

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Συστήματα Μικροϋπολογιστών

Σειρά Ασκήσεων 3	Θανάσουλας Γρηγόριος AM: 03114131 Μόνου Σταματίνα AM: 03114077
------------------	---

Άσκηση 1i

START: LDA 2000H ; Διάβασμα θύρας εισόδου
 LXI H,3000H ; Φόρτωση πόρτας εξόδου

 CPI 00H ; Αν όλα σβηστά
 JZ TURN_OFF

 MOV B,A ; Αποθήκευση του A στον B
 ANI 01H ; Μάσκα ψηφίου 0
 CPI 01H ; Σύγκριση
 JZ TURN_ON_0

 MOV A,B
 ANI 02H ; Μάσκα ψηφίου 1
 CPI 02H
 JZ TURN_ON_1

 MOV A,B
 ANI 04H ; Μάσκα ψηφίου 2
 CPI 04H
 JZ TURN_ON_2

 MOV A,B
 ANI 08H ; Μάσκα ψηφίου 3
 CPI 08H
 JZ TURN_ON_3

 MOV A,B
 ANI 10H ; Μάσκα ψηφίου 4
 CPI 10H
 JZ TURN_ON_4

```

MOV A,B
ANI 20H      ; Μάσκα ψηφίου 5
CPI 20H
JZ TURN_ON_5

MOV A,B
ANI 40H      ; Μάσκα ψηφίου 6
CPI 40H
JZ TURN_ON_6      ;

MOV A,B
ANI 80H      ; Μάσκα ψηφίου MSB 7
CPI 80H
JZ TURN_ON_7

```

```

TURN_ON_0: MVI M,00H      ; ON των αντιστοιχων LEDS
            JMP START
TURN_ON_1: MVI M,01H
            JMP START
TURN_ON_2: MVI M,03H
            JMP START
TURN_ON_3: MVI M,07H
            JMP START
TURN_ON_4: MVI M,0FH
            JMP START
TURN_ON_5: MVI M,1FH
            JMP START
TURN_ON_6: MVI M,3FH
            JMP START
TURN_ON_7: MVI M,7FH
            JMP START
TURN_OFF:  MVI M,FFH
            JMP START
END

```

Άσκηση 1ii

```

LXI H,3000H ; Φόρτωση της διεύθυνσης εξόδου
MVI M,FFH   ; Αρχική απενεργοποίηση LEDS
START:      CALL KIND      ; Κλήση ρουτίνας Key Input

CPI 01H      ; Κουμπί 1
JZ ON_1      ; Ενεργοποίηση LED 1

```

CPI 02H ; Κουμπί 2
JZ ON_2 ; Ενεργοποίηση LED 2

CPI 03H ; Κουμπί 3
JZ ON_3 ; Ενεργοποίηση LED 3

CPI 04H ; Κουμπί 4
JZ ON_4 ; Ενεργοποίηση LED 4

CPI 05H ; Κουμπί 5
JZ ON_5 ; Ενεργοποίηση LED 5

CPI 06H ; Κουμπί 6
JZ ON_6 ; Ενεργοποίηση LED 6

CPI 07H ; Κουμπί 7
JZ ON_7 ; Ενεργοποίηση LED 7

CPI 08H ; Κουμπί 8
JZ ON_8 ; Ενεργοποίηση LED 8

JMP START

ON_1: MVI M,FEH
JMP START
ON_2: MVI M,FDH
JMP START
ON_3: MVI M,FBH
JMP START
ON_4: MVI M,F7H
JMP START
ON_5: MVI M,EFH
JMP START
ON_6: MVI M,DFH
JMP START
ON_7: MVI M,BFH
JMP START
ON_8: MVI M,7FH
JMP START
END

Άσκηση 1iii

```
IN 10H
MVI A,10H    ; Κενό στα δεξιότερα 7-Segments
STA 0B00H
STA 0B01H
STA 0B02H
STA 0B03H
STA 0B04H
STA 0B05H
LXI D,0B00H
CALL STDM
CALL DCD
```

```
START:      MOV H,A
READ_LN_0:  MVI A,FEH    ; 1111 1110
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_IN_ST    ; Εμφάνιση κατάλληλου κωδικού
            CPI 05H      ; 101
            JZ DIS_F_PC ; Display Fetch PC
```

```
READ_LN_1:  MVI A,FDH    ; 1111 1101
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_RUN
            CPI 05H      ; 101
            JZ DIS_FETC_REG
            CPI 03H      ; 011
            JZ DIS_FETC_ADR
```

```
READ_LN_2:  MVI A,FBH    ; 1111 1011
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_0
            CPI 05H      ; 101
            JZ DIS_ST_INC
            CPI 03H      ; 011
            JZ DIS_INCR
```

```
READ_LN_3: MVI A,F7H    ; 1111 0111
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_1
            CPI 05H      ; 101
            JZ DIS_2
            CPI 03H      ; 011
            JZ DIS_3
```

```
READ_LN_4: MVI A,EFH    ; 1110 1111
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_4
            CPI 05H      ; 101
            JZ DIS_5
            CPI 03H      ; 011
            JZ DIS_6
```

```
READ_LN_5: MVI A,DFH    ; 1101 1111
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_7
            CPI 05H      ; 101
            JZ DIS_8
            CPI 03H      ; 011
            JZ DIS_9
```

```
READ_LN_6: MVI A,BFH    ; 1011 1111
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_A
            CPI 05H      ; 101
            JZ DIS_B
            CPI 03H      ; 011
            JZ DIS_C
```

```
READ_LN_7: MVI A,7FH    ; 0111 1111
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_D
            CPI 05H      ; 101
            JZ DIS_E
            CPI 03H      ; 011
            JZ DIS_F
```

```
MOV A,H
```

JMP KEEP_DISPLAY

READ_COLS:; Διαβάζει τις στήλες της γραμμής που έχει οριστεί στον A
; Αποθηκεύει τα 3 ψηφία στον A
STA 2800H ; Ενεργοποίηση Γραμμής
LDA 1800H ; Ανάγνωση στηλών
MVI B,07H ; Μάσκα 0000 0111
ANA B ; Αποθηκεύσει στον A τα τρία ψηφία
CONT: RET

DIS_IN_ST: MVI A,86H
JMP DISPLAY
DIS_F_PC: MVI A,85H
JMP DISPLAY
DIS_RUN: MVI A,84H
JMP DISPLAY
DIS_FETC_REG: MVI A,80H
JMP DISPLAY
DIS_FETC_ADR: MVI A,82H
JMP DISPLAY
DIS_ST_INC: MVI A,83H
JMP DISPLAY
DIS_INCR: MVI A,81H
JMP DISPLAY
DIS_0: MVI A,00H
JMP DISPLAY
DIS_1: MVI A,01H
JMP DISPLAY
DIS_2: MVI A,02H
JMP DISPLAY
DIS_3: MVI A,03H
JMP DISPLAY
DIS_4: MVI A,04H
JMP DISPLAY
DIS_5: MVI A,05H
JMP DISPLAY
DIS_6: MVI A,06H
JMP DISPLAY
DIS_7: MVI A,07H
JMP DISPLAY
DIS_8: MVI A,08H
JMP DISPLAY
DIS_9: MVI A,09H
JMP DISPLAY
DIS_A: MVI A,0AH

```

        JMP DISPLAY
DIS_B:  MVI A,0BH
        JMP DISPLAY
DIS_C:  MVI A,0CH
        JMP DISPLAY
DIS_D:  MVI A,0DH
        JMP DISPLAY
DIS_E:  MVI A,0EH
        JMP DISPLAY
DIS_F:  MVI A,0FH
        JMP DISPLAY

DISPLAY: MOV B,A      ; Δημιουργία αντιγράφου
        RRC          ; Απομόνωση MSB HEX ψηφίου
        RRC
        RRC
        RRC
        ANI 0FH      ; Μάσκα 0000 1111
        STA 0B05H    ; Αποθήκευση αριστερότερου ψηφίου

        MOV A,B      ; Απομόνωση LSB HEX ψηφίου
        ANI 0FH      ; Μάσκα 0000 1111
        STA 0B04H
        LXI D,0B00H
        CALL STDH
        CALL DCD
        MOV A,B
        JMP START

KEEP_DISPLAY: LXI D,0B00H
              CALL STDH
              CALL DCD
              JMP START

END

```

Άσκηση 1iv

```

IN 10H
EQU SYM_0,C0H
EQU SYM_1,F9H
EQU SYM_2,A4H
EQU SYM_3,B0H
EQU SYM_4,99H
EQU SYM_5,92H
EQU SYM_6,82H

```

```
EQU SYM_7,F8H
EQU SYM_8,80H
EQU SYM_9,90H
EQU SYM_A,88H
EQU SYM_B,83H
EQU SYM_C,C6H
EQU SYM_D,A1H
EQU SYM_E,86H
EQU SYM_F,8EH
EQU SYM_OFF,FFH
```

```
MVI C,SYM_OFF
MVI B,SYM_OFF
```

START:

```
READ_LN_0: MVI A,FEH    ; 1111 1110
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_IN_ST    ; Εμφάνιση κατάλληλου κωδικού
            CPI 05H      ; 101
            JZ DIS_F_PC ; Display Fetch PC
```

```
READ_LN_1: MVI A,FDH    ; 1111 1101
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_RUN
            CPI 05H      ; 101
            JZ DIS_FETC_REG
            CPI 03H      ; 011
            JZ DIS_FETC_ADR
```

```
READ_LN_2: MVI A,FBH    ; 1111 1011
            CALL READ_COLS
            CPI 06H      ; 110
            JZ DIS_0
            CPI 05H      ; 101
            JZ DIS_ST_INC
            CPI 03H      ; 011
            JZ DIS_DCR
```

```
READ_LN_3: MVI A,F7H    ; 1111 0111
            CALL READ_COLS
```



```

        CPI 06H      ; 110
        JZ DIS_1
        CPI 05H      ; 101
        JZ DIS_2
        CPI 03H      ; 011
        JZ DIS_3

READ_LN_4: MVI A,EFH  ; 1110 1111
          CALL READ_COLS
          CPI 06H      ; 110
          JZ DIS_4
          CPI 05H      ; 101
          JZ DIS_5
          CPI 03H      ; 011
          JZ DIS_6

READ_LN_5: MVI A,DFH  ; 1101 1111
          CALL READ_COLS
          CPI 06H      ; 110
          JZ DIS_7
          CPI 05H      ; 101
          JZ DIS_8
          CPI 03H      ; 011
          JZ DIS_9

READ_LN_6: MVI A,BFH  ; 1011 1111
          CALL READ_COLS
          CPI 06H      ; 110
          JZ DIS_A
          CPI 05H      ; 101
          JZ DIS_B
          CPI 03H      ; 011
          JZ DIS_C

READ_LN_7: MVI A,7FH  ; 0111 1111
          CALL READ_COLS
          CPI 06H      ; 110
          JZ DIS_D
          CPI 05H      ; 101
          JZ DIS_E
          CPI 03H      ; 011
          JZ DIS_F

        JMP DISPLAY

```

READ_COLS:; Διαβάζει τις στήλες της γραμμής που έχει οριστεί στον A

```
        ; Αποθηκεύει τα 3 ψηφία στον A
        STA 2800H    ; Ενεργοποίηση Γραμμής
        LDA 1800H    ; Ανάγνωση στηλών
        MVI D,07H    ; Μάσκα 0000 0111
        ANA D        ; Αποθηκεύσει στον A τα τρία ψηφία
CONT:    RET
```

```
DIS_IN_ST: MVI B,SYM_8
           MVI C,SYM_6
           JMP DISPLAY
```

```
DIS_F_PC: MVI B,SYM_8
          MVI C,SYM_5
          JMP DISPLAY
```

```
DIS_RUN:      MVI B,SYM_8
              MVI C,SYM_4
              JMP DISPLAY
```

```
DIS_FETC_REG: MVI B,SYM_8
              MVI C,SYM_0
              JMP DISPLAY
```

```
DIS_FETC_ADR: MVI B,SYM_8
              MVI C,SYM_2
              JMP DISPLAY
```

```
DIS_ST_INC: MVI B,SYM_8
            MVI C,SYM_3
            JMP DISPLAY
```

```
DIS_DCR:      MVI B,SYM_8
              MVI C,SYM_1
              JMP DISPLAY
```

```
DIS_0:         MVI B,SYM_0
              MVI C,SYM_0
              JMP DISPLAY
```

```
DIS_1:         MVI B,SYM_0
              MVI C,SYM_1
              JMP DISPLAY
```

```
DIS_2:         MVI B,SYM_0
              MVI C,SYM_2
```

```
                                JMP DISPLAY

DIS_3:    MVI B,SYM_0
          MVI C,SYM_3
          JMP DISPLAY

DIS_4:    MVI B,SYM_0
          MVI C,SYM_4
          JMP DISPLAY

DIS_5:    MVI B,SYM_0
          MVI C,SYM_5
          JMP DISPLAY

DIS_6:    MVI B,SYM_0
          MVI C,SYM_6
          JMP DISPLAY

DIS_7:    MVI B,SYM_0
          MVI C,SYM_7
          JMP DISPLAY

DIS_8:    MVI B,SYM_0
          MVI C,SYM_8
          JMP DISPLAY

DIS_9:    MVI B,SYM_0
          MVI C,SYM_9
          JMP DISPLAY

DIS_A:    MVI B,SYM_0
          MVI C,SYM_A
          JMP DISPLAY

DIS_B:    MVI B,SYM_0
          MVI C,SYM_B
          JMP DISPLAY

DIS_C:    MVI B,SYM_0
          MVI C,SYM_C
          JMP DISPLAY

DIS_D:    MVI B,SYM_0
          MVI C,SYM_D
          JMP DISPLAY
```

```

DIS_E:      MVI B,SYM_0
            MVI C,SYM_E
            JMP DISPLAY

DIS_F:      MVI B,SYM_0
            MVI C,SYM_F
            JMP DISPLAY

DISPLAY:    ; Προβολή 1ου ψηφίου
            MVI A,20H
            STA 2800H    ; Ορισμός αριστερότερου segment
            MOV A,B      ; Εμφάνιση του ψηφίου που έχει ο B
            STA 3800H
            MVI A,FFH    ; CLEAR
            STA 3800H

            ; Προβολή 2ου ψηφίου
            MVI A,10H
            STA 2800H    ; Ορισμός επόμενου αριστερότερου segment
            MOV A,C      ; Εμφάνιση του ψηφίου που έχει ο B
            STA 3800H
            MVI A,FFH    ; CLEAR
            STA 3800H

            JMP START

END

```

Άσκηση 5

Κώδικας μΥΣ-1

```

            MVI A,00H      ; SOD στέλνει 0
            SIM
START:      MVI B,FFH      ; Μετρητής B = 255
            MVI A,C0H      ; A = 1100 0000
            SIM            ; SOD1 = 1.

CHECK1:    RIM            ; Ανάγνωση θύρας SID1
            ANI 80H        ; Απομόνωση MSB
            CPI 80H        ; Σύγκριση με 1000 0000
            JNZ CHECK1     ; Αν δεν αναγνωρίστηκε ακμή, μείνε στο CHECK

```

```

SEND_DATA: MVI A,40H
            SIM                ; SOD1 = 0
            MOV A,M            ; Φόρτωση περιεχομένου μνήμης
            OUT DATA1         ; Αποστολή δεδομένων στην πόρτα DATA!
            INR H               ; Αύξηση διεύθυνσης κατά 1
            DCR B               ; Counter--
            JM END              ; Αν counter < 0, Τέλος
            JMP START           ; Επόμενη επανάληψη
END

```

Κώδικας μΥΣ-2

```

CHECK2A:    MVI C,FFH          ; Αρχικοποίηση C για το μικρότερο δεδομένο στη μέγιστη τιμή
            RIM                ; Έλεγχος τιμής SID2
            ANI 80H             ; Απομόνωση MSB
            CPI 80H             ; Σύγκριση με 1000 0000, αν SID2 = 1
            JNZ CHECK2A         ; Αναμονή μέχρι SID2 = 1

            MVI A,C0H           ; Προετοιμασία ώστε SOD2 = 1, A = 1100 0000
            SIM                ; SOD2 = 1

CHECK2B:    RIM                ; Έλεγχος τιμής SID2
            ANI 80H             ; Απομόνωση MSB
            CPI 00H             ; Σύγκριση με 0000 0000, αν SID2 = 0
            JNZ CHECK2B         ; Αναμονή μέχρι SID2 = 0

READ2:      IN DATA2           ; Ανάγνωση δεδομένων
            MOV M,A             ; Αποθήκευση δεδομένων στη μνήμη
            INR H               ; Αύξηση διεύθυνσης μνήμης
            SUB C               ; A - C
            JP NEXT             ; Αν A > C
            MOV C,A             ; Ορισμός νέου μικρότερου
NEXT:       MVI A,40H           ; Μηδενισμός SOD2
            SIM
            JMP CHECK2A         ; Επαναφορά σε κατάσταση αναμονής για δεδομένα
END

```

Άσκηση 6

Τα chips της RAM 8KB χρειάζονται ως είσοδο 13 ψηφία διεύθυνσης A12-A0.

Τα chips της ROM 16KB χρειάζονται ως είσοδο 14 ψηφία διεύθυνσης A13-A0, συνεπώς για να επιλέξει το κελί της, θα δίνεται ένας αριθμός εσωτερικά από το 00 0000 0000 0000 έως το 11 1111 1111 1111. Στην περίπτωση που βρισκόμαστε στο δεύτερο μισό της ROM τότε πρέπει αν αλλάξουμε την αντιστοίχιση των διευθύνσεων που στέλνει ο 8085 ώστε εσωτερικά να αντιστοιχούν στο δεύτερο μισό της ROM.

Επειδή το A15 στο χάρτη μνήμης μας έχει την τιμή 0, δίνεται ως είσοδος στην επιτρεψη E1' του αποκωδικοποιητή μας (ενεργή χαμηλά επιτρεψη).

HEX	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ROM Chip Shared
1FFF	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	
2000	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	RAM Chip 1
2FFF	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	
3000	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	RAM Chip 2
3FFF	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4000	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RAM Chip 3
4FFF	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
5000	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	ROM Chip Shared
6FFF	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	
7000	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Memory Map Output

Δηλαδή το παρακάτω τμήμα

5000	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	ROM Chip Shared
6FFF	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	

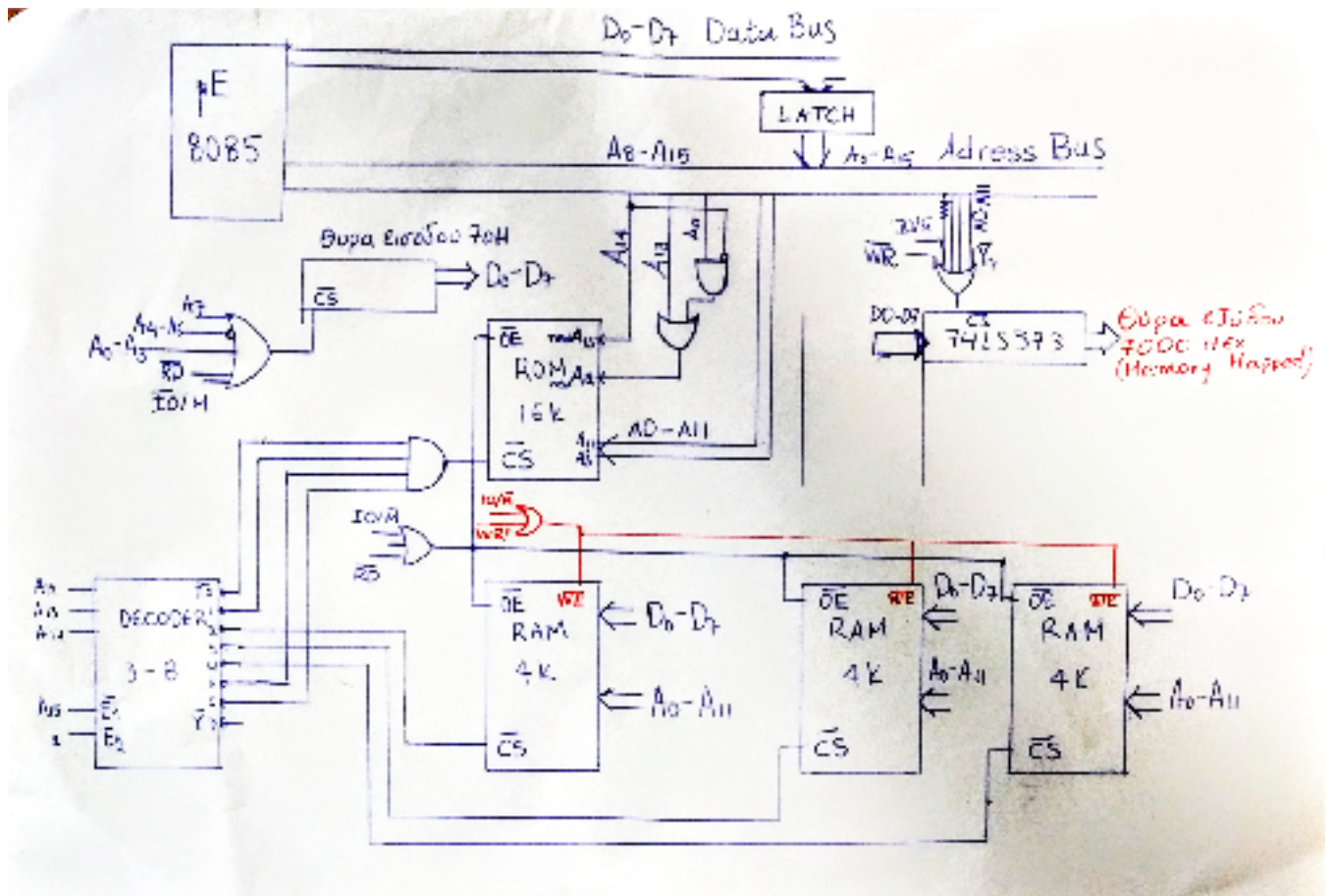
θα πρέπει να μετατρέπεται στο κάτωθι:

5000	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	ROM Chip Shared
6FFF	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Με βάση τον πίνακα αληθείας που βλέπουμε παρακάτω, και έχοντας λάβει υπόψιν ότι το chip select οδηγείται από τον αποκωδικοποιητή οπότε έχουμε την δυνατότητα να προσθέσουμε απροσδιόριστες καταστάσεις X στον πίνακα, βρίσκουμε ότι **NEWA13 = A14** και **NEWA12 = A14' * A12 + A13**

Μεταβλητές			Συναρτήσεις	
A14	A13	A12	NEW A13	NEW A12
0	0	0	0	0
0	0	1	0	1
0	1	0	X	X
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0
1	1	0	1	1
1	1	1	X	X

Η ROM θα είναι ενεργή όταν θα έχουν επιλεγεί από το αποκωδικοποιητή οι γραμμές Y0, Y1, Y5, Y6, και επιπλέον, θα δίνεται ως είσοδος στο 12ο και 13ο pin της ROM η τιμή NEW12 και NEW13 που προκύπτει από την υλοποίηση της λογικής συνάρτησης.



Άσκηση 7

; Ο Q παίρνει τις τιμές A, B, C, D, H, L, M. Αν δώσει όρισμα M εναλλάξει το περιεχόμενο της μνήμης.

SWAP MACRO Q

```
PUSH A
MOV A,Q           ; Φόρτωση του περιεχομένου στον A
RLC               ; Εναλλαγή κάνοντας 4 φορές RLC
RLC
RLC
RLC
MOV Q,A
POP A
```

ENDM

FILL MACRO ADDR,L,K

```
PUSH H           ; Τοποθέτηση καταχωρητών στη στοίβα
```

```

        PUSH B
        MVI C,L           ; Τοποθέτηση μήκους στον C
        LXI H,ADDR        ; Φόρτωση αρχικής διεύθυνσης
LOOP_1:  MVI M,K           ; Εγγραφή σταθεράς στη διεύθυνση μνήμης
        INX H             ; ADDRESS++
        DCR C             ; Μείωση C
        JNZ LOOP
        POP B             ; Επαναφορά καταχωρητών
        POP H
ENDM

```

Σημείωση: Αν L=0, τότε θα γίνει

RHLL MACRO n

```

        PUSH A
        PUSH B
        MOV B,n           ; Φόρτωση πλήθους στον B
LOOP_2:  MOV A,L           ; Φόρτωση του L στον A
        RAL
        MOV L,A
        MOV A,H
        RAL
        MOV H,A
        DCR B
        JNZ LOOP_2       ; Επανάληψη
        POP B
        POP A
END

```