

Αυτοματισμοί και Συστήματα Αυτομάτου Ελέγχου

Ενότητα 5

Ανάπτυξη Προγράμματος σε Γλώσσα
Λίστας Εντολών

ΠΕΡΙΕΧΟΜΕΝΑ

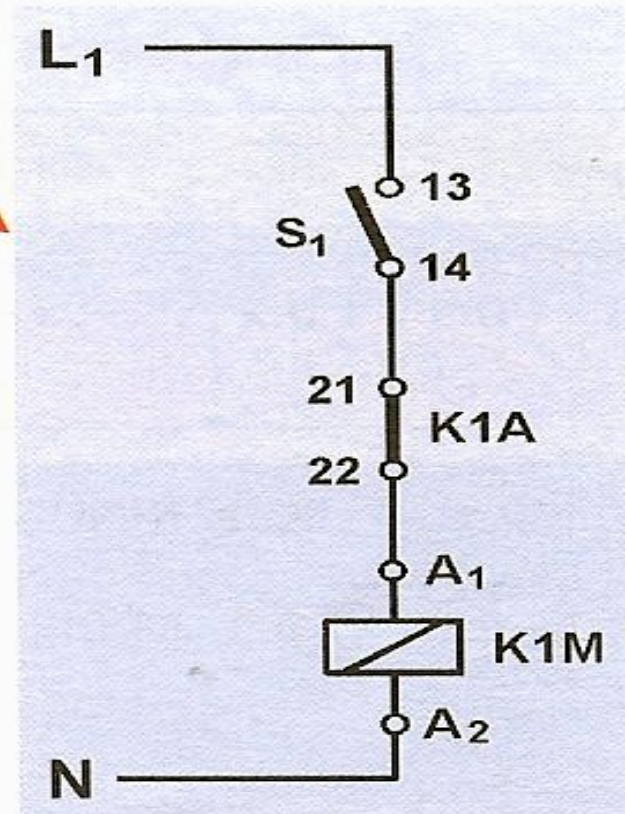
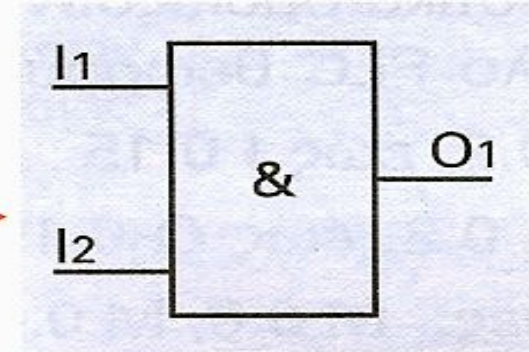
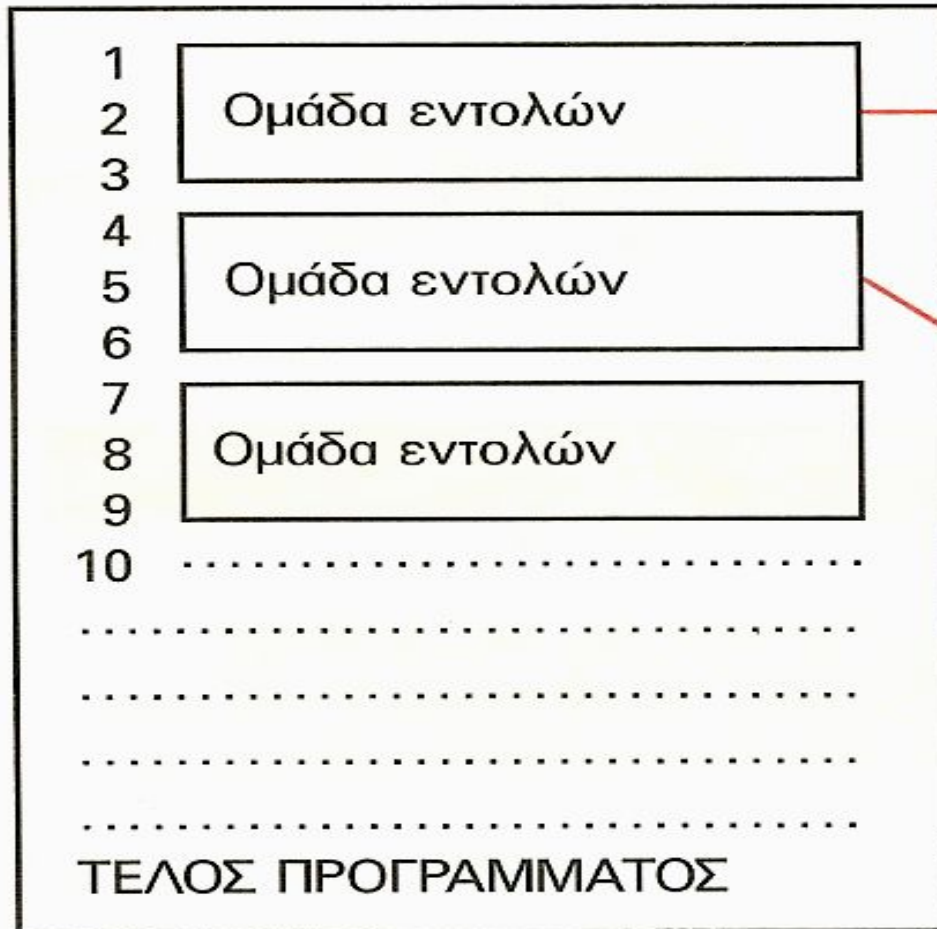
- 5.1 Βασικές εντολές προγραμματισμού στη γλώσσα λίστας εντολών.
- 5.2 Αναπτύσσοντας τα πρώτα προγράμματα.
- 5.3 Παρουσίαση λοιπών εντολών.
- 5.4 Παραδείγματα ανάπτυξης προγράμματος.
- 5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού».

5.1 Βασικές εντολές προγραμματισμού

- **Μορφή του προγράμματος.**
- Το πρόγραμμα αποτελείται από μία σειρά εντολών.
- Κάθε εντολή αποτελεί μια γραμμή προγράμματος.
- Οι εντολές κατανέμονται σε ομάδες εντολών.
- Κάθε ομάδα εντολών αντιστοιχεί σε μία λογική πύλη, ή αλλιώς σε ένα κλάδο του ηλεκτρολογικού κυκλώματος του αυτοματισμού.

5.1 Βασικές εντολές προγραμματισμού

Λίστα εντολών

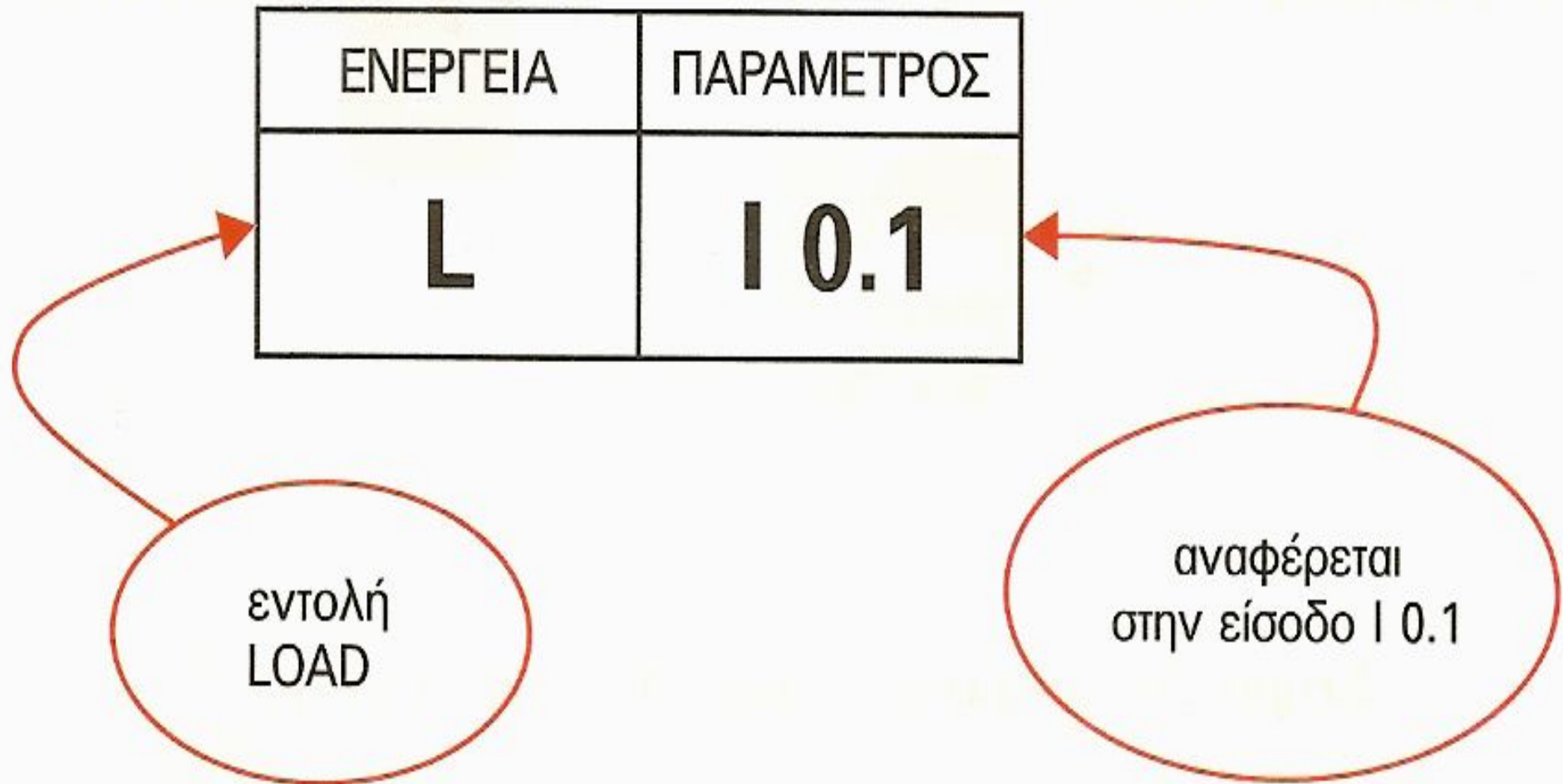


5.1 Βασικές εντολές προγραμματισμού

■ Μορφή εντολής.

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

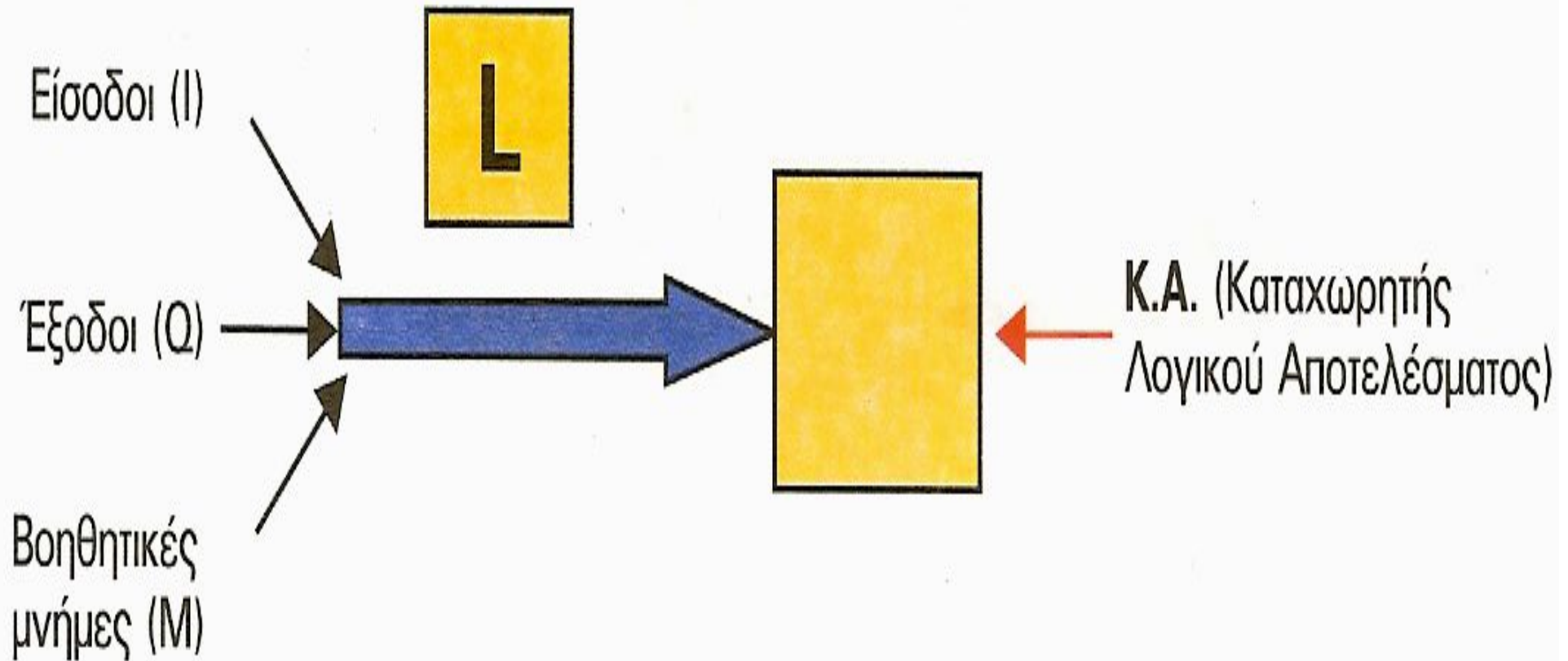
5.1 Βασικές εντολές προγραμματισμού



5.1 Βασικές εντολές προγραμματισμού

- **Παρουσίαση εντολών.**
- **Η εντολή L (Load).**
- Μια ομάδα εντολών ξεκινά με την εντολή L.
- Το PLC με την εντολή αυτή διαβάζει τη λογική κατάσταση («0», «1») μιας εισόδου, εξόδου, βοηθητικής μνήμης, χρονικού, κ.λ.π και την «φορτώνει» σε έναν «καταχωρητή» (μια ειδική θέση μνήμης) τον οποίο θα ονομάζουμε Καταχωρητή Λογικού Αποτελέσματος Κ.Α)

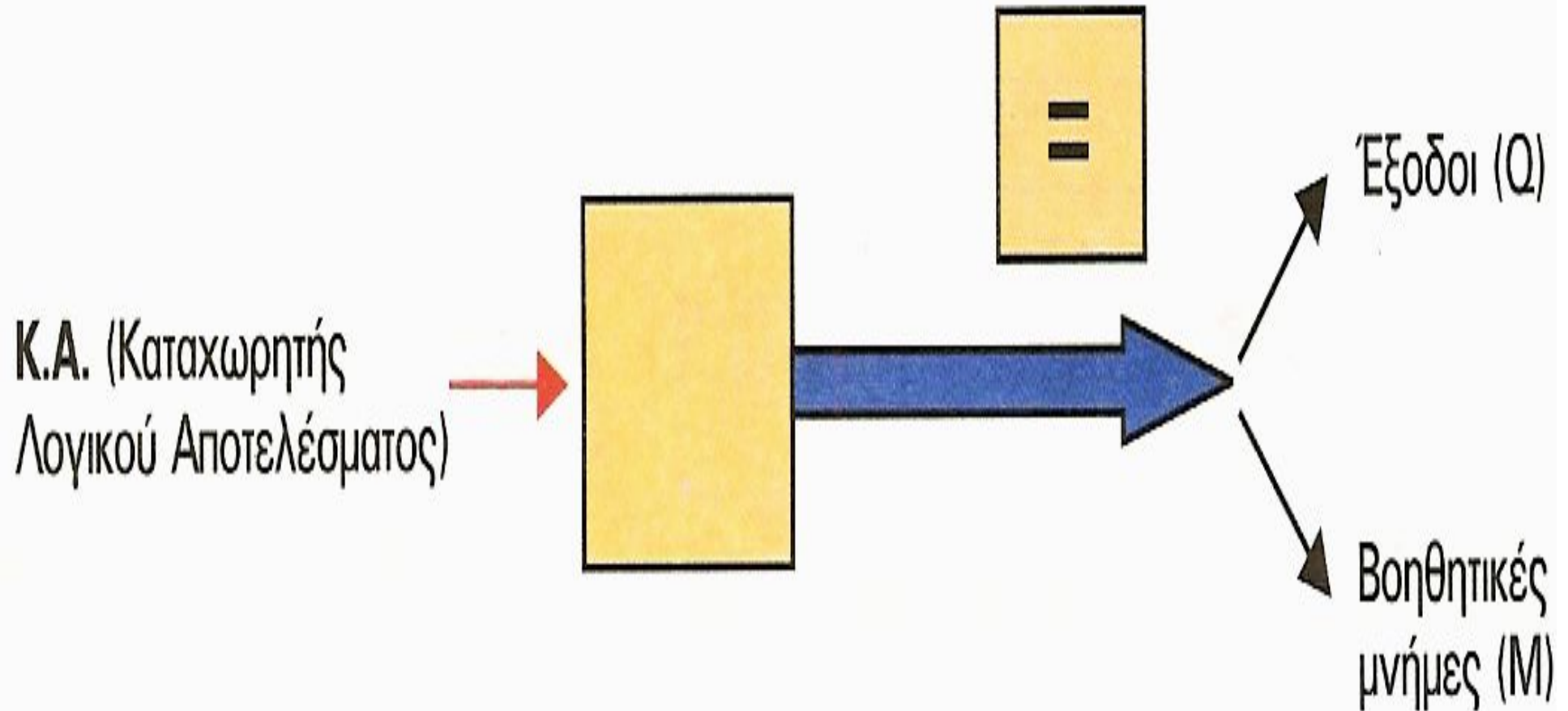
5.1 Βασικές εντολές προγραμματισμού



5.1 Βασικές εντολές προγραμματισμού

- Παρουσίαση εντολών.
- Η εντολή = (ίσον).
- Μια ομάδα εντολών καταλήγει πάντα με την εντολή = (ίσον).
- Η εντολή αναφέρεται σε εξόδους και βοηθητικές μνήμες.
- Το PLC με την εντολή αυτή μεταφέρει στις εξόδους ή στις βοηθητικές μνήμες το περιεχόμενο του Κ.Α.

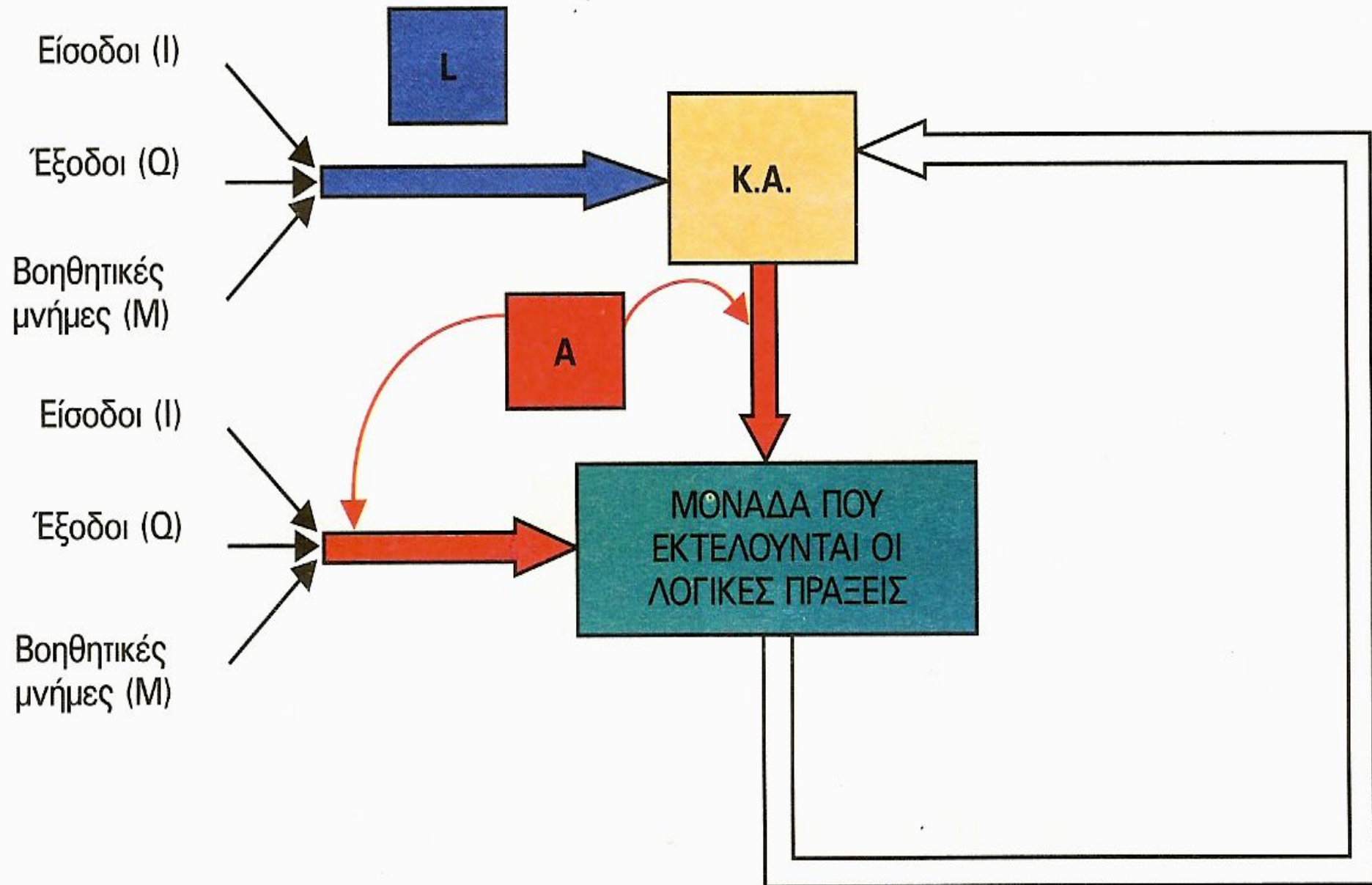
5.1 Βασικές εντολές προγραμματισμού



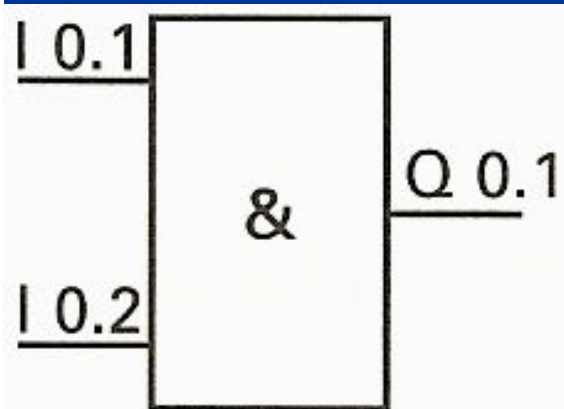
5.1 Βασικές εντολές προγραμματισμού

- Παρουσίαση εντολών.
- Η εντολή A (AND).
- Με την εντολή A το PLC εκτελεί μια λογική πράξη AND μεταξύ της λογικής κατάστασης της εισόδου, εξόδου, βοηθητικές μνήμες, χρονικά, κ.λ.π. και του περιεχόμενου του Κ.Α.
- Το αποτέλεσμα επιστρέφει στο Κ.Α.

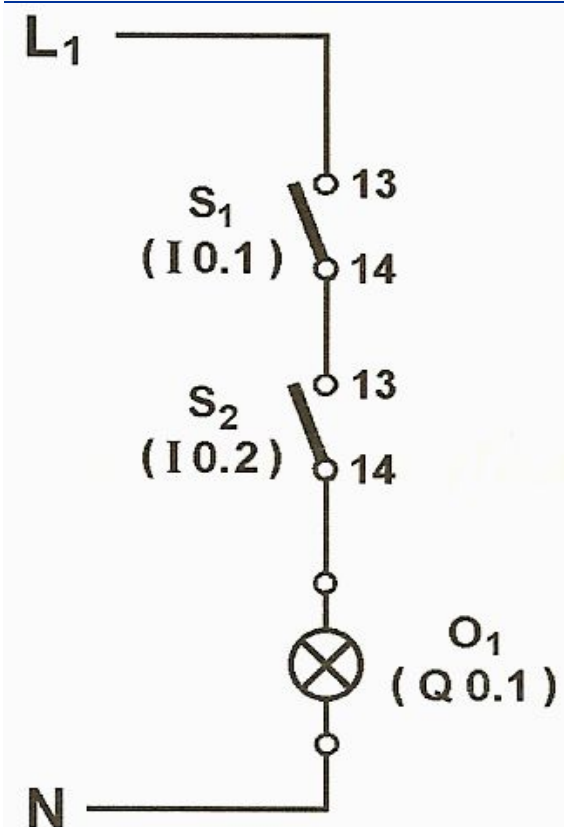
5.1 Βασικές εντολές προγραμματισμού



5.2 Αναπτύσσοντας τα πρώτα προγράμματα



Η εντολή Load φέρνει στον Κ.Α την λογική κατάσταση της εισόδου I 0.1



Η εντολή AND εκτελεί τη λογική πράξη μεταξύ της λογικής κατάστασης της εισόδου I 0.2 και του περιεχομένου του Κ.Α

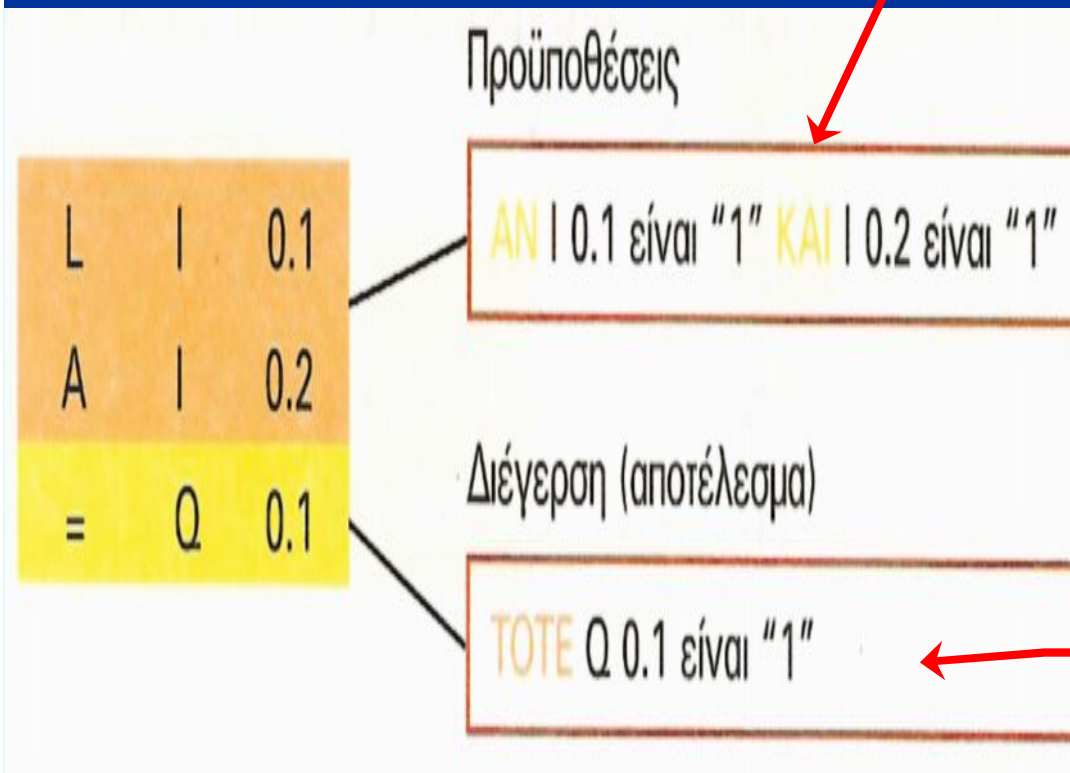
Η εντολή = οδηγεί το περιεχόμενο του Κ.Α (που είναι το λογικό αποτέλεσμα της πράξης AND) στην έξοδο Q 0.1

ΠΡΟΓΡΑΜΜΑ

L	I	0.1
A	I	0.2
=	Q	0.1



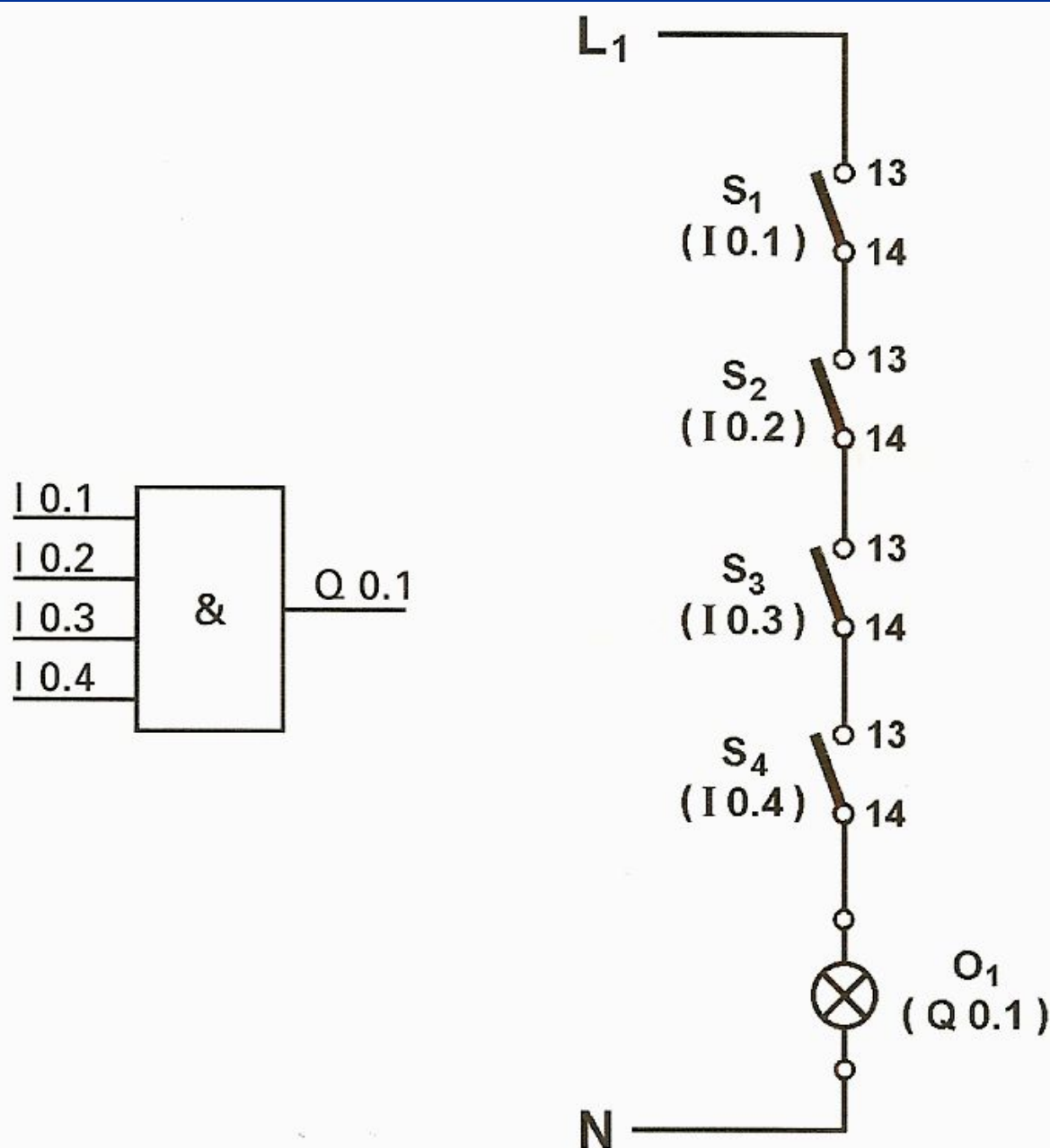
5.2 Αναπτύσσοντας τα πρώτα προγράμματα



Το πρώτο μέρος αποτελούν όλες οι εντολές πριν την εντολή =. Οι εντολές αυτές θέτουν τις προϋποθέσεις, τις ερωτήσεις.

Το δεύτερο μέρος αποτελεί η εντολή =. Η εντολή αυτή δίνει το αποτέλεσμα, τη διέγερση.

5.2 Αναπτύσσοντας τα πρώτα προγράμματα



ΠΡΟΓΡΑΜΜΑ

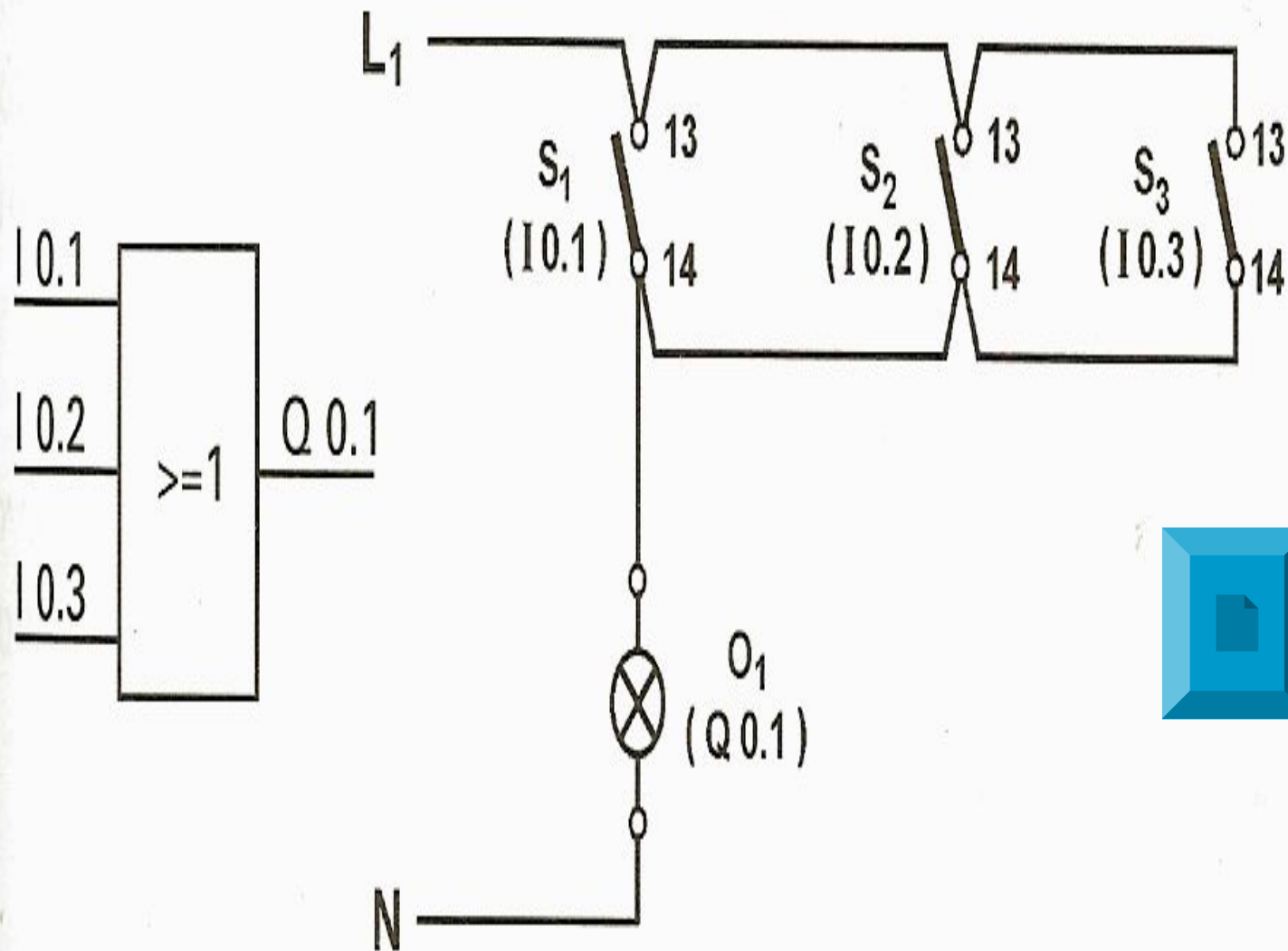
L	I	0.1
A	I	0.2
A	I	0.3
A	I	0.4
=	Q	0.1



5.3 Παρουσίαση λοιπών εντολών

- Η εντολή **O (OR)**.
- Με την εντολή αυτή το PLC θα **εκτελέσει τη λογική πράξη OR** μεταξύ εισόδων, εξόδων, βοηθητικές μνήμες, χρονικά, κ.λ.π και των περιεχομένων του Κ.Α.
- Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελείται η εντολή AND.

5.3 Παρουσίαση λοιπών εντολών



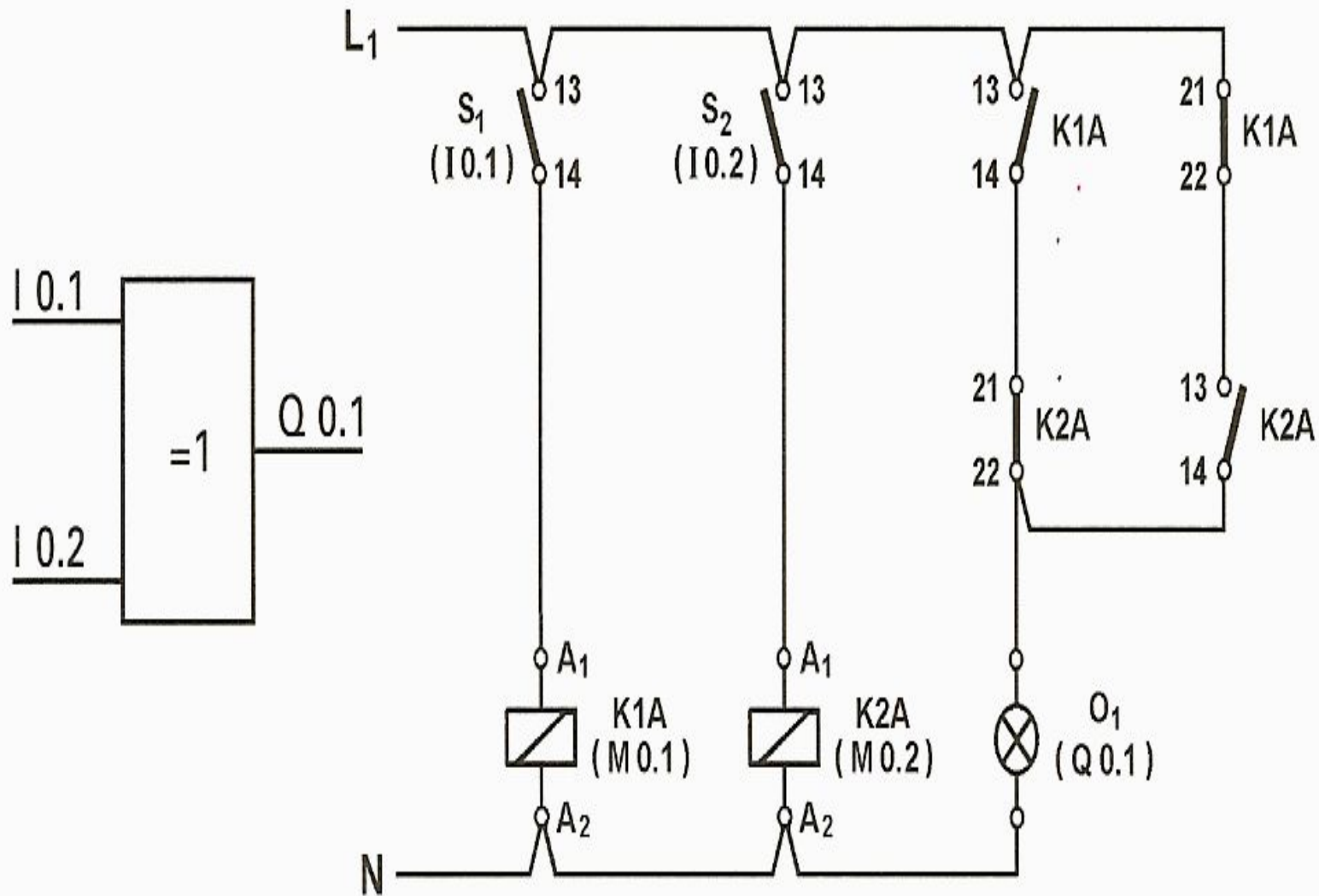
ΠΡΟΓΡΑΜΜΑ

L	I	0.1
0	I	0.2
0	I	0.3
=	Q	0.1

5.3 Παρουσίαση λοιπών εντολών

- Η εντολή **XO (XOR)**.
- Με την εντολή αυτή το PLC θα **εκτελέσει τη λογική πράξη XOR** μεταξύ εισόδων, εξόδων, βοηθητικές μνήμες, χρονικά, κ.λ.π και των περιεχομένων του Κ.Α.
- Εκτελείται με ανάλογο τρόπο, με αυτόν που εκτελείται η εντολή AND και OR.

5.3 Παρουσίαση λοιπών εντολών



ΠΡΟΓΡΑΜΜΑ

L	I	0.1
XO	I	0.2
=	Q	0.1

5.3 Παρουσίαση λοιπών εντολών

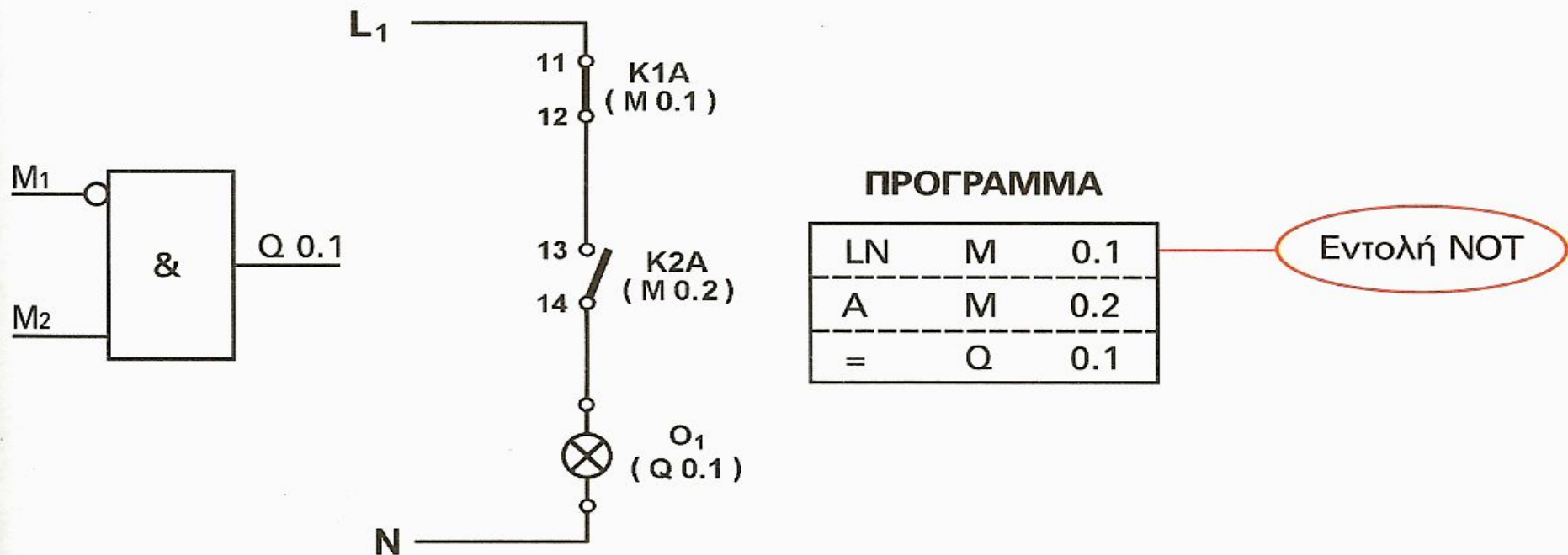
- **Αντιστοιχία συμβολισμών των ηλεκτρολογικών κυκλωμάτων αυτοματισμού με τους συμβολισμούς PLC.**
- $I 0.1 > S1$, διακόπτες, μπουτόνς, αισθητήρες.
- $Q 0.1 > O1$, λάμπες, κινητήρες, ηλεκτρονόμους ισχύος, (αποτέλεσμα), κ.λ.π.
- $K1A > M 0.1$, βοηθητικούς ηλεκτρονόμους.

5.3 Παρουσίαση λοιπών εντολών

- **Η εντολή N (NOT).**
- Δεν πρόκειται για ανεξάρτητη λογική εντολή όπως είναι οι τρεις εντολές λογικών πράξεων AND, OR, XOR.
- Η εντολή N είναι συμπλήρωμα όλων των εντολών που είδαμε προηγουμένως.
- **LN > LOAD NOT**
- **AN > AND NOT**
- **ON > OR NOT**
- **XON > XOR NOT**
- **=N > =NOT**

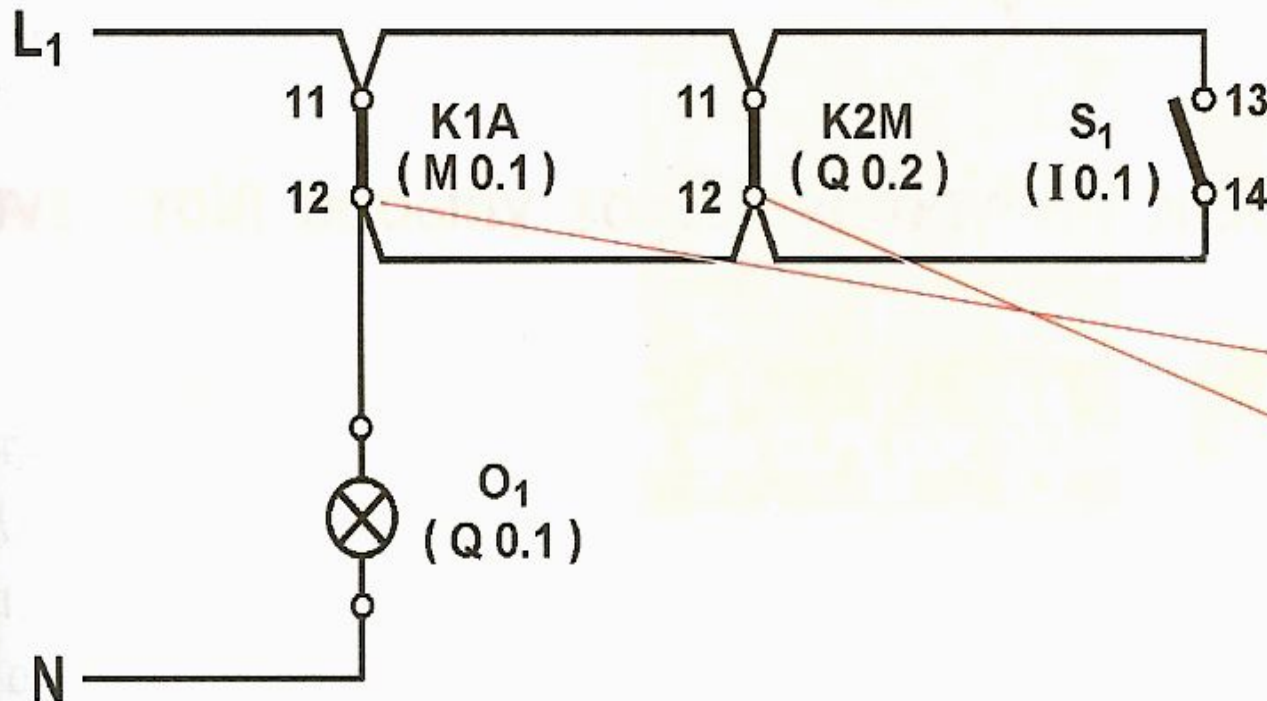
5.3 Παρουσίαση λοιπών εντολών

- Όταν έχουμε το σχέδιο του λογικού κυκλώματος του αυτοματισμού και στο λογικό κύκλωμα έχουμε πύλη NOT τότε στην αντίστοιχη εντολή προσθέτουμε το γράμμα N.



5.3 Παρουσίαση λοιπών εντολών

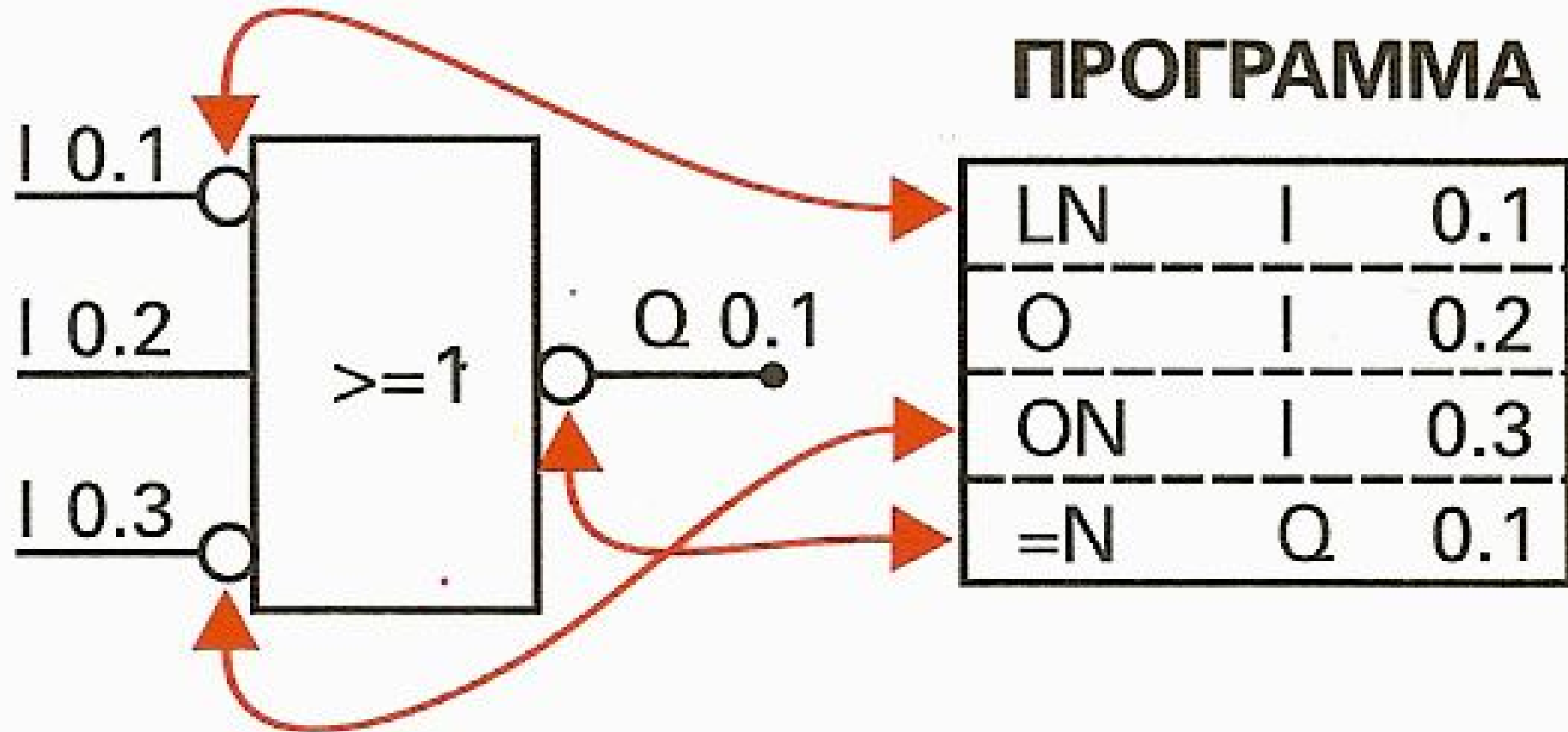
- Όταν έχουμε το ηλεκτρολογικό σχέδιο και σε αυτό έχουμε μια «κανονική κλειστή» επαφή ηλεκτρονόμου τότε στην εντολή προσθέτουμε το γράμμα N. **Δεν ισχύει** για διακόπτες, μπουτόν, αισθητήρια και τις εισόδους του PLC.



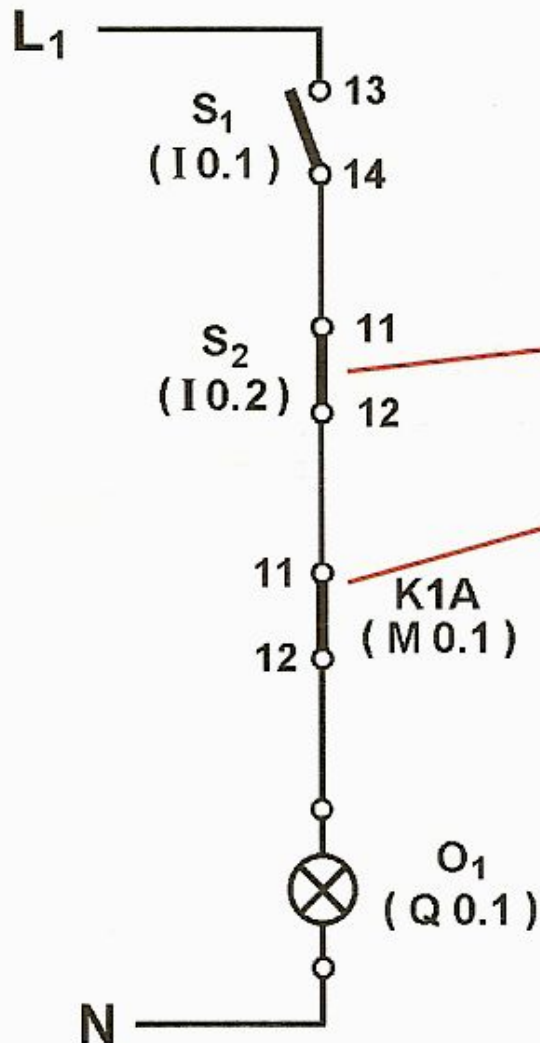
ΠΡΟΓΡΑΜΜΑ

LN	M	0.1
ON	Q	0.2
O	I	0.1
=	Q	0.1

5.3 Παρουσίαση λοιπών εντολών



5.3 Παρουσίαση λοιπών εντολών



ΠΡΟΓΡΑΜΜΑ

L	I	0.1
A	I	0.2
AN	M	0.1
=	Q	0.1

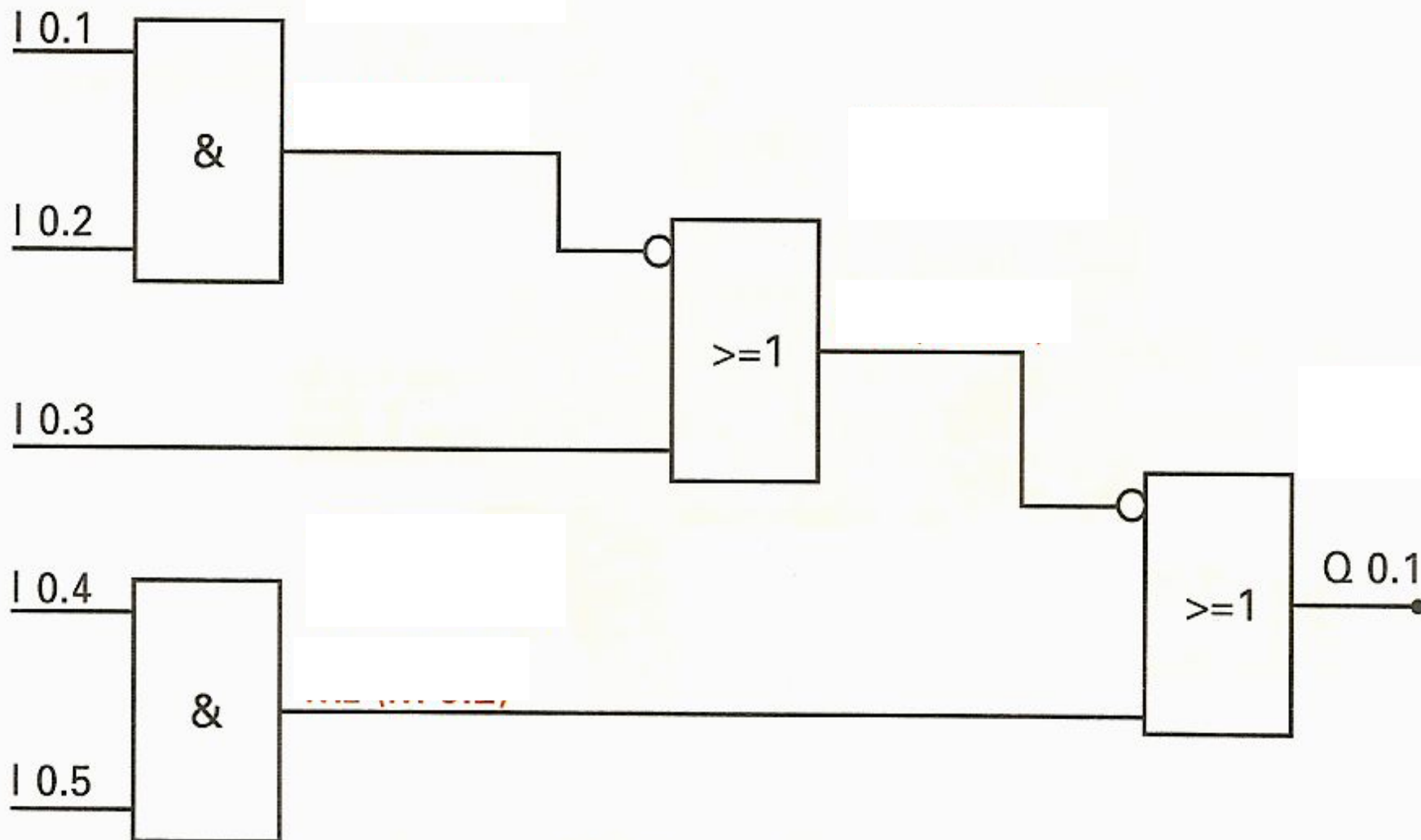
ΠΡΟΣΟΧΗ

Για τις εισόδους του PLC με "κανονικά κλειστές" επαφές (όπως η I 0.2 εδώ) δεν βάζουμε NOT στις αντίστοιχες εντολές του προγράμματος.

5.4 Παραδείγματα

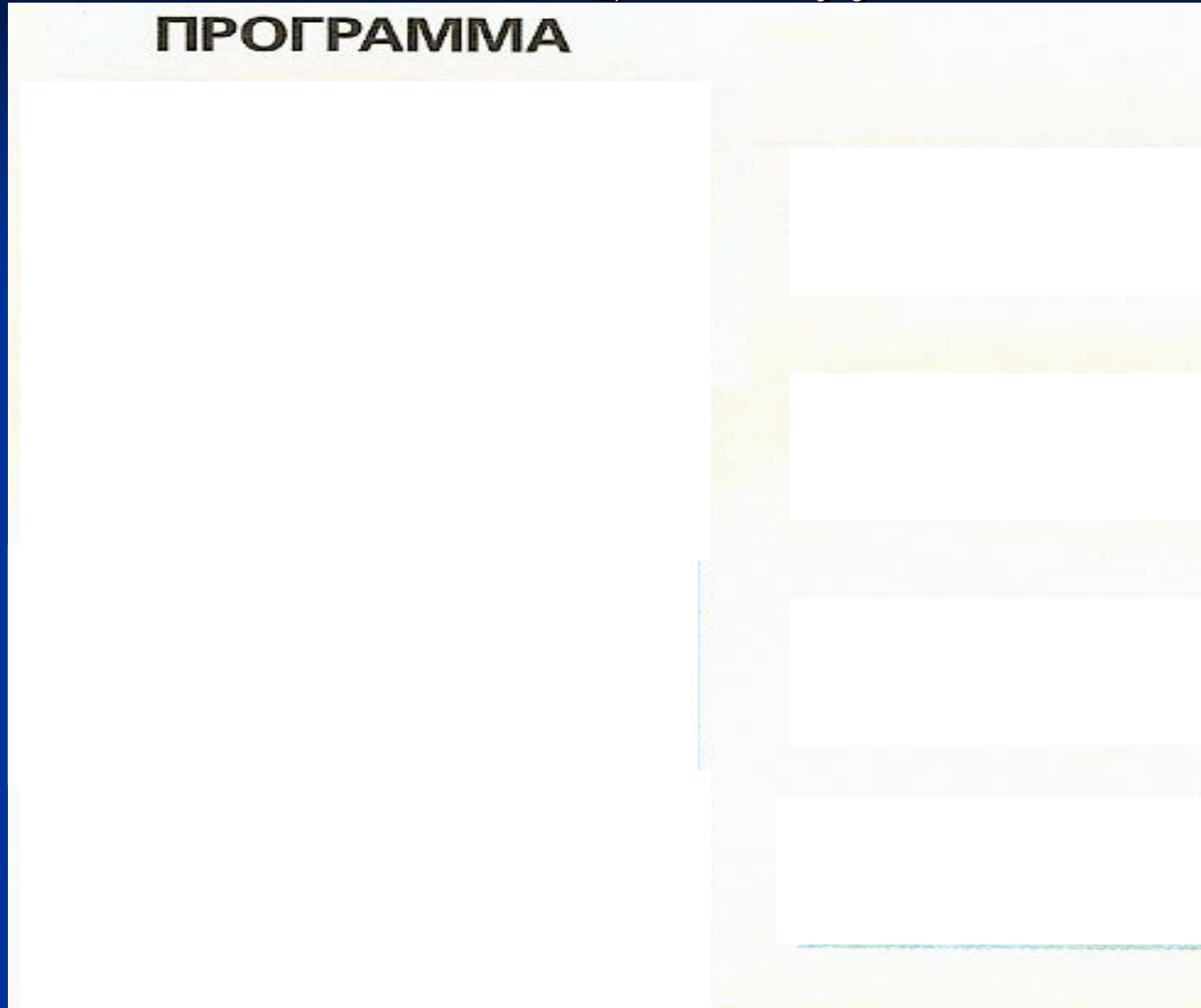
- **Βασικοί κανόνες για την ανάπτυξη του προγράμματος**
- **Όταν έχουμε το σχέδιο του λογικού κυκλώματος του αυτοματισμού.**
 1. Αντιστοιχούμε τις εξόδους των ενδιάμεσων πυλών (δηλαδή τις εξόδους πυλών που είναι είσοδοι σε άλλες πύλες) με βοηθητικές μνήμες (π.χ. M 0.0, M 0.1,).
 2. Χωρίζουμε το σχέδιο νοητά σε ζώνες προτεραιότητας και αριθμούμε τις πύλες.
 3. Για κάθε πύλη γράφουμε την ομάδα εντολών.

5.4 Παραδείγματα 1

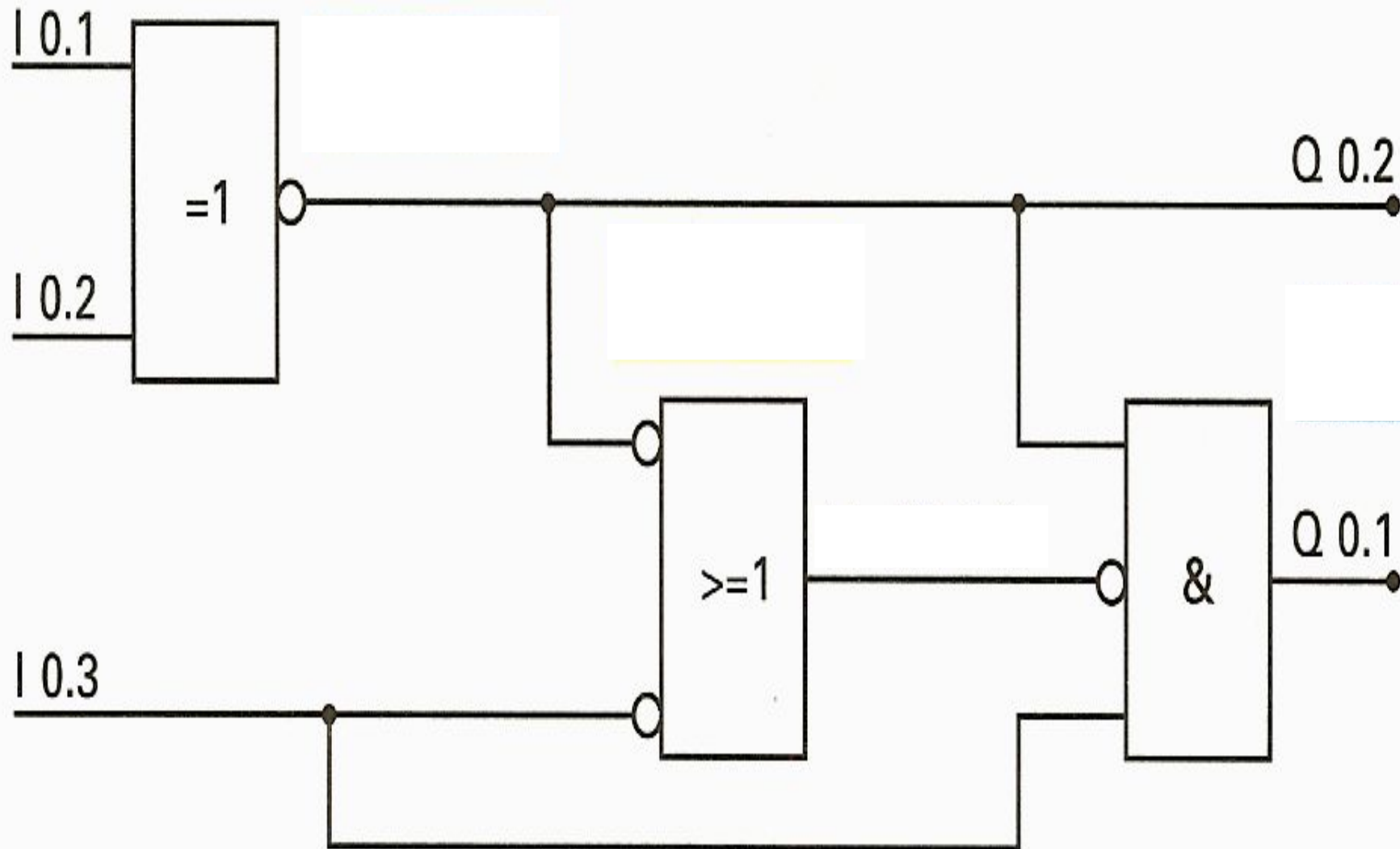


5.4 Παραδείγματα 1

ΠΡΟΓΡΑΜΜΑ

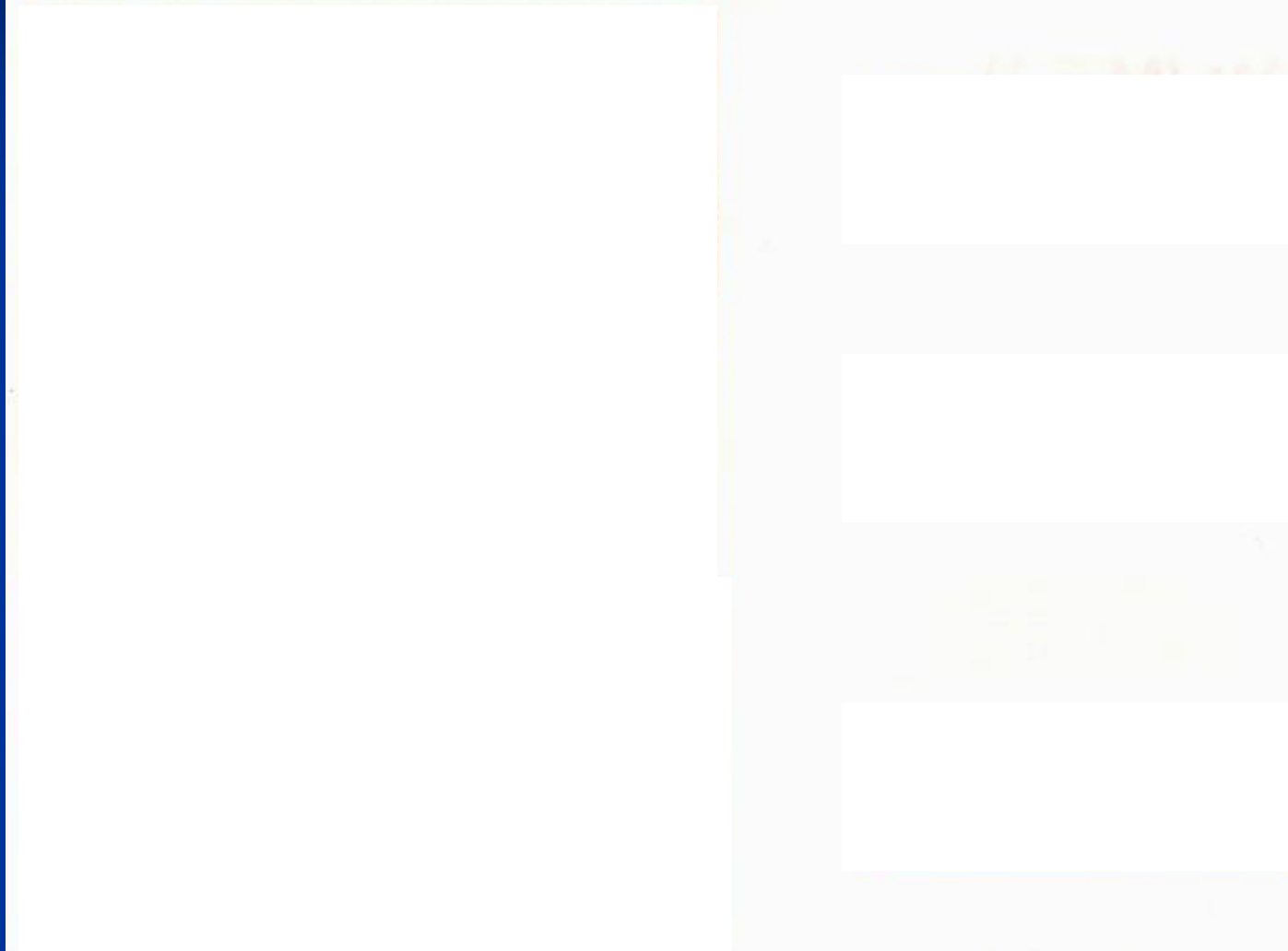


5.4 Παραδείγματα 2



5.4 Παραδείγματα 2

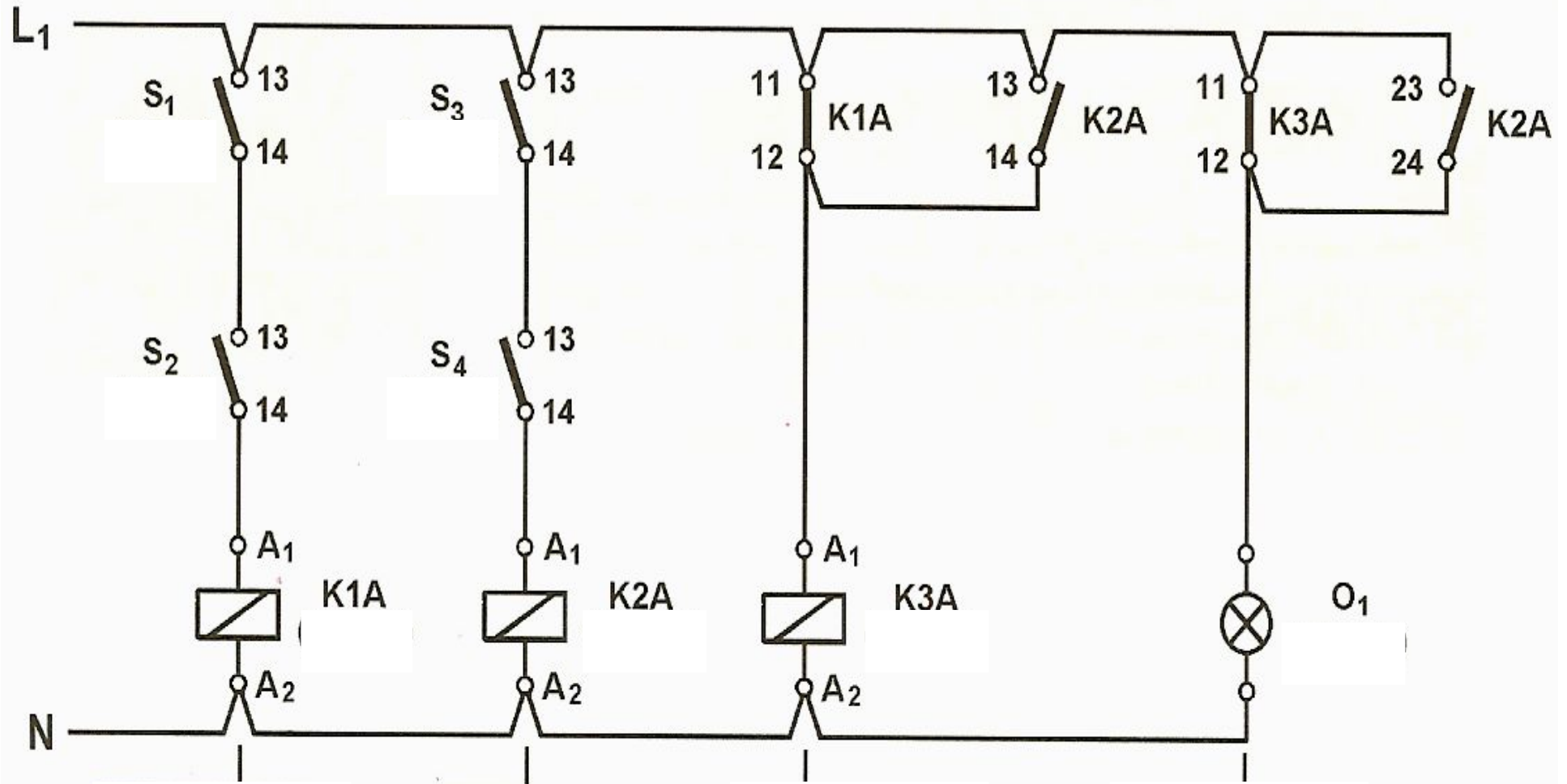
ΠΡΟΓΡΑΜΜΑ



5.4 Παραδείγματα

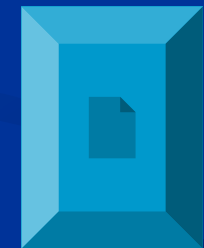
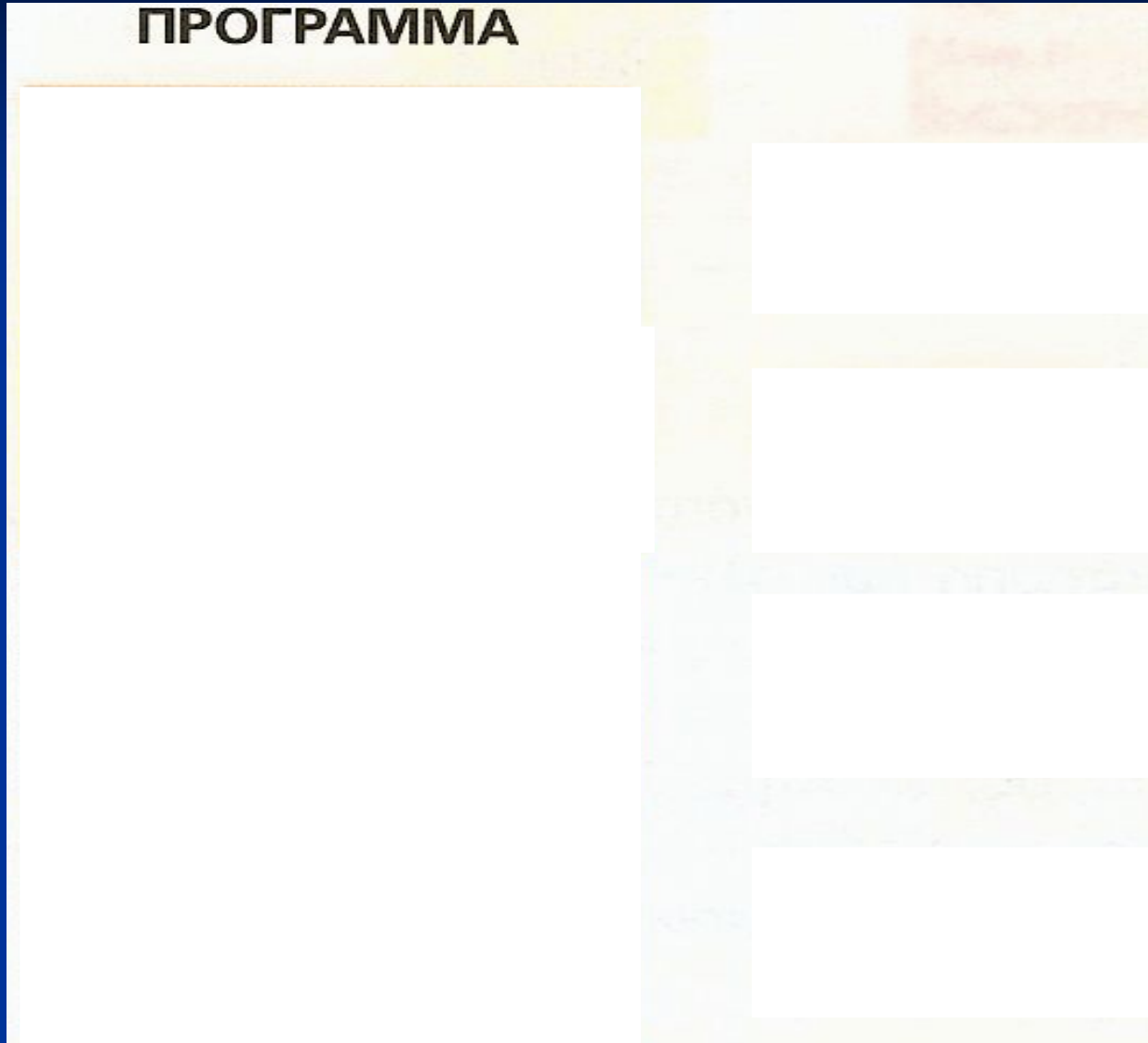
- **Βασικοί κανόνες για την ανάπτυξη του προγράμματος**
- **Όταν έχουμε το ηλεκτρολογικό σχέδιο του αυτοματισμού.**
 1. Αντιστοιχούμε τα πηνία των βοηθητικών ηλεκτρονόμων με βοηθητικές μνήμες (π.χ. M 0.0, M 0.1,
 2. Για κάθε «αυτοτελές» τμήμα του κυκλώματος γράφουμε την ομάδα εντολών. Υπενθυμίζουμε ότι ένα «αυτοτελές» τμήμα κυκλώματος χαρακτηρίζεται από το πηνίο του ηλεκτρονόμου ή από την κατανάλωση στην οποία καταλήγει..

5.4 Παραδείγματα 3

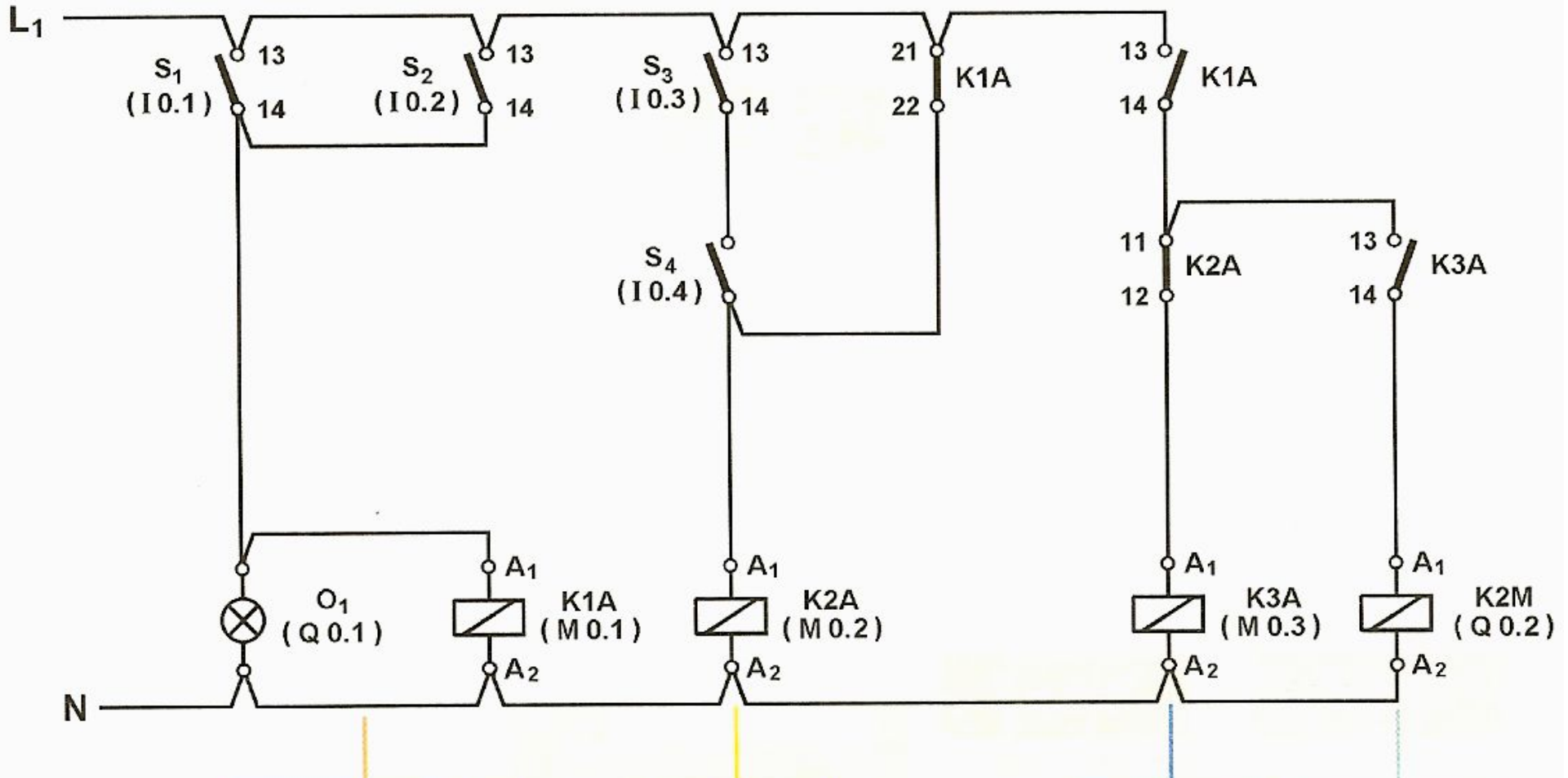


5.4 Παραδείγματα 3

ΠΡΟΓΡΑΜΜΑ



5.4 Παραδείγματα 4



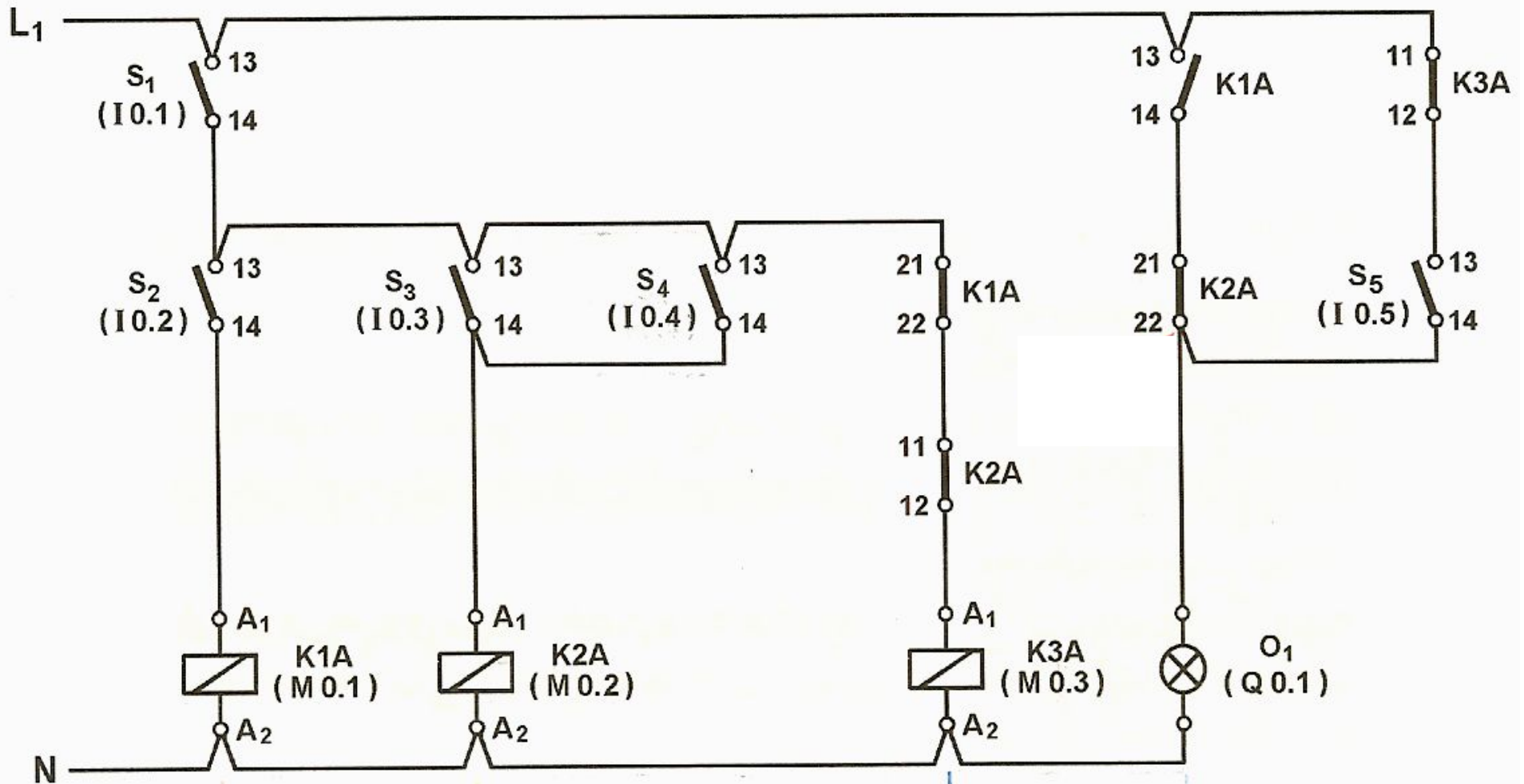
5.4 Παραδείγματα 4

ΠΡΟΓΡΑΜΜΑ



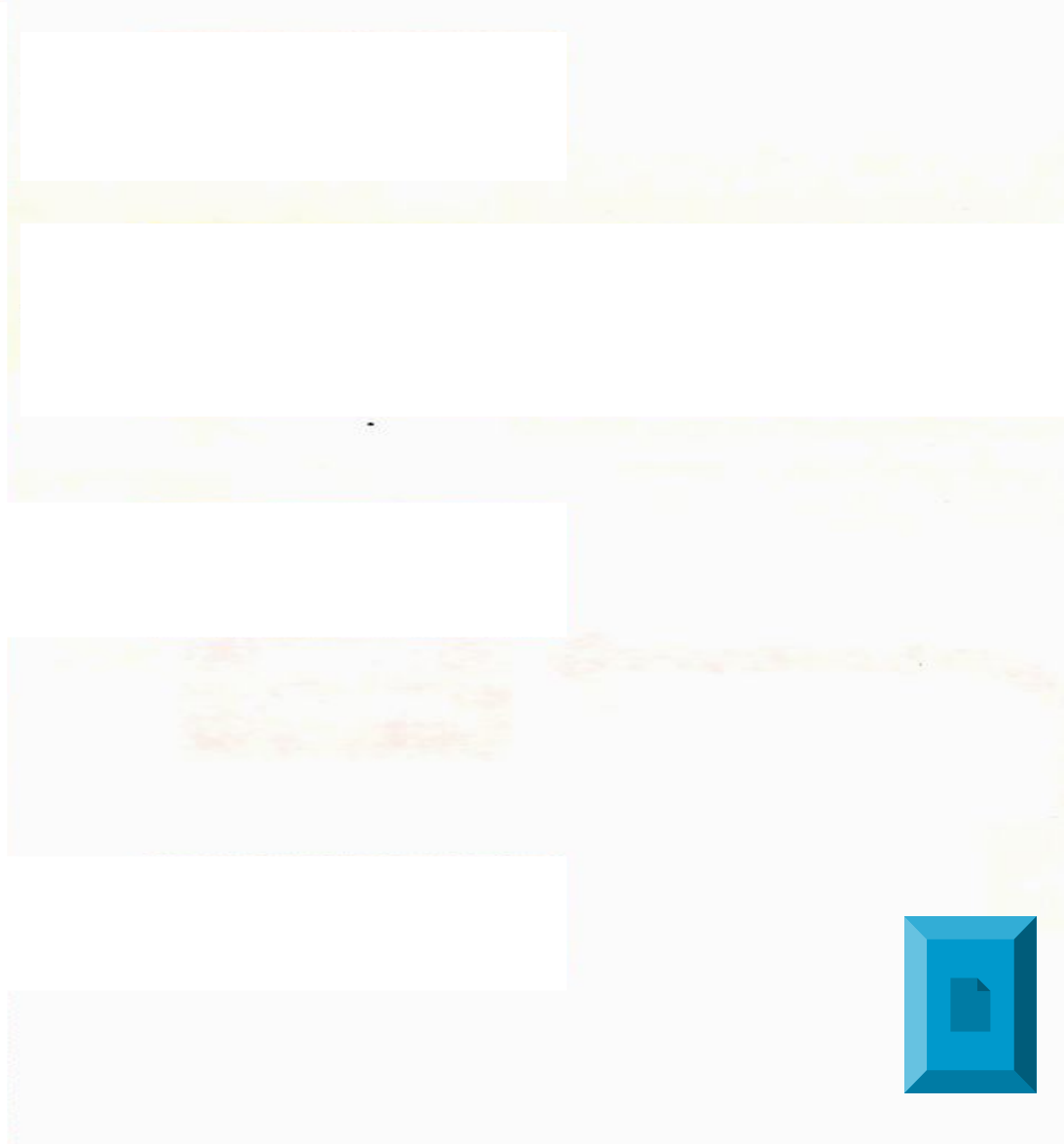
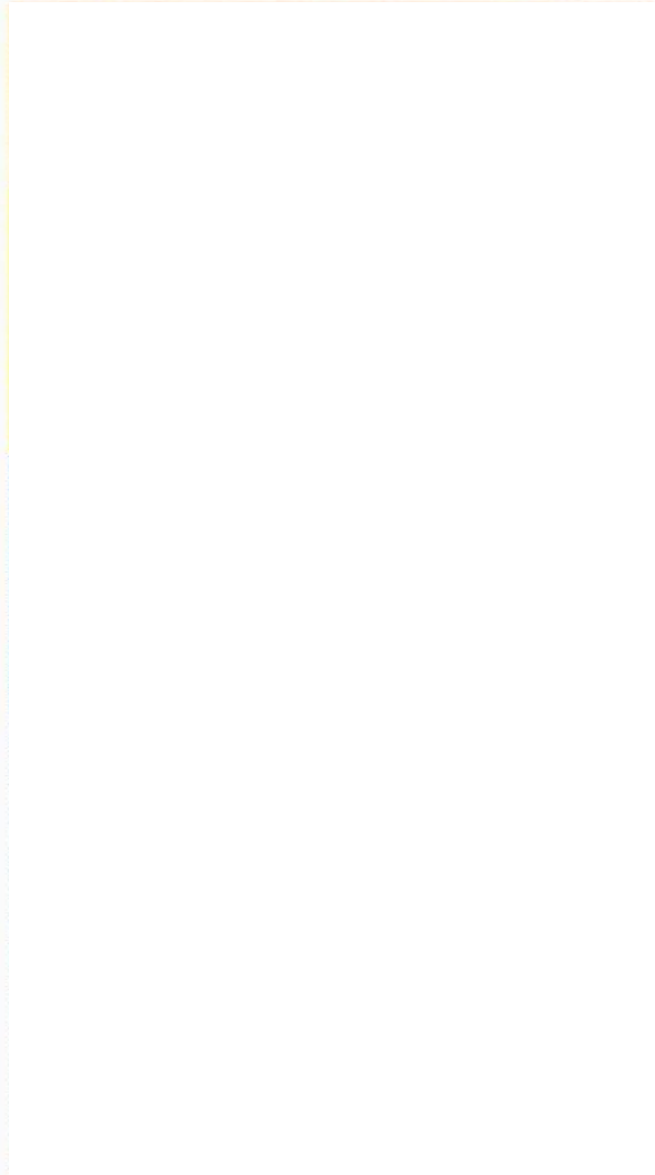
Σχήμα 35

5.4 Παραδείγματα 5



5.4 Παραδείγματα 5

ΠΡΟΓΡΑΜΜΑ



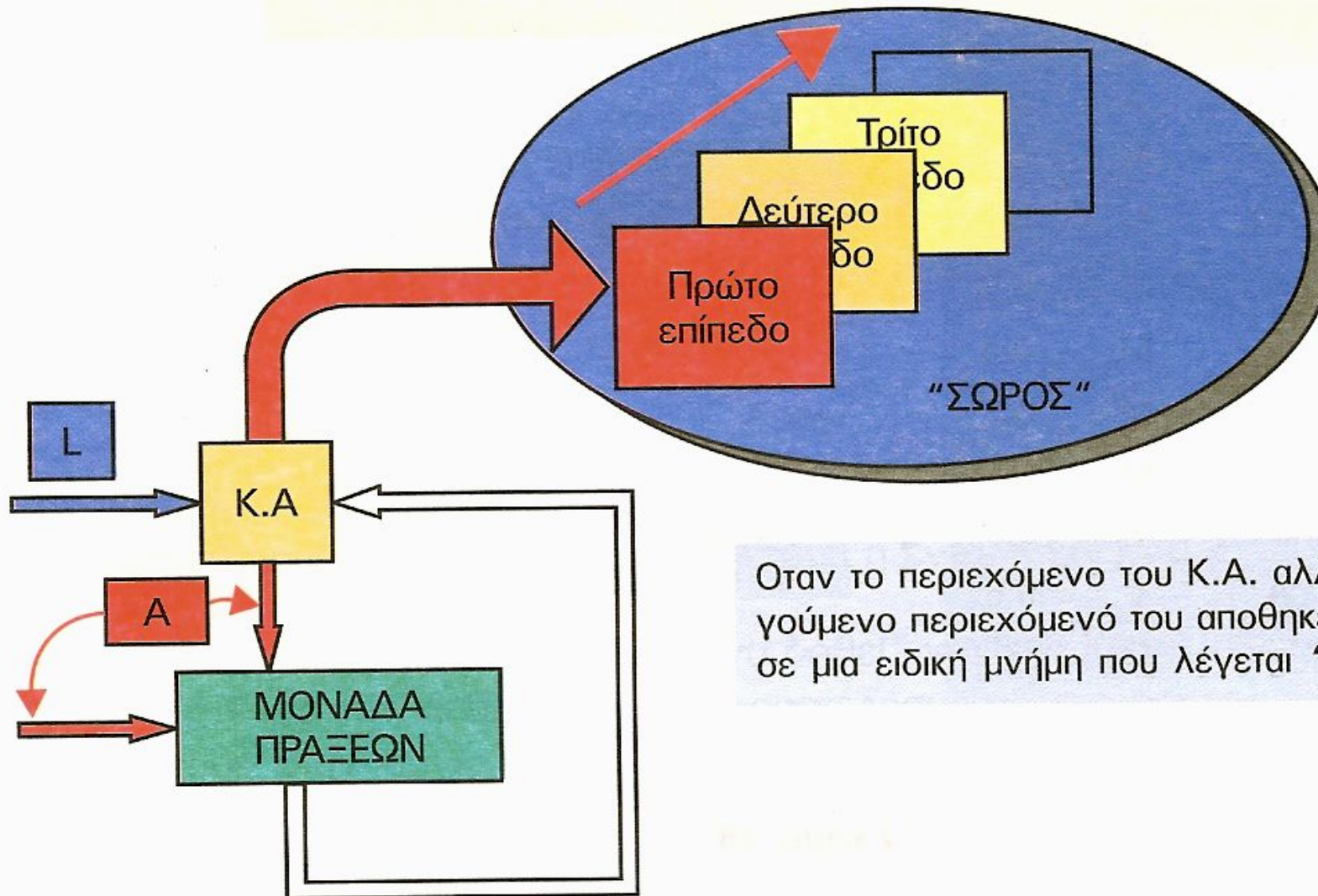
5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

- **Όταν το περιεχόμενο του Κ.Α αλλάζει, το προηγούμενο περιεχόμενο του αποθηκεύεται αυτόματα σε μια περιοχή της μνήμης, που ονομάζεται «σωρός».**
- Ο «σωρός» μπορεί να αποτελείται από μία ή και περισσότερες θέσεις μνήμης. Στην περίπτωση που αποτελείται από περισσότερες θέσεις αυτές διακρίνονται σε επίπεδα.
- Το πρώτο δεδομένο που θα έρθει από τον Κ.Α, θα αποθηκευτεί στο πρώτο επίπεδο, το δεύτερο δεδομένο που θα έρθει θα αποθηκευτεί και πάλι στο πρώτο επίπεδο, ωθώντας το προηγούμενο δεδομένο στο δεύτερο επίπεδο κ.ο.κ.

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

- Με αυτόν τον τρόπο το PLC φυλάσσει τα τελευταία περιεχόμενα του Κ.Α. (τόσα όσες είναι και οι θέσεις του «σωρού»).
- Άρα σε πολύπλοκες συνδεσμολογίες δεν είναι ανάγκη να χρησιμοποιήσουμε την «εικονική» μνήμη για να αποθηκεύσουμε το ενδιάμεσο αποτέλεσμα, αφού το κάνει από μόνο του το PLC με την χρήση του «σωρού».
- Εκεί που διαφέρουν τα PLC μεταξύ τους είναι στις εντολές που χρησιμοποιούν για να ανακαλέσουν τα περιεχόμενα του “σωρού”.

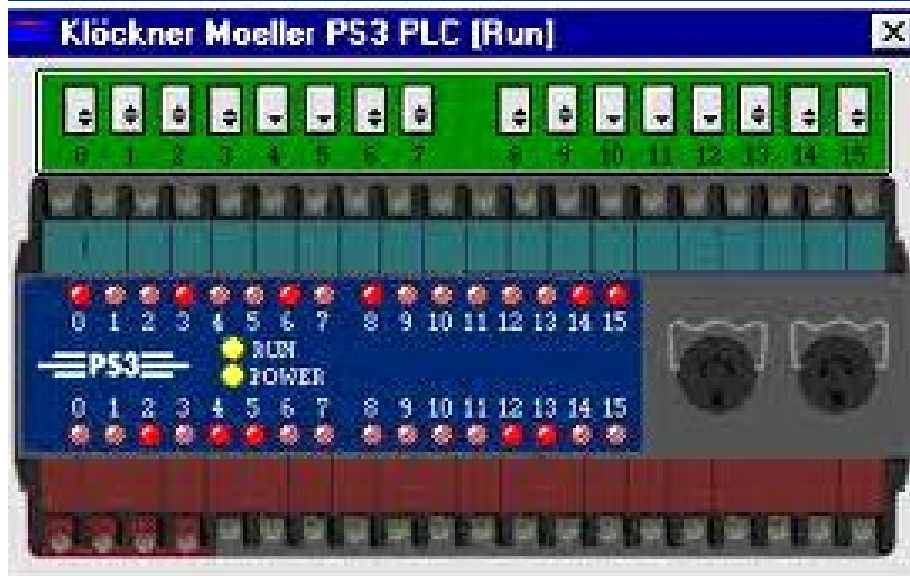
5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



Όταν το περιεχόμενο του Κ.Α. αλλάζει, το προηγούμενο περιεχόμενό του αποθηκεύεται αυτόματα σε μια ειδική μνήμη που λέγεται "σωρός"

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

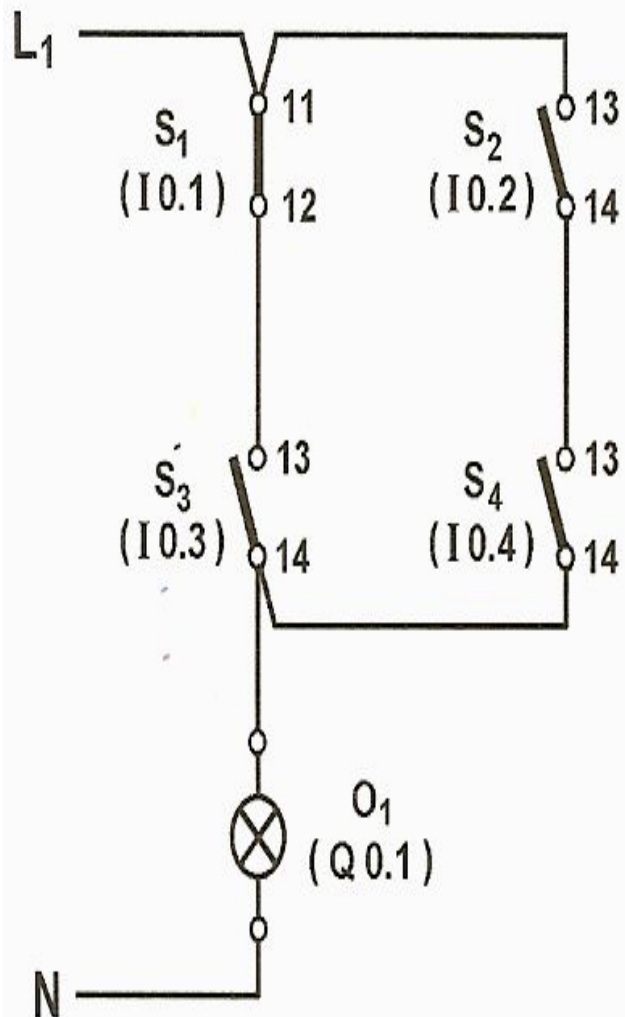
- Η περίπτωση του PLC SUCOS PS 3 της KLOCKER MOELLER.



5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

- **Η περίπτωση του PLC SUCOS PS 3 της KLOCKER MOELLER.**
- Σε μια εντολή **O ή A χωρίς παράμετρο**, το PLC αναλαμβάνει να κάνει την αντίστοιχη λογική πράξη μεταξύ του περιεχομένου του Κ.Α και του περιεχομένου του πρώτου επιπέδου του «σωρού».
- Τα περιεχόμενα του «σωρού» ολισθαίνουν κατά ένα επίπεδο.

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

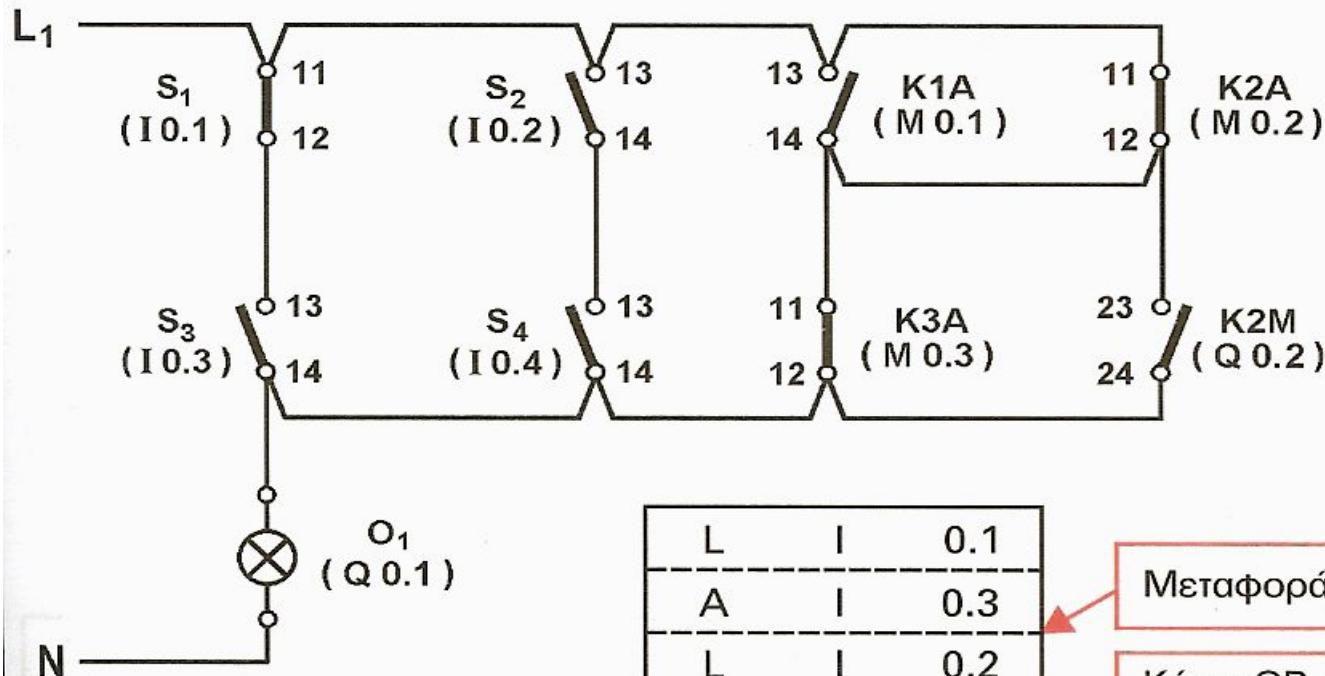


L	I	0.1
A	I	0.3
L	I	0.2
A	I	0.4
O		
=	Q	0.1

Στο σημείο αυτό το αποτέλεσμα της πρώτης λογικής πράξης AND μεταφέρεται στο "σωρό".

Στο σημείο αυτό γίνεται λογική πράξη OR με το περιεχόμενο του Κ.Α. και το περιεχόμενο του πρώτου επιπέδου του "σωρού".

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



L	I	0.1
A	I	0.3
L	I	0.2
A	I	0.4
○		
L	M	0.1
ON	M	0.2
LN	M	0.3
O	Q	0.2
A		
○		
=	Q	0.1

Μεταφορά στο "σωρό"

Κάνει OR με το περιεχόμενο του "σωρού"

Μεταφορά στο "σωρό"

Δεύτερη μεταφορά στο "σωρό"

Λογική πράξη AND με το περιεχόμενο του πρώτου επιπέδου του "σωρού"

Λογική πράξη OR με το νέο περιεχόμενο του πρώτου επιπέδου του "σωρού"

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

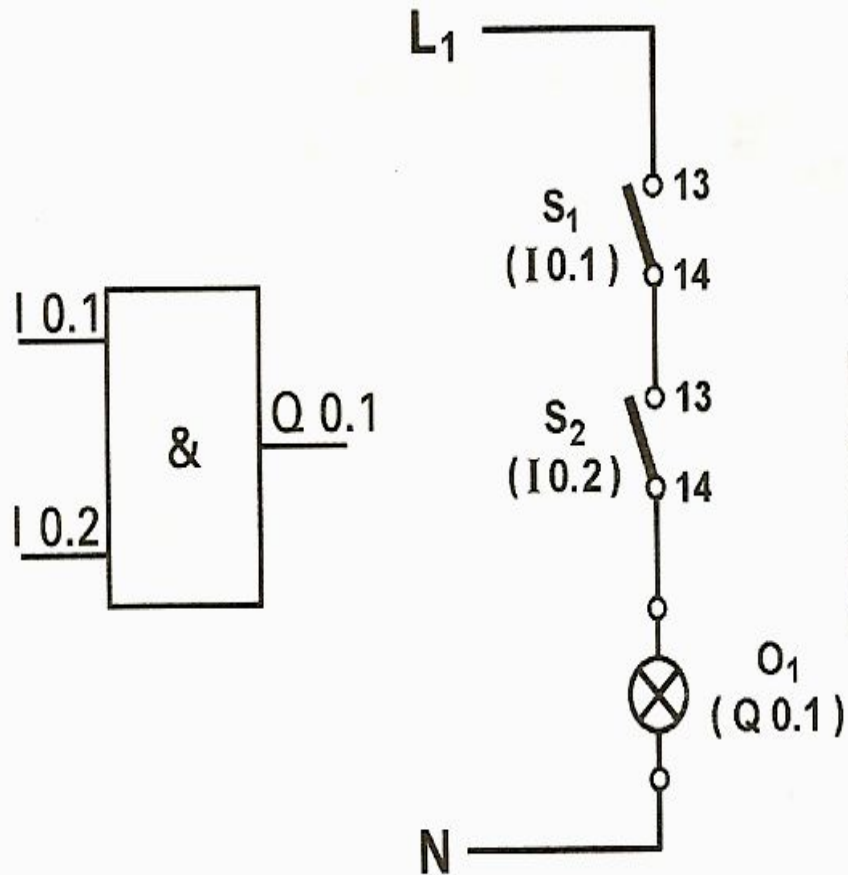
- Η περίπτωση των PLC SIMATIC S7-300 & S7-400 της SIMENS.



5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»

- **Η περίπτωση των PLC SIMATIC S7-300 & S7-400 της SIMENS.**
- Η σειρά S7 της SIMENS παρέχει πολλές διευκολύνσεις στο προγραμματισμό όσον αφορά το θέμα της αντιμετώπισης πολύπλοκων συνδεσμολογιών.
- Υπάρχει μια μικρή διαφοροποίηση μεταξύ S7-300 & S7-400.
- Στον προγραμματισμό των παραπάνω PLC δεν χρησιμοποιείται η εντολή L (Load), αλλά στην θέση της χρησιμοποιείται η εντολή A ή O ανάλογα με το είδος της συνδεσμολογίας που προγραμματίζεται.

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



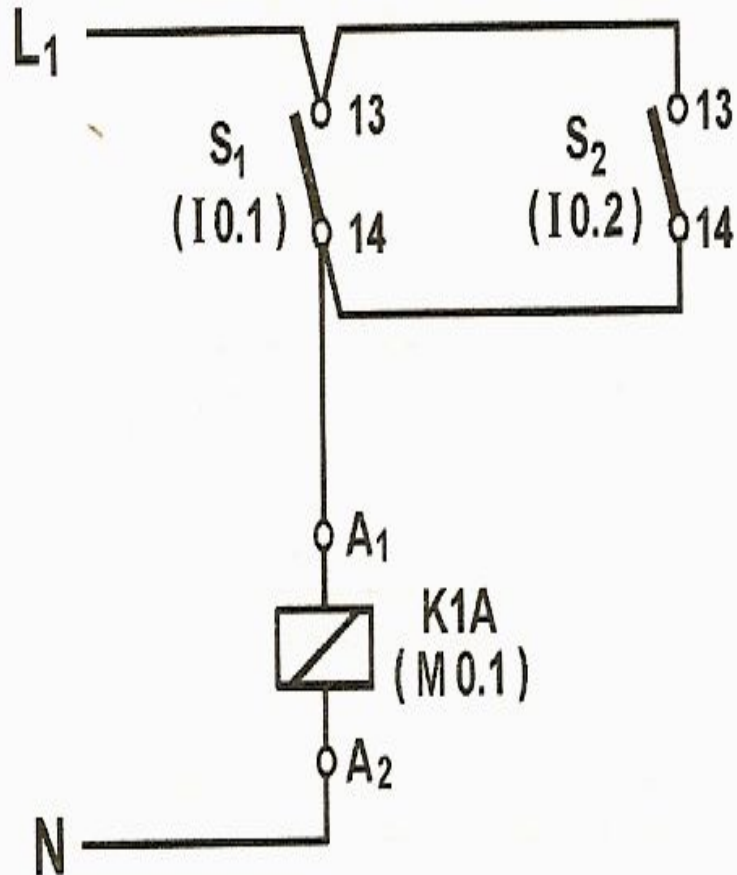
Αντί για:

L	I	0.1
A	I	0.2
=	Q	0.1

Έχουμε:

A	I	0.1
A	I	0.2
=	Q	0.1

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



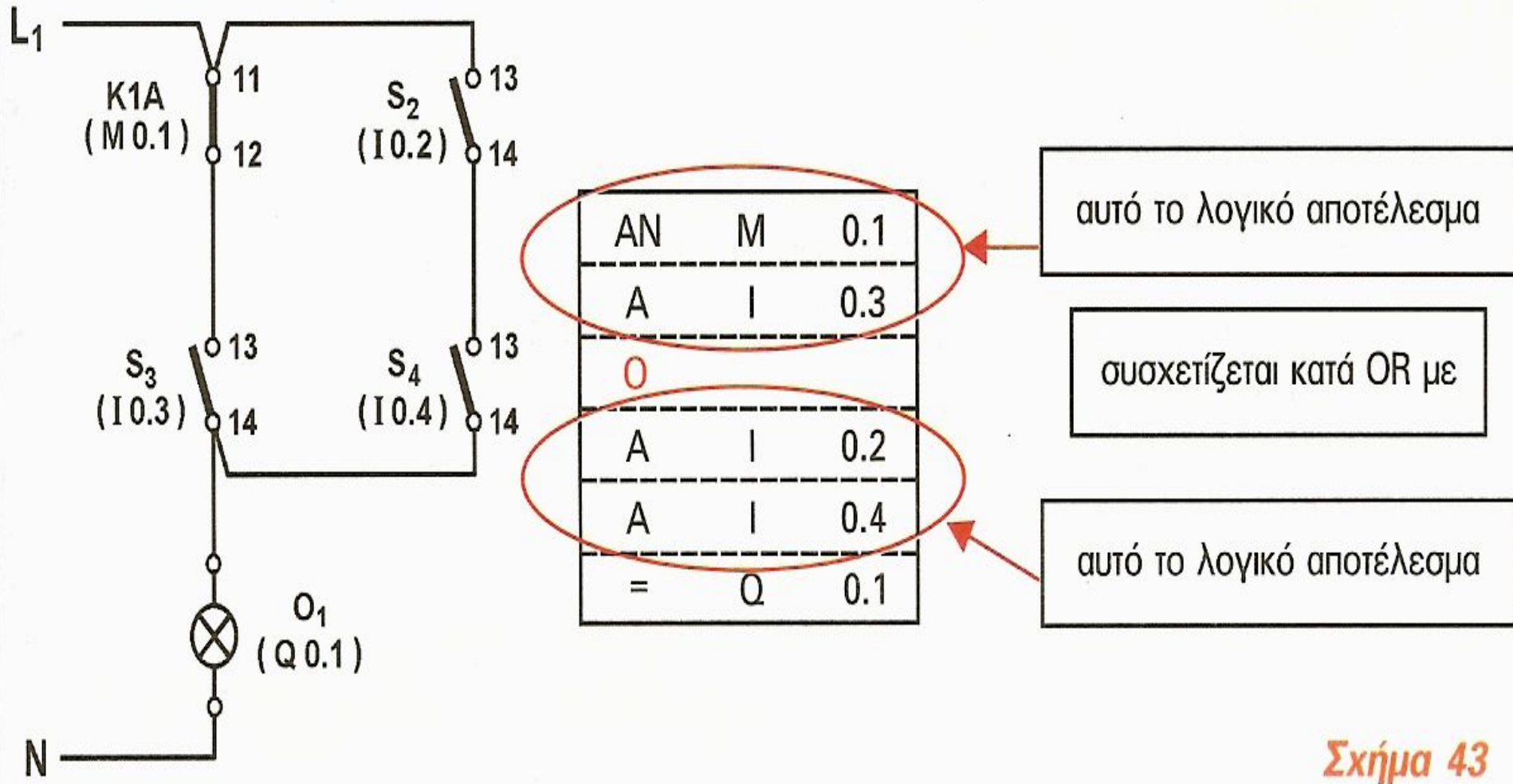
Αντί για:

L	I	0.1
0	I	0.2
=	M	0.1

Έχουμε:

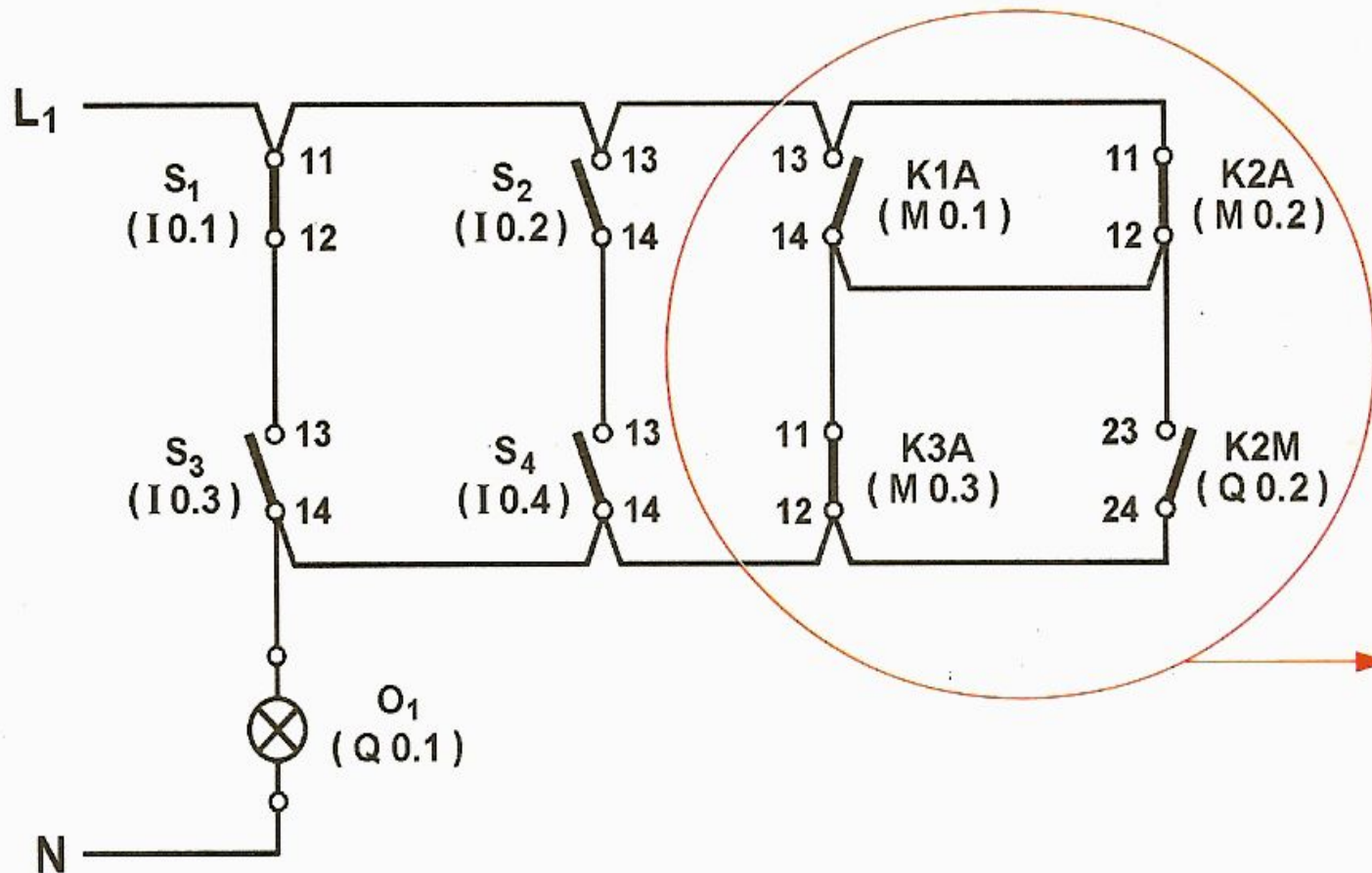
0	I	0.1
0	I	0.2
=	M	0.1

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



Σχήμα 43

5.5 Πολύπλοκες συνδεσμολογίες με την χρήση του «σωρού»



A	I	0.1
A	I	0.3
O		
A	I	0.2
A	I	0.4
O		
A (
O	M	0.1
ON	M	0.2
)		
A (
ON	M	0.3
O	Q	0.2
)		
=	Q	0.1