



Συστήματα Μικροπολογιστών

2^η Ομάδα Ασκήσεων

Μετζάκης Ιωάννης

A.M.: 03116202

ΣΗΜΜΥ 8^ο

1η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας με σχόλια:

```
IN 10H
LXI H,0900H
MVI A,FFH

MVI C,00H
MVI D,00H
MVI E,00H

LOOP1:
    MOV M,A
    MVI B,08H      ;gia ta mhdenika

ERWTIMA2:
    RAR
    JC ASSOS       ;einai assos
    INX D           ;an 0, auksise ton diplo katax. DE
ASSOS:
    DCR B
    JNZ ERWTIMA2

    MOV A,M        ;ksana pairnoume ton A
    CPI 20H
    JC NOTGOOD
    CPI 71H
    JNC NOTGOOD    ;elegxoume an anhkei sto diasthma [20H,70h]
    INR C           ;kai analoga auksanoume h oxi ton C
NOTGOOD:
    MOV A,M        ;ksana pairnoume ton A,kai elegxoume an
    CPI 00H        ;ftasame sto 0, opou kai oloklirwnoume th lounpa
    JZ ENDLOOP
    DCR A          ;alliws proxwrame sthn epomenh thesh mnhmhs
    INX H          ;kai meiwvoume to A kata 1
    JMP LOOP1

ENDLOOP:

    MOV A,C        ;opws perimename einai 81d (31H) ta stoixeia x opou 32d <= x <= 112
    ;MOV A,D       ;kai afairwntas ta sxolio, vlepoume oti einai 1024 ta mhdenika
    ;MOV A,E       ;dhladh o D=04H kai o E=00H
    CMA
    STA 3000H

END
```

Όπως σημειώνω και στο πρόγραμμα, η αναμενόμενη τιμή των ερωτημάτων (β) και (γ) είναι γνωστά, και συγκεκριμένα ίσα με 1024d και 81d αντίστοιχα. Έχω επιλέξει να προβάλω το αποτέλεσμα του (γ), ωστόσο μπορούμε πολύ εύκολα να δούμε το αποτέλεσμα του (β) αφαιρώντας τα σχόλια στο τελευταίο κομμάτι του κώδικα.

2η ΑΣΚΗΣΗ

Παρακάτω ο κώδικας με σχόλια:

```
IN 10H                ;gia thn prostasia mnhmhs
LXI B,00C8H           ;theloume 200ms, ara 200d -> 00C8H

OFF0:
    LDA 2000H          ;sto OFF0 perimenoume na ginei ON
    ANI 01H
    CPI 01H
    JNZ OFF0

ON0:                   ;sto ON0 exei ginei ON, kai perimenoume na ksanaginei OFF
    LDA 2000H
    ANI 01H
    CPI 00H
    JNZ ON0

START_LEDS:           ;edw exei olokhrwthei to zitoumeno motivo, opote anavoume
    MVI A,00H          ;ta LEDS, opou exoun arnhtikh logikh
    MOV L,A

COUNTER:
    MVI D,00H          ;o D einai o metriths pou afksanoume gia to poses fores
COUNT_OFF:           ;tha kanoume th loupa
    MOV A,L
    STA 3000H
    CMA
    MOV L,A           ;kai oso o LSB diakoptis einai OFF, ekteloume th loupa mesa
    CALL DELB          ;se auto to block, kalwntas thn DELB gia thn kathisterish
    INR D
    LDA 2000H
    ANI 01H
    CPI 01H
    JZ COUNT_ON
    MOV A,D
    CPI 4BH           ;75decimal -> mexri ekei pame giati exoume 200msec h DELB,
    JZ END_LEDS        ;ara 15sec/0.2sec => 75 fores th DELB
    JMP COUNT_OFF

COUNT_ON:            ;edw exei ginei ON o LSB, h diadikasia sinexizetai kanonika,
    MOV A,L           ;wstoso twra o elegxos ginetai gia ton an ksanaginei OFF
    STA 3000H
    CMA
    MOV L,A
    CALL DELB
    INR D
    LDA 2000H
    ANI 01H
    CPI 00H
    JZ COUNTER        ;an exei ksanaginei OFF, pame sto COUNTER, opou ousiastika
    MOV A,D           ;arxikopoioume ton metrith D sto 0, wste na ananewthei kai
    CPI 4BH           ;na ksekinhsei apo to 0 h olh diadikasia
    JZ END_LEDS
    JMP COUNT_ON

END_LEDS:             ;edw erxomaste otan apla perasei o xronos xwris ananewsh,
    MVI A,FFH         ;wste na mhdenisoume thn eksodo kai na ksanapame sthn arxh
    STA 3000H
    JMP OFF0

END
```

Όπως εξηγώ και στα σχόλια, οι 75 επαναλήψεις προκύπτουν από το 15sec/0.2sec που είναι ο συνολικός επιθυμητός χρόνος, προς τον χρόνο που διαρκεί η κάθε καθυστέρηση.

3η ΑΣΚΗΣΗ

i) Παρακάτω ο κώδικας με σχόλια:

```
ZEROS:                ;svinoume ta LEDs
    MVI A,FFH
    STA 3000H
START:                ;diavazoume eisodo kai thn apothikevoume ston B
    LDA 2000H
    MOV B,A
    MVI D,80H         ;ousiastika thetουμε ton D iso me 80H, kai
    MVI E,08H         ;kanουμε 8 loupes opou se kathe loupa kanουμε
LOOP1:                ;ena shift aristera ton B pou exei ton A (diladi
    MOV A,B            ;to input mas), kai ama vroume l ws kratoumeno, tote
    RAL                ;vrikame to aristerotero ON diakopti kai printarουμε
    JC LEDSON          ;ton kataxwrith D
    MOV B,A
    DCR E              ;alliws, kanουμε ena shift deksia ton D, prosthetουμε
    MOV A,E            ;to 80H, kai proxwrame sth defterh fora ths loupas ktl
    CPI 00H
    JZ ZEROS           ;ama teleiwsei h loupa(8 fores) kai de vrei asso ws
    MOV A,D            ;kratoumeno tou RAL, tote einai oloi oi diakoptes
    RRC                ;kleistoi, opote pame sto ZEROS kai ksana ap htn arxh
    ADI 80H
    MOV D,A
    JMP LOOP1

LEDSON:
    MOV A,D
    CMA
    STA 3000H
    JMP START
END
```

Εδώ η λογική είναι ότι έχουμε έναν καταχωρητή D, με άσσο στο MSB του, όπου όσο ελέγχουμε τα bits της εισόδου από το MSB στο LSB και δε βρίσκουμε άσσο, τον κάνουμε shift (τον άσσο του D) και προθέτουμε άλλον ένα άσσο στο MSB του D. Όταν βρούμε άσσο στον καταχωρητή A, τότε τυπώνουμε τον D.

ii) Παρακάτω ο κώδικας με σχόλια:

```
IN 10H                ;prostasia mhnms
LXI B,03E8H          ;katallhlos xronos kathisterishs
START:
    CALL KIND         ;diavazoume apo pliktrologio kai an to input
    CPI 09H           ;einai megalitero h iso tou 9 tote ksana START
    JNC START
    MVI D,04H
    CPI 05H           ;alliws, analoga me to an einai 0<x<5 h 4<x<9
    JC MIKRO          ;phgaine sto MIKRO h sto MEGALO
MEGALO:
    MVI A,0FH         ;se aftes tis apla anavoume ta swsta LEDs me ton
    STA 3000H         ;gnwsto tropo, kaloume kathisterish, kai ta svinoume
    CALL DELB
    MVI A,FFH         ;kai olo afto to anapse svise, to kanoume 4 fores
    STA 3000H         ;elegxontas ton arxikopoihmeno apo prin kataxwrhth D
    CALL DELB
    DCR D
    MOV A,D
    CPI 00H
    JZ START
    JMP MEGALO

MIKRO:
    MVI A,FOH         ;edw ta idia alla gia ta alla 4 LEDs
    STA 3000H
    CALL DELB
    MVI A,FFH
    STA 3000H
    CALL DELB
    DCR D
    MOV A,D
    CPI 00H
    JZ START
    JMP MIKRO

END
```

Έχουμε επιλέξει μία αρκετά μεγάλη καθυστέρηση ώστε να είναι εμφανείς οι φορές που αναβοσβήνουν τα leds.

iii) Παρακάτω ο κώδικας με σχόλια:

```
IN 10H
START:
LXI D,0860H      ;redw arxikopoioyme tis 4 LSB theseis tou 7segment
                 ;sto 'keno', afou theloume na emfanisoume stis 2
                 ;MSB theseis mono

                 MVI A,10H
                 STA 0860H
                 STA 0861H
                 STA 0862H
                 STA 0863H

STILI0:
                 MVI A,FEH      ;stn arxh kathe antistoixou block, vle poume oti
                 STA 2800H      ;xrhsimopoioyme to xarakthrstiko grammhs gia na
                 LDA 1800H      ;elegksoume mia triada plhktwn
                 MVI B,07H
                 ANA B          ;ama to B einai 7, den exei patithe kanena apo
                 CPI 07H        ;afth th grammh, opote gia na mhn kanoume elegxous
                 JZ STILI1      ;xwris logo, proxwrame katetheian sthn epomenh

                 MVI C,86H      ;INCR STIP
                 CPI 06H
                 JZ DISPLAY
                 MVI C,85H      ;FETCH PC
                 CPI 05H
                 JZ DISPLAY
                 JMP START

STILI1:          ;h idia logikh pantou
                 MVI A,FDH
                 STA 2800H
                 LDA 1800H
                 MVI B,07H
                 ANA B
                 CPI 07H
                 JZ STILI2

                 MVI C,84H      ;RUN
                 CPI 06H
                 JZ DISPLAY
                 MVI C,80H      ;FETCH REG
                 CPI 05H
                 JZ DISPLAY
                 MVI C,82H      ;FETCH ADRS
                 CPI 03H
                 JZ DISPLAY

STILI2:
                 MVI A,FBH
                 STA 2800H
                 LDA 1800H
                 MVI B,07H
                 ANA B
                 CPI 07H
                 JZ STILI3

                 MVI C,00H      ;0
                 CPI 06H
                 JZ DISPLAY
                 MVI C,83H      ;STORE/INCR
                 CPI 05H
                 JZ DISPLAY
                 MVI C,81H      ;DECR
                 CPI 03H
                 JZ DISPLAY

STILI3:
                 MVI A,F7H
                 STA 2800H
                 LDA 1800H
                 MVI B,07H
                 ANA B
                 CPI 07H
                 JZ STILI4

                 MVI C,01H      ;1
                 CPI 06H
                 JZ DISPLAY
                 MVI C,02H      ;2
                 CPI 05H
                 JZ DISPLAY
                 MVI C,03H      ;3
                 CPI 03H
                 JZ DISPLAY

STILI4:
                 MVI A,EFH
                 STA 2800H
                 LDA 1800H
                 MVI B,07H
                 ANA B
                 CPI 07H
                 JZ STILI5
```

```

MVI C,04H      ;4
CPI 06H
JZ DISPLAY
MVI C,05H      ;5
CPI 05H
JZ DISPLAY
MVI C,06H      ;6
CPI 03H
JZ DISPLAY

STILI5:
MVI A,DFH
STA 2800H
LDA 1800H
MVI B,07H
ANA B
CPI 07H
JZ STILI6

MVI C,07H      ;7
CPI 06H
JZ DISPLAY
MVI C,08H      ;8
CPI 05H
JZ DISPLAY
MVI C,09H      ;9
CPI 03H
JZ DISPLAY

STILI6:
MVI A,BFH
STA 2800H
LDA 1800H
MVI B,07H
ANA B
CPI 07H
JZ STILI7

MVI C,0AH      ;A
CPI 06H
JZ DISPLAY
MVI C,0BH      ;B
CPI 05H
JZ DISPLAY
MVI C,0CH      ;C
CPI 03H
JZ DISPLAY

STILI7:
MVI A,7FH
STA 2800H
LDA 1800H
MVI B,07H
ANA B
CPI 07H
JZ START

MVI C,0DH      ;D
CPI 06H
JZ DISPLAY
MVI C,0EH      ;E
CPI 05H
JZ DISPLAY
MVI C,0FH      ;F
CPI 03H
JZ DISPLAY

DISPLAY:      ;edw provaloume sto 7-segment
MOV A,C
ANI 0FH      ;o C exei ton kwdiko kathe plhktrou
STA 0864H    ;kratame ta 4LSBits tou kai vazoume ton
MOV A,C      ;arithmo pou prokiptei sto 2o MSB tou 7seg
RRC
RRC          ;kai kratame ta 4MSBits, ta kanoume 4 RRC
RRC          ;kai antistoixa me prin, vazoume ton arithmo
RRC          ;sto MSB tou 7seg
ANI 0FH
STA 0865H

CALL STDm    ;kaloume tis STDm kai DCD gia na provlithei
CALL DCD
JMP START

END

```

Εδώ απλά ελέγχουμε κάθε γραμμή ξεχωριστά, και αντιστοιχούμε στον σωστό κωδικό το κάθε πλήκτρο. Όταν γίνει η αντιστοίχιση τυπώνουμε στο 7-segment τον κωδικό αυτό, στα 2 MSBits του, στη ρουτίνα DISPLAY.

4η ΑΣΚΗΣΗ

Στον παρακάτω κώδικα, σε κάθε block με label της μορφής IN_X_Y διαβάζουμε τις τιμές των bits εισόδων στις θέσεις X,Y και κάνουμε τους ελέγχους που αντιστοιχούν στο λογικό διάγραμμα της εκφώνησης.

Στα blocks με label BIT0,BIT1,BIT2,BIT3 μπαίνουμε όταν το αντίστοιχο bit της εξόδου (όπως προκύπτει από τις πύλες) γίνει 1. Αλλιώς, το προσπερνάμε, αφού το έχουμε αρχικοποιήσει ήδη σε 0, και διαβάζουμε τα επόμενα 2 bits εισόδου.

Τέλος, επειδή η έξοδος της πύλης OR με εισόδους τα bits εισόδου 4 και 5 είναι είσοδος της πύλης XOR με εισόδους τα 2 MSBits της εισόδου, αποθηκεύουμε το αποτέλεσμα της OR, και ο έλεγχος για το 4^ο bit της εξόδου μας(μαζί με το διάβασμα των bits 6,7 εισόδου) γίνεται στα blocks με label BIT3, και BIT3_ZERO

Παρακάτω ο κώδικας με σχόλια:

```
START:
    LDA 2000H
    MVI C,00H
    MOV B,A

    MVI D,01H
    ANI 03H
    JNZ BIT0
    JMP IN_2_3

BIT0:
    MOV C,D

IN_2_3:
    MOV A,B
    MVI D,02H
    ANI 0CH
    CPI 0CH
    JZ BIT1
    JMP IN_4_5

BIT1:
    MOV A,C
    ADD D
    MOV C,A
```



```

IN_4_5:
    MOV A,B
    MVI D,04H
    ANI 30H
    JNZ BIT2
    JMP BIT3_ZERO

BIT2:
    MOV A,C
    ADD D
    MOV C,A

BIT3:
    MOV A,B           ;edw mpainoume an h eksodos ths pylhs OR prin,
    ANI C0H           ;einai 1
    CPI C0H           ;pairnoume thn AND tw'n bits 6,7
    JZ DISPLAY        ;1 h and, 1 kai h or, 0 h xor
    MVI D,08H         ;alliws h xor einai 1
    MOV A,C
    ADD D
    MOV C,A
    JMP DISPLAY

BIT3_ZERO:           ;edw mpainoume an h OR tw'n bits eisodou 4,5 einai 0
    MOV A,B
    ANI C0H
    CPI C0H
    JNZ DISPLAY      ;0 oi or/and, ara 0 kai h xor
    MVI D,08H
    MOV A,C
    ADD D
    MOV C,A

DISPLAY:
    MOV A,C
    CMA
    STA 3000H
    JMP START

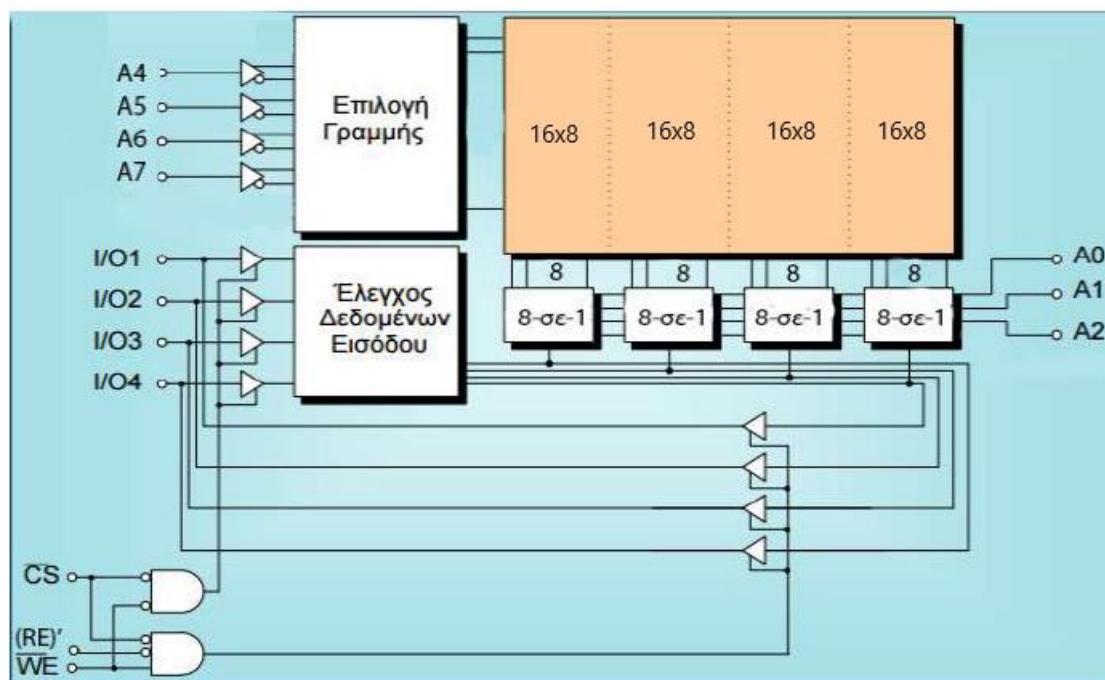
END

```

Όλοι οι παραπάνω κώδικες για τις ασκήσεις προσομοίωσης 1, 2, 3i, 3ii, 3iii και 4, βρίσκονται στο αρχείο που παραδίδω μαζί με την αναφορά.

5η ΑΣΚΗΣΗ

Παρακάτω βρίσκεται το διάγραμμα μίας εσωτερικής οργάνωσης μίας μνήμης **SRAM 128x4 bit**:



Η μνήμη είναι χωρισμένη σε 4 μέρη, όπως υποδηλώνει ο 2^{ος} αριθμός της μνήμης 128x4, με χωρητικότητα 128 bits σε κάθε μέρος. Η κατανομή σε γραμμές και στήλες μπορεί να γίνει με αρκετούς τρόπους (σταθερά όμως το μέγεθος του κάθε πίνακα πρέπει να ισούται με 128). Εδώ, η επιλογή είναι 16 γραμμές επί 8 στήλες. Συνεπώς, χρειαζόμαστε 4 bits για τον υπολογισμό της διεύθυνσης των γραμμών, και 3 bits για τη διεύθυνση των στηλών. Η υλοποίηση έγινε με χρήση πολυπλεκτών 8 σε 1, ενώ υπάρχουν ακόμα και 4 σήματα I/O, ένα για κάθε μέρος της μνήμης. Ανάγνωση γίνεται όταν και τα τρία σήματα RD, WE, CS έχουν τιμή ίση με '1', ενώ εγγραφή γίνεται όταν τα τρία παραπάνω σήματα έχουν τιμή ίση με '0'.

6η ΑΣΚΗΣΗ

Έχουμε τα παρακάτω chips:

6Kbytes ROM: 2Kx8bit , 4Kx8bit

10Kbytes RAM: 2Kx8bit , 8Kx8bit

Άρα, για την παραπάνω σειρά, και για αρχική διεύθυνση της μνήμης ROM την 0000H, έχουμε τις εξής διευθύνσεις:

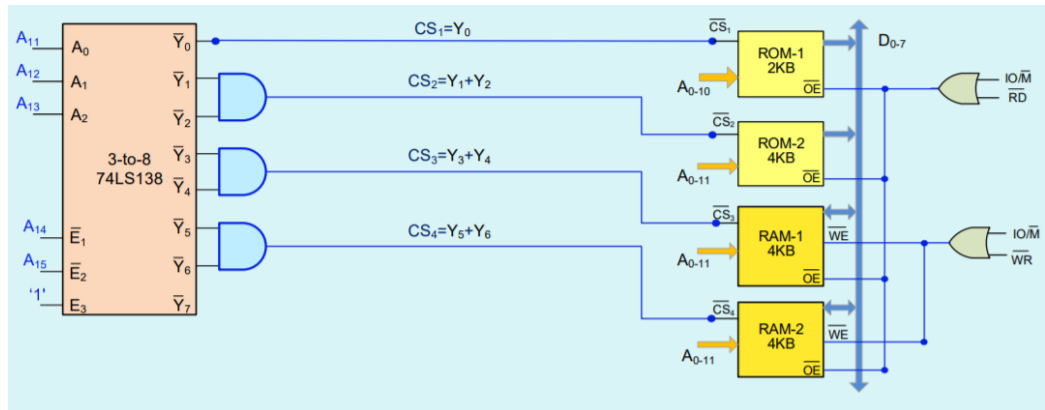
Memory	Chips	Start Address	Finish Address
6Kbytes ROM	2Kx8bit	0000H	07FFH
	4Kx8bit	0800H	17FFH
10Kbytes RAM	2Kx8bit	1800H	1FFFH
	8Kx8bit	2000H	3FFFH

Οπότε και ο χάρτης μνήμης που προκύπτει είναι ο παρακάτω:

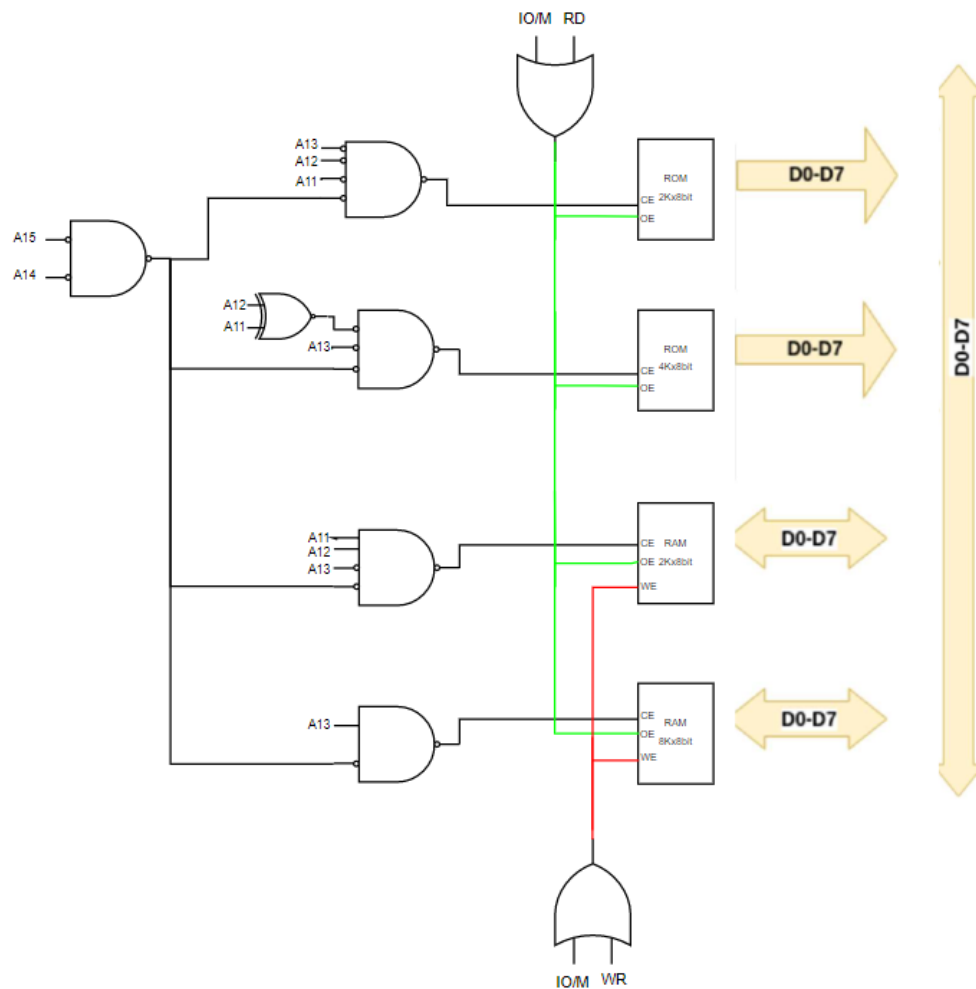
Address (Hex)	Address (Binary)
0000	0000 0000 0000 0000
07FF	0000 0111 1111 1111
0800	0000 1000 0000 0000
17FF	0001 0111 1111 1111
1800	0001 1000 0000 0000
1FFF	0001 1111 1111 1111
2000	0010 0000 0000 0000
3FFF	0011 1111 1111 1111

Συμπεραίνουμε δηλαδή, πως για τον προδιορισμό των chips που αντιστοιχεί κάθε μνήμη αρκούν τα 6 MSBits κάθε διεύθυνσης.

α) Αποκωδικοποίηση με χρήση αποκωδικοποιητή 3:8 (74LS138) και λογικές πύλες:



β) Αποκωδικοποίηση με χρήση μόνο λογικών πυλών:



7η ΑΣΚΗΣΗ

Έχουμε τα παρακάτω chips:

16Kbytes ROM

4Kbytes RAM

8Kbytes RAM

Ο χάρτης μνήμης είναι ο παρακάτω:

Address (Hex)	Address (Binary)	Chips
0000	0000 0000 0000 0000	ROM shared 8Kbytes
0FFF	0000 1111 1111 1111	
1000	0001 0000 0000 0000	RAM 4Kbytes
1FFF	0001 1111 1111 1111	
2000	0010 0000 0000 0000	RAM 8Kbytes
3FFF	0011 1111 1111 1111	
4000	0100 0000 0000 0000	ROM shared 8Kbytes
6FFF	0110 1111 1111 1111	
7000	0111 0000 0000 0000	Memory map Output

Χρειαζόμαστε τα 5 MSBits για τον προσδιορισμό κάθε ολοκληρωμένου, με το κύκλωμα διασύνδεσης να έχει ως εξής:

