

**συστηματα μικρουπολογιστων**

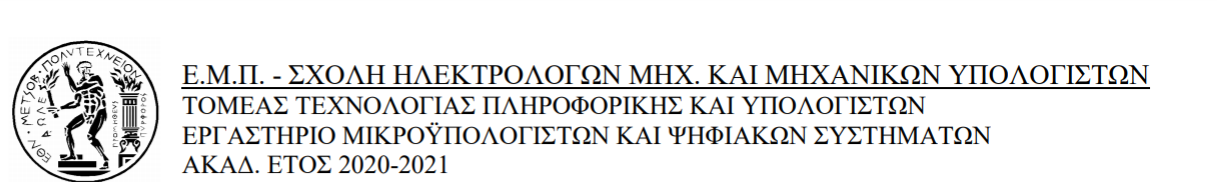
**2η σειρα ασκησεων**



May 4, 2021

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

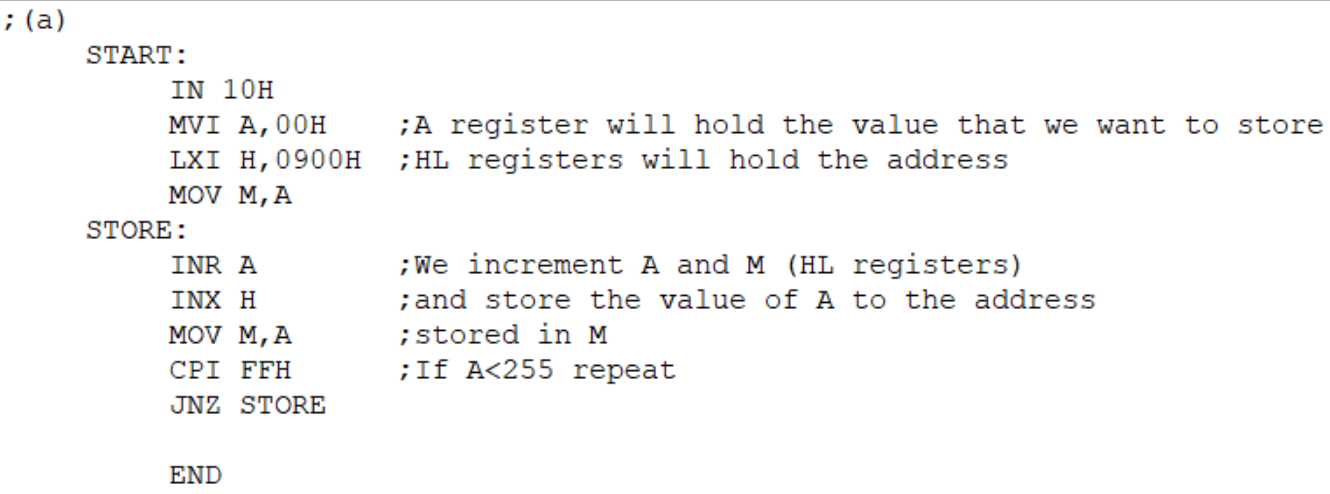
ΚΡΙΣ ΚΟΥΤΣΗ – EL18905

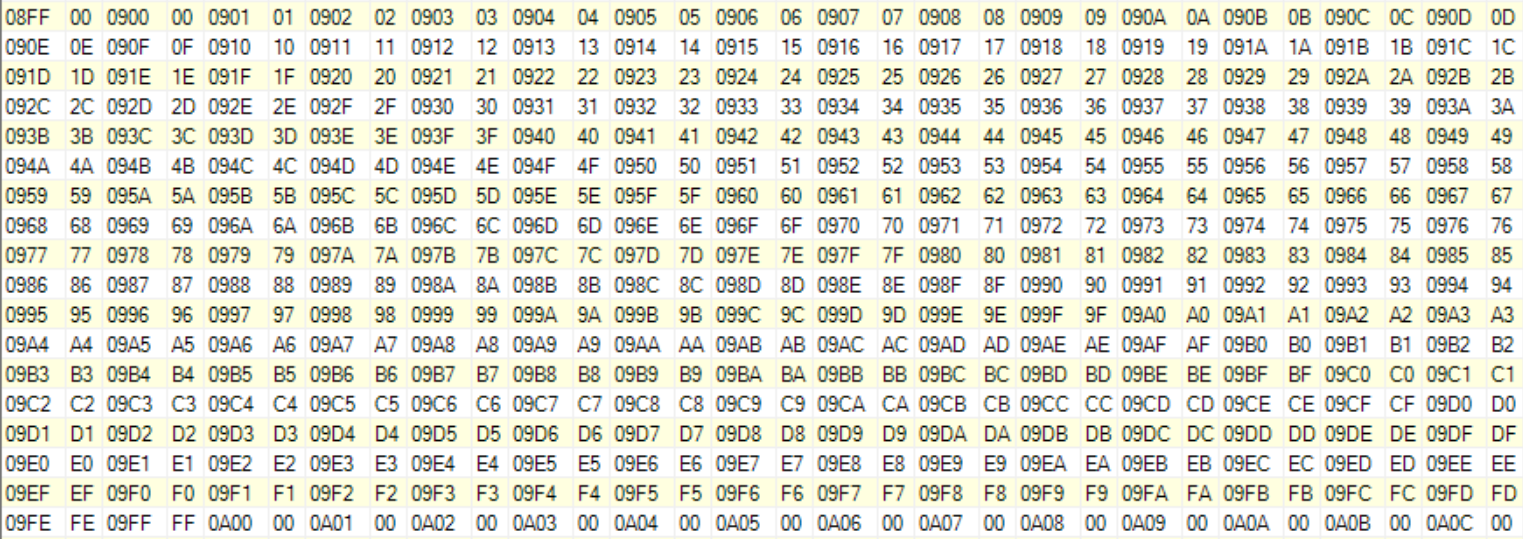


**ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ**

**Άσκηση 1**

**(α)**

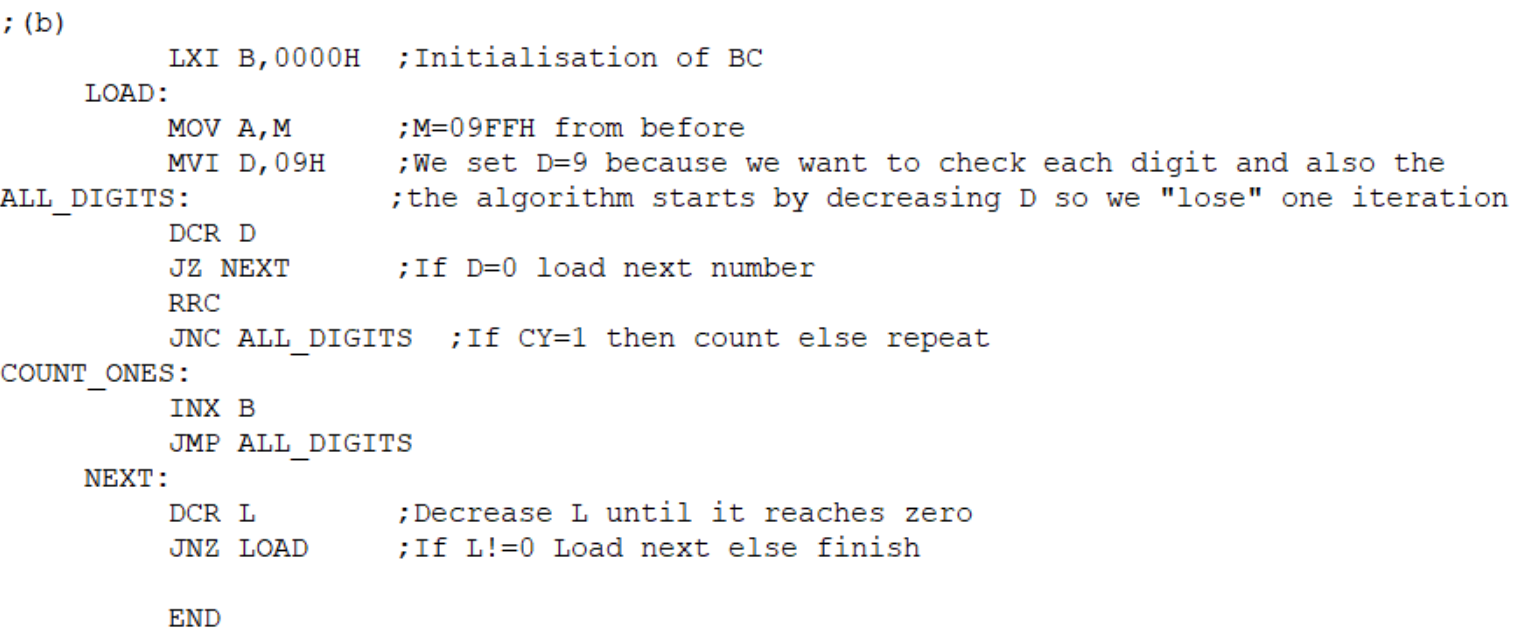
Το ζητούμενο πρόγραμμα είναι το παρακάτω:

Προκειμένου να ελέγξουμε αν λειτουργεί το πρόγραμμα όπως θέλουμε, τοποθετούμε τα δεδομένα στην RAM και έχουμε:

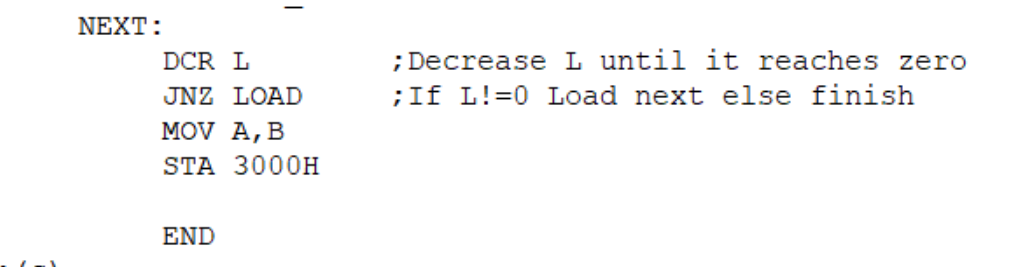
Όπως βλέπουμε παραπάνω, πράγματι οι τιμές αποθηκεύονται με ακριβώς με τον τρόπο που ζητείται.

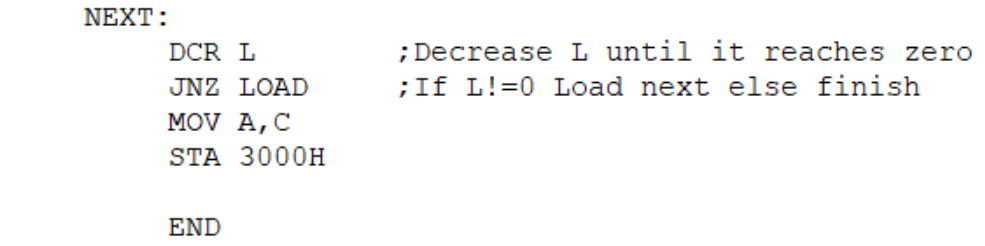
**(β)**

Το πρόγραμμά μας είναι (ως συνέχεια του προηγούμενου προγράμματος):



Γνωρίζουμε ότι στο εύρος αριθμών 0-255 υπάρχουν 2048 ψηφία, από τα οποία τα μισά (1024) είναι μηδέν και τα άλλα μισά άσσοι. Συνεπώς, προκειμένου να ελέγξουμε την ορθότητα του προγράμματός μας, τρέχουμε το πρόγραμμα δύο φορές, τοποθετώντας πριν το τέλος αρχικά τον κώδικα που φαίνεται στην πρώτη φωτογραφία και ύστερα τον κώδικα που φαίνεται στην δεύτερη. Με αυτόν τον τρόπο μπορούμε να δούμε τον αριθμό που περιέχεται στον διπλό καταχωρητή (BC). Σύμφωνα με την ένδειξη των LED (αντίστροφη λογική), ισχύει: (BC) = (0400)HEX = (1024)DEC. Άρα, το πρόγραμμά μας λειτουργεί με τον επιθυμητό τρόπο.





