

Συστήματα Μικροπολογιστών

2^η Ομάδα Ασκήσεων

Μοίρας Αλέξανδρος Α.Μ.: el18081

Παπαδημητρίου Ευθύμιος Α.Μ.: el18129

Ασκήσεις Προσομοίωσης

1^η Άσκηση:

```
; Erwthma a
IN 10H ; Apenergopoihsh prostasias mnhmhs
MVI A,00H ; Prwtos arithmos pou apothikeuoume
LXI H,0900H ; prwth thesi mnhmhs sthn opoia apothikeuoume
LOOP1: MOV M,A ; Apothikeusi arithmou
INX H ; Epomenh thesi mnhmhs
INR A ; Epomenos ari8mos
JNZ LOOP1 ; Elegxos an apothikeuthkan oloi oi arithmoi mexri to FF
```

```
; Erwthma B
LXI H,0900H ; Arxikh thesi mnhmhs apo opou tha fortwthei o arithmos
LXI B,0000H ; Arxikopoihsh metrhth asswn
LOAD_NUMBERS: MOV A,M ; Fortwsh arithmwn
MVI D,08H ; 8 peristrofes gia na elegxoume ola ta pshfia
ROTATE: RAL ; Peristrofh
JNC CONTINUE ; An einai 0 mhn auxhseis ton metrhth asswn
INX B ; Alliws auxhsh metrhth
CONTINUE: DCR D ; Meiwsh metrhth peristrofwn
JNZ ROTATE ; An den exoun ginei 8 epanalave
INX H ; Epomenh 8esh mnhmhs
MOV A,L ; Elegxos an exoume elegxei mexri kai ton arithmo sth
CPI 00H ; thesh 09FF
JNZ LOAD_NUMBERS ; An oxi epanalave
```

```
; Erwthma C
LXI H,0900H ; Thesh prwtou arithmou ston HL
MVI D,00H ; Arxikopoihsh metrhth
COUNTER: MOV A,M ; Fortwsh arithmou
CPI 10H ; An o arithmos einai mikroteros tou 10 proxwra
JC SKIP ; ston epomeno
CPI 60H ; An o arithmos einai isos me 60 auxhse alliws an h
JZ INCREASE ; sygkrish epistrepsei kratoumeno 0 tha einai
JNC SKIP ; ypoxrewtika megalyteros kai proxwrame ston epomeno
INCREASE: INR D ; An einai 10<=arithmos<=60 auxhse metrhth
SKIP: INX H ; epomenh thesh mnhmhs ara epomenos arithmos
MOV A,L ; Elegxos an exoume elegxei mexri kai ton arithmo sth
CPI 00H ; thesh 09FF
JNZ COUNTER ; An oxi epanalave
```

END

2^η Άσκηση:

```
MVI D,00H ; Flag se poia katastash einai o diakopths
MVI E,00H ; "Xronometro" oso den einai 0 anave ta fwta
LXI B,0064H ; 64H = 100 dekadiko gia na exoume kathysterhsh 0.1s
                ; gia na exoume thn epithymhth diakritikh ikanothta
MAIN: MOV A,D ; Fortwsh tou flag
CPI 00H ; flag = 0 shmainei oti einai sto off o diakopths
JZ CHECK1 ; An einai sto off elegxe an egine on
CPI 01H ; An einai sto 1 elegxe an xanagine off
JZ CHECK2

CONTINUE: MOV A,E ; Routina xeirismou fwtwn
CPI 00H ; An to "xronometro" einai 0 svhse ta fwta
JZ TURNOFF
MVI A,00H ; Alliws anapse ta
STA 3000H
CALL DELB ; Perimene 0.1sec
DCR E ; Meiwse to xronometro kata 1
JMP MAIN ; Gyrna sth routina elegxou tou MSB

CHECK1: LDA 2000H ; Fortwsh eisodou
RAL ; Elegxos an to MSB einai 1
JNC CONTINUE ; An oxi phgaine sth routina pou xeirizetai ta fwta
INR D ; An nai kane to flag 1 kai meta phgaine sthn routina
JMP CONTINUE ; tw n fwtwn

CHECK2: LDA 2000H ; Fortwsh eisodou
RAL ; Elegxos an to MSB xanaegine 0
JC CONTINUE ; An oxi phgaine sth routina pou xeirizetai ta fwta
MVI E,C8H ; Vale to "xronometro" sto 200 wste ta fwta na meinoun
MVI D,00H ; anamena gia 200*0.1=20sec kai epanafora tou flag sto 0
JMP CONTINUE ; Ystera phgaine sth routina pou xeirizetai ta fwta

TURNOFF: MVI A,FFH ; Svhsimo fwtwn
STA 3000H
JMP MAIN ; Epistrofh sth routina elegxou tou MSB

END
```

3^η Άσκηση:

i)

```
MAIN: LDA 2000H ; Fortwsh eisodou
MVI C,00H ; Thesh tou prwtou assou arxika ypothetoumai sto bit 0
MOV B,A ; Apothikeush eisodou ston B
ROTATES: MOV A,B ; Epanafora eisodou pou exei peristrafei 0-7 fores
RAR ; peristrofh eisodou gia elegxo tou trexontos LSB
JC FOUND ; An einai 1 vrhkame ton prwto asso
INR C ; An oxi auxhsh ths pithanhs theshs tou prwtou assou
MOV B,A ; Proswrinh apothikeush peristrammenhs eisodou
MOV A,C ; Elegxos an elegxthikan kai ta 8 bits opote den yparxei
CPI 08H ; assos sthn eisodo. An oxi elegxe to epomeno bit
JNZ ROTATES ; epistrefwntas sth ROTATES

NO_ONES: MVI A,FFH ; An den yparxei assos svhse ola ta LED
STA 3000H
JMP MAIN

FOUND: MVI E,01H ; An vrethike o C periexei th thesh sthn opoia
LOOP1: MOV A,C ; vrethike o prwtos assos kai vasei autou
CPI 00H ; peristrefoyme ton E wste o monadikos assos tou
JZ EXIT ; na paei sthn thesh tou prwtou assou ths eisodou
DCR C
MOV A,E
RAL
MOV E,A
JMP LOOP1

EXIT: MOV A,E ; Anapse to LED ths theshs tou prwtou assou
CMA
STA 3000H
JMP MAIN

END
```

ii)

```
READIN: CALL KIND ; Diavasma plhktrologiou
CPI 00H ; Elegxos an auto pou diavasame einai metaxy tou 0
JZ READIN ; kai tou 8 alliws xanadiavase
CPI 09H
JNC READIN

MOV B,A ; Apothikeush arithmou ston B
MVI A,00H ; Arxika ypothetoume ola ta LED anammena
LOOP1: DCR B ; Meiwsh arithmou kata 1
MOV C,A ; Apothikeush LED ston C
MOV A,B ; Elegxos an o arithmos mhdenisthke ara ta anammena LED
CPI 00H ; tha einai stis swstes theseis kai metavainoume sth
JZ END ; rutina ektypwshs
MOV B,A ; Alliws apothikeysh tou meiwmenou arithmou ston B
MOV A,C ; kai svhsimo kai tou epomenou LED
RAL
INR A
JMP LOOP1 ; Kai epanalave ton elegxo th diadikasia

END: MOV A,C ; Fortwsh svhstwn LED
STA 3000H ; Kai ektypwsh tous
JMP READIN ; Epistrofh sthn anagnwsh plhktrologiou

END
```

iii)

```
IN 10H ; Apenergopoishsh prostasias mnhmhs
MVI A,00H ; Arxikopoihsh sto 0 thesewn mnhmhs pou tha
STA 0A00H ; xrhsimopoihthoun gia thn ektypwsh sto 7-segment
STA 0A01H ; display
STA 0A02H
STA 0A03H
STA 0A04H
STA 0A05H

CHECK0: MVI A,FEH ; Elegxos grammhs 0 apothikeuontas sth thesh
STA 2800H ; 2800H to 11111110 dyadiko
LDA 1800H ; diavase sthles twv plhktrwn
ANI 07H ; Maska gia na apomonwsoume ta 3 LSB
CPI 07H ; An kai ta 3 einai 1 ara den paththike kanena plhktro
JZ CHECK1 ; ths grammhs elegxe thn epomenh grammh

CASE01: CPI 06H ; Elegxe an paththike to prwto plhktro ths grammhs
JNZ CASE02 ; An oxi elegxe to an paththike to epomeno plhktro
MVI A,86H ; Alliws apothikeuse ston A ton kwdiko tou plhktrou
JMP DISP_OUTPUT ; pou paththike kai phgaine sth rutina exodou

CASE02: CPI 05H ; Elegxe an paththike to deuthero plhktro ths
JNZ CHECK1 ; Mporei na paraleifthei o elegxos gt an ftasoume edw
MVI A,85H ; sigoura exei paththei to deuthero plhktro
JMP DISP_OUTPUT
```

```
CHECK1: MVI A,FDH ; Elegxos grammhs 1
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK2

CASE11: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
; ths
JNZ CASE12
MVI A,84H
JMP DISP_OUTPUT

CASE12: CPI 05H
JNZ CASE13
MVI A,80H
JMP DISP_OUTPUT

CASE13: CPI 03H
JNZ CHECK2
MVI A,82H
JMP DISP_OUTPUT

CHECK2: MVI A,FBH ; Elegxos grammhs 2
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK3

CASE21: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
; ths
JNZ CASE22
MVI A,00H
JMP DISP_OUTPUT

CASE22: CPI 05H
JNZ CASE23
MVI A,83H
JMP DISP_OUTPUT
```

```
CASE23: CPI 03H
JNZ CHECK3
MVI A,81H
JMP DISP_OUTPUT
```

```
CHECK3: MVI A,F7H ; Elegxos grammhs 3
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK4
```

```
CASE31: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
          ; ths
JNZ CASE32
MVI A,01H
JMP DISP_OUTPUT
```

```
CASE32: CPI 05H
JNZ CASE33
MVI A,02H
JMP DISP_OUTPUT
```

```
CASE33: CPI 03H
JNZ CHECK4
MVI A,03H
JMP DISP_OUTPUT
```

```
CHECK4: MVI A,EFH ; Elegxos grammhs 4
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK5
```

```
CASE41: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
          ; ths
JNZ CASE42
MVI A,04H
JMP DISP_OUTPUT

CASE42: CPI 05H
JNZ CASE43
MVI A,05H
JMP DISP_OUTPUT

CASE43: CPI 03H
JNZ CHECK5
MVI A,06H
JMP DISP_OUTPUT


CHECK5: MVI A,DFH ; Elegxos grammhs 5
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK6


CASE51: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
          ; ths
JNZ CASE52
MVI A,07H
JMP DISP_OUTPUT

CASE52: CPI 05H
JNZ CASE53
MVI A,08H
JMP DISP_OUTPUT

CASE53: CPI 03H
JNZ CHECK6
MVI A,09H
JMP DISP_OUTPUT
```

```

CHECK6: MVI A,BFH ; Elegxos grammhs 6
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ CHECK7

CASE61: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
          ; ths
JNZ CASE62
MVI A,0AH
JMP DISP_OUTPUT

CASE62: CPI 05H
JNZ CASE63
MVI A,0BH
JMP DISP_OUTPUT

CASE63: CPI 03H
JNZ CHECK7
MVI A,0CH
JMP DISP_OUTPUT


CHECK7: MVI A,7FH ; Elegxos grammhs 7
STA 2800H
LDA 1800H
ANI 07H
CPI 07H
JZ DISP_LAST ; An den paththike kanena plhktro kalese thn
          ; DISP_LAST pou xanaektypwnei to teleutaio plhktro
          ; pou paththike

CASE71: CPI 06H ; Elegxos an paththike kapoio apo ta tria plhktra
          ; ths
JNZ CASE72
MVI A,0DH
JMP DISP_OUTPUT

```



```

CASE72: CPI 05H
JNZ CASE73
MVI A,0EH
JMP DISP_OUTPUT

CASE73: CPI 03H
JNZ DISP_LAST
MVI A,0FH
JMP DISP_OUTPUT

; Routina exodou
DISP_OUTPUT: MOV B,A ; Apothikeush kwdikou plhktrou pou paththike
               ; ston B
MVI A,00H ; Mhdenismos tw n prwtwn 4 7-segment displays
STA 0A00H
STA 0A01H
STA 0A02H
STA 0A03H
MOV A,B ; Epanafora plhktrou pou path8hke ston A
ANI 0FH ; Apomonwsh tou prwtou dekaexadikou pshfiou tou me maska
STA 0A04H ; kai apothikeush tou sth thesi 0A04 wste na paei sto 5o
               ; 7-segment display
MOV A,B ; Epanafora plhktrou pou path8hke ston A
RAR ; Metafora most significant dekaexadikou pshfiou sth thesi tou
RAR ; least significant
RAR
RAR
ANI 0FH ; Apomonwsh tou me maska
STA 0A05H ; kai apothikeush tou sth thesi 0A05 wste na paei sto 6o
               ; 7-segment display
LXI D,0A00H ; Apothikeusi sto zeugos DE ths theshs apo thn opoia
CALL STDM ; arxizei to mhnyma kai klhsh ths STDM
CALL DCD ; Klhsh ths DCD gia ektypwsh sta 7 segment displays
JMP CHECK0 ; Epanalave thn anagnwsh tou plhktrologiou

DISP_LAST: LXI D,0A00H ; Ektypwsh teleutaiou plhktrou pou
CALL STDM ; paththike
CALL DCD
JMP CHECK0 ; Epanalave thn anagnwsh tou plhktrologiou

END

```

4^η Άσκηση:

```
IN 10H ; Apenergopoihsh prostasias mnhmhs
START: MVI D,00H ; Tha xrhsimopihthei gia na kataskeuastei h exodos
LDA 2000H ; Fortwsh bits eisodou
MOV B,A ; kai apothikeush tous ston B

ANI 01H ; Apomonwsh B0
MOV C,A
MOV A,B
RAR
ANI 01H ; Metakinshsh A0 sto LSB kai apomonwsh tou
XRA C ; Apothikeush apotelesmatos B0 XOR A0 sth thesi 0A00H
STA 0A00H

MOV A,B
RAR
RAR
ANI 01H ; Metakinshsh B1 sto LSB kai apomonwsh tou
MOV C,A
MOV A,B
RAR
RAR
RAR
ANI 01H ; Metakinshsh A1 sto LSB kai apomonwsh tou
XRA C ; Apothikeush apotelesmatos B1 XOR A1 sth thesi 0A01H
STA 0A01H

MOV A,B
RAR
RAR
RAR
RAR
ANI 01H ; Metakinshsh B2 sto LSB kai apomonwsh tou
MOV C,A
MOV A,B
RAR
RAR
RAR
RAR
ANI 01H ; Metakinshsh A2 sto LSB kai apomonwsh tou
ANA C ; Apothikeush apotelesmatos B2 AND A2 sth thesi 0A02H
```

```
STA 0A02H

MOV A,B
RAR
RAR
RAR
RAR
RAR
RAR
ANI 01H ; Metakinshsh B3 sto LSB kai apomonwsh tou
MOV C,A
MOV A,B
RAR
RAR
RAR
RAR
RAR
RAR
RAR
ANI 01H ; Metakinshsh A3 sto LSB kai apomonwsh tou
ANA C ; Apothikeush apotelesmatos B3 AND A3 sth thesi 0A03H
STA 0A03H

LDA 0A00H ; Ylopoihs ths XOR pou paragei to X0
MOV C,A
LDA 0A01H
XRA C
STA 0A04H

LDA 0A02H ; Ylopoihs ths OR pou paragei to X2
MOV C,A
LDA 0A03H
ORA C
STA 0A05H
```

```

LDA 0A04H ; Fortwsh X0
MOV D,A   ; kai apothikeush tou ston D
LDA 0A01H ; Fortwsh X1
RAL       ; Metakinshsh tou sto bit 1
ANI 02H   ; Maska gia na exaleipsoume tyxon kratoumena apo to RAL
ORA D     ; Topothethshsh tou sto bit 1 tou apotelesmatos
MOV D,A   ; Apothikeusi trexontos apotelesmatos ston D
LDA 0A05H ; Antistoixa fortwsh tou X2 kai topothethshsh tou sto bit
RAL       ; 2 tou apotelesmatos
RAL
ANI 04H
ORA D
MOV D,A
LDA 0A03H ; Fortwsh tou X3 kai metafora tou sto bit 3
RAL
RAL
RAL
ANI 08H   ; Apomonwsh mono tou bit 3;
ORA D     ; Kataskeuh telikou apotelesmatos

CMA       ; Ektypwsh apotelesmatos sta LED
STA 3000H
JMP START ; Epanalave th diadikasia

END

```

Θεωρητικές Ασκήσεις

5^η Άσκηση:

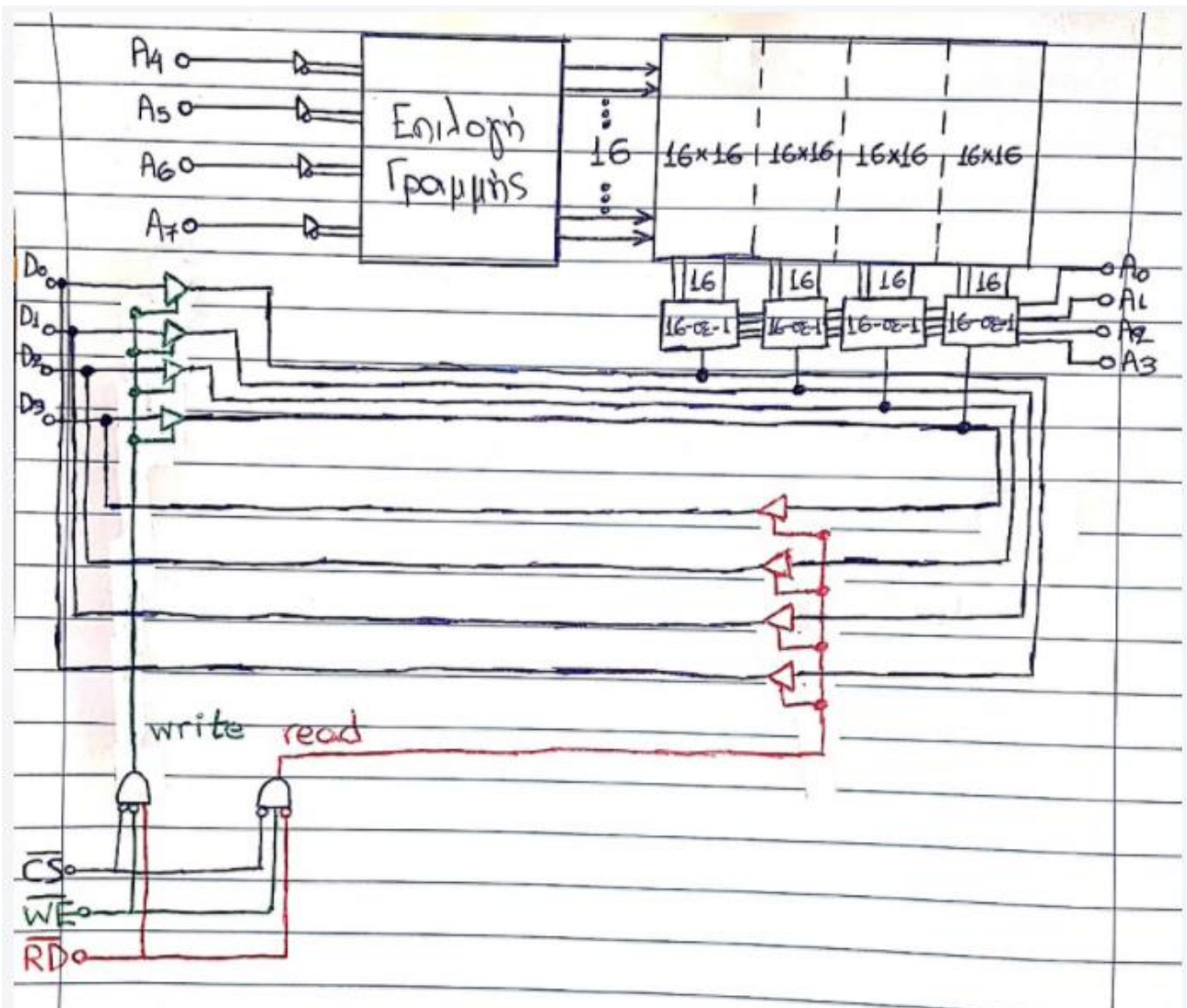
Η διάταξη της μνήμης θα αποτελείται από 16×64 στοιχεία μνήμης. Ο πίνακας της μνήμης θα έχει 16 γραμμές κάθε μία από τις οποίες θα χωρίζεται σε 4 τμήματα με κάθε τμήμα να έχει 16 στήλες. Η διεύθυνση θα αποτελείται από τα bits A0-A7 εκ των οποίων τα 4 υψηλότερης αξίας Bit A4-A7 θα επιλέγουν μία από τις 16 γραμμές του πίνακα της μνήμης μας μέσω κυκλώματος αποκωδικοποίησης και τα bits A0-A3 θα ελέγχουν 4 πολυπλέκτες 16 σε 1 (διπλής κατεύθυνσης) μέσω των οποίων επιλέγεται ποια 4αδα bits της επιλεγμένης γραμμής θα συνδεθεί στη γραμμή των δεδομένων.

Το chip της μνήμης επιλέγεται με το σήμα \overline{CS} και έπειτα με το σήμα \overline{WE} επιλέγεται η λειτουργία εγγραφής, δηλαδή όταν αυτό το σήμα τίθεται στο λογικό 0 τα δεδομένα D0-D3 οδηγούνται μέσω των ενεργοποιημένων buffers εισόδου (πράσινοι στο σχήμα) στους πολυπλέκτες 16 σε 1 και αποθηκεύονται στη μνήμη.

Κατά αντιστοιχία το σήμα \overline{RD} επιλέγει λειτουργία ανάγνωσης ενεργοποιώντας τους κόκκινους buffers και οδηγώντας τα δεδομένα από τη μνήμη προς τις εξόδους D0-D3.

Προφανώς δεν είναι επιθυμητό να είναι ταυτόχρονα στο λογικό 0 και το \overline{WE} και το \overline{RD} οπότε αν από λάθος συμβεί κάτι τέτοιο το κύκλωμα απενεργοποιεί όλους τους buffers που οδηγούν από και προς τη μνήμη.

Παρακάτω φαίνεται σε σχήμα η εσωτερική οργάνωση αυτής της μνήμης.



Αν πχ θέλαμε να διαβάσουμε τα 4 bits από τη διεύθυνση μνήμης 00010000 θα βάζαμε αυτή τη διεύθυνση στα bits A7-A0, το \overline{CS} και το \overline{RD} στο λογικό 0 και το \overline{WE} στο λογικό 1 οπότε ενεργοποιείται το σήμα read και ενεργοποιούνται οι 4 κόκκινοι buffers που οδηγούν από την μνήμη στην έξοδο (και απενεργοποιείται το

write άρα απενεργοποιούνται οι 4 πράσινοι buffers), επιλέγεται η δεύτερη γραμμή του πίνακα (ενεργοποιείται η γραμμή υπ' αριθμόν 1 του επιλογέα γραμμών) και η 1^η τετράδα αυτής της γραμμής οδηγείται στην έξοδο (αφού ενεργοποιείται η γραμμή 0 του κάθε πολυπλέκτη 16 σε 1).

Εντελώς αντίστοιχα αν θέλαμε να γράψουμε τα bits D3-D0 στη θέση μνήμης 00010000 θα θέταμε το \overline{CS} και το \overline{WE} στο λογικό 0 και το \overline{RD} στο λογικό 1 ώστε να ενεργοποιηθεί το σήμα write και μαζί με αυτό οι πράσινοι buffers που οδηγούν την είσοδο στους πολυπλέκτες 16 σε 1 και απενεργοποιείται το σήμα read. Στα Bits της διεύθυνσης θα βάλουμε το 00010000 και θα επιλεγεί η γραμμή και οι 4 στήλες ακριβώς όπως πριν ώστε η 4αδα της εισόδου μέσω των πολυπλεκτών να αποθηκευτεί στις κατάλληλες θέσεις.

6^η Άσκηση:

Ο χάρτης μνήμης αυτού του συστήματος:

Chip	Διεύθυνση	Bits Διεύθυνσης															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM1 2K	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	07FF	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
ROM2 2K	0800	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0FFF	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
ROM3 4K	1000	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	1FFF	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM1 2K	2000	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	27FF	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
RAM2 2K	2800	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	2FFF	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

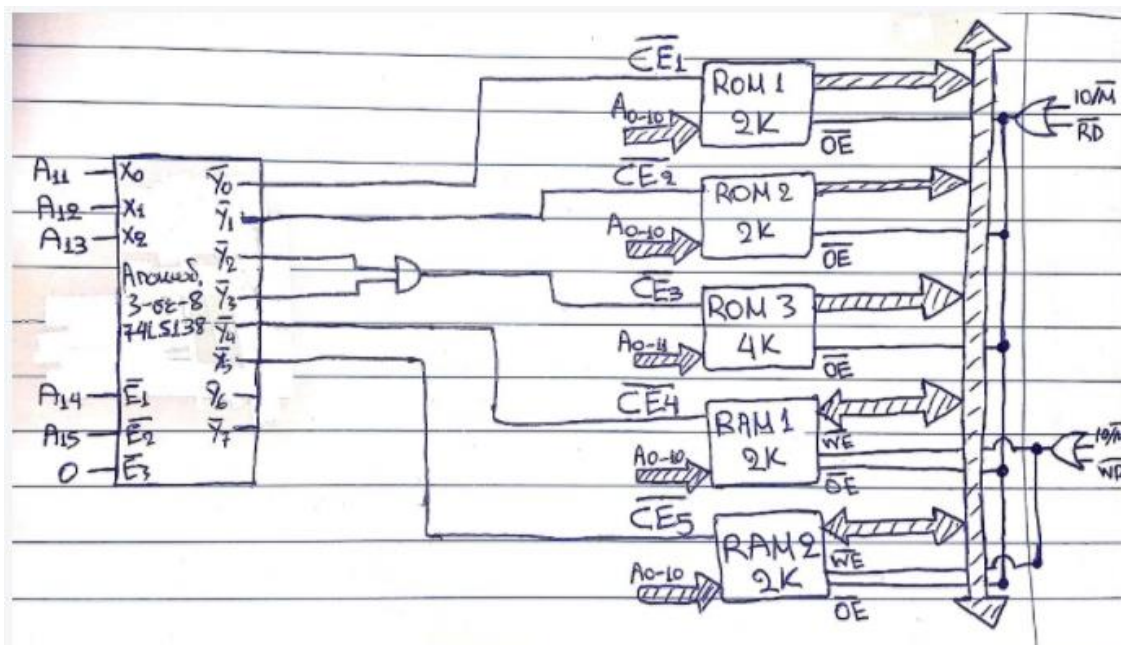
Από τον παραπάνω πίνακα παρατηρούμε ότι τα bits A11A12A13 της διεύθυνσης μπορούν να καθορίσουν ποιο chip μνήμης χρησιμοποιείται κάθε φορά. Όταν αυτή η τριάδα θα έχει την τιμή 000 θα χρησιμοποιείται η ROM1 των 2K, όταν θα έχει την τιμή 001 η ROM2 πάλι των 2K, όταν θα έχει μία από τις τιμές 010 και 011 η ROM3 των 4K, όταν θα έχει την τιμή 100 η RAM1 των 2K και τέλος όταν θα έχει την τιμή 101 η RAM2 των 2K. Τα bits A15,A14 είναι 0 όποιο από τα 5 IC και να χρησιμοποιούμε. Οι μνήμες μεγέθους 2K θα δέχονται πάνω στο chip διευθύνσεις A0-A10 ενώ η ROM των 4K θα δέχεται A0-A11. Στα bits διευθύνσεων A0-A10

όλων των IC μνημών θα οδηγούνται τα bits A0-A10 των διευθύνσεων του μΕ και στο bit A11 της ROM3 θα οδηγείται το bit A11 του μΕ.

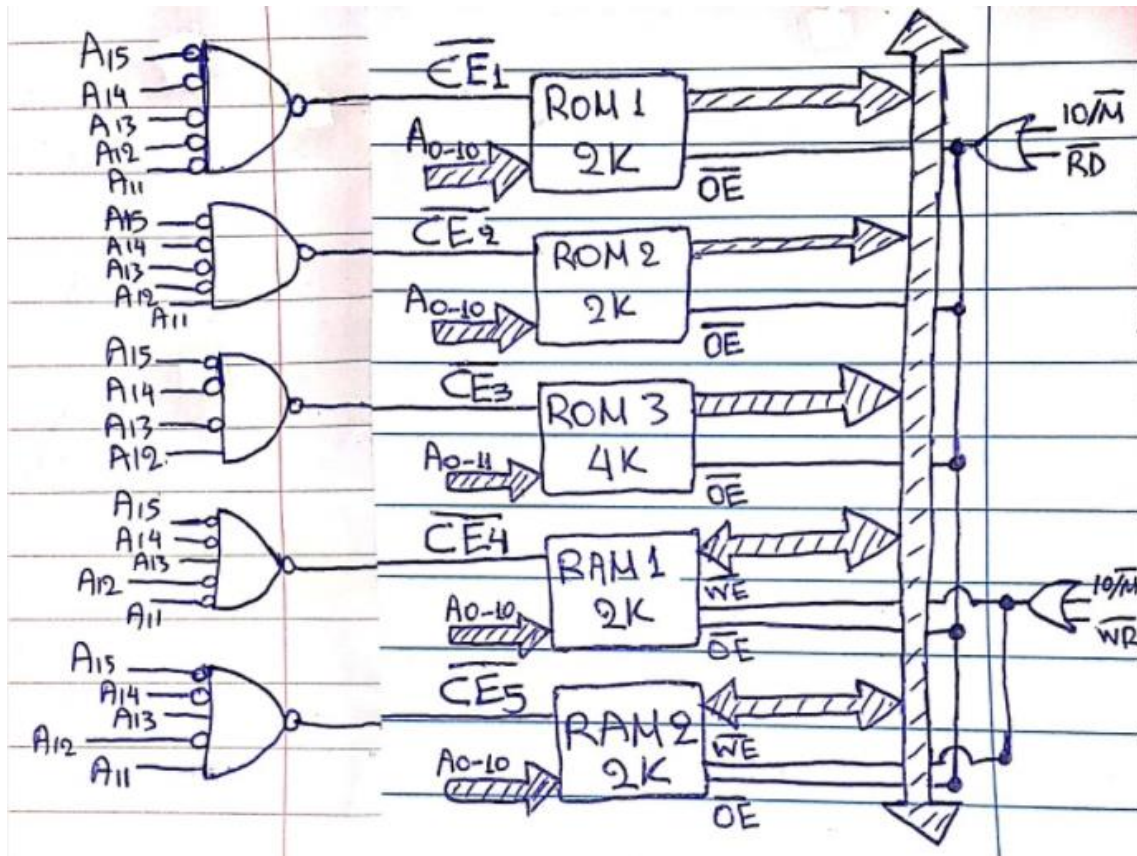
Αν η αποκωδικοποίηση γίνει με αποκωδικοποιητή τότε στις δύο από τις τρεις εισόδους αντίστροφης λογικής Enable θα βάλουμε τα A15, A14 που θέλουμε να είναι πάντα 0 όταν είμαστε στις διευθύνσεις από 0000-2FFF και στην Τρίτη το λογικό 0. Στην είσοδο του αποκωδικοποιητή θα βάλουμε τα A13A12A11 και με βάση τα παραπάνω θα συνδέσουμε την έξοδο $\overline{Y_0}$ του στο \overline{CE} της ROM1, την $\overline{Y_1}$ στο \overline{CE} της ROM2, τις $\overline{Y_2}$ και $\overline{Y_3}$ μέσω μιας πύλης AND στο \overline{CE} της ROM3, την $\overline{Y_4}$ στο \overline{CE} της RAM1 και την $\overline{Y_5}$ στο \overline{CE} της ROM5 ώστε κάθε φορά να φτάνει λογικό 0 στο \overline{CE} μόνο ενός IC.

Αν χρησιμοποιηθούν πύλες, με βάση την παραπάνω παρατήρηση για τα bits A15-A11, χρησιμοποιούμε κατάλληλες πύλες για να φτάνει λογικό 0 στο \overline{CE} μόνο ενός IC κάθε φορά.

α) Χρησιμοποιώντας αποκωδικοποιητή 3:8 (74LS138) και λογικές πύλες:



β) Χρησιμοποιώντας μόνο λογικές πύλες:



7^η Άσκηση:

Ο χάρτης της μνήμης:

[illegible]

Από τον χάρτη παρατηρούμε ότι η επιλογή των chip θα γίνεται με βάση τα bit A14-A12. Το chip της ROM πρέπει να είναι επιλεγμένο όταν το A14A13A12 έχουν μία από τις τιμές 000 001 010 ή 110. Οδηγούμε τις αντίστοιχες εξόδους αρνητικής λογικής του πολυπλέκτη στο \overline{CE} της ROM. Ομοίως στο \overline{CE} της 1^{ης} RAM οδηγούμε την έξοδο του πολυπλέκτη που αντιστοιχεί στο 011, στη 2^η RAM αυτή που αντιστοιχεί στο 100 και στην 3^η αυτή που αντιστοιχεί στο 101. Οι RAM θα έχουν 12 bit διευθύνσεων αφού έχουν μέγεθος 2K ενώ η ROM των 16K θα έχει 14 bit διευθύνσεων. Στις διευθύνσεις A11-A0 όλων των chip θα οδηγούμε τα bits διεύθυνσης του μΕ. Όσον αφορά την μνήμη ROM των 16K που διαιρείται σε ένα τμήμα των 12K και ένα των 4K πρέπει να μεριμνήσουμε για το τι θα οδηγηθεί στα A13 και A12 του chip. Από τον χάρτη της μνήμης παρατηρούμε ότι τα bits A13 και A12 του μΕ είναι συνεπή με τα αντίστοιχα bits του chip στο διάστημα διευθύνσεων 0000-2FFF. Ωστόσο στο διάστημα διευθύνσεων 6000-6FFF του μΕ πρέπει να αντιστοιχίσουμε το διάστημα 3000-3FFF (A13A12=11) των εσωτερικών διευθύνσεων του chip. Για να έχουμε λοιπόν σωστή αντιστοίχιση μεταξύ διευθύνσεων μΕ και εσωτερικών διευθύνσεων του chip οδηγούμε στο bit A13 του chip το bit A13 του μΕ και στο bit A12 του chip το bit A14 του μΕ (προκύπτουν με παρατήρηση του χάρτη μνήμης).

Για να ενεργοποιήσουμε το output του latch της memory mapped εξόδου κάνουμε πλήρη αποκωδικοποίηση της διεύθυνσης 7000H και ταυτόχρονα απαιτούμε το σήμα IO/\overline{M} να βρίσκεται στο λογικό 0 δηλαδή να έχουμε λειτουργία μνήμης καθώς και το σήμα \overline{WR} να είναι στο 0 για να έχουμε εγγραφή.

Αντίστοιχα για να επιτρέψουμε στον buffer της εισόδου να μεταφέρει τα δεδομένα του στο data bus απαιτούμε η διεύθυνση να είναι 70H (την παίρνουμε από τον αποκωδικοποιητή), το σήμα IO/\overline{M} να βρίσκεται στο λογικό 1 δηλαδή να έχουμε λειτουργία εισόδου/εξόδου και το σήμα \overline{RD} να είναι στο 0 για να έχουμε ανάγνωση.

Τα τελικά κυκλώματα διασύνδεσης:

