

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

&

ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Συστήματα Μικροϋπολογιστών

*2η Σειρά Ασκήσεων*

*6Ο Εξάμηνο*

*ΡΟΗ Υ*

*Αθανασίου Νικόλαος*

*ΑΜ 03112074*

*Σταυρακάκης Δημήτριος*

*ΑΜ 03112017*

**Άσκηση 1**

IN 10H ; apenergopoihsh ths prostasias mnhmhs opws ypodykneietai

MVI A,00H ; arxikopoihsh tou kataxwrhth A (deikths gia termatismo)

MVI E,00H ; arxikopoihsh tou kataxwrhth E (ari8moi apo 0-255)

MVI B,00H ; arxikopoihsh tou kataxwrhth B gia to 2o erwthma

MVI C,00H ; arxikopoihsh tou kataxwrhth C gia to 2o erwthma

LXI H,0900H ; fortwnw th diey8ynsh 0900H ston H gia thn apo8hkeysh ari8mwn

QUESTION\_A:

MOV M,A ; vazw sthn katallhlh 8esh mnhmhs ton katallhlo ;ari8mo

MVI D,08H ; twra 8a checkarw ena ena ta bits tou ari8mou pou

; pou molis evala gia na dw posous assous exei

QUESTION\_B:

RAL

JNC NEXT\_BIT

INX B ; an vrw asso pros8etw sto 2plo kataxwrhth BC 1

NEXT\_BIT: DCR D ; meiwnw kata 1 ton counter twn bits

MOV E,A ; vazw th shiftarismenh timh tou A ston E

MOV A,D ; vazw thn timh tou D ston A gia na checkarw

; posa bits menoun

CPI 00H

MOV A,E ;epanaferw ston A th swsth timh tou(shiftarismenh)

JNZ QUESTION\_B

MOV A,M ; epanaferw ston A thn pragmatika swsth timh tou

; apo th mnhmh pou thn apo8hkeysa sthn arxh

INR A ; ay3anw ton A, otan 3anaftasei sto 0 8a exw

; apo8hkeysei olous tous epi8ymhtous ari8mous

CPI 00H ; checkarw an o A 3anaeftase sto 0

JZ QUESTION\_C ; an nai telos to 1o-2o erwthma

INX H ; deixnw sthn epomenh 8esh mnhmhs

JNZ QUESTION\_A ; an den exw ftasei to 255 synexizw th loopa mou

QUESTION\_C:

LXI H,0900H ; fortwnw th diey8ynsh tou prwtou ari8mou mou gia

; na tous checkarw olous osous evala sth mnhmh

MVI D,00H ; arxikopoiw ton D kataxwrhth

MVI E,00H ; arxikopoiw ton E kataxwrhth gia na checkarw

; plh8os epanalhpsewn

CHECK:

MOV A,M ; fernw ston A ton ka8e ari8mo

CPI 10H ;elegxw gia to katw orio pou einai to 10H

JZ ADD\_D\_1 ; an (A)=10H pros8ese 1 sto D

JC NEXT\_NUM ; an (A)<10H tote phgaine ston epomeno ari8mo

CPI 60H ; elegxw gia to anw orio pou einai to 60H

JZ ADD\_D\_1 ; an (A)=60H pros8ese 1 sto D

JNC NEXT\_NUM ; an (A)>60H phgaine ston epomeno ari8mo

ADD\_D\_1:

INR D ; alliws aykshse to D kata 1 kai

NEXT\_NUM:

INX H ; meta checkare ton epomeno

INR E ; ay3anw ton E kata 1

MOV A,E ; ton vazw ston A gia elegxo

CPI 00H ; checkarw an 3anaphge sto 0

JNZ CHECK ; an den phge des ton epomeno alliws

; synexise sto QD

QUESTION\_D:

LDA 2000H ; anagnwsh eisodou - diakoptwn

RAL

RAL

RAL

RAL

RAL ; fernw ta 3 LSB stis 3 MSB 8eseis

RAL ; fernw sto C to prwto kata proteraiothta

; elegxou bit

JC LEDS\_D

RAL ; fernw sto C to 2o kata proteraiothta elegxou bit

JC LEDS\_C

RAL ; fernw sto C to 3o kata proteraiothta elegxou bit

JC LEDS\_B

MVI A,00H ; an den exw kanena anoixto 8elw ta leds kleista

CMA

STA 3000H

JMP PROG\_END

LEDS\_D: MOV A,D ;pernaw ton D ston A

CMA ;pairnw to symplhrwma tou giati ta leds einai

;arnhtikhs logikhs

STA 3000H ;anavw ta leds

JMP PROG\_END ;loop

;antistoixa me ton D anavw ta LEDS gia tis alles periptwseis

LEDS\_C: MOV A,C

CMA

STA 3000H

JMP PROG\_END

LEDS\_B: MOV A,B

CMA

STA 3000H

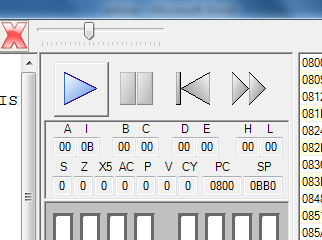
JMP PROG\_END

PROG\_END:

JMP QUESTION\_D ; gia na synexizei to checkarisma real-time

END

**Άσκηση 2**

****

**;se ayth th 8esh o epilogeas taxythtas tou programmatos ry8mizei swsta thn kathysterhsh twn 20 sec**

**;genika exoume balei ston BC thn timh 100 gia na exoume 100\*1ms=0.1 sec gia ka8e antapokrish**

**;alla xwris aythn thn epilogh den antapokrinetai se pragmatiko xrono**

;DIAVASMA EISODOU APO THN 2000H

;EKSODO ME LED STHN 3000H

;OI EVRESEIS MSB KAI LSB GINONTAI ME ENTOLES OLISTHISEWN

;DHLADH RAL KAI EPILEGETAI GIA ELEGXOS ALMATOS TA

;CARRY FLAG KAI ZERO FLAG

;XRHSIMOPOIOUNTAI OI KATAXWRHTES A,B,C,D,E

MVI B,00H ;GIA KATHISTERHSH 1MS\*100=0.1 SEC

MVI C,64H

BEGIN:

LDA 2000H ;DIAVASMA EISODOU

RAL ;ELEGXOS MSB

JC BEGIN ;AN MSB DEN EINAI 0 KSANADIAVASE

;EISODO ALLIWS SUNEXISE KATW

LED\_1ST\_OFF\_ON:

LDA 2000H ;DIAVASMA EISODOU

RAL ;DIAKOPTHS UP MSB=1

JNC LED\_1ST\_OFF\_ON ;SYNEXISE DIAVASMA MEXRI NA

;GINEI UP O DIAKOPTHS

LED\_1ST\_ON\_OFF:

LDA 2000H ;DIAVASMA EISODOU

RAL ;DIAKOPTHS DOWN MSB=1

JC LED\_1ST\_ON\_OFF ;SYNEXISE DIAVASMA MEXRI NA KSANAGINEI OFF

MSLED\_ON: ;TO LED EINAI ON OTAN O DIAKOPTHS GINEI

;DOWN->UP->DOWN

MVI D,00H ;OFF-ON-OFF DHLADH OPWS ZHTEITAI

MVI E,C8H ;E = 200 DEKADIKO GIA THN KATHUSTERHSH

MVI A,7FH ;A=01111111 GIA TO ANAMMA TOU

STA 3000H ;PRWTOU LED AFOU EINAI ARNHTIKHS LOGIKHS

DELAY:

CALL DELB ;0.1 SECOND KATHYSTERHSH

DCR E ;TOY AFAIRW ,KSEKINAEI APO TO 200

MOV A,E ;CHECK AN TELEIWSOUN TA 20 SECONDS

CPI 00H

JZ TURN\_OFF\_LED ;AN NAI KLEISE TO LED

MOV A,D ;AN OXI KATA TH DIARKEIA TWN 20 SECONDS

CPI 00H ;CHECKARW TIS TIMES TOU D GIA NA DW

;TI KATASTASH EIMAI

JZ LED\_IS\_ON ;AN D=0 CHECKARE POU EINAI TWRA

CPI 01H ;HTAN KATW DHLADH OFF?

JZ OFF\_ON ;AN D=1 CHECKARE POU EINAI TWRA

CPI 02H ;HTAN PANW AFOU HTAN KATW DHLADH?

;EXEI GINEI DLD TO OFF-ON?

JZ OFF\_ON\_OFF ;AN D=2 CHECKARE AN PREPEI NA ANANEW8EI

;DHLADH POU EINAI TWRA TO MSB

TURN\_OFF\_LED:

MVI A,FFH

STA 3000H ;KLEISIMO TOU LED

JMP BEGIN

LED\_IS\_ON:

LDA 2000H

RAL

JC DELAY ;APOFASISE AN EIMASTE SE KATASTASH OFF

INR D ;APOTHIKEUSE OFF STATE =1

OFF\_ON: ; TWRA EXW D = 1

LDA 2000H

RAL

JNC DELAY ;APOFASISE AN EIMASTE SE OFF-ON KATASTASH

INR D ;APOTHIKEUSE OFF - ON STATE = 2

OFF\_ON\_OFF: ; TWRA EXW D = 2

LDA 2000H

RAL

JC DELAY ; DES AN EIMAI SE OFF - ON - OFF STATE

REMAINS\_ON:

JMP MSLED\_ON ;ANANEWSE TO ORIO XRONOU OTAN PREPEI

END

**Άσκηση 3**

Η μνήμη είναι 256 x 4 bits. Το 4, δηλώνει σε πόσα ίσα μέρη είναι χωρισμένη η μνήμη και το 256, δηλώνει πόση χωρητικότητα μνήμης έχει το κάθε μέρος, στην περίπτωσή μας 32x8. Κάθε κομμάτι είναι σαν ένα δισδιάστατο πίνακα με 32 γραμμές και 8 στήλες.

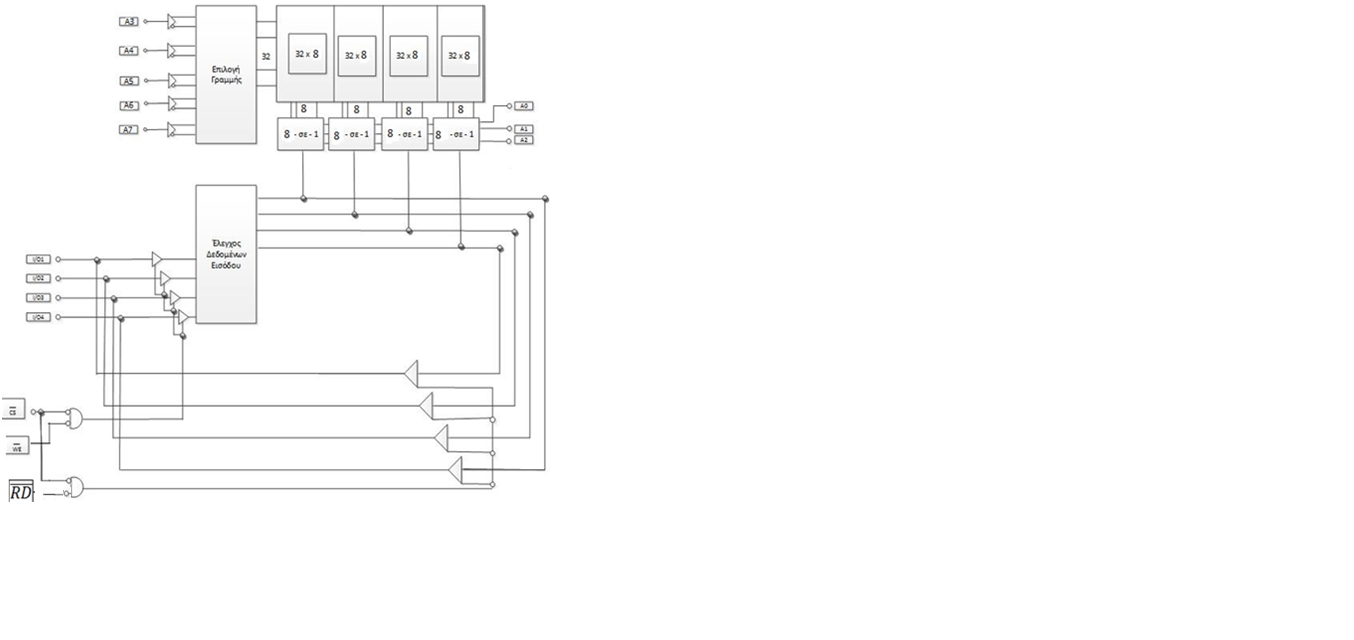
αριθμός γραμμών = αριθμός στηλών = 32

Η επιλογή αυτή επέφερε τη χρήση πολυπλεκτών 8 - σε - 1. Τα 5 Least Significant bits είναι αυτά που θα καθορίσουν τη γραμμή που επιθυμούμε να προσπελάσουμε. Επίσης τα τρία τελευταία bits (ως επιλογείς των πολυπλεκτών) χρησιμοποιούνται για τον προσδιορισμό της διεύθυνσης των στηλών κατά την προσπέλαση στη μνήμη. Τέλος, το γεγονός ότι έχουμε χωρίσει τη μνήμη σε 4 κομμάτια για να λάβουμε τα 4 bits που επιθυμούμε, επιβάλλει την ύπαρξη και 4 σημάτων Ι/Ο για κάθε μία από αυτές.

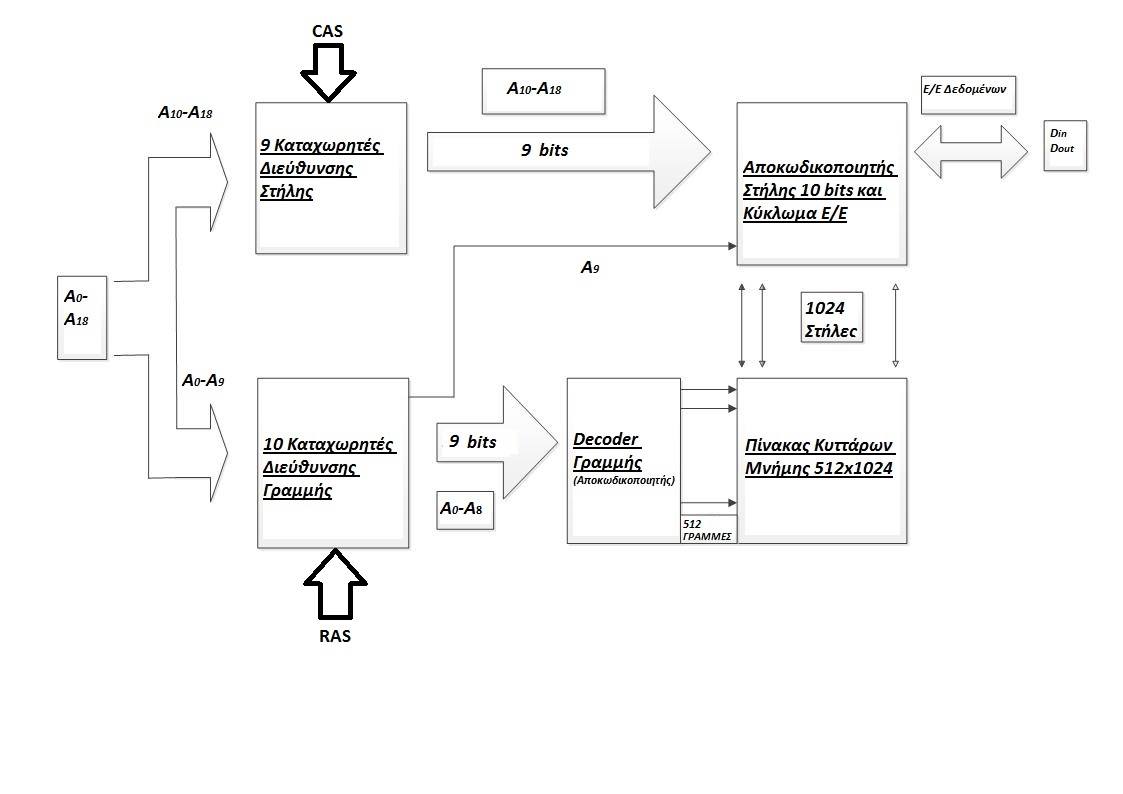
Ας υποθέσουμε για παράδειγμα ότι θέλουμε να διαβάσουμε το περιεχόμενο της μνήμης που βρίσκεται στη διεύθυνση 00010100. Έτσι η παραπάνω διεύθυνση εφαρμόζεται στις εισόδους Α0-Α7. Εφόσον πρόκειται για ανάγνωση, η είσοδος είναι 0 , η είσοδος είναι 0 και η  είναι 1. Ως αποτέλεσμα η κάτω πύλη AND έχει έξοδο 1 και η πάνω πύλη AND έχει έξοδο 0. Λόγω της πρώτης εμποδίζεται το σήμα εισόδου να καταγραφεί στη μνήμη ενώ λόγω της δεύτερης επιτρέπεται η ανάγνωση από τη μνήμη. Τα περισσότερο σημαντικά ψηφία 00010 εφαρμόζονται ως είσοδος στον επιλογέα γραμμής, ο οποίος είναι ένας αποκωδικοποιητής 5 σε 25=32. Έτσι, επιλέγεται η 2η γραμμή της μνήμης. Από αυτές, επιλέγεται μόνο το 4ο bit από κάθε ένα από τα 4 πεδία της μνήμης λόγω των ελάχιστα σημαντικών bits 100 τα οποία εισάγονται ως επιλογείς στους πολυπλέκτες. Ως αποτέλεσμα, επειδή οι 4απομονωτές έχουν επίτρεψη 1, η τετράδα που επιλέγεται περνάει στις 4 γραμμές Ι/Ο.

Ανάλογη με την παραπάνω είναι και η λειτουργία εγγραφής. Σ’αυτή την περίπτωση, η είσοδος είναι 0 και η είσοδος  είναι 0 και η είσοδος είναι 1. Ως αποτέλεσμα η πάνω πύλη AND έχει έξοδο 1 και η κάτω πύλη AND έχει έξοδο 0. Έτσι ενεργοποιούνται οι απομονωτές που επιτρέπουν τη διέλευση των γραμμών Ι/Ο ενώ απενεργοποιούνται οι απομονωτές μέσω των οποίων γίνεται διέλευση των δεδομένων της μνήμης. Η επιλογή της μνήμης με διεύθυνση 00010100 γίνεται με ακριβώς ίδιο τρόπο αλλά η μόνη διαφορά είναι ότι εδώ, αποθηκεύεται στη μνήμη μέσω των πολυπλεκτών, η πληροφορία που βρίσκεται στις εισόδους Ι/Ο.

Σύμφωνα με το βιβλίο δε θα έχουμε conflict μεταξύ και .

Το σχεδιάγραμμα της SRAM που καλούμαστε να σχεδιάσουμε είναι η ακόλουθη:

Αντίστοιχη είναι και η εσωτερική οργάνωση μίας μνήμης **DRAM 512Κ1bit**.



Η συγκεκριμένη μνήμη έχει διεύθυνση μήκους (512Κ=29\*210 =219) 19bits και είναι οργανωμένη σε λέξεις του 1bit. Επιλέγουμε τη δημιουργία του πίνακα κυττάρων μνήμης με διαστάσεις (5121024). Επομένως, για τον προσδιορισμό της διεύθυνσης γραμμής απαιτούνται τα Less Significant 9bits (29 =512 γραμμές) και για τον προσδιορισμό στήλης απαιτούνται τα 10 More Significant bits (210=1024 στήλες).

Η διαδικασία ανάγνωσης και εγγραφής περιλαμβάνει την μεταφορά της διεύθυνσης στην είσοδο. Τα ψηφία Α0­-Α9 που αποτελούν τη διεύθυνση γραμμής, κρατούνται στο ΤΜS4416 με τον παλμό  και στη συνέχεια τα ψηφία Α10-A18 κρατούνται στο ίδιο chip με τον παλμό . Η εγγραφή της μνήμης γίνεται στο αρνητικό μέτωπο του παλμού *W* κι ενώ το σήμα *G*ισούται με 1. Η ανάγνωση γίνεται στο αρνητικό μέτωπο του *G*κι ενώ το σήμα *W*ισούται με 1.

**Άσκηση 4**

Χάρτες Μνημών

Ζητούμενο της παρούσας άσκησης είναι η κατασκευή ενός συστήματος μνήμης για μΥ σύστημα του μΕ 8085. Το σύστημα αυτό θα περιέχει μνήμη EPROM 10kByte και μνήμη RAM 6kByte.

Για το σκοπό αυτό χρησιμοποιείται δύο ολοκληρωμένα κυκλώματα 4K x 8bit και ένα 2K x 8bit για την μνήμη ΕPROM και μία 4K x 8bit και ένα 2K x 8bit για την RAM. H EPROM θέλουμε να ξεκινάει από τη θέση 0000H και αμέσως μετά να βρίσκεται ο χώρος που χρησιμοποιεί η μνήμη RAM. Επομένως θα τοποθετήσουμε τα ολοκληρώμενα της EPROM πρώτα και έπειτα της RAM. Επιλέγουμε για κάθε είδος μνήμης να τοποθετήσουμε πρώτα το chip μεγαλύτερης χωρητικότητας και έπειτα το μικρότερο, χωρίς να άλλαζε η λειτουργία του συστήματος μνήμης αν γινόταν ανάποδη τοποθέτηση. Αρχικά κατασκευάζουμε το χάρτη μνήμης. Το πρώτο chip EPROM που χρησιμοποιούμε έχει διαστάσεις

4K x 8bit = 212 x 8bit, δηλαδή χρειαζόμαστε 12bits διεύθυνσης για να προσδιορίσουμε ποιο τμήμα της σελίδας της EPROM θέλουμε να προσπελάσουμε και επίσης από την έξοδο του συγκεκριμένου chip μπορούμε να παίρνουμε κάθε φορά μία πλήρη λέξη των 8bit. Ο χάρτης μνήμης για το συγκεκριμένο chip είναι:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EPROM 1 (4KX8bit) | | | | |
| Αρχή διευθύνσεων | | | | |
| BIN | 0000 | 0000 | 0000 | 0000 |
| HEX | 0 | 0 | 0 | 0 |
| Τέλος διευθύνσεων | | | | |
| BIN | 0000 | 1111 | 1111 | 1111 |
| HEX | 0 | F | F | F |

Αντίστοιχα το επόμενο chip EPROM ξεκινάει από την αμέσως επόμενη διεύθυνση μνήμης και λόγω της μίας διάστασης του (4 Κ = 212) χρειάζεται 12bit διεύθυνσης για να προσδιοριστεί το τμήμα της θέλουμε από αυτό. Έτσι, ο χάρτης μνήμης στην περίπτωση αυτή είναι:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EPROM 2 (4Kx8bit) | | | | |
| Αρχή διευθύνσεων | | | | |
| BIN | 0001 | 0000 | 0000 | 0000 |
| HEX | 1 | 0 | 0 | 0 |
| Τέλος διευθύνσεων | | | | |
| BIN | 0001 | 1111 | 1111 | 1111 |
| HEX | 1 | F | F | F |

Στη συνέχεια ακολουθεί το επόμενο chip της μνήμης ΕPRΟM. Αυτό το chip της EPROM επειδή θα έχουν μία διάσταση 2 K = 211, χρειάζονται 11 bits για την παράσταση μιας διεύθυνσης τους. Οι χάρτες μνήμης για το τρίτο chip της EPROM είναι ο ακόλουθος:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EPROM 3 (2Kx8bit) | | | | |
| Αρχή διευθύνσεων | | | | |
| BIN | 0010 | 0000 | 0000 | 0000 |
| HEX | 2 | 0 | 0 | 0 |
| Τέλος διευθύνσεων | | | | |
| BIN | 0010 | 0111 | 1111 | 1111 |
| HEX | 2 | 7 | F | F |

Στη συνέχεια ακολουθούν τα chip της μνήμης RAM. Το πρώτο chip της RAM (SRAM) θα έχει μία διάσταση 4 K = 212, χρειάζονται 12 bits για την παράσταση μιας διεύθυνσης τους, ενώ το δεύτερο (2 Κ = 211 ) 11bits. Οι χάρτες μνήμης για τα δύο chip της RAM είναι:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SRAM 1 (4KX8bit) | | | | |
| Αρχή διευθύνσεων | | | | |
| BIN | 0010 | 1000 | 0000 | 0000 |
| HEX | 2 | 8 | 0 | 0 |
| Τέλος διευθύνσεων | | | | |
| BIN | 0011 | 0111 | 1111 | 1111 |
| HEX | 3 | 7 | F | F |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SRAM 2 (2KX8bit) | | | | |
| Αρχή διευθύνσεων | | | | |
| BIN | 0011 | 1000 | 0000 | 0000 |
| HEX | 3 | 8 | 0 | 0 |
| Τέλος διευθύνσεων | | | | |
| BIN | 0011 | 1111 | 1111 | 1111 |
| HEX | 3 | F | F | F |

Από όλα τα παραπάνω προκύπτει ότι ο συνολικός χάρτης μνήμης θα είναι:

2x EPROMs 4K x 8 bits

1x EPROM 2K x 8 bits

1x SRAM 4K x 8

1x SRAM 2K x 8

2800H

3FFFH

4000H

27FFH

SRAMs 6Kbytes

EPROMs  
10Kbytes

ΑΧΡΗΣΙΜΟΠΟΙΗΤΗ ΜΝΗΜΗ

0000Η

FFFFΗ

Στη συνέχεια στόχος είναι να εντοπίσουμε ένα μοτίβο με το οποίο θα επιλέγεται το κάθε chip μνήμης (πότε δηλαδή η υποδοχή CS’ του κάθε chip θα πρέπει να γίνεται 0). Για το σκοπό αυτό τοποθετούμε σε ένα πίνακα την αρχική και την τελική διεύθυνση κάθε chip. Έτσι έχουμε:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EPROM 1 | 0000 | 0000 | 0000 | 0000 |
| 0000 | 1111 | 1111 | 1111 |
| EPROM 2 | 0001 | 0000 | 0000 | 0000 |
| 0001 | 1111 | 1111 | 1111 |
| EPROM 3 | 0010 | 0000 | 0000 | 0000 |
| 0010 | 0111 | 1111 | 1111 |
| SRAM 1 | 0010 | 1000 | 0000 | 0000 |
| 0011 | 0111 | 1111 | 1111 |
| SRAM 2 | 0011 | 1000 | 0000 | 0000 |
| 0011 | 1111 | 1111 | 1111 |

Παρατηρώντας τον παραπάνω πίνακα συμπεραίνουμε ότι τα 5 πιο σημαντικά bit είναι ικανά να μας προσδιορίσουν ποιο είναι το επιθυμητό chip. Επίσης βλέπουμε ότι είτε στην αρχική είτε στην τελική διεύθυνση του κάθε chip τα ψηφία Α15-Α14 (πράσινα ψηφία) είναι 0. Άρα διαφοροποίηση υπάρχει μόνο στα ψηφία Α13-Α12-Α11 (κόκκινα ψηφία). Έτσι, για την επιλογή του chip θα χρησιμοποιήσουμε τον αποκωδικοποιητή 3 σε 8 **74LS138** ο οποίος έχει 3 εισόδους ενεργοποίησης (Ε1-Ε3) και 3 εισόδους που δίνουν τις 8 εξόδους. Στις δύο εισόδους ενεργοποίησης θα συνδεθούν τα bit Α15-Α14 αφού περάσουν από μία πύλη αντιστροφέα, ώστε η μνήμη να μην είναι προσπελάσιμη στην περίπτωση διευθύνσεων με πιο σημαντικά ψηφία τα 11 ή 10 ή 01, καθώς τότε η επίτρεψη του αποκωδικοποιητή είναι στο λογικό 0 λόγω των αντιστροφέων και με βάση το datasheet του ολοκληρωμένου **74LS138**. (οι επιτρέψεις του ολοκληρωμένου στην ουσία είναι 2: Ε3 και Ε1-2 =Ε1 ΟR E2 ) Για να επιλέξουμε ποια έξοδος του αποκωδικοποιητή ενεργοποιεί ποιο chip κατασκευάζουμε τον ακόλουθο πίνακα :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Α13 | Α12 | Α11 | EPROM1 | EPROM2 | EPROM3 | SRAM1 | SRAM2 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

Όταν δύο τριάδες διευθύνσεων ενεργοποιούν το ίδιο chip τότε εισάγονται σε μία AND, της οποίας η έξοδος οδηγείται στο επιθυμητό chip. Αυτό συμβαίνει επειδή ο αποκωδικοποιητής έχει συμπληρωματικές εξόδους σε δεδομένη είσοδο και στο CS’ πρέπει να εισαχθεί 0 για να ενεργοποιηθεί. Επίσης, σύμφωνα με τον πίνακα με τις αρχικές και τελικές διευθύνσεις, πρέπει να κυκλοφορούν στο ADDRESS BUS τα bits A0-A13 για να μπορούν να προσδιοριστούν πλήρως όλες οι λέξεις που υπάρχουν στην μνήμη (αυτό συμβαίνει επειδή στην περίπτωση της SRAM1 παρατηρούμε διαφοροποίηση της αρχικής και τελικής διεύθυνσης στα 14 λιγότερο σημαντικά bit). Έτσι η διάταξη έχει ως ακολούθως:

