

Συστήματα Μικροϋπολογιστών

Παραδείγματα
προγραμματισμού του
μΕ Intel 8085

Παράδειγμα 1

Δίνεται αριθμός των 8 bit στη θέση μνήμης 0040H. Να βρεθούν τα δεκαεξαδικά ψηφία υψηλότερης και χαμηλότερης τάξης (σε δυαδική παράσταση) και να αποθηκευτούν στις θέσεις 0041H και 0042H αντίστοιχα.

0040:	X	Y
0041:	0	X
0042:	0	Y

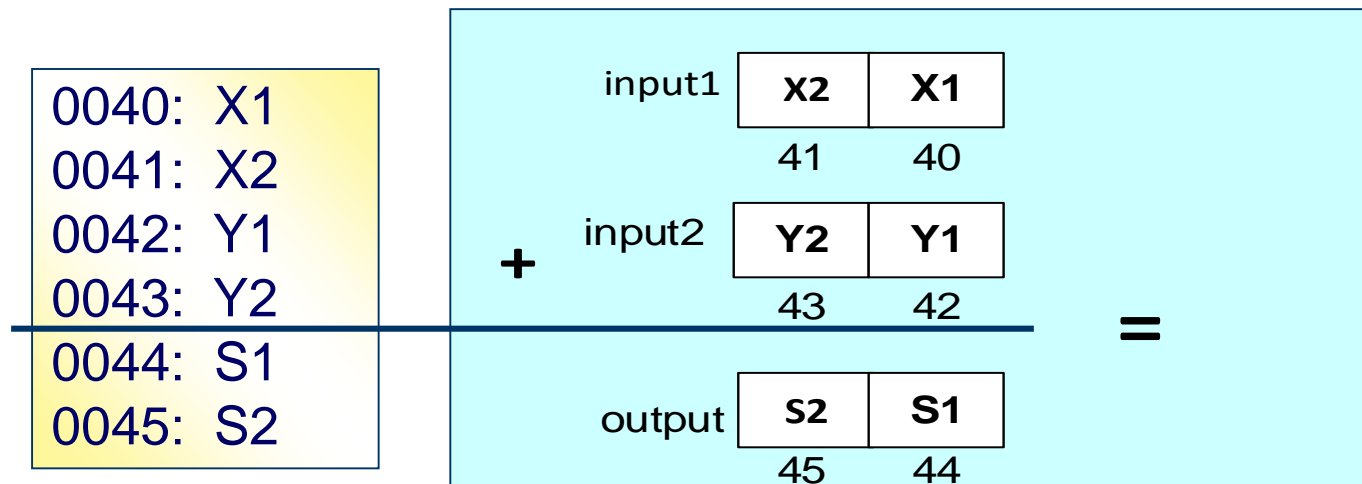
Παράδειγμα 1 – Λύση

```
LXI H,0040
MOV A, M
MOV B, A      ; Κρατώ αντίγραφο
RRC           ; 4 περιστροφές δεξιά
RRC
RRC
RRC           ; Το υψηλότερης αξίας Hex ψηφίο
              ; πάει στη χαμηλότερη θέση
ANI 0F        ; Απομόνωση X,  $0F_{16} = 00001111_2$ 
INX H
MOV M, A      ; Αποθηκεύω το X στη θέση 0041H
MOV A, B      ; Επαναφέρω
ANI 0F        ; Απομονώνω το χαμηλότερης
              ; Hex αξίας ψηφίο Y
INX H
MOV M, A      ; Αποθηκεύω το Y στη θέση 0042H
HLT           ; Τέλος
```

0040:	X	Y
0041:	0	X
0042:	0	Y

Παράδειγμα 2

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 βρίσκονται στις θέσεις 41, 43 και 45. Να βρεθεί και δεύτερος τρόπος.



