



Σχολή Ηλεκτρολόγων Μηχανικών
και
Μηχανικών Υπολογιστών

Συστήματα Μικροϋπολογιστών [Ροή Υ]

2^η Ομάδα Ασκήσεων

Δημήτρης Δήμος

031 17 165

6^ο Εξάμηνο

Μάιος 2020

1^η, 2^η, 3^η και 4^η Άσκηση

Στον προσομοιωτή του εκπαιδευτικού προγράμματος μLab αναπτύχθηκαν τα πρόγραμματα [group2.ex1.8085](#), [group2.ex2.8085](#), [group2.ex3i.8085](#), [group2.ex3ii.8085](#), [group2.ex3iii.8085](#) και [group2.ex4.8085](#) σε assembly, τέτοια ώστε να πληρούν τα ζητούμενα των αντίστοιχων εκφωνήσεων. Οι κώδικες βρίσκονται στο συμπιεσμένο αρχείο που παραδόθηκε και περιλαμβάνουν διευκρινιστικά σχόλια αναφορικά με τις επιμέρους λειτουργίες που επιτελούν οι υπορουτίνες προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα.

5^η Άσκηση

Η οργάνωση που παρουσιάζεται αφορά μνήμη SRAM μεγέθους 128 x 4 bit. Επομένως, συμπεραίνουμε ότι η μνήμη έχει χώρο για 128 λέξεις μεγέθους 4 bit. Χωρίζουμε, λοιπόν, την μνήμη σε 4 τμήματα - τράπεζες που το καθένα έχει μέγεθος 128 bit. Κάθε λέξη “μοιράζει” από ένα bit σε κάθε τράπεζα. Η κάθε τράπεζα θα έχει δύο διαστάσεις α και β , τέτοιες ώστε $\alpha \times \beta = 128$. Επιλέγουμε, αυθαίρετα (με σκοπό να επιτύχουμε σχήμα όσο πιο κοντά στο τετραγωνικό), διαστάσεις 16 x 8.

Για να επιλέξουμε γραμμή χρειαζόμαστε **$\log_2 16 = 4$ bit διεύθυνσης**

Για να επιλέξουμε στήλη χρειαζόμαστε **$\log_2 8 = 3$ bit διεύθυνσης**

Για τους λόγους αυτούς, οι ακροδέκτες διευθύνσεων A0 έως A2 χρησιμοποιούνται για την επιλογή στήλης, με χρήση πολυπλεκτών-αποπλεκτών 8-σε-1 και οι ακροδέκτες διευθύνσεων A3 έως A6 χρησιμοποιούνται για την επιλογή γραμμής, με χρήση αποκωδικοποιητή 4-σε-16.

Οι ακροδέκτες D0 έως D3 είναι οι ακροδέκτες εισόδου/εξόδου 4 bit και τα σήματα \overline{CS} , \overline{WE} , \overline{RD} μέσω των αντίστοιχων ακροδεκτών επιτρέπουν ή αποτρέπουν τις λειτουργίες εγγραφής και ανάγνωσης. Τα παραπάνω εξηγούνται αναλυτικότερα μέσα από ένα παράδειγμα.

Έστω ότι επιθυμούμε να:

- 1) εγγράψουμε την λέξη 0101 στην θέση μνήμης 0010101 και ύστερα να
- 2) διαβάσουμε την λέξη της θέσης μνήμης 0100010

Για την πραγματοποίηση του (1):

Αρχικά, στις εισόδους διεύθυνσης A0 έως A7 εφαρμόζεται η διεύθυνση εγγραφής, δηλαδή η 0010101. Εφαρμόζονται τα δεδομένα προς εγγραφή (δηλαδή η λέξη 0101) τους ακροδέκτες εισόδου D0 έως D3. Στέλνεται αρνητικός παλμός στον ακροδέκτη επιλογής \overline{CS} του κατάλληλου ολοκληρωμένου κυκλώματος SRAM και προκειμένου να γίνει η εγγραφή της λέξης στέλνεται αρνητικός παλμός στον ακροδέκτη \overline{WE} , ενώ στον ακροδέκτη \overline{RD} στέλνεται θετικός παλμός. Έτσι η έξοδος της άνω πύλης AND (write) γίνεται 1 και της κάτω (read) 0. Επιλέγεται, δηλαδή, το chip για χρήση και σημαίνεται η εγγραφή στη μνήμη, ενώ αποτρέπεται η ανάγνωση.

