

Ανάπτυξη προγράμματος σε γλώσσα λίστα εντολών



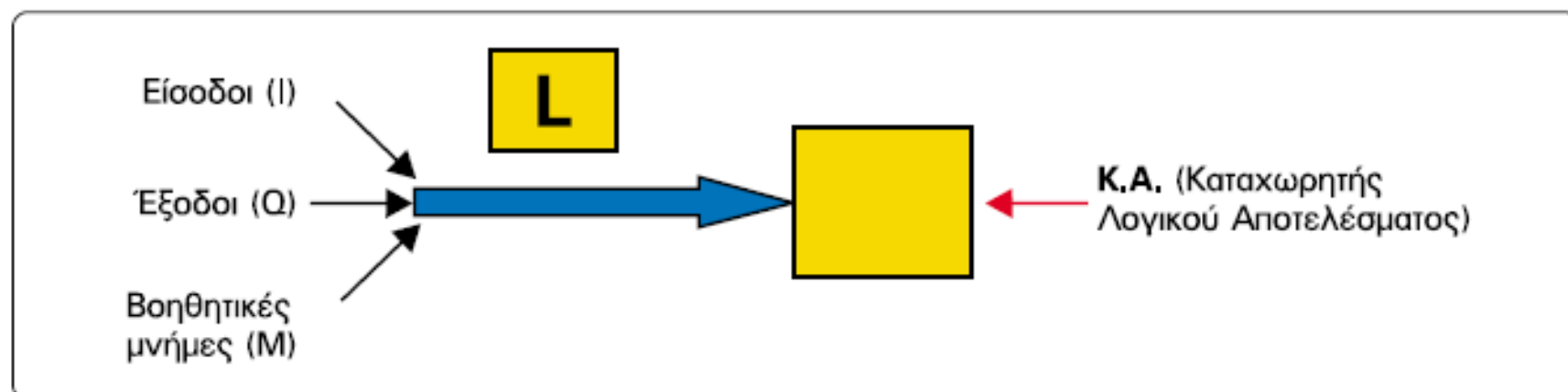
Μορφή εντολής στη γλώσσα λίστα εντολών

Κάθε εντολή του προγράμματος αποτελείται από δύο μέρη (σχήμα 18). Το πρώτο μέρος καθορίζει την *ενέργεια* την οποία θα εκτελέσει το PLC, δηλαδή χαρακτηρίζει την ίδια την εντολή. Το δεύτερο μέρος καθορίζει την *παράμετρο*, δηλαδή καθορίζει σε ποια είσοδο, έξοδο, βοηθητική μνήμη κ.λπ. αναφέρεται η ενέργεια της εντολής.

Η εντολή L (Load)

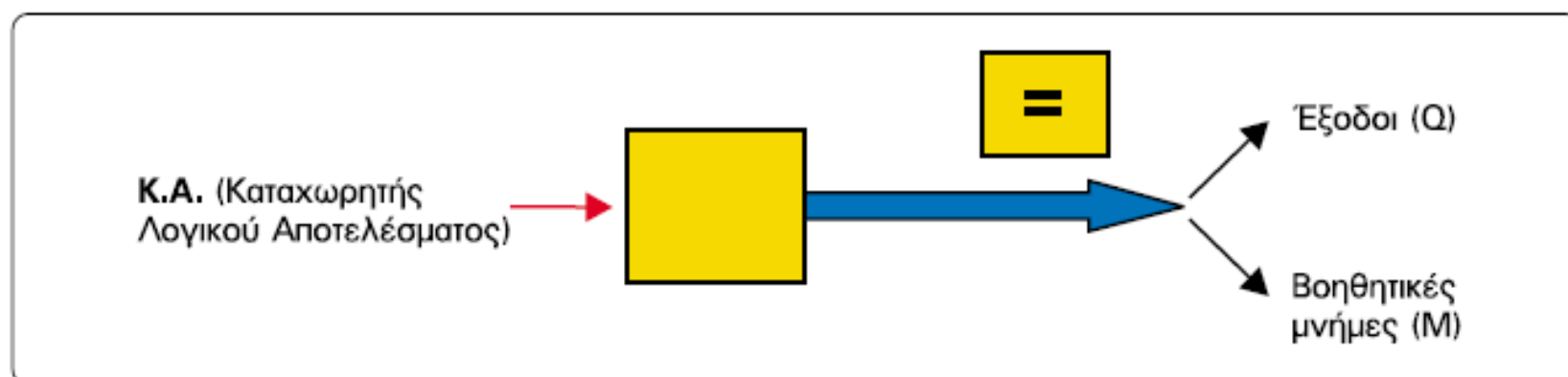
Το πρόγραμμα το οποίο αντιστοιχεί σε μια πύλη λογικού κυκλώματος (ή κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού) ξεκινά με την εντολή L (Load, “φόρτωση”). Το PLC με την εντολή Load διαβάζει τη λογική κατάσταση (“0” ή “1”) μιας εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού κ.λπ., και τη “φορτώνει” σε ένα “καταχωρητή” (μια ειδική θέση μνήμης) τον οποίο θα ονομάζουμε **Καταχωρητή Λογικού Αποτελέσματος (Κ.Α.)**.

Η εντολή L μπορεί να αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Π.χ. L I 0.1, L Q 0.2, L M 0.1.



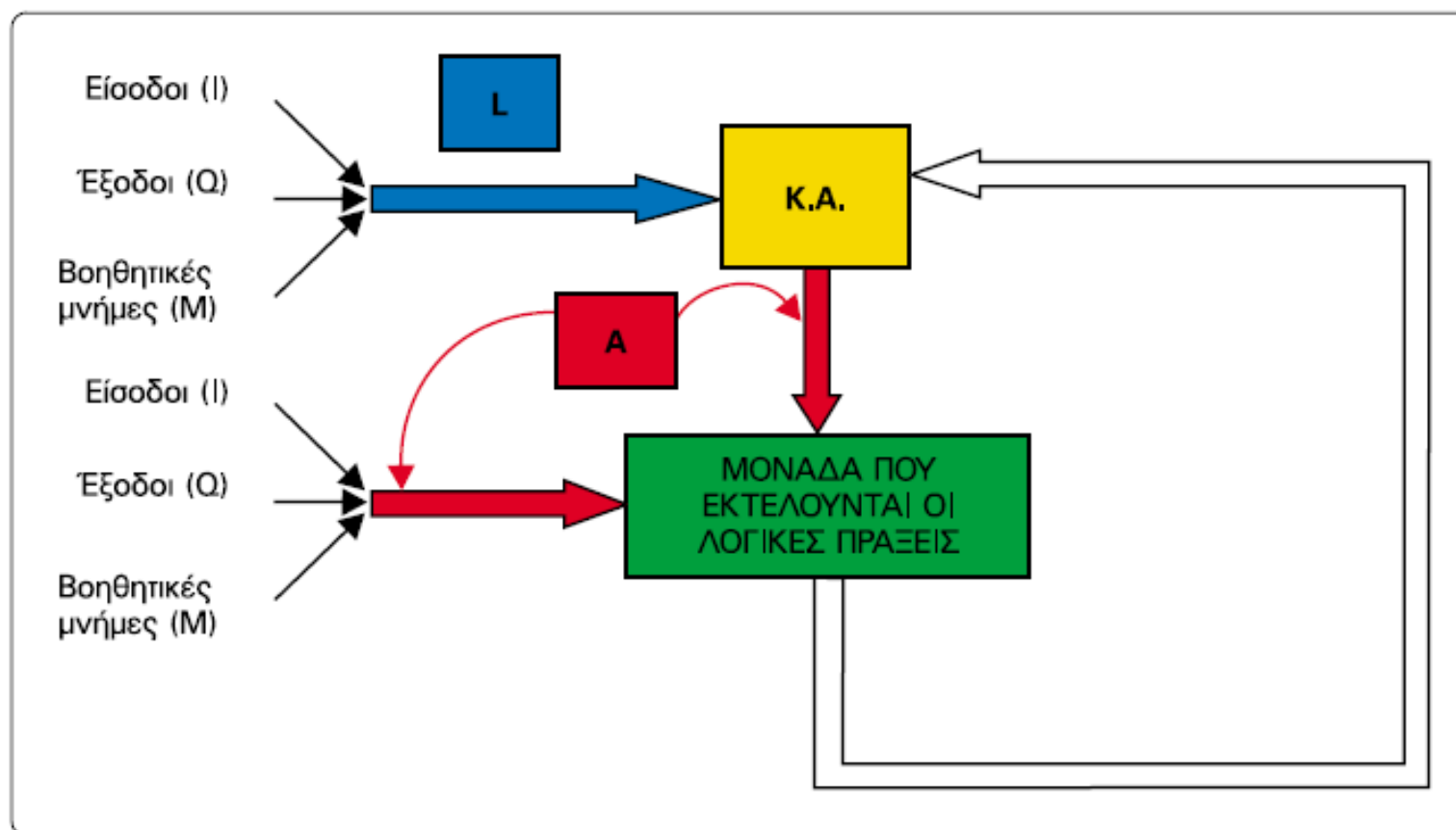
Η εντολή = (ίσον)

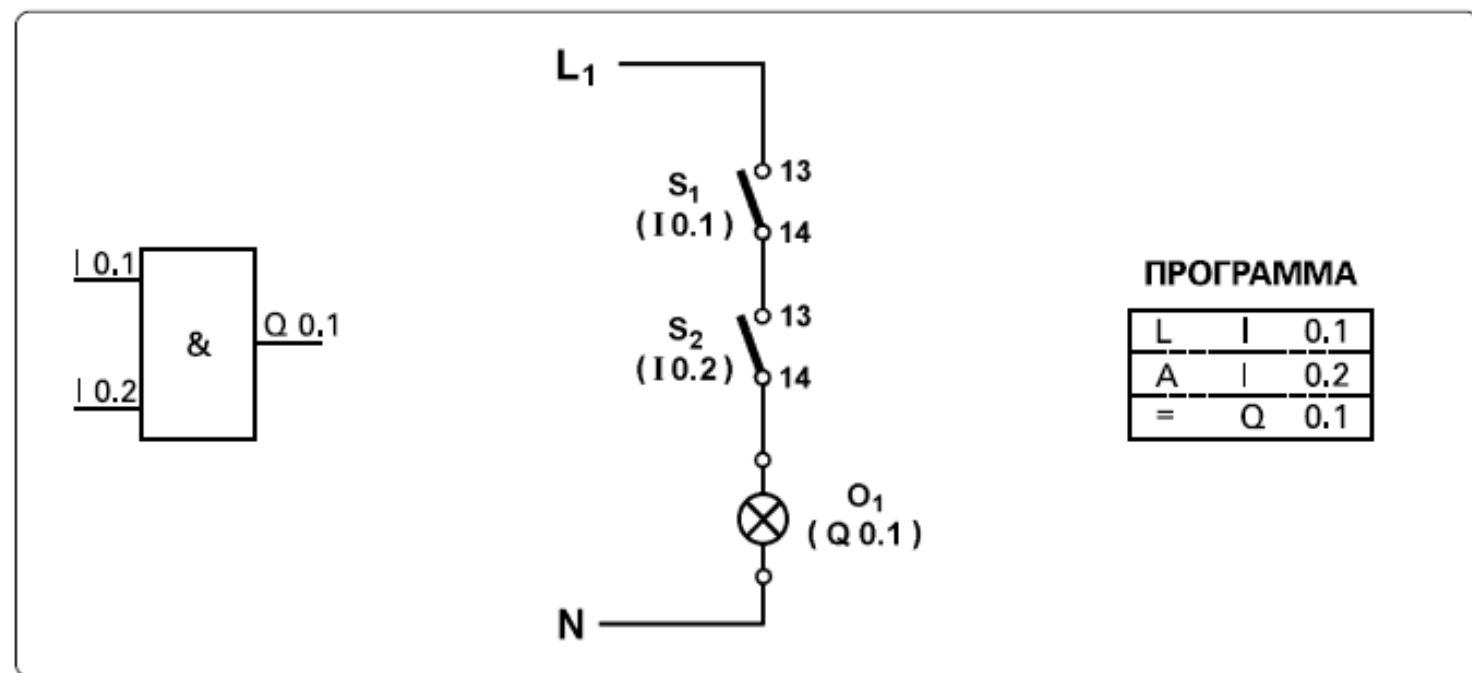
Το πρόγραμμα που αντιστοιχεί σε μια πύλη λογικού κυκλώματος (ή κλάδο ηλεκτρολογικού κυκλώματος αυτοματισμού) καταλήγει πάντα με την εντολή = (ίσον). Η εντολή αναφέρεται σε εξόδους και βοηθητικές μνήμες. Το PLC με την εντολή = μεταφέρει στις εξόδους ή στις βοηθητικές μνήμες το περιεχόμενο του Καταχωρητή Λογικού Αποτελέσματος. Π.χ. = Q 1.2, = M 0.1



Η εντολή A (AND)

Η εντολή **A** υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη AND. Η εντολή AND αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά κ.λπ. Η λογική πράξη γίνεται μεταξύ της λογικής κατάστασης της εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού, κ.λπ. το οποίο αναφέρεται στην εντολή και του περιεχομένου του Κ.Α. Το αποτέλεσμα επιστρέφει στον Κ.Α. Προσέξτε το σχήμα 21, το οποίο αποδίδει πολύ καλά τον τρόπο που το PLC εκτελεί τη λογική πράξη AND.





Επεξήγηση:

Γραμμή 1: Η εντολή Load φέρνει στον Κ.Α. τη λογική κατάσταση (λογική τιμή "0" ή "1") της εισόδου I 0.1.

Γραμμή 2: Η εντολή AND εκτελεί τη λογική πράξη μεταξύ της λογικής κατάστασης της εισόδου I 0.2 και του περιεχομένου του Κ.Α. (δηλαδή της λογικής κατάστασης της εισόδου I 0.1), και το λογικό αποτέλεσμα επιστρέφει στον Κ.Α.

Γραμμή 3: Η εντολή = οδηγεί το περιεχόμενο του Κ.Α. (που είναι το λογικό αποτέλεσμα της πράξης AND) στην έξοδο Q 0.1.

L	I	0.1
A	I	0.2
=	Q	0.1

Προϋποθέσεις

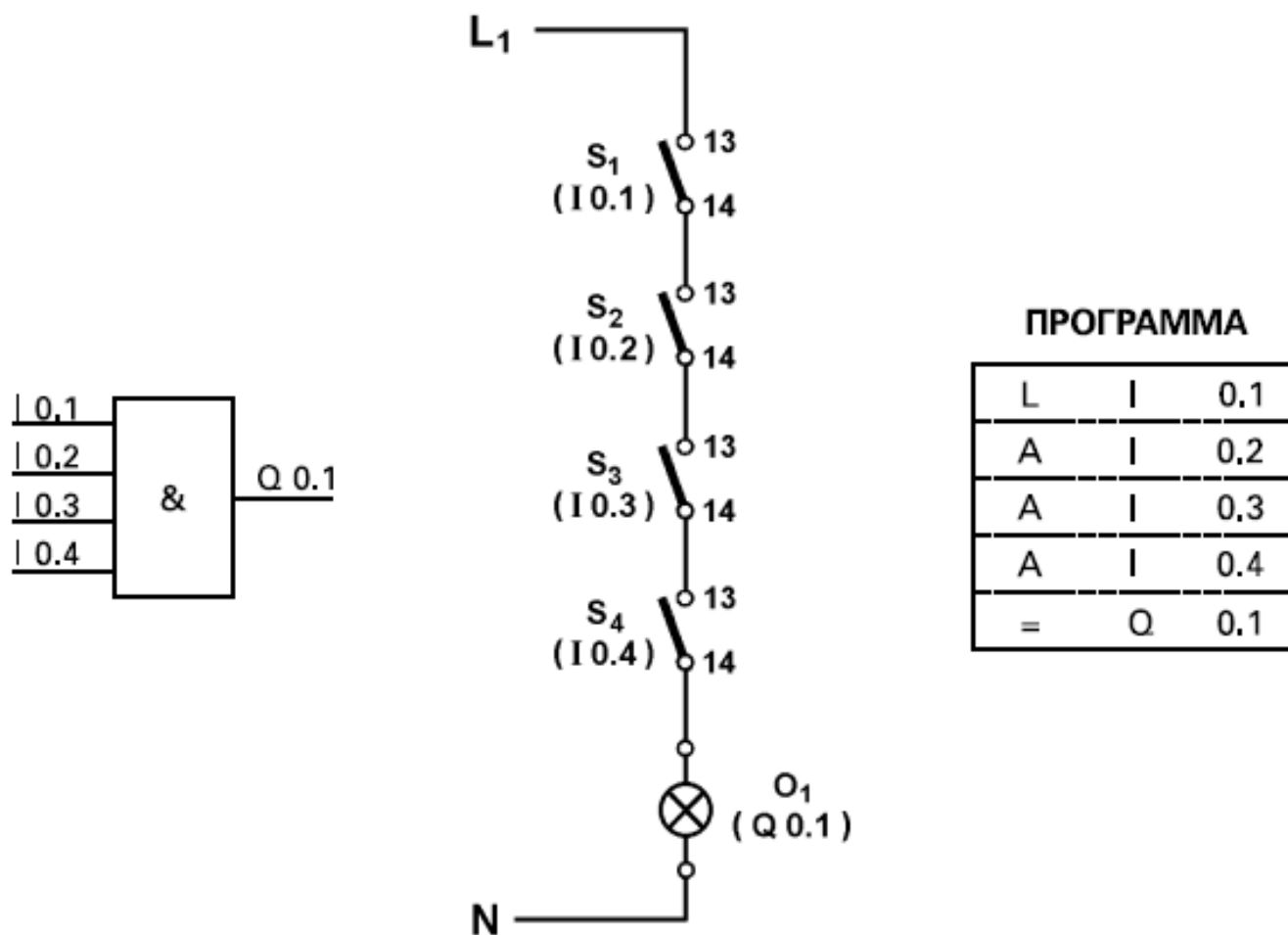
ΑΝ | 0.1 είναι "1" ΚΑΙ | 0.2 είναι "1"

Διέγερση (αποτέλεσμα)

ΤΟΤΕ Q 0.1 είναι "1"

Πύλη AND πολλών εισόδων.

Το πρόγραμμα για μια πύλη AND πολλών εισόδων είναι σχεδόν το ίδιο με το πρόγραμμα της πύλης δύο εισόδων.



Επεξήγηση:

Αφού γνωρίζουμε ότι στον Κ.Α. βρίσκεται πάντα το αποτέλεσμα της προηγούμενης λογικής πράξης, εύκολα μπορούμε να καταλάβουμε τη διαδικασία.

Γραμμή 1: Η εντολή Load φέρνει στον Κ.Α. τη λογική τιμή της εισόδου I 0.1.

Γραμμή 2: Η εντολή AND εκτελεί τη λογική πράξη μεταξύ της λογικής τιμής της εισόδου I 0.2 και του περιεχομένου του Κ.Α. (δηλαδή της λογικής τιμής της εισόδου I 0.1) και το αποτέλεσμα επιστρέφει στον Κ.Α.

Γραμμή 3: Η εντολή AND εκτελεί τη λογική πράξη AND μεταξύ του προηγούμενου λογικού αποτελέσματος (που βρίσκεται στον Κ.Α.) και της λογικής τιμής της εισόδου I 0.3. Το αποτέλεσμα επιστρέφει και πάλι στον Κ.Α.

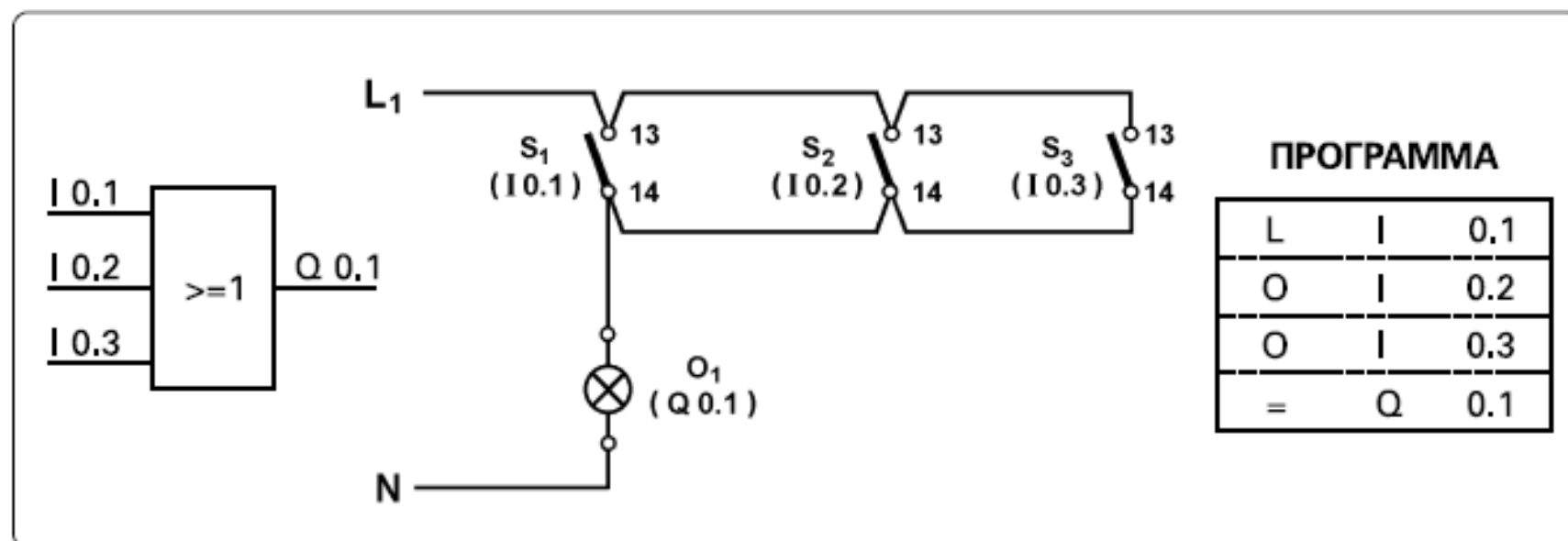
Γραμμή 4: Επαναλαμβάνεται η διαδικασία της εντολής της γραμμής 3, δηλαδή εκτελείται η λογική πράξη AND μεταξύ του νέου περιεχομένου του Κ.Α. και της λογικής τιμής της εισόδου I 0.4.

Γραμμή 5: Το περιεχόμενο του Κ.Α., δηλαδή η λογική τιμή, που είναι το τελικό λογικό αποτέλεσμα της προηγούμενης ομάδας εντολών, μεταφέρεται στην έξοδο Q 0.1.

Η εντολή O (OR)

Η εντολή OR υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη OR. Η εντολή OR αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελείται η εντολή AND.

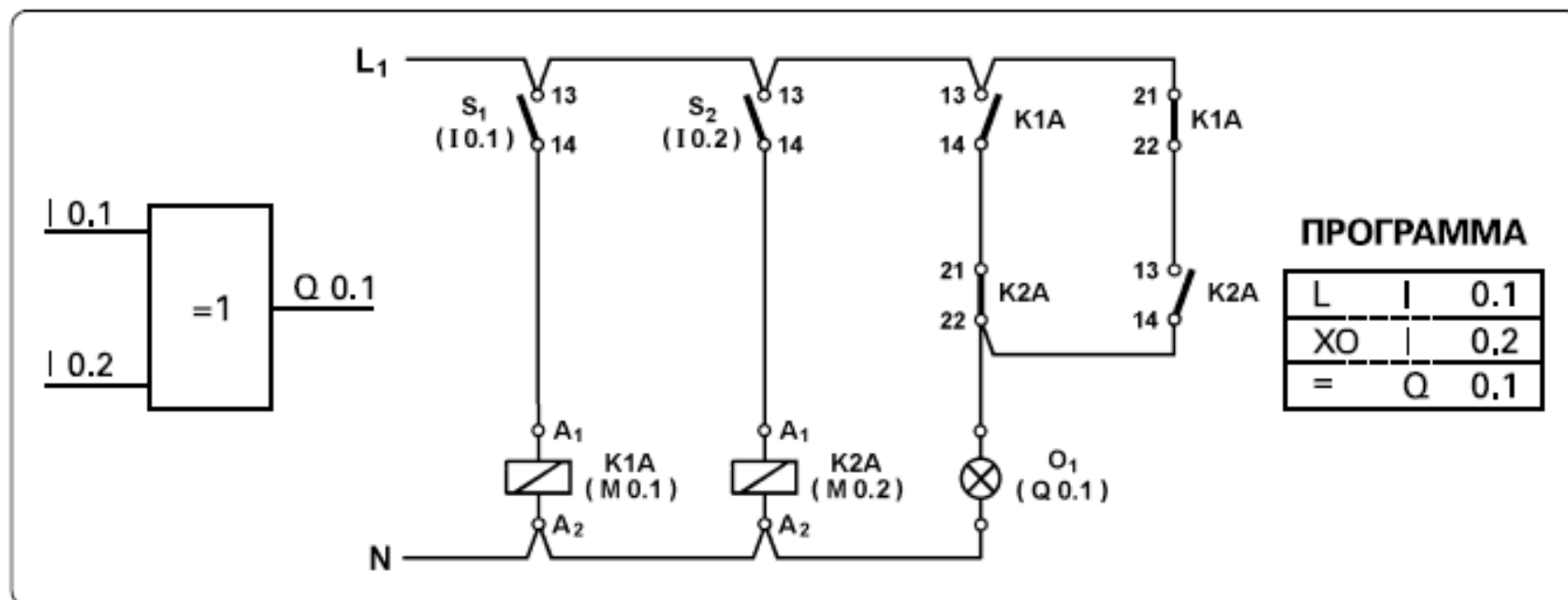
Παράδειγμα προγράμματος:



Η εντολή XO (EXOR)

Η εντολή EXOR υπαγορεύει στο PLC να εκτελέσει τη λογική πράξη EXOR. Η εντολή EXOR αναφέρεται σε εισόδους, εξόδους, βοηθητικές μνήμες, χρονικά, κ.λπ. Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελούνται οι εντολές AND και OR. Υπενθυμίζουμε ότι η λογική πύλη EXOR αποτελεί σύνθετο λογικό κύκλωμα και όχι βασική πύλη.

Παράδειγμα προγράμματος:



Αντιστοιχία συμβολισμών των ηλεκτρολογικών κυκλωμάτων αυτοματισμού με τους συμβολισμούς των PLC.

Όπως ήδη έχουμε δει, στα PLC διακρίνουμε τις εισόδους, στις οποίες δίνουμε τις ονομασίες I 0.0, I 0.1, I 0.2, I 0.3, ..., τις βοηθητικές μνήμες, τις οποίες ονομάζουμε M 0.0, M 0.1, M 0.2, M 0.3, ..., και τις εξόδους, τις οποίες ονομάζουμε Q 0.0, Q 0.1, Q 0.2, Q 0.3, Είναι επίσης γνωστοί οι συμβολισμοί που χρησιμοποιούμε στα ηλεκτρολογικά κυκλώματα αυτοματισμού (S_1 , S_2 ,... για διακόπτες, μπουτόνς και αισθητήρες, K1M, K2M,... για ηλεκτρονόμους ισχύος, K1A, K2A,... για βοηθητικούς ηλεκτρονόμους κ.λπ.). Στα παραδείγματα που θα ακολουθήσουν, όταν αναφερόμαστε σε ηλεκτρολογικά κυκλώματα αυτοματισμού θα υπάρχουν στα σχέδια των κυκλωμάτων και οι ονομασίες των αντίστοιχων στοιχείων του PLC.

Υπενθυμίζουμε ότι οι διακόπτες, τα μπουτόνς και οι αισθητήρες είναι είσοδοι σε ένα PLC, οι ηλεκτρονόμοι ισχύος είναι έξοδοι του PLC και οι βοηθητικοί ηλεκτρονόμοι είναι βοηθητικές μνήμες του PLC.

Η εντολή N (NOT)

Δεν πρόκειται για ανεξάρτητη λογική εντολή όπως οι τρεις εντολές λογικών πράξεων AND, OR, EXOR. Η εντολή **N** είναι συμπλήρωμα όλων των εντολών που είδαμε προηγουμένως. Μπορούμε επομένως να πούμε ότι έχουμε όχι μια εντολή NOT, αλλά πέντε εντολές:

LN → LOAD NOT

AN → AND NOT

ON → OR NOT

XON → XOR NOT

= N → = NOT

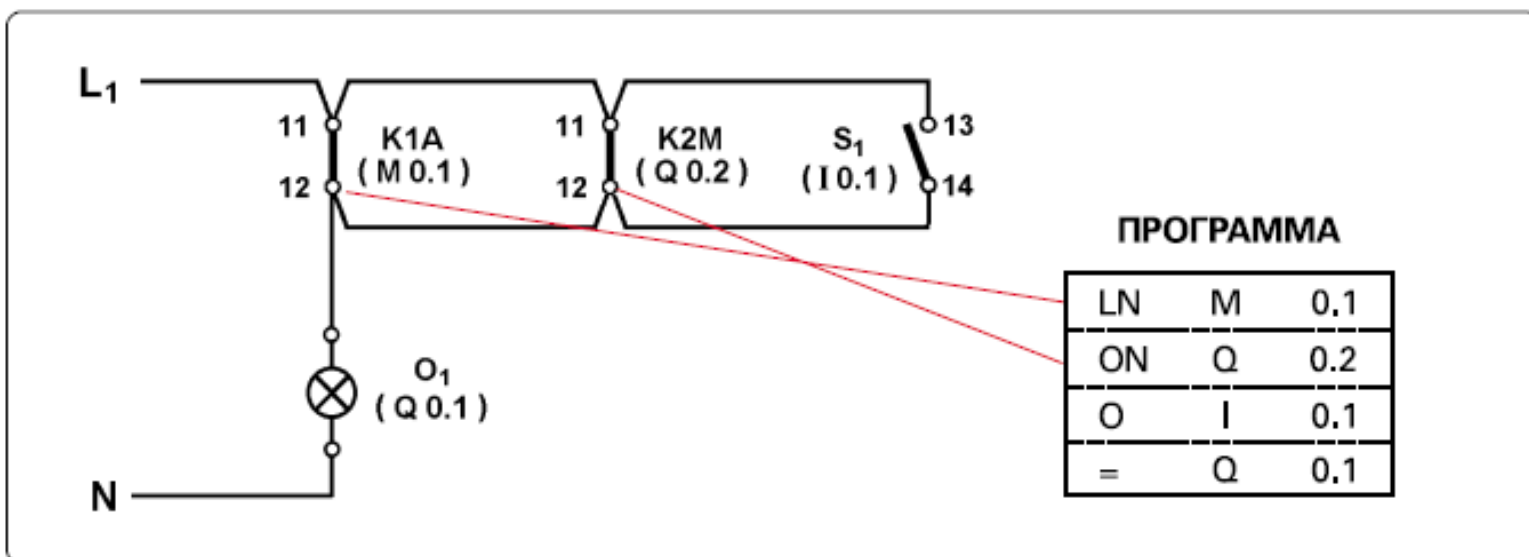
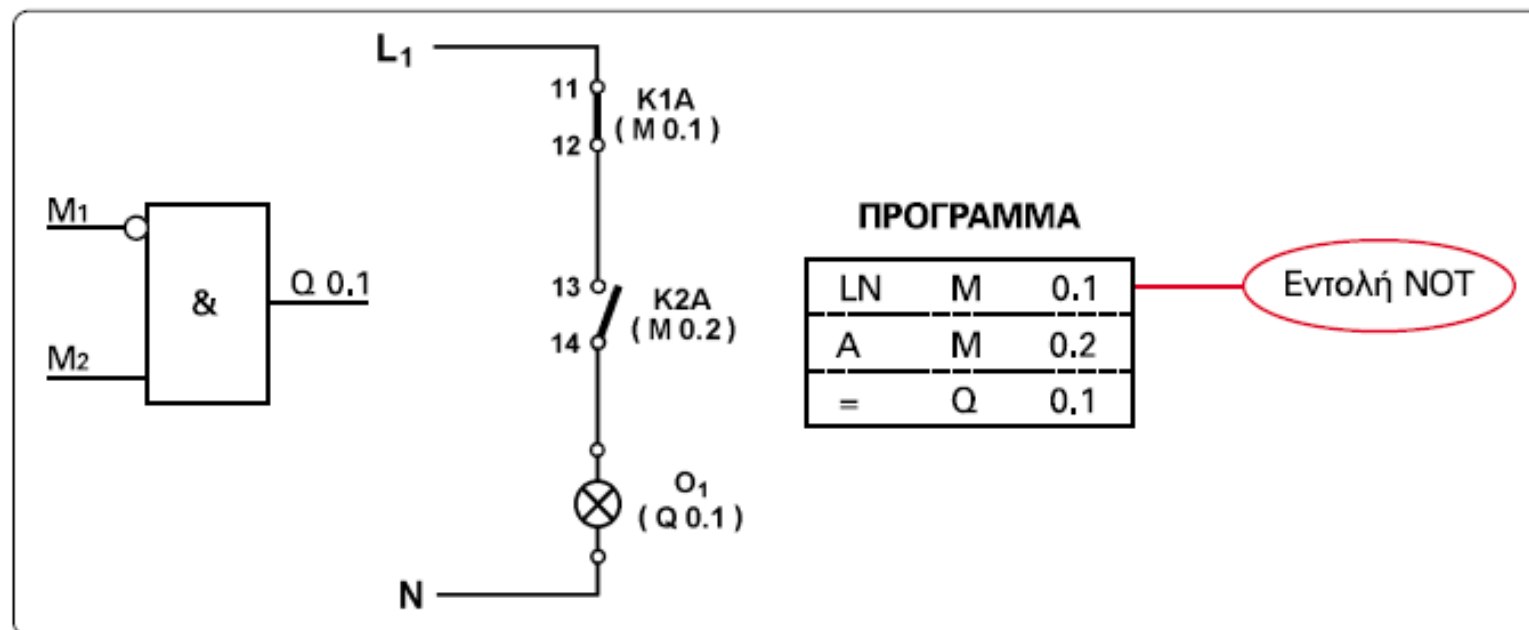
Η εντολή NOT χρησιμοποιείται ως εξής:

- Όταν έχουμε το σχέδιο λογικού κυκλώματος του αυτοματισμού.

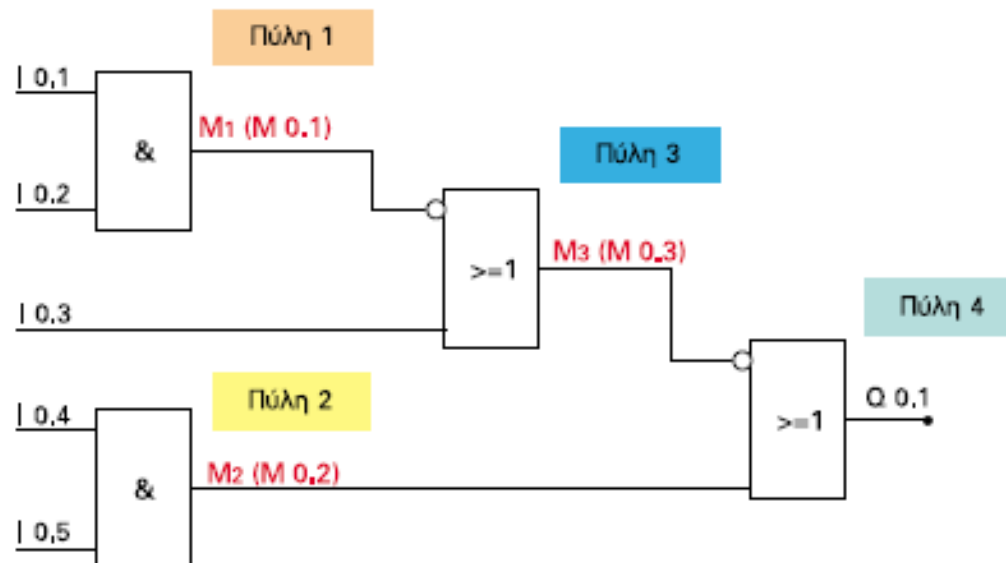
Όταν στο λογικό κύκλωμα έχουμε πύλη NOT, στην εντολή που θα είχαμε αν δεν υπήρχε η πύλη NOT (A, O, XO, L, =) προσθέτουμε το N.

- Όταν έχουμε το ηλεκτρολογικό σχέδιο του αυτοματισμού.

Όταν στο ηλεκτρολογικό σχέδιο έχουμε “κανονική κλειστή” επαφή ηλεκτρονόμου, στην εντολή που θα είχαμε αν η επαφή του ηλεκτρονόμου ήταν “κανονικά ανοιχτή” (A, O, L) προσθέτουμε το N. **ΠΡΟΣΟΧΗ!** Αυτό δεν ισχύει όταν έχουμε “κανονική κλειστή” επαφή εξωτερικού διακόπτη, μπουτόν ή αισθητήρα.



Παράδειγμα 1. Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω λογικό κύκλωμα.



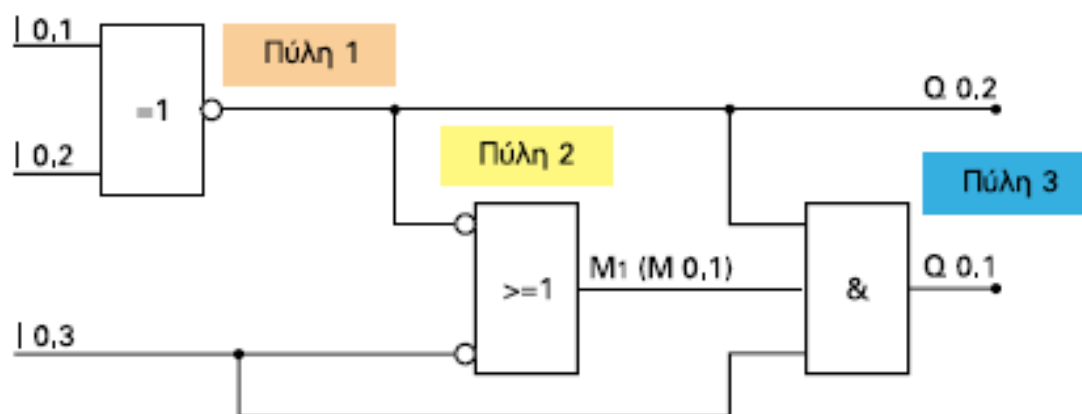
ΠΡΟΓΡΑΜΜΑ

L	I	0,1
A	I	0,2
=	M	0,1
L	I	0,4
A	I	0,5
=	M	0,2
LN	M	0,1
O	I	0,3
=	M	0,3
LN	M	0,3
O	M	0,2
=	Q	0,1

Πύλη 1
Πύλη 2
Πύλη 3
Πύλη 4

Σχήμα 31

Παράδειγμα 2 Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω λογικό κύκλωμα.



ΠΡΟΓΡΑΜΜΑ

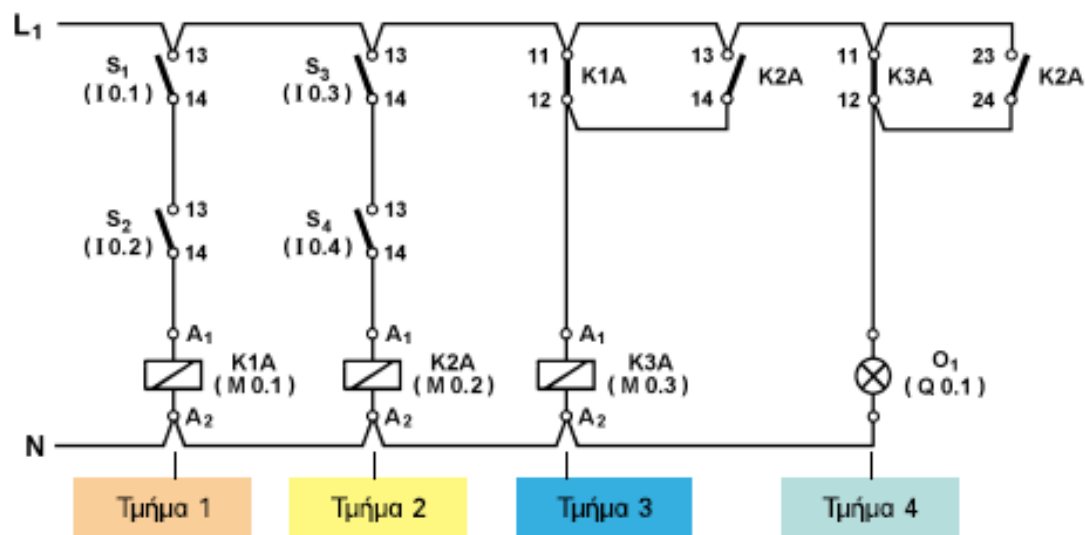
L	I	0,1
XO	I	0,2
=N	Q	0,2
LN	Q	0,2
ON	I	0,3
=	M	0,1
L	Q	0,2
AN	M	0,1
A	I	0,3
=	Q	0,1

Πύλη 1

Πύλη 2

Πύλη 3

Παράδειγμα 3. Να γραφεί το πρόγραμμα που αντιστοιχεί στο παρακάτω ηλεκτρολογικό σχέδιο αυτοματισμού.



