【*Instruction*】

Chapter 1 PLC Ladder Diagram and the Coding Rules of Mnemonic

In this chapter, we would like to introduce you the basic principles of ladder diagram, in addition, the coding rules of mnemonic will be introduced as well, it's essential for the user who use FP-07C as a programming tool. If you are familiar with PLC Ladder Diagram and mnemonic coding rules, you may skip this chapter.

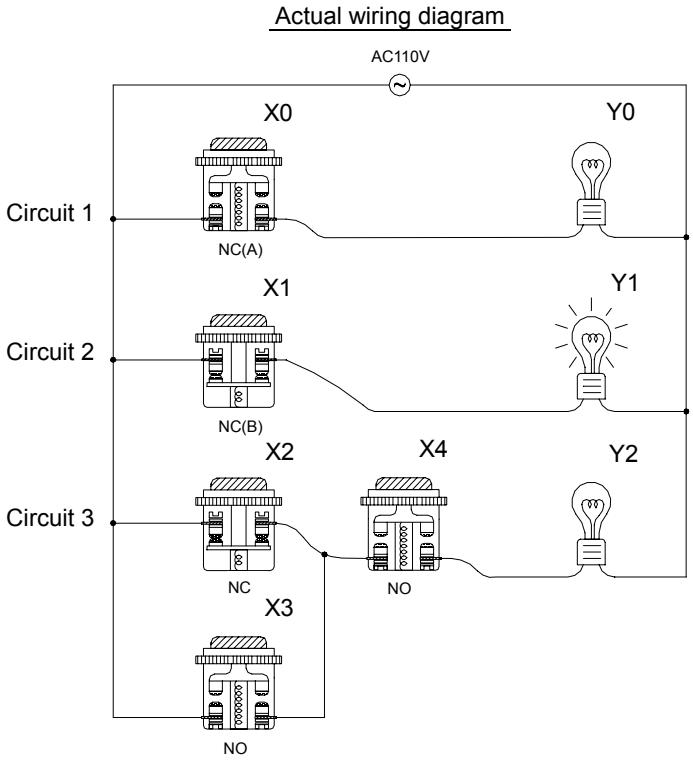
1.1 The Operation Principle of Ladder Diagram

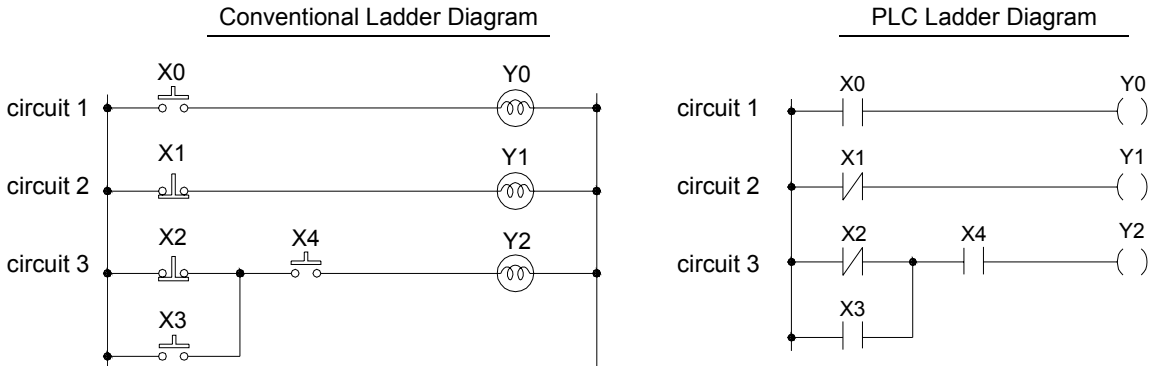
Ladder Diagram is a type of graphic language for automatic control systems it had been used for a long period since World War II. Until today, it is the oldest and most popular language for automatic control systems. Originally there are only few basic elements available such as A-contact (Normally ON), B contact (Normally OFF), output Coil, Timers and Counters. Not until the appearance of microprocessor based PLC, more elements for Ladder Diagram, such as differential contact, retentive coil (refer to page 1-6) and other instructions that a conventional system cannot provide, became available.

The basic operation principle for both conventional and PLC Ladder Diagram is the same. The main difference between the two systems is that the appearance of the symbols for conventional Ladder Diagram are more closer to the real devices, while for PLC system, symbols are simplified for computer display. There are two types of logic system available for Ladder Diagram logic, namely combination logic and sequential logic. Detailed explanations for these two logics are discussed below.

1.1.1 Combination Logic

Combination logic of the Ladder Diagram is a circuit that combines one or more input elements in series or parallel and then send the results to the output elements, such as Coils, Timers/Counters, and other application instructions.





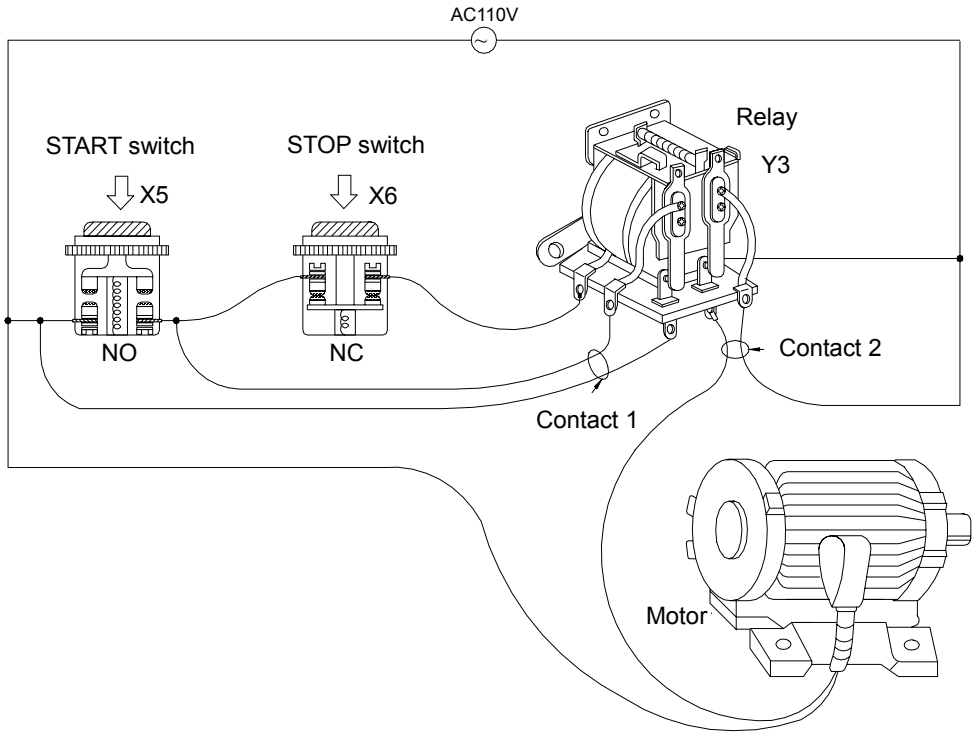
The above example illustrated the combination logic using the actual wiring diagram, conventional Ladder Diagram, and PLC Ladder Diagram. Circuit 1 uses a NO (Normally Open) switch that is also called "A" switch or contact. Under normal condition (switch is not pressed), the switch contact is at OFF state and the light is off. If the switch is pressed, the contact status turns ON and the light is on. In contrast, circuit 2 uses a NC (Normally Close) switch that is also called "B" switch or contact. Under normal condition, the switch contact is at ON state and the light is on. If the switch is pressed, the contact status turns OFF and the light also turns off.

Circuit 3 contains more than one input element. Output Y2 light will turn on under the condition when X2 is closed or X3 switches to ON, and X4 must switch ON too.

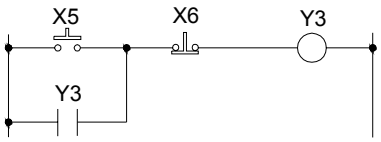
1.1.2 Sequential Logic

The sequential logic is a circuit with feedback control; that is, the output of the circuit will be feedback as an input to the same circuit. The output result remains in the same state even if the input condition changes to the original position. This process can be best explained by the ON/OFF circuit of a latched motor driver as shown in below.

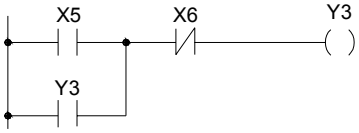
Actual wiring diagram



Conventional Ladder Diagram



PLC Ladder Diagram



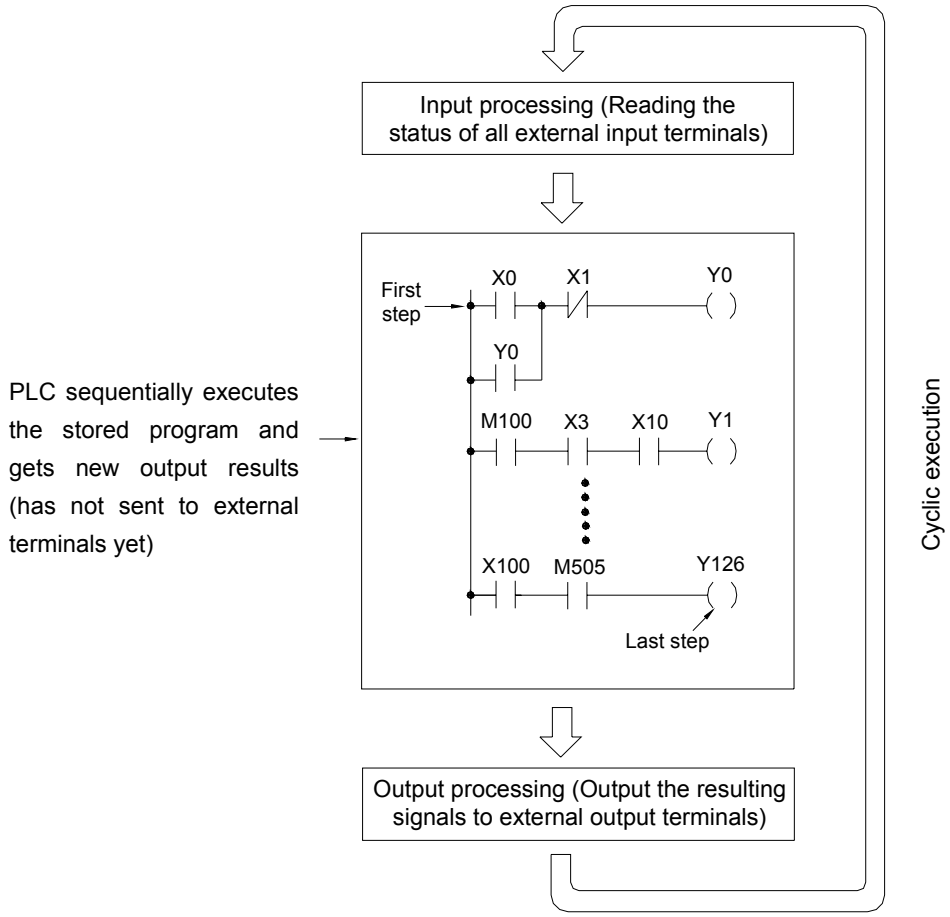
When we first connect this circuit to the power source, X6 switch is ON but X5 switch is OFF, therefore the relay Y3 is OFF. The relay output contacts 1and 2 are OFF because they belong to A contact (ON when relay is ON). Motor does not run. If we press down the switch X5, the relay turns ON as well as contacts 1and 2 are ON and the Motor starts. Once the relay turns ON, if we release the X5 switch (turns OFF), relay can retain its state with the feedback support from contact 1 and it is called Latch Circuit. The following table shows the switching process of the example we have discussed above.

	X5 switch (NO)	X6 switch (NC)	Motor（Relay）status
①	Released	Released	OFF
↓			
②	Pressed	Released	ON
↓			
③	Released	Released	ON
↓			
④	Released	Pressed	OFF
↓			
⑤	Released	Released	OFF

From the above table we can see that under different stages of sequence, the results can be different even the input statuses are the same. For example, let's take a look at stage ① and stage ③ , X5 and X6 switches are both released, but the Motor is ON (running) at stage ③ and is OFF (stopped) at stage ① . This sequential control with the feedback of the output to the input is a unique characteristic of Ladder Diagram circuit. Sometimes we call the Ladder Diagram a "Sequential Control Circuit" and the PLC a "Sequencer". In this section, we only use the A/B contacts and output coils as the example. For more details on sequential instructions please refer to chapter 5 - "Introduction to Sequential Instructions."

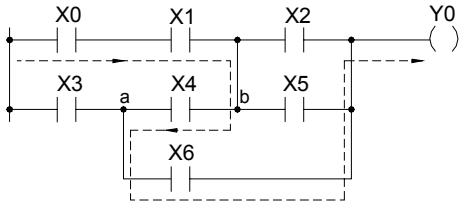
1.2 Differences Between Conventional and PLC Ladder Diagram

Although the basic operation principle for both conventional and PLC Ladder Diagram are the same, but in reality, PLC uses the CPU to emulate the conventional Ladder Diagram operations; that is, PLC uses scanning method to monitor the statuses of input elements and output coils, then uses the Ladder Diagram program to emulate the results which are the same as the results produced by the conventional Ladder Diagram logic operations. There is only one CPU, so the PLC has to sequentially examine and execute the program from its first step to the last step, then returns to the first step again and repeats the operation (cyclic execution). The duration of a single cycle of this operation is called the scan time. The scan time varies with the program size. If the scan time is too long, then input and output delay will occur. Longer delay time may cause big problems in controlling fast response systems. At this time, PLCs with short scan time are required. Therefore, scan time is an important specification for PLCs. Due to the advance in microcomputer and ASIC technologies nowadays the scan speed has been enhanced a great deal. A typical FB_E-PLC takes approximately 0.33 ms for IK steps of contact. The following diagram illustrates the scanning process of a PLC Ladder Diagram.



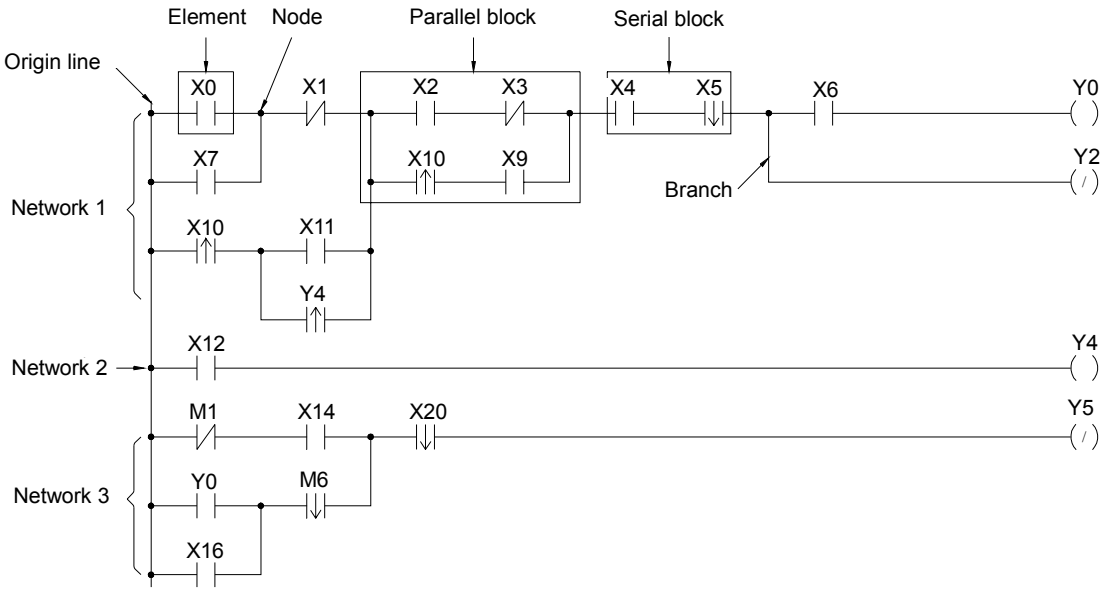
Besides the time scan difference mentioned above, the other difference between the conventional and PLC Ladder Diagram is “Reverse flow” characteristic. As shown in the diagram below, if X0, X1, X4 and X6 are ON, and the remaining elements are OFF. In a conventional Ladder Diagram circuit, a reverse flow route for output Y0 can be defined by the dashed line and Y0 will be ON. While for PLC, Y0 is OFF because the PLC Ladder Diagram scans from left to right, if X3 is off then CPU believes node “a” is OFF, although X4 and node “b” are all ON, since the PLC scan reaches X3 first. In other words, the PLC ladder can only allow left to right signal flow while conventional ladder can flow bi-directional.

Reverse flow of conventional Ladder diagram



1.3 Ladder Diagram Structure and Terminology

Sample Ladder Diagram



(Remark：The maximum size of FBs-PLC network is 16 rows × 22 columns)

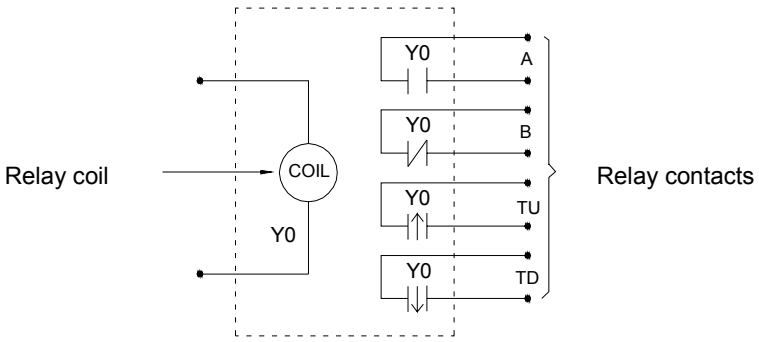
As shown above, the Ladder Diagram can be divided into many small cells. There are total 88 cells (8 rows X 11 columns) for this example Ladder Diagram. One cell can accommodate one element. A completed Ladder Diagram can be formed by connecting all the cells together according to the specific requirements. The terminologies related to Ladder Diagram are illustrated below.

① Contact

Contact is an element with open or short status. One kind of contact is called "Input contact"(reference number prefix with X) and its status reference from the external signals (the input signal comes from the input terminal block). Another one is called "Relay contact" and its status reflects the status of relay coil (please refer to ②). The relation between the reference number and the contact status depends on the contact type. The contact elements provided by FB series PLC include: A contact, B contact, up/down differential (TU/TD) contacts and Open/Short contacts. Please refer to ④ for more details.

② Relay

Same as the conventional relay, it consists of a Coil and a Contact as shown in the diagram below.

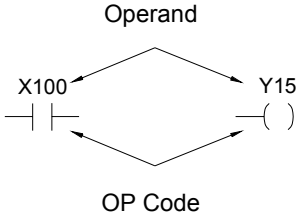


We must energize the coil of relay first (using OUT instruction) in order to turn on the relay. After the coil is energized, its contact status will be ON too. As shown in the example above, if Y0 turns ON, then the relay contact A is ON and contact B is OFF, TU contact only turns ON for one scan duration and TD contact is OFF. If Y0 turns OFF, then the relay contact A is ON and contact B is ON, TU contact is OFF and TD contact only turns ON for one scan duration (Please refer to chapter 5 “Introduction to Sequential Instructions” for operations of A,B,TU and TD contacts).

There are four types of FB-PLC relays, namely Y△△△（output relay）, M△△△△（internal relay）, S△△△（step relay）and TR△△（temporary relay）. The statuses of output relays will be sent to the output terminal block.

③ Origin-line: The starting line at the left side of the Ladder Diagram.

④ Element: Element is the basic unit of a Ladder Diagram. An element consists of two parts as shown in the diagram below. One is the element symbol which is called “OP Code” and another is the reference number part which is called “Operand”.



Element type	Symbol	Mnemonic instructions	Remark
A Contact （Normally OPEN）	□△△△△ ⊣ ⊢	(ORG、LD、AND、OR) □△△△△	□ can be X、Y、M、S、T、C（please refer to section 3.2）
B Contact （Normally CLOSE）	□△△△△ ⊣/⊢	(ORG、LD、AND、OR) NOT □△△△△	
Up Differential Contact	□△△△△ ⊣↑⊢	(ORG、LD、AND、OR) TU □△△△△	□ can be X、Y、M、S
Down Differential Contact	□△△△△ ⊣↓⊢	(ORG、LD、AND、OR) TD □△△△△	
Open Circuit Contact	—◦ ◦—	(ORG、LD、AND、OR) OPEN	
Short Circuit Contact	•——•	(ORG、LD、AND、OR) SHORT	
Output Coil	□△△△△ —( )—	OUT □△△△△	□ can be Y、M、S
Inverse Output Coil	□△△△△ —(/)—	OUT NOT □△△△△	
Latching Output Coil	Y△△△ —(L)—	OUT L Y△△△	

Remark：please refer to section 3.2 for the ranges of X、Y、M、S、T and C contacts. Please refer to section 5.2 for the characteristics of X、Y、M、S、T and C contacts.

There are three special sequential instructions, namely OUT TRn, LD TRn and FOn, which were not displayed on the Ladder Diagram. Please refer to section 1.6 “Using the Temporary Relay” and section 5.1.4 “Function Output FO”.

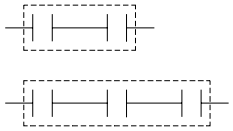
⑤ Node: The connection point between two or more elements（please refer to section 5.3）

⑥ Block: a circuit consists of two or more elements.

There are two basic types of blocks:

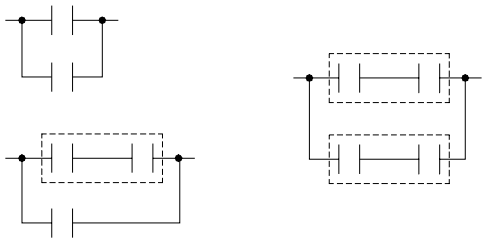
- Serial block：Two or more elements are connected in series to form a single row circuit.

Example:



- Parallel block: Parallel block is a type of a parallel closed circuit formed by connecting elements or serial blocks in parallel.

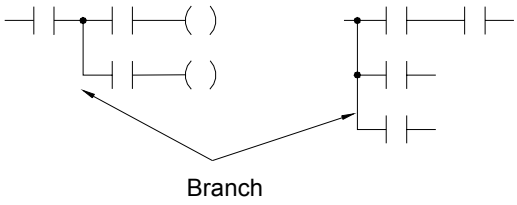
Example:



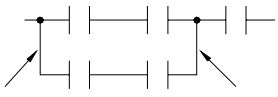
Remark: Complicated block can be formed by the combination of the single element, serial blocks and parallel blocks. When design a Ladder Diagram with mnemonic entry, it is necessary to break down the circuits into element, serial, and parallel blocks. Please refer to section 1.5.

⑦ Branch: In any network, branch is obtained if the right side of a vertical line is connected with two or more rows of circuits.

Example:

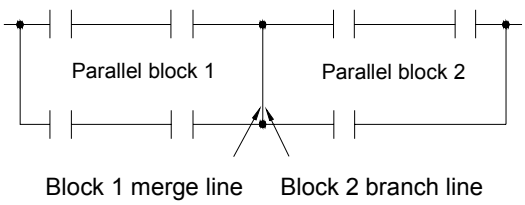


Merge line is defined as another vertical line at the right side of a branch line that merges the branch circuits into a closed circuit (forming a parallel block). This vertical line is called “Merge line”.



If both the right and the left sides of the vertical line are connected with two or more rows of circuits, then it is both a branch line and a merge line as shown in the example below.

Example:



⑧ Network: Network is a circuit representing a specified function. It consists of the elements, branches, and blocks. Network is the basic unit in the Ladder Diagram which is capable of executing the completed functions, and the program of Ladder Diagram is formed by connecting networks together. The beginning of the network is the origin line. If two circuits are connected by a vertical line, then they belong to the same network. If there is no vertical line between the two circuits, then they belong to two different networks. Figure 1, shows three (1～3) networks.

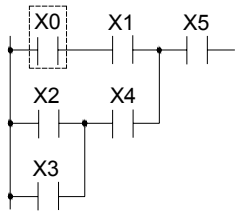
1.4 The Coding Rules of Mnemonic

(Users of WinProladder can skip this section)

It's very easy to program FB-PLC with WinProladder software package, just key-in the ladder symbols as they appear on your CRT screen directly to form a ladder diagram program. But for the users who are using FPC-07 to program FB-PLC they have to translate ladder diagram into mnemonic instructions by themselves. Since FPC-07 only can input program with mnemonic instruction, this section till section 1.6 will furnish you with the coding rules to translate ladder diagrams into mnemonic instructions.

- The program editing directions are from left to right and from top to bottom. Therefore the beginning point of the network must be at the upper left corner of the network. Except the function instruction without the input control, the first instruction of a network must begin with the ORG and only one ORG instruction is permissible per network. Please refer to section 6.1.1 for further explanations.

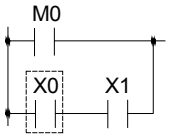
Example:




```
ORG    X    0
AND    X    1
LD     X    2
OR     X    3
AND    X    4
ORLD
AND    X    5
```

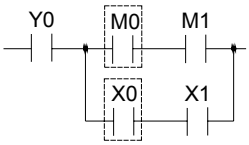
- Using LD instruction for connecting vertical lines (origin line or branch line) except at the beginning of the network.

Example 1:




```
ORG    M    0
LD     X    0
AND    X    1
ORLD
```

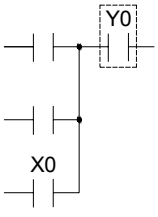
Example 2:




```
AND    Y    0
LD     M    0
AND    M    1
LD     X    0
AND    X    1
ORLD
```

Remark 1: Using the AND instruction directly if only one row of elements is serially connected to the branch line.

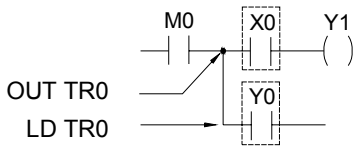
Example:




```
AND    X    0
ORLD
AND    Y    0
```

Remark 2: Also using the AND instruction directly if an OUT TR instruction has been used at a branch line to store the node statuses.

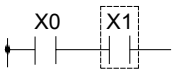
Example:




```
AND     M   0
OUT TR  0
AND     X   0
OUT     Y   1
LD  TR  0
AND     Y   0
```

● Using AND instruction for serial connection of a single element.

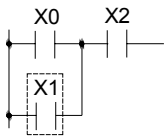
Example:




```
ORG     X   0
AND     X   1
```

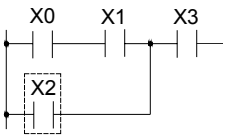
● Using OR instruction for parallel connection of a single element.

Example:




```
ORG     X   0
OR      X   1
AND     X   2
```

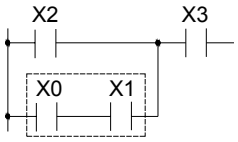
Example:




```
ORG     X   0
AND     X   1
OR      X   2
AND     X   3
```

● If the parallel element is a serial block, ORLD instruction must be used.

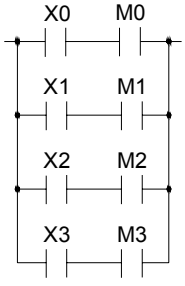
Example:




```
ORG     X   2
LD      X   0
AND     X   1
ORLD
AND     X   3
```

Remark：If more than two blocks are to be connected in parallel, they should be connected in a top to bottom sequence. For example, block 1 and block 2 should be connected first, then connect block 3 to it and so on.

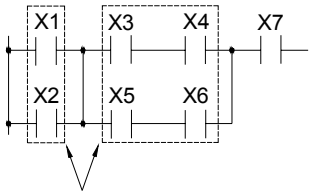
Example:




```
LD      X   0
AND     M   0
LD      X   1
AND     M   1
ORLD
LD      X   2
AND     M   2
ORLD
LD      X   3
AND     M   3
```

● ANDLD instruction is used to connect parallel blocks in series.

Example:



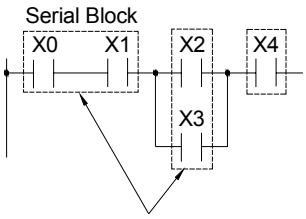
Must use ANDLD instruction

```
ORG   X   1
OR    X   2
LD    X   3
AND   X   4
LD    X   5
AND   X   6
ORLD
ANDLD
AND   X   7
```

● The ANDLD instruction must be used if the element or serial block is in front of the parallel block. If the parallel block is in front of the element or serial block, AND instruction can be used to connect all parts together.

Example:

ANDLD instruction is not necessary

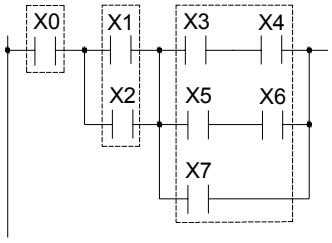


Must use ANDLD instruction

```
ORG   X   0
AND   X   1
LD    X   2
OR    X   3
ANDLD
AND   X   4
```

Remark: If there are more than two blocks are to be connected serially, they should be connected in a top to bottom sequence. For example, block 1 and 2 should be connected first, then connect block 3 to it and so on.

Example:




```
ORG   X   0
LD    X   1
OR    X   2
ANDLD
LD    X   3
AND   X   4
LD    X   5
AND   X   6
ORLD
OR    X   7
ANDLD
```

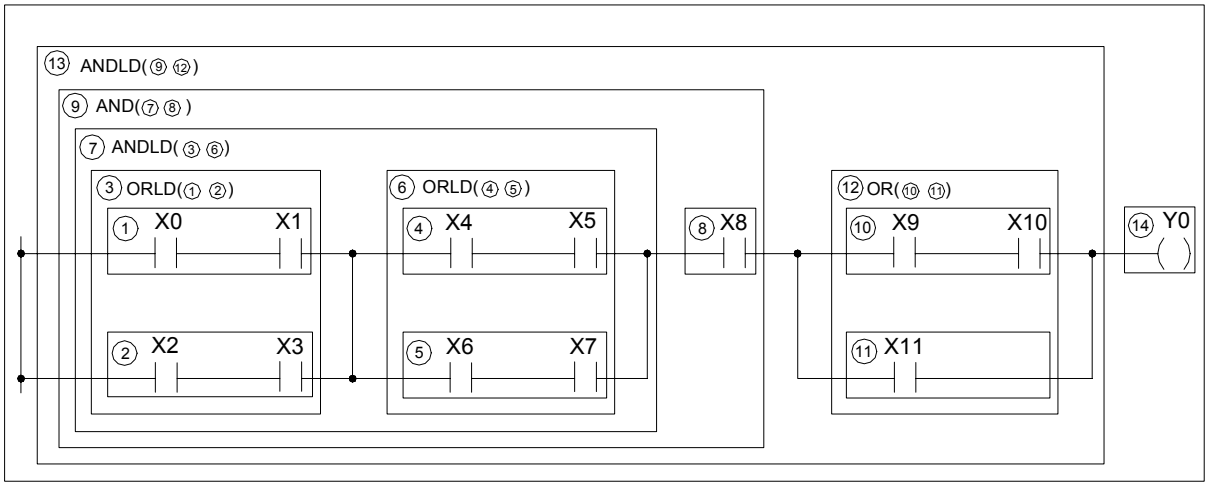
● The output coil instruction （OUT）can only be located at end of the network (the right end) and no other elements can be connected to it afterwards. The output coil can not connect to the origin line directly. If you want to connect the output coil to the origin line, connect it serially with a short circuit contact.

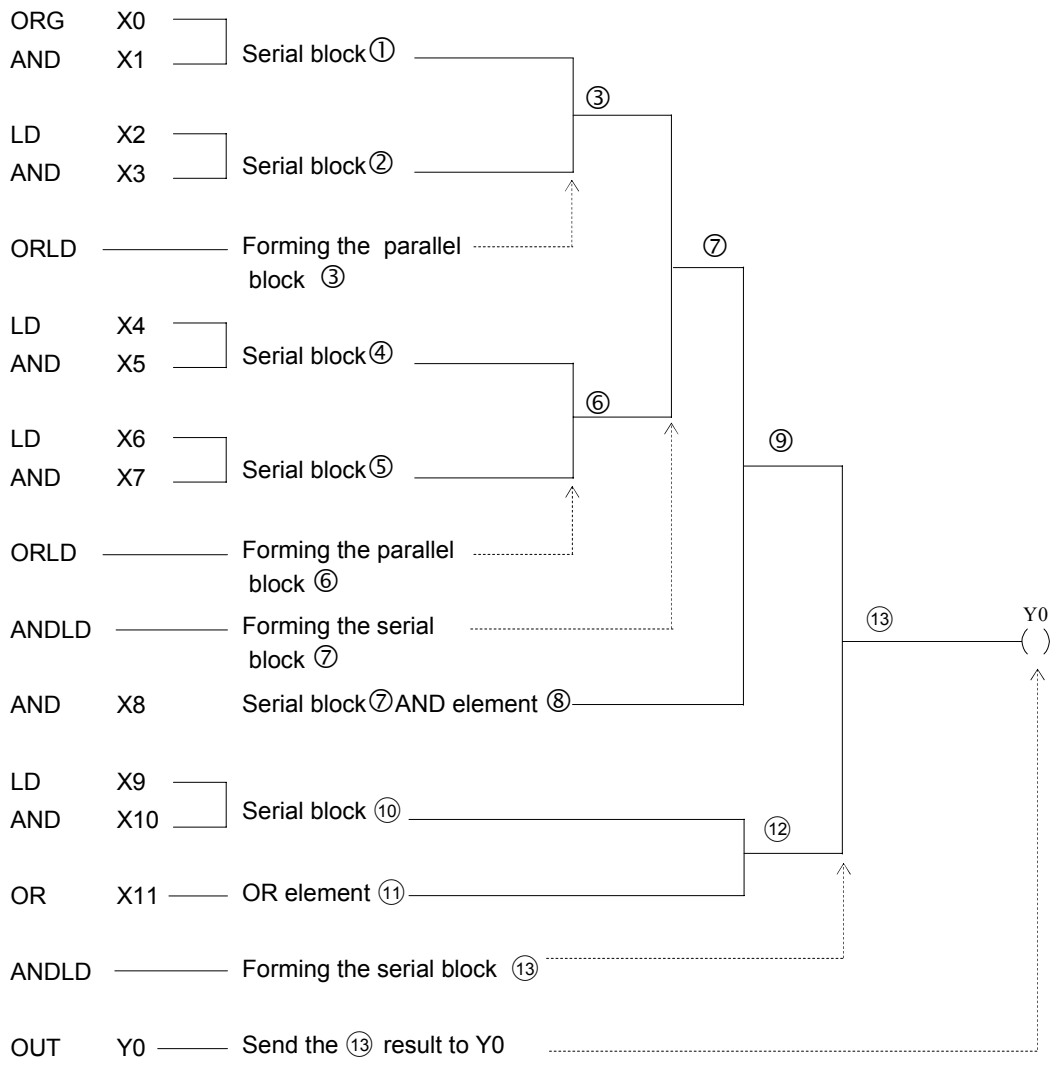

```
ORG SHORT
OUT          Y   0
```

1.5 The De-Composition of a Network (Users of WinProladder can skip this section)

The key process of de-composition of a network is to separate the circuits that appear between two vertical lines into independent elements and serial blocks, then coding those elements and serial blocks according to the mnemonic coding rules and then connect them (with ANDLD or ORLD instruction) from left to right and top to bottom to form a parallel or a serial-parallel blocks, and finally to form a complete network.

Sample diagram:





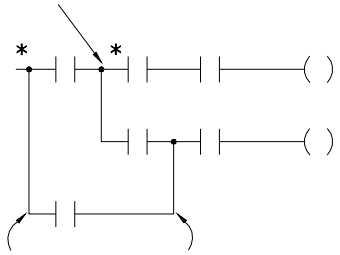
1.6 Using Temporary Relays (Users of WinProladder can skip this section)

The network de-composition method for mnemonic coding demonstrated in section 1.5 does not apply to the branched circuit or branched block. In order to input the program using the method shown in section 1.5, It must first to store the statuses of branched nodes in temporary relays. The program design should avoid having branched circuit or branched block as much as possible. Please refer the next section “Program Simplification Techniques”. Two situations that must use the TR are described at below.

- Branched circuit: Merge line does not exist at the right side of the branch line or there is a merge line at the right side of the branch line but they are not in the same row.

Example： * indicates setting of TR relay

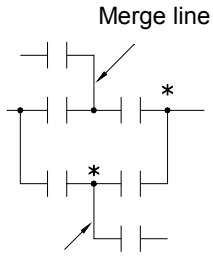
Without merge line



Although this branch has merge lines but they are not in the same row, so this is also a branched circuit

- Branched block：The horizontal parallel blocks with a branch in one of the blocks.

Example：



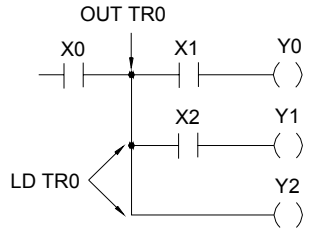
Remark 1: The OUT TR instruction must be programmed at the top of the branched point. LD TRn instruction is used at the starting point of the circuits after second rows of the branch line for regaining the branch line status before you can connect any element to the circuits. AND instruction must be used to connect the first element after OUT TRn or LD TRn instruction. LD instruction is not allowed in this case.

Remark 2: A network can have up to 40 TR points and the TR number can not be used repeatedly in the same network. It is recommended to use the numbers 1,2,3… with sequence. The TR number must be the same in the same branch line. For example, if a branch line uses OUT TR0, then starting from row 2, LD TR0 must be used for connection.

Remark 3: If the branch line of a branched circuit or a branched block is the origin line, then ORG or LD instructions can be used directly and TR contact is not necessary.

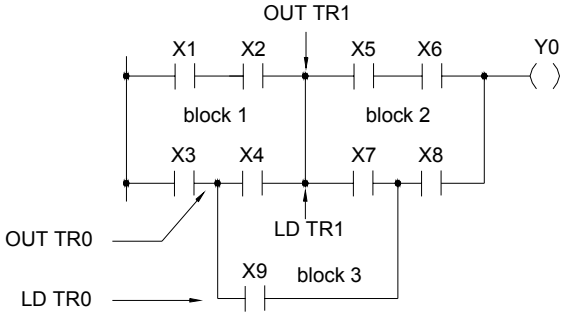
Remark 4: If any one of the branched circuit rows is not connected to the output coil (there are serially connected elements in between), and other circuits also exist after the second row, a TR instruction must be used at the branch points.

Example:




```
AND     X   0
OUT TR  0
AND     X   1
OUT     Y   0
LD  TR  0       <- Begins from row 2
AND     X   2
OUT     Y   1
LD  TR  0       <- Begins from row 3
OUT     Y   2
```

Example:

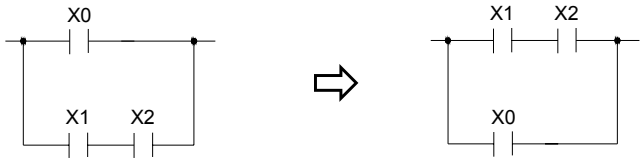



```
ORG     X   1
AND     X   2
LD      X   3
OUT TR  0
AND     X   4
ORLD
OUT TR  1
AND     X   5       <- Uses AND Instruction after TR instruction
AND     X   6
LD  TR  1           <- Uses LD TR instruction to return to TR branch line
AND     X   7
LD  TR  0
AND     X   9       <- Uses AND instruction after TR instruction
ORLD
AND     X   8
ORLD
OUT     Y   0
```

- The above sample diagram shows a typical example of connecting two parallel blocks in series. Block 3 is formed when the element X9 is introduced into the network and the two parallel blocks become the branched blocks.
- TR instruction is not necessary because the（*）point is the origin line.
- If have already used TR relay to connect two blocks serially, then ANDLD instruction is not necessary.

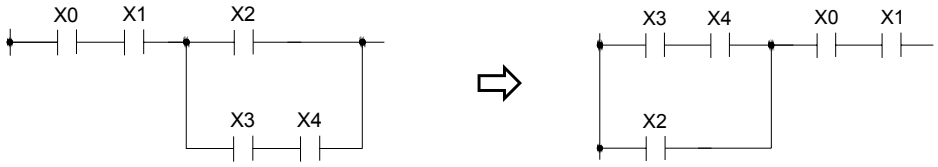
1.7 Program Simplification Techniques

- If a single element is connected in parallel to a serial block, The ORLD instruction can be omitted if the serial block is connected on top of this single element.




```
LD      X   0          LD      X   1
LD      X   1          AND     X   2
AND     X   2          OR      X   0
ORLD
```

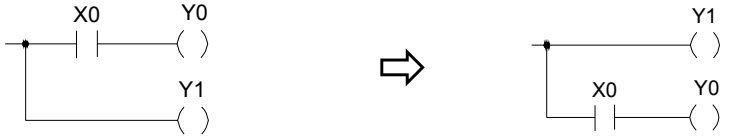
- When a single element or a serial block is connected in parallel with a parallel block, ANDLD instruction can be omitted if put the parallel block in front.



ORG	X	0
AND	X	1
LD	X	2
LD	X	3
AND	X	4
ORLD		
ANDLD		

ORG	X	3
AND	X	4
OR	X	2
AND	X	0
AND	X	1

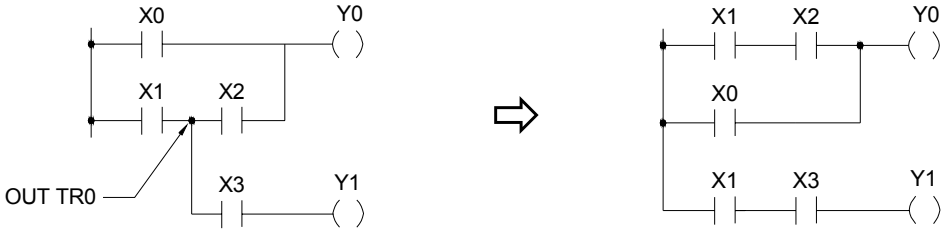
- If the branch node of a branch circuit is directly connected to the output coil, this coil could be located on top of the branch line (first row) to reduce the code.



OUT TR	0	
AND	X	0
OUT	Y	0
LD TR	0	
OUT	Y	1

OUT	Y	1
AND	X	0
OUT	Y	0

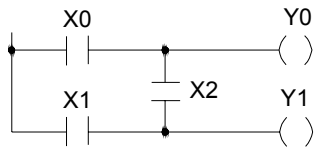
- The diagram shown below indicates the TR relay and the ORLD instruction can be omitted.



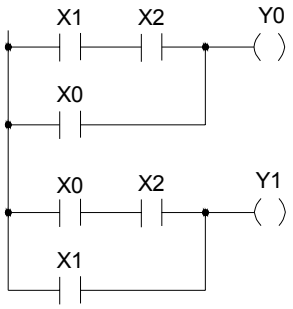
ORG	X	0
LD	X	1
OUT TR	0	
AND	X	2
ORLD		
OUT	Y	0
LD TR	0	
AND	X	3
OUT	Y	1

ORG	X	1
AND	X	2
OR	X	0
OUT	Y	0
ORG	X	1
AND	X	3
OUT	Y	1

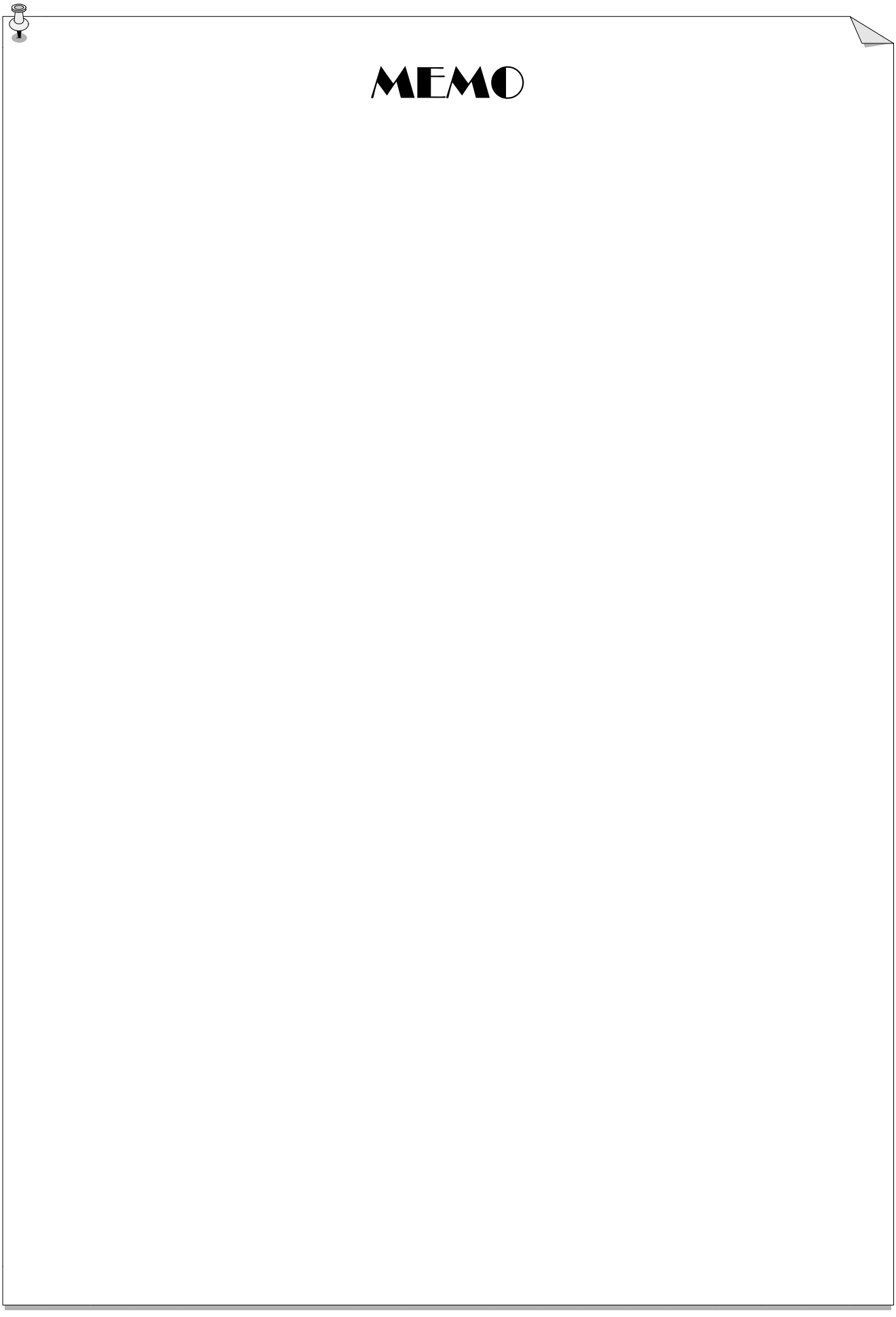
● Conversion of the bridge circuit



This network structure is not allowed in PLC program




```
ORG   X   1
AND   X   2
OR    X   0
OUT   Y   0
ORG   X   0
AND   X   2
OR    X   1
OUT   Y   1
```


MEMO