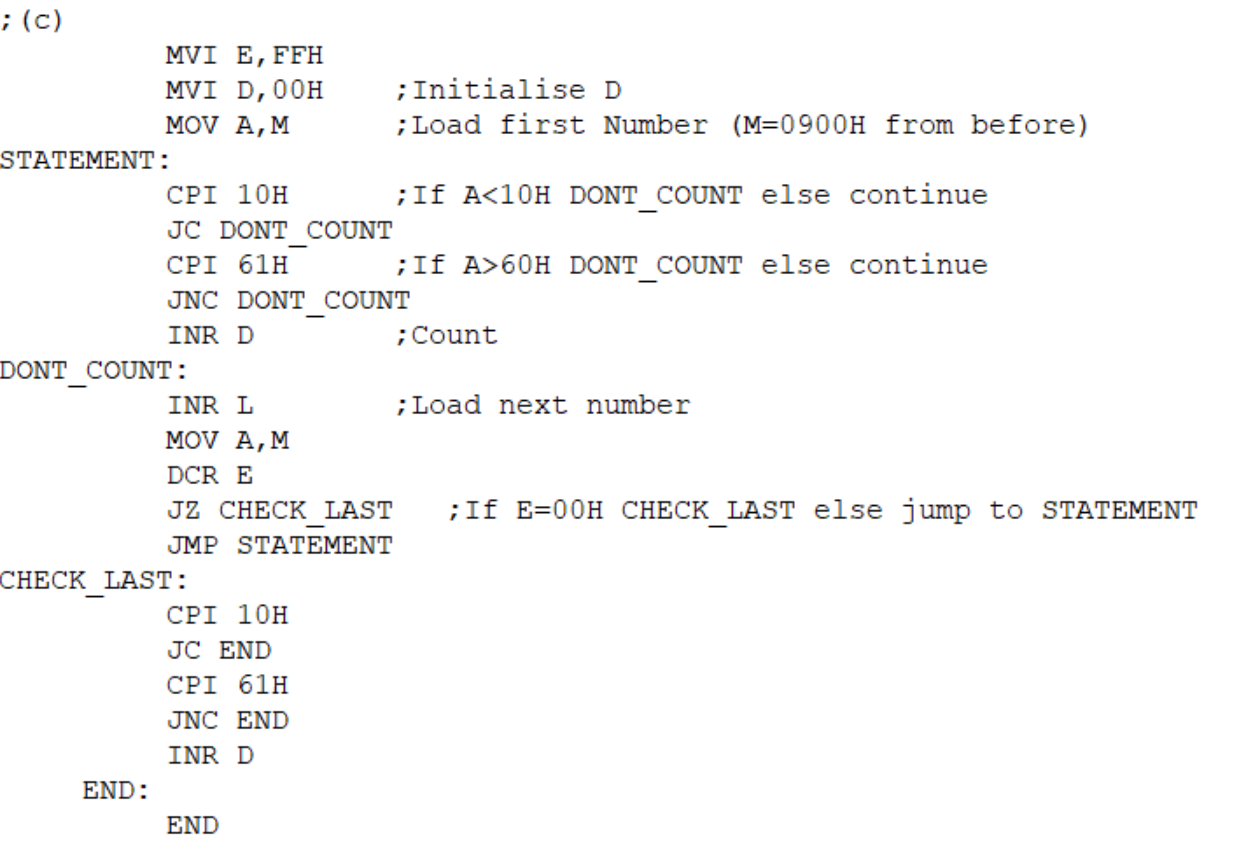


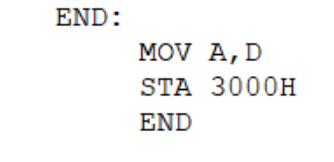
**(γ)**

Γράφουμε το πρόγραμμα με την λογική ότι τα νούμερα (0-255) μπορεί να έχουν αποθηκευτεί σε τυχαίες θέσεις στο κομμάτι μνήμης 0900H – 09FFH επιπλέον υποθέτουμε ακόμα ότι κάποιο νούμερο μπορεί να υπάρχει περισσότερες από μία φορές ή να μην υπάρχει καθόλου. Χωρίς αυτές τις υποθέσεις, το η απάντηση θα ήταν ξεκάθαρη (μια απλή αφαίρεση). Η απάντηση που θα θέλαμε στο συγκεκριμένο πρόβλημα είναι:

(60)HEX – (10)HEX + 1 = (51)HEX ή (96)DEC – (16)DEC + 1 = (81)DEC

Ελέγχουμε την ορθότητα του προγράμματός μας εκτυπώνοντας την τιμή του καταχωρητή D, εφαρμόζοντας την αλλαγή που φαίνεται κάτω από το πρόγραμμα. Η ένδειξη των LED συμφωνεί με τα παραπάνω αποτελέσματα, άρα το πρόγραμμα λειτουργεί σωστά.

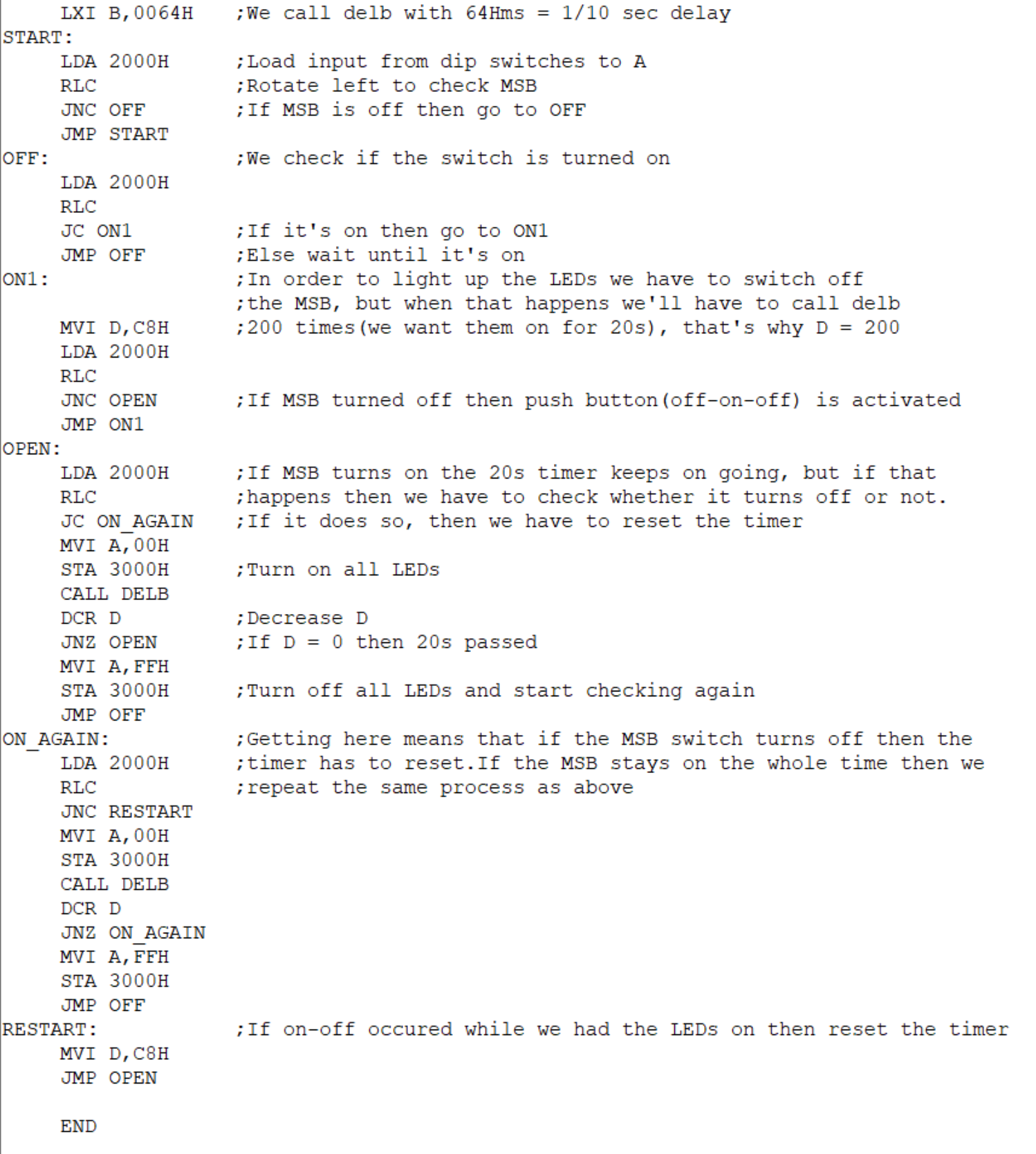


Στην ταμπέλα END κάνουμε την ακόλουθη αλλαγή προκειμένου να ελέγξουμε την ορθότητα του προγράμματος:

Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_1.8085*.

**Άσκηση 2**

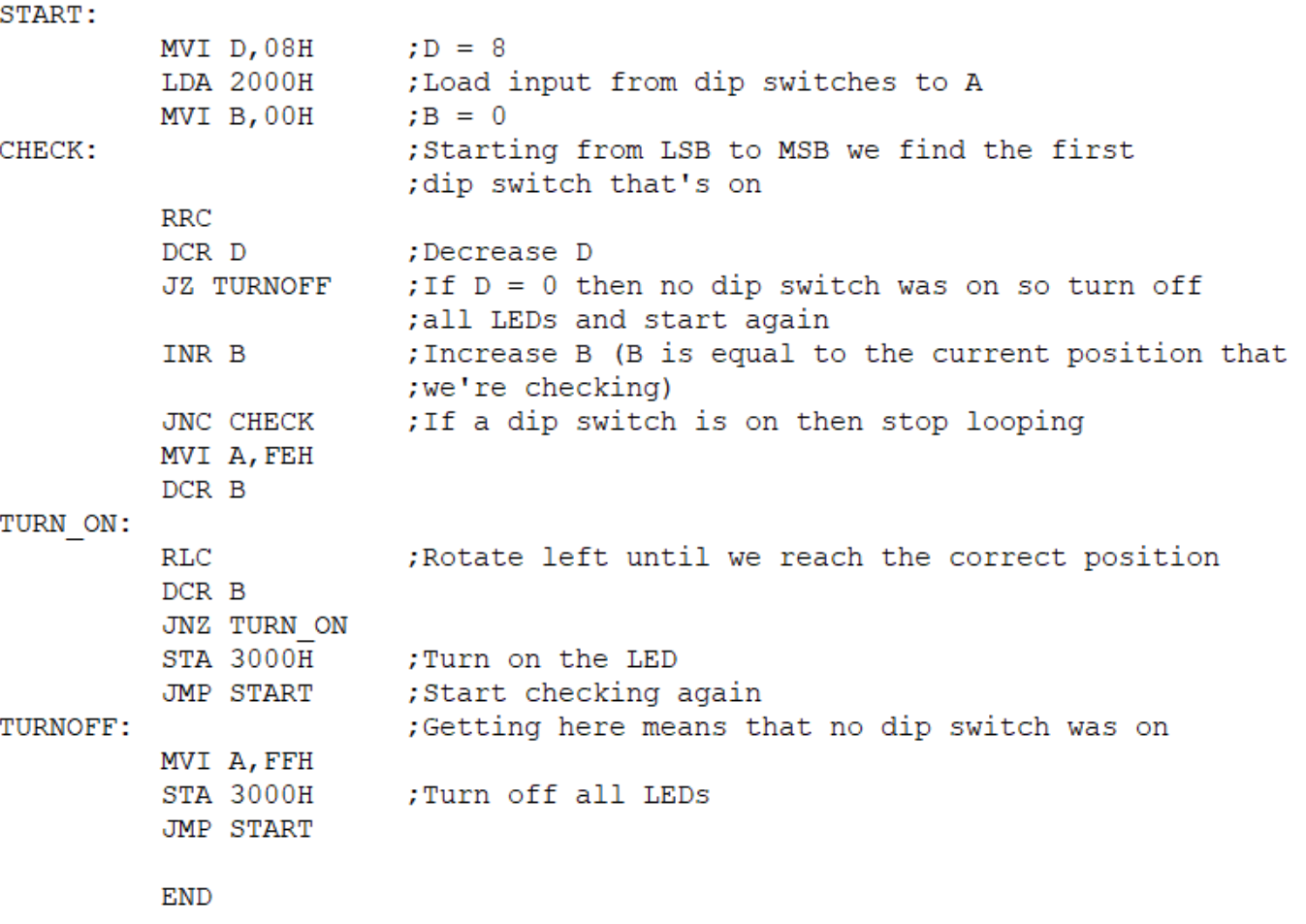
Το πρόγραμμά μας είναι το ακόλουθο:



Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_2.8085*

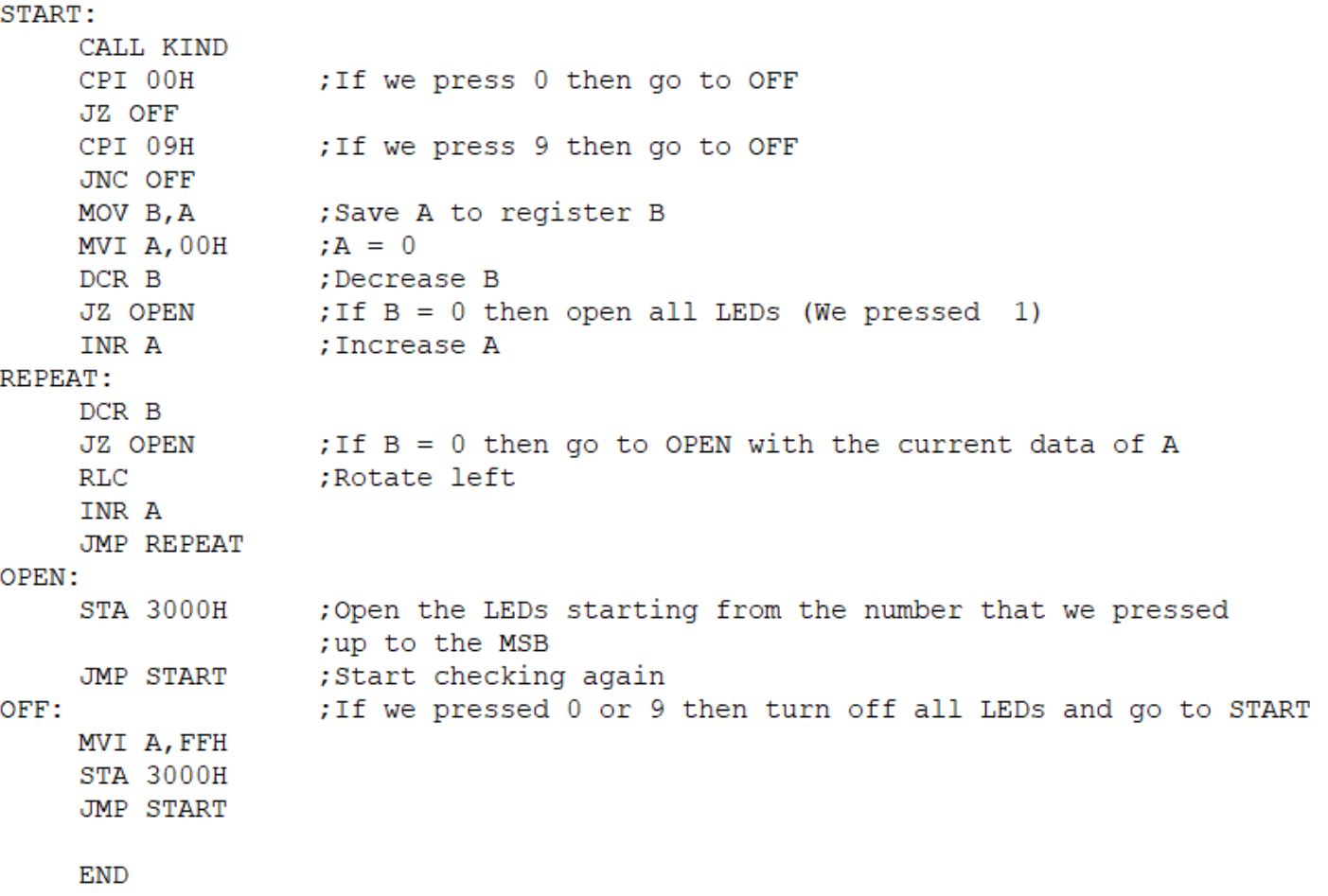
**Άσκηση 3**

Τα προγράμματά μας για κάθε ερώτημα δίνονται παρακάτω:

**(i)**

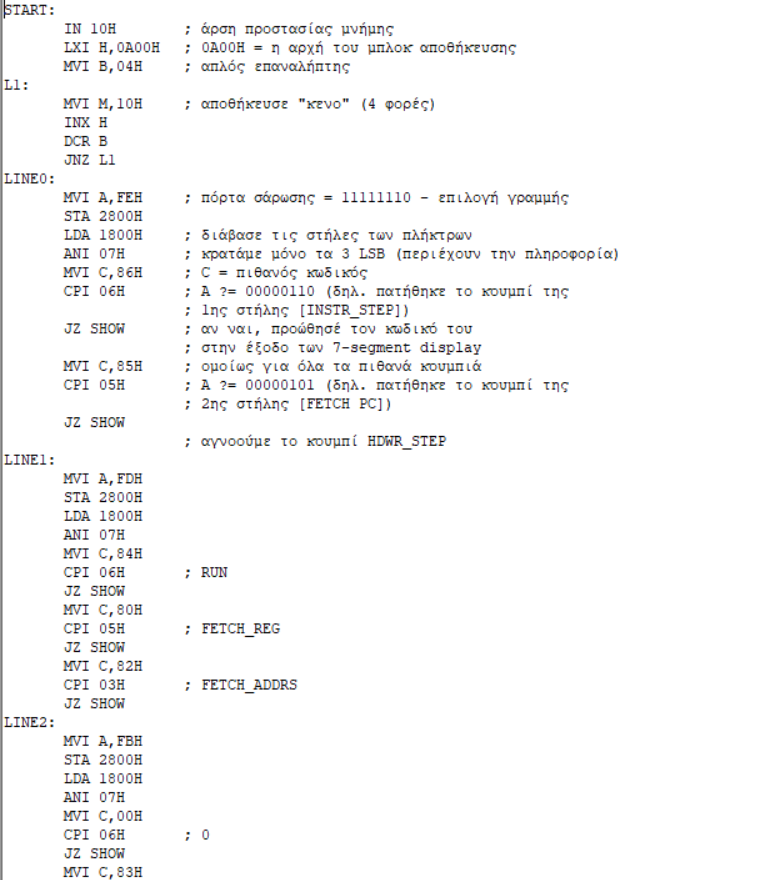
Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_3i.8085*

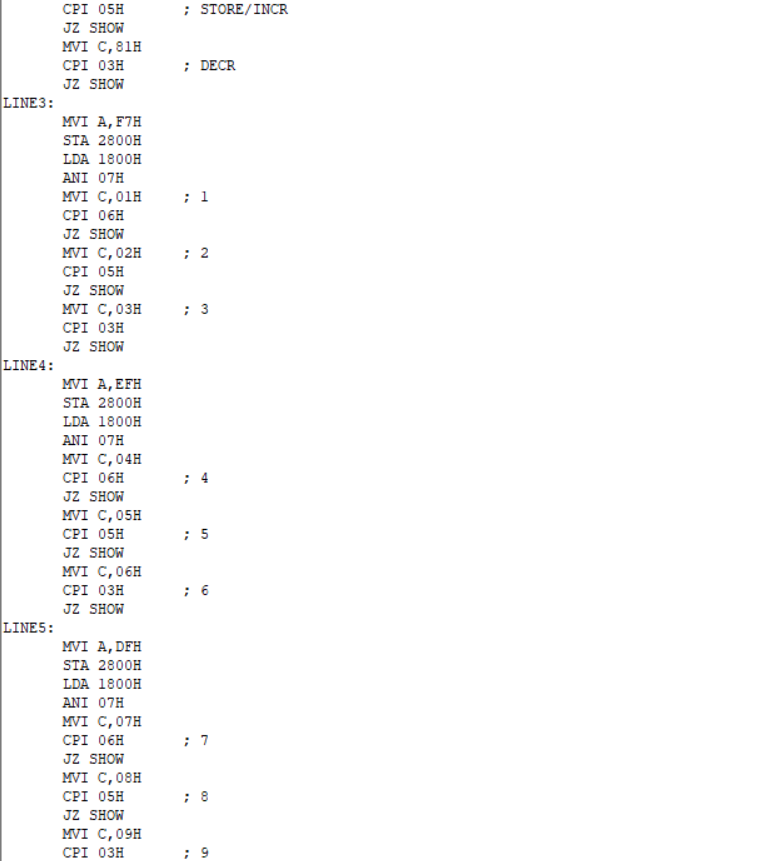
**(ii)**

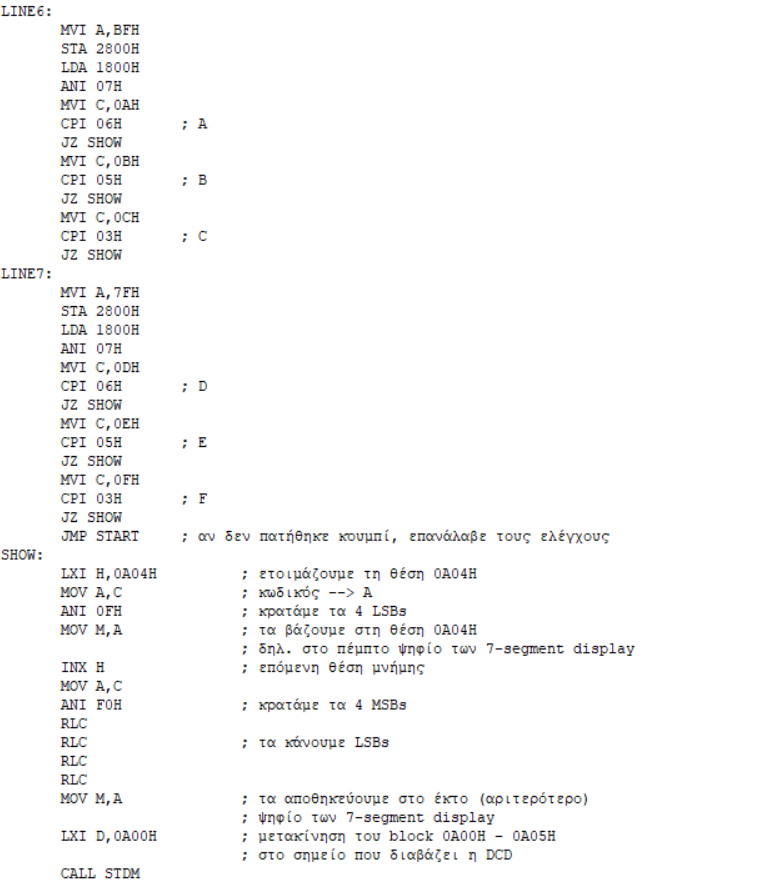
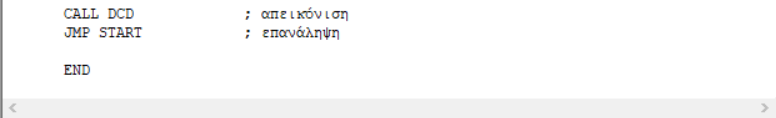


Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_3ii.8085*

**(iii)**

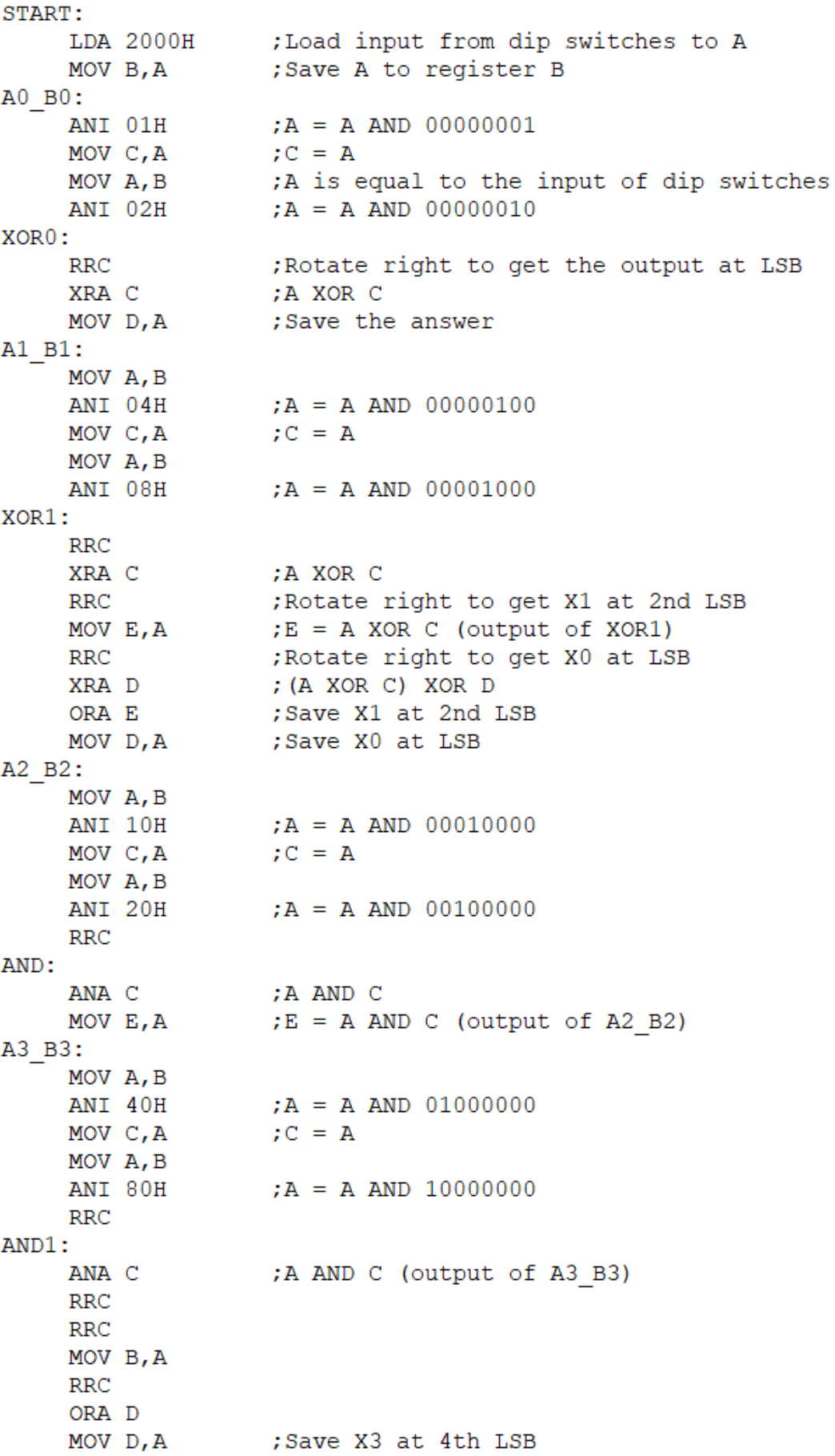
****

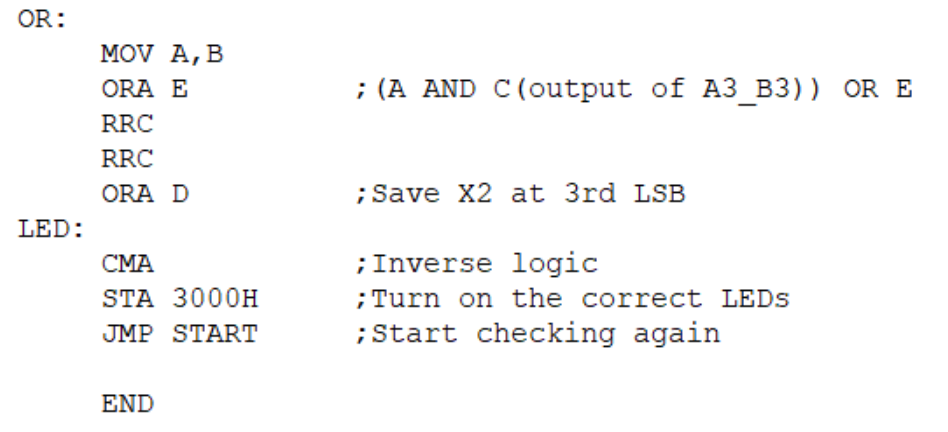
****

****

Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_3iii.8085*

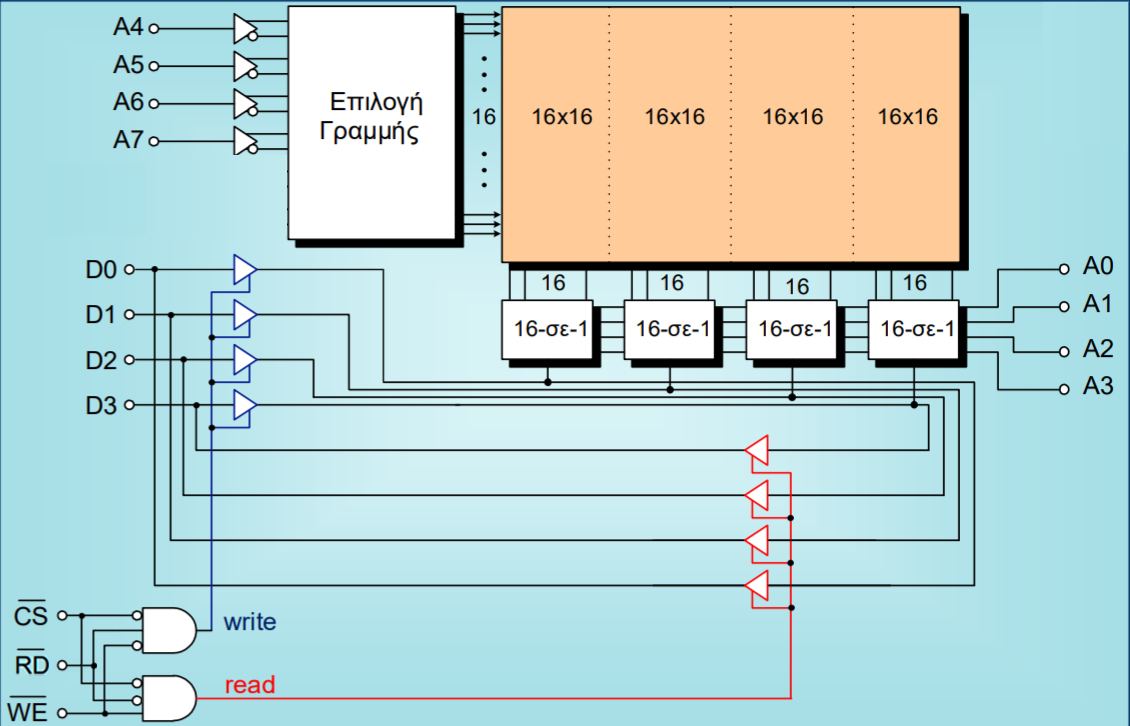
**Άσκηση 4**

Το πρόγραμμά μας είναι το ακόλουθο:



Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση\_4.8085*

**Άσκηση 5**



Παραπάνω φαίνεται η εσωτερική δομή μίας SRAM 256x4 bit (16x16x4). Από τον πίνακα της μνήμης επιλέγεται με βάση τις γραμμές διεύθυνσης Α4-Α7 μια από τις 16 γραμμές. Οι γραμμές D0-D3 αποτελούν τις γραμμές δεδομένων, οι οποίες συνδέονται με τον πίνακα της μνήμες μέσω τεσσάρων πολυπλεκτών 16-σε-1. Οι πολυπλέκτες αυτοί επιλέγουν μία από τις 16 τετράδες-στήλες του πίνακα μνήμης (μία ο καθένας), με βάση τις γραμμές διευθύνσεων Α0-Α3, όπου, σε συνδυασμό με την επιλεγμένη γραμμή του πίνακα, είτε εγγράφονται τα δεδομένα D0-D3 στις θέσεις αυτές, είτε διαβάζονται, δηλαδή μεταφέρονται στις γραμμές D0-D3, τα δεδομένα των θέσεων αυτών.

Για παράδειγμα αν είχαμε μία διεύθυνση Α0…Α7 = 0110 1001, τότε επιλέγεται η 9η (1001) γραμμή και η 6η (0110) τετράδα του πίνακα μνήμης.

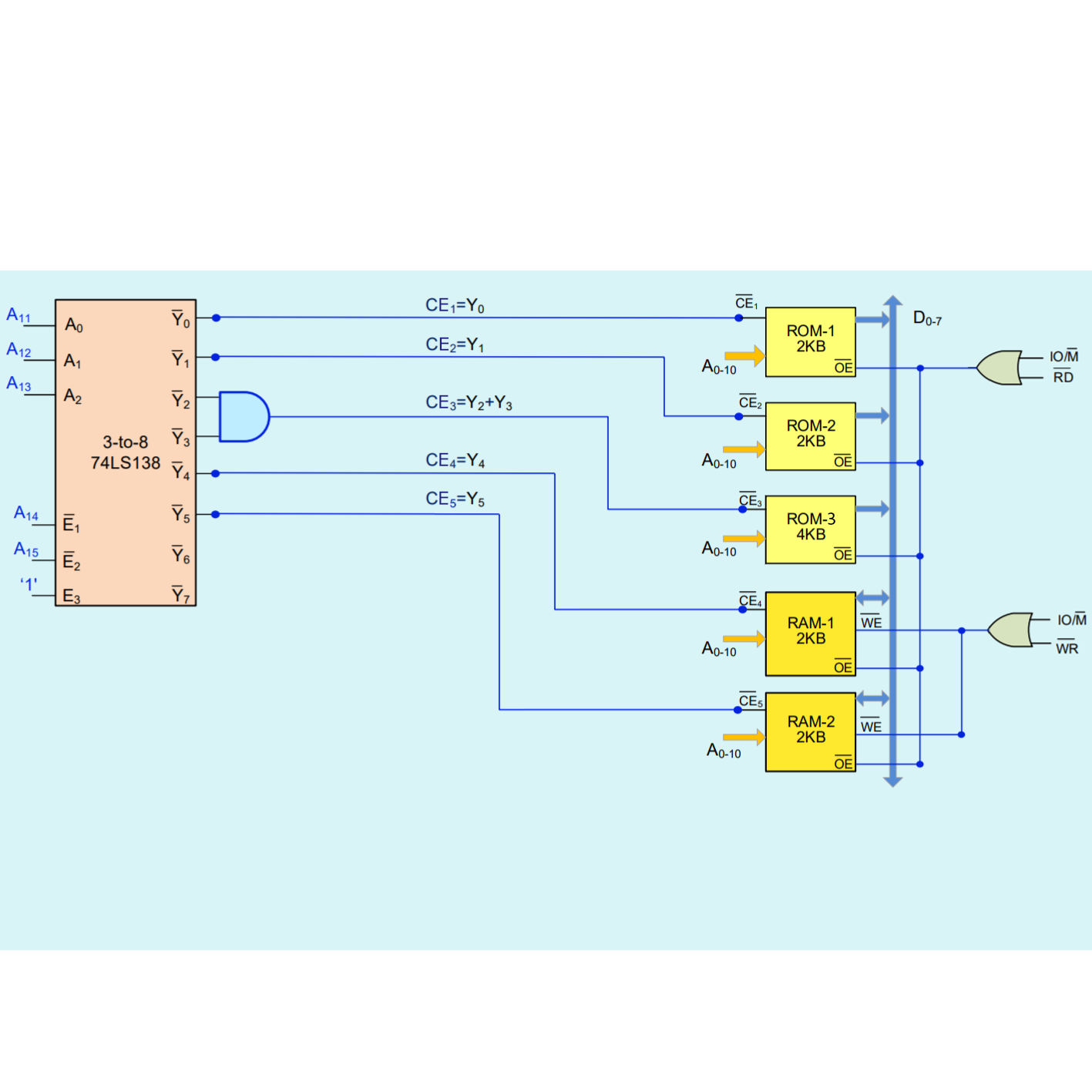
Όσον αφορά το πότε γίνεται εγγραφή ή ανάγνωση, αυτό καθορίζεται από τα τρία σήματα Όταν το σήμα γίνει 0, ενεργοποιείται η λειτουργία της μνήμης. Στη συνέχεια, Αν το σήμα γίνει 0 και το γίνει 1, τότε θα ενεργοποιηθούν οι απομονωτές με μπλε περίγραμμα και θα έχουμε εγγραφή στην μνήμη. Αν το σήμα γίνει 1 και το γίνει 0, τότε θα ενεργοποιηθούν οι απομονωτές με κόκκινο περίγραμμα και θα έχουμε διάβασμα από τη μνήμη.

**Άσκηση 6**

Ο χάρτης μνήμης του συστήματος μνήμης είναι:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** | **Address** | **Memory** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | ROM1-2K |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 07FF |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0800 | ROM2-2K |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0FFF |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 | ROM3-4K |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1FFF |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2000 | RAM1-2K |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 27FF |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2800 | RAM2-2K |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2FFF |

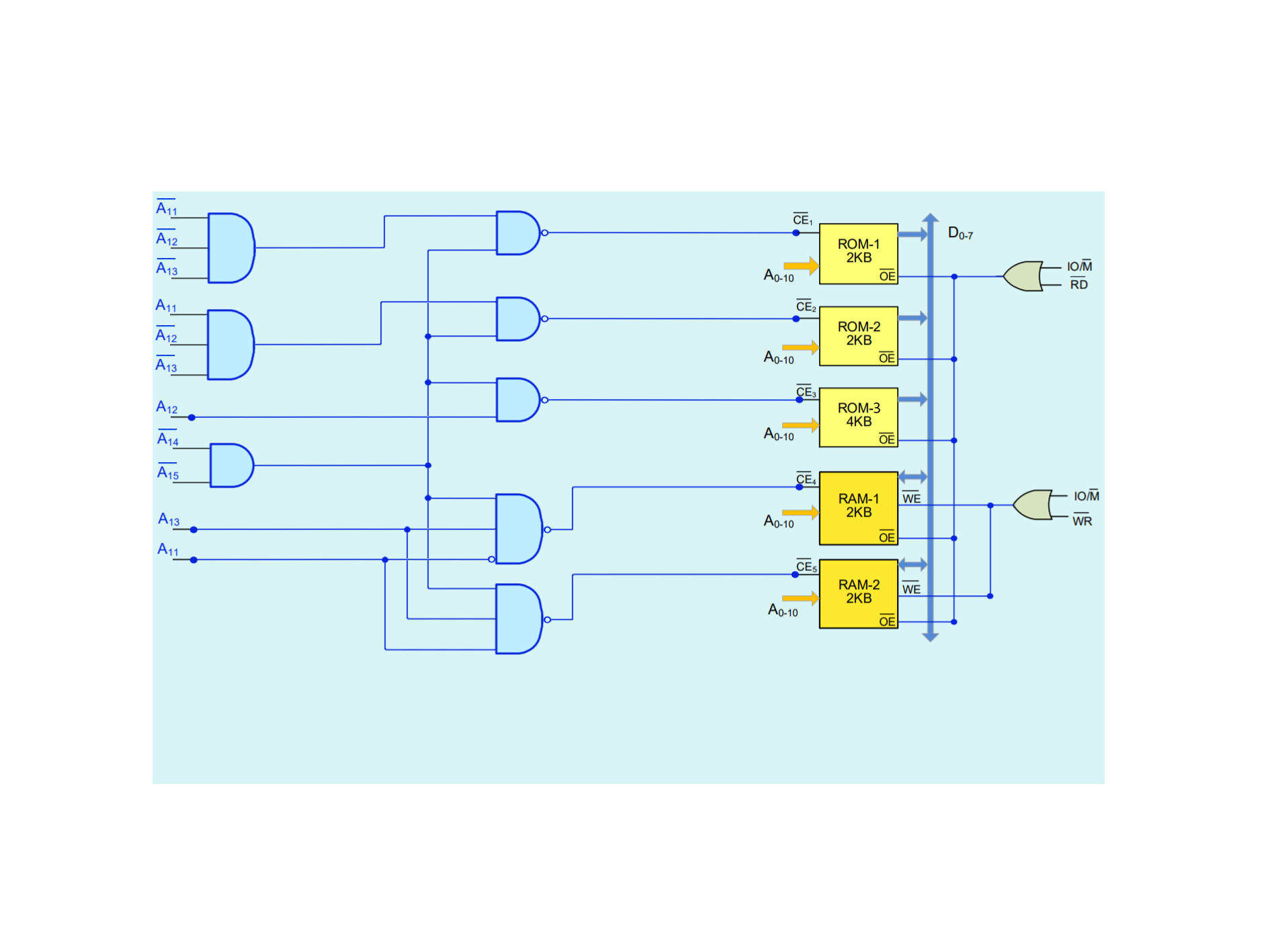
**(α)**



Το σύστημα μνήμης υλοποιημένο με αποκωδικοποιητή και λογικές πύλες:

**(β)**

Το σύστημα μνήμης υλοποιημένο μόνο με λογικές πύλες:



**Άσκηση 7**

Ο χάρτης μνήμης είναι:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** | **Address** | **Memory** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | ROM1-12K |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2FFF |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3000 | RAM1-4K |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3FFF |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4000 | RAM2-4K |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4FFF |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5000 | RAM3-4K |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5FFF |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6000 | ROM1-4K |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 6FFF |

Το σύστημα μνήμης είναι:

