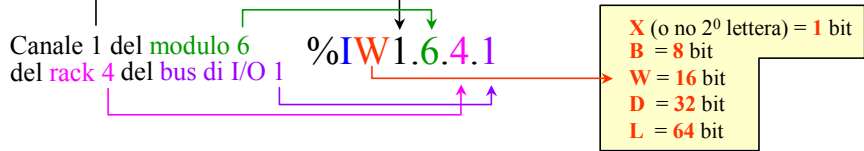


## Ladder diagram (LD) o diagramma a contatti

(12) Scrivendo il programma che comanda il PLC rispettando la norma IEC 1131-3 si assicura la sua portabilità su PLC di marche diverse. In base a tale norma, un PLC (Programmable Logic Controller) è un computer destinato all'uso in ambito industriale, dove legge degli ingressi e poi eroga delle uscite che vanno ad alimentare o a togliere l'alimentazione ai motori usati in quell'impianto industriale

Un **diagramma a contatti** (in inglese un **Ladder Diagram, LD, diagramma a scala** perché si presenta come una scala a pioli) è un linguaggio di programmazione grafica che ha come elemento base il **contatto**, che può essere **normalmente aperto (NO)** o **normalmente chiuso (NC)**. Questo simbolo --| |-- o questo --! |-- è il simbolo di un contatto **normalmente aperto (NO)**, ovvero che è **aperto se la variabile che ci dice il suo stato (%I1.0) vale 0**, quindi se essa è **off, non attivata**. Con la lineetta in mezzo --| / |-- oppure --! / |-- è un contatto **normalmente chiuso (NC)**, ovvero che risulta **chiuso se la variabile associata al contatto (%I1.1) vale 0**, quindi se essa è **off, non attivata**. I nomi delle variabili sono in realtà più complessi perché specificano anche la lunghezza e la locazione fisica o logica dell'elemento:



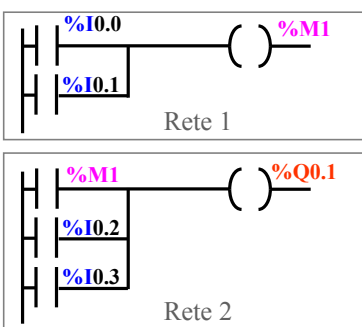
La linea verticale a sinistra è la **barra di alimentazione positiva** e la linea a destra (che può anche essere omessa nel disegno) è la **barra di alimentazione negativa**.

I contatti portano o non portano il + dell'alimentazione alla **bobina** di un relè, che ha l'altro terminale collegato fisso al - della tensione di alimentazione.

Il nome di una **variabile di uscita** comincia con **Q**. **%M1.0 Variabile di memoria**

Se comincia con **M** è una **variabile di memoria** (non è un ingresso e non è un'uscita)

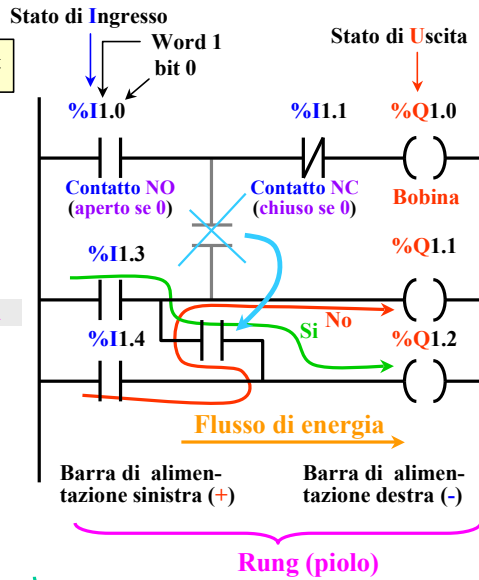
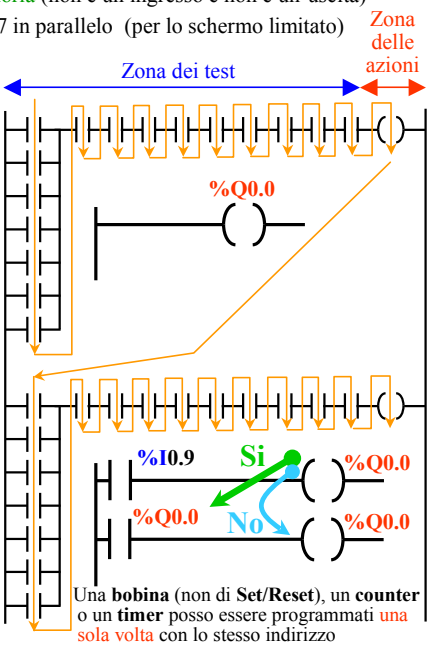
Sono ammessi fino a 10 contatti in serie e 7 in parallelo (per lo schermo limitato)



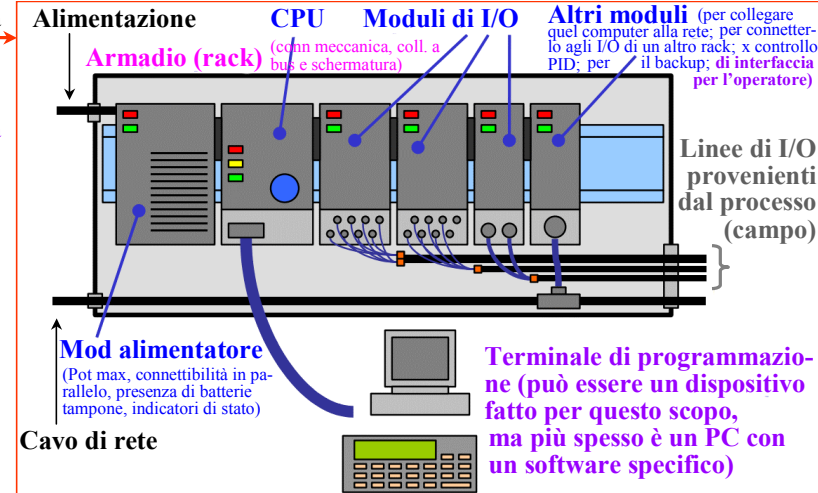
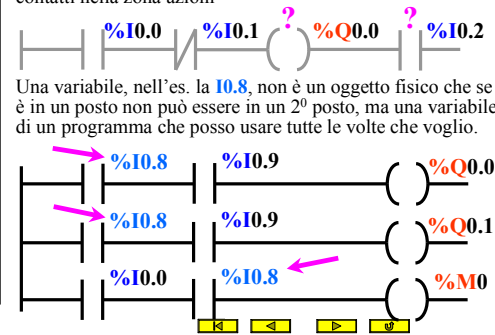
Il PLC:

- prima **legge tutti gli ingressi** (che quindi restano uguali per tutta una esecuzione del programma);
- Poi **esegue tutti i rung**, a partire dall'alto, e **ogni volta assegna un valore alla bobina** (che lo conserverà fino alla nuova esecuzione del programma).
- Poi **aggiorna tutte le uscite**

A questo punto ricomincia il ciclo successivo, tornando a leggere tutti gli ingressi



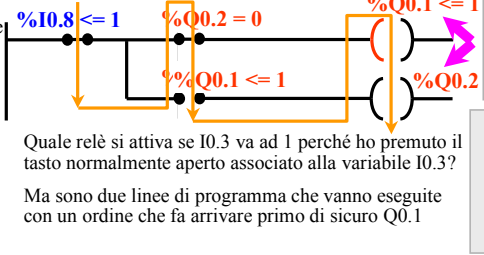
No bobine nella zona test e no contatti nella zona azioni



- | |-- oppure --! |-- Contatto normalmente aperto (NO) (OFF se V=0)
- | / |-- oppure --! / |-- Contatto normalmente chiuso (NC) (ON se V=0)
- | P |-- o --! P |-- ON se V ha una transizione positiva (fronte di salita)
- | N |-- o --! N |-- ON se V ha una transizione negativa (fronte di discesa)
- ( )-- Bobina OFF se la sua Variabile = 0
- ( / )-- Bobina ON se la sua Variab. = 0

Le due barre di alimentazione sono chiamate anche **montanti**, perché se il disegno somiglia a una scala allora i lati della scala sono i montanti

Inoltre nulla mi vieta di usare la variabile Q0.1, che chiamandosi Q è associata ad una bobina, anche per controllare un contatto che va a influire su una seconda bobina che si chiama Q0.2, e nulla mi vieta di usare Q0.2 per controllare un contatto che va a influire su Q0.1



Ma sono due linee di programma che vanno eseguite con un ordine che fa arrivare primo di sicuro Q0.1