Παράδειγμα 2 – Λύση Α

LXI H,0040H

MOV A,M ; (A) = X1

INX H

INX H ; (HL) = 0042H

ADD M ; Προσθέτουμε τα λιγότερο
σημαντικά bytes $X1+Y1=S1$

MOV B,A ; (B) = S1

DCX H ; (HL) = 0041H

MOV A,M ; (A) = X2

INX H

INX H ; (HL) = 0043H

ADC M ; Προσθ. τα περισσότερα
σημαντικά bytes $X2+Y2+CY$

INX H ; (HL) = 0044H

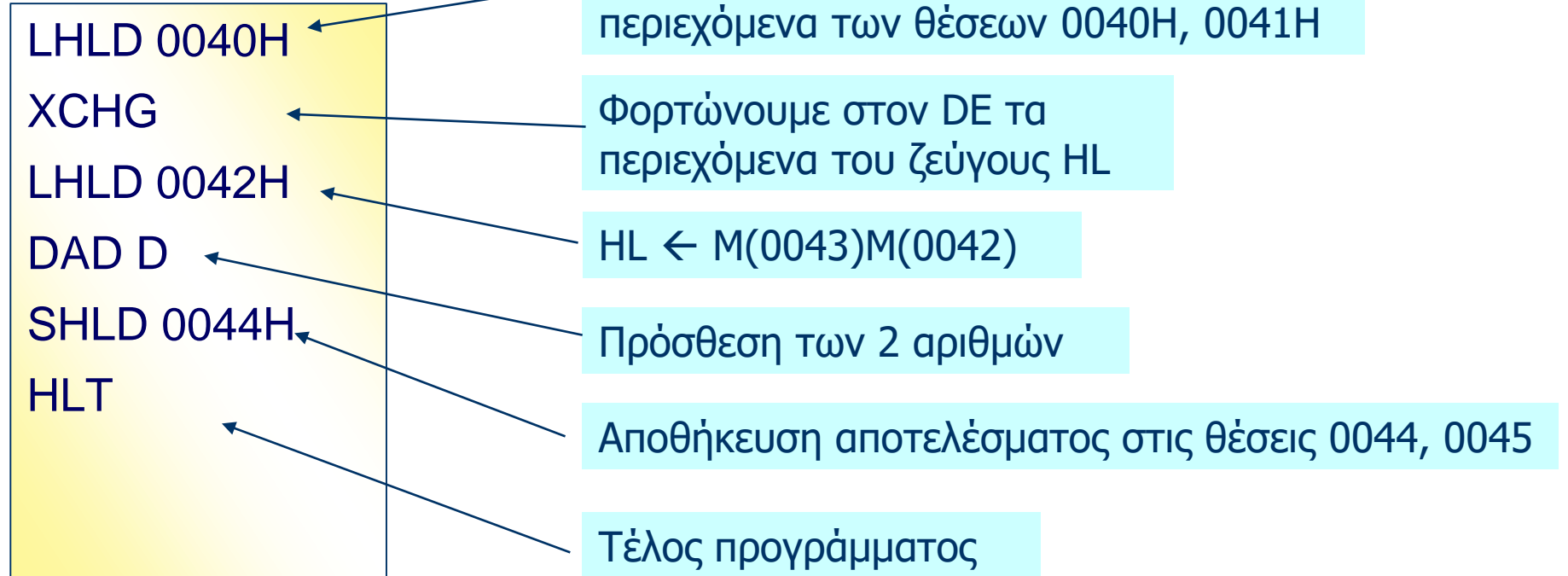
MOV M,B ; Αποθήκευση S1

INX H ; (HL) = 0045H

MOV M,A ; Αποθήκευση
αποτελέσματος S2

HLT ; Τέλος προγράμματος

Παράδειγμα 2 – Λύση Β



Παράδειγμα 3

Εδώ υπολογίζεται το τετράγωνο ενός αριθμού που περιέχεται στη θέση 0040H της μνήμης με τη βοήθεια ενός πίνακα. Ο αριθμός υποθέτουμε ότι ανήκει στο διάστημα [0,15]. Ο πίνακας τοποθετείται στις θέσεις 0100H έως 010FH της μνήμης. Το αποτέλεσμα τοποθετείται στη θέση 0041H.

