



Ch 3: Phương pháp lập trình

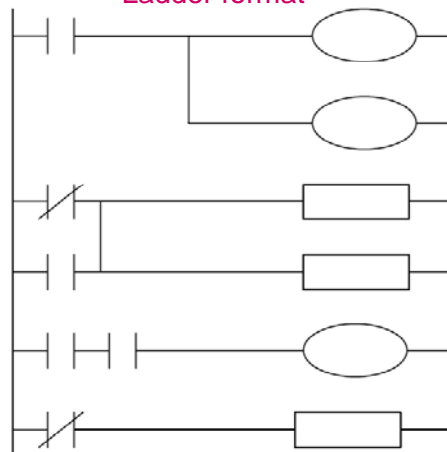
• Chương trình:

Chương trình là một dãy các lệnh có liên kết với nhau và được biểu diễn ở dạng ngôn ngữ nào đó mà PLC có thể hiểu được. Với PLC, có hai dạng ngôn ngữ lập trình cơ bản: ngôn ngữ “dòng lệnh” (**Instruction format**) và ngôn ngữ “bậc thang” (**Ladder format**)

Instruction format

LD	X10
OUT	Y7
AND	M38
SET	S5
LD	X21
OUT	T01
	K40

Ladder format

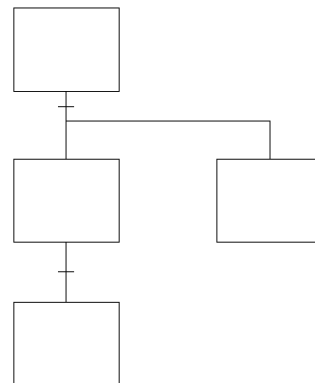


3.1. Ngôn ngữ lập trình

• **Ngôn ngữ dòng lệnh:** là ngôn ngữ lập trình cơ bản dễ học, dễ dùng nhưng phải mất nhiều thời gian kiểm tra đối chiếu để tìm ra mối quan hệ giữa một đoạn chương trình lớn với chức năng thể hiện trong đó.

• **Ngôn ngữ bậc thang:** là ngôn ngữ có dạng đồ họa, cho phép nhập chương trình có dạng như một sơ đồ mạch điện logic và dùng các ký hiệu điện để biểu diễn các công tắc logic ngõ vào và rơ-le logic ngõ ra.

Ngoài hai dạng ngôn ngữ trên, những phần mềm lập trình chuyên nghiệp còn hỗ trợ chức năng lập trình ở dạng biểu đồ tuần tự chức năng SFC (Sequential Function Chart) – có kiểu tương tự sơ đồ Grafset – rất hữu ích trong những hệ thống điều khiển theo trình tự.

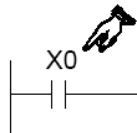




3.2. Các lệnh cơ bản – họ FX Mitsubishi

- **Lệnh LD:** dùng để đặt một tiếp điểm thường mở vào chương trình.

Mnemonic	Function	Format	Devices	Program steps
LD (LoaD)	Initial logical operation contact type NO (normally open)		X, Y, M, S, T, C	1



0 LD X 0
1 OUT Y 0

Trong chương trình instruction, lệnh LD luôn xuất hiện ở vị trí đầu tiên của một dòng chương trình hoặc mở đầu cho một khối logic; còn trong chương trình ladder, lệnh LD thể hiện một tiếp điểm thường mở nối trực tiếp vào nhánh bên trái hoặc là một tiếp điểm thường mở đầu tiên của một khối logic.



3.2. Các lệnh cơ bản – họ FX Mitsubishi

- **Lệnh LDI:** dùng để đặt một tiếp điểm thường đóng vào chương trình.

