

2η Σειρά Ασκήσεων

Συστήματα Μικροϋπολογιστών

Παναγιώτης Μπέλσης, AM: el20874

Ανδρόνικος Γιαννουσιάδης, AM: el20241

1η Ασκηση

α)

```
START:  
    IN 10H  
    MVI A,00H      ;Ο A krataei thn timh pou theloume na apothikeusoume  
    LXI H,0900H    ;HL dieuthunsh  
    MOV M,A  
  
STORE:  
    INR A          ;aukshsh A kai HL kata 1  
    INX H          ;  
    MOV M,A        ;apothikeuoume ton A sth mnhmh  
    CPI 7FH        ;if A<127 epanalabe  
    JNZ STORE  
  
END
```

Θέσεις Μνήμης:

08FA	00	08FB	00	08FC	00	08FD	00	08FE	00	08FF	00	0900	00	0901	01	0902	02	0903	03
0904	04	0905	05	0906	06	0907	07	0908	08	0909	09	090A	0A	090B	0B	090C	0C	090D	0D
090E	0E	090F	0F	0910	10	0911	11	0912	12	0913	13	0914	14	0915	15	0916	16	0917	17
0918	18	0919	19	091A	1A	091B	1B	091C	1C	091D	1D	091E	1E	091F	1F	0920	20	0921	21
0922	22	0923	23	0924	24	0925	25	0926	26	0927	27	0928	28	0929	29	092A	2A	092B	2B
092C	2C	092D	2D	092E	2E	092F	2F	0930	30	0931	31	0932	32	0933	33	0934	34	0935	35
0936	36	0937	37	0938	38	0939	39	093A	3A	093B	3B	093C	3C	093D	3D	093E	3E	093F	3F
0940	40	0941	41	0942	42	0943	43	0944	44	0945	45	0946	46	0947	47	0948	48	0949	49
094A	4A	094B	4B	094C	4C	094D	4D	094E	4E	094F	4F	0950	50	0951	51	0952	52	0953	53
0954	54	0955	55	0956	56	0957	57	0958	58	0959	59	095A	5A	095B	5B	095C	5C	095D	5D
095E	5E	095F	5F	0960	60	0961	61	0962	62	0963	63	0964	64	0965	65	0966	66	0967	67
0968	68	0969	69	096A	6A	096B	6B	096C	6C	096D	6D	096E	6E	096F	6F	0970	70	0971	71
0972	72	0973	73	0974	74	0975	75	0976	76	0977	77	0978	78	0979	79	097A	7A	097B	7B
097C	7C	097D	7D	097E	7E	097F	7F	0980	00	0981	00	0982	00	0983	00	0984	00	0985	00

Εξήγηση:

Αποθηκεύονται οι αριθμοί 0-127 με αύξουσα σειρά στις διαδοχικές θέσεις της μνήμης με αρχή τη διεύθυνση 0900 H.

β)

```
IN 10H
MVI A,00H
LXI D,0900H
STAX D
LXI B,0000H

COUNT:
    INR A
    INX D
    STAX D
    MVI H,08H
FORE8:
    RLC
    JC ADD
    DCR H
    JNZ FORE8
    JMP COUNT
ADD:
    DCR H
    INX B
    JNZ FORE8

    CPI 7FH
    JNZ COUNT

LDA 0980H
MOV A,B
STA 0980H
LDA 0981H
MOV A,C
STA 0981H
END
```

Εξήγηση:

Υπολογίζουμε τον αριθμό των μονάδων (το πλήθος των δυαδικών ψηφίων '1') των παραπάνω δεδομένων.

Αποτέλεσμα σε hex:

0980	01	0981	C0	
------	----	------	----	--

δλδ 448.

Λογικό, εφόσον $7 \times 128 / 2 = 448$.

γ)

```
IN 10H
MVI A,00H
LXI B,0900H
STAX B
MVI D,00H

COUNT:
    INR A
    INX B
    STAX B
    CPI 10H
    JNC LESS60H
    CPI 7FH
    JNZ COUNT

LESS60H:
    CPI 61H
    JC ADD
    CPI 7FH
    JC COUNT
    JMP END

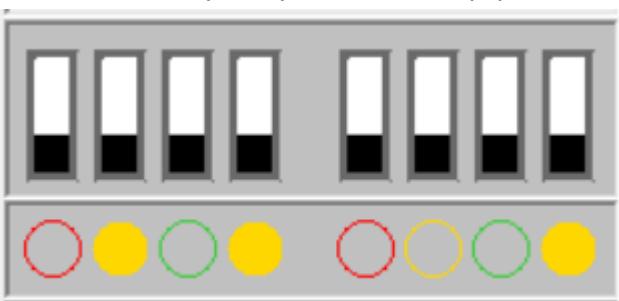
ADD:
    INR D
    JMP COUNT

END:
MOV A,D
CMA
STA 3000H
END
```

Εξήγηση:

Υπολογίζουμε το πλήθος από τους παραπάνω αριθμούς (0-127) που είναι μεταξύ των αριθμών 10H και 60H περιλαμβανομένων ($10 \leq X \leq 60$) και φυλάπτουμε το αποτέλεσμα στον καταχωρητή D.

Εκτυπώνουμε την τιμή του καταχωρητή D στα LED:



2η Άσκηση

```
START:
    LDA 2000H
    LXI B,0064H ;delay 100ms
    RLC
    JNC OFFACTIVATED
    JMP START

OFFACTIVATED:
    LDA 2000H
    RLC
    JC OFFONACTIVATED
    JMP OFFACTIVATED

OFFONACTIVATED:
    MVI D,32H
    LDA 2000H
    RLC
    JNC OFFONOFFFACTIVATED
    JMP OFFONACTIVATED

OFFONOFFFACTIVATED:
    LDA 2000H
    RLC
    JC OFFONACTIVATED AGAIN
    MVI A,00H
    STA 3000H
    CALL DELB
    DCR D
    JNZ OFFONOFFFACTIVATED
    CMA
    STA 3000H
    JMP OFFACTIVATED

OFFONACTIVATED AGAIN:
    LDA 2000H
    RLC
    JNC KALO_MHNA
    MVI A,00H
    STA 3000H
    CALL DELB
    DCR D
    JNZ OFFONACTIVATED AGAIN
    CMA
    STA 3000H
    JMP OFFACTIVATED

KALO_MHNA:
    MVI D,32H
    JMP OFFONACTIVATED

END
```

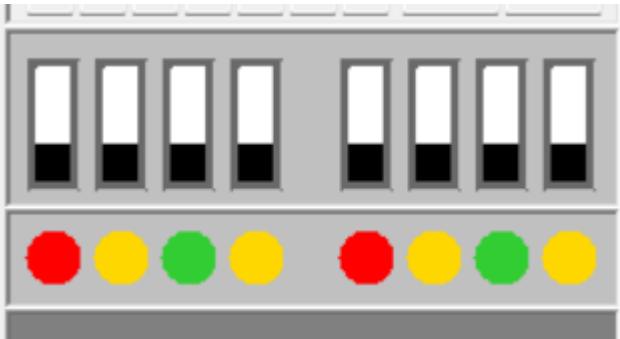
Εξήγηση:

Όταν το MSB της θύρας εισόδου dip switch (θέση μνήμης 2000 Hex) από OFF γίνει ON και ξανά OFF τότε να ανάβουν όλα τα LED της πόρτας εξόδου. Αυτά παραμένουν ανοιχτά για περίπου 20 sec και μετά σβήνουν. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί το push-button

(OFF - ON - OFF των dip switch) να ανανεώνεται ο χρόνος των 20 sec. Θεωρούμε ότι το σύστημα παρακολουθεί με διακριτική ικανότητα όχι μεγαλύτερη του 1/10 sec.

Παράδειγμα Εκτέλεσης:

Παρατηρούμε τη στιγμή, που το MSB των dip switch ήταν αρχικά OFF κι έπειτα έγινε ON και μετά ξανά OFF. Τότε τα λαμπάκια άνοιξαν για 20 δευτερόλεπτα:



3η Άσκηση

α)

```
START:
    LDA 2000H      ; dip switches input
    MVI C,FFH      ; arxikopoish Count=-1.*
    MVI B,00H
    CMP B          ; check if all switches are OFF
    JNZ COUNT_INR ; if at least one switch is ON, continue
                    ; else
    MVI A,FFH      ; close LED's
    STA 3000H      ; close LED's
    JMP START      ; check dip switches again

COUNT_INR:
    RRC
    INR C          ; Count++
    JC OPEN         ; if CY=1, OPEN LED's
    JMP COUNT_INR

OPEN:           ; here, Count has correct value*
    MVI A,FEH      ; 1111 1110
COUNT_DCR:     ; open the correct LED, according to Count
    DCR C
    RLC
    JNZ COUNT_DCR
    STA 3000H
    JMP START

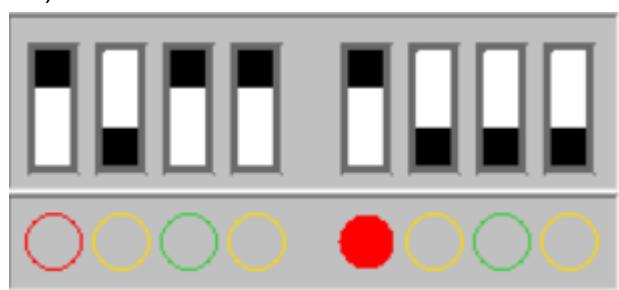
END
```

Εξήγηση:

με βάση το 1ο δεξιότερο ON, ανάβει το αντίστοιχης τάξης led

Παράδειγμα Εκτέλεσης:

Για 1011 1000 (dip switches) => OOOO OXOO (Led), όπου O=σβηστό led, X=αναμμένο led).



3)

```
START:
    CALL KIND      ; A is now pressed number
    CPI 00H
    JZ OTHER      ; if A=0, close all LED's
    CPI 09H
    JNC OTHER     ; if A>=9, close all LED's
    JMP OPEN_LEDs ; here, 1<=A<=8

OTHER:           ; Non valid number, close all LED's
    MVI A,FFH
    STA 3000H
    JMP START     ; and get another number

OPEN_LEDs:      ; compare with 1-8 and open corresponding LED's
    CPI 08H
    JZ OPEN8
    CPI 07H
    JZ OPEN7
    CPI 06H
    JZ OPEN6
    CPI 05H
    JZ OPEN5
    CPI 04H
    JZ OPEN4
    CPI 03H
    JZ OPEN3
    CPI 02H
    JZ OPEN2
    CPI 01H
    JZ OPEN1

OPEN8:
    MVI A,7FH
    STA 3000H
    JMP START

OPEN7:
    MVI A,3FH
    STA 3000H
    JMP START

OPEN6:
    MVI A,1FH
    STA 3000H
    JMP START

OPEN5:
    MVI A,0FH
    STA 3000H
    JMP START

OPEN4:
    MVI A,07H
    STA 3000H
    JMP START

OPEN3:
    MVI A,03H
    STA 3000H
    JMP START

OPEN2:
    MVI A,01H
    STA 3000H
    JMP START

OPEN1:
    MVI A,00H
    STA 3000H
    JMP START

END
```

Εξηγηση:

Αναμένει το πάτημα του δεκαεξαδικού πληκτρολογίου και μόνο των αριθμών 1 έως 8. Κάθε φορά ανάβει το led της αντίστοιχης θέσης και όλα τα υψηλότερης τάξης led μετά από αυτό. Είναι συνεχούς λειτουργίας.

Παράδειγμα Εκτέλεσης:

Πατήσαμε το νούμερο 3.

Επομένως, τα λαμπτάκια που άνοιξαν είναι το 3ο δεξιότερο και τα αριστερότερά του.



γ)

START:
 IN 10H

L1:
 MVI A,FEH ; arxikopoihsh tou A = 1h grammh
 STA 2800H
 LDA 1800H ; diavase sthlh
 MVI B,07H
 ANA B ; 3LSB
 MVI C,86H ; INSTR STEP
 CPI 06H
 JZ DISPLAY
 MVI C,85H ; FETCH PC
 CPI 05H
 JZ DISPLAY

L2:
 MVI A,FDH ; A = 2h grammh
 STA 2800H
 LDA 1800H ; diavase sthlh
 MVI B,07H
 ANA B ; 3LSB
 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

L3:
 MVI A,FBH ; A = 3h grammh
 STA 2800H
 LDA 1800H ; diavase sthlh
 MVI B,07H
 ANA B ; 3LSB
 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

L4:
 MVI A,F7H ; A = 4h grammh
 STA 2800H
 LDA 1800H ; diavase sthlh
 MVI B,07H
 ANA B ; 3LSB
 MVI C,01H ; 1
 CPI 06H
 JZ DISPLAY
 MVI C,02H ; 2
 CPI 05H
 JZ DISPLAY
 MVI C,03H ; 3
 CPI 03H
 JZ DISPLAY

```
-- -----
L5:
    MVI A,EFH      ; A = 5h grammh
    STA 2800H
    LDA 1800H      ; diavase sthlh
    MVI B,07H
    ANA B          ; 3LSB
    MVI C,04H      ; 4
    CPI 06H
    JZ DISPLAY
    MVI C,05H      ; 5
    CPI 05H
    JZ DISPLAY
    MVI C,06H      ; 6
    CPI 03H
    JZ DISPLAY
L6:
    MVI A,DFH      ; A = 6h grammh
    STA 2800H
    LDA 1800H      ; diavase sthlh
    MVI B,07H
    ANA B          ; 3LSB
    MVI C,07H      ; 7
    CPI 06H
    JZ DISPLAY
    MVI C,08H      ; 8
    CPI 05H
    JZ DISPLAY
    MVI C,09H      ; 9
    CPI 03H
    JZ DISPLAY
L7:
    MVI A,BFH      ; A = 7h grammh
    STA 2800H
    LDA 1800H      ; diavase sthlh
    MVI B,07H
    ANA B          ; 3LSB
    MVI C,0AH      ; A
    CPI 06H
    JZ DISPLAY
    MVI C,0BH      ; B
    CPI 05H
    JZ DISPLAY
    MVI C,0CH      ; C
    CPI 03H
    JZ DISPLAY
L8:
    MVI A,7FH      ; A = 8h grammh
    STA 2800H
    LDA 1800H      ; diavase sthlh
    MVI B,07H
    ANA B          ; 3LSB
    MVI C,0DH      ; D
    CPI 06H
    JZ DISPLAY
    MVI C,0EH      ; E
    CPI 05H
    JZ DISPLAY
    MVI C,0FH      ; F
    CPI 03H
    JZ DISPLAY
JMP START
```

```

DISPLAY:
LXI H,0924H ; So (ari8misi apo ta deksia) pshfio
MOV A,C ; A=code
ANI 0FH ; 4 LSB
MOV M,A ; apothikeush 4 LSB tou code (eg. if code=86 -> now: Block=060000)

INX H ; 6o pshfio
MOV A,C
ANI F0H ; 4 MSB
RLC
RLC
RLC ; metakinsh sta 4 LSB
MOV M,A ; apothikeush 4 MSB tou code (eg. if code=86 -> now: Block=860000)

LXI D,0920H ; dieuthunsh mhnumatos

CALL STDM ; metakinisi mhnumatos
CALL DCD ; emfanish mhnumatos
JMP START ; diavase epomeno mhnuma

END

```

Εξήγηση:

Γίνεται απευθείας ανάγνωση του πληκτρολογίου χωρίς τη χρήση της ρουτίνας KIND. Το αποτέλεσμα του κωδικού (βάσει του πίνακα 1):

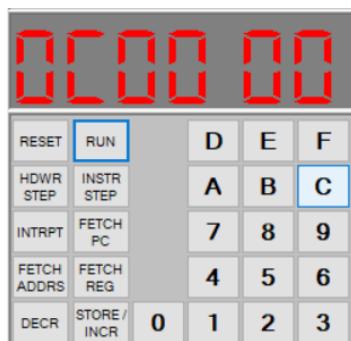
Πίνακας 1. Κωδικοί των πλήκτρων για τη ρουτίνα KIND

Πλήκτρο	Κωδικός	Πλήκτρο	Κωδικός
0	00	C	0C
1	01	D	0D
2	02	E	0E
3	03	F	0F
4	04	FETCH REG	80
5	05	DECR	81
6	06	FETCH ADRS	82
7	07	STORE/INCR	83
8	08	RUN	84
9	09	FETCH PC	85
A	0A	INSTR STEP	86
B	0B	HDWR STEP	F7

εμφανίζεται στα 2 αριστερότερα 7-segment display με βάση τις ρουτίνες DCD (Display Character Decoder) και STDM (Store Display Message).

Παράδειγμα Εκτέλεσης:

Πατήθηκε το πλήκτρο "C", επομένως βλέπουμε στην οθόνη τον διψήφιο κωδικό του (0C) στα 2 αριστερότερα 7-segment display.



4η Άσκηση

```
        IN 10H
START1:
        LDA 2000H
        LXI H,0928H
        MVI C,09H
START:
        DCR L
        DCR C
        RRC
        MVI M,00H
        JZ X3      ; Z=0 and
        JNC START  ; C=0
        MVI M,01H
        JNZ START

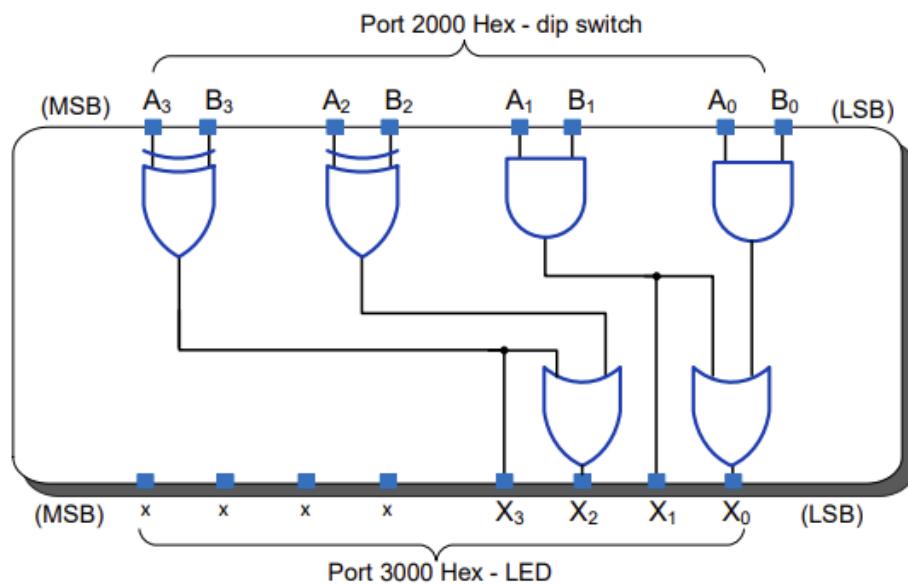
                                ; here, memory is correct
X3:
        LDA 0920H
        LXI H,0921H
        XRA M      ; X3 etoimo
        MOV C,A    ; antigrafo X3 for X2
        RLC
        RLC
        RLC
        MOV B,A    ; antigrafo me swsth 8esh X3

X2:
        LDA 0922H
        LXI H,0923H
        XRA M
        ORA C
        RLC
        RLC
        ADD B
        MOV B,A    ; corrected B for X3X2 here

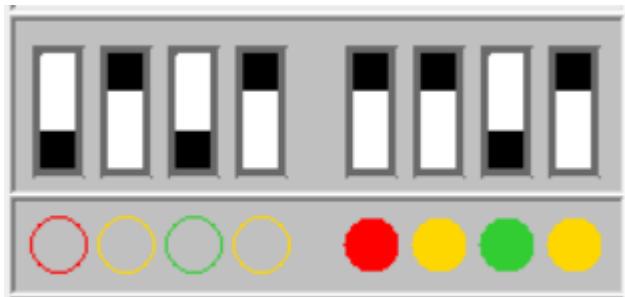
X1:
        LDA 0924H
        LXI H,0925H
        ANA M
        MOV C,A    ; antigrafo X1 for X0
        RLC
        ADD B
        MOV B,A

X0:
        LDA 0926H
        LXI H,0927H
        ANA M
        ORA C
        ADD B
        CMA
        STA 3000H
        JMP START1

END
```



Παράδειγμα εκτέλεσης:



Έχουμε:

$$B_0 = 1 \Rightarrow X_0 = 1$$

$$A_0 = 1 \Rightarrow$$

$$B_1 = 1 \Rightarrow X_2 = 1$$

$$A_1 = 1 \Rightarrow$$

$$B_2 = 1 \Rightarrow X_3 = 1$$

$$A_2 = 1 \Rightarrow$$

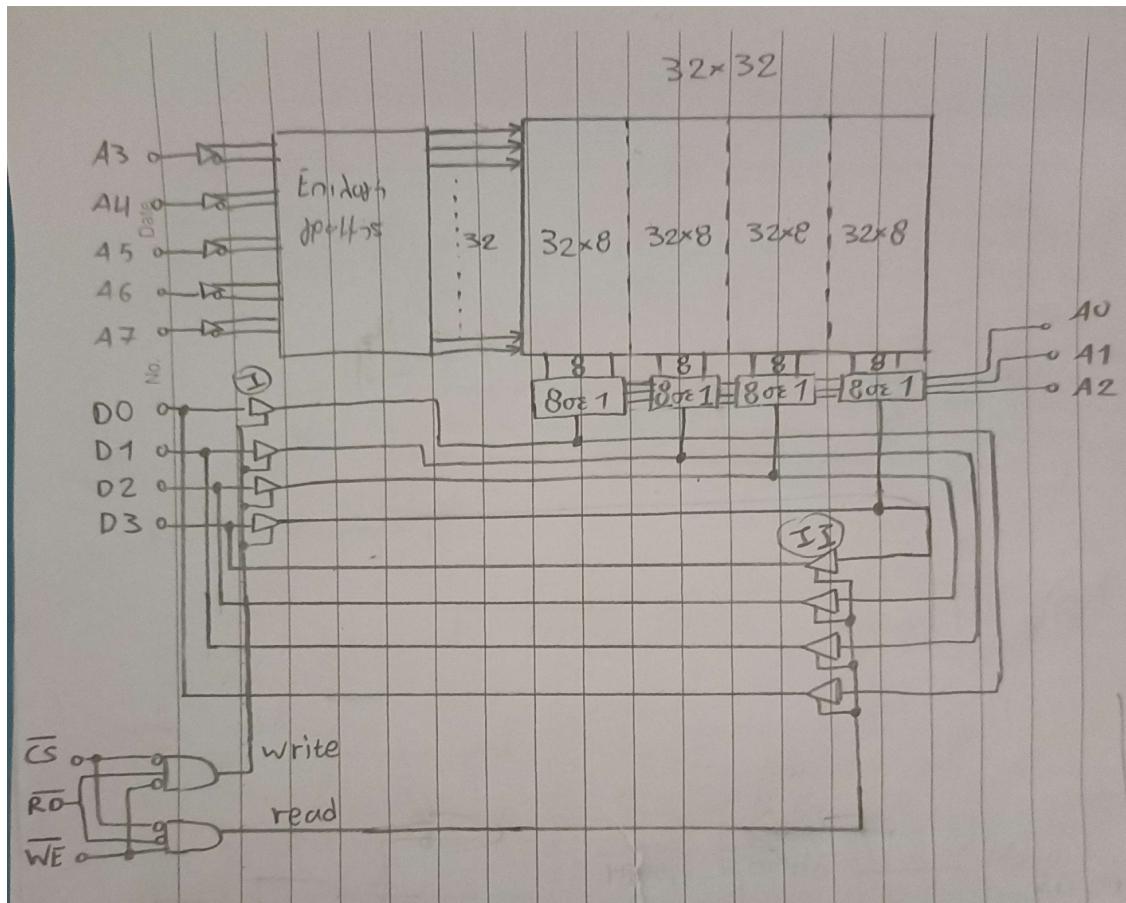
$$B_3 = 1 \Rightarrow X_4 = 1$$

$$A_3 = 1 \Rightarrow$$

Εξήγηση:

-Με βάση τα dip switches και την λογική των πυλών, ανοίγουν τα σωστά LED.

5η Ασκηση



Εξήγηση:

Επιλέγουμε τυχαία μια διεύθυνση πχ. 11001010

Τα 3 τελευταία bit (010) της παραπάνω διεύθυνσης εισέρχονται στους 4 MUX για την επιλογή της κατάλληλης τετράδας.

Τα 5 πρώτα bit της διεύθυνσης (11001) αντιστοιχούν στα A3-A7 και επιλέγουν την κατάλληλη γραμμή της μνήμης.

Όταν $\overline{CS} = 1 \rightarrow$ απομονώνεται η είσοδος/έξοδος

Όταν $\overline{CS} = 0 \rightarrow$ τότε ανάλογα με την τιμή του \overline{WE} και του \overline{RD} γίνεται εγγραφή ή ανάγνωση ($\overline{WE}=0 \& \overline{RD}=1 \rightarrow$ write / $\overline{WE}=1 \& \overline{RD}=0 \rightarrow$ read).

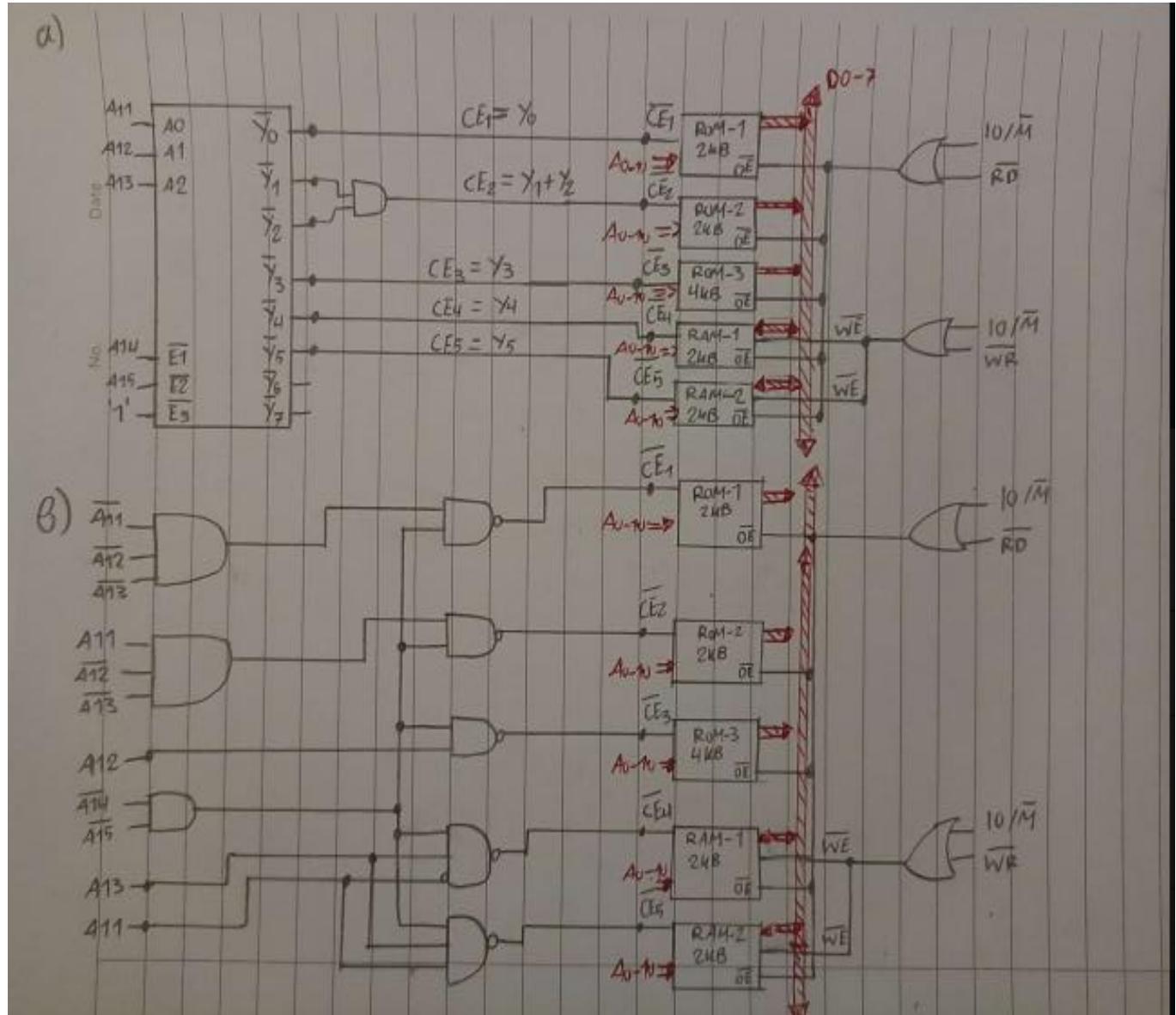
Όταν $write = 1$ τα buffer στο (I) θα γίνουν enable και οι τιμές των D0-D3 θα περάσουν στους MUX και επιπλέον αφού $read = 0$ τα buffer στο (I I) θα είναι disable.

Αντίστοιχα συμβαίνει και για όταν έχουμε $read = 1$, $write = 0$ τα buffer στο (I I) θα γίνουν enable και τα buffer στο (I) θα είναι disable.

6η Άσκηση

ΧΑΡΤΗΣ ΜΝΗΜΗΣ:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	address
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 (ROM-1)
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FF (ROM-1)
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800 (ROM-2)
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF (ROM-2)
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 (ROM-3)
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF (ROM-3)
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 (RAM-1)
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	27FF (RAM-1)
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	2800 (RAM-2)
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	2FFF (RAM-2)



7η Ασκηση

ΧΑΡΤΗΣ ΜΝΗΜΗΣ:

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	ROM1-8K
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	RAM1-4K
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	RAM2-4K
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	RAM3-4K
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	ROM2-8K
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	6FFF	

Το σύστημα μνήμης:

