

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΠΕΞΕΡΓΑΣΤΗ 8085



ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ
ΦΡΟΝΤΙΣΤΗΡΙΑΚΗ ΔΙΑΛΕΞΗ 4

ΕΙΣΟΔΟΣ – ΕΞΟΔΟΣ 8085 (I/O)

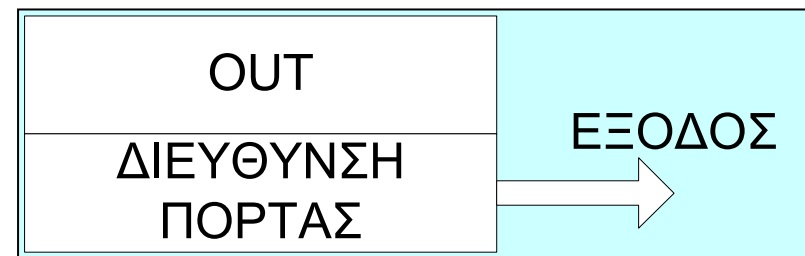
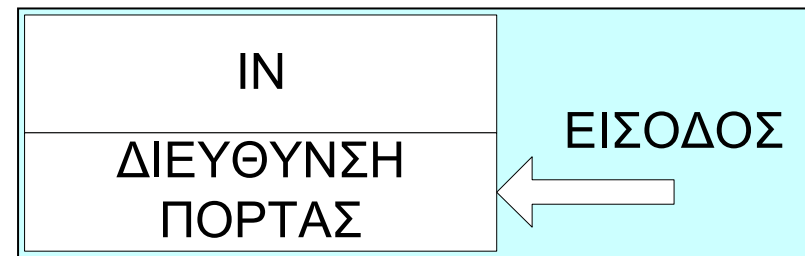
Εντολές Εισόδου – Εξόδου
Δεδομένων στον 8085

IN port

π.χ. IN 20H

OUT port

π.χ. OUT 30H



ΜΑΚΡΟΕΝΤΟΛΕΣ

- Μια μακροεντολή, όπως μια διαδικασία, είναι μια ομάδα εντολών που εκτελούν μια εργασία.
- Οι εντολές τοποθετούνται στο πρόγραμμα από τον assembler στο σημείο που καλείται.
- Η χρήση μακροεντολών βοηθά στη δημιουργία νέων οδηγιών που θα αναγνωρίζονται από το συμβολομεταφραστή.
- Με την χρήση μακροεντολών μπορούμε να επεκτείνουμε το βασικό σύνολο εντολών του 8085 που έχουμε στην διάθεσή μας

ΜΑΚΡΟΕΝΤΟΛΕΣ

- Ο ορισμός της μακροεντολής ξεκινά με το όνομα.
- Οι παράμετροι είναι προαιρετικές και επιτρέπουν την ίδια μακροεντολή να χρησιμοποιείται σε διαφορετικά σημεία μέσα σε ένα πρόγραμμα με διαφορετικά δεδομένα σαν είσοδο.

όνομα **MACRO** <παράμετρος>

Σώμα της macro

ENDM

ΠΑΡΑΔΕΙΓΜΑ ΜΑΚΡΟΕΝΤΟΛΗΣ

Υλοποιήστε μια μακροεντολή, SWAP Q,R που εναλλάσσει τα περιεχόμενα οποιονδήποτε δύο καταχωρητών γενικού σκοπού B, C, D, E, H και L. Η εκτέλεση της μακροεντολής δεν πρέπει να επηρεάζει τα περιεχόμενα των υπολοίπων καταχωρητών που δεν μετέχουν στην εναλλαγή.

SWAP MACRO

```
SWAP MACRO Q,R
    PUSH PSW    ; στοίβα←A, F
    MOV A,Q
    MOV Q,R
    MOV R,A
    POP PSW     ; A, F←στοίβα
ENDM
```

Για παράδειγμα η εντολή SWAP H,B
αντικαθίσταται από το παρακάτω σύνολο εντολών:

```
PUSH PSW
MOV A,H
MOV H,B
MOV B,A
POP PSW
```

ΣΤΟΙΒΑ

- Μια δομή LIFO
- Χρησιμοποιείται από την αριθμητική λογική μονάδα (ALU) για αποθήκευση διευθύνσεων επιστροφής
- Χρειάζεται τον δείκτη στοίβας (SP – Stack Pointer)
- Χρησιμεύει επίσης για την αποθήκευση καταχωρητών όταν ζητούνται δεδομένα και οι καταχωρητές δεν επαρκούν (push, pop)

ΕΝΤΟΛΕΣ ΣΤΟΙΒΑΣ - PUSH

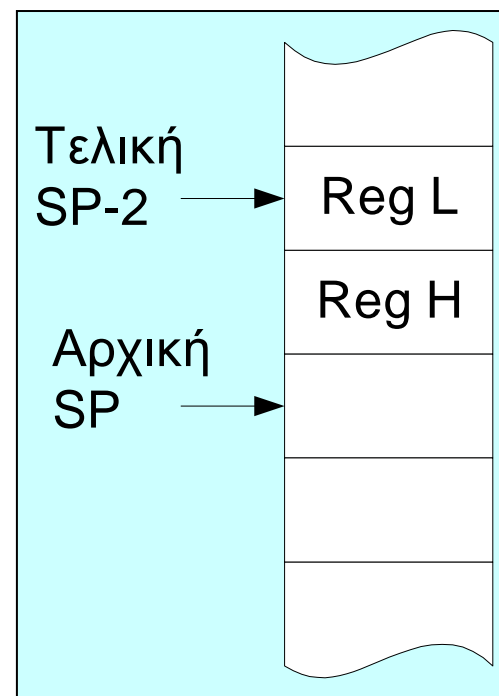
PUSH Reg Pair

$((SP) - 1) \leftarrow (Reg\ H)$

$((SP) - 2) \leftarrow (Reg\ L)$

$(SP) \leftarrow (SP) - 2$

	Reg H	Reg L
PUSH B	B	C
PUSH D	D	E
PUSH H	H	L
PUSH PSW	A	Flags



ΕΝΤΟΛΕΣ ΣΤΟΙΒΑΣ - POP

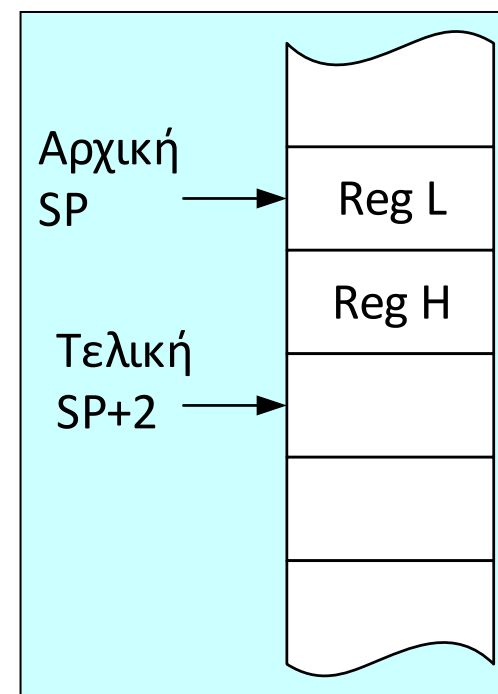
POP Reg Pair

(Reg L) \leftarrow ((SP))

(Reg H) \leftarrow ((SP) + 1)

(SP) \leftarrow (SP) + 2

	Reg H	Reg L
POP B	B	C
POP D	D	E
POP H	H	L
POP PSW	A	Flags



ΥΠΟΡΟΥΤΙΝΕΣ

- Ανάγκη να εκτελείται η ίδια σειρά εντολών σε διαφορετικά σημεία
- Γιατί όχι μακροεντολές?
 - Μεγάλη μνήμη
 - Στις μακροεντολές ο assembler πάει και βάζει κάθε φορά τις εντολές της μακροεντολής στο σημείο που την καλούμε

ΚΛΗΣΗ ΥΠΟΡΟΥΤΙΝΑΣ

CALL Address

$((SP) - 1) \leftarrow (PCH)$

$((SP) - 2) \leftarrow (PCL)$

$(SP) \leftarrow (SP) - 2$

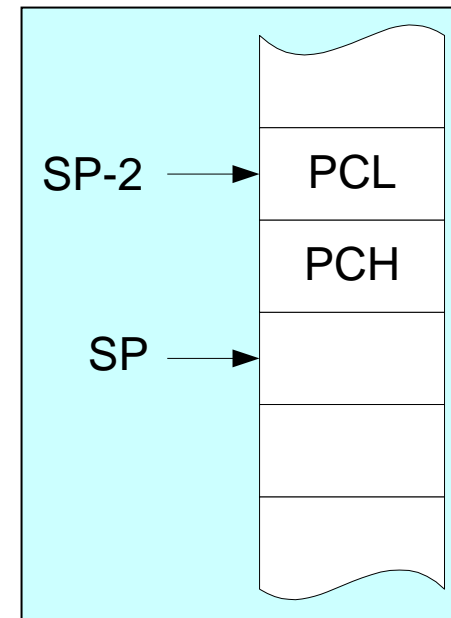
$(PC) \leftarrow \text{Address}$

RET

$(PCL) \leftarrow (SP)$

$(PCH) \leftarrow ((SP)+1)$

$(SP) \leftarrow (SP) + 2$



ΚΛΗΣΗ ΥΠΟΡΟΥΤΙΝΑΣ ΥΠΟ ΣΥΝΘΗΚΗ

Συνθήκη Address

Αν ισχύει η συνθήκη (NZ, Z, NC, C, PO, PE, P, M) τότε:

$((SP) - 1) \leftarrow (PCH)$

$((SP) - 2) \leftarrow (PCL)$

$(SP) \leftarrow (SP) - 2$

$(PC) \leftarrow \text{Address}$

ΠΑΡΑΔΕΙΓΜΑ ΚΛΗΣΗΣ ΥΠΟΡΟΥΤΙΝΩΝ

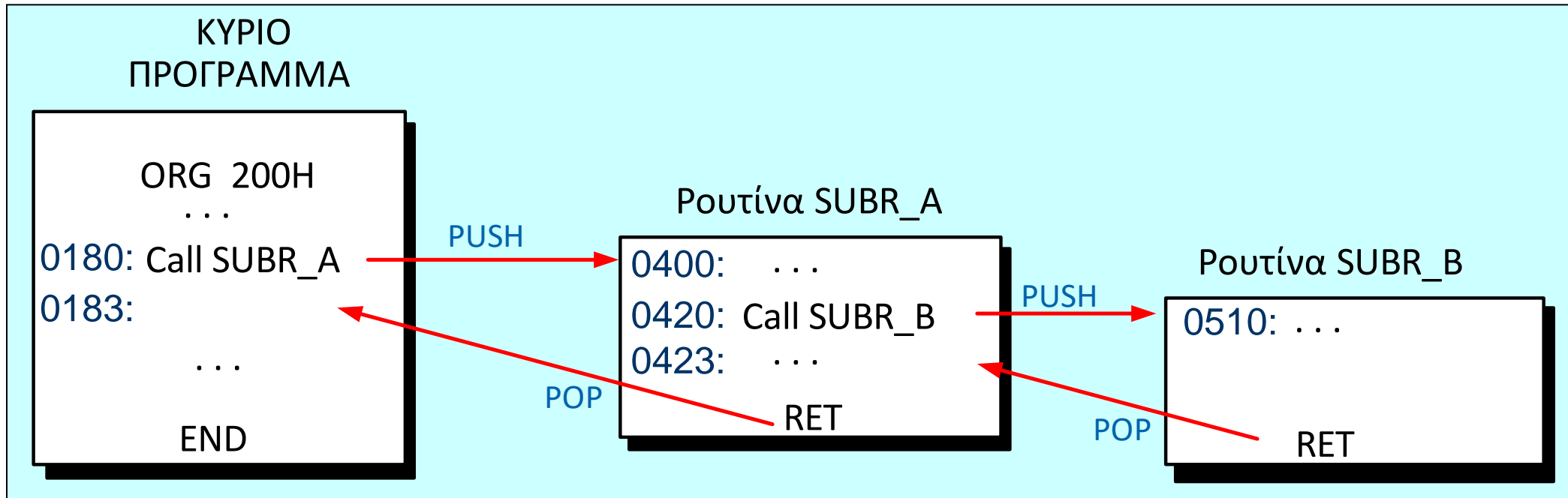
```
0000 LXI SP,610H
      Κύριο Πρόγραμμα
      .....
0180 CALL SUBR_A
0183 (επόμενη εντολή)
      Υπόλοιπο Κυρίου Προγρ.
      .....
      END

;Υπορουτίνα A
0400 SUBR_A:  PUSH H
0401          PUSH D
0402          PUSH B
0403          PUSH PSW
      Κύριο Σώμα Υπορ. A
      .....
0420          CALL SUBR_B
0423 Υπόλοιπες Εντολές Υπορ. A
      ....
```

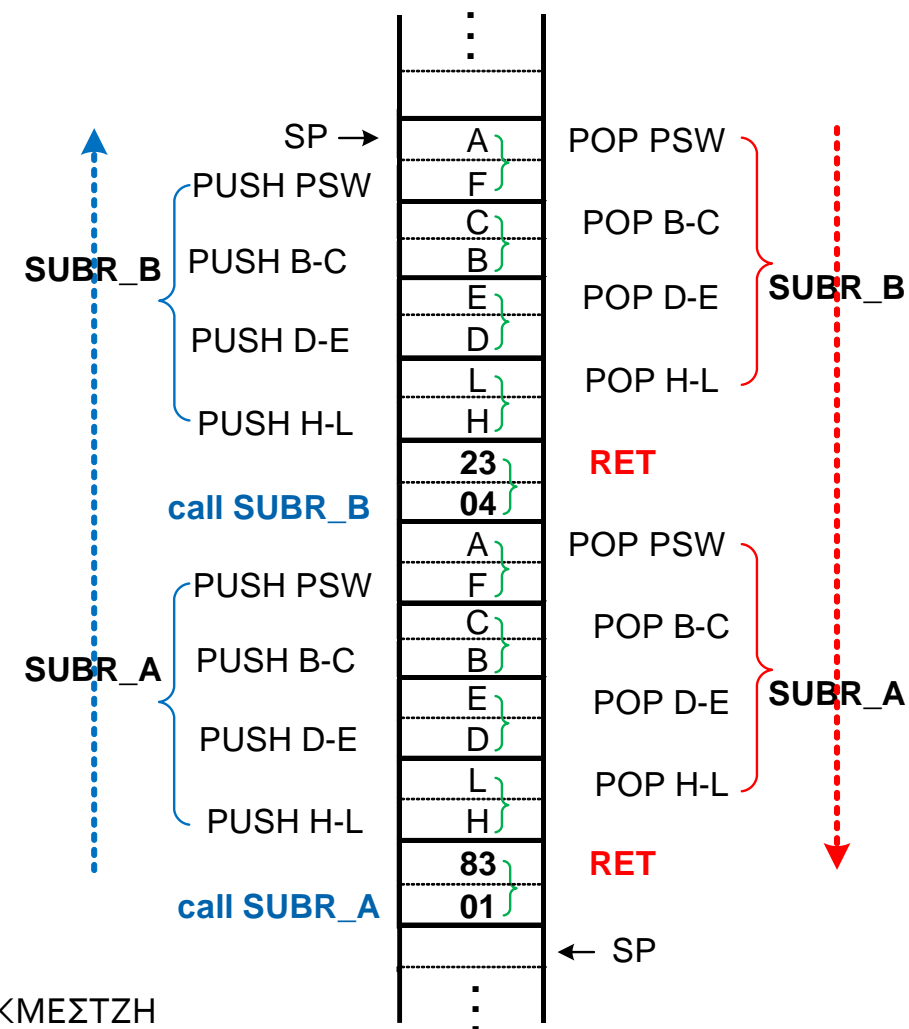
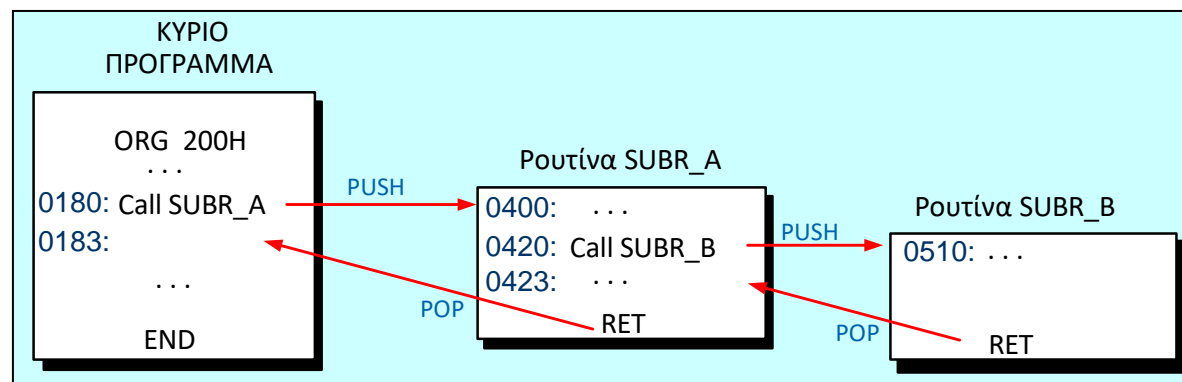
```
0436      POP PSW
0437      POP B
0438      POP D
0439      POP H
0440      RET

;Υπορουτίνα B
0510 SUBR_B:  PUSH  H
0511          PUSH  D
0512          PUSH  B
0513          PUSH  PSW
      Κύριο Σώμα Υπορ. B
      .....
0536      POP PSW
0537      POP B
0538      POP D
0539      POP H
0540      RET
```

ΠΑΡΑΔΕΙΓΜΑ ΚΛΗΣΗΣ ΥΠΟΡΟΥΤΙΝΩΝ



ΠΑΡΑΔΕΙΓΜΑ ΚΛΗΣΗΣ ΥΠΟΡΟΥΤΙΝΩΝ



ΠΑΡΑΔΕΙΓΜΑ ΚΛΗΣΗΣ ΥΠΟΡΟΥΤΙΝΩΝ - ΠΡΟΣΟΜΟΙΩΣΗ

START:

LXI SP,610H

MVI H,01H

MVI C,05H

CALL SUBR_A

HLT

SUBR_A:

PUSH H

PUSH D

PUSH B

PUSH PSW

CALL SUBR_B

POP PSW

POP B

POP D

POP H

RET

SUBR_B:

PUSH H

PUSH D

PUSH B

PUSH PSW

POP PSW

POP B

POP D

POP H

RET

ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΕΙΣ

- Σε συστήματα αυτοματισμού απαραίτητες
- Η εκτέλεση εντολών εισάγει χρονοκαθυστέρηση έτσι κι αλλιώς

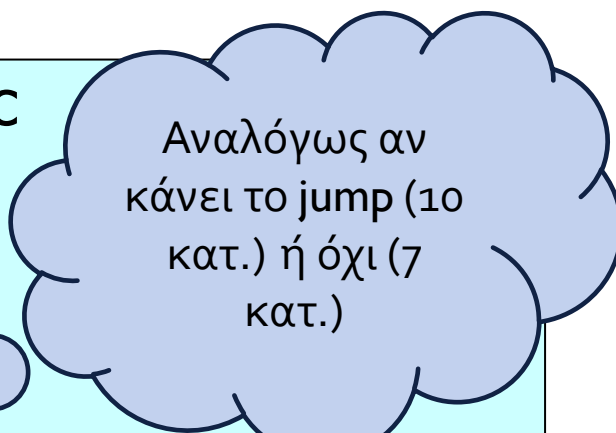
Δίνεται το παρακάτω πρόγραμμα σε assembly (8085 στα 2 MHz). Ποιος είναι ο ελάχιστος και ποιος ο μέγιστος χρόνος καθυστέρησης που μπορεί να επιτευχθεί;

```
MVI A,DELAY  
NOP  
NOP  
LOOP:  
DCR A  
JNZ LOOP
```

ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΕΙΣ – ΠΑΡΑΔΕΙΓΜΑ 1

2 MHz άρα $1/(2\text{MHz}) = 0.5 \mu\text{sec}$ χρόνος ανά κατάσταση

MVI A,DELAY ; 7 κατ. $\times 500 \text{ nsec/κατ.} = 3.5 \mu\text{sec}$
NOP ;4 κατ.
NOP ;4 κατ.
LOOP:
DCR A ; 4 κατ. $500 \text{ nsec/κατ.} = 2 \mu\text{sec}$
JNZ LOOP ; 7 κατ. $500 \text{ nsec/κατ.} = 3.5 \mu\text{sec}$ ή 10 κατ. $= 5 \mu\text{sec}$



Αναλόγως αν κάνει το jump (10 κατ.) ή όχι (7 κατ.)

Για DELAY=1 έχουμε $7+4+4+4+7 = 26$ καταστάσεις ή $26 \times 0.5 \mu\text{sec} = 13 \mu\text{sec}$

Για DELAY>1 έχουμε $7+4+4+(4+10) \times (\text{DELAY}-1) + (4+7) = 26+14 \times (\text{DELAY}-1)$

Για DELAY = 142 έχουμε $26+14 \times 141 = 2000$ κατ. $\rightarrow 1000 \mu\text{sec} = 1 \text{msec}$

ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΕΙΣ – ΠΑΡΑΔΕΙΓΜΑ 2

DELAY:

LXI B,1000D

mSEC:

<ο κώδικας του παραδείγματος 1 για DELAY=140>

NOP

DCX B

MOV A,B

ORA C

JNZ mSEC

RET

ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΕΙΣ – ΠΑΡΑΔΕΙΓΜΑ 2

DELAY:

LXI B,1000D ;10 καταστάσεις

mSEC:

<ο κώδικας του παραδείγματος 1 για DELAY=140> ;1972 κατ.

NOP ; 4 καταστάσεις

DCX B ; 6 καταστάσεις

MOV A,B ; 4 καταστάσεις

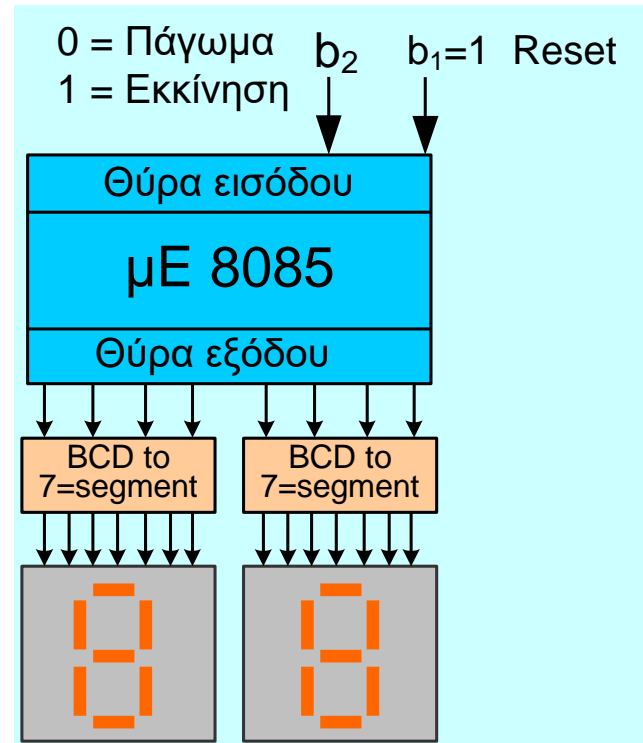
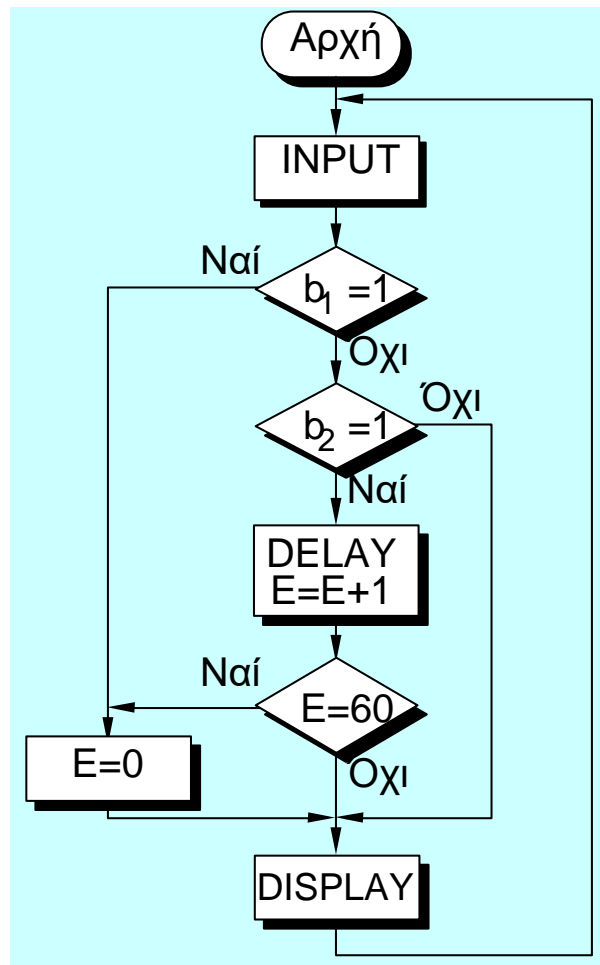
ORA C ; 4 καταστάσεις

JNZ mSEC ; 7 ή 10 καταστάσεις

RET ; 10 καταστάσεις

$$T = 10 + 1000 \times (1972 + 4 + 6 + 4 + 4 + 10) - 3 + 10 = 2000017 \text{ κατ.} \rightarrow 1000008.5 \text{ } \mu\text{sec} \approx 1 \text{ sec}$$

ΠΑΡΑΔΕΙΓΜΑ 3: ΧΡΟΝΟΜΕΤΡΟ



Πόρτα Εισόδου: 10H

Πόρτα Εξόδου: 20H

ΠΑΡΑΔΕΙΓΜΑ 3: ΧΡΟΝΟΜΕΤΡΟ - ΛΥΣΗ

```
MVI    E,00 ; Μηδενισμός της ένδειξης
INPUT:
IN      10H ; Είσοδος από την πόρτα 10H
RRC     ; Το bit 1(b1) στο κρατούμενο
JC      RESET ; Αν b1=1 το χρονόμετρο γίνεται RESET
RRC     ; Το bit 2(b2) στο κρατούμενο
JNC     DISPLAY ; Αν είναι 0 το χρονόμετρο παγώνει
          ; (Παρακάμπτεται η αύξηση του μετρητή)
CALL    DELAY ; Καθυστέρηση 1 sec
MOV     A,E
INR     A ; και αύξηση της ένδειξης
DAA     ; Προσαρμογή της ένδειξης σε BCD μορφή
MOV     E,A
CPI     60H ; Έφτασε το χρονόμετρο στο 59?
JNZ     DISPLAY ; Αν όχι συνεχίζεται η χρονομέτρηση
RESET:
MVI     E,0 ; Μηδενίζεται το χρονόμετρο
DISPLAY:
MOV     A,E
OUT     20H ; Εμφάνιση της ένδειξης στους ενδείκτες
JMP     INPUT
END
```

ΠΑΡΑΔΕΙΓΜΑ 3: ΧΡΟΝΟΜΕΤΡΟ – ΛΥΣΗ ΣΤΟΝ TSIK

- Είσοδος: Διακόπτες στην 2000h
 - B1 για reset στο 2ο LSB (στο bit 1)
 - B2 για εκκίνηση/πάγωμα στο 3ο LSB (στο bit 2)
- Έξοδος στην 7-segment display στα 2 δεξιότερα ψηφία

ΠΑΡΑΔΕΙΓΜΑ 3: ΧΡΟΝΟΜΕΤΡΟ – ΛΥΣΗ ΣΤΟΝ TSIK

<pre>MVI A,10H ;Καθαρισμός STA 0BF2H ;οθόνης STA 0BF3H STA 0BF4H STA 0BF5H MVI E,00H INPUT: LDA 2000H RRC RRC JC RESET1 RRC JNC DISPLAY</pre>	<pre>JNC DISPLAY MVI B,03H MVI C,E8H ;1000d=03E8h CALL DELB MOV A,E INR A CMC DAA MOV E,A CPI 60H JNZ DISPLAY</pre>	<pre>RESET1: MVI E,00H DISPLAY: MOV A,E ANI 0FH STA 0BF0H ;Μονάδες MOV A,E RRC RRC RRC RRC ANI 0FH STA 0BF1H ; Δεκάδες CALL DCD JMP INPUT END</pre>
---	---	---

ΧΡΟΝΟΚΑΘΥΣΤΕΡΗΣΗ ΣΤΟΥΣ ΠΡΟΣΟΜΟΙΩΤΕΣ

- TSIK:

- CALL DELB (έτοιμη ρουτίνα)

- Παίρνει σαν είσοδο στο ζεύγος B-C τον χρόνο που θέλουμε σε msec

- JUBINMITRA:

- Βοηθητικό πρόγραμμα Subroutine > Insert Delay Subroutine όπου ο χρήστης ορίζει τις παραμέτρους (πόσοι καταχωρητές πόσα msec κάθε loop) και παράγεται αυτόματα μια ρουτίνα