Mnemonic	Function	Format	Devices	Program steps
LDI (LoaD Inverse)	Initial logical operation contact type NC (normally closed)		X, Y, M, S, T, C	1



2 LDI X 1
3 OUT M 100

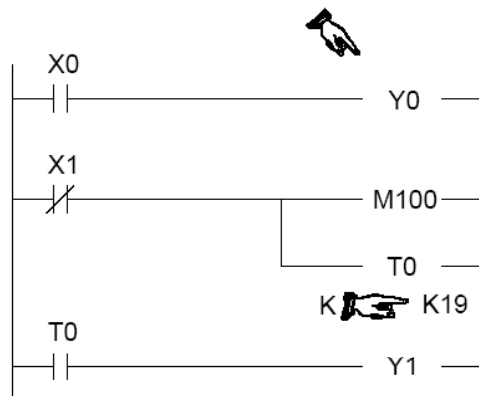
Trong chương trình instruction, lệnh LDI luôn xuất hiện ở vị trí đầu tiên của một dòng chương trình hoặc mở đầu cho một khối logic; còn trong chương trình ladder, lệnh LDI thể hiện một tiếp điểm thường đóng nối trực tiếp vào nhánh bên trái hoặc là một tiếp điểm thường đóng đầu tiên của một khối logic.



3.2. Các lệnh cơ bản – họ FX Mitsubishi

- **Lệnh OUT:** dùng để đặt một rơ-le logic vào chương trình.

Mnemonic	Function	Format	Devices	Program steps
OUT (OUT)	Final logical operation type coil drive		Y, M, S, T, C	Y, M: 1 S, special M coils: 2 T: 3 C (16 bit): 3 C (32 bit): 5



```

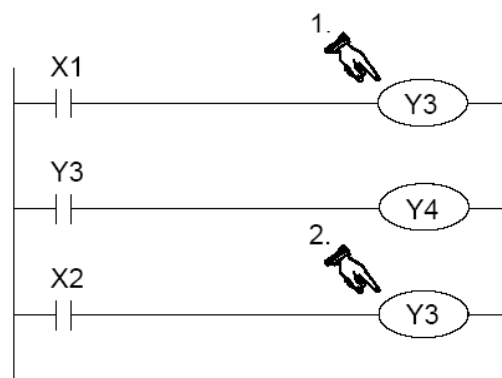
0  LD    X    0
1  OUT   Y    0
2  LDI   X    1
3  OUT   M    100
4  OUT   T    0
   K19
7  LD    T    0
8  OUT   Y    1
  
```



3.2. Các lệnh cơ bản – họ FX Mitsubishi

Trong chương trình ladder, lệnh OUT ký hiệu bằng () được nối trực tiếp vào nhánh bên phải. Lệnh OUT được thực hiện khi điều kiện phía bên trái của nó được thỏa. Tham số của lệnh OUT không duy trì được trạng thái.

Lưu ý: lỗi double coil designation



Xét trường hợp khi X1 = ON, và X2 = OFF.



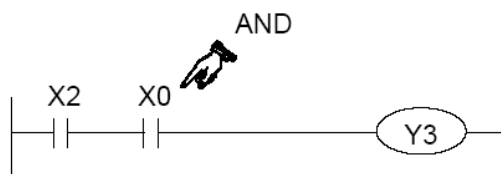
Sau khi 3 nhánh chương trình được quét, kết quả sẽ là: Y3 = OFF, và Y4 = ON



3.2. Các lệnh cơ bản – họ FX Mitsubishi

- **Lệnh AND / ANI:** dùng để đặt nối tiếp một tiếp điểm thường mở / thường đóng .

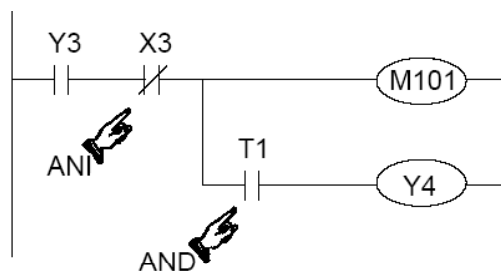
Mnemonic	Function	Format	Devices	Program steps
AND (AND)	Serial connection of NO (normally open) contacts		X, Y, M, S, T, C	1
ANI (AND Inverse)	Serial connection of NC (normally closed) contacts		X, Y, M, S, T, C	1



0	LD	X	2
1	AND	X	0
2	OUT	Y	3



3.2. Các lệnh cơ bản – họ FX Mitsubishi



3	LD	Y	3
4	ANI	X	3
5	OUT	M 101	
6	AND	T	1
7	OUT	Y	4

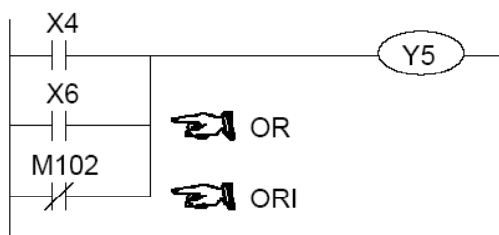
Lưu ý: PLC không giới hạn số tiếp điểm mắc nối tiếp hoặc song song cũng như số nhánh ngõ ra liên tiếp (follow-on output). Tuy nhiên, các thiết bị phần cứng như thiết bị lập trình, máy in và màn hình không thể hiển thị được hết nếu số tiếp điểm / số nhánh liên tiếp quá nhiều. Thông thường, nên giới hạn từ 10 tiếp điểm trở xuống trên một nhánh; và số nhánh ngõ ra liên tiếp từ 24 trở xuống.



3.2. Các lệnh cơ bản – họ FX Mitsubishi

- **Lệnh OR / ORI:** dùng để đặt song song một tiếp điểm thường mở / thường đóng .

Mnemonic	Function	Format	Devices	Program steps
OR (OR)	Parallel connection of NO (normally open) contacts		X, Y, M, S, T, C	1
ORI (OR Inverse)	Parallel connection of NC (normally closed) contacts		X, Y, M, S, T, C	1



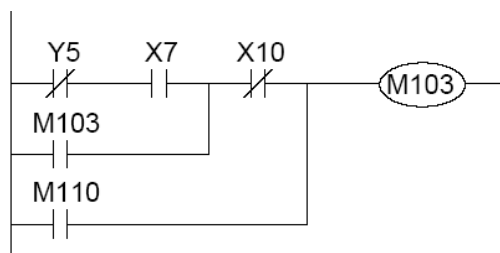
```

0 LD X 4
1 OR X 6
2 ORI M 102
3 OUT Y 5

```



3.2. Các lệnh cơ bản – họ FX Mitsubishi



```

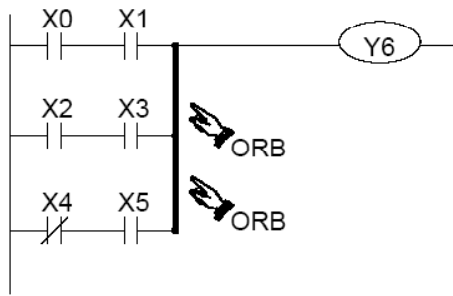
4 LDI Y 5
5 AND X 7
6 OR M 103
7 ANI X 10
8 OR M 110
9 OUT M 103

```

- **Lệnh ANB:** dùng để mắc nối tiếp với 1 khối có từ 2 tiếp điểm mắc song song trở lên.
- **Lệnh ORB:** dùng để mắc song song với 1 khối có từ 2 tiếp điểm mắc nối tiếp trở lên.

Mnemonic	Function	Format	Devices	Program steps
ANB (ANd Block)	Serial connection of multiple parallel circuits		N/A	1
ORB (OR Block)	Parallel connection of multiple contact circuits		N/A	1

3.2. Các lệnh cơ bản – họ FX Mitsubishi



Recommended sequential programming method

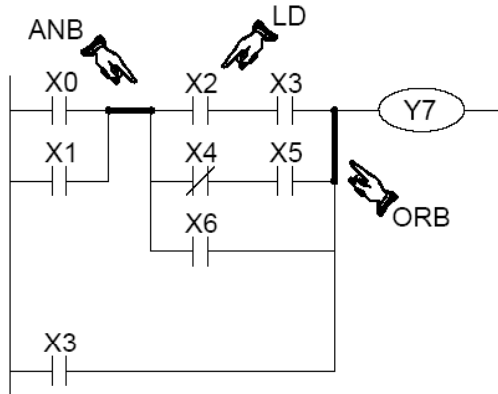
```

0 LD X 0
1 AND X 1
2 LD X 2
3 AND X 3
4 ORB
5 LDI X 4
6 AND X 5
7 ORB
8 OUT Y 6
  
```

Non-preferred batch programming method

```

0 LD X 0
1 AND X 1
2 LD X 2
3 AND X 3
4 LDI X 4
5 AND X 5
6 ORB
7 ORB
8 OUT Y 6
  
```



```

0 LD X 0
1 OR X 1
2 LD X 2
3 AND X 3
4 LDI X 4
5 AND X 5
6 ORB
7 OR X 6
8 ANB
9 OR X 3
10 OUT Y 7
  
```

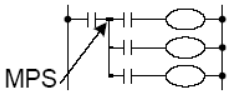
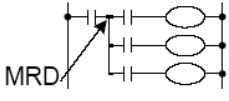
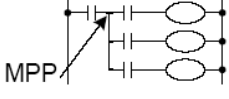
© C.B. Pham

Bộ điều khiển lập trình

3-11

3.2. Các lệnh cơ bản – họ FX Mitsubishi

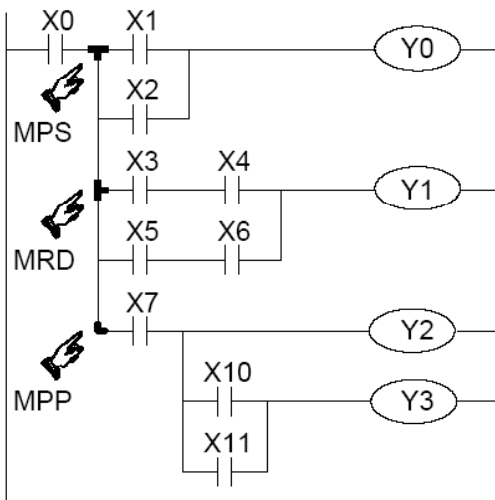
- **Lệnh MPS / MRD / MPP:** dùng để thực hiện việc rẽ nhánh cho các tác vụ phía bên phải của nhánh ở phần thi hành.

Mnemonic	Function	Format	Devices	Program steps
MPS (Point Store)	Stores the current result of the internal PLC operations		N/A	1
MRD (Read)	Reads the current result of the internal PLC operations		N/A	1
MPP (PoP)	Pops (recalls and removes) the currently stored result		N/A	1

Lưu ý: Đối với mỗi lệnh MPS, thì phải có lệnh MPP tương ứng. Số cặp MPS-MPP phải luôn giới hạn nhỏ hơn 11.



3.2. Các lệnh cơ bản – họ FX Mitsubishi

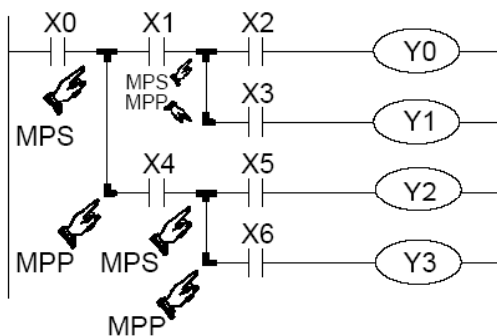


0	LD	X 0
1	MPS	
2	LD	X 1
3	OR	X 2
4	ANB	
5	OUT	Y 0
6	MRD	
7	LD	X 3
8	AND	X 4
9	LD	X 5
10	AND	X 6
11	ORB	

12	ANB	
13	OUT	Y 1
14	MPP	
15	AND	X 7
16	OUT	Y 2
17	LD	X 10
18	OR	X 11
19	ANB	
20	OUT	Y 3



3.2. Các lệnh cơ bản – họ FX Mitsubishi



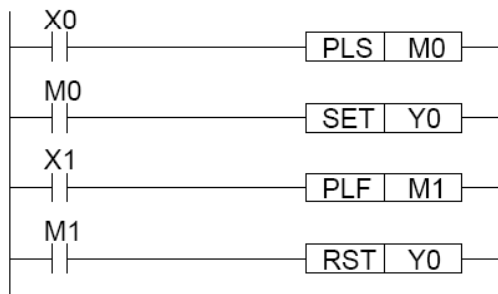
0	LD	X 0
1	MPS	
2	AND	X 1
3	MPS	
4	AND	X 2
5	OUT	Y 0
6	MRD	
7	LD	X 3
8	AND	X 4
9	LD	X 5
10	AND	X 6
11	ORB	
12	ANB	
13	OUT	Y 1
14	MPP	
15	AND	X 7
16	OUT	Y 2
17	LD	X 10
18	OR	X 11
19	ANB	
20	OUT	Y 3

9	MPP	
10	AND	X 4
11	MPS	
12	AND	X 5
13	OUT	Y 2
14	MPP	
15	AND	X 6
16	OUT	Y 3

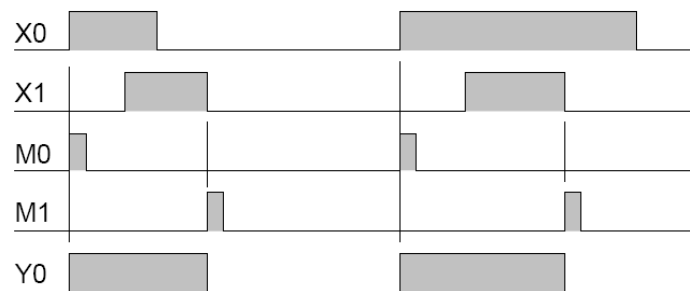


3.2. Các lệnh cơ bản – họ FX Mitsubishi

3.2. Các lệnh cơ bản – họ FX Mitsubishi

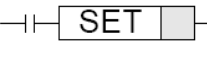



0	LD	X	0
1	PLS	M	0
3	LD	M	0
4	SET	Y	0
5	LD	X	1
6	PLF	M	1
8	LD	M	1
9	RST	Y	0



3.2. Các lệnh cơ bản – họ FX Mitsubishi

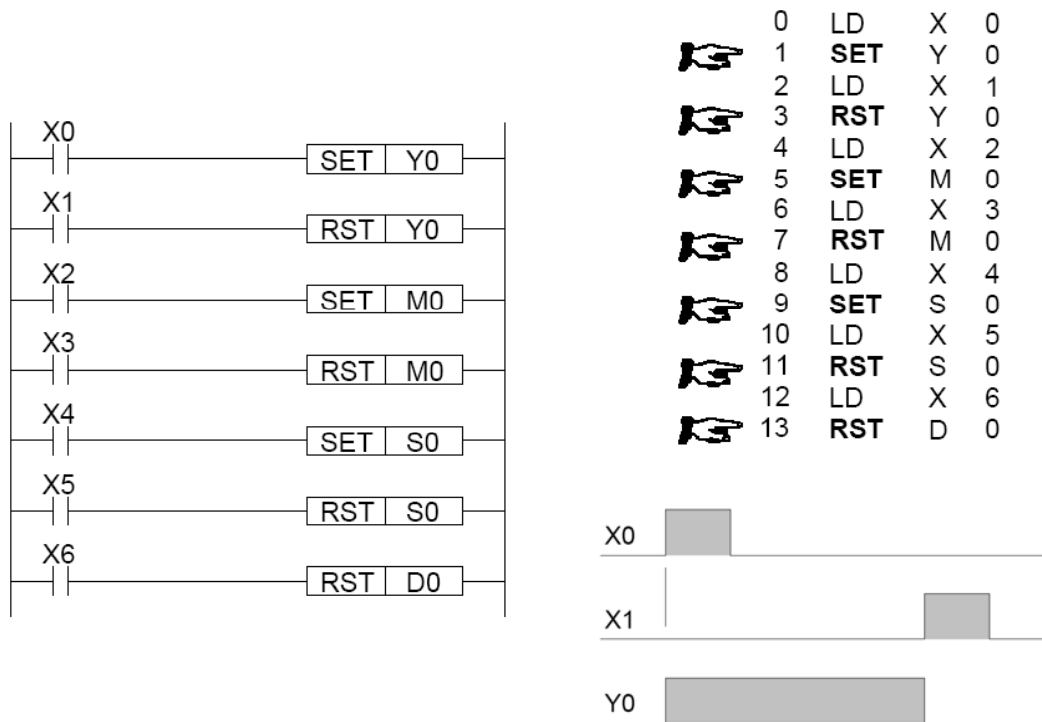
- **Lệnh SET / RST:** dùng để đặt trạng thái của tham số lên mức 1 / xuống mức 0.

Mnemonic	Function	Format	Devices	Program steps
SET (<i>SET</i>)	Sets a bit device permanently ON		Y, M, S	Y, M:1 S, special M coils:2
RST (<i>ReSeT</i>)	Resets a bit device permanently OFF		Y, M, S, D, V, Z	D, special D registers, V and Z:3

Lưu ý:

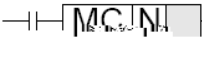

- Trong chương trình ladder, lệnh SET / RST luôn xuất hiện ở phía cuối nhánh (được nối trực tiếp vào nhánh bên phải).
- Các tham số của lệnh SET / RST duy trì trạng thái của nó sau khi bị tác động.
- Lệnh RST còn được dùng để gán giá trị 'không' cho các thanh ghi (D, V, Z)

3.2. Các lệnh cơ bản – họ FX Mitsubishi



3.2. Các lệnh cơ bản – họ FX Mitsubishi

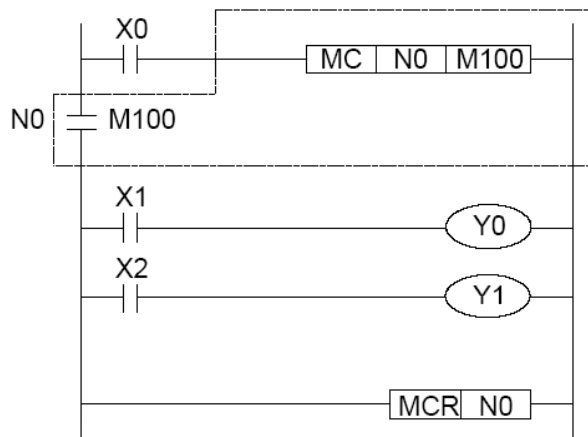
- **Lệnh MC / MCR:** dùng để khai báo / kết thúc một khối lệnh

Mnemonic	Function	Format	Devices	Program steps
MC (Master Control)	Denotes the start of a master control block		Y, M (no special M coils allowed) N denotes the nest level (N0 to N7)	3
MCR (Master Control Reset)	Denotes the end of a master control block		N denotes the nest level (N0 to N7) to be reset.	2

- Trong trường hợp muốn thực hiện một phần hay một số phần của chương trình thì lệnh MC và MCR được dùng để phân đoạn chương trình.
- Lệnh MC gồm có tham số N – chỉ số tầng. Các tầng có thể được sử dụng lồng vào nhau. Tầng 0 là tầng cao nhất và tầng 7 là tầng thấp nhất.
- Lệnh MCR kết thúc một khối lệnh (tầng) có chỉ số N tương ứng. Khi một tầng kết thúc, thì các tầng thấp hơn (được lồng trong đó) cũng bị kết thúc.



3.2. Các lệnh cơ bản – họ FX Mitsubishi



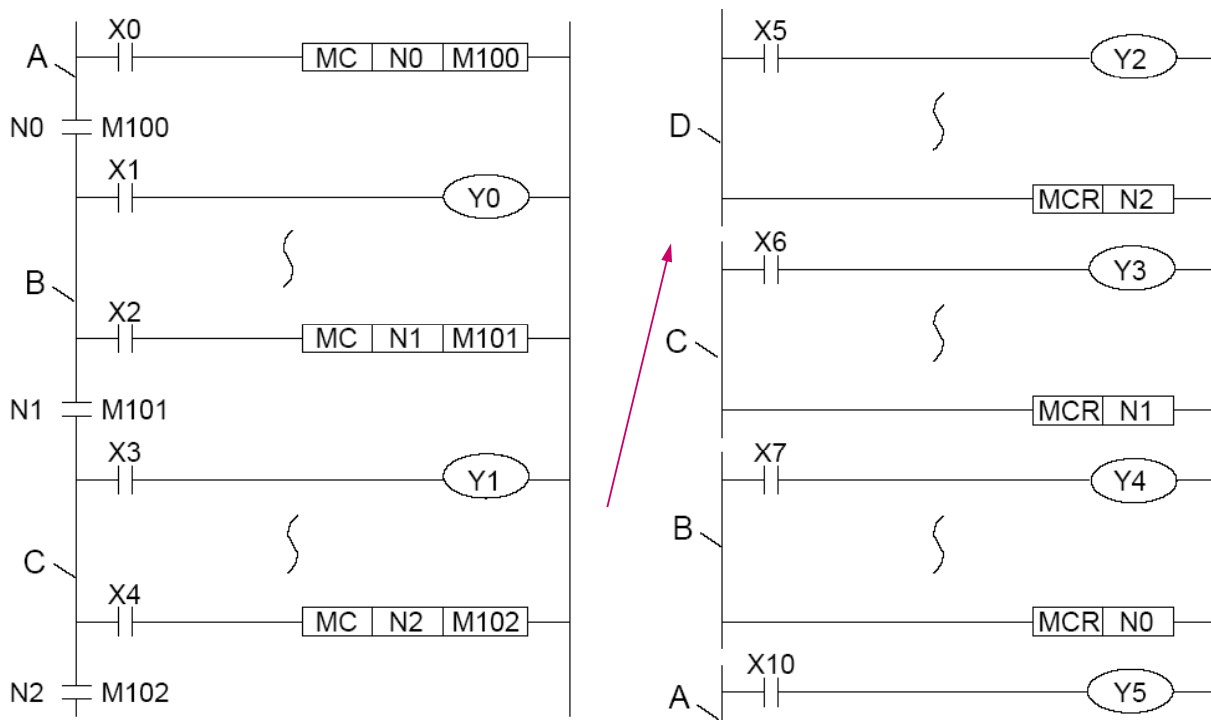
0	LD	X	0
1	MC	N	0
	SP	M	100
4	LD	X	1
5	OUT	Y	0
6	LD	X	2
7	OUT	Y	1
8	MCR	N	0

Note: SP - space key
N - nest level of MC (N0 to N7)

- Khi X0 = ON, tất cả các lệnh giữa MC và MCR được thực hiện.
- Khi X0 = OFF, chẳng có lệnh nào nằm giữa MC và MCR được thực hiện.



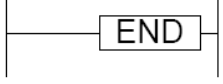
3.2. Các lệnh cơ bản – họ FX Mitsubishi





3.2. Các lệnh cơ bản – họ FX Mitsubishi

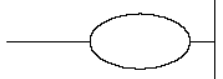

- **Lệnh END:** dùng để đặt khai báo điểm cuối của chương trình.

Mnemonic	Function	Format	Devices	Program steps
END (END)	Forces the current program scan to end		N/A	1

- Đặt lệnh END vào trong chương trình sẽ bắt buộc chương trình đó kết thúc quá trình quét hiện hành và thực hiện các quá trình cập nhật cho ngõ vào và ngõ ra.
- Mặc dù không luôn luôn cần thiết, nhưng sử dụng lệnh END làm cho chương trình được quét nhanh hơn. Do đó làm tăng tốc độ thực thi các lệnh.



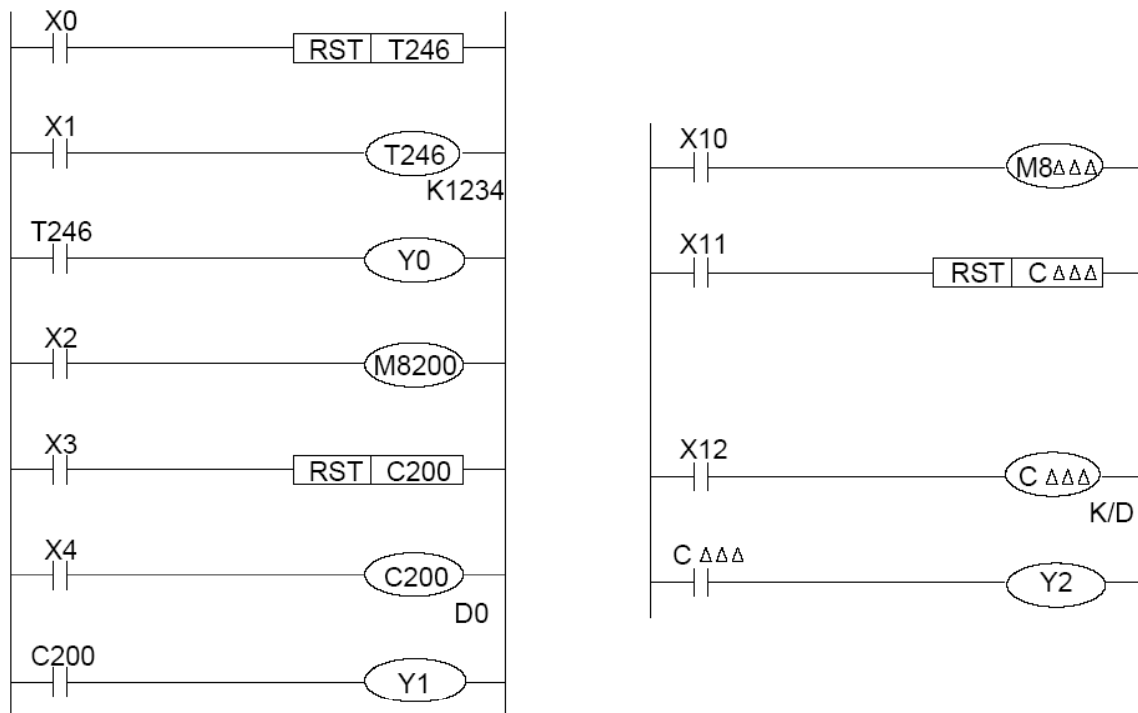
Lập trình với bộ định thì, bộ đếm

Mnemonic	Function	Format	Devices	Program steps
OUT (OUT)	Driving timer or counter coils		T, C	32 bit counters:5 Others: 3
RST (ReSeT)	Resets timer and counter, coils contacts and current values		T, C	

- Lệnh OUT dùng để kích hoạt bộ định thì; và làm tăng / giảm giá trị của bộ đếm.
- Đối với bộ định thì thông thường (không có khả năng tự duy trì), thì điều kiện kích hoạt bộ định thì phải được duy trì trong suốt thời gian định thì. Nếu điều kiện kích hoạt không thỏa thì bộ định thì trở lại trạng thái ban đầu.
- Lệnh RST dùng để đưa bộ định thì (có khả năng tự duy trì) và bộ đếm về trạng thái ban đầu – giá trị thời gian và giá trị đếm bằng không.



Lập trình với bộ định thì, bộ đếm



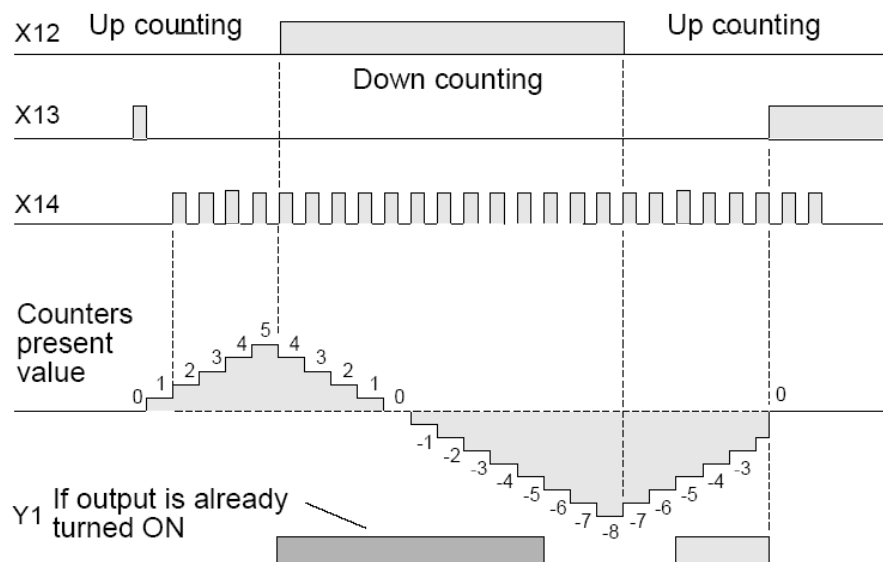
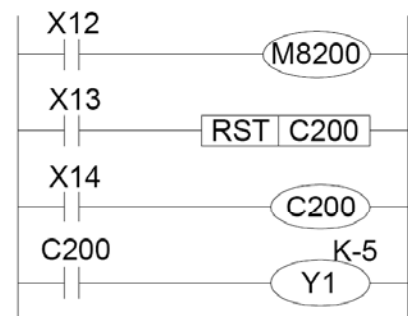
© C.B. Pham

Bộ điều khiển lập trình

3-25



Lập trình với bộ định thì, bộ đếm



© C.B. Pham

3-26