Οι τρισταθείς buffers που “κοιτούν” προς τα δεξιά ενεργοποιούνται κι επιτρέπουν την διέλευση των σημάτων D0 έως D3, ενώ οι τρισταθείς buffers που “κοιτούν” προς τα αριστερά τίθενται σε κατάσταση υψηλής αντίστασης.

Έτσι, τα δεδομένα D0 έως D3 φτάνουν στους πολυπλέκτες-αποπλέκτες (λειτουργία αποπλέκτη) και τα σήματα επιλογής A0 έως A2 οδηγούν το καθένα στη στήλη 101 (δηλαδή 5 στο δεκαδικό) της αντίστοιχης τράπεζας. Τα σήματα A3 έως A6 επιλέγουν στη σειρά που αντιστοιχεί στον ελαχιστόρο 0010, δηλαδή τη σειρά 2 στο δεκαδικό.

Έτσι, το bit D0 της λέξης μας εγγράφεται στην πρώτη από αριστερά τράπεζα στη θέση (2, 5), το bit D1 της λέξης εγγράφεται στη δεύτερη από αριστερά τράπεζα στη θέση

(2, 5), το bit D2 της λέξης εγγράφεται στην τρίτη από αριστερά τράπεζα στη θέση (2, 5) και το bit D3 της λέξης εγγράφεται στην τέταρτη από αριστερά τράπεζα στη θέση (2, 5).

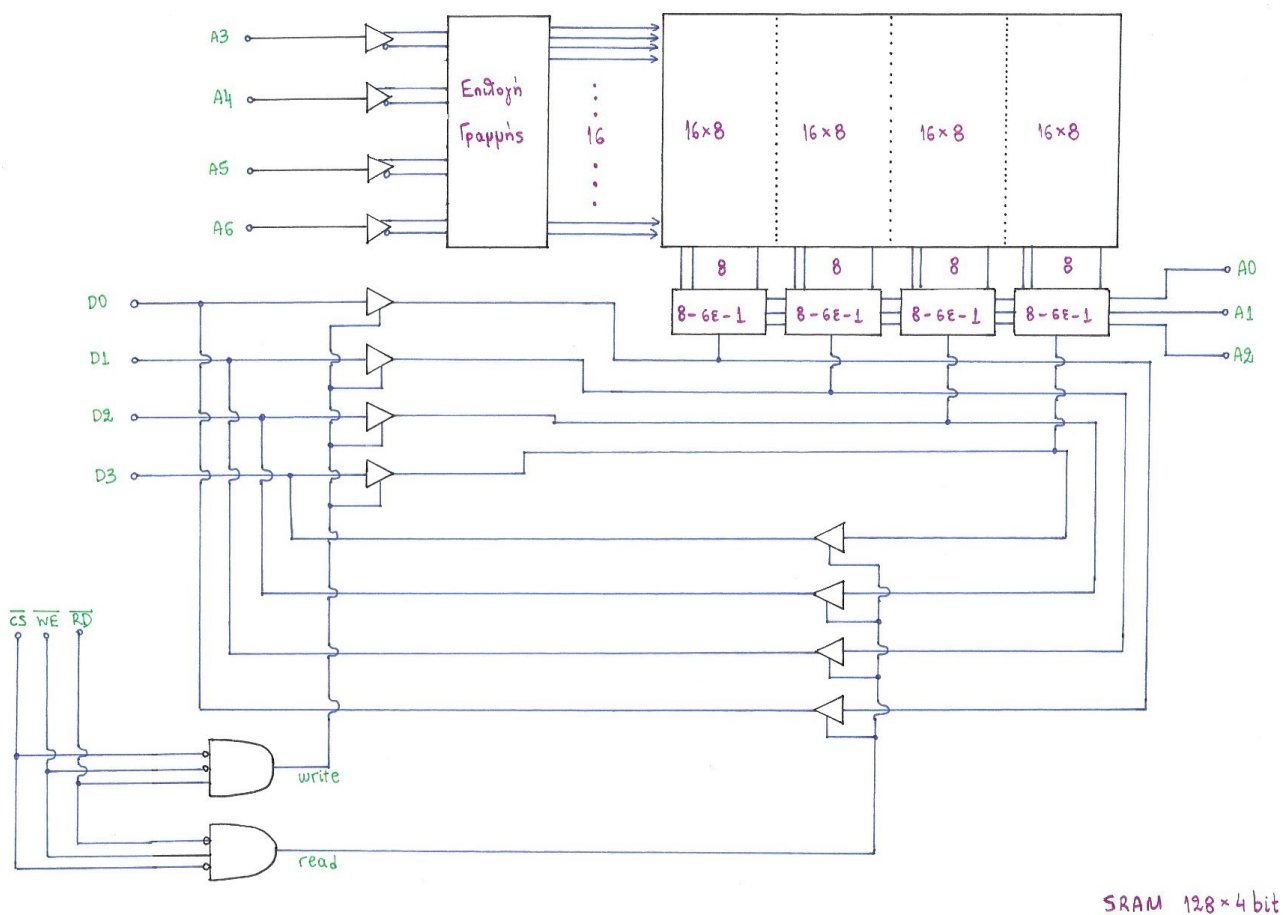
Για την πραγματοποίηση του (2):

Αρχικά, στις εισόδους διεύθυνσης A0 έως A7 εφαρμόζεται η διεύθυνση ανάγνωσης, δηλαδή η 0100010. Στέλνεται αρνητικός παλμός στον ακροδέκτη επιλογής \overline{CS} του κατάλληλου ολοκληρωμένου κυκλώματος SRAM και προκειμένου να γίνει η ανάγνωση της λέξης στέλνεται αρνητικός παλμός στον ακροδέκτη \overline{RD} , ενώ στον ακροδέκτη \overline{WE} στέλνεται θετικός παλμός. Έτσι η έξοδος της κάτω πύλης AND (read) γίνεται 1 και της άνω (write) 0. Επιλέγεται, δηλαδή, το chip για χρήση και σημαίνεται η ανάγνωση της μνήμης, ενώ αποτρέπεται η εγγραφή.

Οι τρισταθείς buffers που “κοιτούν” προς τα δεξιά τίθενται σε κατάσταση υψηλής αντίστασης, ενώ οι τρισταθείς buffers που “κοιτούν” προς τα αριστερά ενεργοποιούνται.

Τα σήματα A3 έως A6 επιλέγουν στη σειρά που αντιστοιχεί στον ελαχιστόρο 0100, δηλαδή τη σειρά 4 στο δεκαδικό. Τα σήματα επιλογής A0 έως A2 επιτρέπουν στα bit των στηλών 010 (δηλαδή 2 στο δεκαδικό) της σειράς 4 να φτάσουν στις εξόδους των πολυπλεκτών – αποπλεκτών (λειτουργία πολυπλέκτη). Τα bits περνούν μέσα από τους τρισταθείς buffers που “κοιτούν” αριστερά και φτάνουν στις εξόδους D0 έως D3.

Εάν συμβεί σφάλμα και σταλεί αρνητικός παλμός στους ακροδέκτες \overline{WE} και \overline{RD} ταυτόχρονα, τότε τα σήματα αλληλοακυρώνονται, αφού δίνουν από μία είσοδο 0 και στις δύο AND, δημιουργώντας σε αυτές έξοδο 0 και θέτοντας όλους τους τρισταθείς buffers σε κατάσταση υψηλής αντίστασης, ώστε να αποφευχθεί η σύγκρουση και πιθανή ζημιά πληροφορίας.



6^η Άσκηση

Το σύστημα μνήμης προς σχεδίαση συνίσταται από τα εξής ολοκληρωμένα, που βρίσκονται σε διαδοχικές θέσεις χωρίς κενά:

- ✓ ROM1: 2k x 8 bit = 2k byte
- ✓ ROM2: 4k x 8 bit = 4k byte
- ✓ SRAM1: 2k x 8 bit = 2k byte
- ✓ SRAM2: 8k x 8 bit = 8k byte

Εφόσον, στο σχεδιαζόμενο σύστημα συμμετέχει ο μE 8085, η αναπαράσταση των δεδομένων απαιτεί 8 bits, δηλαδή οι λέξεις είναι 8 bit = 1 byte η κάθε μία.