ΠΡΟΓΡΑΜΜΑ

L	I	0.1
A	I	0.2
=	M	0.1

Τμήμα 1

L	I	0.3
A	I	0.4
=	M	0.2

Τμήμα 2

LN	M	0.1
O	M	0.2
=	M	0.3

Τμήμα 3

LN	M	0.3
O	M	0.2
=	Q	0.1

Τμήμα 4

Ανάπτυξη προγράμματος σε γλώσσα LADDER

Η γλώσσα Ladder (LAD) είναι γλώσσα που χρησιμοποιεί τα ηλεκτρολογικά γραφικά. Το πρόγραμμα σε γλώσσα LADDER μοιάζει με το ηλεκτρολογικό σχέδιο του αυτοματισμού. Οι ιδιαιτερότητες που έχουμε να αντιμετωπίσουμε στη γλώσσα αυτή είναι:

- Χρησιμοποιούνται σύμβολα από την Αμερικάνικη τυποποίηση και όχι από την Ευρωπαϊκή με την οποία είμαστε εξοικειωμένοι.
- Το “σχέδιο-πρόγραμμα” είναι τυποποιημένο, δεν έχουμε δηλαδή την ελευθερία που έχουμε κατά τη σχεδίαση. Για παράδειγμα σε κάθε κλάδο μπορούμε να έχουμε περιορισμένο αριθμό στοιχείων προγράμματος (διακόπτες και επαφές). Επίσης, δεν μπορούμε να κάνουμε οποιασδήποτε μορφής διακλάδωση.

Άρα η δουλειά που έχει να κάνει ο προγραμματιστής στη γλώσσα LADDER είναι να προσαρμόσει το σχέδιο του αυτοματισμού, στα δεδομένα που απαιτεί η γλώσσα.

Δομή προγράμματος στη γλώσσα LADDER

Το πρόγραμμα σε γλώσσα LADDER είναι ένα “σχέδιο”, “ένα διάγραμμα επαφών”, δηλαδή σχεδόν το ηλεκτρολογικό σχέδιο του αυτοματισμού.

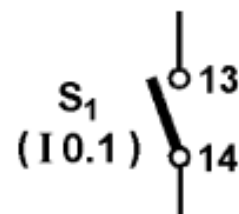
Το διάγραμμα επαφών της γλώσσας LADDER σχεδιάζεται όχι “κατακόρυφα” αλλά “οριζόντια”. Δηλαδή σε ένα πρόγραμμα LADDER έχουμε δύο παράλληλες κατακόρυφες γραμμές (μπάρες), η αριστερή γραμμή παριστάνει τη μπάρα τροφοδοσίας με το υψηλό δυναμικό (+) και η δεξιά γραμμή τη μπάρα τροφοδοσίας με το χαμηλό δυναμικό (-). Μεταξύ των δύο γραμμών σχεδιάζουμε οριζόντια τους κλάδους του “κυκλώματος”.

Κάθε κλάδος του διαγράμματος Ladder, που ξεκινά από την αριστερή μπάρα και καταλήγει στη δεξιά μπάρα, αποτελεί μια “γραμμή προγράμματος”, η οποία αντιστοιχεί στην ομάδα εντολών της γλώσσας λίστα εντολών.

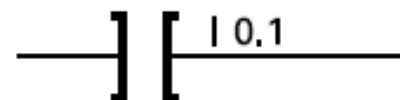
Τα σύμβολα που χρησιμοποιούνται είναι σύμβολα από την Αμερικανική τυποποίηση του ηλεκτρολογικού σχεδίου (ANSI). Τα βασικά σύμβολα δίνονται σε σχήμα 45. Δίπλα σε κάθε σύμβολο γράφεται το στοιχείο (η παράμετρος) στο οποίο αναφέρεται το σύμβολο.

“Κανονικά ανοικτή” επαφή → λογικό “1”

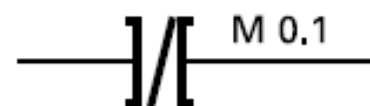
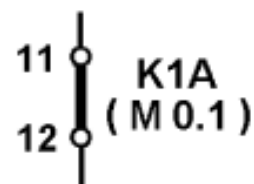
Σύμβολο
ηλεκτρολογικού
σχεδίου



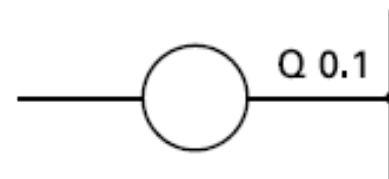
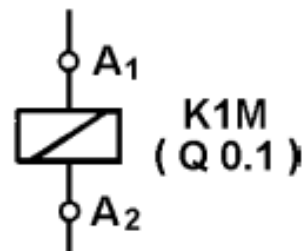
Σύμβολο στη
γλώσσα LADDER

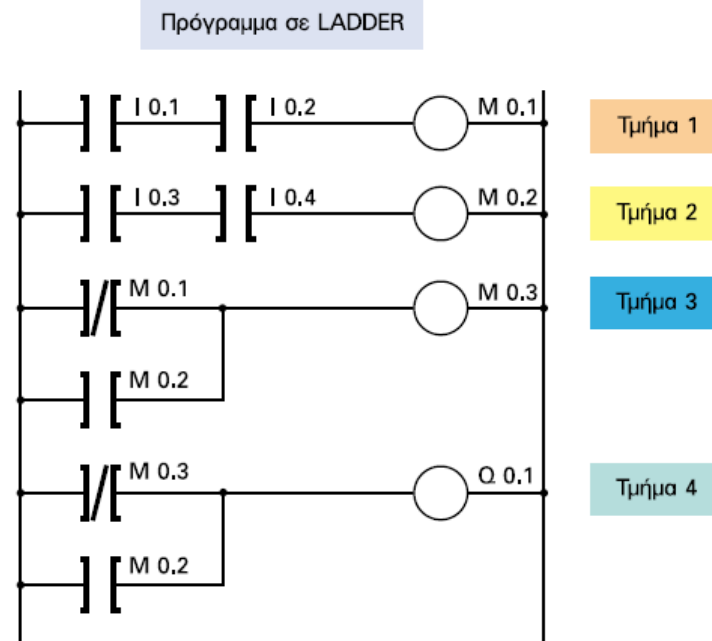
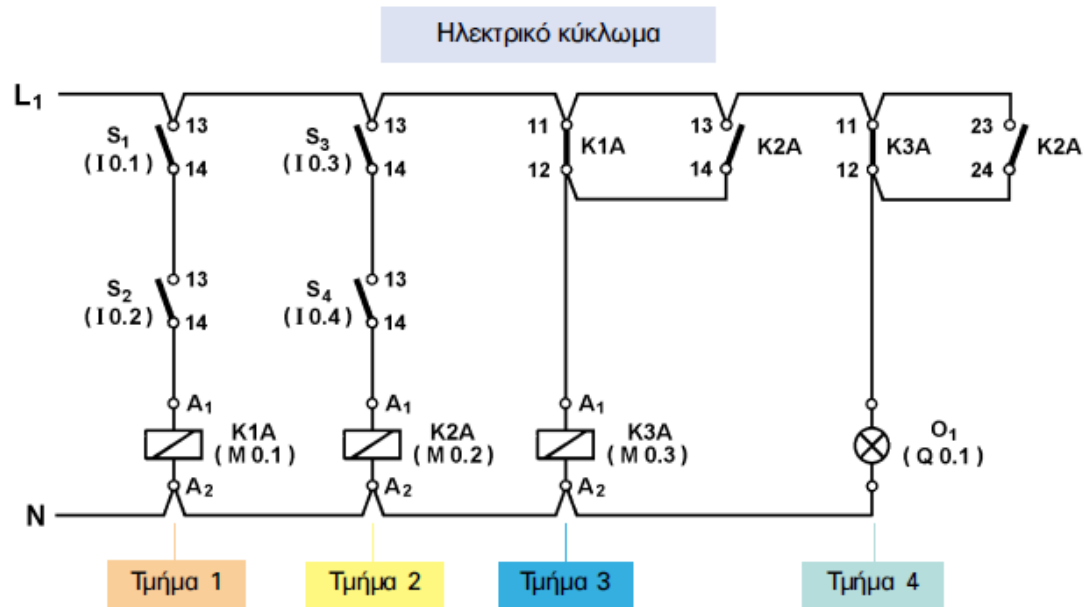


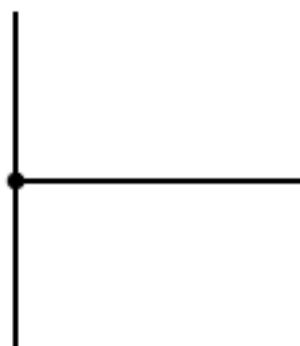
“Κανονικά κλειστή” επαφή ηλεκτρονόμου ισχύος
ή βοηθητικού ηλεκτρονόμου → λογικό “0”



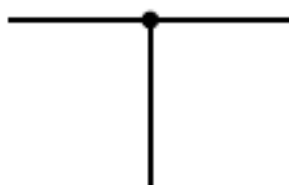
Πηνίο ηλεκτρονόμου ισχύος ή βοηθητικού
ηλεκτρονόμου → έξοδος προγράμματος



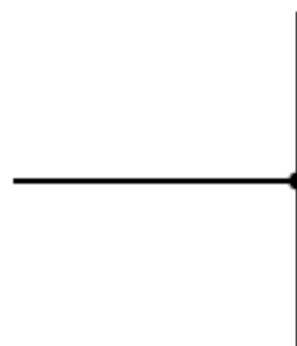




Κόμβος έναρξης

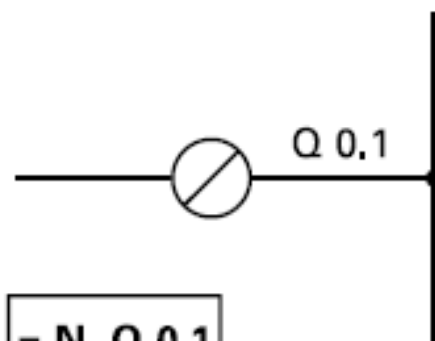


Κόμβος διακλάδωσης



Κόμβος τερματισμού

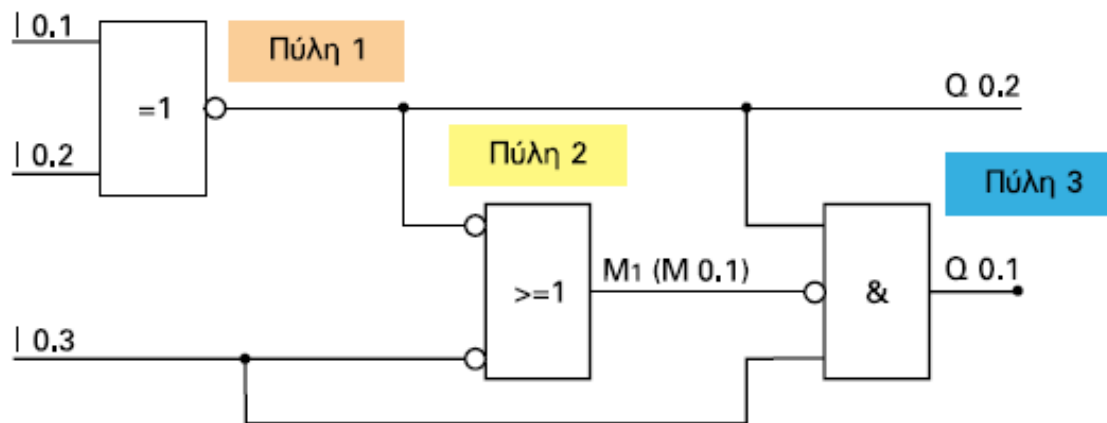
Άρνηση της εξόδου



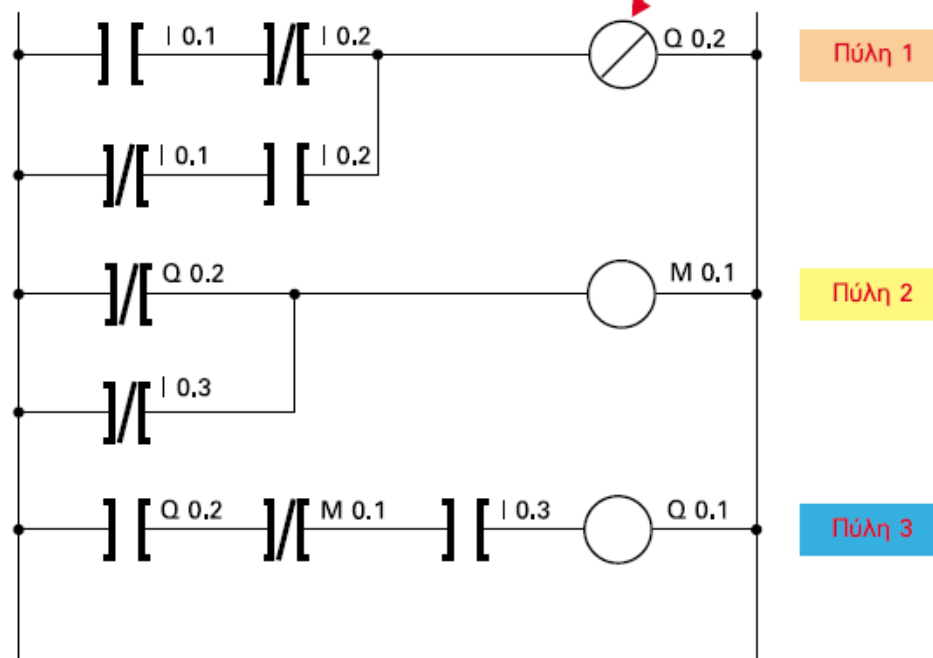
Αντιστοιχεί με την εντολή:

= N Q 0.1

Λογικό κύκλωμα



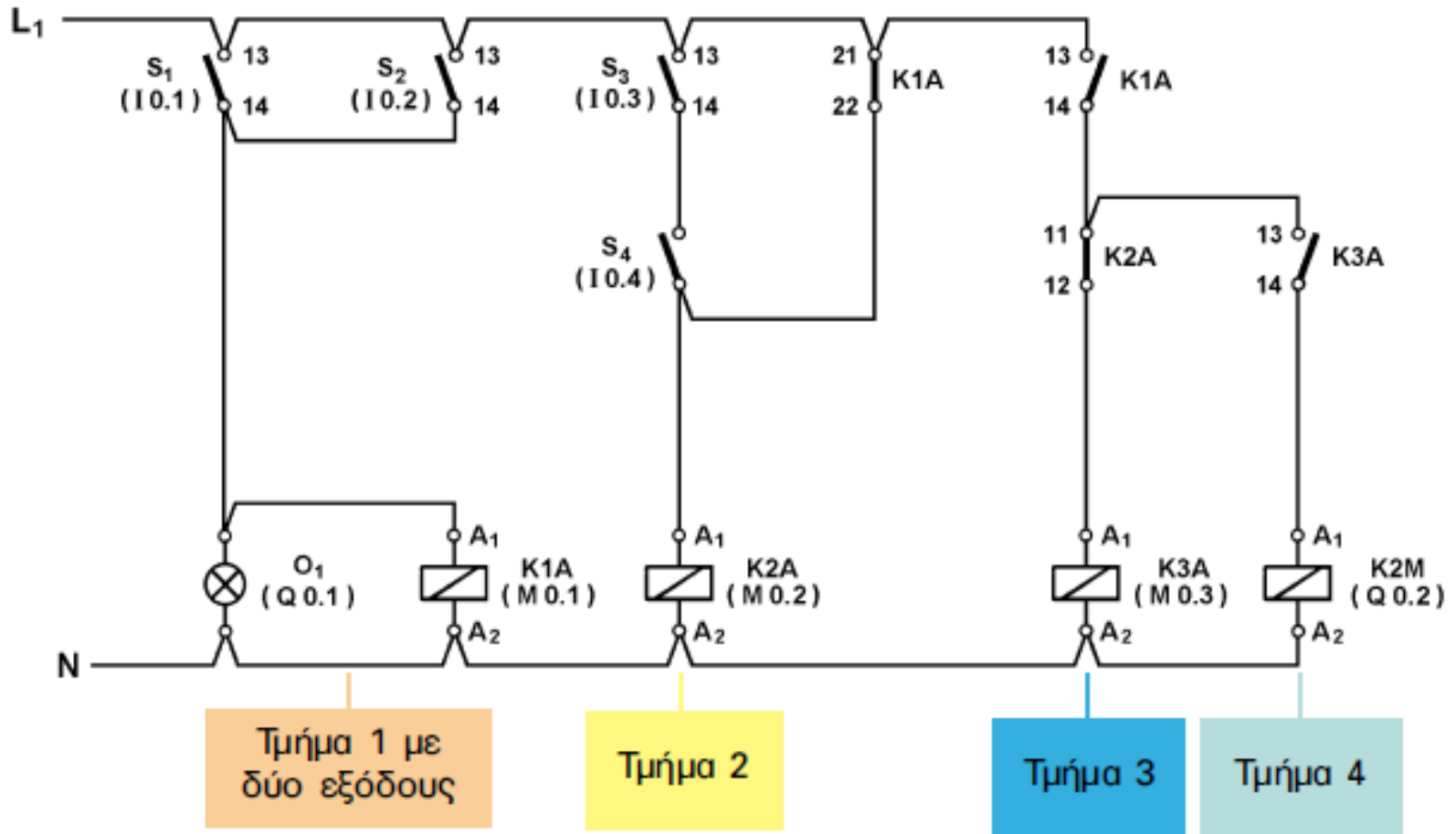
Πρόγραμμα σε LADDER



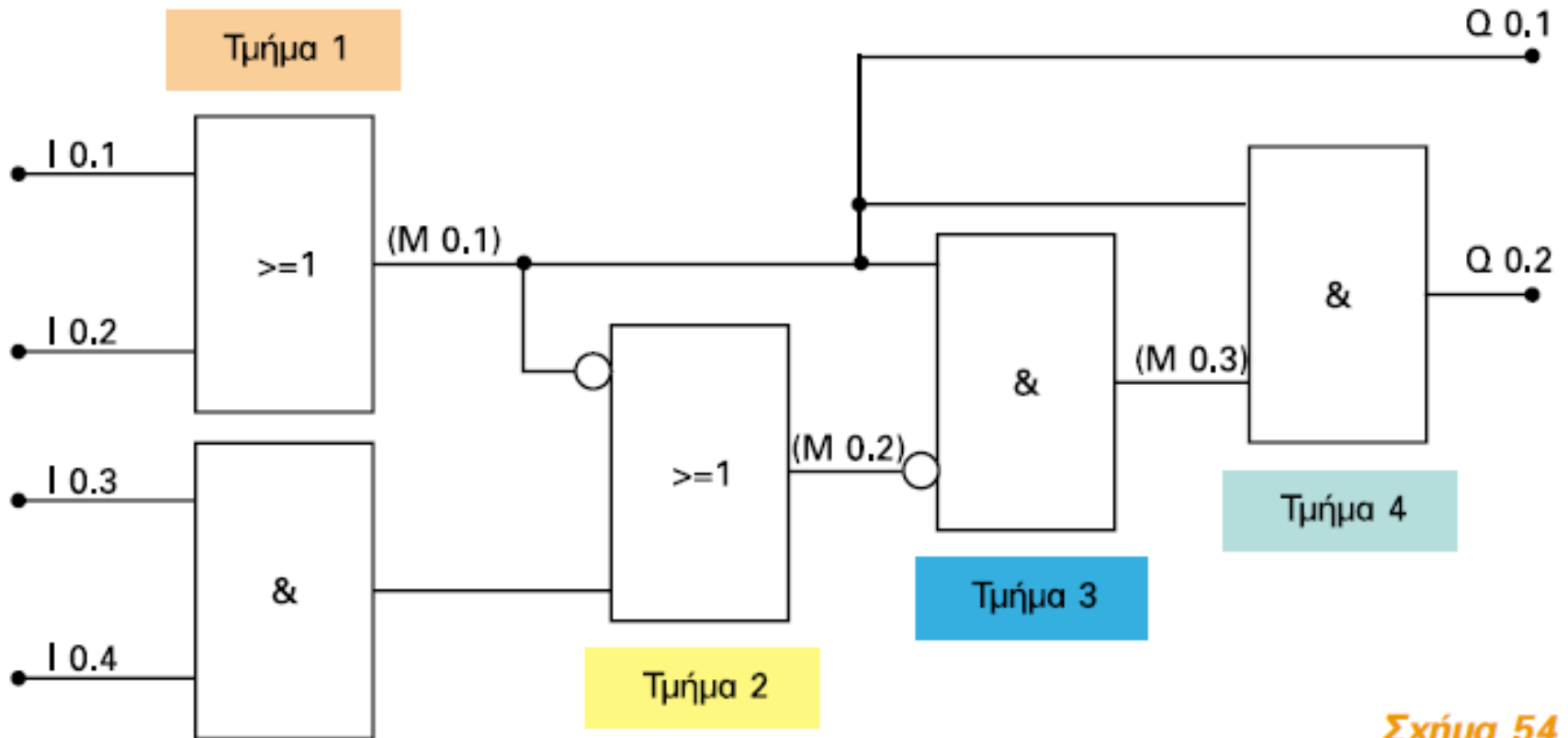
Προσέξτε τη χρήση της άρνησης εξόδου

Ανάπτυξη προγράμματος σε γλώσσα λογικών γραφικών

Ηλεκτρικό κύκλωμα

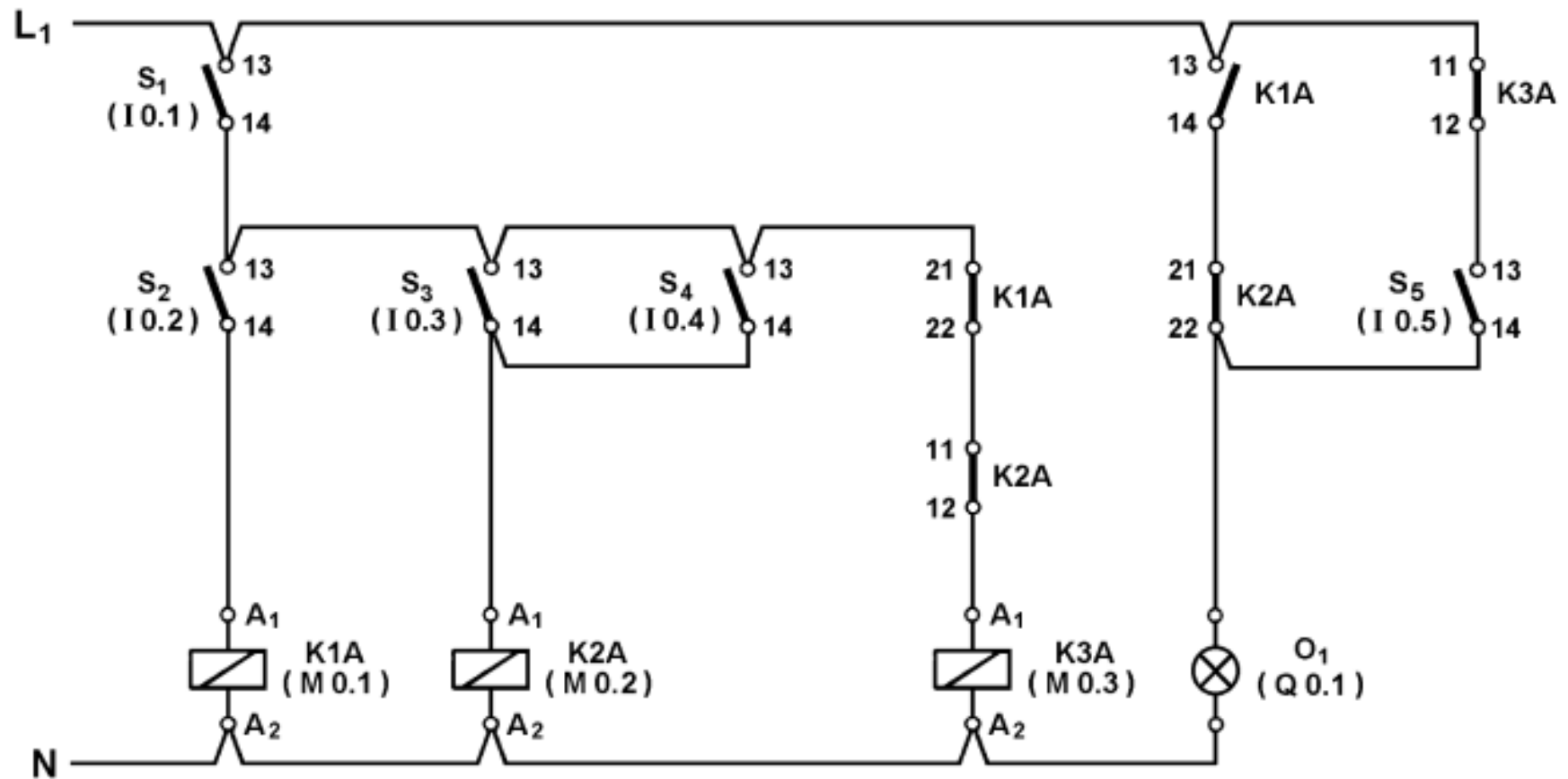


Πρόγραμμα σε γλώσσα λογικών γραφικών



Σχήμα 54

Ηλεκτρικό κύκλωμα



Πρόγραμμα σε γλώσσα λογικών γραφικών

