Για την εμπέδωση των εντολών που έχουν παρουσιαστεί θα δοθούν παραδείγματα χρήσης τους.

Παραδείγματα χρήσης αριθμητικών και εντολών μετακίνησης δεδομένων

Παράδειγμα 1ο

Αριθμός των 16 bits που βρίσκεται στις θέσεις 40 και 41 της μνήμης να προστεθεί στον αριθμό των 16 bits που βρίσκεται στις θέσεις 42 και 43 της μνήμης. Το αποτέλεσμα να τοποθετηθεί στις θέσεις 44 και 45. Τα περισσότερα σημαντικά bytes είναι στις θέσεις 41, 43 και 45. (Με δυο τρόπους).

1ος τρόπος.

```
LXI  H,40H
MOV  A,M
INX  H
INX  H     ;HL=42H
ADD  M     ;Προσθέτουμε τα λιγότερο σημαντικά byte
MOV  B,A
DCX  H     ;HL=41H
MOV  A,M
INX  H
INX  H     ;HL=43H
ADC  M     ;Προσθέτουμε τα περισσότερο σημαντικά bytes.
INX  H     ;HL=44H
MOV  M,B
INX  H     ;HL=45H
MOV  M,A   ;Αποθηκεύω το αποτέλεσμα.
HLT        ;Εντολή που σταματά την εκτέλεση εντολών από τον μΕ και
           ;τοποθετείται για να δηλώσει το τέλος του προγράμματος
```

2ος τρόπος.

```
LHLD 40H   ;Φορτώνουμε στο ζεύγος (H)(L) τα περιεχόμενα των θέσεων
           ;40H, 41H
XCHG       ;Φορτώνουμε στον (D)(E) τα περιεχόμενα του ζεύγους (H)(L)
LHLD 42H   ;(H)(L) ← M(43)M(42)
DAD  D     ;Πρόσθεση των δύο αριθμών
SHLD 44H   ;Αποθήκευση αποτελέσματος στις θέσεις 44, 45
HLT        ;Σταματώ τον μΕ.
```

Παράδειγμα 2ο

Εδώ υπολογίζεται το τετράγωνο ενός αριθμού που περιέχεται στη θέση 40H της μνήμης με τη βοήθεια ενός πίνακα. Ο αριθμός υποθέτουμε ότι ανήκει στο διάστημα [0,15]. Ο πίνακας τοποθετείται στις θέσεις 100H εως 10FH της μνήμης. Το αποτέλεσμα τοποθετείται στη θέση 41H. Τα τετράγωνα των αριθμών που αποτελούν τον πίνακα και το αντίστοιχο πρόγραμμα δίνονται παρακάτω:

```
0100H: 00H      0108H:40H
0101H: 01H      0109H:51H
0102H: 04H      010AH:64H
0103H: 09H      010BH:79H
0104H: 10H      010CH:90H
0105H: 19H      010DH:A9H
0106H: 24H      010EH:C4H
0107H: 31H      010FH:E1H
```

```
LDA  40H        ;Φορτώνεται ο αριθμός στον καταχωρητή A
MOV  L,A        ;Προστίθεται στην αρχική διεύθυνσης του πίνακα η
MVI  H,0        ; σχετική απόσταση που καθορίζεται από τον αριθμό,
LXI  D,100H     ;για δημιουργία της διεύθυνσης του προενταμιευμένου
DAD  D          ; αποτελέσματος στον πίνακα των τετραγώνων.
MOV  A,M        ;Μεταφέρεται στον καταχ. A το αντίστοιχο τετράγωνο
STA  41H        ;Αποθήκευση του αποτελέσματος στη θέση 41H
HLT
```

Παράδειγμα 3ο

Ένας διψήφιος BCD αριθμός είναι αποθηκευμένος στις θέσεις 40 και 41 της μνήμης, με το περισσότερο σημαντικό ψηφίο (MSD - Most Significant Digit) στη θέση 40. Ο αριθμός αυτός να μετατραπεί σε δυαδικό και να αποθηκευτεί στη θέση 42. Για να μετατρέψουμε ένα διψήφιο αριθμό BCD σε δυαδικό δεκαπλασιάζουμε το MSD και μετά προσθέτουμε το LSD (Least Significant Digit).

```
LXI  H,40H      ;Φορτώνεται ο διπλός καταχωρητής HL με 0040H
MOV  A,M        ;(A) ← MSD
ADD  A          ;(A) ← 2×MSD
MOV  B,A        ;(B) ← (A)=2×MSD
ADD  A          ;(A) ← 2×(A)=4×MSD
ADD  A          ;(A) ← 2×(A)=8×MSD
ADD  B          ;(A) ← (A)+(B)=8×MSD+2×MSD=10×MSD
INX  H
```

```
ADD  M          ;(A) ← 10×MSD+LSD
INX  H
MOV  M,A        ;Αποθήκευση του αποτελέσματος
HLT
```

2.6.3. Εντολές Λογικών πράξεων

Αυτό το σύνολο εντολών υλοποιεί λογικές πράξεις σε δεδομένα που βρίσκονται στους καταχωρητές, στη μνήμη και σε σημαίες κατάστασης. Οι λογικές πράξεις που ακολουθούν γίνονται μεταξύ του καταχωρητή A και ενός καταχωρητή , αριθμού ή θέσης μνήμης των 8 bits. Η λογική πράξη επιτελείται μεταξύ ίδιας τάξεως δυαδικών ψηφίων. Ετσι με μια εντολή έχουμε ταυτόχρονη εκτέλεση 8 λογικών πράξεων και το αποτέλεσμα επιστρέφεται στον καταχωρητή A.

Εκτός αν αναφέρεται διαφορετικά, όλες οι εντολές αυτού του συνόλου επηρεάζουν τις σημαίες.

α) Λογικές εντολές AND του καταχωρητή A με καταχωρητές, μνήμη και άμεσα με ένα δεδομένο.

ANA r (μήκους ενός byte)

(A) ← (A) AND (r)

Πραγματοποιείται η λογική πράξη "ΚΑΙ" μεταξύ του περιεχομένου του καταχωρητή r (ένας οποιοσδήποτε από τους A, B, C, D, E, H, L) και του περιεχομένου του καταχωρητή A. Το αποτέλεσμα τοποθετείται στον A.

ANA M (μήκους ενός byte)

(A) ← (A) AND ((H)(L))

Πραγματοποιείται η λογική πράξη "ΚΑΙ" μεταξύ του περιεχομένου της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στο ζευγάρι των καταχωρητών H και L, και του περιεχομένου του καταχωρητή A. Ο καταχωρητής H περιέχει το περισσότερο σημαντικό byte της διεύθυνσης και ο L το λιγότερο. Το αποτέλεσμα τοποθετείται στον A.

ANI byte (μήκους δύο bytes)

(A) ← (A) AND byte

Πραγματοποιείται η λογική πράξη "ΚΑΙ" μεταξύ του byte (που αποτελεί το δεύτερο byte της όλης εντολής) και του περιεχομένου του καταχωρητή A. Το αποτέλεσμα τοποθετείται στον A.

β) Λογικές εντολές XOR του καταχωρητή Α με καταχωρητές, μνήμη και άμεσα με ένα δεδομένο.

XRA κ (μήκους ενός byte)

(A) ← (A) XOR (κ)

Πραγματοποιείται η λογική πράξη "αποκλειστικό-Η" μεταξύ του περιεχομένου του καταχωρητή κ (ένας οποιοσδήποτε από τους A, B, C, D, E, H, L) και του περιεχομένου του καταχωρητή A. Το αποτέλεσμα τοποθετείται στον A.

XRA M (μήκους ενός byte)

(A) ← (A) XOR ((H)(L))

Πραγματοποιείται η λογική πράξη "αποκλειστικό-Η" μεταξύ του περιεχομένου της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στο ζευγάρι των καταχωρητών H και L, και του περιεχομένου του καταχωρητή A. Ο καταχωρητής H περιέχει το περισσότερο σημαντικό byte της διεύθυνσης και ο L το λιγότερο. Το αποτέλεσμα τοποθετείται στον A.

XRI byte (μήκους δύο bytes)

(A) ← (A) XOR byte

Πραγματοποιείται η λογική πράξη "αποκλειστικό-Η" μεταξύ του byte (που αποτελεί το δεύτερο byte της όλης εντολής) και του περιεχομένου του καταχωρητή A. Το αποτέλεσμα τοποθετείται στον καταχωρητή A.

γ) Λογικές εντολές OR του καταχωρητή Α με καταχωρητές, μνήμη και άμεσα με ένα δεδομένο.

ORA κ (μήκους ενός byte)

(A) ← (A) OR (κ)

Πραγματοποιείται η λογική πράξη "Η" μεταξύ του περιεχομένου του καταχωρητή κ (ένας οποιοσδήποτε από τους A, B, C, D, E, H, L) και του περιεχομένου του καταχωρητή A. Το αποτέλεσμα τοποθετείται στον A. Η σημαία κρατούμενου (CY) και η σημαία βοηθητικού κρατούμενου (AC) γίνονται μηδέν.

ORA M (μήκους ενός byte)

(A) ← (A) OR ((H)(L))

Πραγματοποιείται η λογική πράξη "Η" μεταξύ του περιεχομένου της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στο ζευγάρι των καταχωρητών H και L, και του περιεχομένου του καταχωρητή A. Ο καταχωρητής H περιέχει το περισσότερο σημαντικό byte της διεύθυνσης και ο L το λιγότερο. Το αποτέλεσμα τοποθετείται στον A.

Η σημαία κρατούμενου (CY) και η σημαία βοηθητικού κρατούμενου (AC) γίνονται μηδέν.

`ORI byte` (μήκους δύο bytes)

(A) ← (A) OR byte

Πραγματοποιείται η λογική πράξη "Η" μεταξύ του byte (που αποτελεί το δεύτερο byte της όλης εντολής) και του περιεχομένου του καταχωρητή Α. Το αποτέλεσμα τοποθετείται στον Α. Η σημαία κρατούμενου (CY) και η σημαία βοηθητικού κρατούμενου (AC) γίνονται μηδέν.

δ) Εντολές σύγκρισης με καταχωρητή, μνήμη και άμεσα με ένα δεδομένο.

`CMP κ` (μήκους ενός byte)

(A) - (κ)

Το περιεχόμενο του καταχωρητή κ (ένας οποιοσδήποτε από τους A, B, C, D, E, H, L) αφαιρείται από τον καταχωρητή Α. Ο καταχωρητής Α μένει ανεπηρέαστος. Οι σημαίες κατάστασης επηρεάζονται σύμφωνα με το αποτέλεσμα της αφαίρεσης. Η σημαία μηδενισμού (Z) γίνεται ένα εάν (A)=(κ). Η σημαία κρατούμενου (CY) γίνεται ένα εάν (A)<(κ).

`CMP M` (μήκους ενός byte)

(A) - ((H)(L))

Το περιεχόμενο της θέσης μνήμης της οποίας η διεύθυνση περιέχεται στο ζευγάρι των καταχωρητών Η και L, αφαιρείται από το περιεχόμενο του καταχωρητή Α. Ο καταχωρητής Η περιέχει το περισσότερο σημαντικό byte της διεύθυνσης και ο L το λιγότερο. Ο καταχωρητής Α μένει ανεπηρέαστος. Οι σημαίες κατάστασης επηρεάζονται σύμφωνα με το αποτέλεσμα της αφαίρεσης. Η σημαία μηδενισμού (Z) γίνεται ένα εάν (A)=((H)(L)). Η σημαία κρατούμενου (CY) γίνεται ένα εάν (A) < ((H)(L)).

`CPI byte` (μήκους ενός byte)

(A) - byte

Το byte (που αποτελεί το δεύτερο byte της όλης εντολής) αφαιρείται από τον καταχωρητή Α. Ο καταχωρητής Α μένει ανεπηρέαστος. Οι σημαίες κατάστασης επηρεάζονται σύμφωνα με το αποτέλεσμα της αφαίρεσης. Η σημαία μηδενισμού (Z) γίνεται ένα εάν (A) = byte. Η σημαία κρατούμενου (CY) γίνεται ένα εάν (A)<byte.

ε) Εντολές περιστροφής του συσσωρευτή.

`RLC` (μήκους ενός byte)

(An+1) ← (An)

(A0) ← (A7)
(CY) ← (A7)
Το περιεχόμενο του καταχωρητή Α περιστρέφεται αριστερά κατά μία θέση. Το λιγότερο σημαντικό bit A0 και η σημαία κρατούμενου (CY) παίρνουν την τιμή του περισσότερο σημαντικού bit A7. Μόνο η σημαία κρατούμενου επηρεάζεται.

RRC (μήκους ενός byte)
(An) ← (An+1)
(A7) ← (A0)
(CY) ← (A0)
Το περιεχόμενο του καταχωρητή Α περιστρέφεται δεξιά κατά μία θέση. Το περισσότερο σημαντικό bit A7 και η σημαία κρατούμενου (CY) παίρνουν την τιμή του λιγότερο σημαντικού bit A0. Μόνο η σημαία κρατούμενου επηρεάζεται.

RAL (μήκους ενός byte)
(An+1) ← (An)
(CY) ← (A7)
(A0) ← (CY)
Το περιεχόμενο του καταχωρητή Α περιστρέφεται αριστερά κατά μία θέση μέσω της σημαίας κρατούμενου. Το λιγότερο σημαντικό bit A0 παίρνει την τιμή της σημαίας κρατούμενου (CY) και η σημαία κρατούμενου (CY) παίρνει την τιμή του περισσότερο σημαντικού bit A7. Μόνο η σημαία κρατούμενου επηρεάζεται.

RAR (μήκους ενός byte)
(An) ← (An+1)
(CY) ← (A0)
(A7) ← (CY)
Το περιεχόμενο του καταχωρητή Α περιστρέφεται δεξιά κατά μία θέση μέσω της σημαίας κρατούμενου. Το περισσότερο σημαντικό bit A7 παίρνει την τιμή της σημαίας κρατούμενου (CY) και η σημαία κρατούμενου (CY) παίρνει την τιμή του λιγότερο σημαντικού bit A0. Μόνο η σημαία κρατούμενου επηρεάζεται.

στ) Εντολή συμπληρώματος συσσωρευτή.

CMA (μήκους ενός byte)
(A) ← ($\overline{A}$)
Το περιεχόμενο του καταχωρητή Α συμπληρώνεται (τα bits 0 γίνονται 1 και αντίστροφα). Καμιά σημαία δεν επηρεάζεται.

ζ) Εντολή αναστροφής και τοποθέτησης κρατουμένου.

`CMC` (μήκους ενός byte)

$(CY) \leftarrow (\overline{CY})$

Η σημαία κρατούμενου (CY) συμπληρώνεται (τα bits 0 γίνονται 1 και αντίστροφα). Καμιά άλλη σημαία δεν επηρεάζεται.

`STC` (μήκους ενός byte)

$(CY) \leftarrow 1$

Η σημαία κρατούμενου (CY) γίνεται 1. Καμιά άλλη σημαία δεν επηρεάζεται.

Παραδείγματα χρήσης λογικών εντολών

Παράδειγμα 1ο.

Τα 4 περισσότερο σημαντικά και τα 4 λιγότερα σημαντικά bits του περιεχομένου της θέσης 40 της μνήμης τοποθετούνται στα 4 λιγότερο σημαντικά bits των θέσεων 41 και 42 αντίστοιχα. Τα 4 περισσότερο σημαντικά bits των θέσεων 41 και 42 μηδενίζονται.

```
    LXI  H,40H      ;Φορτώνουμε στο ζεύγος (H)(L) τον αριθμό 40H
    MOV  A,M        ;(A) ← M((H)(L)) = M(40)
    MOV  B,A        ;(B) ← (A)
    RRC             ;Τα 4 περισσότερο σημαντικά ψηφία του
    RRC             ;αρχικού αριθμού είναι τώρα τα τέσσερα λιγότερο
    RRC             ;σημαντικά ψηφία του A.
    RRC
    ANI  0FH        ;Μηδενισμός των 4 MSBits του A
    INX  H          ;Το ζεύγος (H)(L) δείχνει τη διεύθυνση 41H
    MOV  M,A        ;M((H)(L)) ← (A)
    MOV  A,B        ;(A) ← (B) = M(40)
    ANI  0FH        ;Κρατάμε στον A τα 4 LSBits του αρχικού αριθμού
    INX  H          ;και τα αποθηκεύουμε στην διεύθυνση
    MOV  M,A        ;μνήμης 42H
    HLT
```

Παράδειγμα 2ο.

Στο πρόγραμμα αυτό υπολογίζεται το συμπλήρωμα ως προς 2 αριθμού που βρίσκεται στη θέση 40 και να τοποθετείται στη θέση 41.

```
    LXI  H,40H
    MVI  B,01H
    MOV  A,M        ;Φορτώνουμε στον A τον αριθμό
    CMA             ;συμπλήρωμα ως προς 1 του αριθμού
    ADD  B          ;Προσθέτοντας 1 έχουμε στον A το συμπλήρωμα
    INX  H          ;ως προς δύο
```

```
MOV  M,A        ;Αποθήκευση του αποτελέσματος στη θέση 41
HLT
```

Το συμπλήρωμα ως προς 2 ενός αριθμού θα μπορούσε να υλοποιηθεί απλούστερα με αφαίρεση από το μηδέν του αριθμού αυτού.

2.6.4. Εντολές Αλματος.

Αυτό το σύνολο εντολών αλλάζει την κανονική ακολουθιακή ροή ενός προγράμματος γι'αυτό και οι εντολές αυτές ονομάζονται και εντολές διακλάδωσης. Έτσι όταν εκτελείται μια εντολή άλματος, αντί να εκτελεστεί η επόμενη εντολή, ο έλεγχος μεταφέρεται σε άλλο σημείο του προγράμματος. Το σημείο αυτό μπορεί να είναι οπουδήποτε στο πρόγραμμα, σε κάποια διεύθυνση που καθορίζεται από την εντολή άλματος. Υπάρχουν εντολές άλματος υπό και χωρίς συνθήκη. Οι εντολές άλματος χωρίς συνθήκη απλά μεταφέρουν την καθορισμένη διεύθυνση στον καταχωρητή PC. Οι εντολές υπό συνθήκη εξετάζουν την κατάσταση μιας από τις τέσσερις σημαίες για να καθορίσουν αν πρέπει να μεταφερθεί ή όχι ο έλεγχος. Οι συνθήκες που πρέπει να καθοριστούν είναι οι εξής:

Συνθήκη	Ερμηνεία
NZ	- όχι μηδέν (Z = 0)
Z	- μηδέν (Z = 1)
NC	- όχι κρατούμενο (CY = 0)
C	- κρατούμενο (CY = 1)
PO	- μονή ισοτιμία (P = 0)
PE	- ζυγή ισοτιμία (P = 1)
P	- θετικό (S = 0)
M	- αρνητικό (S = 1)

`JMP διεύθυνση` (μήκους 3 bytes)

(PC) ← byte3 byte2

Ο έλεγχος του προγράμματος μεταφέρεται στην εντολή της οποίας η διεύθυνση καθορίζεται από το τρίτο και δεύτερο byte της παρούσας εντολής. Το τρίτο byte είναι το περισσότερο σημαντικό byte της διεύθυνσης και το δεύτερο το λιγότερο.

`Jσυνθήκη διεύθυνση` (μήκους τριών bytes)

Εάν ισχύει η συνθήκη τότε (PC) ← byte3 byte2

Εάν ισχύει η συνθήκη της εντολής τότε ο έλεγχος του προγράμματος μεταφέρεται στην εντολή της οποίας η διεύθυνση καθορίζεται από το τρίτο και δεύτερο byte της παρούσας εντολής, αλλιώς το πρόγραμμα συνεχίζεται με την εκτέλεση της επόμενης εντολής

κανονικά. Το τρίτο byte είναι το περισσότερο σημαντικό byte της διεύθυνσης και το δεύτερο το λιγότερο. Οι σημαίες συνθηκών δεν επηρεάζονται.

Οι υπόλοιπες εντολές διακλάδωσης **CALL, RET** και **RST** πρόκειται να παρουσιαστούν μαζί με παραδείγματα στο κεφάλαιο 5.

Παράδειγμα 1ο

Τα περιεχόμενα των θέσεων μνήμης 40 και 41 είναι απλοί δυαδικοί αριθμοί (χωρίς πρόσημο). Προσδιορίζεται ο μεγαλύτερος από τους δύο αριθμούς και τοποθετείται στη θέση 42.

```
    LXI  H,40H
    MOV  A,M
    INX  H
    CMP  M          ;Συγκρίνονται οι δυο αριθμοί
    JNC  DONE       ;Αλμα αν ο πρώτος είναι μεγαλύτερος
    MOV  A,M        ;διαφορετικά προετοιμάζεται ο δεύτερος
DONE:
    INX  H          ;Το ζεύγος (H)(L) δείχνει τώρα στη 42H
    MOV  M,A        ;Ο μεγαλύτερος στέλνεται στη 42H
    HLT
```

Παράδειγμα 2ο

Στο πρόγραμμα αυτό προσδιορίζεται τό μεγαλύτερο ενός συνόλου αριθμών. Το πλήθος των αριθμών βρίσκεται στη θέση 41 της μνήμης και οι αριθμοί αρχίζουν από την θέση 42. Το αποτέλεσμα καταχωρείται στη θέση 40.

```
    LXI  H,41H
    MOV  B,M        ;(B) = πλήθος αριθμών
    SUB  A          ;Μηδενίζεται ο A
FOR:
    INX  H
    CMP  M          ;Είναι ο επόμενος αριθμός μεγαλύτερος του A
    JNC  NEXT       ;δηλαδή του τοπικού μεγίστου.
    MOV  A,M        ;Αν ναι, αντικαθιστάται ο μέγιστος
NEXT:
    DCR  B          ;Διαφορετικά προχωράμε στον έλεγχο του
    JNZ  FOR        ;επομένου αριθμού
    STA  40H
    HLT
```

Παράδειγμα 3ο

Προσδιορίζεται το μήκος ενός συνόλου χαρακτήρων ASCII (string) όπου κάθε χαρακτήρας αντιστοιχεί σ' έναν κωδικό του ενός byte. Το string είναι αποθηκευμένο στη μνήμη, απο την θέση 41 και μετά. Το τέλος του προσδιορίζεται από ένα χαρακτήρα "CR" (Carriage Return, κωδικός: 0DH). Αφού βρεθεί το ζητούμενο μήκος (χωρίς το "CR") τοποθετείται στη θέση 40.

```
     LXI  H,41H
     MVI  B,0        ;(B)=μηκος=0
     MVI  A,0DH      ;Φορτώνεται ο A με τον κωδικό "CR"
FOR:
     CMP  M          ;Είναι ο χαρακτήρας "CR";
     JZ   DONE       ;Αν ναι τέλος
     INR  B          ;Διαφορετικά το μήκος αυξάνεται
     INX  H          ;κατά ένα
     JMP  FOR        ;Ελεγχος επόμενου χαρακτήρα
DONE:
     MOV  A,B
     STA  40H        ;Αποθήκευση του μήκους
     HLT
```

Παράδειγμα 4ο

Στο παρακάτω πρόγραμμα δυαδικός αριθμός που βρίσκεται στη θέση 42H όταν είναι μικρότερος του 100_{10} μετατρέπεται σε δεκαδικό και τα δύο ψηφία αποθηκεύονται στις θέσεις 40H και 41H.

Το πρόγραμμα βασίζεται σε διαδοχικές αφαιρέσεις του 10 μέχρι να εμφανιστεί αρνητικός αριθμός. Τότε στο υπόλοιπο προσθέτω το 10 και βγάζω τις μονάδες, ενώ ταυτόχρονα καταμετρώ τις δεκάδες.

```
     LDA  42H
     CPI  63H        ;Είναι μεγαλύτερος του 99;
     JNC  END        ;Αν ναι τέλος
     MVI  B,FFH
DECA:
     INR  B
     SUI  0AH        ;Συνεχής αφαίρεση του 10
     JNC  DECA       ;Αν είναι θετικός συνέχισε
     ADI  0AH        ;Διόρθωσε το αρνητικό υπόλοιπο
     STA  40H        ;Αποθήκευσε τις μονάδες
     MOV  A,B
     STA  41H        ;Αποθήκευσε τις δεκάδες
END:
     HLT
```

Διαγράμματα Ροής

Συχνά η σχεδίαση του αλγορίθμου απαιτεί διάγραμμα ροής όπως κάνουμε και στις γλώσσες ανωτέρου επιπέδου (BASIC, FORTRAN).

Εδώ υπενθυμίζουμε τα βασικά γραφικά σύμβολα με τα οποία κάθε διαφορετική λειτουργία του προγράμματος παριστάνεται. Ένα γενικό διάγραμμα ροής φαίνεται στο σχήμα 2.14.

α) **Αρχή:** μακρόστενο ορθογώνιο με στρογγυλεμένα άκρα.
β) **Επεξεργασία** (πράξεις): ορθογώνιο
γ) **Είσοδος-έξοδος**: παραλληλόγραμμο
δ) **Απόφαση**: ρόμβος
ε) **Υπορουτίνα**: ορθογώνιο με διπλές πλάγιες γραμμές
στ) **Τέλος**: μακρόστενο ορθογώνιο με στρογγυλεμένα άκρα.

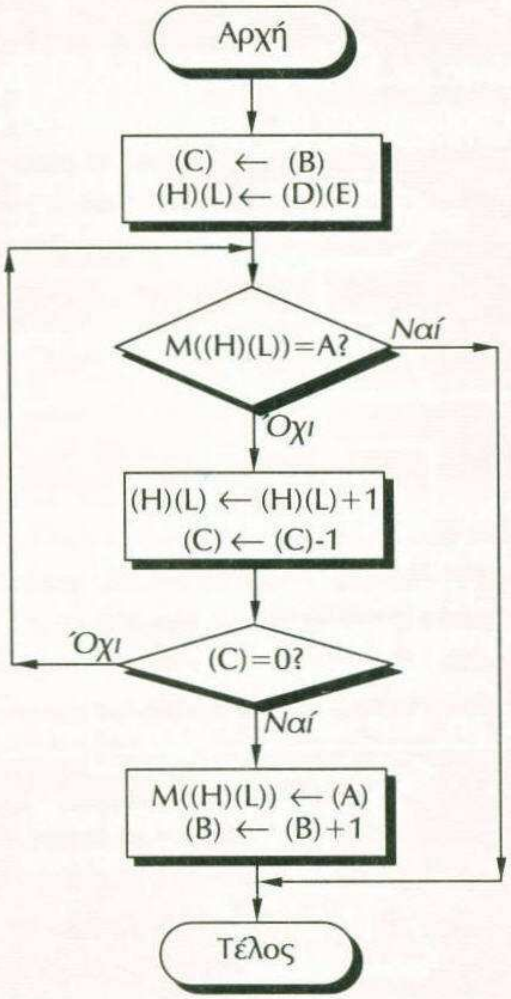
Σχήμα 2.14 Ενα γενικό διάγραμμα ροής

Τα δύο παραδείγματα που ακολουθούν συνοδεύονται από διαγράμματα ροής για την καλύτερη κατανόηση των προγραμμάτων.

Παράδειγμα 5ο

Δίνεται ένα block N δεδομένων στη μνήμη. Η πρώτη διεύθυνση βρίσκεται στον καταχωρητή D-E και το πλήθος N στον καταχωρητή B. Συγκρίνεται το περιεχόμενο του καταχωρητή A με το block των δεδομένων και αν δεν περιλαμβάνεται προστίθεται στο τέλος του block με αντίστοιχη ενημέρωση του πλήθους N.

Το διάγραμμα ροής δίνεται στο σχήμα 2.15 ενώ αμέσως παρακάτω παρατίθεται το αντίστοιχο πρόγραμμα.



Σχήμα 2.15 Διάγραμμα ροής του 5ου παραδείγματος

```
     MOV  C,B        ;Αποθηκεύω το πλήθος στον καταχωρητή C.
     MOV  H,D        ;Το περιεχόμενο του D-E
     MOV  L,E        ;μεταφέρεται στο HL
ADR1:
     CMP  M          ;Συγκρίνεται ο A με τις τιμές του πίνακα
     JZ   ADR2       ;Αν περιλαμβάνεται η διαδικασία περατώνεται
```

```
    INX  H          ;Αλλιώς...
    DCR  C
    JNZ  ADR1       ;...σαρώνουμε όλον τον πίνακα
    MOV  M,A        ;Δεν περιλαμβάνεται άρα θα προστεθεί στο τέλος
    INR  B          ;Ενημερώνω το πλήθος Ν.
ADR2:
    HLT
```

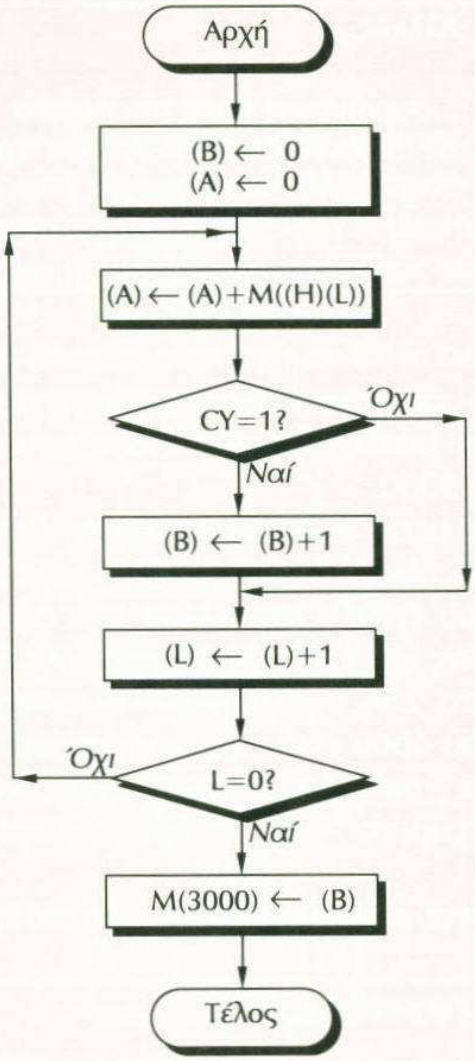
Παράδειγμα 6ο

Δίνονται 256 μη αρνητικοί αριθμοί στην περιοχή 2000-20FF της μνήμης. Υπολογίζεται ο μέσος όρος των αριθμών με ακρίβεια 8 bits και καταχωρείται στη θέση 3000.

Υπολογίζουμε τον μέσο όρο 256 αριθμών, δηλαδή το άθροισμα τους το διαιρούμε με το 256. Διαίρεση με το 256 σημαίνει ολίσθηση 8 θέσεις δεξιά. Με τον τρόπο που έγινε η πρόσθεση των αριθμών ο καταχωρητής Β έχει τα υψηλότερης αξίας bits ενώ ο καταχωρητής Α τα χαμηλότερης αξίας bits. Ετσι διαίρεση με το 256 σημαίνει ότι ο καταχωρητής Β περιέχει το ακέραιο μέρος του μέσου όρου ενώ ο Α το κλασματικό που αγνοείται. Αρκεί λοιπόν να αθροίσουμε τα κρατούμενα στον καταχωρητή Β και να αγνοήσουμε το περιεχόμενο του Α και έχουμε το μέσο όρο στον καταχωρητή Β.

Το διάγραμμα ροής δίνεται στο σχήμα 2.16. Ακολουθεί το πρόγραμμα:

```
    LXI  H,2000H    ;Αρχικοποιήσεις
    MVI  B,00H
    MVI  A,00H
ADR1:
    ADD  M          ;Άθροιση στον καταχωρητή Α
    JNC  ADR2
    INR  B          ;Οι υπερχειλίσεις καταμετρούνται στον καταχωρητή Β
ADR2:
    INR  L          ;Επόμενος αριθμός
    JNZ  ADR1       ;Αν συμπληρωθούν 256 αθροίσεις τέλος
    LXI  H,3000H    ;Αποθήκευση αποτελέσματος
    MOV  M,B
    HLT
```



Σχήμα 2.16 Διάγραμμα ροής του 6ου παραδείγματος

2.6.5. Εντολές Σωρού, Ε/Ε και Ελέγχου μΕ

Ομοίως και αυτές οι εντολές θα παρουσιαστούν με παραδείγματα στα αντίστοιχα κεφάλαια. Εδώ θα αρκεστούμε στην αναφορά μόνο των δύο επόμενων εντολών.

`NOP`

Δεν εκτελείται καμία λειτουργία. Οι καταχωρητές και οι σημαίες μένουν ανεπηρέαστες. Χρησιμοποιείται στις τεχνητές καθυστερήσεις που δημιουργούνται μέσα στο πρόγραμμα.

`HLT`

Ο επεξεργαστής σταματά. Οι καταχωρητές και οι σημαίες μένουν ανεπηρέαστες. Τυπική εντολή για τερματισμό προγραμμάτων.

Ασκήσεις προς λύση

2.1 Υποθέστε ότι πριν εκτελεστούν οι ακόλουθες εντολές ο καταχωρητής A περιέχει 4AH, ο καταχωρητής B περιέχει 8CH και ο καταχωρητής F περιέχει 00X0X0X1B. Βρείτε το περιεχόμενο των καταχωρητών A, B και F σε δυαδική μορφή αμέσως μετά την εκτέλεση κάθε εντολής. Χρησιμοποιήστε το X για να παραστήσετε τα απροσδιόριστα bit του F.

```
MVI  B,24H
ANA  B
RLC
INR  A
```

2.2 Γράψτε ένα πρόγραμμα που βρίσκει το μικρότερο αριθμό σε μια ομάδα δεδομένων (bytes) το μήκος της οποίας βρίσκεται αποθηκευμένο στη θέση 2001H και που ξεκινά απο τη θέση 2002H. Οι αριθμοί είναι 8-bit δυαδικοί χωρίς πρόσημο. Αποθηκεύστε το μικρότερο αριθμό στη θέση 2000H.

2.3 Υλοποιήστε την μετατροπή δυαδικού σε BCD ταχύτερα με βάση τη σχέση:

$$X = 16 \times MSB + LSB = 10 \times MSB + LSB + 6 \times MSB$$

Μετά από κάθε πράξη να γίνεται χρήση της εντολής **DAA** που διορθώνει τις δυαδικές πράξεις σε δεκαδικές.

2.4 Δίνονται 256 8ψήφιοι μη αρνητικοί αριθμοί αποθηκευμένοι στις διευθύνσεις 2000 - 20FF. Υπολογίστε το πλήθος των μηδενικών αριθμών του πίνακα και φυλάξτε το αποτέλεσμα στον καταχωρητή B. Επίσης αντιγράψτε τον πίνακα στην περιοχή διευθύνσεων 3000-30FF με συμπληρωμένες όμως τις μηδενικές τιμές του από το ημιάθροισμα της προηγούμενης και επόμενης κάθε φορά τιμής (ανεξάρτητα αν αυτές είναι ή όχι μηδενικές). Αν χρειαστεί υπολογισμός γιά την πρώτη και τελευταία τιμή του πίνακα (δηλ. έχουμε μηδενικές τιμές), να ληφθεί το μισό της επόμενης (2ης) ή το μισό της προηγούμενης (255ης) τιμής αντίστοιχα. Το πρόγραμμα να γραφεί στή συμβολική γλώσσα του μΕ 8085 και να συνοδεύεται απαραίτητα από σχόλια.

2.5 Στη μνήμη ενός μΥ-Σ 8085 και μεταξύ των διευθύνσεων 1000H-2000H βρίσκονται δεδομένα (8 bits). Ταυτόχρονα υπάρχει ένας πίνακας 256 θέσεων στη περιοχή 0100H-01FFH της μνήμης. Να γραφεί πρόγραμμα που να τοποθετεί τη συχνότητα εμφάνισης στον πίνακα των 256 θέσεων ως εξής: Αν το δεδομένο είναι το xy HEX τη συχνότητα του να την τοποθετεί

στη θέση 01xy HEX της μνήμης. Αν η συχνότητα είναι μεγαλύτερη από 255 να παραμένει στη τιμή αυτή.

2.6 Να γραφεί ρουτίνα που να υπολογίζει το άθροισμα N δεδομένων (με $N<2^8$ και τα δεδομένα ακέραιοι θετικοί των 8 bits) που βρίσκονται σε διαδοχικές θέσεις της μνήμης. Τα δεδομένα αρχίζουν από την διεύθυνση ADDR.

2.7 Υπολογίστε τον αριθμό των μονάδων (δυαδικών ψηφίων 1) που βρίσκονται στην περιοχή μνήμης που αρχίζει από τη διεύθυνση 1000H και έχει μήκος N που δίνεται στον καταχωρητή B. Το αποτέλεσμα να καταχωρηθεί στον D-E.

2.8 Εστω δύο σειρές χαρακτήρων ASCII ίσου μήκους N που αρχίζουν από τις διευθύνσεις 100 και 200H. Το N δίνεται στον καταχωρητή B. Να δημιουργηθεί:
(α) η ένωση U
(β) η τομή Λ και
(γ) το σύνολο που δεν περιέχει τους κοινούς χαρακτήρες των δύο γραμματοσυνόλων. Τα νέα σύνολα να αρχίζουν από τις διευθύνσεις 300, 400, 500H αντίστοιχα.

2.9 Εξηγήστε πώς πρέπει να προσθέσετε τους δεκαδικούς αριθμούς:
(α) 45 + 37 και (β) 89 + 99 στον μΕ 8085.

2.10 Δίνεται πίνακας N διαδοχικών χαρακτήρων που αρχίζουν από τη διεύθυνση 100. Το N βρίσκεται στον καταχωρητή B. Δίνεται επίσης ένας επιπλέον χαρακτήρας στον καταχωρητή C. Να γραφεί πρόγραμμα που να αφαιρεί τον χαρακτήρα αυτόν από τον πίνακα εφόσον το βρεί και να επεναφέρει τα στοιχεία του πίνακα σε διαδοχικές θέσεις της μνήμης.

2.11 Να επεκταθεί το 4ο παράδειγμα που αφορά μετατροπή δυαδικού των 8 bits σε δεκαδική μορφή χωρίς τον περιορισμό να είναι μικρότεροι του 100_{10}.

2.12 Να γραφεί πρόγραμμα που να υπολογίζει το συμπλήρωμα ως προς 2 αριθμού 16-bit, που βρίσκεται στις θέσεις 400H και 401H (MSD) και να το αποθηκεύει στις θέσεις 402H και 403H (MSD).