Πίνακας
τετραγώνων

0100H: 00H
0101H: 01H
0102H: 04H
0103H: 09H
0104H: 10H
0105H: 19H
0106H: 24H
0107H: 31H

0108H: 40H
0109H: 51H
010AH: 64H
010BH: 79H
010CH: 90H
010DH: A9H
010EH: C4H
010FH: E1H

Παράδειγμα 3 – Λύση

```
LDA 0040H  
MOV L,A  
MVI H,0  
LXI D,0100H  
DAD D  
MOV A,M  
STA 0041H  
HLT
```

Δημιουργία της κατάλληλης διεύθυνσης του πίνακα τετραγώνων

Μεταφέρεται στον A το αντίστοιχο τετράγωνο

Αποθήκευση του αποτελέσματος στη θέση 41H

Παράδειγμα 3 – Απλούστερη Λύση

```
LDA 0040H  
MOV L, A  
MVI H, 01  
MOV A, M  
STA 0041H  
HLT
```

Δημιουργία στο HL της κατάλληλης διεύθυνσης του πίνακα τετραγώνων

Μεταφέρεται στον A το αντίστοιχο τετράγωνο

Αποθήκευση του αποτελέσματος στη θέση 41H

Παράδειγμα 4

Ένας διψήφιος δεκαδικός αριθμός είναι αποθηκευμένος στις θέσεις 0040 και 0041 της μνήμης, με το περισσότερο σημαντικό ψηφίο (MSD) στη θέση 0040. Ο αριθμός αυτός να μετατραπεί σε δυαδικό και να αποθηκευτεί στη θέση 0042.

0040: MSD

0041: LSD

0042: BIN

Μετατροπή BCD σε δυαδικό: $10 \times \boxed{\text{MSD}} + \boxed{\text{LSD}} = \boxed{\text{BIN}}$

Παράδειγμα 4 – Λύση

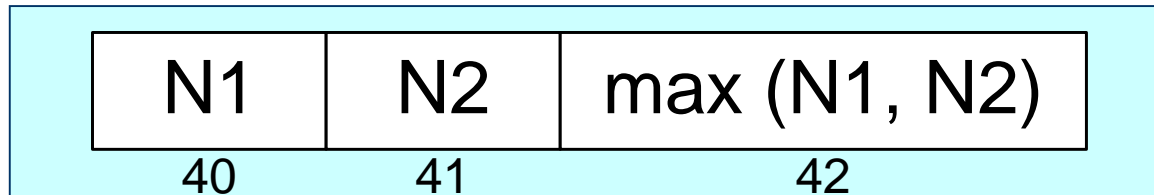
```
LXI H, 0040H
MOV A,M      ; (A) ← MSD
ADD A        ; (A) ← 2×MSD
MOV B,A      ; (B) ← 2×MSD
ADD A        ; (A) ← 2×(A) = 4×MSD
ADD A        ; (A) ← 2× (A) = 8×MSD
ADD B        ; (A) ← (A)+(B) = 8×MSD + 2×MSD
              ;                      =10×MSD

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

Παράδειγμα 5

Τα περιεχόμενα των θέσεων 0040H και 0041H της μνήμης είναι απλοί δυαδικοί αριθμοί (χωρίς πρόσημο). Να προσδιοριστεί ο μεγαλύτερος από τους δύο αριθμούς και να αποθηκευτεί στη θέση 0042^H.

0040: N1
0041: N2
0042: max



Παράδειγμα 5 – Λύση

```
LXI H, 0040H
MOV A,M
INX H
CMP M           ; Συγκρίνονται οι 2 αριθμοί
JNC DONE        ; Άλμα αν ο 1ος είναι μεγαλύτερος
                ;  $N1(A) > N2(M) \Rightarrow CY=0$ 
MOV A,M         ; διαφορετικά ( $N1 < N2$ ) προετοιμάζεται ο 2ος
DONE:
INX H           ;  $(H)(L) \leftarrow 0042H$ 
MOV M,A        ; Αποθήκευση μεγαλύτερου στη 0042H
HLT
```

Παράδειγμα 6

Στο πρόγραμμα αυτό προσδιορίζεται το μεγαλύτερο ενός συνόλου αριθμών.

Το πλήθος των αριθμών βρίσκεται στη θέση 0041H της μνήμης και οι αριθμοί αρχίζουν από τη θέση 0042H.

Το αποτέλεσμα καταχωρείται στη θέση 0040H.

```
0040: max  
0041: n  
0042: x1  
0042: x2  
...  
...  
: xn
```

Παράδειγμα 6 - Λύση

```
LXI H, 0041H
MOV B,M      ; (B) ← πλήθος αριθμών
SUB A        ; Μηδενίζεται ο A

FOR:
    INX H
    CMP M     ; Είναι ο επόμενος αριθμός > A
    JNC NEXT  ; δηλ. του τοπικού μεγίστου;
    MOV A,M   ; Ναι, αντικαθίσταται ο μέγιστος.

NEXT:
    DCR B     ; Αλλιώς προχωράμε στον έλεγχο
    JNZ FOR   ; του επόμενου αριθμού
    STA 0040H
    HLT
```

Παράδειγμα 7

Στο πρόγραμμα αυτό προσδιορίζεται το μήκος ενός συνόλου χαρακτήρων (ASCII string) όπου κάθε χαρακτήρας αντιστοιχεί σε έναν κωδικό του ενός byte. Το string είναι αποθηκευμένο στη μνήμη, από τη θέση 0041H και μετά. Το τέλος του προσδιορίζεται από ένα χαρακτήρα “CR” (0DH). Αφού βρεθεί το ζητούμενο μήκος (χωρίς το CR) τοποθετείται στη θέση 0040H.

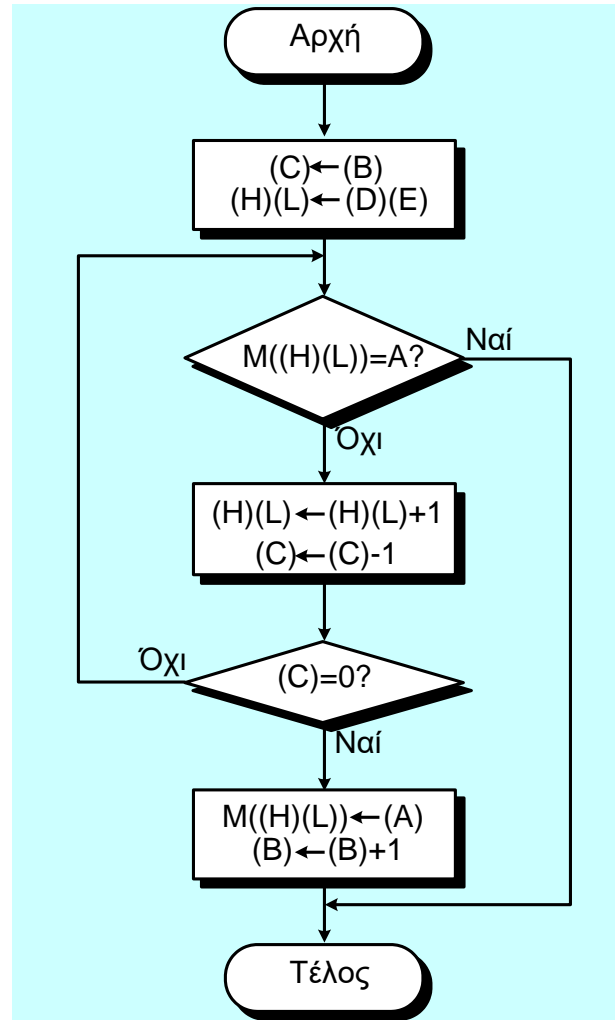
Παράδειγμα 7 – Λύση

```
LXI H,0041H
MVI B,0          ; (B)=αρχικό μήκος 0
MVI A,0DH        ; (A)← Κωδικός CR
FOR:
    CMP M         ; Είναι ο χαρακτήρας CR;
    JZ DONE       ; Αν ναι τέλος
    INR B         ; Διαφορετικά το μήκος αυξάνεται
    INX H         ; κατά 1
    JMP FOR       ; Έλεγχος επόμενου χαρακτήρα
DONE:
    MOV A,B
    STA 0040H     ; Αποθήκευση μήκους
    HLT
```

Παράδειγμα 8

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

Παράδειγμα 8 – Λογικό Διάγραμμα



Παράδειγμα 8 – Λύση

```
MOV C,B           ; (B) ← Πλήθος δεδομένων
MOV H,D           ; (D)(E) ← (H)(L)
MOV L,E

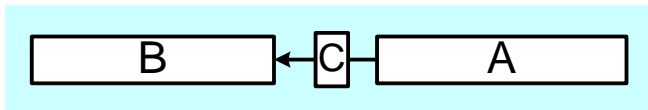
ADR1:
    CMP M         ; Συγκρίνεται ο A με τις τιμές του πίνακα
    JZ ADR2       ; Αν περιλαμβάνεται, η διαδικασία
                  ; περατώνεται

    INX H         ; Αλλιώς σαρώνουμε όλο τον πίνακα
    DCR C
    JNZ ADR1
    MOV M,A       ; Δεν βρέθηκε, άρα θα προστεθεί στο τέλος
    INR B         ; Ενημέρωση πλήθους N

ADR2:
    HLT
```

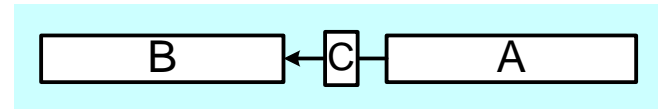
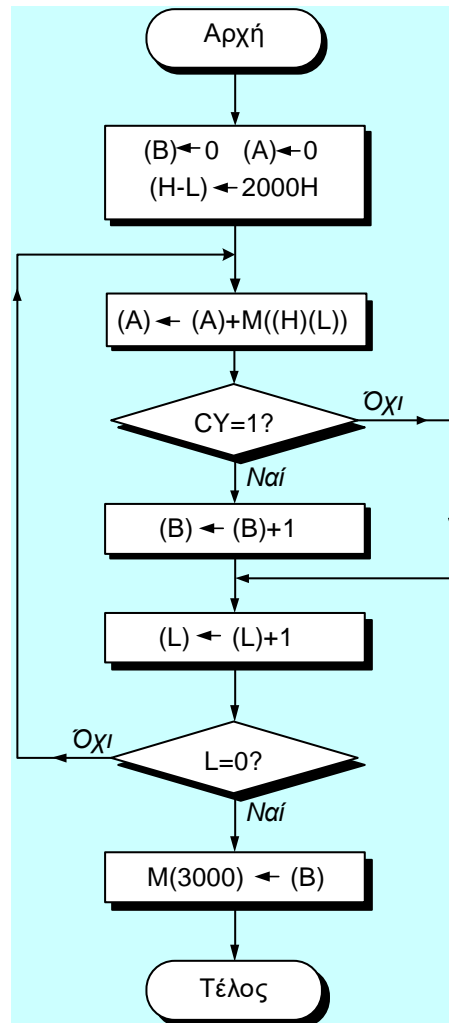
Παράδειγμα 9

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



2000:	x0
2001:	x1
2002:	x2
...	
...	
20FF:	x255
...	
...	
3000:	mean

Παράδειγμα 9 – Λογικό Διάγραμμα

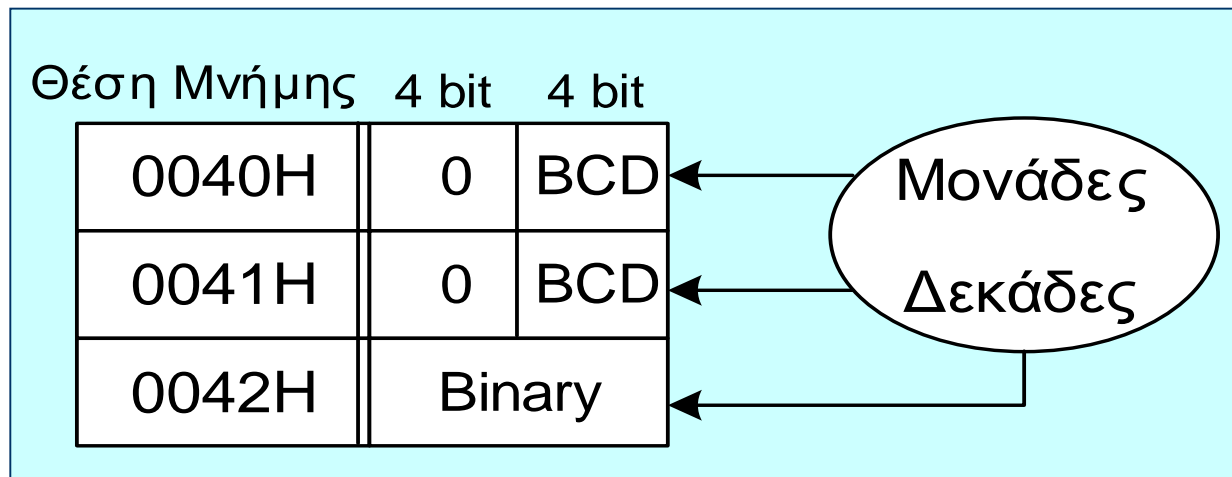


Παράδειγμα 9 - Λύση

	LXI H, 2000H	; Αρχικοποιήσεις
	MVI B, 00H	
	MVI A, 00H	
ADR1:		
	ADD M	; Άθροιση στον συσσωρευτή A
	JNC ADR2	
	INR B	; Υπερχειλίσσεις στον καταχωρητή B
ADR2:		
	INR L	; Επόμενος αριθμός
	JNZ ADR1	; Αν συμπληρωθούν 256 αθροίσεις, τέλος
	LXI H, 3000H	; και αποθήκευση αποτελέσματος
	MOV M,B	
	HLT	

Παράδειγμα 10

Μετατροπή Δυαδικού αριθμού σε BCD



Παράδειγμα 10 – Λύση

