

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΠΕΞΕΡΓΑΣΤΗ 8085



ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ  
ΦΡΟΝΤΙΣΤΗΡΙΑΚΗ ΔΙΑΛΕΞΗ 3

# ΕΙΣΟΔΟΣ – ΕΞΟΔΟΣ 8085 (I/O)

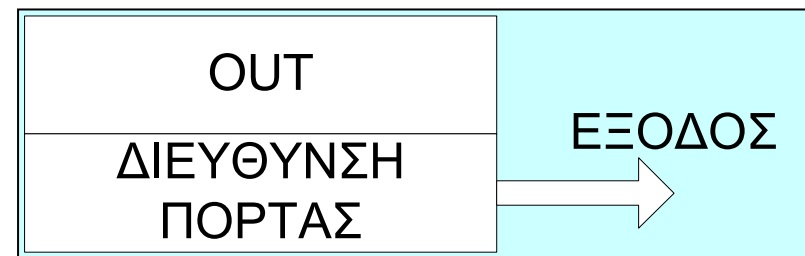
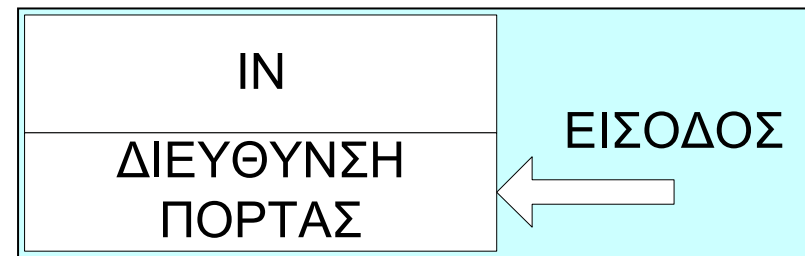
Εντολές Εισόδου – Εξόδου  
Δεδομένων στον 8085

IN port

π.χ. IN 20H

OUT port

π.χ. OUT 30H



# ΠΑΡΑΔΕΙΓΜΑ Ι

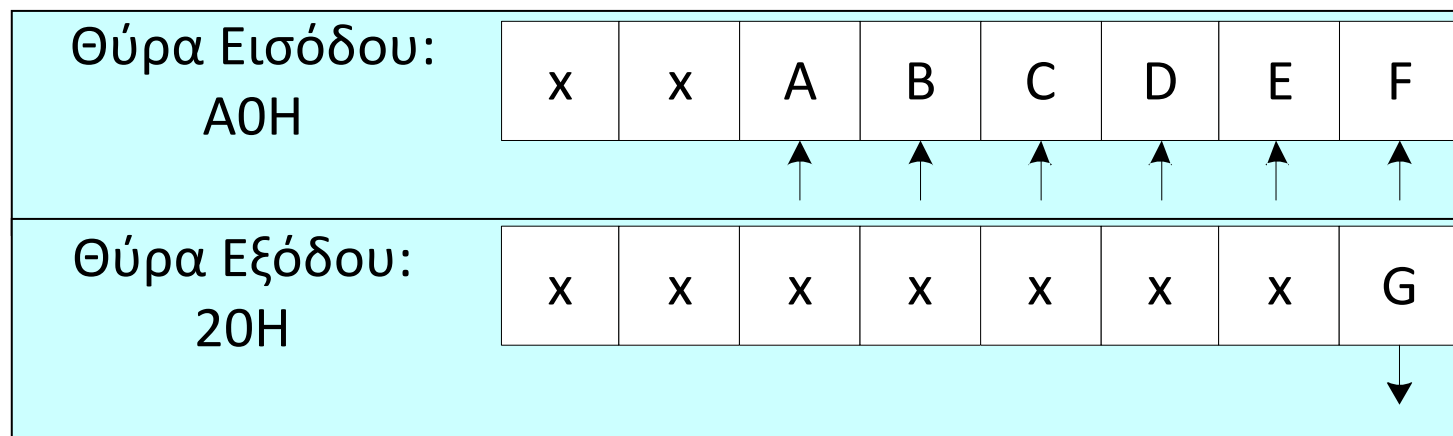
Να εξομοιωθεί σε μΥ-Σ 8085 η λογική συνάρτηση

$$G = AB'CD + AB'C'DE'F$$

Θα χρησιμοποιηθούν οι θύρες:

A0H (είσοδος)

20H (έξοδος)



## ΠΑΡΑΔΕΙΓΜΑ Ι - ΛΥΣΗ

```
APXH:  IN A0h
        MOV B,A
        ANI 00111100b    ; Απομονώνω τα A,B,C,D
        CPI 00101100b; Έλεγχος του όρου AB'CD
        JZ ADRI
        MOVA,B
        ANI 00111111b    ; Έλεγχος 2ου όρου
        CPI 00100101b; A B' C' D E' F
        JZ ADRI
        XRA A            ; A  $\square$  0
        JMP ADR0        ; G=0
ADRI:   MVI A,01H        ; G=1
ADR0:   OUT 20H; Έξοδος
        JMP APXH
        HLT
```

## ΠΑΡΑΔΕΙΓΜΑ 2

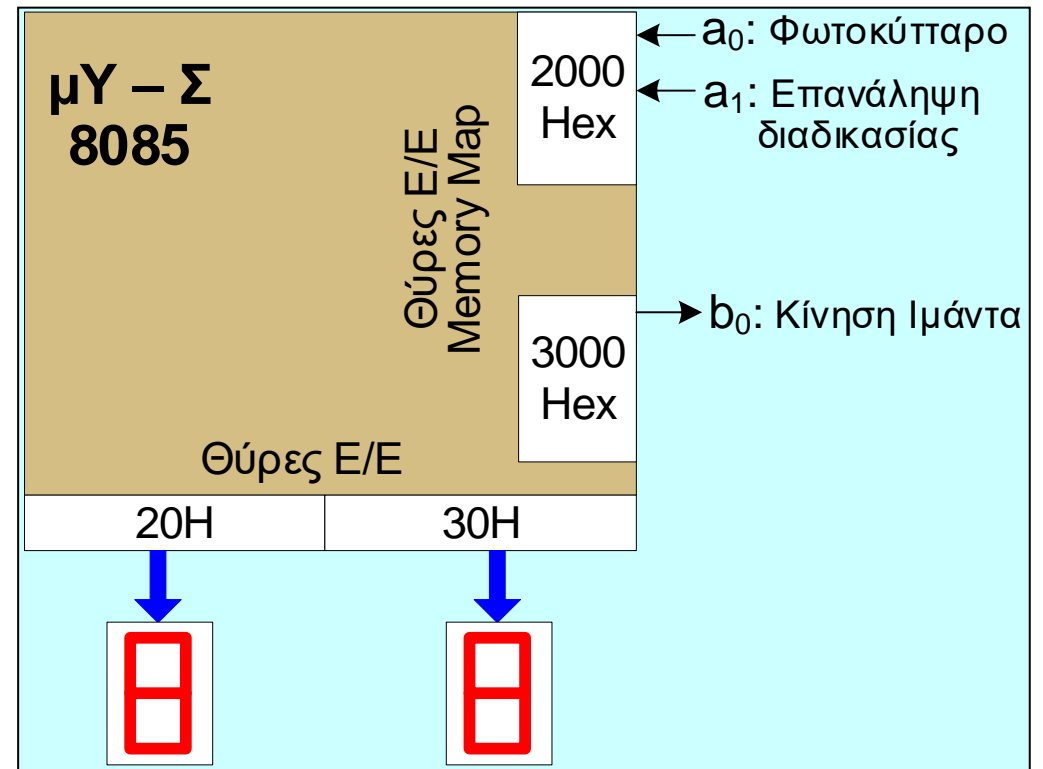
### Πρόγραμμα διακίνησης κιβωτίων:

Το bit  $a_0$  της θύρας εισόδου  $2000H$  οδηγείται από φωτοκύτταρο που απενεργοποιείται όταν διέρχεται κιβώτιο.

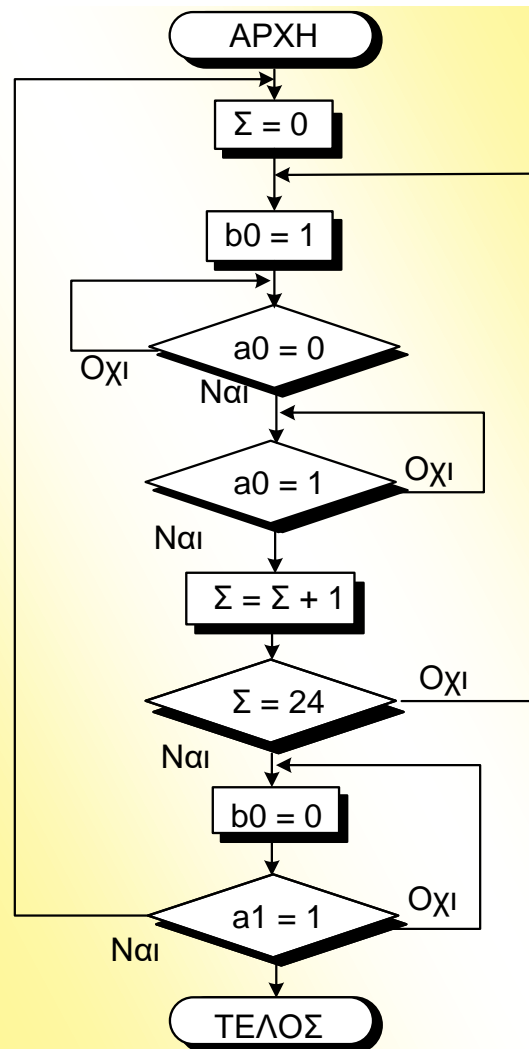
Το σύστημα αφού μετρήσει **24** κιβώτια σταματάει και αναμένει το bit  $a_1$  να γίνει '1', οπότε και επαναλαμβάνει.

Η κίνηση ελέγχεται από το bit  $b_0$  της θύρας εξόδου  $3000H$ :  $b_0=1$  κίνηση,  $b_0=0$  στάση.

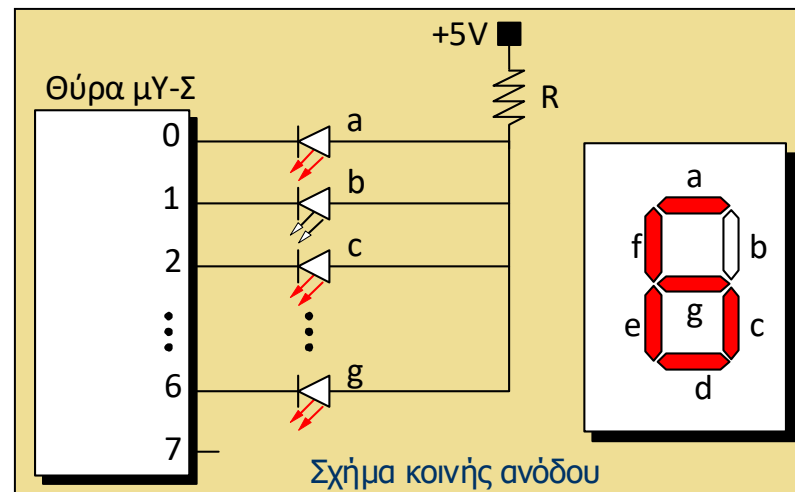
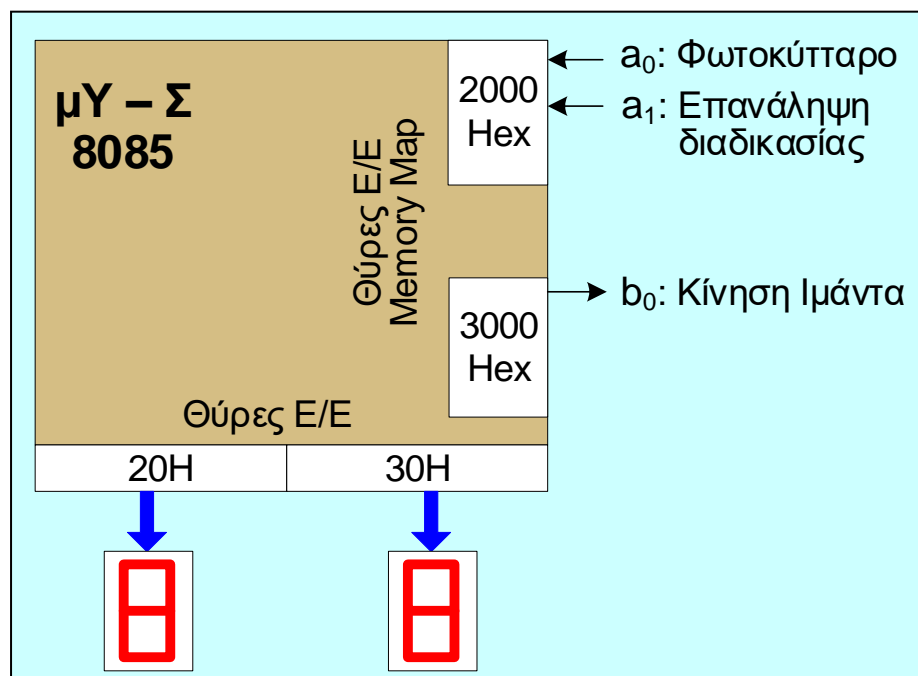
Ο αριθμός των κιβωτίων να απεικονίζεται σε δεκαδική μορφή σε δυο 7-segments που βρίσκονται στις θύρες  $20H$  και  $30H$  και οδηγούνται από τα 7 LSB των θυρών αυτών.



## ΠΑΡΑΔΕΙΓΜΑ 2 - ΛΥΣΗ

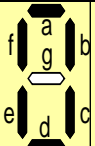
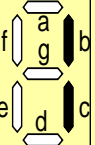
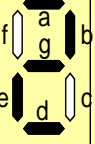
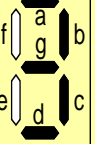
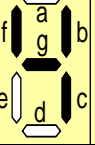


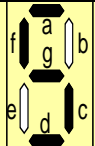
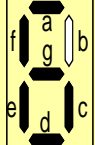
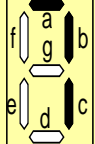
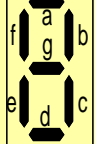
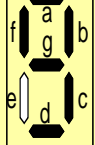
## ΠΑΡΑΔΕΙΓΜΑ 2 - ΔΙΑΤΑΞΗ



## ΠΑΡΑΔΕΙΓΜΑ 2 - ΚΩΔΙΚΟΙ

Οι κωδικοί για το 7-segment display

		x	g	f	e	d	c	b	a	
0		0	1	0	0	0	0	0	0	40H
1		0	1	1	1	1	0	0	1	79H
2		0	0	1	0	0	1	0	1	25H
3		0	0	1	1	0	0	0	0	30H
4		0	0	0	1	1	0	0	1	19H

		x	g	f	e	d	c	b	a	
5		0	1	0	0	0	1	0	0	12H
6		0	0	0	0	0	1	0	0	02H
7		0	1	1	1	0	1	0	0	78H
8		0	0	1	1	0	0	0	0	00H
9		0	0	0	1	1	0	0	0	10H



## Η ΧΡΗΣΗ ΤΗΣ ΕΝΤΟΛΗΣ ΔΕΚΑΔΙΚΗΣ ΡΥΘΜΙΣΗΣ – DAA

- Αν  $A_L > 9$  ή  $AC=1$  τότε  $A \leftarrow A+6$  (ή  $A_L \leftarrow A_L+6$ )
- Αν  $A_H > 9$  ή  $CY=1$  τότε  $A \leftarrow A+60$  (ή  $A_H \leftarrow A_H+6$ )

```
MVI A,27  
ADI 36 ;  $A \leftarrow 5DH = 93_{10}$   
DAA ;  $A \leftarrow 63$ 
```

```
MVI A,9  
INR A ;  $A \leftarrow 0AH = 10_{10}$   
DAA ;  $A \leftarrow 10_{10}$ 
```

```
MVI A,29  
ADI 18 ;  $A \leftarrow 41H$  ( $AC=1$ )  
DAA ;  $A \leftarrow 47$ 
```

## ΠΑΡΑΔΕΙΓΜΑ 2

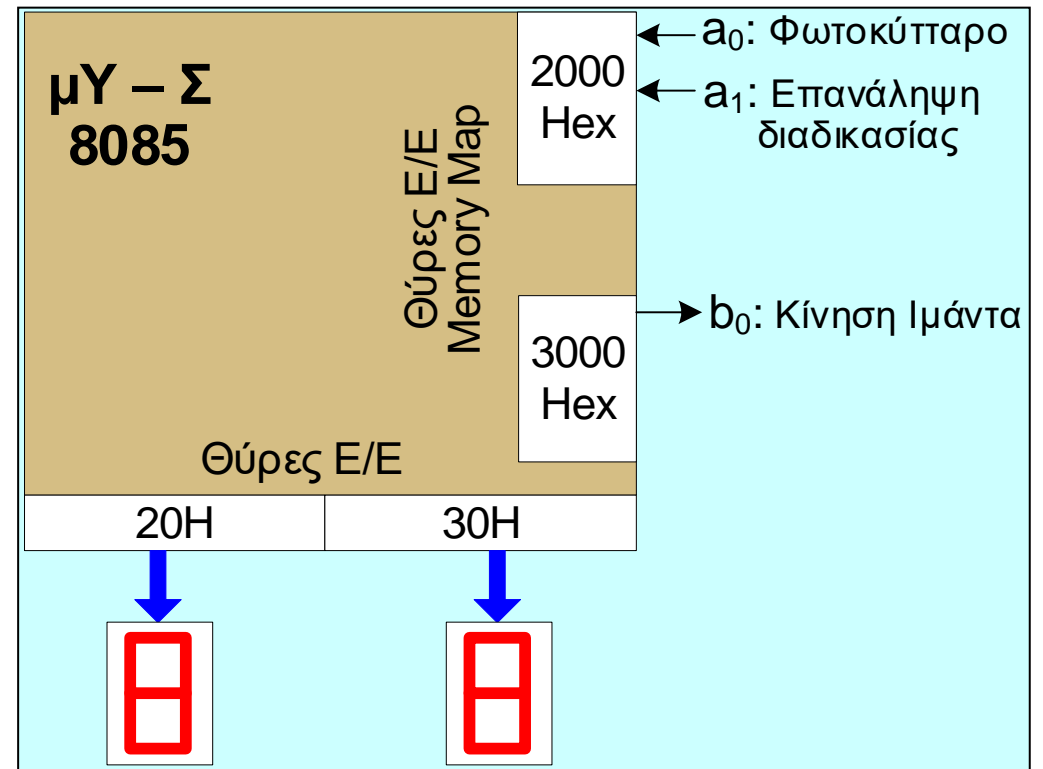
### Πρόγραμμα διακίνησης κιβωτίων:

Το bit  $a_0$  της θύρας εισόδου  $2000H$  οδηγείται από φωτοκύτταρο που απενεργοποιείται όταν διέρχεται κιβώτιο.

Το σύστημα αφού μετρήσει **24** κιβώτια σταματάει και αναμένει το bit  $a_1$  να γίνει '1', οπότε και επαναλαμβάνει.

Η κίνηση ελέγχεται από το bit  $b_0$  της θύρας εξόδου  $3000H$ :  $b_0=1$  κίνηση,  $b_0=0$  στάση.

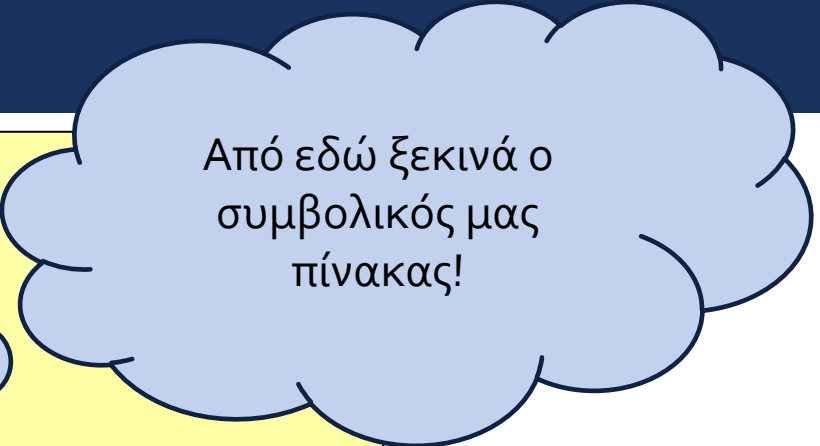
Ο αριθμός των κιβωτίων να απεικονίζεται σε δεκαδική μορφή σε δυο 7-segments που βρίσκονται στις θύρες  $20H$  και  $30H$  και οδηγούνται από τα 7 LSB των θυρών αυτών.



## ΠΑΡΑΔΕΙΓΜΑ 2 - ΛΥΣΗ

A1: MVI B,00H ; B: Μετρητής κιβωτίων  
LXI H,0900H ; Στη συμβολική TABLE  
MOV A,M ; Αντιστοιχεί η διεύθυνση 0900H  
OUT 20H ; Απεικόνιση αρχικής  
OUT 30H ; Τιμής στα 7-segments

A2: MVI A,01H ;  $b_0 = 1$   
STA 3000H  
LDA 2000H ; Αν  $a_0 = 0$  συνέχισε  
RRC  
JCA2

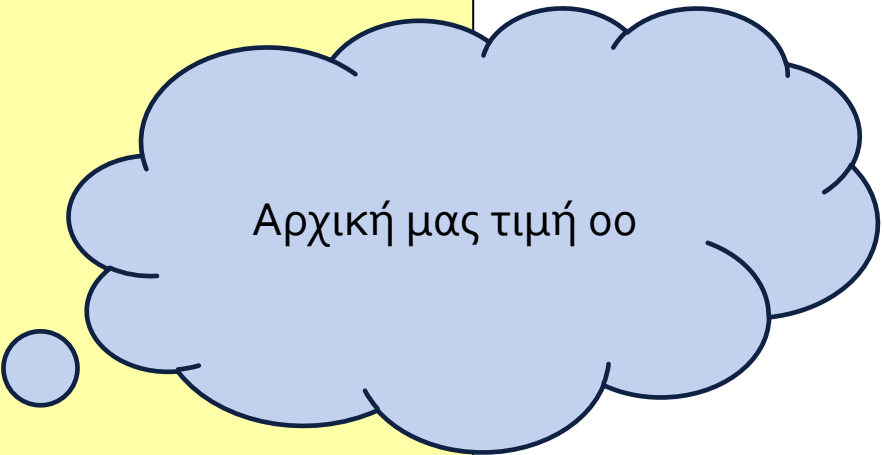


Από εδώ ξεκινά ο  
συμβολικός μας  
πίνακας!

## ΠΑΡΑΔΕΙΓΜΑ 2 - ΛΥΣΗ

A1: MVI B,00H ; B: Μετρητής κιβωτίων  
LXI H,0900H ; Στη συμβολική TABLE  
MOV A,M ; Αντιστοιχεί η διεύθυνση 0900H  
OUT 20H ; Απεικόνιση αρχικής  
OUT 30H ; Τιμής στα 7-segments

A2: MVI A,01H ;  $b_0 = 1$   
STA 3000H  
LDA 2000H ; Αν  $a_0 = 0$  συνέχισε  
RRC  
JCA2



Αρχική μας τιμή 00

## ΠΑΡΑΔΕΙΓΜΑ 2 - ΛΥΣΗ

A1:	MVI B,00H	; B: Μετρητής κιβωτίων
	LXI H,0900H	; Στη συμβολική TABLE
	MOV A,M	; Αντιστοιχεί η διεύθυνση 0900H
	OUT 20H	; Απεικόνιση αρχικής
	OUT 30H	; Τιμής στα 7-segments
A2:	MVI A,01H	; $b_0 = 1$
	STA 3000H	
	LDA 2000H	; Αν $a_0 = 0$ συνέχισε
	RRC	
	JCA2	

Αν το  $a_0$  ήταν ακόμα 1  
θα ήταν από πριν!  
Πρέπει πρώτα να γίνει 0  
γιατί αλλιώς θα  
μετράμε πολλές φορές  
το ένα κυβώτιο

## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ Β ΜΕΡΟΣ

Εδώ περιμένουμε το  
φωτοκύτταρο για το  
επόμενο

A3: LDA 2000H ; Αναμονή μέχρι να γίνει  
RRC ;  $a_0 = 1$   
JNC A3  
MOV A,B  
INR A  
CMC ; Καθαρισμός κρατουμένου!!!  
DAA ; Δεκαδική διόρθωση (π.χ.  $0A^H \Rightarrow 10^H$ )  
MOV B,A  
RRC (4 φορές) ; Τα 4 MSB μεταφέρονται στις 4 LSB θέσεις  
ANI 0FH ; Απομόνωση του MSD (δεκάδες)  
MOV L,A ; Το έχει τη τιμή  $H=09^H$  και το L έχει τις δεκάδες  
MOV A,M ; Ανάγνωση του 7-segment κώδικα  
OUT 20H ; Απεικόνιση των δεκάδων στα 7-segments (θύρα  $20^H$ )

## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ Β ΜΕΡΟΣ

A3:	LDA 2000H	; Αναμονή μέχρι να γίνει
	RRC	; $a_0 = 1$
	JNC A3	
	MOV A,B	
	INR A	
	CMC	; Καθαρισμός κρατούμενου!!!
	DAA	; Δεκαδική διόρθωση (π.χ. $0A^H \Rightarrow 10^H$ )
	MOV B,A	
	RRC (4 φορές)	; Τα 4 MSB μεταφέρονται στις 4 LSB θέσεις
	ANI 0FH	; Απομόνωση του MSD (δεκάδες)
	MOV L,A	; Το έχει τη τιμή $H=09^H$ και το L έχει τις δεκάδες
	MOV A,M	; Ανάγνωση του 7-segment κώδικα
	OUT 20H	; Απεικόνιση των δεκάδων στα 7-segments (θύρα $20^H$ )

DAA θα μας το μετατρέψει σε δεκαδικό! Αλλά θέλουμε να μην υπάρχει κρατούμενο!

## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ Β ΜΕΡΟΣ

A3: LDA 2000H ; Αναμονή μέχρι να γίνει  
RRC ;  $a_0 = 1$   
JNC A3  
MOV A,B  
INR A  
CMC ; Καθαρισμός κρατουμένου!!!  
DAA ; Δεκαδική διόρθωση (π.χ.  $0A^H \Rightarrow 10^H$ )  
MOV B,A  
RRC (4 φορές) ; Τα 4 MSB μεταφέρονται στις 4 LSB θέσεις  
ANI 0FH ; Απομόνωση του MSD (δεκάδες)  
MOV L,A ; Το έχει τη τιμή  $H=09^H$  και το L έχει τις δεκάδες  
MOV A,M ; Ανάγνωση του 7-segment κώδικα  
OUT 20H ; Απεικόνιση των δεκάδων στα 7-segments (θύρα  $20^H$ )

Από τον τρόπο που  
φτιάξαμε τον πίνακα θα  
αντιστοιχεί το 0  
7segment σε  $L=0$  και το  
9 – 7segment σε  $L=9$



## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ Γ ΜΕΡΟΣ

	MOV A,B	; Επαναφορά τιμής
	ANI 0FH	; Απομόνωση του LSD (μονάδες)
	MOV L,A	; των 7-segments (θύρα 30 <sup>H</sup> )
	MOV A,M	; Ανάγνωση του 7-segment κώδικα
	OUT 30H	; Απεικόνιση των μονάδων
	MVI A, <b>24H</b>	; Έλεγχος αν πέρασαν 24 κιβώτια
	CMP B	
	JNZ A2	
A4:	MVI A,00H	
	STA 3000H	; b0 = 0
	LDA 2000H	
	RRC	
	RRC	
	JCAI	; Αν AI = 1 πήγαινε στο A1
	JMP A4	
	HLT	

TABLE: 40H,79H,25H,30H,19H,12H,02H,78H,00H,10H

Γιατί 24H κι όχι 18H???

## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ ΣΤΟΝ ΤΣΙΚ

```
MVI A,10H                ;Κενο στα αχρησιμοποιήτα leds
STA 0BF2H
STA 0BF3H
STA 0BF4H
STA 0BF5H
MVI A,00H
STA 0BF0H
STA 0BF1H
CALL DCD
A1:  MVI B,00H            ; B: Μετρητής κιβωτίων
A2:  MVI A,01H            ; b0 = 1
      STA 3000H
      LDA 2000H           ; Αν a0 = 0 συνέχισε
      RRC
      JC A2
A3:  LDA 2000H            ; Αναμονή μέχρι να γίνει
      RRC                ; a0 = 1
      JNC A3
      MOV A,B
      INR A
      CMC                ;Καθαρισμός του κρατουμένου!
      DAA                ; Δεκαδική διόρθωση (π.χ. 0AH =>10H)
      MOV B,A
```

## ΠΑΡΑΔΕΙΓΜΑ 2 – ΛΥΣΗ ΣΤΟΝ ΤΣΙΚ (Β ΜΕΡΟΣ)

```
RRC
RRC
RRC
RRC                ;Τα 4 MSB μεταφέρονται στις 4 LSB θέσεις
ANI 0FH            ;Απομόνωση του MSD (δεκάδες)
STA 0BF1H
CALL DCD
MOV A,B            ;Επαναφορά τιμής
ANI 0FH            ;Απομόνωση του LSD (μονάδες)
STA 0BF0H
CALL DCD
MVI A,24H          ;Έλεγχος αν πέρασαν 24 κιβώτια
CMP B
JNZ A2
A4: MVI A,00H
STA 3000H          ;b0 = 0
LDA 2000H
RRC
RRC
JCAI                ;Αν AI = 1 πήγαινε στο A1
JMP A4
END
```

## ΠΑΡΑΔΕΙΓΜΑ 3

Δίνονται 25 αριθμοί. Να αποθηκευτούν σε διαδοχικές θέσεις, αρχίζοντας από την  $300H$  όσοι βρίσκονται μεταξύ  $10H$  και  $F0H$ . Αλλιώς στην αντίστοιχη θέση να αποθηκευτεί μηδέν. Τα δεδομένα παρέχονται από τη θύρα εισόδου  $10h$ .

## ΠΑΡΑΔΕΙΓΜΑ 3 - ΛΥΣΗ

```
LXI H,0300H      ; Αρχικοποιήσεις
MVI B,10H        ; Τα 2 όρια
MVI C,F0H
MVI D,19H        ; Ο μετρητής
INPUT: IN 10H     ; Είσοδος
CMP B
JC ZERO          ; X<10H
CMP C
JNC ZERO         ; X>F0H
MOV M,A          ; Εντός των ορίων
JMP CONT
ZERO: MVI M,00    ; Εκτός ορίων, συμπληρώνεται με 0
CONT: INX H
DCR D            ; Ελέγχω για 25 Αριθμούς
JNZ INPUT
HLT
```

## ΠΑΡΑΔΕΙΓΜΑ 4

Εδώ υποθέτουμε ότι το πρωτόκολλο επιτρέπει είσοδο νέου δεδομένου μόνο αν η τιμή του είναι διαφορετική από το προηγούμενο δεδομένο.

```
MVI E,0      ; θεωρώ στο  $x_1=0$   
Input: IN 10h  
CMP E  
JZ Input  
MOV E,A  
; Επεξεργασία, πράξεις, κλπ  
JMP Input
```

Προϋπόθεση: Η συσκευή είναι *αργή* και ο μΥ κάνει polling συνεχώς την θύρα των δεδομένων

## ΠΑΡΑΔΕΙΓΜΑ 5

Υπολογισμός συχνότητας εμφάνισης δεδομένων

Δίνεται η περιοχή μνήμης 0000 – 07FFH. Να υπολογισθεί η συχνότητα εμφάνισης ενός αριθμού XY (HEX) και να τοποθετηθεί στον πίνακα και στη θέση μνήμης 08XYH.

## ΠΑΡΑΔΕΙΓΜΑ 5 - ΛΥΣΗ

```
MVI D,08H
LXI H,0000
LOOP: MOV E,M ; Με βάση το δεδομένο XY
LDAX D ; Διαβάζεται η θέση 08XY
INR A ; Αυξάνεται η συχνότητα
STAX D ; και αποθηκεύεται
INX H ; Επόμενος αριθμός
MOV A,H
CPI 08H ; Τερματίζεται για 0080h
JNZ LOOP
HLT
```

Υποθέτουμε ότι η συχνότητα κάθε αριθμού XY δεν ξεπερνάει το 255. Αλλιώς χρειάζεται πρόβλεψη για υπερχείλιση (με 2 θέσεις για κάθε αριθμό).



## ΕΝΤΟΛΕΣ ΠΑΡΑΔΕΙΓΜΑΤΟΣ 5

- LDAX κ

- Load accumulator indirect
- Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στο ζεύγος καταχωρητών μετακινείται στον A

- STAX κ

- Store accumulator indirect
- Το περιεχόμενο του A μεταφέρεται στη θέση μνήμης της οποίας η διεύθυνση περιέχεται στο ζεύγος καταχωρητών

## ΠΑΡΑΔΕΙΓΜΑ 6

Πρόγραμμα διαίρεσης :  $X / Y$

```
LHLD 0040H    ; H-L ← X διαιρετέος
LDA 0042H      ; A ← Y διαιρέτης
MOV C,A
MVI B,08H

DIV:
DAD H
MOV A,H        ; Αφαιρώ διαιρέτη από
SUB C          ; από διαιρετέο
JC ADDR        ; Αν δεν χωράει πηλίκο 0
MOV H,A        ; Το υπόλοιπο στον κατ. H-L
INR L          ; Και το bit πηλίκου=1

ADDR: DCR B     ; 8 Βήματα
JNZ DIV
SHLD 0043H
HLT
```

40	$X_{\text{LSB}}$
41	$X_{\text{MSB}}$
42	Y
43	Πηλίκο
44	Υπόλοιπο


## ΠΑΡΑΔΕΙΓΜΑ 7

Κατάταξη αριθμών σε σειρά φθίνοντος μεγέθους.

Το πλήθος των αριθμών βρίσκεται στη θέση  $00_4oH$  και η αρχή τους είναι στη θέση  $00_41H$ .

## ΠΑΡΑΔΕΙΓΜΑ 7 - ΛΥΣΗ

```
SORT: MVI B,0           ; B=Δείκτης
      Εναλλαγής
      LXI H,0040         ; (H-L) ← 0040
      MOV C,M            ; C=Πλήθος
      DCR C
      INX H              ; (H-L) ← 0041
COMP: MOV A,M            ; (A) ← Αριθμός x(n)
      INX H
      CMP M              ; Σύγκρινε x(n), x(n+1)
      JNC CNT            ; x(n)-x(n+1)>=0
```



```
MOV D,M
MOV M,A
DCX H
MOV M,D
INX H
MVI B,1
CNT: DCR C
      JNZ COMP
      DCR B
      JZ  SORT
      HLT
```

Αντάλλαξε  $x(n)$ ,  $x(n+1)$

; Έγινε μια εναλλαγή  
; Να επαναληφθεί N-1 φορές