

# Chương 4: Thiết kế hệ điều khiển dùng PLC

---

**4.1. Trình tự các bước thiết kế**

**4.2. Lập trình cho hệ điều khiển logic dùng PLC**

**4.3. Biểu diễn chương trình bằng Giải đồ thang (LAD)**

**4.4. Một số khái niệm**

**4.5. Biểu diễn chương trình bằng dãy lệnh (STL) và các lệnh cơ bản dạng STL**

**4.6. Hệ lệnh họ PLC CQM1 – OMRON và các ví dụ**

# Chương 4: Thiết kế hệ điều khiển dùng PLC

---

## 4.1. Trình tự các bước thiết kế

4.2. Lập trình cho hệ điều khiển logic dùng PLC

4.3. Biểu diễn chương trình bằng Giải đồ thang (LAD)

4.4. Một số khái niệm

4.5. Biểu diễn chương trình bằng dãy lệnh (STL) và các lệnh cơ bản dạng STL

4.6. Hệ lệnh họ PLC CQM1 – OMRON và các ví dụ

## 4.1. Trình tự các bước thiết kế

---

**Bước 1: Mô tả bài toán**

**Bước 2: Thống kê thiết bị vào ra**

**Bước 3: Chọn thiết bị**

**Bước 4: Thiết lập cấu hình của hệ điều khiển logic**

**Bước 5: Địa chỉ hóa vào/ra**

**Bước 6: Xây dựng phần mềm**

**Bước 7: Chạy mô phỏng**

**Bước 8: Chạy công nghệ**

**Bước 9: Lưu giữ phần mềm**

**Bước 10: Xây dựng tài liệu**

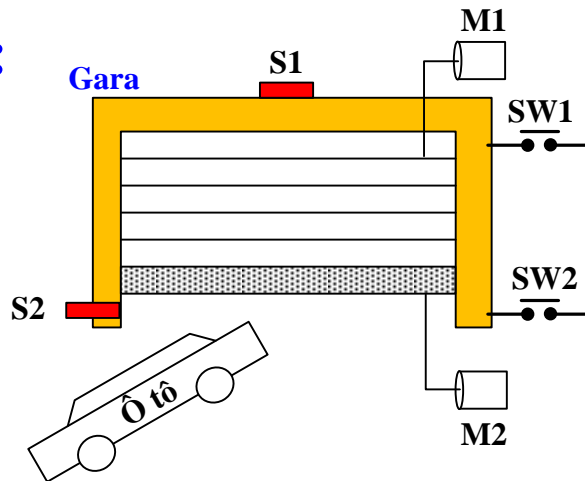
# Bước 1: Mô tả bài toán

---

- ❑ Mô tả bài toán có thể ở dạng 1 hàm hoặc biểu thức logic, có thể ở dạng đồ thị thời gian
- ❑ Trong trường hợp tổng quát, mô tả bài toán là một đoạn văn bản. Đây là cách phổ biến nhất vì nó tương ứng với yêu cầu của người sử dụng.
- ❑ Người thiết kế phải chuyển được từ ngôn ngữ văn bản thành ngôn ngữ điều khiển (hàm logic, bảng, đồ thị thời gian).
- ❑ Yêu cầu cơ bản của bước này là phải mô tả 1 cách đầy đủ và chính xác các yêu cầu của bài toán.

# Bước 1: Mô tả bài toán (tiếp)

**VD1:**



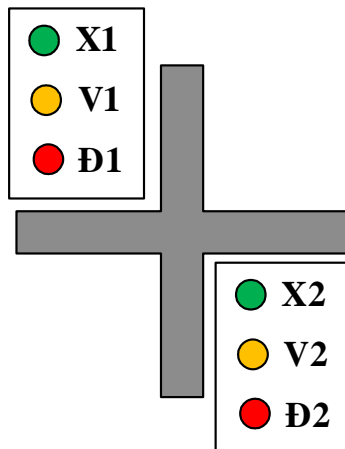
**Yêu cầu bài toán:** khi có xe đến, cửa tự động mở và tự động đóng sau khi xe đi qua.

**Hoạt động:** Khi phát hiện oto tới (S1 on), động cơ M1 on kéo cửa lên. Khi cửa đến giới hạn trên (SW1 on), M1 off.

S2 on khi oto chạy ngang qua cửa và off khi oto vào trong gara.

S2 off, M2 on kéo cửa xuống. Khi cửa đến giới hạn dưới (SW2 on), M2 off. Kết thúc.

**VD2:**



**Điều khiển đèn giao thông ở ngã tư:**

**X1, X2:** sáng 30s

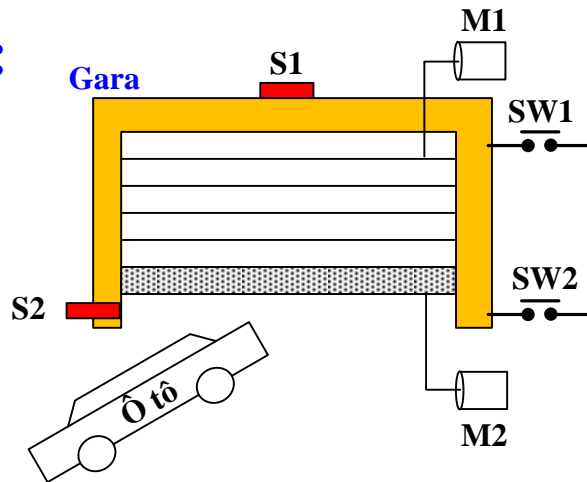
**V1, V2:** sáng 5s

**Đ1, Đ2:** sáng 35s

## Bước 2: Thống kê thiết bị vào ra

- Trên cơ sở các yêu cầu đã có ở bước 1, người thiết kế phải thống kê đầy đủ chủng loại và kiểu của các thiết bị vào/ra.
- Các thiết bị vào/ra thông thường là thiết bị tiêu chuẩn được mô tả trong các tài liệu kỹ thuật.

**VD1:**



**Thiết bị vào:**

**S1:** Sensor phát hiện oto tới

**S2:** Photo-Sensor báo oto qua cửa

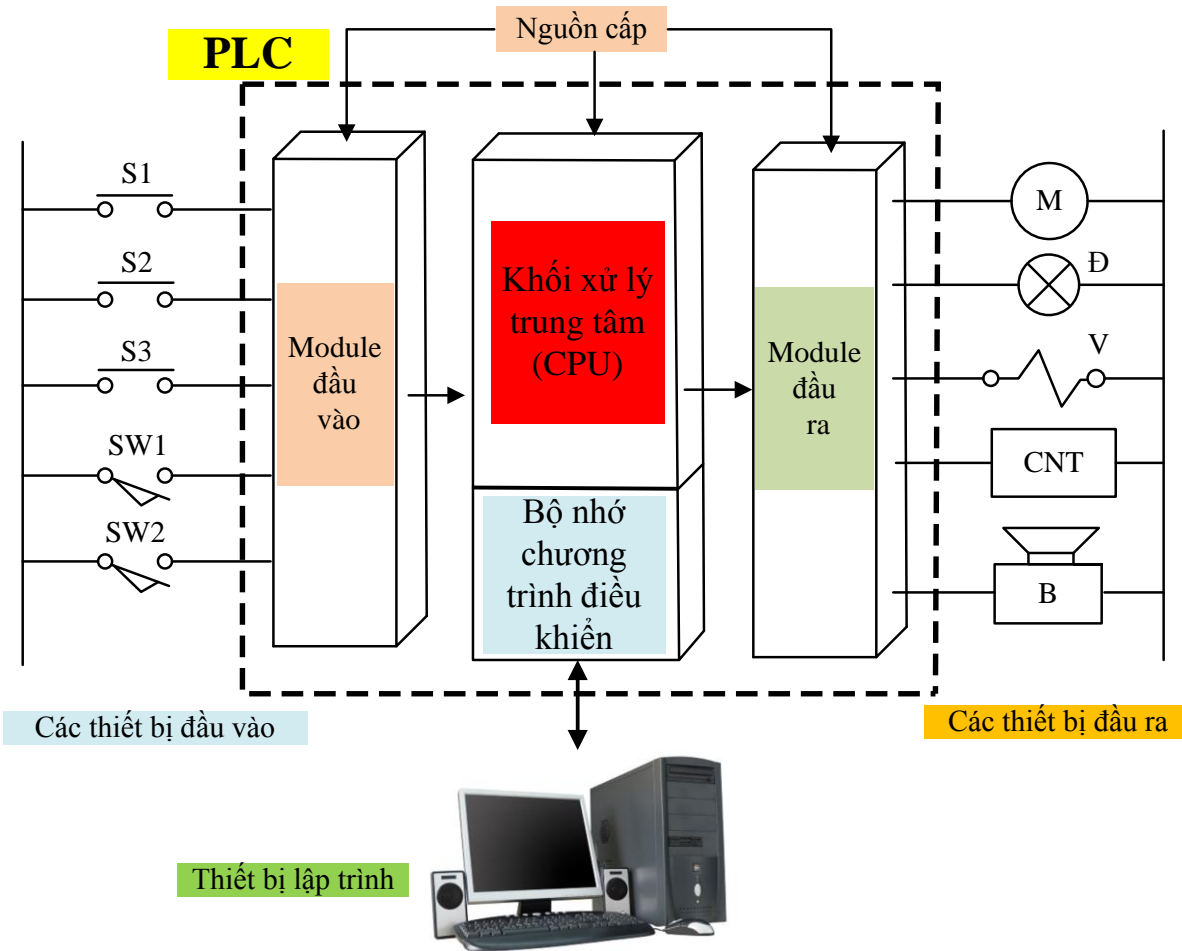
**SW1:** Giới hạn trên của cửa

**SW2:** Giới hạn dưới của cửa

**Thiết bị ra:**

**Động cơ M1 và động cơ M2**

# Bước 3: Chọn thiết bị



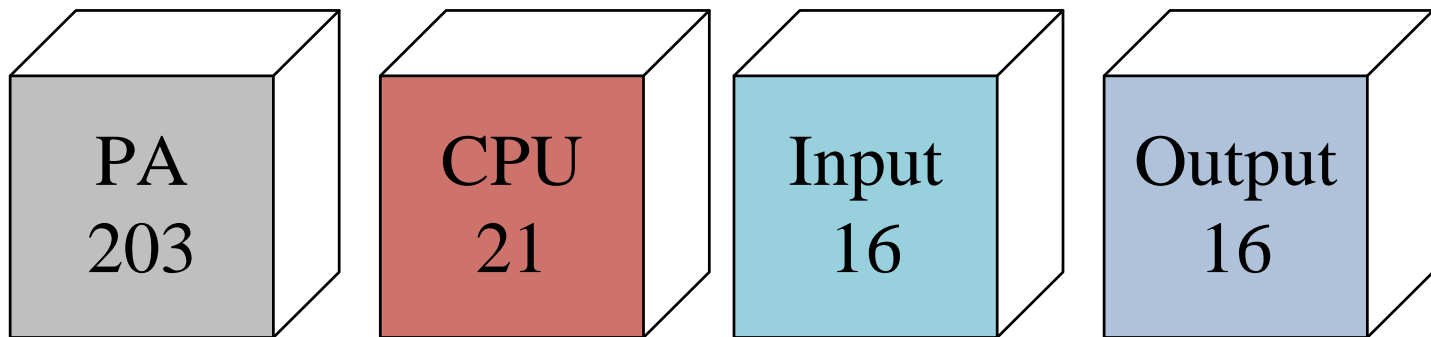
- ☐ Chọn họ PLC sẽ sử dụng
- ☐ Chọn các thành phần cấu trúc của hệ thống
- ☐ Chọn module vào/ra
- ☐ Chọn module nguồn
- ☐ Chọn các thiết bị phụ trợ: đầu nối, cáp truyền thông, các bộ chuyển đổi, ...

# Bước 4: Thiết lập cấu hình của hệ điều khiển

- ❑ Cấu hình của hệ được thiết lập bằng cách lắp ghép các module thành phần trên bảng mạch bus.
- ❑ Ngoài các module nguồn và module CPU có vị trí xác định, các module còn lại có thể lắp đặt ở vị trí bất kỳ trên bảng mạch Bus sao cho thuận lợi nối ghép với các thiết bị vào/ra.

**VD:** Chọn PLC họ CQM1 – CPU21, chọn modules 16 – Inputs và 16 – outputs, chọn Module nguồn PA203.

Cấu hình:





## Bước 5: Địa chỉ hóa vào/ra

- ❑ Trên cơ sở các bước 2, 3, và 4, địa chỉ của các đầu vào/ra được xác định bằng cách địa chỉ hóa.
- ❑ Kết quả của công việc này là thiết lập được hai bảng địa chỉ vào/ra có dạng như sau:

Tên thiết bị vào	Địa chỉ
Start (Nút khởi động)	000.00
Button (Nút nhấn)	000.01
...	...

Tên thiết bị ra	Địa chỉ
M (Động cơ)	100.00
Đ (Đèn)	100.01
...	...

## Bước 5: Địa chỉ hóa vào/ra

**VD: PLC có slot 1 là module vào/ra gồm 8 đầu vào và 8 đầu ra, slot 2 là module vào/ra gồm 16 đầu vào và 16 đầu ra. Địa chỉ hóa các đầu vào/ra như sau:**

	<b>SIEMENS</b>		<b>ALLEN BRADLEY</b>		<b>OMRON</b>	
	Địa chỉ vào	Địa chỉ ra	Địa chỉ vào	Địa chỉ ra	Địa chỉ vào	Địa chỉ ra
Slot 1	I1.0÷I1.7	Q1.0÷Q1.7	I:1/0÷I:1/7	Q:1/0÷Q:1/7	000.00÷000.07	100.00÷100.07
Slot 2	I2.0÷I2.15	Q2.0÷Q2.15	I:2/0÷I:2/15	Q:2/0÷Q:2/15	001.00÷001.15	101.00÷101.15

# Bước 6: Xây dựng phần mềm

**Phần mềm gọi là một thiết kế (Project hoặc Processor File) được lưu giữ dưới 1 tên duy nhất. Bao gồm 2 file:**

- ☐ File chương trình (Program file) chứa các lệnh được tạo ra khi soạn thảo chương trình bằng phần mềm lập trình của họ PLC đã chọn.
- ☐ File dữ liệu (Data file) chứa dữ liệu thực hiện lệnh và cấu trúc hệ thống. File dữ liệu được tạo bằng 2 cách; (1) File dữ liệu được tự động tạo ra khi tiến hành xây dựng một thiết kế mới và soạn thảo File chương trình. (2) File dữ liệu được tạo ra bằng cách soạn thảo trực tiếp từ phần mềm lập trình.

**Tiến hành xây dựng chương trình là công việc chủ yếu của xây dựng phần mềm.**

# Bước 7: Chạy mô phỏng (Dry run)

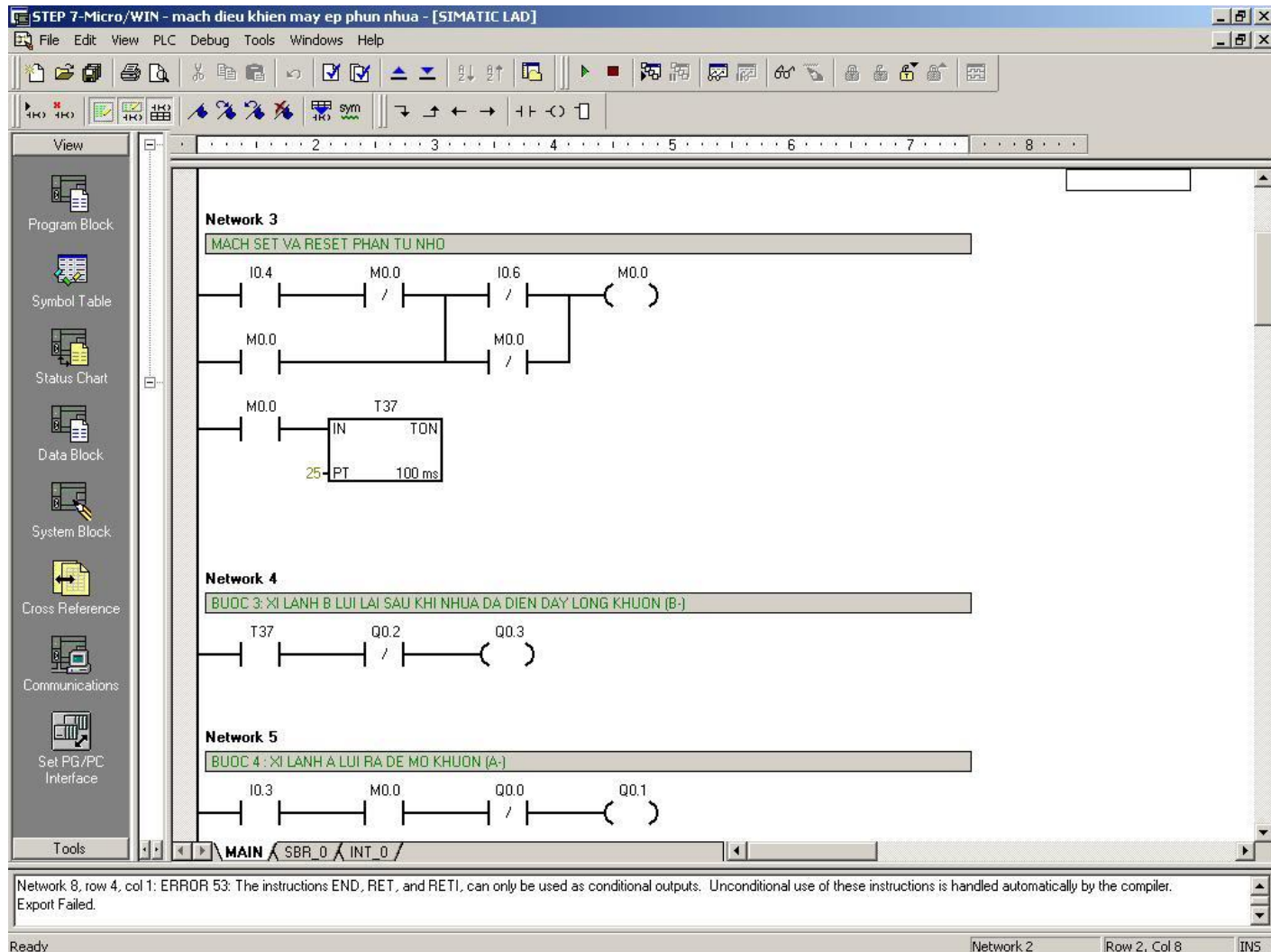
Phần mềm sau khi hoàn thành được nạp vào bộ nhớ của PLC. Thao tác này được thực hiện trên thiết bị lập trình. Để đánh giá chương trình, người thiết kế phải tiến hành chạy thử. Bước chạy thử đầu tiên là chạy mô phỏng, còn gọi là chạy khô (Dry Run) gồm các thủ tục sau:

- Dùng các thiết bị mô phỏng thiết bị vào/ra thực tế và nối thiết bị mô phỏng với các đầu nối trên module vào/ra. VD: các công tắc, chuyển mạch thay thế cho các thiết bị vào; đèn, rơ le thay thế cho thiết bị ra.
- Chạy từng bước của chương trình theo cách mô tả bài toán ở Bước 1.
- Dùng các công cụ phần mềm trên thiết bị lập trình để quan sát, đánh giá.
- Sửa chương trình, kết thúc.

**Kết quả của việc chạy mô phỏng phải đạt được như sau:**

- Chương trình không có lỗi cú pháp.
- Các bước thực hiện tuần tự đúng với yêu cầu công nghệ.

# Bước 7: Chạy mô phỏng (Dry run)



# Bước 8: Chạy công nghệ (Wet run)

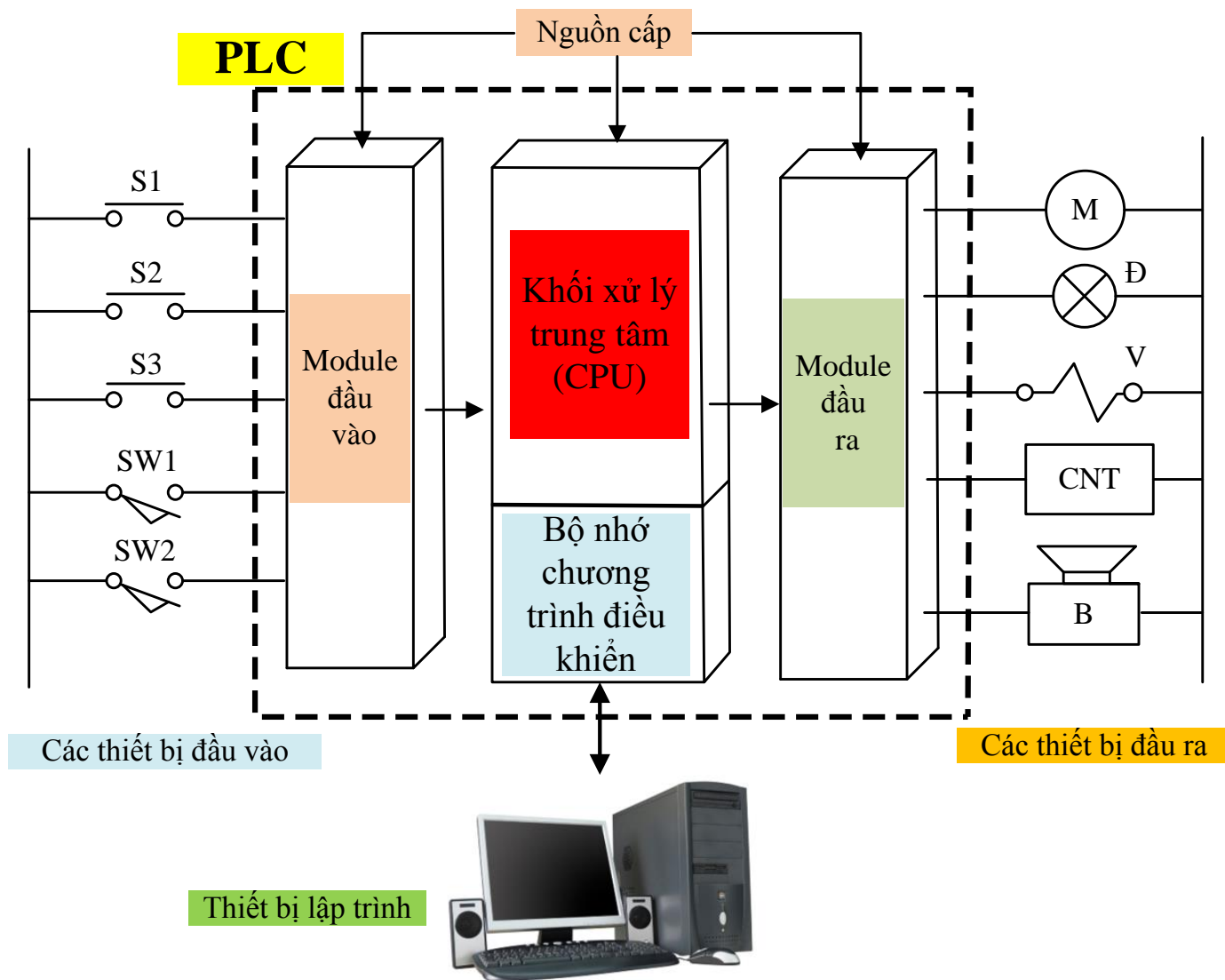
Sau khi hoàn thành bước 7 là điều kiện để thực hiện chạy công nghệ, còn gọi là chạy ướt (Wet Run). Chạy công nghệ bao gồm các bước sau:

- Lắp đặt các thiết bị của hệ điều khiển logic PLC vào dây chuyền công nghệ.
- Kết nối các thiết bị vào/ra với module vào/ra.
- Nạp phần mềm vào PLC và tiến hành chạy thử nghiệm toàn bộ hệ thống theo yêu cầu công nghệ đã mô tả ở bước 1.
- Dùng các công cụ phần mềm trên thiết bị lập trình để quan sát, đánh giá.
- Thêm vào chương trình các khâu thời gian trễ, thay đổi dữ liệu trong File dữ liệu để phối hợp hoạt động 1 cách đồng bộ các khâu, các cơ cấu chấp hành của dây chuyền.

**Kết quả của việc chạy công nghệ phải đạt được như sau:**

- Toàn bộ hệ thống chạy đúng theo yêu cầu công nghệ.
- Hệ thống điều khiển chạy ổn định.

# Bước 8: Chạy công nghệ (Wet run)



# Bước 9: Lưu giữ phần mềm

- Phần mềm sau khi chạy công nghệ là phần mềm gốc, cần phải được cất giữ để phục vụ cho việc bảo trì, bảo dưỡng, thay thế thiết bị sau này.
- Cần nhấn mạnh ở đây là phải cất giữ cả File chương trình và File dữ liệu
- Có thể sử dụng các phương pháp cất giữ như sau: cất giữ trong các thẻ nhớ cố định (EPROM, EEPROM), cất giữ bằng File chương trình, File dữ liệu, cất giữ bằng tài liệu in từ File chương trình, File dữ liệu.
- Vấn đề bản quyền phần mềm cần được chú ý. Để bảo vệ phần mềm, có thể sử dụng công cụ lập trình theo các bước sau:
  - Bảo vệ không cho phép mở File chương trình
  - Bảo vệ không cho phép truy nhập vào bộ nhớ chương trình của PLC.



# Bước 10: Xây dựng tài liệu

---

**Một sản phẩm chỉ được hoàn chỉnh khi có đủ tài liệu đi kèm. Tài liệu gồm 2 loại sau:**

- ❑ Tài liệu hướng dẫn lắp đặt, vận hành hệ thống dành cho người sử dụng.
- ❑ Tài liệu kỹ thuật dành cho cán bộ kỹ thuật bảo trì, bảo dưỡng, thay thế thiết bị gồm có: sơ đồ lắp ráp, cấu hình hệ thống, bảng địa chỉ hóa vào/ra, bảng liệt kê cò, biến trung gian (nếu có), File chương trình, File dữ liệu.



# Chương 4: Thiết kế hệ điều khiển dùng PLC

---

4.1. Trình tự các bước thiết kế

**4.2. Lập trình cho hệ điều khiển logic dùng PLC**

4.3. Biểu diễn chương trình bằng Giải đồ thang (LAD)

4.4. Một số khái niệm

4.5. Biểu diễn chương trình bằng dãy lệnh (STL) và các lệnh cơ bản dạng STL

4.6. Hệ lệnh họ PLC CQM1 – OMRON và các ví dụ

## 4.2. Lập trình cho hệ điều khiển logic dùng PLC

### 4.2.1. Chương trình và ngôn ngữ lập trình

- Chương trình là dãy các lệnh liên tiếp mà PLC phải thực hiện.
- Chương trình được lưu giữ trong bộ nhớ ở dạng mã máy.
- Chương trình do người sử dụng soạn thảo bằng ngôn ngữ lập trình do nhà sản xuất cung cấp
- Ngôn ngữ lập trình là môi trường giao tiếp giữa người và máy.
- Mỗi họ PLC có một hệ lệnh và phần mềm lập trình riêng => sử dụng họ PLC nào phải trang bị phần mềm phù hợp. Phần mềm được cài đặt trong thiết bị lập trình chuyên dụng hoặc PC

VD: PLC họ S7-200 của SIEMENS phải có phần mềm STEP7-200

PLC họ SYSMAC của OMRON phải có phần mềm SYSWIN

## 4.2. Lập trình cho hệ điều khiển logic dùng PLC

### 4.2.2. Chế độ lập trình

- **OFFLINE:** Việc soạn thảo chương trình hoàn toàn tiến hành trên các thiết bị lập trình, không kết nối với PLC => chương trình được lưu ở file => Việc nạp chương trình vào PLC được thực hiện ở thời điểm bất kỳ.
- **ONLINE:** đây là chế độ nạp từng lệnh trực tiếp vào bộ nhớ chương trình. Khi đó PLC được kết nối với thiết bị lập trình. Sử dụng phương pháp này trong hai trường hợp sau:
  - + Soạn thảo trực tiếp một chương trình mới trong PLC
  - + Thay đổi một số lệnh của chương trình đã có sẵn trong PLC khi chương trình đang chạy
- **ONLINE EDIT:** Đây là chế độ đặc biệt giúp cho người sử dụng dễ dàng thay đổi các lệnh của chương trình một cách nhanh chóng mà không cần phải lập lại các thao tác nạp chương trình mới.

## 4.2. Lập trình cho hệ điều khiển logic dùng PLC

### 4.2.3. Các phương pháp biểu diễn chương trình

■ **Dãy lệnh (STL – Statement List):** Mỗi lệnh được biểu diễn bằng một cụm từ viết tắt mô tả lệnh tương tự như hợp ngữ, bao gồm lệnh vào (*Input Instruction*) và lệnh ra (*Output Instruction*).

Cấu trúc của câu lệnh: <Tên lệnh>

Toán hạng 1, toán hạng 2

Tên lệnh (operation) là cụm từ viết tắt chỉ ra chức năng của lệnh, gọi là mã nhớ.

Toán hạng (Operand) là địa chỉ của dữ liệu thực hiện lệnh. Tùy thuộc vào kiểu lệnh, dữ liệu có thể ở dạng Bit, Byte, Word, Double Word, ...

Phương pháp địa chỉ hóa dữ liệu gọi là mode địa chỉ hóa, bao gồm:

- Mode địa chỉ hóa trực tiếp, toán hạng là địa chỉ của dữ liệu.
- Mode địa chỉ hóa gián tiếp, toán hạng là địa chỉ chứa địa chỉ của dữ liệu.
- Mode địa chỉ hóa tức thời, toán hạng là dữ liệu

## 4.2. Lập trình cho hệ điều khiển logic dùng PLC

### 4.2.3. Các phương pháp biểu diễn chương trình

#### ▪ Các lệnh cơ bản dạng STL

- **LD (LoaD)**: lệnh nạp một phần tử mới. Bắt đầu một mạch logic mới hoặc khôi mới.
- **NOT**: lệnh phủ định
- **AND**: lệnh mắc nối tiếp hai phần tử.
- **OR**: lệnh mắc song song hai phần tử.
- **OUT**: lệnh đầu ra

## 4.2. Lập trình cho hệ điều khiển logic dùng PLC

### 4.2.3. Các phương pháp biểu diễn chương trình

#### ■ Ưu điểm của STL:

- Dễ dàng soạn thảo chương trình và có thể biểu diễn tất cả các lệnh ở dạng STL
- File chương trình dạng text, kích thước nhỏ => phù hợp cho chương trình cỡ lớn
- Thuận lợi cho người lập trình chuyên nghiệp. Ngôn ngữ tương tự như hợp ngữ.

#### ■ Nhược điểm của STL:

- Khó thiết kế mạch điều khiển dùng PLC trực tiếp bằng STL vì đòi hỏi kỹ năng lập trình cao.
- STL không trực quan.

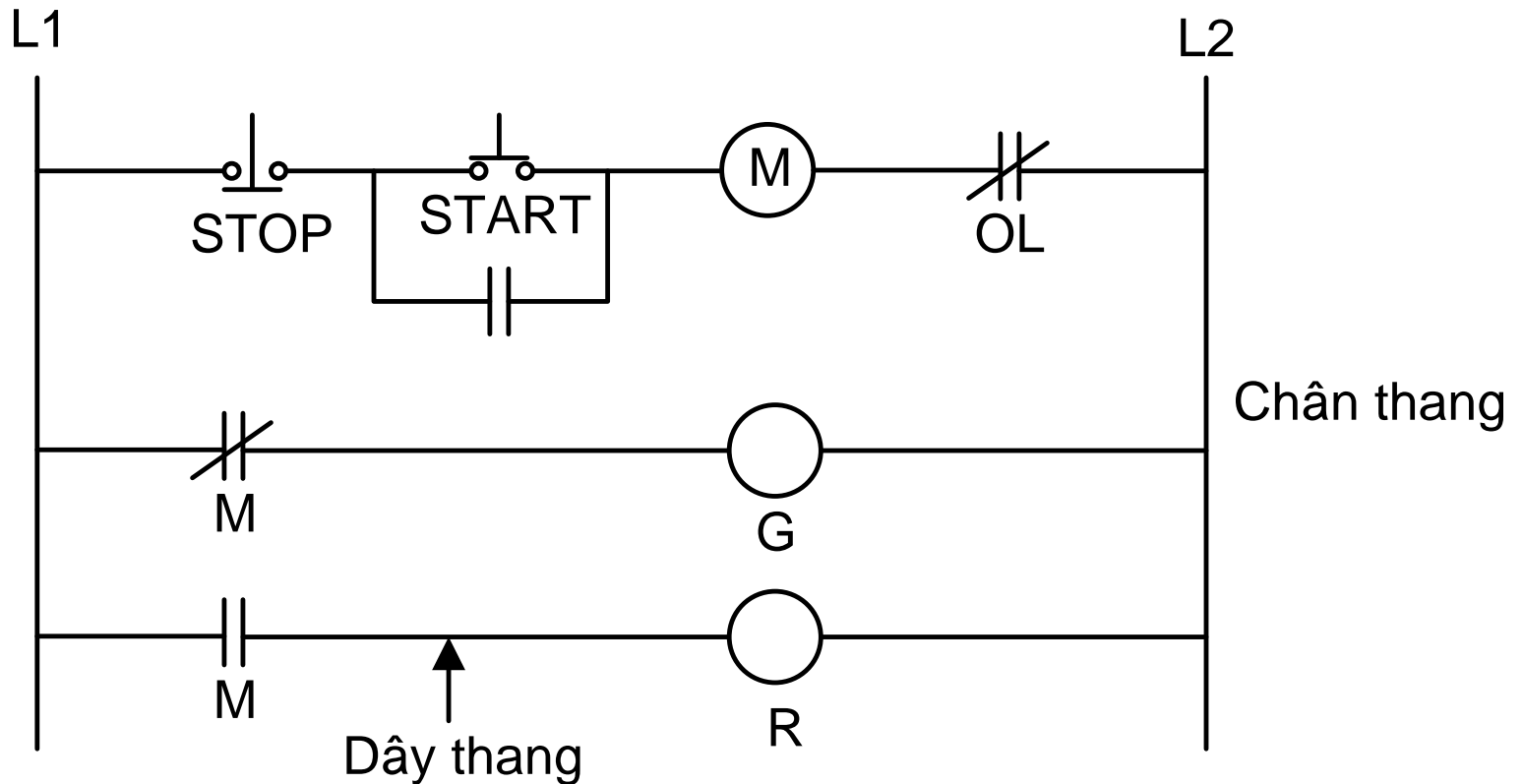
=> STL ít được sử dụng.

**Trong biểu diễn bằng giản đồ thang (LAD) các lệnh được biểu diễn ở dạng biểu tượng đặc trưng và chương trình biểu diễn dưới dạng đồ họa => trở thành cách biểu diễn chuẩn của chương trình cho PLC**

**Biểu diễn bằng sơ đồ khối chức năng logic như sơ đồ các phần tử logic cơ bản.**

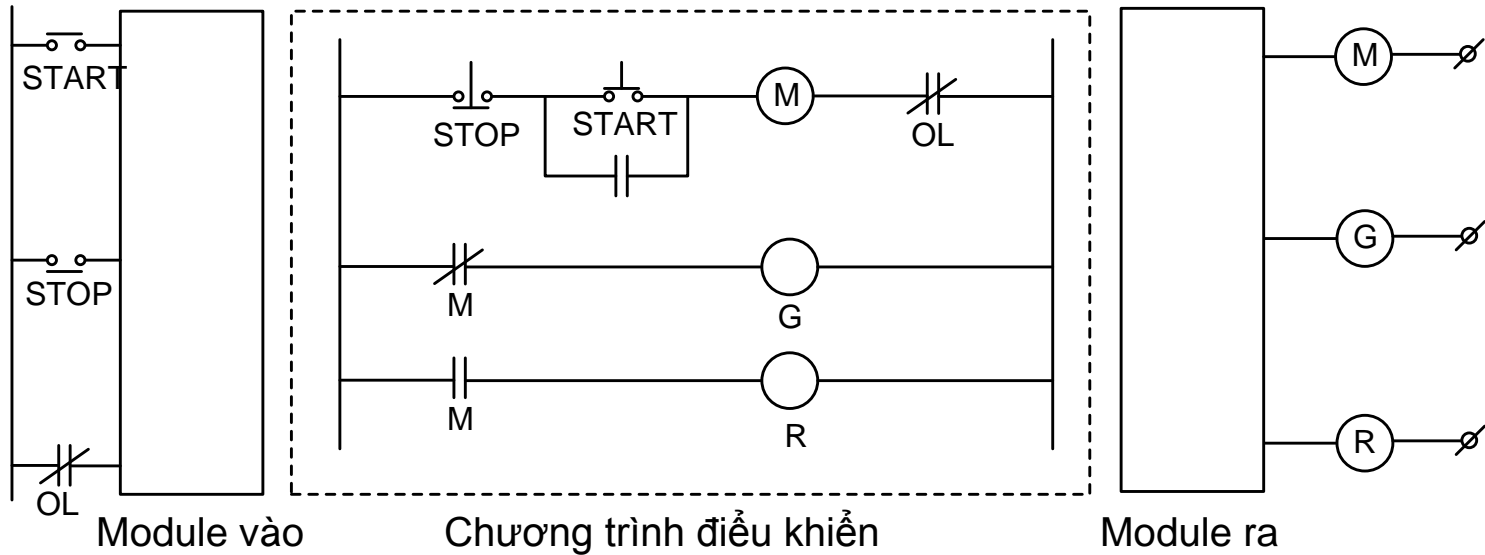


## 4.3. Giải đồ hình thang (LAD)



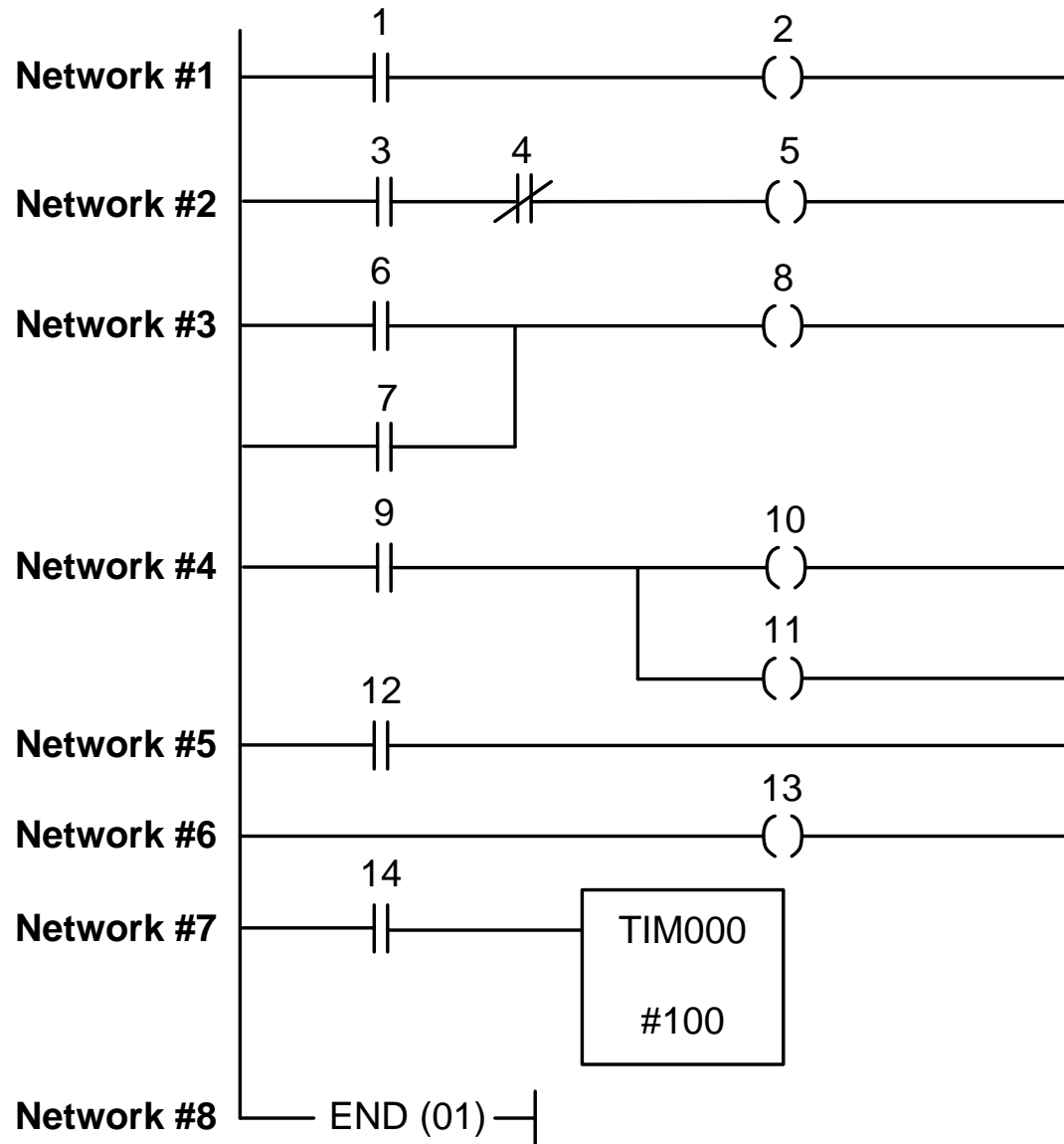
**Mạch điều khiển nối dây cứng**

## 4.3. Giải đồ hình thang (LAD)



**Sơ đồ dạng LAD**

## 4.3. Giải đồ hình thang (LAD)



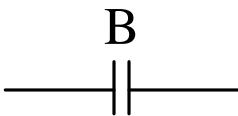


**Hệ lệnh:**

**Họ CQM1 - OMRON**

# Hệ lệnh: Họ CQM1 - OMRON

## ■ Lệnh mô tả tiếp điểm thường mở NO (Normal Opened)

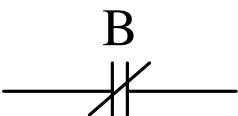
✓ Cú pháp: 