Επιπλέον, οι διευθύνσεις μνήμης αποτελούνται από το πολύ 16 bit ($A_0 - A_{15}$). Άρα:

- ✓ οι λέξεις που αποθηκεύονται στην ROM1 είναι σε πλήθος $2k = 2^{11}$ λέξεις και απαιτούνται 11 bytes ($A_0 - A_{10}$) για την διευθυνσιοδότησή τους.
- ✓ οι λέξεις που αποθηκεύονται στην ROM2 είναι σε πλήθος $4k = 2^{12}$ λέξεις και απαιτούνται 12 bytes ($A_0 - A_{11}$) για τη διευθυνσιοδότησή τους.
- ✓ οι λέξεις που αποθηκεύονται στην SRAM1 είναι σε πλήθος $2k = 2^{11}$ λέξεις και απαιτούνται 11 bytes ($A_0 - A_{10}$) για την διευθυνσιοδότησή τους.
- ✓ οι λέξεις που αποθηκεύονται στην SRAM2 είναι σε πλήθος $8k = 2^{13}$ λέξεις και απαιτούνται 13 bytes ($A_0 - A_{12}$) για την διευθυνσιοδότησή τους.

Τα παραπάνω δεδομένα είναι αρκετά για να σχεδιάσουμε τον χάρτη μνήμης του συστήματός μας.

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 - 4k
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FF	
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800	SRAM1 - 2k
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	SRAM2 - 8k
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF	

Παρατηρούμε ότι τα bit A_{15} και A_{14} δεν χρησιμοποιούνται για τον προσδιορισμό καμίας θέσης μνήμης (έχουν για όλες τις θέσεις μνήμης τιμή 00). Χρησιμοποιούνται, λοιπόν, ως επίτρεψη στον αποκωδικοποιητή (1η υλοποίηση) ή στις λογικές πύλες του δευτέρου τρόπου.

Τα bit A_{11} , A_{12} και A_{13} χρησιμοποιούνται για την επιλογή του επιθυμητού ολοκληρωμένου (ROM1, ROM2, SRAM1 ή SRAM2) καθώς ένας ή δύο συνδυασμοί αυτών προσδιορίζουν μοναδικά τις περιοχές μνήμης που αντιστοιχούν στο κάθε ολοκληρωμένο.

Συγκεκριμένα:

- ✓ $A_{13}A_{12}A_{11} = 000 \rightarrow \text{ROM1}$
- ✓ $A_{13}A_{12}A_{11} = 001$ και $A_{13}A_{12}A_{11} = 010 \rightarrow \text{ROM2}$
- ✓ $A_{13}A_{12}A_{11} = 011 \rightarrow \text{SRAM1}$
- ✓ $A_{13}A_{12}A_{11} = 100$, $A_{13}A_{12}A_{11} = 101$, $A_{13}A_{12}A_{11} = 110$, $A_{13}A_{12}A_{11} = 111 \rightarrow \text{SRAM2}$

Πριν προχωρήσουμε στον τελικό σχεδιασμό μένει να προσδιορίσουμε εξονυχιστικά τον τρόπο επιλογής ολοκληρωμένου.

Για το σκοπό αυτό, έχουμε στη διάθεσή μας τα bit A_{13} , A_{12} και A_{11} .

α) Αν κάνουμε **χρήση αποκωδικοποιητή και πυλών** τότε ακολουθούμε τις εξής σχεδιαστικές κινήσεις:

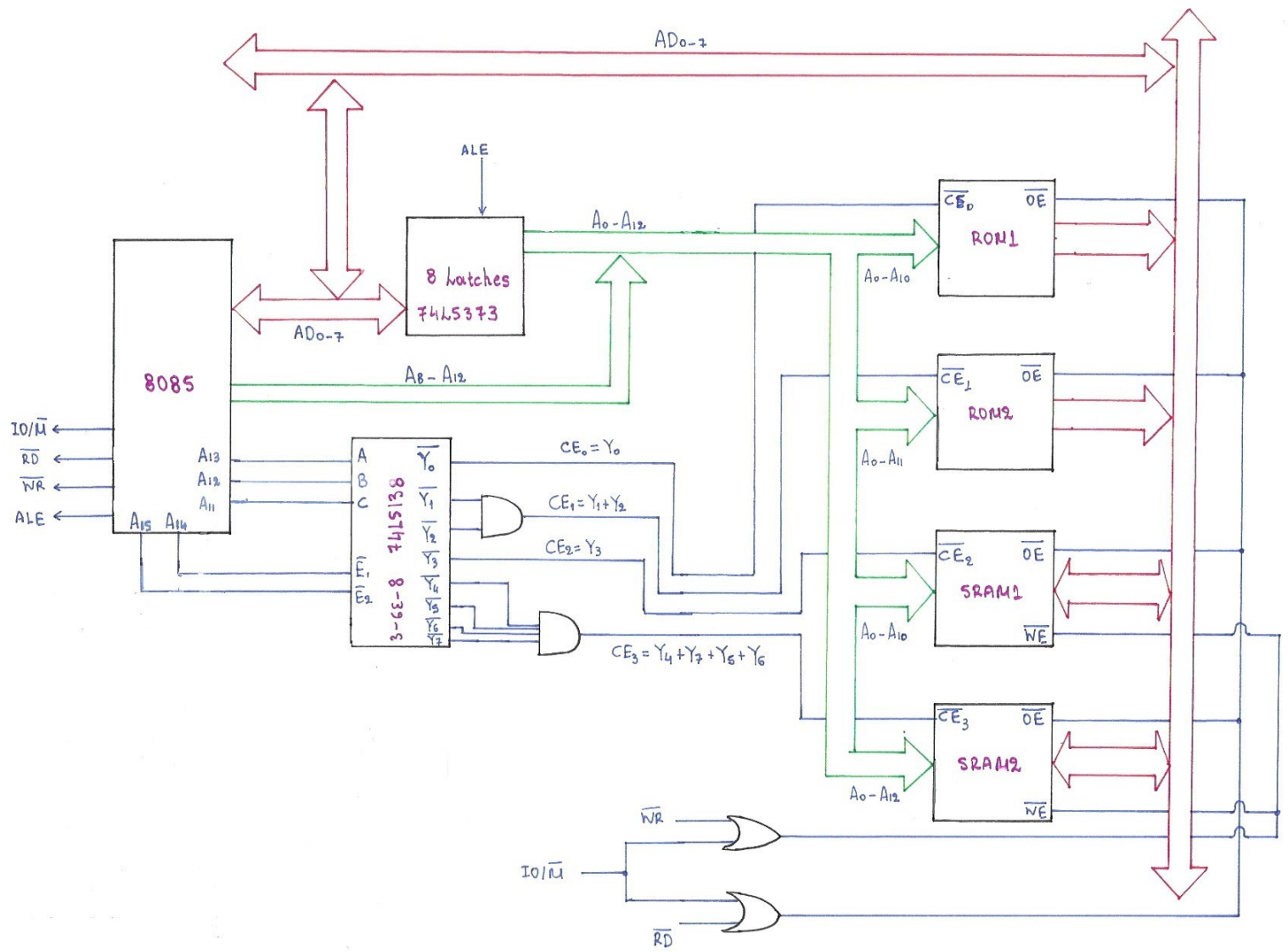
Χρησιμοποιούμε αποκωδικοποιητή 3 – σε – 8 [74LS138]. Ο αποκωδικοποιητής παίρνει ως εισόδους τα A_{13} , A_{12} , A_{11} και τα A_{15} , A_{14} ως επιτρίψεις. Έτσι, μόνο όταν $A_{15} = A_{14} = 0$ θα μπορεί μέσω του αποκωδικοποιητή να γίνει επιλογή μνήμης. Ανάλογα, με τον συνδυασμό των A_{13} , A_{12} , A_{11} όλες οι έξοδοι του αποκωδικοποιητή γίνονται 1 εκτός από αυτή που αντιστοιχεί στο συμπλήρωμα του ελαχιστόρου που καθορίζει η είσοδος. Οι έξοδοι του αποκωδικοποιητή συνδέθηκαν με τρόπο που φαίνεται στο παρακάτω σχήμα, σύμφωνα με τις παρακάτω σχέσεις:

$$CE_0 = 1 (\overline{CE}_0 = 0) \text{ όταν } A_{13}A_{12}A_{11} = 000, \text{ δηλαδή όταν } Y_0 = 1 \Rightarrow \overline{Y}_0 = 0 \\ \Rightarrow \overline{CE}_0 = \overline{Y}_0$$

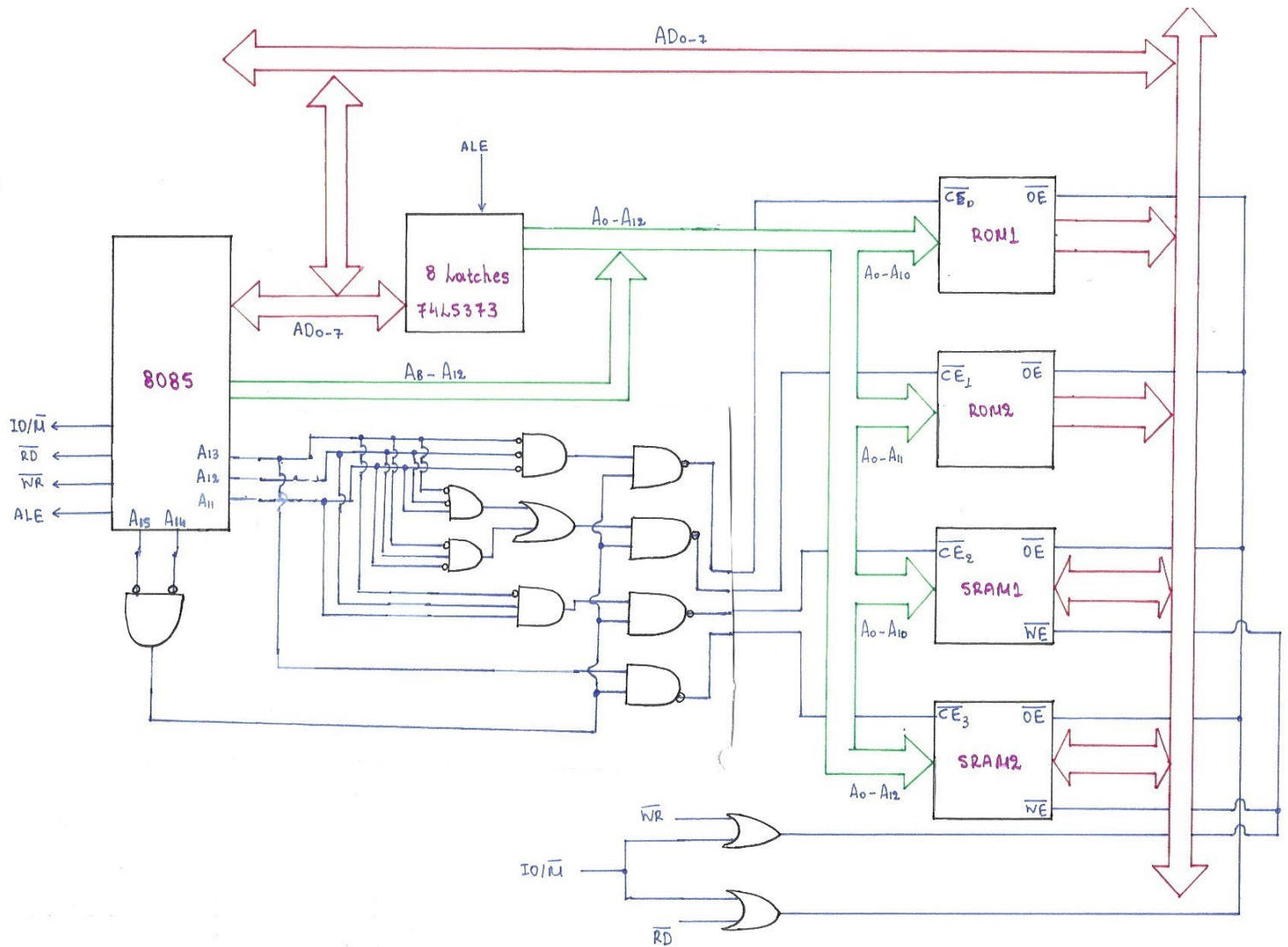
$$CE_1 = 1 (\overline{CE}_1 = 0) \text{ όταν } A_{13}A_{12}A_{11} = 001 \text{ ή } A_{13}A_{12}A_{11} = 010, \text{ δηλαδή όταν } Y_1 + Y_2 = 1 \Rightarrow \overline{Y}_1 \overline{Y}_2 = 0 \\ \Rightarrow \overline{CE}_1 = \overline{Y}_1 \overline{Y}_2$$

$$CE_2 = 1 (\overline{CE}_2 = 0) \text{ όταν } A_{13}A_{12}A_{11} = 011, \text{ δηλαδή όταν } Y_3 = 1 \Rightarrow \overline{Y}_3 = 0 \\ \Rightarrow \overline{CE}_2 = \overline{Y}_3$$

$$CE_3 = 1 (\overline{CE}_3 = 0) \text{ όταν } A_{13}A_{12}A_{11} = 100 \text{ ή } A_{13}A_{12}A_{11} = 101 \text{ ή } A_{13}A_{12}A_{11} = 110 \text{ ή } A_{13}A_{12}A_{11} = 111, \\ \text{δηλαδή όταν } Y_3 + Y_4 + Y_5 + Y_6 = 1 \Rightarrow \overline{Y}_4 \overline{Y}_5 \overline{Y}_6 \overline{Y}_7 = 0 \\ \Rightarrow \overline{CE}_3 = \overline{Y}_4 \overline{Y}_5 \overline{Y}_6 \overline{Y}_7$$



β) Αν κάνουμε **χρήση αποκλειστικά λογικών πυλών** τότε χρησιμοποιούμε κατάλληλο συνδυασμό πυλών ώστε να φτάνει αρνητικός παλμός μόνο στο chip που θέλουμε κάθε φορά να προσπελάσουμε. Επισημαίνεται ότι μόνο το bit A_{13} είναι υπεύθυνο για την επιλογή της SRAM2, καθώς μόνο αυτή μπορεί να επιλεγθεί με $A_{13} = 1$.



7^η Άσκηση

Τα διαθέσιμα ολοκληρωμένα κυκλώματα είναι:

- ✓ ROM 16kB
- ✓ RAM 4kB
- ✓ RAM 8kB

Ένας συνδυασμός που να επιτρέπει τον σχεδιασμό χάρτη μνήμης, όπως αυτός ζητείται από την εκφώνηση, είναι να χρησιμοποιήσουμε:

- 1) μία μνήμη ROM των 16kB, της οποίας τα πρώτα 4kB θα αντιστοιχιστούν στις θέσεις μνήμης 0000H έως 0FFFH και τα υπόλοιπα 12kB θα αντιστοιχιστούν στις θέσεις μνήμης 4000H έως 6FFFH,
- 2) μία μνήμη RAM των 4kB αντιστοιχιζόμενη στις θέσεις μνήμης 1000H έως 1FFFH
- 3) μία μνήμη RAM των 8kB αντιστοιχιζόμενη στις θέσεις μνήμης 2000H έως 3FFFH.

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	0000H	ROM
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	RAM1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H	RAM2
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H	ROM
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	6FFFH	

Παρατηρούμε ότι το bit A_{15} δεν χρησιμοποιείται για τον προσδιορισμό καμίας θέσης μνήμης (έχει για όλες τις θέσεις μνήμης τιμή 0). Χρησιμοποιείται, λοιπόν, ως επίτρεψη στον αποκωδικοποιητή που επιλέγει ολοκληρωμένο για προσπέλαση.

Ακόμη:

- ✓ οι λέξεις που αποθηκεύονται στην ROM είναι σε πλήθος $16k = 2^{14}$ λέξεις και απαιτούνται 14 bytes ($A_0 - A_{13}$) για την διευθυνσιοδότησή τους.

- ✓ οι λέξεις που αποθηκεύονται στην RAM1 είναι σε πλήθος $4k = 2^{12}$ λέξεις και απαιτούνται 12 bytes ($A_0 - A_{11}$) για την διευθυνσιοδότησή τους.
- ✓ οι λέξεις που αποθηκεύονται στην RAM2 είναι σε πλήθος $8k = 2^{13}$ λέξεις και απαιτούνται 13 bytes ($A_0 - A_{12}$) για την διευθυνσιοδότησή τους.

Τα bit A_{12} , A_{13} και A_{14} χρησιμοποιούνται για την επιλογή του επιθυμητού ολοκληρωμένου (ROM, RAM1 ή RAM2) καθώς ένας ή δύο συνδυασμοί αυτών προσδιορίζουν μοναδικά τις περιοχές μνήμης που αντιστοιχούν στο κάθε ολοκληρωμένο.

Συγκεκριμένα:

- ✓ $A_{14}A_{13}A_{12} = 000$ και $A_{14}A_{13}A_{12} = 100$, $A_{14}A_{13}A_{12} = 101$ και $A_{14}A_{13}A_{12} = 110 \rightarrow \text{ROM}$
- ✓ $A_{14}A_{13}A_{12} = 001 \rightarrow \text{RAM1}$
- ✓ $A_{14}A_{13}A_{12} = 010$ και $A_{14}A_{13}A_{12} = 011 \rightarrow \text{RAM2}$

Οι έξοδοι του αποκωδικοποιητή συνδέθηκαν σύμφωνα με τις παρακάτω σχέσεις. Από τον χάρτη μνήμης συμπεραίνουμε:

$CS_0 = 1$ ($\overline{CS}_0 = 0$) όταν $A_{13}A_{12}A_{11} = 000$, $A_{13}A_{12}A_{11} = 100$, $A_{13}A_{12}A_{11} = 101$, $A_{13}A_{12}A_{11} = 110$ δηλαδή όταν $Y_0 + Y_4 + Y_5 + Y_6 = 1 \Rightarrow \overline{Y}_0 \overline{Y}_4 \overline{Y}_5 \overline{Y}_6 = 0$
 $\Rightarrow \overline{CS}_0 = \overline{Y}_0 \overline{Y}_4 \overline{Y}_5 \overline{Y}_6$

$CS_1 = 1$ ($\overline{CS}_1 = 0$) όταν $A_{13}A_{12}A_{11} = 001$, δηλαδή όταν $Y_1 = 1 \Rightarrow \overline{Y}_1 = 0$
 $\Rightarrow \overline{CS}_1 = \overline{Y}_1$

$CS_2 = 1$ ($\overline{CS}_2 = 0$) όταν $A_{13}A_{12}A_{11} = 010$, $A_{13}A_{12}A_{11} = 011$ δηλαδή όταν $Y_2 + Y_3 = 1 \Rightarrow \overline{Y}_2 \overline{Y}_3 = 0$
 $\Rightarrow \overline{CS}_2 = \overline{Y}_2 \overline{Y}_3$

Για την επίτρεψη του Latch της θύρας εισόδου 7000H κάνουμε χρήση πύλης **AND** με 16 εισόδους, η οποία δίνει στην έξοδο 1 αν:

$$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0 = 0111000000000000$$

Για την επίτρεψη του Latch της θύρας εισόδου 70H κάνουμε χρήση του \overline{Y}_7 .

Η μνήμη ROM λαμβάνει τα bits A_0 έως A_{11} από το **address bus** και τα bit A_{12} και A_{13} από τις εξόδους των πυλών **XOR** που φαίνονται στο παρακάτω σχήμα. Αυτό συμβαίνει, ώστε τα bit A_{12} και A_{13} να μετατρέπουν τις διευθύνσεις του χάρτη μνήμης που αντιστοιχούν σε θέσεις της ROM που δεν είναι στο πρώτο τμήμα της (0000H – 0FFFH) σε συνεχόμενες θέσεις εσωτερικά στο ολοκληρωμένο (δηλαδή γίνεται αντιστοίχιση των διευθύνσεων 4000H έως 6FFFH του χάρτη μνήμης στις διευθύνσεις 1000H έως 3FFFH της ROM).

Παρουσιάζεται, στη συνέχεια, το ζητούμενο μΥ – Σ:

