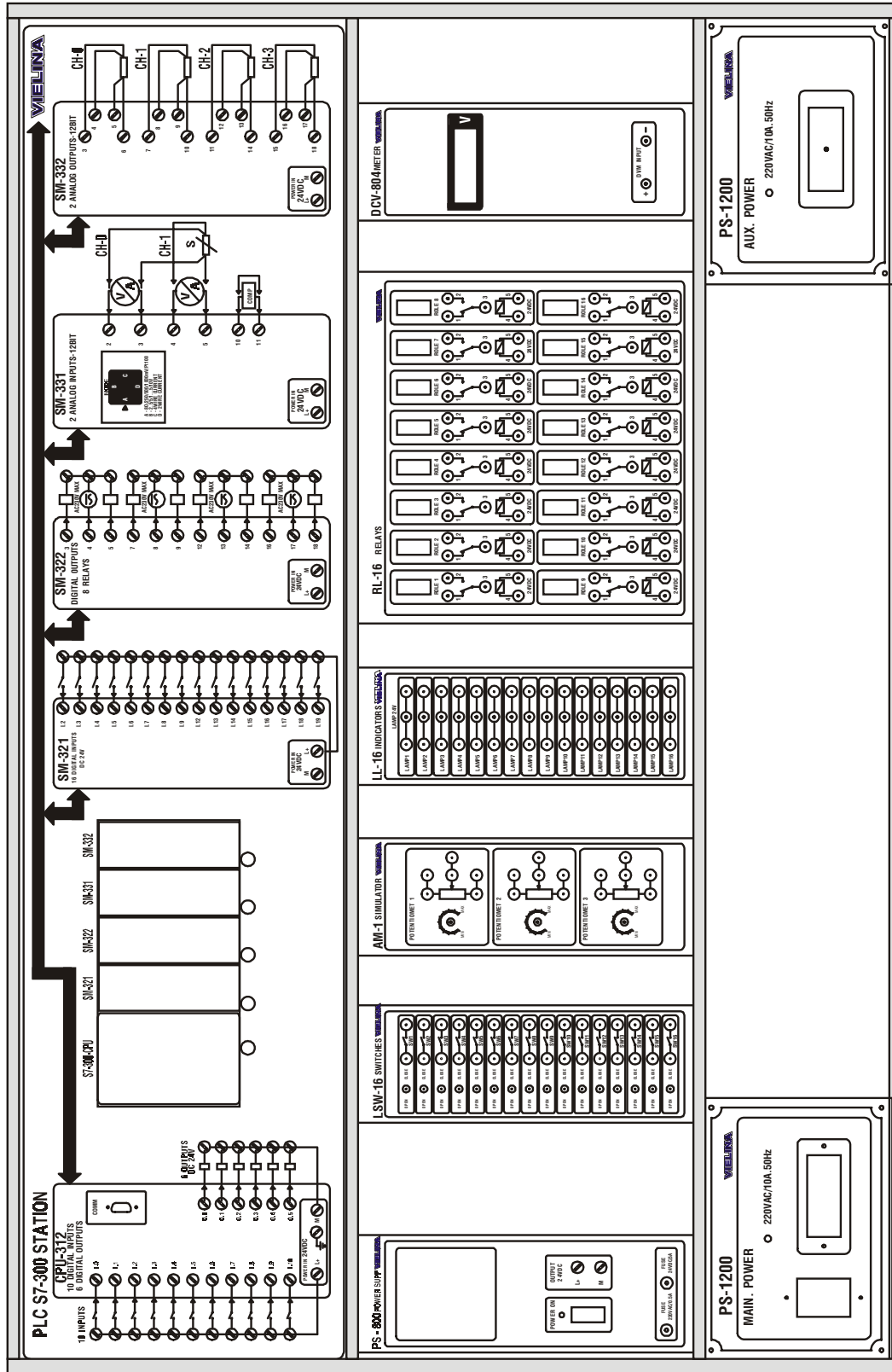
Chứa 3 biến trở 10kΩ, điện thế cấp 24V, cho phép tạo các điện thế DC cho thí nghiệm.

➤ **Khối DCV-804 Meter**

Chứa ADC với hiển thị  $4^{1/2}$  digits, cho phép đo điện thế DC tạo ra từ hệ PLC.

➤ **Khối nguồn 24V / 5A**

Cung cấp nguồn nuôi cho các modules.



Hình 2.15

**2.4.2 S7-200**

Hệ thống bao gồm các thiết bị :

1. Bộ điều khiển PLC-Station 1200 chứa :
  - CPU-214 : AC Power Supply, 24VDC Input, 24VDC Output
  - Digital Input / Output EM 223 : 4x DC 24V Input, 4x Relay Output
  - Analog Input / Output EM 235 : 3 Analog Input, 1 Analog Output 12bit
2. Khối Contact LSW-16
3. Khối Relay RL-16
4. Khối Đèn LL-16
5. Khối AM-1 Simulator
6. Khối DCV-804 Meter
7. Khối nguồn 24V PS-800
8. Máy tính.
9. Các dây nối với chốt cắm 2 đầu

**Mô tả hoạt động của hệ thống**

1. Các lối vào và lối ra CPU cũng như của các khối Analog và Digital được nối ra các chốt cắm.
2. Các khối PLC STATION – 1200, DVD – 804 và PS – 800 sử dụng nguồn 220VAC
3. Khối RELAY – 16 dùng các RELAY 24VDC
4. Khối đèn LL – 16 dùng các đèn 24V
5. Khối AM – 1 dùng các biến trở 10KΩ

Dùng các dây nối có chốt cắm 2 đầu và tùy từng bài toán cụ thể để đấu nối các lối vào / ra của CPU 214, khối Analog EM235, khối Digital EM222 cùng với các đèn, contact, Relay, biến trở, và khối chỉ thị DCV ta có thể bố trí rất nhiều bài thực tập để làm quen với cách hoạt động của một hệ thống PLC, cũng như cách lập trình cho một hệ PLC.

## Chương 3: KIẾN THỨC CƠ SỞ

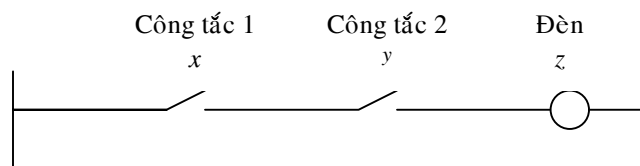
### 3.1 KỸ THUẬT SỐ VÀ LOGIC SỐ CƠ BẢN

#### 3.1.1 Biến và hàm số hai giá trị

Biến hai trị, hay còn gọi biến Boole là loại hàm số mà miền giá trị của nó chỉ có hai phần tử. Ta sẽ ký hiệu chúng bằng những chữ nhỏ in nghiêng như  $x, y, u, v, \dots$  và phần tử của chúng là 0 và 1. Ví dụ

- Công tắc là một biến Boole với 2 giá trị: đóng (ký hiệu là 1) và mở (ký hiệu là 0).
- Đèn hiệu cũng là một biến Boole với hai trạng thái: Sáng (ký hiệu là 1) và tắt (ký hiệu là 0).

Hai biến Boole được gọi là **độc lập nhau** nếu sự thay đổi giá trị của biến số này không ảnh hưởng đến giá trị của biến số kia. Ví dụ 2 công tắc trong hình 3.1 là 2 biến Boole độc lập với nhau.



Hình 3.1

Ngược lại, nếu giá trị của một biến số  $y$  phụ thuộc vào giá trị của biến số  $x$  thì biến  $y$  được gọi là **biến phụ thuộc** của biến  $x$ . Ví dụ trong hình 3.1 thì đèn là 2 biến phụ thuộc vào biến công tắc. Đèn sẽ sáng nếu cả 2 biến công tắc có giá trị 1 và sẽ tắt nếu một trong hai biến có giá trị 0.

Hàm hai trị là mô hình toán học mô tả sự phụ thuộc của một biến Boole vào các biến Boole khác. Chẳng hạn như để biểu diễn sự phụ thuộc của đèn, ký hiệu là  $z$ , vào 2 biến công tắc, ký hiệu là  $x$  và  $y$ , ta viết

$$z = f(x, y)$$

Một cách tổng quát hàm hai trị mô tả sự phụ thuộc của biến số  $y$  vào  $n$  biến  $x_1, x_2, \dots, x_n$  có dạng

$$y = f(x_1, x_2, \dots, x_n).$$

Việc mô tả sự phụ thuộc của một biến Boole này vào các biến Boole khác thành hàm hai trị dựa vào ba phép tính cơ bản. Đó là phép tính và (ký hiệu là  $\wedge$ ), hoặc (ký hiệu là  $\vee$ ), phủ định (ký hiệu là  $\bar{\phantom{x}}$ ) được định nghĩa như sau:

Bảng 3.1

Phép tính và			Phép tính hoặc			Phép phủ định	
$x$	$y$	$x \wedge y$	$x$	$y$	$x \vee y$	$x$	$\bar{x}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Ví dụ, hàm  $f(x,y)$  biểu diễn biến đèn  $z$  phụ thuộc vào hai biến công tắc  $x, y$  sẽ là:

$$z = f(x,y) = x \wedge y = x.y$$

### 3.1.2 Tính chất

$x \wedge 1 = 1 \wedge x = x$ , với  $x$  thuộc  $\mathbf{B} \rightarrow 1$  là phần tử đơn vị của phép toán  $\wedge$ .

$x \vee 0 = 0 \vee x = x$ , với  $x$  thuộc  $\mathbf{B} \rightarrow 0$  là phần tử đơn vị của phép toán  $\vee$ .

$x \vee y = y \vee x$  (Tính giao hoán)

$x \vee (y \wedge z) = (x \vee y) \wedge z$  (Tính kết hợp)

$(x \wedge y) \vee (x \vee \bar{y}) = x$

$\bar{\bar{x}} = x, \forall x \in B$

$x = x.x = x \vee x, \forall x \in B$

$\bar{x}.x = 0, \forall x \in B$

$1 \vee x = 1, \forall x \in B$

$0 \vee x = x, \forall x \in B$

$\bar{x \vee x} = 1, \forall x \in B$

$\overline{x.y} = \bar{x} \vee \bar{y}, \forall x, y \in B$

$\overline{x \vee y} = \bar{x} \wedge \bar{y}, \forall x, y \in B$

$(x \vee y).z = (x.z) \vee (y.z), \forall x, y, z \in B$

$(x.y) \vee z = (x \vee z).(y \vee z), \forall x, y, z \in B$

### 3.1.3 Xác định công thức hàm hai trị từ bảng chân lý

Ta sẽ xét bài toán ngược là tìm công thức biểu diễn hàm  $f(x)$  từ bảng giá trị chân lý đã biết của hàm đó. Công việc này là cần thiết vì trong thực tế nhiều bài toán tổng hợp bộ điều khiển được bắt đầu từ bảng chân lý.

Trước hết hãy làm quen với hai khái niệm mới là **biểu thức nguyên tố tích** và **biểu thức nguyên tố tổng** nếu trong  $T(x)$ :

- Có mặt tất cả các biến số  $x_k$ ,  $k=1, 2, \dots, n$  và mỗi biến chỉ xuất hiện một lần,
- Chỉ cấu thành bởi hai phép tính  $\wedge, \bar{\phantom{x}}$  hoặc  $\vee, \bar{\phantom{x}}$ .

Ví dụ:

$$T_1(x_1, x_2, x_3) = \bar{x}_1 \cdot x_2 \cdot x_3 \quad (\text{tạo bởi 2 phép toán } \wedge, \bar{\phantom{x}}),$$

$$T_2(x_1, x_2, x_3) = \bar{x}_1 \vee x_2 \vee \bar{x}_3 \quad (\text{tạo bởi 2 phép toán } \vee, \bar{\phantom{x}})$$

Là các biểu thức nguyên tố.

Biểu thức nguyên tố với 2 phép tính  $\wedge, \bar{\phantom{x}}$  được gọi là **biểu thức nguyên tố tích** còn biểu thức nguyên tố với 2 phép tính  $\vee, \bar{\phantom{x}}$  gọi là **biểu thức nguyên tố tổng**. Trong ví dụ trên  $T_1$  là biểu thức nguyên tố tích còn  $T_2$  gọi là **biểu thức nguyên tố tổng**.

Để tiện cho việc trình bày ta quy ước:

$$\bar{x}_k = x_k^0 \quad \text{và} \quad x_k = x_k^1$$

vậy một biểu thức nguyên tố tích  $T_N(x)$  với  $n$  biến 2 trị  $x_1, x_2, \dots, x_n$  có dạng

$$T_N(x) = x_1^{q_1} x_2^{q_2} \dots x_n^{q_n} = \prod_{k=1}^n x_k^{q_k}$$

và một biểu thức nguyên tố tổng  $T_C(x)$  với  $n$  biến 2 trị  $x_1, x_2, \dots, x_n$  có dạng

$$T_C(x) = x_1^{q_1} \vee x_2^{q_2} \vee \dots \vee x_n^{q_n} = \sum_{k=1}^n x_k^{q_k}$$

Trong đó

$q_k=0$  nếu biến  $x_k$  xuất hiện dưới dạng phủ định

$q_k=1$  nếu biến  $x_k$  xuất hiện dưới dạng không phủ định

Từ định nghĩa ta thấy các biểu thức nguyên tố có các đặc điểm:

- Biểu thức nguyên tố tích  $T_N(x)$  có giá trị 1 khi và chỉ khi tất cả các thừa số cùng có giá trị 1. Như vậy nếu  $x_k$  xuất hiện trong biểu thức dạng phủ định ( $q_k=0$ ) thì  $x_k$  phải có giá trị 0 và ngược lại nếu  $q_k=1$  thì  $x_k$  phải có giá trị 1.
- Biểu thức nguyên tố tổng  $T_C(x)$  có giá trị 0 khi và chỉ khi tất cả các thương số cùng có giá trị 0. Như vậy nếu  $x_k$  xuất hiện trong biểu thức dạng phủ định ( $q_k=0$ ) thì  $x_k$  phải có giá trị 1 và ngược lại nếu  $q_k=1$  thì  $x_k$  phải có giá trị 0.

Bây giờ ta xác định biểu thức hàm hai trị từ bảng chân lý của nó.

**Xác định nhờ biểu thức nguyên tố tích**

Biểu thức hàm hai trị  $f(x)$  sẽ tương đương với kết quả phép HOẶC của tất cả các biểu thức nguyên tố tích của các hàng có giá trị 1 trong bảng chân lý. Ta sẽ minh họa nguyên tắc bằng một ví dụ.

**Bảng 3.2**

$x_1$	$x_2$	$x_3$	$x_4$	$f(x)$	$T_N^i(x)$
0	0	0	0	0	$x_1 \vee x_2 \vee x_3 \vee x_4$
0	0	0	1	0	$x_1 \vee x_2 \vee x_3 \vee \bar{x}_4$
0	0	1	0	1	$\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4$
0	0	1	1	0	$x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4$
0	1	0	0	0	$x_1 \vee \bar{x}_2 \vee x_3 \vee x_4$
0	1	0	1	1	$\bar{x}_1 x_2 \bar{x}_3 x_4$
0	1	1	0	0	$x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4$
0	1	1	1	0	$x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$
1	0	0	0	0	$\bar{x}_1 \vee x_2 \vee x_3 \vee x_4$
1	0	0	1	1	$x_1 \bar{x}_2 \bar{x}_3 x_4$
1	0	1	1	0	$\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4$
1	1	0	0	0	$\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4$
1	1	0	1	1	$x_1 x_2 \bar{x}_3 x_4$
1	1	1	0	0	$\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4$
1	1	1	1	0	$\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$

Suy ra:  $f(x) = (\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4) \vee (\bar{x}_1 x_2 \bar{x}_3 x_4) \vee (x_1 \bar{x}_2 \bar{x}_3 x_4) \vee (x_1 x_2 x_3 x_4)$

### Xác định nhờ biểu thức nguyên tố tổng

Biểu thức hàm hai trị  $f(x)$  sẽ tương đương với kết quả phép AND của tất cả các biểu thức nguyên tố tổng của các hàng có giá trị 0 trong bảng chân lý. Ta sẽ minh họa nguyên tắc bằng ví dụ trên.

Suy ra:  $f(x) = (x_1 \vee x_2 \vee x_3 \vee x_4)(x_1 \vee x_2 \vee x_3 \vee \bar{x}_4)(x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4)$

$(x_1 \vee \bar{x}_2 \vee x_3 \vee x_4)(x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4)(x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)(\bar{x}_1 \vee x_2 \vee x_3 \vee x_4)$

$(\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4)(\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$

### 1.1.4 Biểu diễn số nguyên dương

#### ❖ Biểu diễn trong hệ cơ số 10

Một số nguyên dương  $u_k$  bất kỳ, trong hệ cơ số 10 bao giờ cũng được biểu diễn đầy đủ bằng dãy các con số nguyên từ 0 đến 9. Ví dụ  $u_k = 259$  được biểu diễn bằng 3 con số: 2, 5 và 9 và cách biểu diễn đó được hiểu là

$$U_k = 2 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0$$

Một cách tổng quát khi biểu diễn trong hệ cơ số 10  $u_k$  có dạng

$$U_k = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 \quad (3.1) \quad \text{với } 0 \leq a_i \leq 9$$

Như vậy việc biểu diễn  $u_k$  trong hệ cơ số 10 là sự biến đổi  $u_k$  thành tập hữu hạn  $n+1$  số nguyên  $a_i, i=0, 1, \dots, n$  thoả  $0 \leq a_i \leq 9$ .

Số các giá trị mà  $a_i$  có được do hệ cơ số biểu diễn  $u_k$  quyết định. Trong trường hợp này  $u_k$  được biểu diễn trong hệ cơ số 10 nên  $a_i$  có 10 giá trị.

### ❖ Biểu diễn trong hệ cơ số 2

Cách biểu diễn  $u_k$  trong hệ cơ số 10 chưa phù hợp với nguyên tắc mạch điện của bộ điều khiển số vì  $u_k$  có các phần tử đa trị  $0 \leq a_i \leq 9$ . Ta biến đổi biểu thức (3.1) về dạng sau

$$U_k = x_n \cdot 2^n + x_{n-1} \cdot 2^{n-1} + \dots + x_1 \cdot 2^1 + x_0 \cdot 2^0 \quad \text{với } a_i = 0, 1 \quad (3.2)$$

Với việc thay đổi này các tham số  $x_i, i=0, 1, \dots, n$  sẽ trở thành những đại lượng hai trị 0 hoặc 1. Nếu sử dụng vector hàng để biểu diễn ta sẽ có

$$U_k \rightarrow \begin{array}{|c|c|c|c|c|} \hline x_n & x_{n-1} & \dots & x_1 & x_0 \\ \hline \end{array}$$

ta sẽ đi đến dạng biểu diễn thông dụng bằng mạch điện cho tín hiệu số. Mỗi ô vuông trong cách biểu diễn trên gọi là một bit và mỗi bit là một biến hai trị.

Số các bit của vector  $x$  quyết định miền giá trị cho  $u_k$ . Với  $n+1$  bit trong (3.2) thì miền giá trị của  $u_k$  sẽ là tập các số nguyên dương trong khoảng  $0 \leq u_k \leq 2^{n+1} - 1$

Một dãy 8 bit được gọi là 1 byte. Hai byte gọi là 1 từ (word) và 2 từ gọi là từ kép (double word). Trong kỹ thuật PLC nói riêng và điều khiển số nói chung người ta thường biểu diễn  $u_k$  bằng một byte, 1 từ hoặc 1 từ kép.

Biểu diễn  $u_k=205$  thành một byte:

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

Một từ:

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

Cách biểu diễn trong hệ cơ số 2 như vậy không ảnh hưởng tới thói quen tính toán của ta trong hệ thập phân như cộng trừ. Tuy nhiên vẫn phải để ý rằng do  $x_i$  chỉ bằng 0 hoặc 1 nên khi cộng có tổng lớn hơn 1 ta phải viết  $x_i = 0$  và nhớ 1 sang hàng sau. Ví dụ khi cộng 53 và 27 trong hệ cơ số 2 sẽ có

$$\begin{array}{r} 53 = \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \\ 27 = \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \\ \text{nhớ} \quad \quad \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\ \text{tổng} \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \end{array}$$

### Mã hexadecimal của số nguyên dương



Giống như cách biểu diễn  $u_k$  trong hệ cơ số 10 và 2, trong hệ cơ số 16, số nguyên  $u_k$  có dạng

$$U_k = h_n \cdot 16^n + h_{n-1} \cdot 16^{n-1} + \dots + h_1 \cdot 16^1 + h_0 \cdot 16^0 \quad \text{với } 0 \leq h_i \leq 15 \quad (3.3)$$

và tham số  $h_i$  là những biến 16 trị. Các số của  $h_i$  ký hiệu là

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

trong đó các ký tự khi chuyển sang hệ thập phân sẽ tương đương với

A=10, B=11, C=12, D=13, E=14, F=15

Để bộ điều khiển số hiểu được dạng biểu diễn của  $u_k$ , người ta đã chuyển các tham số  $h_i$  sang hệ cơ số 2. Do mỗi tham số có 16 giá trị nên người ta cũng chỉ cần 4 bit là đủ để biểu diễn chúng.

Một mảng 4 bit có tên gọi là một *Nipple*

Ví dụ, số nguyên dương  $u_k = 7723$  trong hệ cơ số 10, khi chuyển sang cơ số 16 sẽ là 1E2B vì

$$7723 = 1 \cdot 16^3 + 14 \cdot 16^2 + 2 \cdot 16 + 11$$

1      E      2      B

và do đó dạng Hexadecimal của nó sẽ là

0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1				
				1				E				2				B			

### Mã BCD của số nguyên dương

Ta đã biết mã Hexadecimal là kiểu sử dụng biến hai trị để thể hiện các chữ số  $h_i$ , khi  $u_k$  được biểu diễn trong hệ cơ số 16. Hoàn toàn tương tự, mã BCD là dạng dùng biến hai trị thể hiện những chữ số  $0 < a_i < 9$  khi biểu diễn  $u_k$  trong hệ cơ số 10 theo công thức

$$U_k = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0 \quad \text{với } a_i \text{ thuộc } \{0,9\} \quad (3.3)$$

Ví dụ.  $U_k = 259$  được biểu diễn nhờ 3 con số 2, 5 và 9 và do đó mã BCD của nó có dạng

0	0	1	0	0	1	0	1	1	0	0	1
2				5				9			

### Chuyển đổi số:

#### ➤ Thập phân $\Leftrightarrow$ Nhị phân

VD: Chuyển số nhị phân 11011001 thành số thập phân

$$U_k = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

$$128 + 64 + 0 + 16 + 8 + 0 + 0 + 1 = 217$$

VD: Chuyển số thập phân 217 thành số nhị phân

$$217:2=108 \quad \text{dư } 1$$

$$108:2=54 \quad \text{dư } 0$$

$$54:2=27 \quad \text{dư } 0$$

$$27:2=13 \quad \text{dư } 1$$

$$13:2 = 6 \quad \text{dư } 1$$

$$6:2 = 3 \quad \text{dư } 0$$

$$3:2 = 1 \quad \text{dư } 1$$

1

$$\Rightarrow u_k = 11011001$$

➤ **Thập phân  $\Leftrightarrow$  Hexa**

VD: Chuyển số Hexa 2AC4 thành số thập phân

$$U_k = 2.16^3 + 10.16^2 + 12.16^1 + 4.16^0 \\ 8192 + 2560 + 192 + 4 = 10948$$

VD: Chuyển số thập phân 10948 thành số Hexa

$$10948:16 = 684 \quad \text{dư } 4$$

$$684:16 = 42 \quad \text{dư } 12 = C$$

$$42 :16 = 2 \quad \text{dư } 10 = A$$

2

$$\Rightarrow u_k = 2AC4$$

➤ **Hexa  $\Leftrightarrow$  Nhị phân**

VD: Chuyển số Nhị phân 1100 1011 0011 1001 thành số Hexa

$$1100 = 1.2^3 + 1.2^2 + 0.2^1 + 0.2^0 = 12 = C$$

$$1011 = 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 11 = B$$

$$0011 = 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 3$$

$$1001 = 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0 = 9$$

$$\Rightarrow u_k = CB39$$

VD: Chuyển số Hexa CB39 thành số Nhị phân

Chú ý cách nhớ giá trị thập phân của 4 bit: 8 4 2 1

$$C=12 = 8+4 = 1100$$

$$B=11 = 8+3 = 1011$$

$$3 = 2+1 = 0011$$

$$9 = 8+1 = 1001$$

$$\Rightarrow u_k = 1100 1011 0011 1001$$

➤ **BCD  $\Leftrightarrow$  Nhị phân**

Cách chuyển hoàn toàn tương tự với BCD  $\Leftrightarrow$  Hexa

VD: Chuyển số Nhị phân 0010 0111 0101 thành số BCD

$$0010 = 2$$

$$0111 = 0 + 4 + 2 + 1 = 7$$

$$0101 = 0 + 4 + 0 + 1 = 5$$

$$\Rightarrow u_k = 275$$

VD: Chuyển số BCD 275 thành số Nhị phân

$$2 = 0010$$

$$7 = 4+2+1 = 0111$$

$$5 = 4+1 = 0101$$

$\Rightarrow u_k = 0010\ 0111\ 0101$

## 3.2 CÁC VÙNG DỮ LIỆU, VÙNG NHỚ, CÁC QUY ĐỊNH VỀ DỮ LIỆU VÀ CÁCH TRUY CẬP ĐỊA CHỈ TRÊN PLC

### 3.2.1 S7-300

Kiểu dữ liệu và phân chia bộ nhớ

- Kiểu dữ liệu

**Bảng 3.3**

Dạng	Kích thước	Dạng Format	Tầm và ký hiệu (từ giá trị nhỏ nhất đến giá trị lớn nhất)	Ví dụ
BOOL (bit)	1	Boolean text	TRUE\FALSE	TRUE
BYTE (Byte)	8	Thập lục phân	B#16#0 đến B#16#FF	B#16#10 byte#16#10
WORD (Word)	16	Nhị phân Thập lục phân BCD Thập phân không dấu	2#0 đến 2#1111_1111_1111_1111 W#16#0 đến W#16#FFFF C#0 đến C#999 B#(0,0) đến B#(255,255)	2#0001_0010_0000_0011 W#16#1CBF Word#16#1CBF C#998 B#(10,20) Byte#(10,20)
DWORD (Double Word)	32	Nhị phân Thập lục phân Thập phân không dấu	2#0 đến 2#1111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 đến DW#16#FFFF_FFFF B#(0,0,0,0) đến B#(255,255,255,255)	2#0010_0111_1001_0000_0011_0100_1111_1000 DW#16#00A2_0FAB dword#16#00A2-0FAB B#(1,14,65,245) byte#(1,14,65,245)
INT (Integer)	16	Thập phân có	-32768 đến 32767	2

r)		dấu		
DINT (Double Integer)	32	Thập phân có dấu	L#-2147483648 đến L#2147483647	L#1
S5TIMER (Simatic Timer)	16	S5 time với đơn vị là 10ms	S5T#0H_0M_0S_10MS đến S5T#2H_46M_30S_0MS	S5T#1M S5TIME#1M
TIME (IEC Time)	32	IEC time với đơn vị là 1ms (số integer có dấu)	T#-24D_20H_31M_23S_648MS đến T#24D_20H_31M_23S_647MS	T#1H_1M TIME#1H_1M
DATE (IEC date)	16	Ngày hệ IEC với đơn vị 1ngày	D#1990-1-1 đến D#2168-12-31	D#1994-3-15 DATE#1994-3-15
TIME_OF_DAY (time of day)	32	Thời gian trong một ngày với đơn vị 1ms	TOD#0:0:0.0 đến TOD#23:59:59.999	TOD#1:10:3.3 TIME_OF_DAY#1:10:3.3
CHAR (character)	8	Ký tự	'A', 'B', 'c', .....	'c'

### Cấu trúc bộ nhớ của CPU

Bộ nhớ của S7-300 được chia thành 3 vùng chính

- Vùng chứa chương trình ứng dụng. Vùng nhớ chương trình được chia thành 3 miền OB (Organisation Block): Miền chứa chương trình tổ chức.

FC (Function): Miền chứa chương trình con được tổ chức thành hàm có biến hình thức để trao đổi dữ liệu với chương trình đã gọi nó.

FB (Function Block): Miền chứa chương trình con, được tổ chức thành hàm và có khả năng trao đổi dữ liệu với bất cứ một khối chương trình nào khác. Các dữ liệu này phải được xây dựng thành một khối dữ liệu riêng (gọi là DB-Data Block).

- Vùng chứa tham số của hệ điều hành và chương trình ứng dụng, được phân chia thành 7 miền khác nhau gồm

**I (Process image Input):** Miền bộ đệm các dữ liệu cổng vào số. Trước khi bắt đầu thực hiện chương trình, PLC sẽ đọc giá trị logic của tất cả các cổng đầu vào và cất giữ chúng trong vùng nhớ I. Thông thường chương trình ứng dụng không đọc trực tiếp trạng thái logic của cổng vào số mà chỉ lấy dữ liệu của cổng vào từ bộ đệm I.

**Q (Process image output):** Miền bộ đệm các dữ liệu cổng ra số. Kết thúc giai đoạn thực hiện chương trình, PLC sẽ chuyển giá trị logic của bộ đệm Q tới các cổng ra số. Thông thường chương trình ứng dụng không trực tiếp gán giá trị tới cổng ra mà chỉ chuyển vào bộ đệm Q.

**M:** Miền các biến cờ. Chương trình ứng dụng sử dụng vùng nhớ này để lưu giữ các tham số cần thiết và có thể truy cập nó theo bit (M), Byte (MB), từ (MW), hay từ kép (MD).

**T:** Miền nhớ phục vụ bộ thời gian (Timer) bao gồm việc lưu trữ giá trị thời gian đặt trước (PV- preset value), giá trị đếm thời gian tức thời (CV-Current value) cũng như giá trị logic đầu ra của bộ timer.

**C:** Miền nhớ phục vụ đếm (counter) bao gồm việc lưu trữ giá trị đặt trước (PV-preset value), giá trị đếm tức thời (CV-Current value) cũng như giá trị logic đầu ra của bộ counter.

**PI:** Miền địa chỉ cổng vào của các module tương tự (I/O external input). Các giá trị tương tự tại cổng vào của module tương tự sẽ được module đọc và chuyển tự động theo những địa chỉ. Chương trình ứng dụng có thể truy cập miền nhớ PI theo từng byte (PIB), từng từ (PIW) hoặc theo từ kép (PID).

**PQ:** Miền địa chỉ cổng ra của các module tương tự (I/O external Output). Các giá trị theo những địa chỉ này sẽ được module đọc và chuyển tới các cổng ra tương tự. Chương trình ứng dụng có thể truy cập miền nhớ PQ theo từng byte (PQB), từng từ (PQW) hoặc theo từ kép (PQD).

- Vùng chứa các khối dữ liệu, được chia làm 2 loại

**DB (Data block):** Miền chứa các dữ liệu được tổ chức thành khối. Kích thước cũng như số lượng khối do người sử dụng quy định, phù hợp với từng bài toán điều khiển. Chương trình có thể truy cập miền này theo từng bit (DBX), byte (DBB), từ (DBW) hoặc từ kép (DBD).

**L (Local data block):** Miền dữ liệu địa phương, được các khối chương trình OB, FC, FB tổ chức và sử dụng cho các biến nhấp tức thời và trao đổi dữ liệu của biến hình thức với những khối chương trình đã gọi nó. Nội dung của một số dữ liệu trong miền nhớ này sẽ bị xoá khi kết thúc chương trình tương ứng OB, FC, FB. Miền này có thể truy nhập từ chương trình theo bit (L), byte (LB), từ (LW) hoặc từ kép (LD).

#### Những khối OB đặt biệt

- OB10: Time of day Interrupt
- OB20: Time delay Interrupt
- Hardware Interrupt

- OB 81: Powersupply fault
- OB 100: Start Up information

### Cách truy cập địa chỉ

Địa chỉ ô nhớ trong Step7-300 gồm hai phần: phần chữ và phần số. Ví dụ



**Phần chữ** chỉ vị trí và kích thước của ô nhớ. Chúng có thể là:

- M: Chỉ ô nhớ trong miền các biến cờ có kích thước 1 bit
- MB: Chỉ ô nhớ trong miền các biến cờ có kích thước 1 byte (8bit)
- MW: Chỉ ô nhớ trong miền các biến cờ có kích thước 2 byte (16bit)
- MD: Chỉ ô nhớ trong miền các biến cờ có kích thước 4 byte (32 bit)
- I: Chỉ ô nhớ trong miền bộ đếm cổng vào số có kích thước 1 bit
- IB: Chỉ ô nhớ trong miền bộ đếm cổng vào số có kích thước 1 byte
- IW: Chỉ ô nhớ trong miền bộ đếm cổng vào số có kích thước 1 từ
- ID: Chỉ ô nhớ trong miền bộ đếm cổng vào số có kích thước 1 từ kép
- Q: Chỉ ô nhớ trong miền bộ đếm cổng ra số có kích thước 1 bit
- QB: Chỉ ô nhớ trong miền bộ đếm cổng ra số có kích thước 1 byte
- QW: Chỉ ô nhớ trong miền bộ đếm cổng ra số có kích thước 1 từ
- QD: Chỉ ô nhớ trong miền bộ đếm cổng ra số có kích thước 1 từ kép
- PIB: Chỉ ô nhớ có kích thước 1 byte thuộc vùng peripheral input. Thường là địa chỉ cổng vào của các module tương tự.
- PIW: Chỉ ô nhớ có kích thước 2 byte thuộc vùng peripheral input. Thường là địa chỉ cổng vào của các module tương tự.
- PID: Chỉ ô nhớ có kích thước 2 từ thuộc vùng peripheral input. Thường là địa chỉ cổng vào của các module tương tự.
- PQB: Chỉ ô nhớ có kích thước 1 byte thuộc vùng peripheral output. Thường là địa chỉ cổng ra của các module tương tự.
- PQW: Chỉ ô nhớ có kích thước 2 byte thuộc vùng peripheral output. Thường là địa chỉ cổng ra của các module tương tự.
- PQD: Chỉ ô nhớ có kích thước 2 từ thuộc vùng peripheral output. Thường là địa chỉ cổng ra của các module tương tự.

**Phần số:** Chỉ địa chỉ của byte hoặc bit trong miền nhớ đã xác định. Nếu ô nhớ đã được xác định thông qua phần chữ là có kích thước 1 bit thì phần số sẽ gồm địa chỉ của byte và số thứ tự của bit trong byte đó được tách với nhau bằng dấu chấm. Ví dụ

I 1.3 Chỉ bit thứ 3 trong byte 1 của miền nhớ bộ đếm cổng vào số.

M 101.5 Chỉ bit thứ 5 trong byte 101 của miền các biến cờ M.

Q 4.5 Chỉ bit thứ 5 trong byte 4 của miền nhớ bộ đếm cổng ra số.

Trong trường hợp ô nhớ đã được xác định là byte, từ hoặc từ kép thì phần số sẽ là địa chỉ byte đầu tiên trong mảng byte của ô nhớ đó. Ví dụ

MB15 Chỉ ô nhớ có kích thước 1 byte (byte 15) trong miền các biến cờ M  
 MW 18 Chỉ ô nhớ có kích thước 1 từ gồm 2 byte 18 và 19 trong miền các biến cờ M.

MD105 Chỉ ô nhớ có kích thước 1 từ kép gồm 4 byte 105, 106, 107 và 108 trong miền các biến cờ M.

### 3.2.2 S7-200

#### CẤU TRÚC BỘ NHỚ CỦA PLC S7 – 200

Bộ nhớ của S7 – 200 có tính năng động cao, có thể đọc và ghi được trong toàn vùng, ngoại trừ phần các bit nhớ đặc biệt được ký hiệu SM (Special Memory) chỉ có thể truy cập để đọc.

Bộ nhớ có một tụ nhớ để giữ thế nuôi, duy trì dữ liệu trong một khoảng thời gian khi mất điện.

Bộ nhớ của S7 – 200 được chia thành 4 vùng :

**2.2.2.1 Vùng nhớ chương trình :** Là vùng lưu giữ các lệnh chương trình. Vùng này thuộc kiểu không bị mất dữ liệu (non – volatile), đọc / ghi được.

**2.2.2.2 Vùng nhớ tham số :** Là vùng lưu giữ các thông số như : từ khóa, địa chỉ trạm. Cũng như vùng chương trình vùng tham số thuộc kiểu đọc ghi / được .

**2.2.2.3 Vùng nhớ dữ liệu:**

Được sử dụng để trữ các dữ liệu của chương trình. Đối với CPU 214, 1KByte đầu tiên của vùng nhớ này thuộc kiểu đọc / ghi được. Vùng dữ liệu là một miền nhớ động. Nó có thể được truy cập theo từng bit, từng byte, từng từ đơn (word), hoặc theo từng từ kép (Double word) và được dùng để lưu trữ dữ liệu cho các thuật toán, các hàm truyền thông, lập bảng, các hàm dịch chuyển, xoay vòng thanh ghi, con trỏ địa chỉ ...

Vùng dữ liệu được chia thành những vùng nhớ nhỏ với các công dụng khác nhau. Chúng được ký hiệu bằng chữ cái đầu tiếng Anh, đặc trưng cho công dụng riêng của chúng :

<b>V</b>	Variable memory
<b>I</b>	Input image resister
<b>O</b>	Ouput image resister
<b>M</b>	Internal memory bits
<b>SM</b>	Special memory bits

Tất cả các miền này đều có thể truy cập theo từng bit, từng byte, từng từ đơn, hoặc từng từ kép.

#### Vùng dữ liệu của CPU 214

- **Miền V ( đọc / ghi ) :**

7	6	5	4	3	2	1	0
V0							
...							
V4095							

- **Vùng đệm cổng vào I ( đọc / ghi ):**

7	6	5	4	3	2	1	0
I0.x ( x = 0 ÷ 7 )							
...							
I7.x ( x = 0 ÷ 7 )							

- **Vùng đệm cổng ra Q ( đọc / ghi ):**

7	6	5	4	3	2	1	0
Q0.x ( x = 0 ÷ 7 )							
...							
Q7.x ( x = 0 ÷ 7 )							

- **Vùng nhớ nội M ( đọc / ghi ):**

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

M0.x ( x = 0 ÷ 7 )
...
M31.x ( x = 0 ÷ 7 )

- **Vùng nhớ đặc biệt SM ( chỉ đọc ):**

7      6      5      4      3      2      1      0
SM0.x ( x = 0 ÷ 7 )
...
SM29.x ( x = 0 ÷ 7 )

- **Vùng nhớ đặc biệt ( đọc / ghi ) :**

7      6      5      4      3      2      1      0
SM30.x ( x = 0 ÷ 7 )
...
SM85.x ( x = 0 ÷ 7 )

**Địa chỉ truy nhập được với công thức :**

- **Truy nhập theo bit :** Tên miền (+) địa chỉ byte (+). (+) chỉ số bit.

*Ví dụ :* V150.4 chỉ bit 4 của byte 150.

- **Truy nhập theo byte :** Tên miền (+) B (+) địa chỉ của byte trong miền.

*Ví dụ :* VB150 chỉ byte 150 của miền V.

- **Truy nhập theo từ :** Tên miền (+) W (+) địa chỉ byte cao của từ trong miền.

*Ví dụ :* VW150 chỉ từ đơn gồm 2 byte 150 và 151 thuộc miền V trong đó byte 150 là byte cao trong từ.

15	14	13	12	11	10	9	8	7	5	4	3	2	1	0
VB150 (byte cao)								VB151 (byte thấp)						

- **Truy nhập theo từ kép :** Tên miền (+) D (+) địa chỉ của byte cao của từ trong miền.

*Ví dụ :* VD150 là từ kép 4 byte 150, 151, 152, 153 thuộc miền V trong đó byte 150 là byte cao và 153 là byte thấp trong từ kép.

63	32	31	16	15	8	7	0
VB150 (byte cao)	VB151			VB152	VB153 (byte thấp)		

Tất cả các byte thuộc vùng dữ liệu đều có thể truy nhập được bằng con trỏ. Con trỏ được định nghĩa trong miền V hoặc các thanh ghi AC1, AC2, AC3. Mỗi con trỏ chỉ địa chỉ gồm 4 byte (từ kép).

Quy ước dùng con trỏ để truy nhập như sau :

- **&địa chỉ byte (cao) :** Là toán hạng lấy địa chỉ của byte, từ hoặc từ kép.

*Ví dụ :*

AC1 = &VB150 : Thanh ghi AC1 chứa địa chỉ byte 150 thuộc miền V

VD100 = &VW150 : Từ kép VD100 chứa địa chỉ byte cao (VB150) của từ đơn VW150 AC2 = &VD150

: Thanh ghi AC2 chứa địa chỉ byte cao (VB150) của từ kép VD150.

- **contrô :** là toán hạng lấy nội dung của byte, từ, từ kép mà con trỏ đang chỉ vào.

*Ví dụ :* như với phép gán địa chỉ trên, thì :

\*AC1 : Lấy nội dung của byte VB150.

\*VD100 : Lấy nội dung của từ đơn VW100.

\*AC2 : Lấy nội dung của từ kép VD150



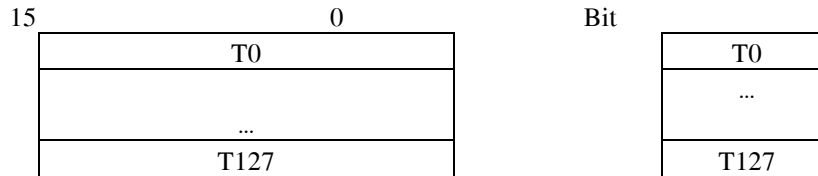
### 2.2.2.4 Vùng nhớ đối tượng

Vùng đối tượng được sử dụng để giữ dữ liệu cho các đối tượng lập trình như các giá trị tức thời, giá trị đặt trước của bộ đếm hay Timer. Dữ liệu kiểu đối tượng bao gồm các thanh ghi của Timer, bộ đếm, các bộ đếm tốc độ cao, bộ đệm vào/ra Analog và các thanh ghi Accumulator ( AC ).

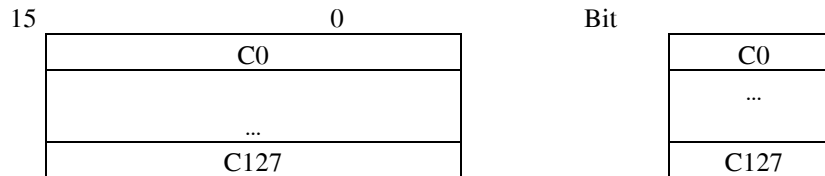
Kiểu dữ liệu đối tượng bị hạn chế rất nhiều vì các dữ liệu đối tượng chỉ được ghi theo mục đích cần sử dụng đối tượng đó.

**Vùng nhớ đối tượng được phân chia như sau :**

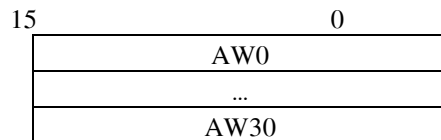
- Timer ( đọc / ghi ) :



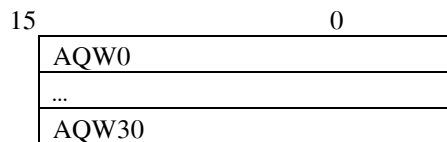
- Bộ đếm ( đọc / ghi ) :



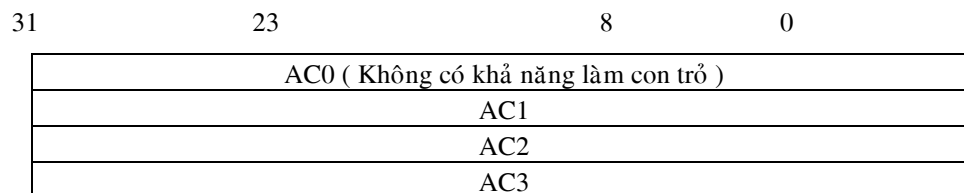
- Bộ đệm cổng vào tương tự ( chỉ đọc ) :



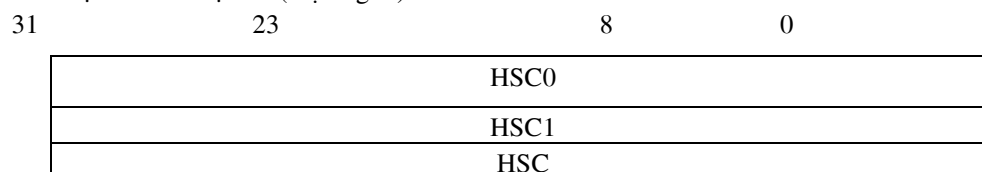
- Bộ đệm cổng ra tương tự ( chỉ ghi ) :



- Thanh ghi Accumulator ( đọc / ghi ) :



- Bộ đếm tốc độ cao ( đọc / ghi ) :



## Chương 4: Tập lệnh

### 4.1 CÁC LỆNH VÀO / RA

LAD	Mô tả	TOÁN HẠNG
n —   —	Tiếp điểm thường mở được đóng nếu n=1	n: I, Q, M, L, D, T, C
n — / —	Tiếp điểm thường đóng sẽ mở khi n=1	n: I, Q, M, L, D, T, C

**OUTPUT:** Sao chép nội dung của bit đầu tiên trong ngăn xếp vào bit được chỉ định trong lệnh. Nội dung của ngăn xếp không thay đổi.

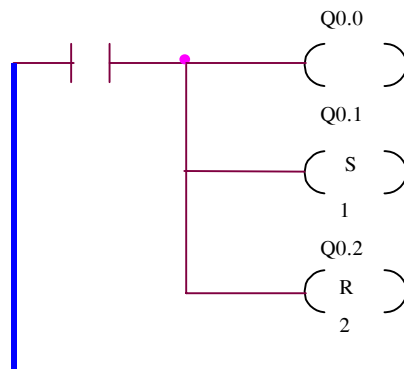
LAD	Mô tả	TOÁN HẠNG
n  —( )	Cuộn dây đầu ra được kích thích khi được cấp dòng điều khiển	n: I, Q, M, L, D, T, C

### 4.2 CÁC LỆNH GHI / XOÁ GIÁ TRỊ CHO TIẾP ĐIỂM

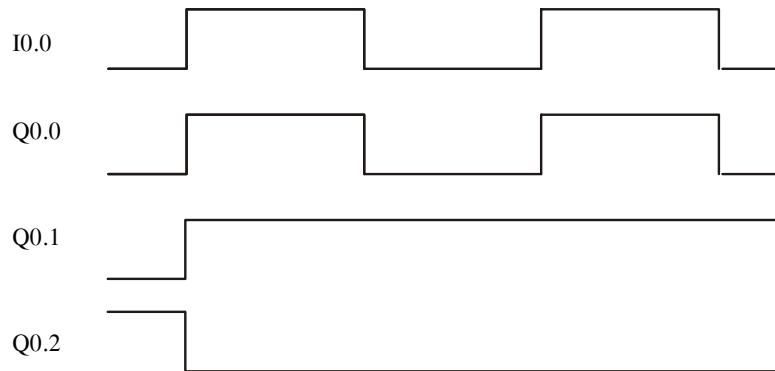
SET ( S )

RESET ( R )

Ví dụ mô tả các lệnh vào ra và S, R :



Giải đồ tín hiệu thu được ở các lối ra theo chương trình trên như sau :



Hình 4.1

### 4.3 CÁC LỆNH LOGIC ĐẠI SỐ BOOLEAN

Các lệnh làm việc với tiếp điểm theo đại số Boolean cho phép tạo sơ đồ điều khiển logic không có nhớ.

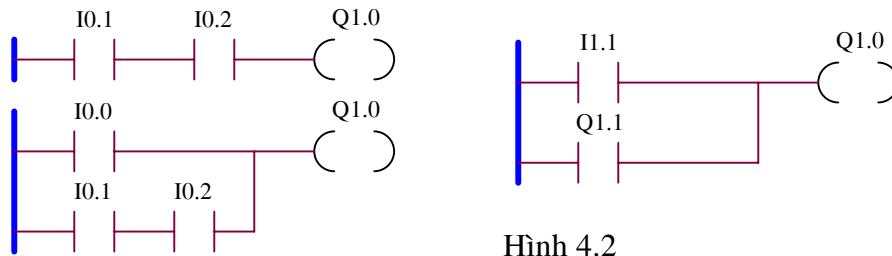
Trong LAD lệnh này được biểu diễn thông qua cấu trúc mạch mắc nối tiếp hoặc song song các tiếp điểm thường đóng hay thường mở.

Trong STL có thể sử dụng các lệnh A (And) và O (Or) cho các hàm hở hoặc các lệnh AN (And Not) và ON (Or Not) cho các hàm kín. Giá trị của ngăn xếp thay đổi phụ thuộc vào từng lệnh.

Các hàm logic boolean làm việc trực tiếp với tiếp điểm bao gồm :

**O** (Or) , **A** (And), **AN** (And Not), **ON** (Or Not)

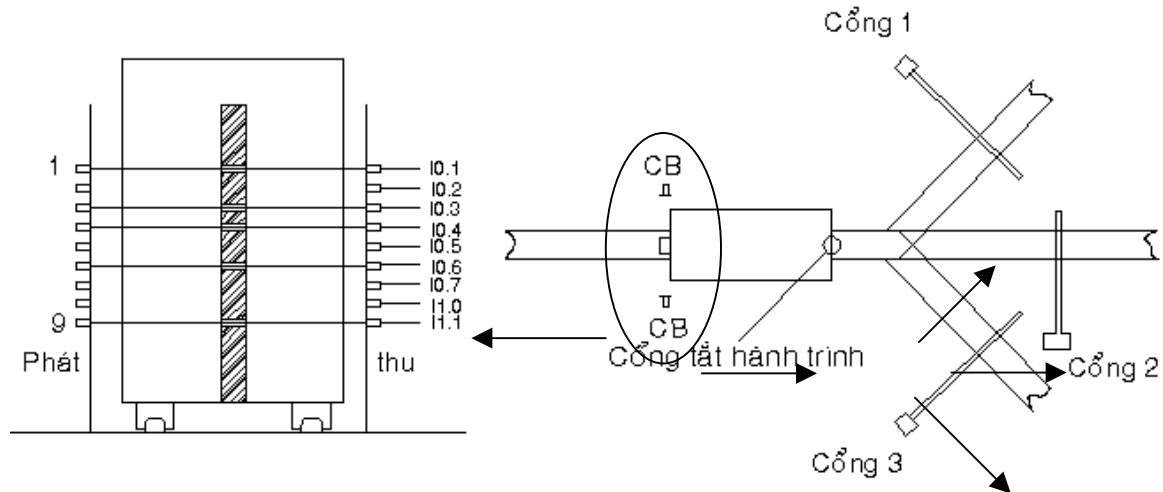
Ví dụ về việc thực hiện lệnh A ( And ), O ( Or ) và OLD theo LAD:



Hình 4.2

**Bài tập ứng dụng:****Bài 1:**

Một hệ thống phân loại xe chở hàng đơn giản trong nhà máy như sau:



I0.0: Công tắc hành trình

Q0.0: Mở cổng 1, Q0.1: Đóng cổng 1, Q0.2: Mở cổng 2, Q0.3: Đóng cổng 2, Q0.4: Mở cổng 3, Q0.5: Đóng cổng 3

Các xe sẽ cùng đi trên một ray chính sau đó tùy loại xe sẽ cho phép rẽ vào các đường khác nhau.

Sau mỗi xe có một thanh dọc có khoét lỗ (tương ứng với số). Khi tia laser (mức thấp) chiếu qua lỗ thì ngõ tương ứng sẽ lên 1. Theo hình vẽ ta sẽ có các ngõ: I0.1, I0.3, I0.4, I0.6, I1.1 sẽ lên 1 (được kích) tức xe có số 13469.

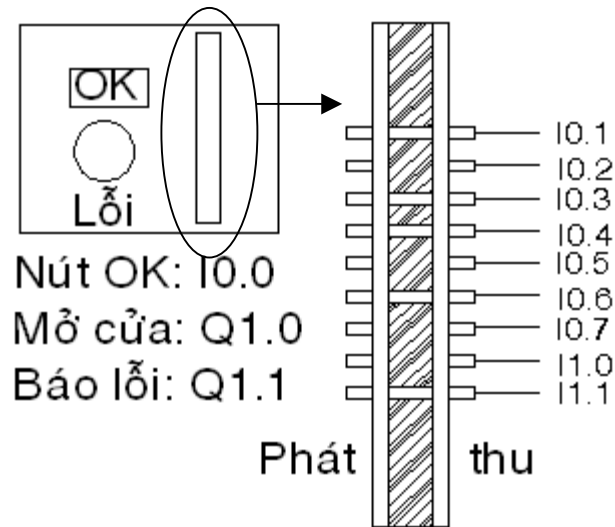
Khi xe chạy đến chạm vào công tắc hành trình (I0.0) thì PLC sẽ bắt đầu đọc mã. Tùy loại mã nhận được sẽ mở cổng tương ứng trong 5s rồi đóng cổng lại.

Mã 12579: cổng 1, mã 23679: cổng 2, mã 13689: cổng 3.

Viết chương trình điều khiển hệ thống. (Dùng PLC S7-300)

**Bài 2:**

Một hệ thống đọc mã thẻ đơn giản có cấu tạo như sau:



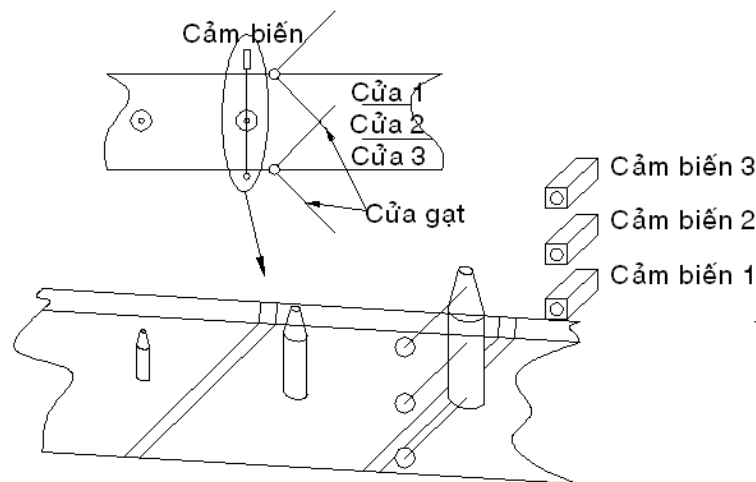
Trên thẻ có khắc lỗ (tương ứng với số). Khi ánh sáng hồng ngoại chiếu qua lỗ thì ngõ tương ứng sẽ lên 1. Theo hình vẽ ta sẽ có các ngõ: I0.1, I0.3, I0.4, I0.6, I1.1 sẽ lên 1 (được kích) tức thẻ có số 13469.

Khi chèn thẻ vào, nhấn nút OK, nếu đúng mã thì mở cửa (Q1.0) 5s rồi đóng lại, nếu sai sẽ bật đèn báo lỗi (Q1.1).

Viết chương trình để hệ thống chỉ nhận dạng 3 loại thẻ sau: 12579, 23679, 13689. (Dùng PLC S7-300)

**Bài 3:**

Một hệ thống phân loại sản phẩm có cấu tạo như sau:



Hệ thống sẽ phân ra 3 loại chạy theo 3 chiều cao khác nhau do 3 cảm biến quang xác định.

Loại 1 (Cao nhất, cả 3 cảm biến điều lên mức 1): Sẽ đi theo đường 1.

Loại 2 (Cao thứ 2, cảm biến 1 và 2 sẽ lên mức 1, cảm biến 3 ở mức 0): Sẽ đi theo đường 2.

Loại 3 (Thấp nhất, chỉ có cảm biến 1 lên mức 1, cảm biến 2 và 3 ở mức 0): Sẽ đi theo đường 3.

Việc chọn đường đi do vị trí của cửa gạt quyết định.

Ngõ vào → Start: I0.0, Stop: I0.1, CB 1: I0.2, CB 2: I0.3, CB 3: I0.4.

Ngõ ra → Cửa mở sang 1: Q0.0, Cửa mở sang 3: Q0.1.

*Chú ý: Cảm biến quang khi bị chắn ngang thì sẽ lên mức 1. Khoá lẩn khi điều khiển cửa gạt. Cửa ở vị trí 2 khi Q0.0 và Q0.1 ở mức 0.*

#### 4.4 TIMER

Timer là bộ tạo thời gian trễ giữa tín hiệu vào và tín hiệu ra nên trong điều khiển thường được gọi là khâu trễ. Các công việc điều khiển cần nhiều chức năng Timer khác nhau. Một Word (16bit) trong vùng dữ liệu được gán cho một trong các Timer.

Một Timer có các ngõ vào và ngõ ra tương ứng như sau:

Ngõ vào Start (bắt đầu): Timer được bắt đầu với sự thay đổi tín hiệu từ mức “0” lên mức “1” ở ngõ vào Start của nó. Thời gian (thí dụ L S5T#1S) và hoạt động của Timer (thí dụ SP T1) phải được lập trình ngay sau hoạt động quét điều kiện bắt đầu (thí dụ A I0.0).

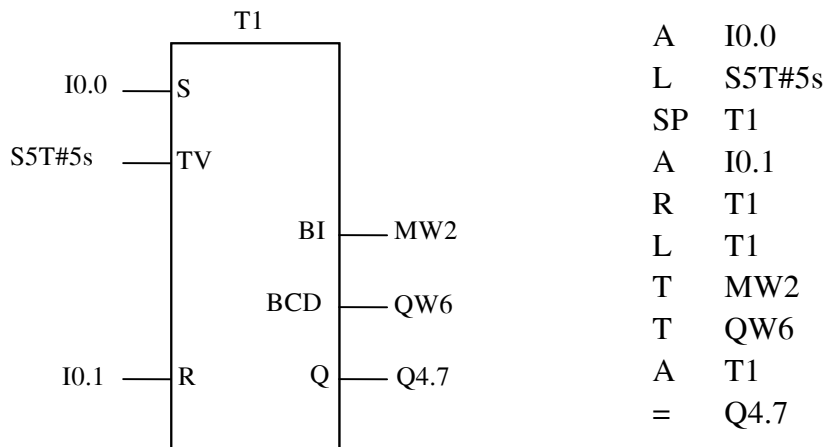
Ngõ vào Reset (xóa): tín hiệu mức “1” ở ngõ vào Reset làm dừng Timer. Lúc này thời gian hiện hành được đặt về 0 và ngõ ra Q của timer được xoá về “0”.

Các ngõ ra số: giá trị thời gian thực sự có thể đọc được từ hai ngõ ra số BI (số nhị phân) và BCD (số thập phân). Ví dụ xuất ra hiển thị dạng số ở ngõ ra.

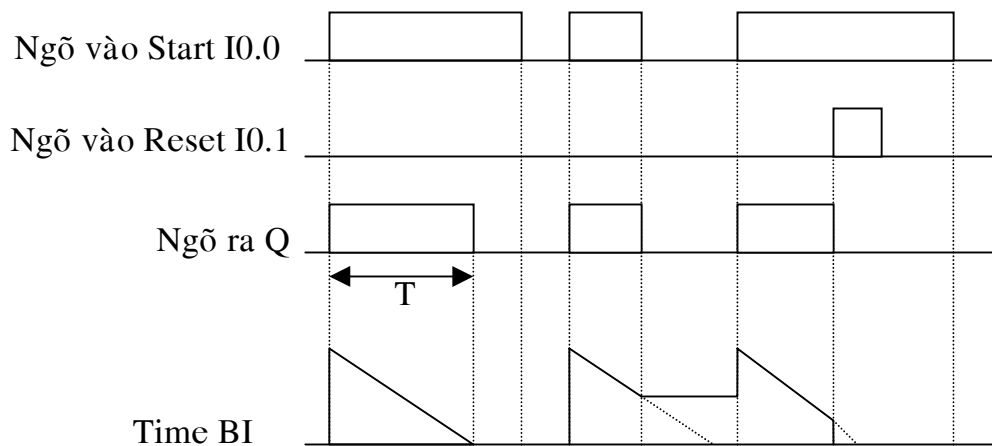
Ngõ ra nhị phân: trạng thái tín hiệu ở ngõ ra nhị phân Q của Timer phụ thuộc vào chức năng Timer được lập trình. Thí dụ khi bắt đầu, ngõ ra Q ở mức “1” khi có tín hiệu Start và Timer đang chạy.

Thí dụ:

Chương trình và giản đồ định thì của bộ định thì xung ( pulse Timer ):



Giải đồ định thì:

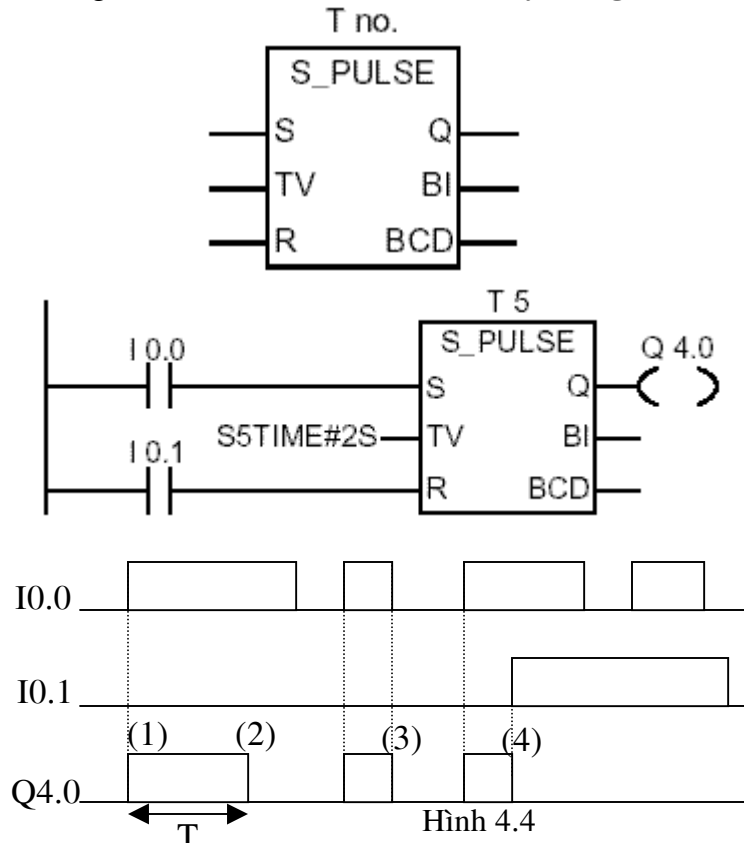


Hình 4.3

S7-300 có từ 128 Timer được chia làm nhiều loại khác nhau: Định thời xung (Pulse Timer), định thời xung mở rộng (extended-pulse Timer), định thời ON trễ (ON delay Timer), định thời gian ON trễ có chốt (latching ON delay Timer) và định thời OFF trễ (OFF delay Timer).

4.4.1 Pulse Timer (SP):

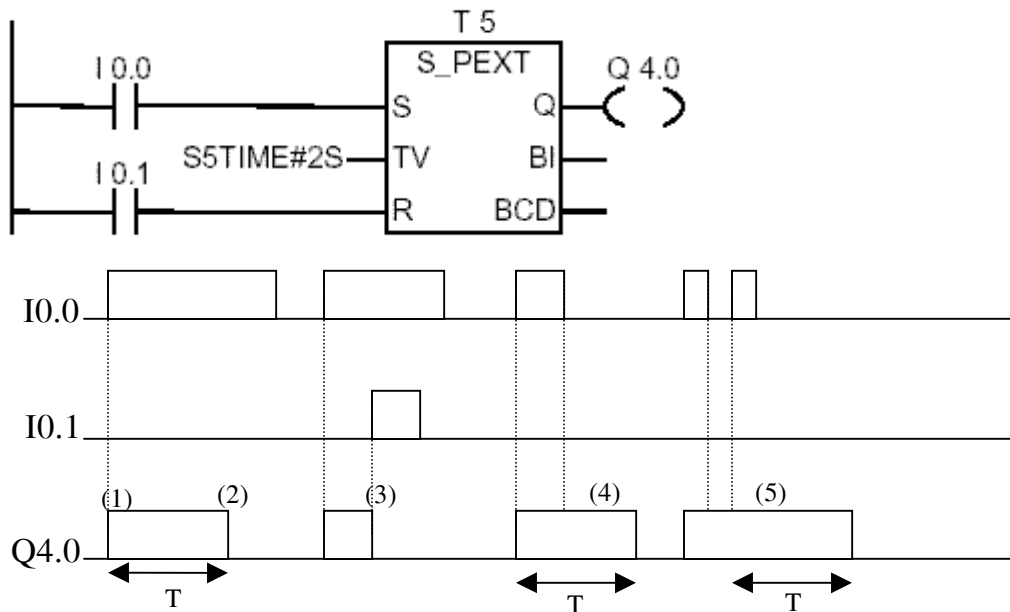
Ngõ ra của “pulse Timer” là “1” sau khi Timer được bắt đầu (1). Ngõ ra bị Reset nếu quá thời gian lập trình (2), nếu tín hiệu Start bị reset về “0” (3) hay nếu có một tín hiệu “1” đưa vào ngõ Reset của Timer (4). **Phải duy trì ngõ S**



Hình 4.4

#### 4.4.2 Extended pulse Timer (SE):

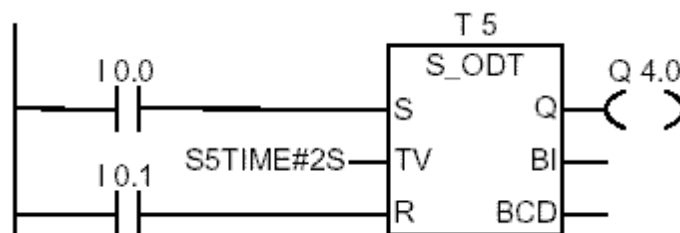
Ngõ ra của Extended Pulse Timer là “1” sau khi Timer được bắt đầu (1). Ngõ ra bị reset nếu quá thời gian được lập trình (2), hoặc ngõ vào Reset bị tác động. Việc reset ngõ vào Start trong quá trình Timer đang chạy (4) không làm cho ngõ ra bị reset. Nếu sự thay đổi tín hiệu “1” được lập lại trong quá trình Timer đang chạy thì Timer được bắt đầu lại, nghĩa là được kích trở lại (5). **Không cần duy trì ngõ S**



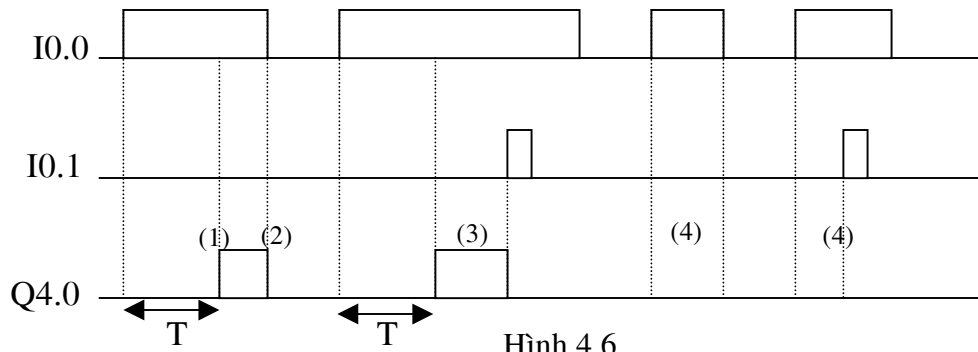
Hình 4.5

#### 4.4.3 On delay Timer (SD):

Ngõ ra On Delay Timer là “1” nếu quá thời gian được lập trình, và ngõ vào Start vẫn còn ở mức “1” (1). Kết quả là việc đặt ngõ vào Start lên “1” làm cho ngõ ra Q sẽ được đặt lên “1” với thời gian trì hoãn tương ứng đã được lập trình. Ngõ ra bị reset nếu ngõ vào Start bị reset(2) hoặc nếu có tín hiệu mức “1” ở ngõ vào Reset của Timer(3). Việc reset ngõ vào Start hoặc đưa “1” vào ngõ vào Reset của Timer trong khi Timer đang chạy (4) không làm cho ngõ ra đặt lên mức “1”. **Phải duy trì ngõ S**



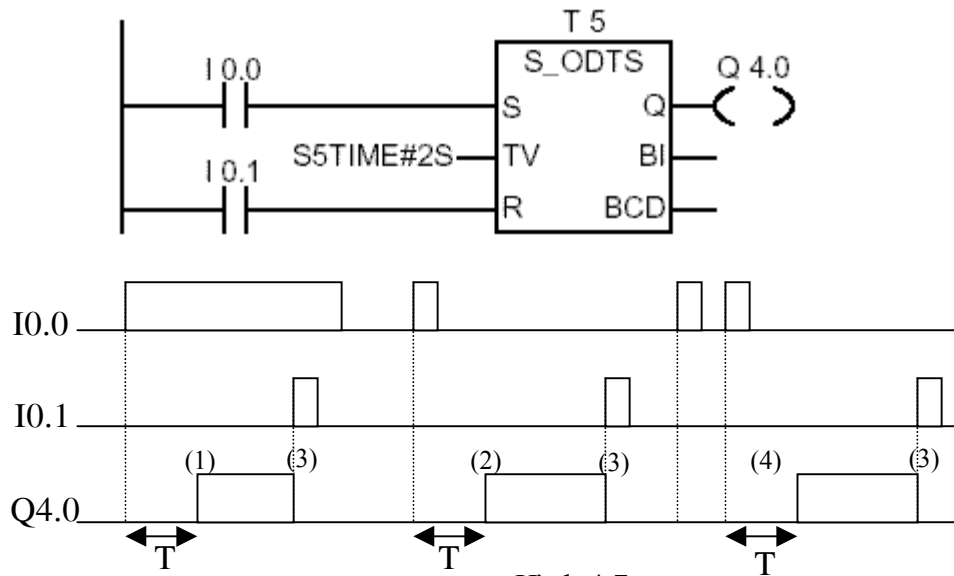




Hình 4.6

**4.4.4 Latching ON delay Timer (SS): (On delay không cần duy trì)**

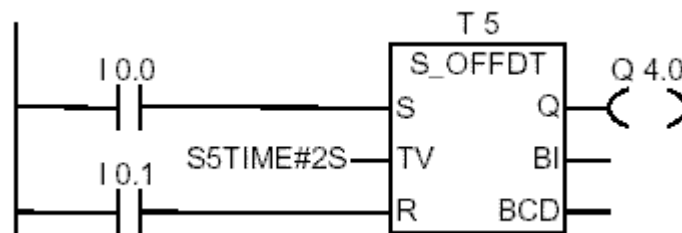
Ngõ ra của SS là “1” nếu vượt quá thời gian được lập trình (1). Ngõ ra Q của Timer vẫn giữ mức “1” (được chốt) ngay cả ngõ vào bị reset trong khi Timer đang chạy (2). Ngõ ra chỉ bị reset khi ngõ vào Reset của Timer bị tác động (3). Việc set và reset tiếp theo của ngõ vào Start trong khi Timer đang chạy chỉ được thực hiện khi nó bắt đầu được kích lại (4).

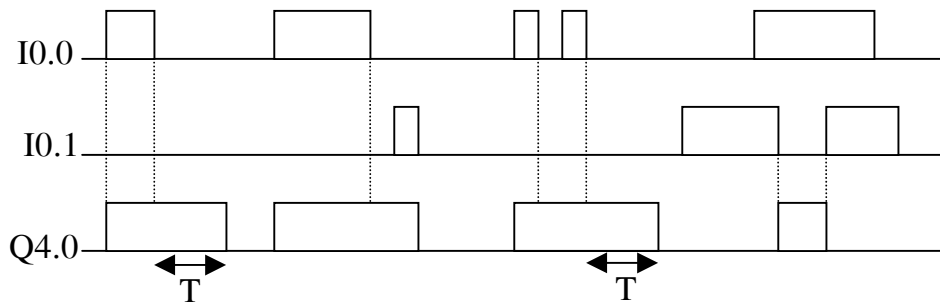


Hình 4.7

**4.4.5 OFF delay Timer (SF):**

Ngõ ra Q của SF được đặt lên mức “1” nếu có sự thay đổi tín hiệu từ “0” lên “1” ở ngõ vào Start. Nếu ngõ vào Start bị reset, ngõ ra vẫn giữ cho đến khi quá thời gian lập trình (2).





Hình 4.8

**Bài tập ứng dụng:**

Đèn 1: Q0.1    Đèn 2: Q0.2    Đèn 3: Q0.3

Start: I0.0, Stop: I0.1

Viết chương trình điều khiển 3 đèn theo trình tự:

Start → Đèn 1 sáng 1s → Đèn 2 sáng 1s → Đèn 3 sáng 1s → Đèn 1 và 3 sáng 2s →  
Đèn 2 sáng 2s → Lặp lại.

Stop → Dừng chương trình.

**4.5 COUNTER**

Trong công nghiệp, bộ đếm rất cần cho các quá trình đếm khác nhau như: đếm số chai, đếm xe hơi, đếm số chi tiết, ...

Một word 16bit (counter word) được lưu trữ trong vùng bộ nhớ dữ liệu hệ thống của PLC dùng cho mỗi counter. Số đếm được chứa trong vùng nhớ dữ liệu hệ thống dưới dạng nhị phân và có giá trị trong khoảng 0 đến 999.

Các phát biểu dùng để lập trình cho bộ đếm có các chức năng như sau:

**Đếm lên (CU = Counting Up):** Tăng counter lên 1. Chức năng này chỉ được thực hiện nếu có một tín hiệu dương ( từ “0” chuyển sang “1” ) xảy ra ở ngõ vào CU. Một khi số đếm đạt đến giới hạn trên là 999 thì nó không được tăng nữa.

**Đếm xuống (CD = Counting Down):** Giảm counter đi 1. Chức năng này chỉ được thực hiện nếu có sự thay đổi tín hiệu dương ( từ “0” sang “1” ) ở ngõ vào CD. Một khi số đếm đạt đến giới hạn dưới 0 thì nó không còn giảm được nữa.

**Đặt counter ( S = Setting the counter):** Counter được đặt với giá trị được lập trình ở ngõ vào PV khi có cạnh lên ( có sự thay đổi từ mức “0” lên mức “1” ) ở ngõ vào S này. Chỉ có sự thay đổi mới từ “0” sang “1” ở ngõ vào S này mới đặt giá trị cho counter một lần nữa.

**Đặt số đếm cho Counter ( PV = Presetting Value ):** Số đếm PV là một word 16 bit ở dạng BCD. Các toán hạng sau có thể được sử dụng ở PV là:

Word IW, QW, MW,...

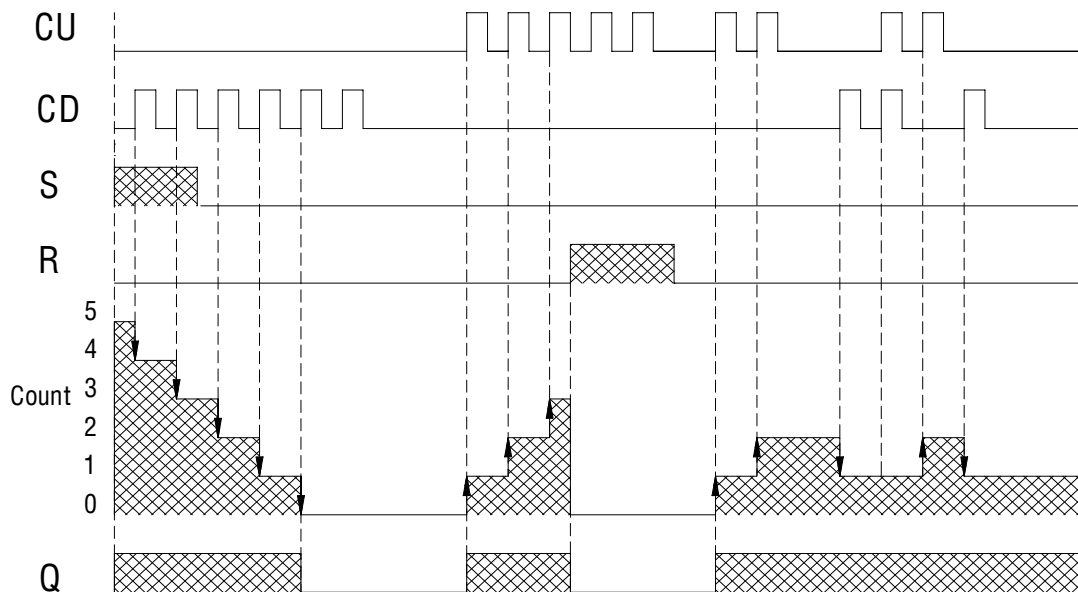
Hằng số: C#0,...,999

Xóa Counter ( R = Resetting the counter ): Counter được đặt về 0 (bị reset) nếu ở ngõ vào R có sự thay đổi tín hiệu từ mức “0” lên mức “1”. Nếu tín hiệu ở ngõ vào R là “0” thì không có gì ảnh hưởng đến bộ đếm.

Quét số của số đếm: (CV, CV\_BCD ): số đếm hiện hành có thể được nạp vào thanh ghi tích lũy ACCU như một số nhị phân (CV = Counter Value) hay số thập phân ( CV\_BCD ). Từ đó có thể chuyển các số đếm đến các vùng toán hạng khác.

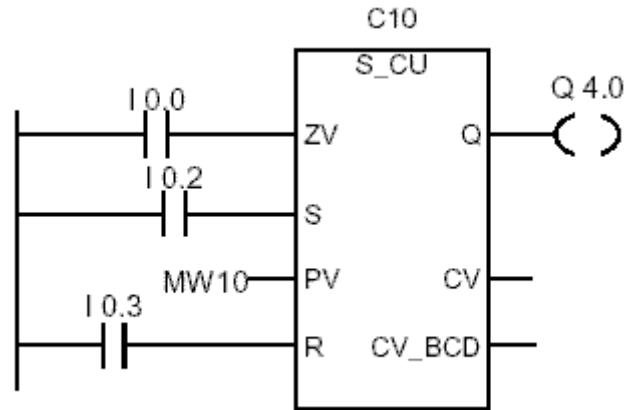
Quét nhị phân trạng thái tín hiệu của Counter (Q): ngõ ra Q của counter có thể được quét để lấy tín hiệu của nó. Nếu Q = “0” thì counter ở zero, nếu Q = “1” thì số đếm ở counter lớn hơn zero.

Biểu đồ chức năng:



Hình 4.9

### 4.5.1 Up counter



Hình 4.10

I0.2: đặt giá trị bắt đầu và cho phép Counter đếm.

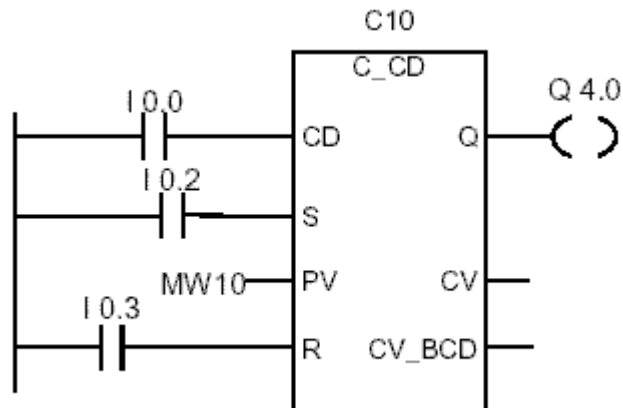
I0.0: Counter đếm lên

I0.3: Reset Counter

Q4.0 = 1 khi giá trị của Counter khác 0.

MW10: chứa giá trị bắt đầu đếm cho Timer.

### 4.5.2 Down counter



Hình 4.11

I0.2: đặt giá trị bắt đầu và cho phép Counter đếm.

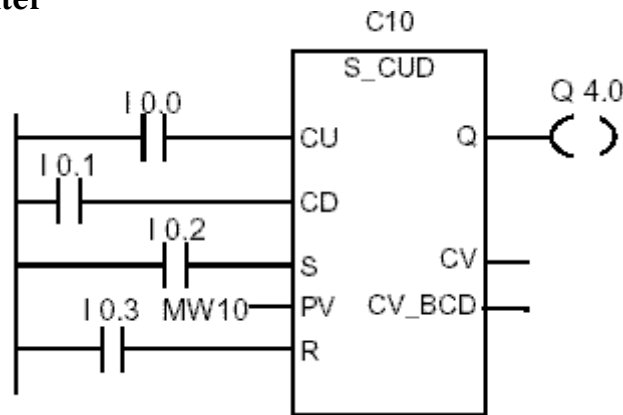
I0.0: Counter đếm xuống

I0.3: Reset Counter

Q4.0 = 1 khi giá trị của Counter khác 0.

MW10: chứa giá trị bắt đầu đếm cho Timer.

### 4.5.3 Up-Down Counter



Hình 4.12

I0.2: đặt giá trị bắt đầu và cho phép Counter đếm.

I0.0: Counter đếm lên

I0.1: Counter đếm xuống

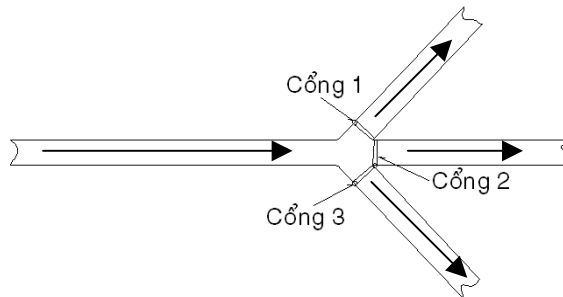
I0.3: Reset Counter

Q4.0 = 1 khi giá trị của Counter khác 0.

MW10: chứa giá trị bắt đầu đếm cho Timer.

#### **Bài tập ứng dụng:**

Một bầy gia súc 300 con, được phân ra 3 chuồng khác nhau, mỗi chuồng 100 con.



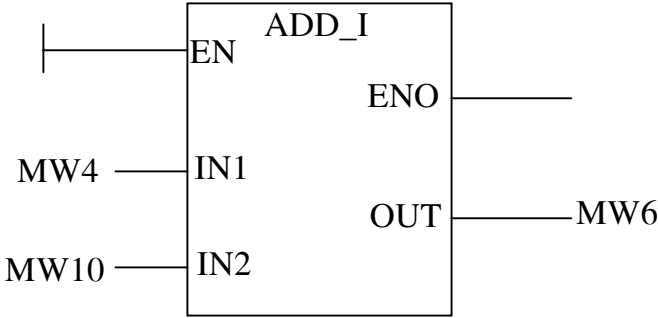
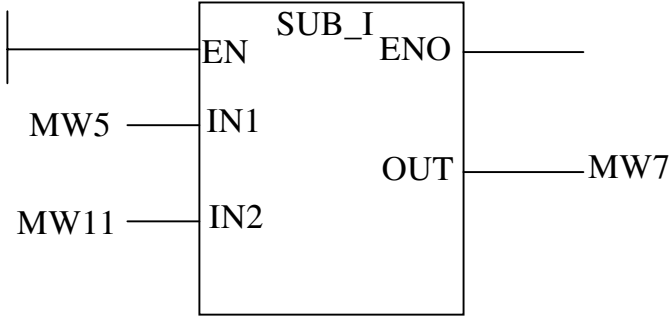
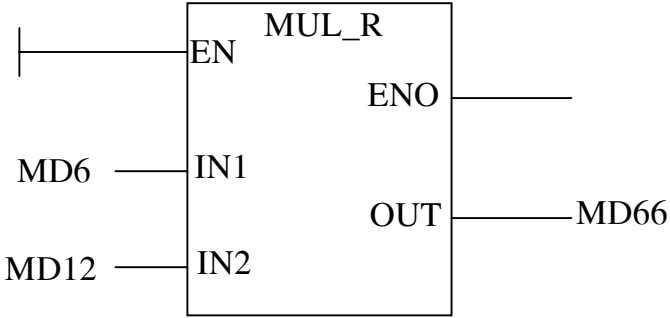
Gia súc sẽ đi theo một đường chung sao đó sẽ phân ra mỗi chuồng 100 con.

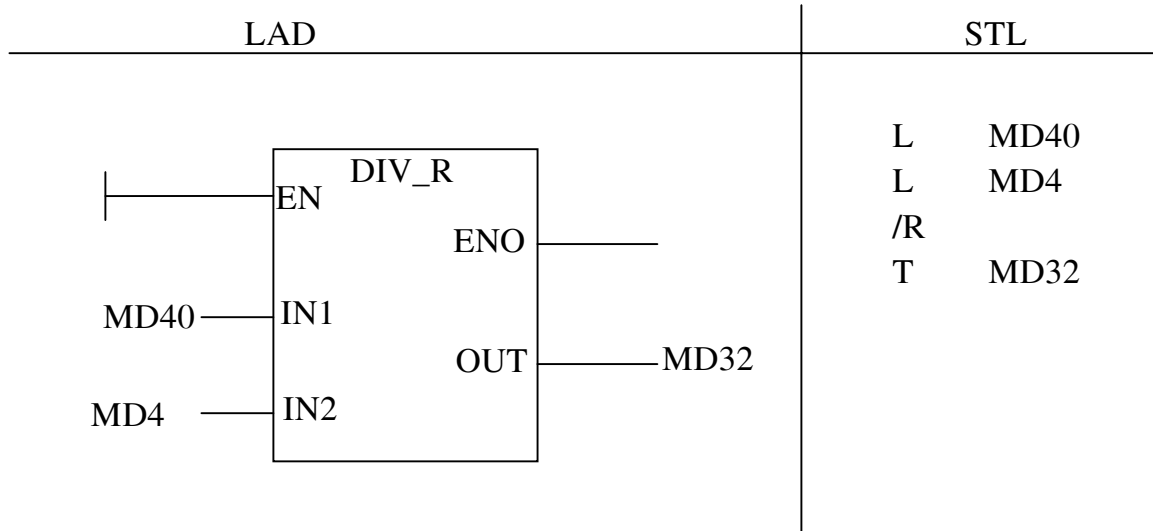
Nhấn Start → Mở cổng 1 cho gia súc vào (100 con) → đóng cổng 1, mở cổng 2 (100 con) → đóng cổng 2, mở cổng 3 (100 con) → đóng cổng 3.

Hãy giúp nông trại:

- Thiết kế phần cứng cho hệ thống điều khiển
- Viết chương trình điều khiển (dùng PLC S7-300)

## 4.6 LỆNH TOÁN HỌC CƠ BẢN

LAD	STL
	<pre> L   MW4 L   MW10 +I T   MW6 </pre>
	<pre> L   MW5 L   MW11 -I T   MW7 </pre>
	<pre> L   MD6 L   MD12 *R T   MD66 </pre>



S7 – 300 có nhiều lệnh cho phép tính toán số học. Tất cả những câu lệnh có cùng một định dạng.

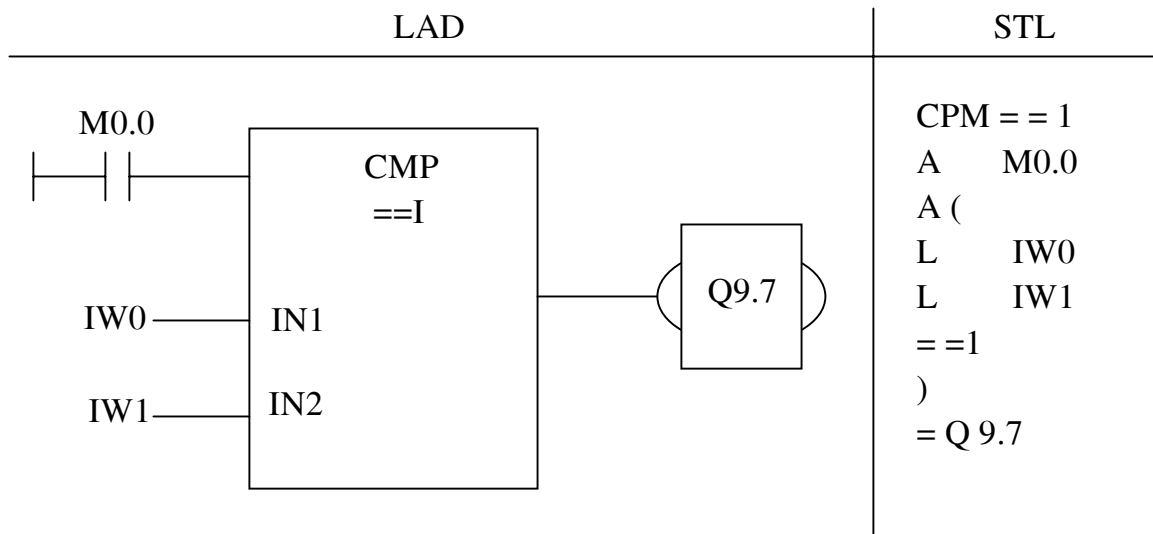
- EN      Lệnh được thực hiện nếu có sự thay đổi tín hiệu từ mức “0” lên mức “1” ở ngõ vào EN
- ENO     Nếu kết quả nằm ngoài phạm vi cho phép của loại dữ liệu tương ứng thì cờ tràn (bit tràn) OV và cờ tràn có nhớ (bit tràn có nhớ) OS sẽ được set lên “1” và ENO = “0”. Qua đó các phép tính tiếp theo qua ENO sẽ không được thực hiện.
- IN1, IN2   Giá trị tại IN1 được đọc vào như toán tử thứ nhất và giá trị tại IN2 được đọc vào như toán tử thứ 2. (Chú ý sự tương thích của kiểu dữ liệu và kích thước ô nhớ)
- OUT      Kết quả của phép tính toán học được lưu tại ngõ ra out. (Chú ý sự tương thích của kiểu dữ liệu và kích thước ô nhớ)

#### Các câu lệnh:

Cộng	ADD_I	Cộng số nguyên
	ADD_DI	Cộng số nguyên kép
	ADD_R	Cộng số nguyên thực
Trừ	SUB_I	Trừ số nguyên
	SUB_DI	Trừ số nguyên kép
	SUB_R	Trừ số thực
Nhân	MUL_I	Nhân số nguyên
	MUL_DI	Nhân số nguyên kép
	MUL_R	Nhân số thực
Chia	DIV_I	Chia số nguyên
	DIV_DI	Chia số nguyên kép
	DIV_R	Chia số thực

## 4.7 LỆNH XỬ LÝ DỮ LIỆU

### 4.7.1 Lệnh So Sánh



Có thể dùng lệnh so sánh để so sánh các cặp giá trị số sau:

- I: So sánh những số nguyên ( dựa trên cơ sở số **16bit**)
- D: So sánh những số nguyên ( dựa trên cơ sở số **32bit**)
- R: So sánh những số thực ( dựa trên cơ sở số thực **32bit**).

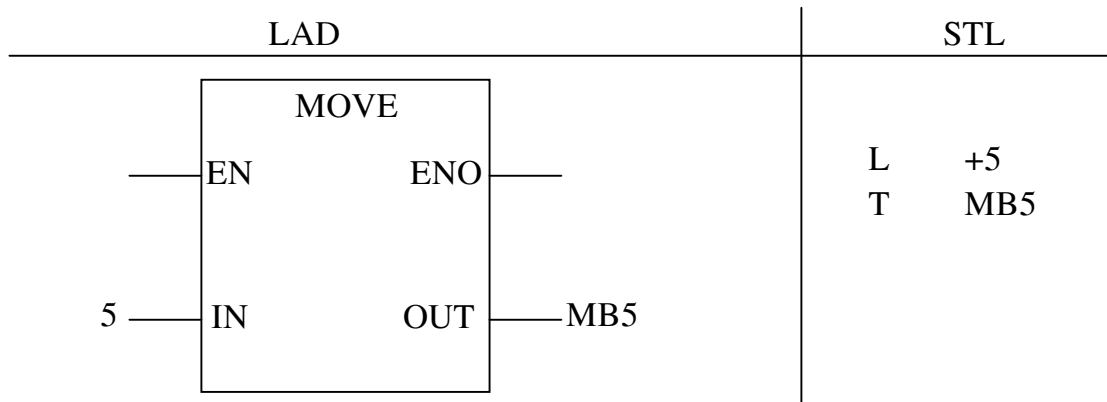
Nếu kết quả so sánh là TRUE thì ngõ ra của phép toán là “1” ngược lại ngõ ra của phép toán là “0”.

Sự so sánh ở ngõ ra và ngõ vào tương ứng với các loại sau:

- == (I, D, R) IN1 bằng IN2
- <> (I, D, R) IN1 không bằng IN2
- > (I, D, R) IN1 lớn hơn IN2
- < (I, D, R) IN1 nhỏ hơn IN2
- >= (I, D, R) IN1 lớn hơn hoặc bằng IN2
- <= (I, D, R) IN1 nhỏ hơn hoặc bằng IN2.



### 4.7.2 Lệnh nạp và truyền dữ liệu

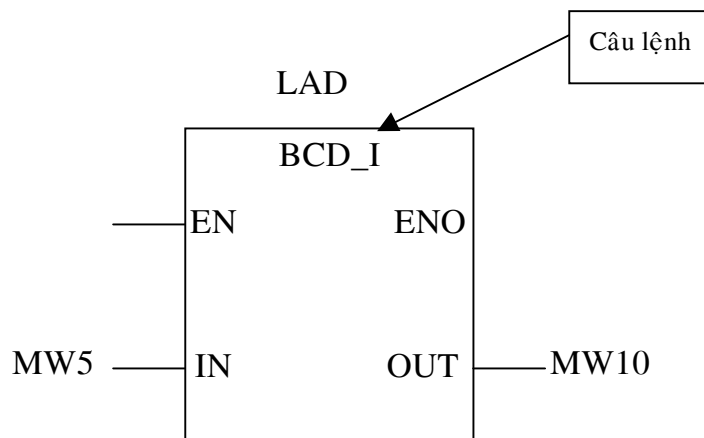


Khi có tín hiệu EN thì lệnh sẽ chuyển giá trị ở ngõ vào IN vào ô nhớ ở ngõ ra OUT. Ngõ vào IN có thể là số hoặc ô nhớ, ngõ ra OUT chỉ có thể là ô nhớ. Kiểu dữ liệu giữa ngõ IN và ngõ OUT phải tương thích nhau. Ví dụ

Nếu ngõ vào là MW thì ngõ ra cũng phải là MW hoặc MD

Nếu ngõ vào là số nguyên thì ngõ ra phải là MW hoặc MD.

### 4.7.3 Các lệnh chuyển đổi dữ liệu



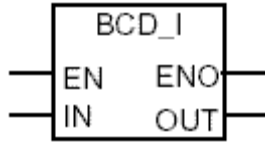
Hình 4.13

S7 – 300 có nhiều lệnh cho phép chuyển đổi các kiểu dữ liệu. Tất cả những câu lệnh có cùng một định dạng.

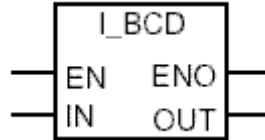
- EN      Lệnh được thực hiện nếu có sự thay đổi tín hiệu từ mức “0” lên mức “1” ở ngõ vào EN
- ENO     Lên 1 nếu phép chuyển đổi được thực hiện.
- IN      Dữ liệu cần chuyển đổi . Có thể là hằng hoặc ô nhớ (phải tương thích kiểu dữ liệu và kích thước ô nhớ) (I, Q, M, Const, D, L...)
- OUT     Kết quả của phép chuyển đổi được lưu tại ngõ ra out. Chỉ có thể là ô nhớ (phải tương thích kiểu dữ liệu và kích thước ô nhớ). (I, Q, M, D, L...)

**Các câu lệnh**

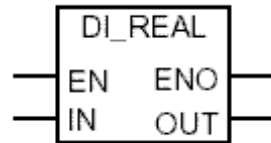
**BCD\_I:** Chuyển đổi số nhị phân thập phân 16 bit thành số nguyên 16 bit và kết quả ghi vào OUT .



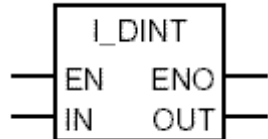
**I\_BCD:** Chuyển đổi số nguyên 16 bit IN thành số nhị phân thập phân 16 bit và kết quả ghi vào OUT.



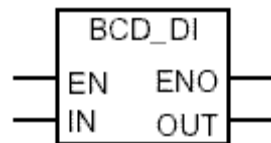
**DI\_REAL:** Chuyển đổi số nguyên 32 bit có dấu IN thành số thực 32 bit và ghi kết quả vào OUT.



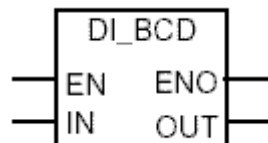
**I\_DINT:** Chuyển đổi số nguyên 16 bit thành số nguyên 32 bit và ghi kết quả vào OUT.



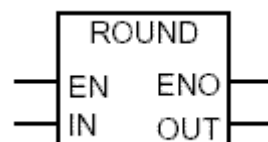
**BCD\_DI:** Chuyển đổi số BCD thành số nguyên 32 bit và ghi kết quả vào OUT.



**DI\_BCD:** Chuyển đổi số nguyên 32 bit thành số BCD và ghi kết quả vào OUT.



Làm tròn giá trị ngõ vào thành số nguyên và ghi kết quả vào OUT.



## Chương 5: Ngôn ngữ lập trình Step7

### 5.1 CÀI ĐẶT STEP 7

Cấu hình phần cứng

Để cài đặt STEP 7 yêu cầu tối thiểu cấu hình như sau:

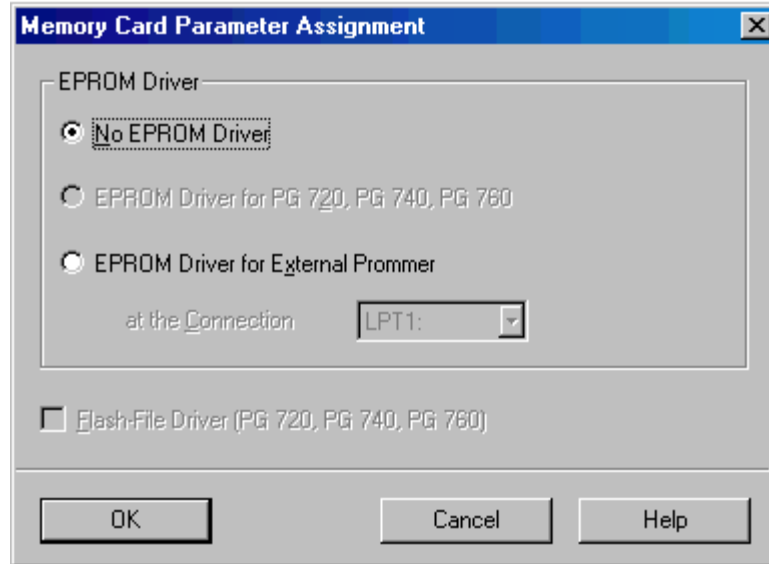
- 80486 hay cao hơn, đề nghị Pentium
- Đĩa cứng trống: Tối thiểu 300MB
- RAM: > 32MB, đề nghị 64MB
- Giao tiếp: CP5611, MPI card hay tiếp hợp PC để lập trình với mạch nhớ
- Mouse: Có
- Hệ điều hành: Windows 95/ 98/ NT

Có nhiều phiên bản của bộ phần mềm gốc của STEP 7 hiện có tại Việt Nam. Đang được sử dụng nhiều nhất là phiên bản 4.2 và 5.0. Trong khi phiên bản 4.2 khá phù hợp với những PC có cấu hình trung bình nhưng lại đòi hỏi phải tuyệt đối có bản quyền thì phiên bản 5.0, đòi hỏi cấu hình PC phải mạnh tốc độ cao, có thể chạy ở chế độ không cài bản quyền (ở mức hạn chế).

Phần lớn các đĩa gốc của STEP 7 đều có khả năng tự thực hiện chương trình cài đặt (autorun). Bởi vậy ta chỉ cần bỏ đĩa vào và thực hiện theo những chỉ dẫn. Ta cũng có thể chủ động thực hiện cài đặt bằng cách gọi chương trình **setup.exe** có trên đĩa. Công việc cài đặt STEP 7 nói chung không khác gì nhiều so với việc cài đặt các phần mềm ứng dụng khác như Windows, Office...

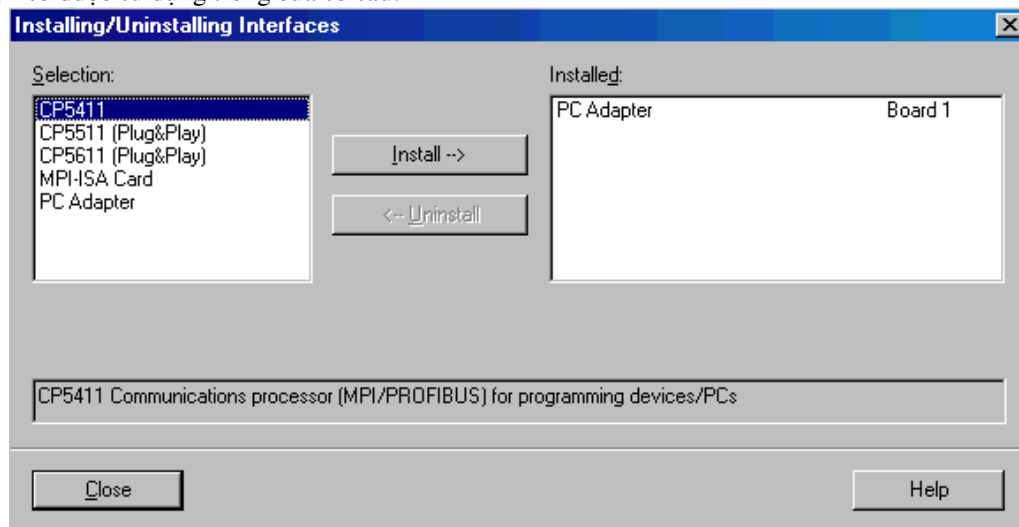
Tuy nhiên, so với các phần mềm khác thì việc cài đặt STEP 7 sẽ có vài điểm khác biệt cần được giải thích rõ thêm:

- **Khai báo mã hiệu sản phẩm:** Mã hiệu sản phẩm luôn đi kèm theo phần mềm STEP 7 và in ngay trên đĩa chứa bộ cài STEP 7. Khi trên màn hình hiện ra cửa sổ yêu cầu cho biết mã hiệu sản phẩm, ta điền đầy đủ vào tất cả các mục trong ô cửa sổ đó thì mới có thể tiếp tục cài đặt phần mềm.
- **Đăng ký bản quyền:** Bản quyền của STEP 7 nằm trên một đĩa mềm riêng (thường có màu vàng hoặc đỏ). Ta có thể cài đặt bản quyền trong quá trình cài đặt hay sau khi cài đặt phần mềm xong thì chạy chương trình đăng ký **AuthorsW.exe** có trên đĩa CD cài đặt.
- **Khai báo thiết bị đốt EPROM:** Chương trình STEP 7 có khả năng đốt chương trình ứng dụng lên thẻ EPROM cho PLC. Nếu máy tính của ta có thiết bị đốt EPROM thì cần thông báo cho STEP 7 biết khi trên màn hình xuất hiện cửa sổ (hình dưới):



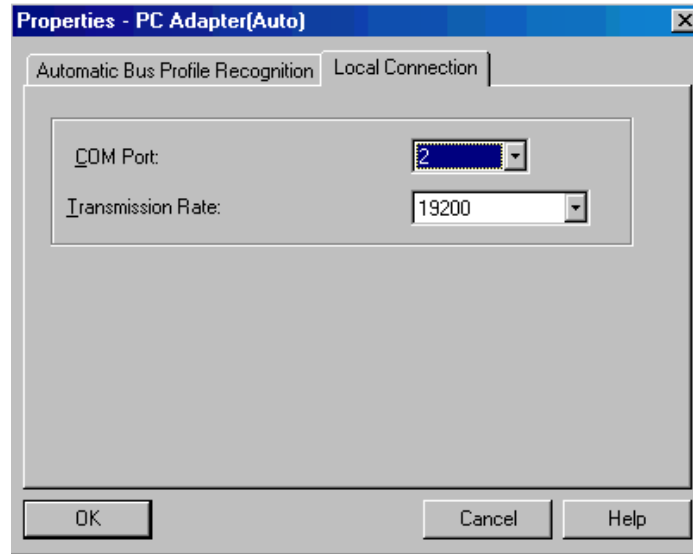
Hình 5.1 Cài đặt thiết bị đối EPROM

**Chọn giao diện PC/PLC:** Chương trình được cài đặt trên PG/PC để hỗ trợ việc soạn thảo cấu hình phần cứng cũng như chương trình cho PLC. Ngoài ra, STEP 7 còn có khả năng quan sát việc thực hiện chương trình của PLC. Muốn như vậy ta cần tạo bộ giao diện ghép nối giữa PC và PLC để truyền thông tin, dữ liệu. STEP 7 có thể được ghép nối giữa PC và PLC qua nhiều bộ giao diện khác nhau và ta có thể chọn giao diện sẽ được sử dụng trong cửa sổ sau:



Hình 5.2 Các bộ giao diện có thể chọn

Sau khi chọn bộ giao diện ta phải cài đặt tham số làm việc cho nó thông qua cửa sổ màn hình dưới đây khi chọn mục “**Set PG/PC Interface...**”.




Hình 5.3 Cài đặt thông số cho bộ giao diện

**Đặt tham số làm việc:**

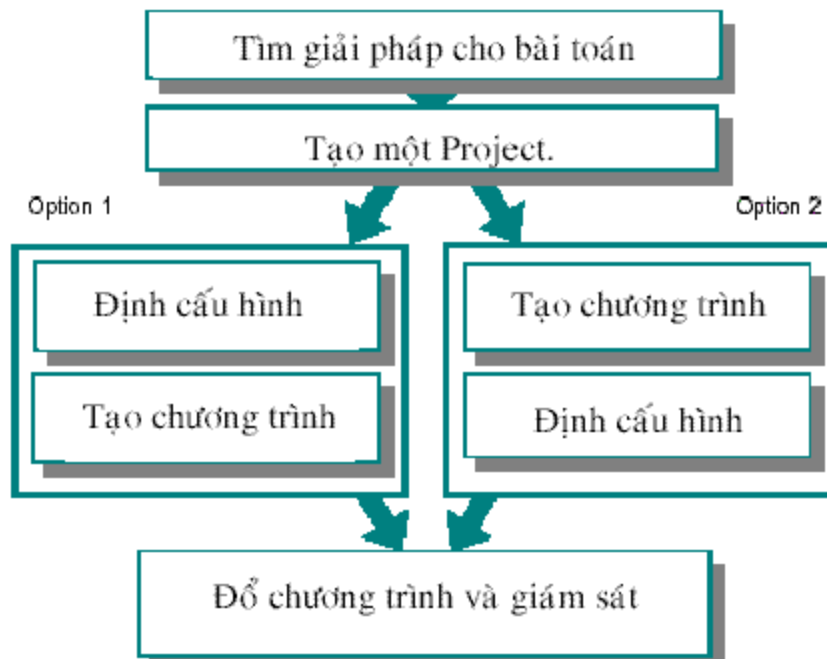
Sau khi cài đặt xong STEP 7, trên màn hình desktop sẽ xuất hiện biểu tượng của phần mềm STEP 7.



Hình 5.4 Biểu tượng của STEP 7

Đồng thời trong menu  của Windows cũng có thư mục Simatic với tất cả các tên của những thành phần liên quan, từ các phần mềm trợ giúp đến các phần mềm cài đặt cấu hình, chế độ làm việc của STEP 7 ....

**5.2 TRÌNH TỰ CÁC BƯỚC THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN**



### 5.3 KHỞI ĐỘNG CHƯƠNG TRÌNH TẠO PROJECT

Chương trình quản lý SIMATIC là giao diện đồ họa với người dùng bằng chương trình soạn thảo trực tuyến/ngoại tuyến đối tượng S7 (đề án, tập tin người dùng, khối, các trạm phần cứng và công cụ)

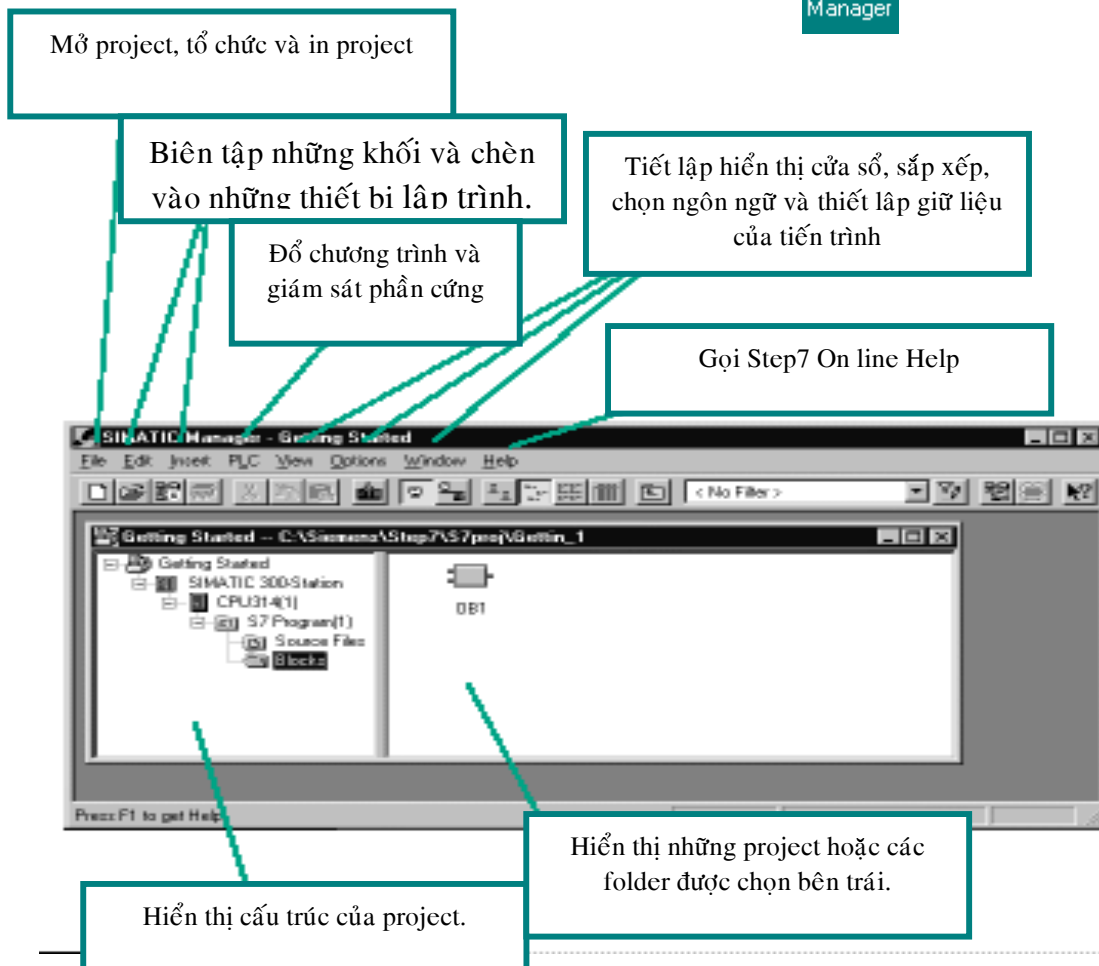
Với chương trình quản lý SIMATIC có thể:

- Quản lý đề án và thư viện
- Tác động công cụ của STEP 7
- Truy cập trực tuyến PLC
- Soạn thảo thẻ nhớ

Các công cụ của STEP 7 có ở trong SIMATIC Manager. Để khởi động có thể làm theo hai cách:

- Bằng *Task bar* → *Start* → *SIMATIC* → *STEP7* → *SIMATIC Manager*

- Nhấn kép vào biểu tượng SIMATIC Manager



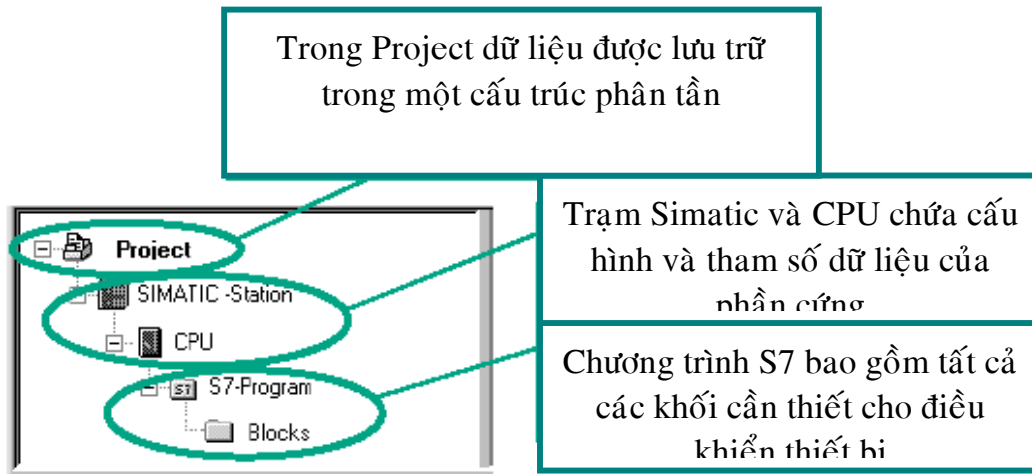
Hình 5.5 Các thành phần cửa sổ Manager

- **Thanh tiêu đề:**  
Thanh tiêu đề gồm cửa sổ và các nút để điều khiển cửa sổ
- **Thanh thực đơn:**  
Gồm các thực đơn cho các cửa sổ đang mở
- **Thanh công cụ**  
Gồm các thao tác thường dùng nhất dưới dạng ký hiệu. Những ký hiệu này có thể tự giải thích
- **Thanh trạng thái:**  
Hiện ra trạng thái hiện tại và nhiều thông tin khác
- **Thanh công tác**  
Chứa các ứng dụng đang mở và cửa sổ dưới dạng các nút. Thanh công tác có thể đặt 2 bên màn hình bằng cách nhấn chuột phải

**Thanh công cụ chương trình quản lý SIMATIC bao gồm:**

- |                                      |                     |
|--------------------------------------|---------------------|
| • New (File Menu)                    | Tạo mới             |
| • Open (File Menu)                   | Mở file             |
| • Display Accesible Nodes (PLC Menu) | Hiển thị các nút    |
| • S7 Memory Card (File Menu)         | Thẻ nhớ S7          |
| • Cut (Edit menu)                    | Cắt                 |
| • Paste (Edit Menu)                  | Dán                 |
| • Copy (Edit Menu)                   | Sao chép            |
| • Download (PLC Menu)                | Tải xuống           |
| • Online (View Menu)                 | Trực tuyến          |
| • Offline (View Menu)                | Ngoại tuyến         |
| • Large Icons (View Menu)            | Biểu tượng lớn      |
| • Small Icons (View Menu)            | Biểu tượng nhỏ      |
| • List (View Menu)                   | Liệt kê             |
| • Details (View Menu)                | Chi tiết            |
| • Up on level (View Menu)            | Lên một cấp         |
| • Simulate Modules (OptionMenu)      | Khởi mô phỏng       |
| • Help Symbol                        | Biểu tượng trợ giúp |

### 5.3 CẤU TRÚC PROJECT STEP7



Hình 5.6 Cấu trúc project step7


### 5.4 VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN

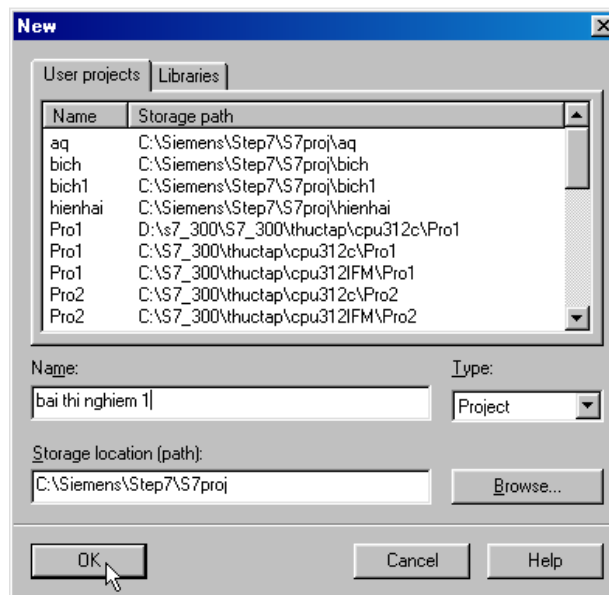
#### 5.4.1 Khai báo phần cứng

Ta phải xây dựng cấu hình phần cứng khi tạo một project. Dữ liệu về cấu hình sẽ được truyền đến PLC sau đó.

Ta sẽ thử khai báo phần cứng cho các Module sau:

CPU 312C-5BD01-0AB0, DI 321-1BH02-0AA0, DO 322-1HF01-0AA0, AI 331-7KB02-0AB0, AO 332-5H501-0AB0

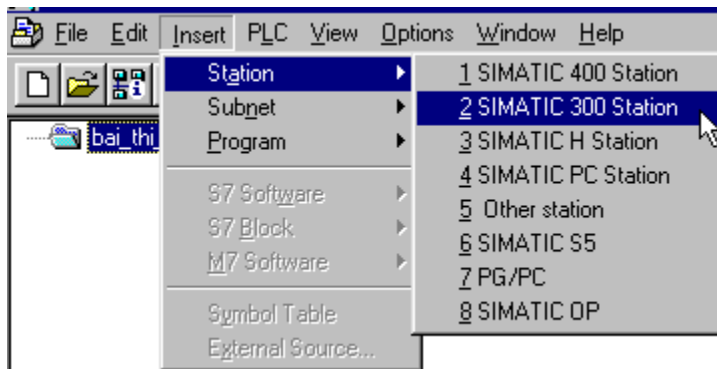
- Click vào biểu tượng  để mở chương trình mới. Khi cửa sổ New hiện ra, ta nhập tên của chương trình vào và Click OK như hình sau:



Hình 5.7

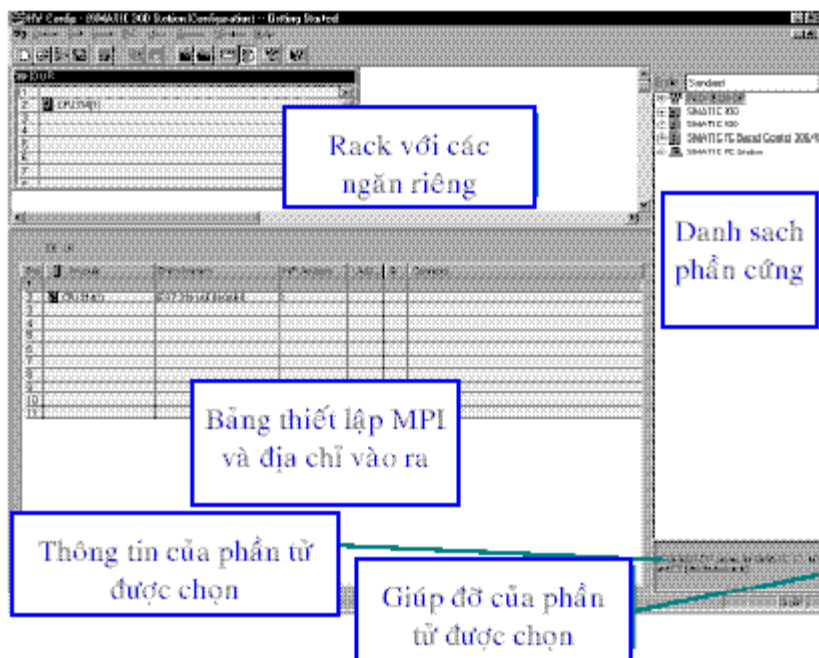


◆ Trở vào màn hình chính ta vào *Insert -> Station -> SIMATIC 300 Station* để chèn cấu hình cho chương trình (module CPU, module IM,...). Xem hình sau:



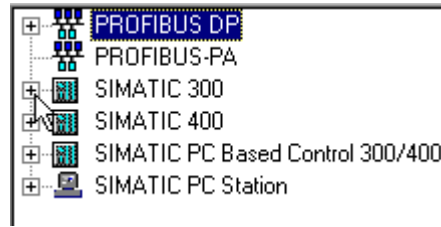
Hình 5.8

◆ Khi Click vào biểu tượng SIMATIC 300 bên phải màn hình xuất hiện biểu tượng Hardware. Ta D\_Click vào biểu tượng Hardware để khai báo cấu hình cho chương trình. Cửa sổ HW Config được mở (xem hình dưới), ta phải chèn rack cho project.



Hình 5.9 Cửa sổ khai báo Hardware

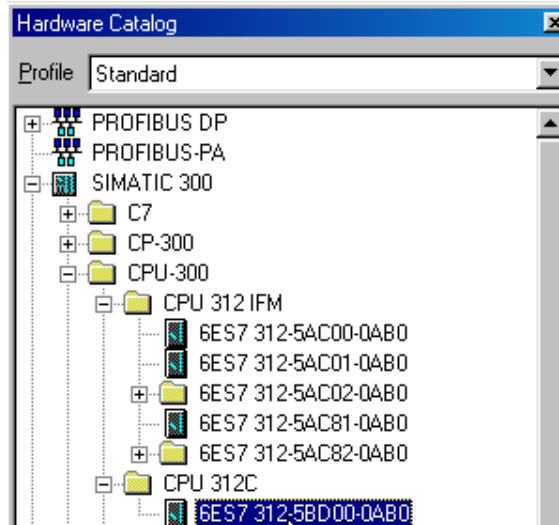
◆ Trong cửa sổ HW\_config ta Click vào biểu tượng để mở thư viện.  
 ◆ Trong thư viện, ta Click vào SIMATIC 300 (hình 15) để lấy các thành phần cần thiết.



\_Hình 5.10

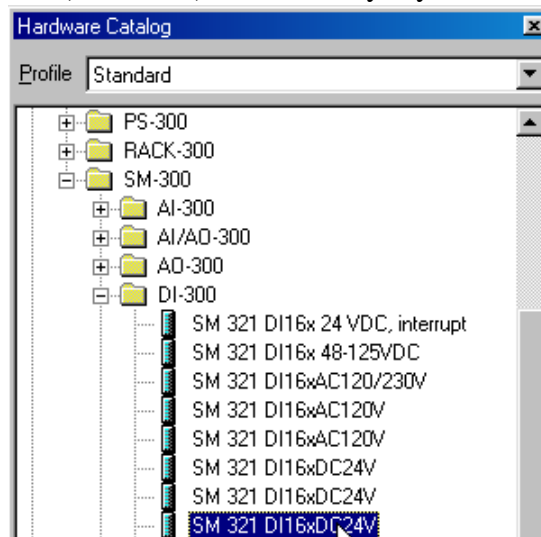
◆ Tiếp theo, ta tìm thư mục RACK 300 và D\_Click vào biểu tượng  để tạo Rail chứa các Module.

◆ Một Rail sẽ hiện ra gồm 11 Slot (xem hình). Ta Click vào Slot 2 (tô đậm Slot 2), sau đó Click vào *CPU\_300* -> *CPU\_312C* -> *6ES7 312-5BD00-0AB0*.



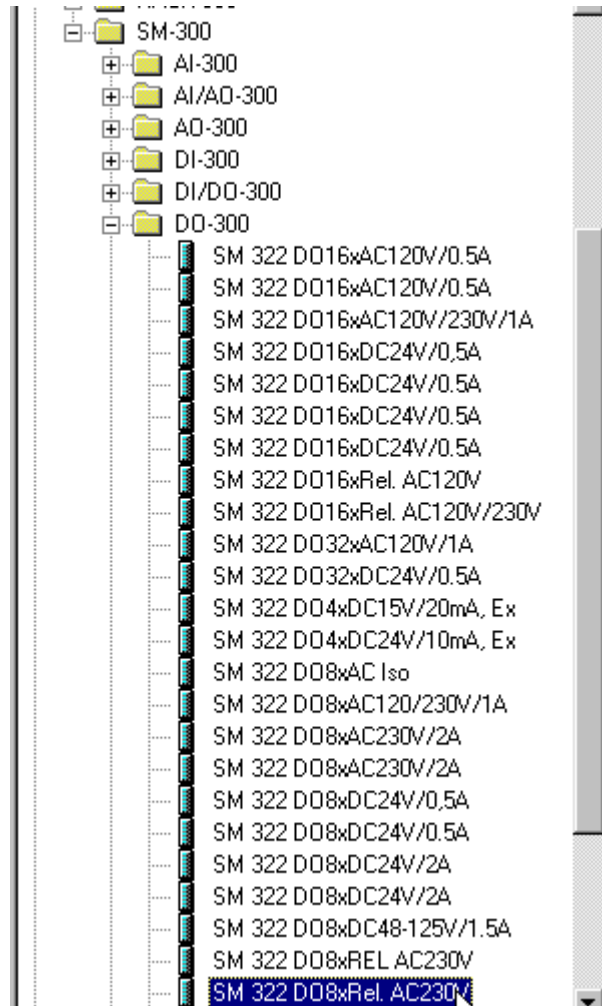
\_Hình 5.11

◆ Tiếp theo, Click vào Slot 4 và Click vào *SM\_300* -> *DI\_300* -> D\_Click vào *SM 321 DI16xDC24V* (hình 5.12). Số hiệu này tùy thuộc loại Module DI mà ta có.



Hình 5.12

- ◆ Tương tự, ta Click vào Slot 5 và Click vào DO\_300 -> D\_Click vào SM 322 DO16xDC24V/0.5A. (Hình 5.13):



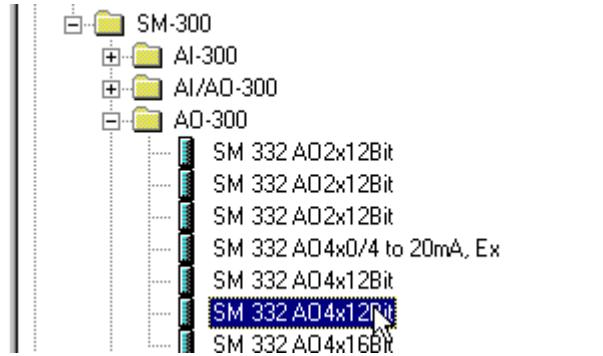
Hình 5.13

- ◆ Tương tự, ta Click vào Slot 6 và Click vào AI\_300 -> D\_Click vào SM 331 AI2x12Bit. (hình 5.14):




Hình 5.14

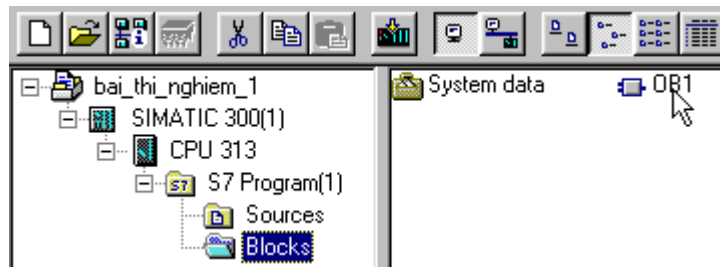
- ◆ Tương tự, ta Click vào Slot 7 và Click vào A0\_300 -> D\_Click vào SM 332 AO4 x 12Bit.(Hình 5.15):



Hình 5.15

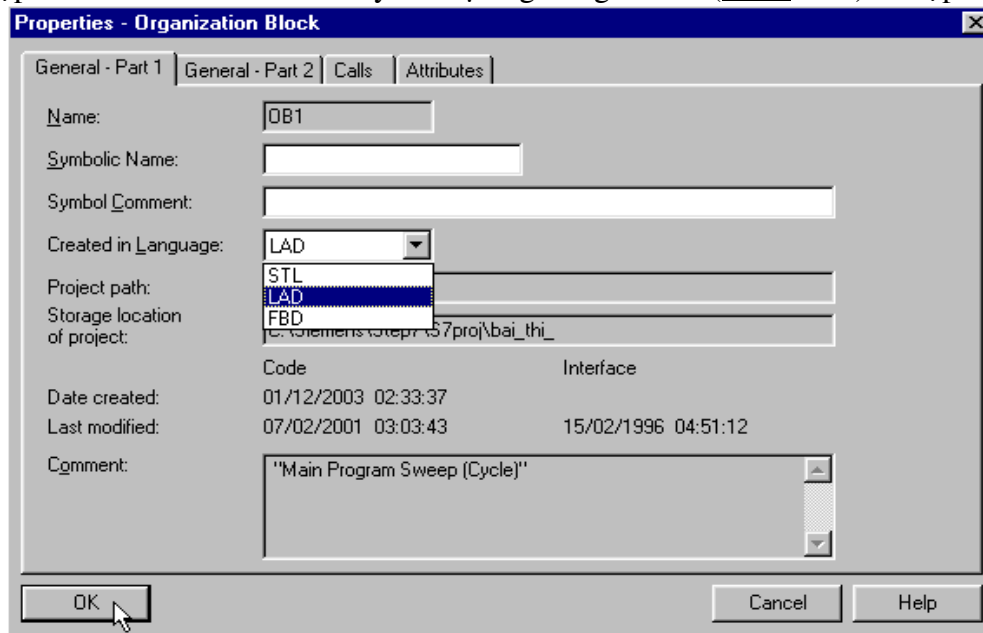
- ◆ Ta Click vào biểu tượng  để Save và Compile cấu hình cứng. Ta đóng cửa sổ HW\_Config để vào màn hình Manager.

Khi trở về màn hình Manager ta D\_Click vào biểu tượng khối OB1 (hình 5.16) để mở khối OB1.



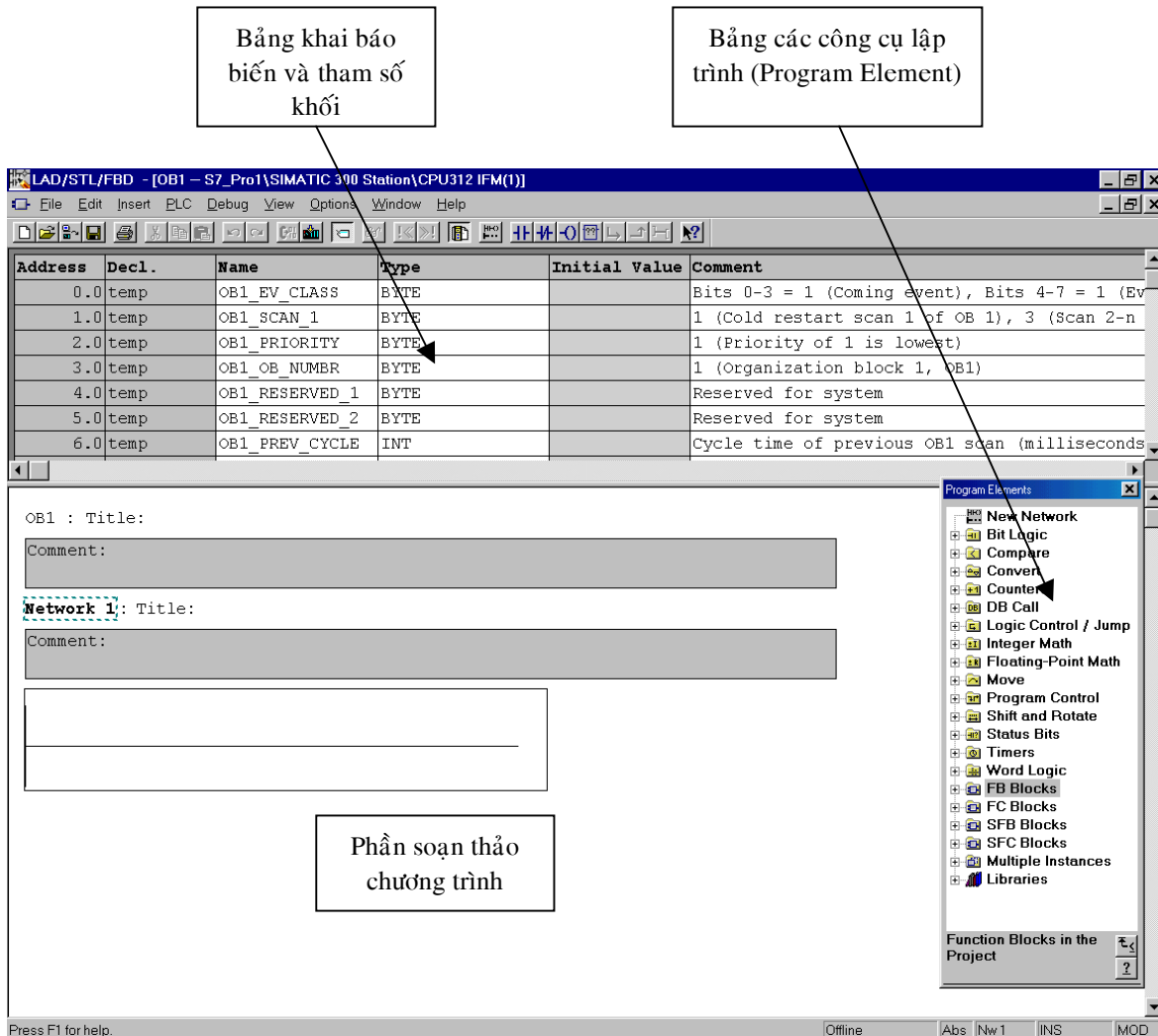
Hình 5.16

- ◆ Khi cửa sổ Properties\_ Organization Block hiện ra (hình 5.17) ta có thể chọn ngôn ngữ lập trình cho khối OB1. Ở đây ta chọn ngôn ngữ LAD (**L**ADDER) để lập trình.



Hình 5.17

## 5.4.2 Cấu trúc cửa sổ lập trình












Hình 5.18 Các thành phần của cửa sổ lập trình

- Bảng khai báo phụ thuộc khối. Dùng để khai báo biến và tham số khối
- Phần soạn thảo chứa một chương trình, nó chia thành từng Network. Các thông số nhập được kiểm tra lỗi cú pháp

Nội dung cửa sổ “Program Element” tùy thuộc ngôn ngữ lập trình đã lựa chọn. Có thể nhấn đúp vào phần tử lập trình cần thiết trong danh sách để chèn chúng vào danh sách. Cũng có thể chèn các phần tử cần thiết bằng cách nhấn và thả chuột.

Các thanh công cụ thường sử dụng:

-  : Mở chương trình mới
-  : Mở chương trình đã có sẵn
-  : Lưu chương trình
-  : Đổ chương trình xuống PLC

-  : Hiển thị địa chỉ dạng tên gọi nhớ (Symbol representation)
-  : Giám sát hoạt động chương trình của PLC
-  : Mở cửa sổ các phần tử lập trình (Program Element)
-  : Rẽ nhánh chương trình
-  : Tạo network mới.

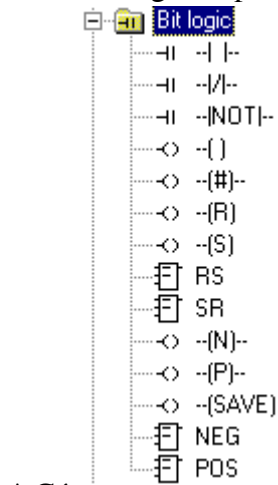
Các Menu công cụ thường dùng:

- |                             |                                      |
|-----------------------------|--------------------------------------|
| • New (File Menu)           | Tạo mới                              |
| • Open (File Menu)          | Mở file                              |
| • Cut (Edit menu)           | Cắt                                  |
| • Paste (Edit Menu)         | Dán                                  |
| • Copy (Edit Menu)          | Sao chép                             |
| • Download (PLC Menu)       | Tải xuống                            |
| • Network (Insert)          | Chèn network mới                     |
| • Program Elements (Insert) | Mở cửa sổ các phần tử lập trình      |
| • Clear/Reset (PLC)         | Xóa chương trình hiện thời trong PLC |
| • LAD, STL, FBD (View)      | Hiển thị dạng ngôn ngữ yêu cầu       |

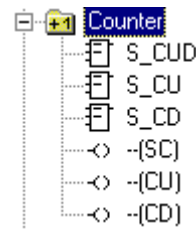
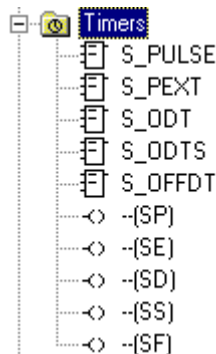
Các phần tử lập trình thường dùng (cửa sổ Program Elements):

\* Các lệnh logic tiếp điểm:

\* Các loại counter.



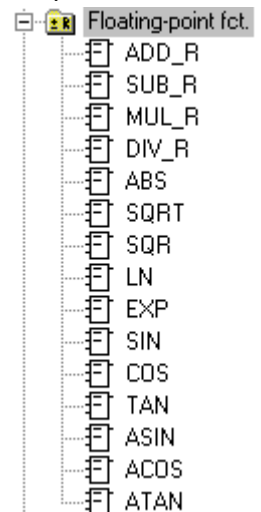
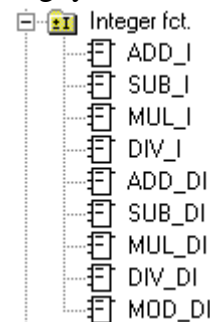
\* Các loại Timer



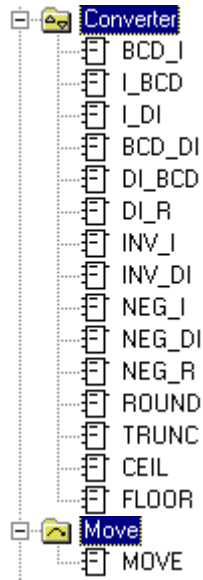
\* Các lệnh toán học

Số nguyên:

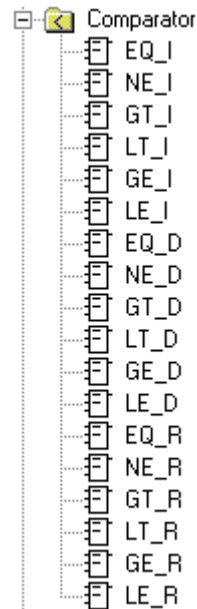
Số thực:



\* Các lệnh chuyển đổi dữ liệu:

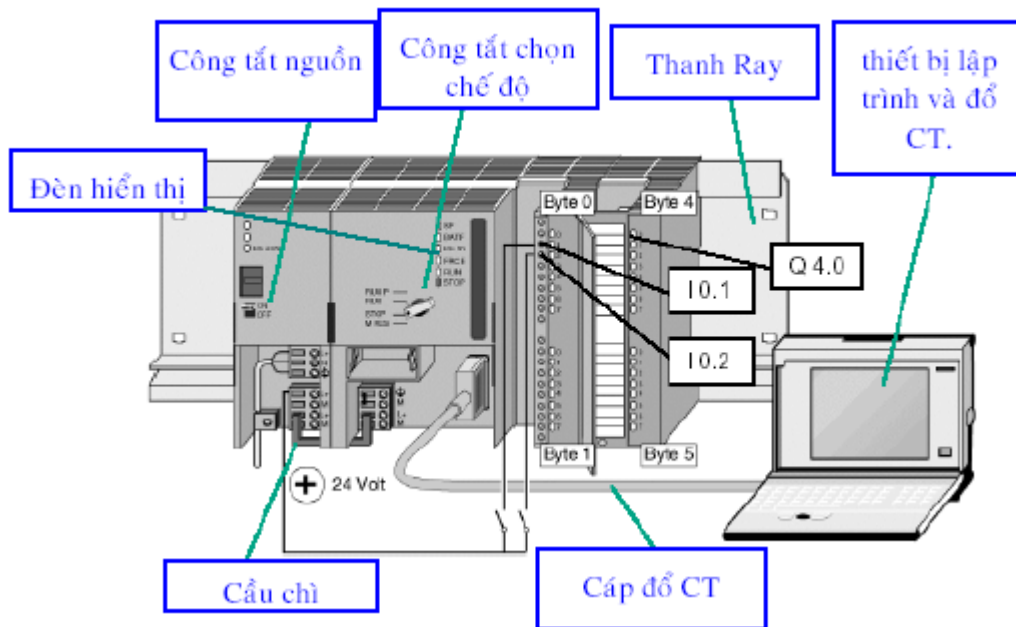


\* Các lệnh so sánh:



**5.4.3 Đổ chương trình**

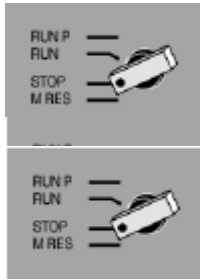
Ta phải thiết lập sẵn sàng sự kết nối đến PLC (hình 5.19) để đổ chương trình.



Hình 5.19



Mở nguồn cho PLC.



Chuyển sang trạng thái stop. Đèn stop hiện lên.

Chuyển cần gạt sang chế độ MRES và giữ khoảng 3s để reset trước khi đổ.  
Chuyển nút gạt trở về vị trí stop và đổ chương trình.




Chọn những khối mà ta sẽ download (từ màn hình Manager), PLC -> Download.

### 5.4.4 Giám sát hoạt động của chương trình

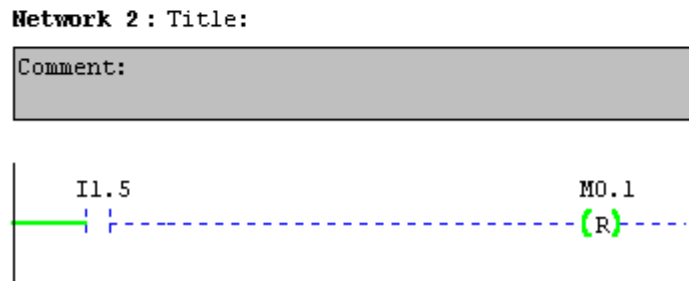
Để quan sát trạng thái hoạt động hiện thời của PLC ta dùng chức năng **Kiểm tra và quan sát**.

Trong chế độ kiểm tra các phần tử trong LAD/FBD được hiển thị ở các màu khác nhau. Có thể định dạng các màu này trong menu *Option -> Customize*.

Để kích hoạt chức năng Kiểm tra và quan sát ta Click vào biểu tượng mắt kính  trên thanh công cụ hoặc vào menu *Debug -> Monitor*

**Khi đó trong chương trình có các đặc điểm:**

- Trạng thái được thực hiện có màu xanh lá và liền nét.
- Trạng thái không thực hiện có dạng đường đứt nét.
- ❖ *Chú ý: Ở chế độ kiểm tra, sự thay đổi trong chương trình là không thể thực hiện được.*





## PHỤ LỤC

PLC SIMATIC S7-200 có các thông số kỹ thuật sau :

Đặc trưng cơ bản của các khối vi xử lý CPU212 và CPU214 được giới thiệu trong bảng :

	CPU212	CPU214
Bộ nhớ chương trình	512 words(1KB) có nhớ	2048 words(4KB) có nhớ
Bộ nhớ dữ liệu	512 words, chứa 100 words có nhớ	2048 words(4KB),chứa 512 words có nhớ
Số cổng logic vào	8	14
Số cổng logic ra	6	10
Số module I/O mở rộng	2	7
Tổng số cổng logic vào	64	64
Tổng số cổng logic ra	64	64
Số bộ tạo thời gian trễ	64/2:1ms,8:10ms,54:100ms	128/4:1ms,16:10ms108:100ms
Số bộ đếm	64	128
Số bộ đếm tốc độ cao	0	3
Số bộ phát xung nhanh	0	2
Số bộ đ. chỉnh tương tự	0	2
Số bit nhớ đặc biệt	368	688
Chế độ ngắt & xử lý tín hiệu	x	X
Thời gian lưu trữ bộ nhớ	50 giờ	190 giờ
Pin kéo dài thời gian nhớ	x	X
Led chỉ thị trạng thái I/O	x	X
Ghép nối máy tính	x	X

• CÁC ĐẶC TRƯNG KỸ THUẬT CỦA CPU 312IFM

CPU và Product Version

- Mã hiệu: 6ES7312-5AC02-0A0B
- Phiên bản phần cứng: 01
- Phiên bản của Hãng: V1.1.0
- Phần mềm thích hợp: STEP 7 V5.0 Service Pack 03

Memory

Bộ nhớ làm việc

- Bộ nhớ nội: 6K
- Bộ nhớ mở rộng: Không

Bộ nhớ LOAD

- Bộ nhớ tích phân: 20KB RAM  
20KB EEPROM
- FEPRM mở rộng: Không
- RAM mở rộng: Không

Backup

- Có Pin: Không
- Không có Pin: Giữ được 72 bytes thông số  
(dữ liệu, cờ, timer)

Thời gian xử lý

- Xử lý với các lệnh bit: Cực tiểu 0.6 μs
- Xử lý với các lệnh từ: Cực tiểu 2 μs
- Phép toán với số nguyên kép: Cực tiểu 3 μs
- Các phép toán với dấu phẩy trôi: Cực tiểu 60 μs

Bộ định thời/dếm và đặc tính lưu giữ

Các bộ đếm S7:

- Điều chỉnh lưu giữ: 32
- Đặt trước (Preset): Từ C 0 tới C 31
- Dải đếm: Từ C 0 tới C 7  
1 tới 999

Các bộ đếm IEC:

- Loại: Có  
SFBs

Bộ định thời S7:

- Điều chỉnh lưu giữ: 64
- Dải định thời gian: Không  
10 ms tới 9990 s

Các bộ định thời IEC:

- Loại: Có  
SPBs

**Miền dữ liệu và các đặc tính lưu giữ**

Toàn bộ miền lưu giữ dữ liệu (các cờ liên hợp, các bộ thời gian, bộ đếm)

Các bộ nhớ Bit:

- Điều chỉnh lưu giữ:
- Đặt trước (Preset)

Các bộ nhớ Clock:

Các khối dữ liệu:

- Kích thước:
- Điều chỉnh lưu giữ:
- Đặt trước (Preset):

Vùng dữ liệu (không thay đổi):

- Lớp quyền ưu tiên

**Các Khối**

Obs

- Kích thước:

Chiều sâu ngăn xếp

- Lớp quyền ưu tiên
- Các mức thêm vào trong vòng 1 lõi OB: Không

FBs

- Kích thước:

FCs

- Kích thước:

**Miền địa chỉ (các lối vào / ra)**

Vùng địa chỉ cho ngoại vi:

- Số / Digital:
- Tích hợp:
- Tương tự / Analog:

Xử lý vẽ hình (không thay đổi được):

Các kênh Digital:

Các kênh Analog:

**Cấu hình**

Khung gắn:

Số module trên giá đỡ:

DB chủ

- Tích hợp
- Qua CP

**Các chức năng thông báo S7**

Kích hoạt ngay lập tức

Báo động - các khối S

**Thời gian**

Đồng hồ thời gian thực:

- Lưu giữ:
- Độ chính xác:

Hoạt động của bộ đếm giờ

Đồng bộ clock

- Trên PLC
- Trên MPI

**Các chức năng kiểm tra và uỷ thác**

Trạng thái /thay đổi các biến

- Các biến

- Số

Các biến màn hình

Các biến thay đổi

Force

- Biến
- Số

Khối monitor

Dây đơn

Điểm gẫy

Cực đại 1 DB, 72 byte dữ liệu

1024

MB 0 tới MB 71

MB 0 tới MB 15

8 (1 byte bộ nhớ)

Cực đại 63 (DB 0 để dự trữ)

Cực đại 6KB

Cực đại 1DB, 72 bytes

Không lưu giữ

Cực đại 512 byte

256 bytes

Xem danh sách lệnh

cực đại 6 KB

8

Cực đại 32

Cực đại 6KB

Cực đại 32

Cực đại 6KB

0 tới 31/0 tới 31

124, 125 E/124 A

256 tới 383/256 tới 383

32 byte + 4 byte integral/

32 byte + 4 byte integral

256 + 10 integral /256 + 6 integral

64/32

1

Cực đại 8

Không

Có

Không

Có

Không

Xem phần 8.1.6

Không

Có

Chủ

Chủ/Tớ

Có

Lối vào, lối ra, các cờ,  
DBs, thời gian, bộ đếm

Cực đại 30

Cực đại 14

Các lối vào, lối ra

Cực đại 10

Có

Có

2

Bộ đệm chuẩn đoán	Có	
- Số lối vào		100
<b><u>Chức năng truyền thông</u></b>		
Truyền thông PD/OP	Có	
Truyền dữ liệu tổng thể	Có	
- Số gói GD		
Gửi		1
Nhận		1
Kích thước của gói GD	Cực đại 22 byte	
Kích thước thích hợp	8 byte	
Truyền thông S7 cơ bản	Có	
- Dữ liệu Người dùng	Cực đại 76 byte	
Kích thước thích hợp	32 byte với X/I_PUT/_GET	
	76 Byte với X_SEND/_RCV	
Truyền Thông S7	Có (server)	
- Dữ liệu Người dùng	Cực đại 160 byte	
Kích thước thích hợp	32 byte	
Tương thích truyền thông S5	Không	
Truyền thông chuẩn	Không	
Số tài nguyên kết nối	6 cho PD/OP/S7 cơ bản/S7 truyền thông	
- Đặt trước cho:		
Truyền thông PD	Cực đại 5	
Người dùng có thể định nghĩa	từ 1 tới 5	
Mặc định	1	
Truyền thông OP	Cực đại 5	
Người dùng có thể định nghĩa	từ 1 tới 5	
Mặc định	1	
Truyền thông S7 cơ bản	Cực đại 2	
Người dùng có thể định nghĩa	từ 0 tới 2	
Mặc định	2	
<b><u>Giao diện</u></b>		
<b><u>Giao diện thứ nhất</u></b>		
<b><u>Chức năng</u></b>		
- MPI		Có
- DP chủ	Không	
- DP tớ	Không	
- Cách điện galvanic	Không	
<b><u>MPI</u></b>		
- Các dịch vụ		
PD/OP		Có
Truyền thông toàn bộ dữ liệu	Có	
Truyền thông S7 cơ bản		Có
Truyền thông S7		Có (Server)
- Vận tốc truyền		19,2; 187,5kbps
<b><u>Kích thước</u></b>		
- Kích thước lắp đặt W x H x D (mm)	80 x 125 x 130	
- Trọng lượng		Khoảng 0,45kg
<b><u>Lập trình</u></b>		
- Ngôn ngữ lập trình		STEP 7
- Tập lệnh lưu trữ		Xem danh sách lệnh
- Ngăn xếp		8
- Các lệnh hệ thống (SFCs)		Xem danh sách lệnh
- Các khối lệnh hệ thống (SFBs)		Xem danh sách lệnh
- Bảo mật chương trình người dùng		Mật khẩu bảo vệ
<b><u>Điện thế, Dòng</u></b>		
Nguồn nuôi		24VDC
- Dải nguồn cho phép		20,4 tới 28,8VDC
Công suất tiêu thụ (không tải)	0.7A (điển hình)	
Dòng khởi động	8A	
$I^2 t$		0,4 A <sup>2</sup> s
Câu chì ngoài cho đường cấp nguồn (khuyến cáo)	Bộ ngắt mạch; 10A, loại B hoặc C	

Nguồn PG cho MPI (15 tới 30VDC)	Cực đại 200mA
Mất mát công suất	9W (điển hình)
Pin	Không
Bộ ắc quy	Không
<b><u>Các lối vào / ra tích hợp</u></b>	
Địa chỉ:	
- Các lối vào Digital	E 124.0 tới E 127.7
- Các lối ra Digital	A 124.0 tới A 124.7
<b><u>Các hàm tích hợp</u></b>	
Đếm	1 (xem Integrated Functions Manual)
Đo tần số	cực đại tới 10 KHz (xem Integrated Functions Manual)

• **ĐẶC TRƯNG KỸ THUẬT CÁC LỐI VÀO ĐẶC BIỆT CỦA CPU 312IFM**

**Module – Dữ liệu đặc biệt**

Số các lối vào	4 từ I 124.6 tới I 125.1
Chiều dài cáp	
- Bọc kim	Cực đại 100m (109 yd.)

**Thế, dòng, điện thế**

Số các lối vào có thể xử lý đồng thời	4
- (Cấu hình ngang) tới 60°C	4
- (Cấu hình đứng) tới 40°C	4

**Các trạng thái, các ngắt, các chuẩn đoán**

Hiển thị trạng thái	1 đèn LED xanh cho mỗi kênh
Ngắt	
- Xử lý ngắt	Có thể thông số hoá

Các hàm chuẩn đoán

Không

**Tài liệu lựa chọn cảm biến**

Thế lối vào	
- Giá trị	24VDC
- Cho tín hiệu "1"	
I 125.0 và I 125.1	15 tới 30 V
I 124.6 và I 24.7	15 tới 30 V
- Cho tín hiệu "0"	-3 tới 5V

Dòng lối vào

- Cho tín hiệu "1"	
I 125.0 và I 125.1	Cực tiểu 2 mA
I 124.6 và I 24.7	Cực tiểu 6.5 mA

Thời gian trễ lối vào

- Từ "0" tới "1"	Cực đại 50µs
- Từ "1" tới "0"	Cực đại 50µs

Các đặc trưng lối vào

- E125.0 và E125.1	Theo IEC 1131, loại 1
- E124.6 và E124.7	Theo IEC 1131, loại 1

Connections nối 2 dây

Không

BEROs

Dòng tĩnh cho phép

I 125.0 và I 125.1	Cực đại 0.5 mA
I 124.6 và I 24.7	Cực đại 2 mA

**Thời gian, tần số**

Thời gian điều phối nội cho

- Xử lý ngắt	Cực đại 1.5ms
--------------	---------------

Tần số lối vào

≤ 10 kHz

• **ĐẶC TRƯNG KỸ THUẬT CÁC LỐI VÀO DIGITAL CỦA CPU 312 IFM**

**Ghi chú:** Có thể gán các thông số các lối vào I 124.6 và I 24.7 như là các lối vào đặc biệt, Khi đó các đặc tính kỹ thuật liệt kê cho các lối vào đặc biệt áp dụng cho các lối vào I 124.6 và I 24.7.

**Module – Đặc trưng Dữ liệu**

Số các lối vào	8
Chiều dài cáp	
- Cáp không bọc kim	Cực đại 600m
- Cáp bọc kim	Cực đại 1000m

**Thế, dòng, điện thế**

Số lối vào có thể xử lý đồng thời	8	
- (Cấu hình ngang) tới 60°C		8
- (Cấu hình đứng) tới 40°C		8
Cách ly điện galvanic		Không
<b><u>Các trạng thái, các ngắt, các chuẩn đoán</u></b>		
Hiển thị trạng thái		1 đèn LED xanh cho mỗi kênh
Các ngắt		Không
Các hàm chuẩn đoán		Không
<b><u>Tài liệu lựa chọn cảm biến</u></b>		
Thế lối vào		
- Giá trị		24VDC
- Cho tín hiệu "1"		11 tới 30V
- Cho tín hiệu "0"		-3 tới 5V
Dòng lối vào		
- Cho tín hiệu "1"		7 mA
Thời gian trễ lối vào		
- Từ "0" tới "1"		1.2 tới 4.8 ms
- Từ "1" tới "0"		1.2 tới 4.8 ms
Các đặc trưng lối vào1		Theo IEC 1131, loại 2
Connections nối 2 dây		Có thể
BEROs		
Dòng tĩnh cho phép		Cực đại 2 mA
• ĐẶC TRƯNG KỸ THUẬT CÁC LỐI RA DIGITAL CỦA CPU 312 IFM		
<b><u>Tài liệu đặc thù - module</u></b>		
Số các lối ra		8
Chiều dài cáp		
- Cáp không bọc kim		Cực đại 600m
- Cáp bọc kim		Cực đại 1000m
<b><u>Thế, dòng, điện thế</u></b>		
Dòng tổng cộng của các lối ra (trên nhóm)		
- (Cấu hình ngang) tới 40°C		Cực đại 3A
- (Cấu hình ngang) tới 60°C		Cực đại 3A
- (Cấu hình đứng) tới 40°C		Cực đại 3A
Cách ly điện galvanic		Không
<b><u>Các trạng thái, các ngắt, các chuẩn đoán</u></b>		
Hiển thị trạng thái		1 đèn LED xanh cho mỗi kênh
Các ngắt		Không
Các hàm chuẩn đoán		Không
<b><u>Tài liệu lựa chọn khởi động</u></b>		
Thế lối ra		
- Cho tín hiệu "1"		Cực tiểu L + (-0.8V)
Dòng lối ra		
- Cho tín hiệu "1" / định mức	0.5 A	
Khoảng cho phép		5 mA đến 0.6 A
- Cho tín hiệu "0"		
Dòng còn dư		cực đại 0.5 mA
Dải tổng trở tải		48Ω tới 4kW
Tải đèn		cực đại 5W
Nối song song của 2 lối ra		
- Cho điều khiển kênh kép tải	Có thể	
- Cho nâng cao chất lượng	Không thể	
Điều khiển của một lối vào số	Có thể	
Tần số chuyển mạch (switching)		
- Đối với tải trở		Cực đại 100Hz
- Đối với tải cảm		
theo IEC 947-5-1, DC13		cực đại 0.5 Hz
- Đối với tải đèn		Cực đại 100Hz
Giới hạn của thế cảm ứng ngắt	30V (điển hình)	
Bảo vệ ngắn mạch lối ra		Có, định thời bằng điện tử
- Ngưỡng nhạy		1A (điển hình)