✓ B: địa chỉ Bit

(B) = 1 trạng thái là ON (*Active*)

(B) = 0 trạng thái là OFF (*Inactive*)

## ■ Lệnh mô tả tiếp điểm thường kín NC (Normal Closed)

✓ Cú pháp: 

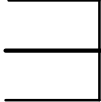
✓ B: địa chỉ Bit

(B) = 1 trạng thái là OFF (*Inactive*)

(B) = 0 trạng thái là ON (*Active*)

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

## ■ Lệnh mô tả cuộn dây rơ le

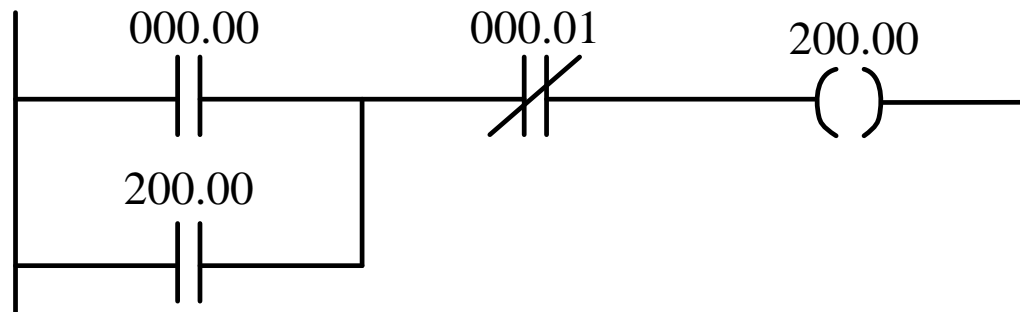
✓ Cú pháp: đ/k logic  —  $\overset{B}{( )}$  — |

✓ B: địa chỉ Bit

đ/k logic = 0  $\Rightarrow$  (B) = 0

đ/k logic = 1  $\Rightarrow$  (B) = 1

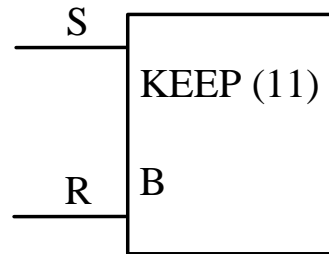
**VD:** Chương trình thực hiện mạch tự duy trì. Sử dụng 2 công tắc S1 và S2 để điều khiển động cơ M: Nhấn S1  $\Rightarrow$  M chạy và nhấn S2  $\Rightarrow$  M tắt. Mạch tự duy trì:



# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- Lệnh KEEP (11) trong PLC của OMRON là lệnh duy trì Bit ra 2 trạng thái ổn định.

Cú pháp:

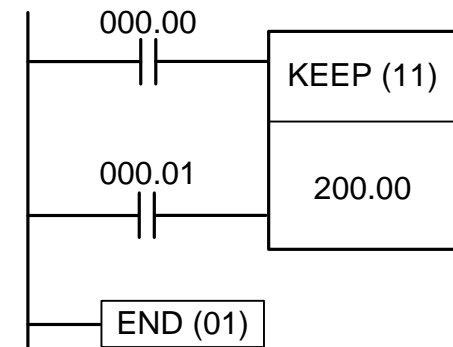
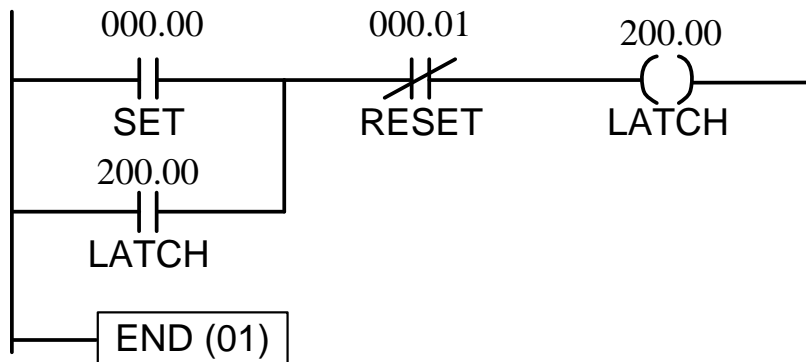


S là đầu vào SET

R là đầu vào RESET

B là địa chỉ Bit ra

**VD2:** Mạch tự duy trì sử dụng lệnh KEEP. Sử dụng lệnh KEEP cho VD 1.

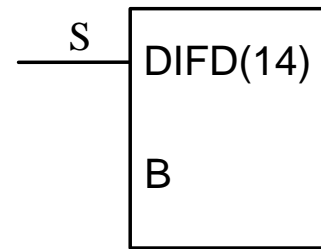
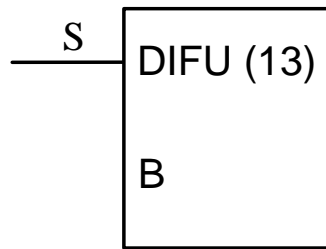




# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- **DIFU(13)** là lệnh tạo xung ngắn khi điều kiện vào chuyển từ mức thấp lên mức cao (Positive Edge).
- **DIFD(14)** là lệnh tạo xung ngắn khi điều kiện vào chuyển từ mức cao xuống thấp (Negative Edge).

- Cú pháp :



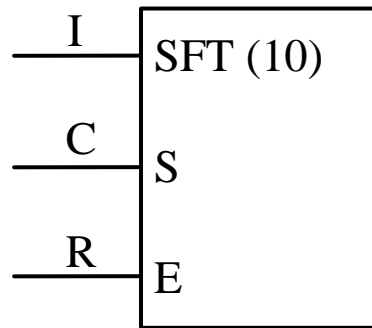
- Hoạt động của lệnh DIFU(13) và DIFD(14):



# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- **Lệnh thanh ghi dịch SFT(10)**
- Định nghĩa thanh ghi: là tập hợp từ các từ liên nhau trong 1 vùng dữ liệu

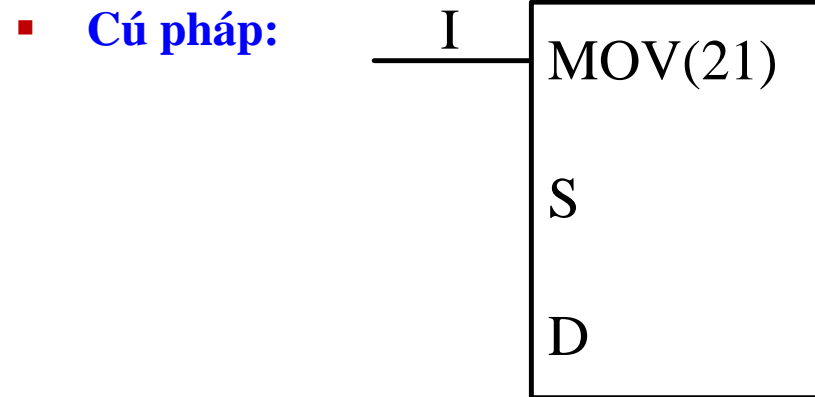
- Cú pháp:



- **S (start):** địa chỉ thanh ghi đầu tiên
- **E (end):** địa chỉ thanh ghi cuối
- **S = E:** thanh ghi là 1 từ
- **Hoạt động:** **C = 1:** dịch trái 1 bit, **I** điền vào Bit trẻ nhất, Bit già nhất mất
- Thanh ghi dịch dùng để phát hiện và loại bỏ sản phẩm lỗi

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- Lệnh chuyển dữ liệu MOV(21)

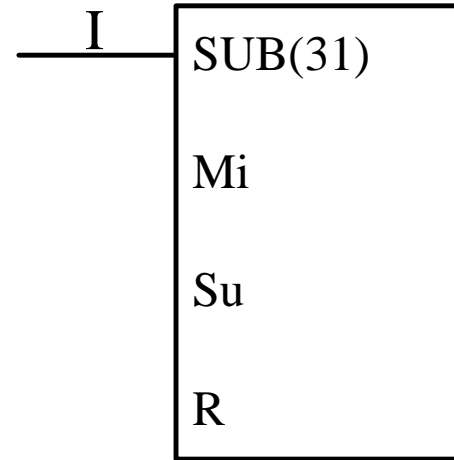
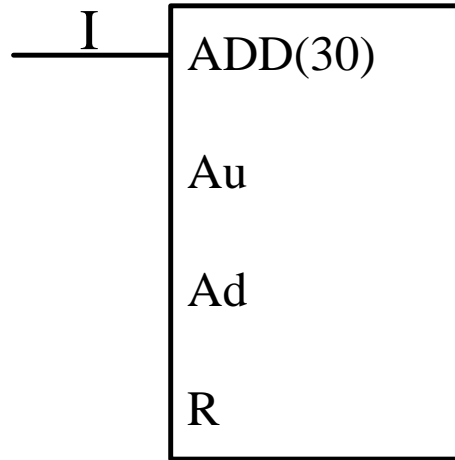


- S (source): địa chỉ của word nguồn
- D (destination): địa chỉ của word đích
- Hoạt động: I = 1, word nguồn (S) => word đích (D)
- Chú ý: trong trường hợp phân toán hạng là dữ liệu, chúng ta phải thêm dấu #

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- Lệnh cộng, trừ số học (BCD)

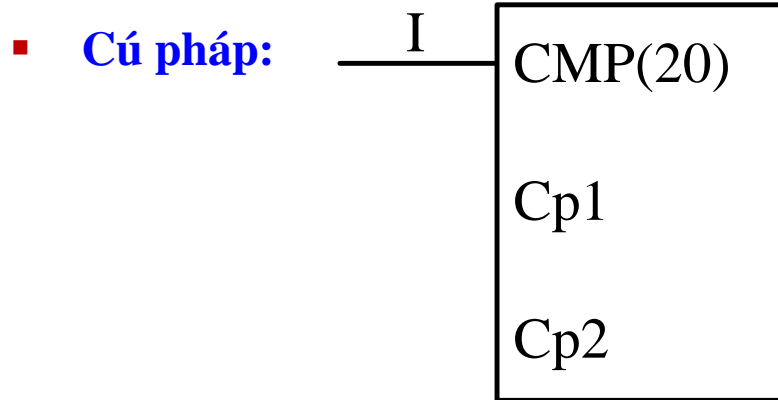
- Cú pháp:



- **Au, Ad:** địa chỉ của toán hạng
- **R:** địa chỉ của kết quả
- **Mi:** địa chỉ của số bị trừ, **Su:** địa chỉ của số trừ
- **R:** địa chỉ của kết quả

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

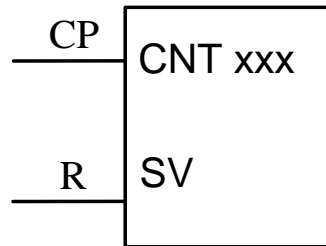
- **Lệnh so sánh CMP(20)**



- **Cp1:** địa chỉ của từ so sánh thứ nhất
- **Cp2:** địa chỉ của từ so sánh thứ hai
- **Hoạt động:**  $I = 1$  so sánh lập cờ,  $(Cp1) - (Cp2) \Rightarrow$  lập cờ
  - ✓ 255.05: cờ lớn hơn
  - ✓ 255.06: cờ bằng nhau
  - ✓ 255.07: cờ nhỏ hơn

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- **COUNTER** dùng để đếm sự kiện xảy ra.
- COUNTER có bộ đếm tăng (COUNTER-Up), bộ đếm giảm (COUNTER-Down), và bộ đếm thuận nghịch (COUNTER-Up/Down).
- Các họ PLC khác nhau cung cấp các lệnh COUNTER khác nhau.
- Cú pháp lệnh:



**CP** là đầu vào xung đếm. Khi CP chuyển từ thấp lên cao thì giá trị COUNTER tăng 1 đơn vị.

**xxx** là số TC trong vùng nhớ dữ liệu dành cho TIMER/COUNTER

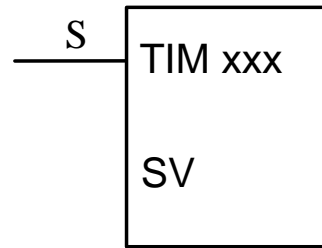
**CNTxxx** là toán hạng chỉ ra địa chỉ của giá trị hiện tại của PV của COUNTER hoặc cờ hoàn thành (Complement Flag)

**SV** là địa chỉ chứa giá trị đặt của COUNTER

**R** là đầu vào RESET

# Hệ lệnh: Họ CQM1 – OMRON (tiếp)

- **TIMER** là bộ định thời gian có rất nhiều ứng dụng trong điều khiển. TIMER thực chất là bộ đếm xung thời gian. Hệ thống cung cấp cho người sử dụng các cờ định thời trong vùng dữ liệu (SR) với các chu kỳ: 0.01s, 0.1s, 0.2s, ...
- TIMER có 2 loại chính: ON-Delay và OFF-Delay. Tuy nhiên, từ 1 loại (ON-Delay) có thể xây dựng các kiểu TIMER khác như: OFF-Delay, ON-OFF Delay, ...
- Các họ PLC khác nhau cung cấp các lệnh TIMER kiểu ON-Delay, OFF-Delay, ... khác nhau. Dữ liệu dành cho TIMER được xác định trong vùng nhớ dữ liệu dành cho TIMER.
- Cú pháp lệnh TIMER ON-Delay:



**S (Start)** là đầu vào điều kiện cho phép TIMER hoạt động

**xxx** là số TC trong vùng nhớ dữ liệu dành cho TIMER/COUNTER

**TIM xxx** là toán hạng chỉ ra địa chỉ của giá trị hiện tại của TIMER (Present Value) hoặc cờ hoàn thành (Complement Flag)

**SV** là địa chỉ chứa giá trị đặt của TIMER