```
LDA 0042H
CPI 64H      ; Είναι μεγαλύτερος του 99? (A>99)
JNC END      ; Αν ναι τέλος (A>99)
MVI B, FFH   ; Αλλιώς A ≤ 99
DECA:
INR B
SUI 0AH      ; Αλληπάλληλες αφαιρέσεις του 10
JNC DECA     ; Αν είναι θετικός συνέχισε
ADI 0AH      ; Διόρθωση του αρνητικού υπολοίπου
STA 0040H    ; Αποθήκευση μονάδων
MOV A,B
STA 0041H    ; Αποθήκευση δεκάδων
END:
HLT
```

Παράδειγμα 11

Η χρήση της εντολής δεκαδικής ρύθμισης – 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 ;  $A \leftarrow 5DH = 63_{10}$   
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 ;  $A \leftarrow 47$ 
```

Παράδειγμα 12

Μετατροπή Δυαδικού Αριθμού σε Δεκαδική Μορφή

Παράδειγμα

$4E = 4 \cdot 16 + 14 = 4 \cdot 10 + E + 4 \cdot 6 = 78$
Σχηματίζω $4 \cdot 6$ σε δεκαδική μορφή

4

8

10

16

1E

Εδώ είναι το $4 \cdot 6$

+6 DAA

24

+4E

72

+6 DAA

78

27

Παράδειγμα 12 – Λύση

```
MOV B,A      ; B ← 16X+Y (αποθήκευση)
ANI F0       ; A ← 16X
RRC
RRC
RRC
RRC          ; A ← X
ADD A        ; A ← 2X (Hex)
DAA          ; A ← 2X (Dec)
MOV C,A      ; Αποθήκευση
ADD A        ; A ← 4X (Hex)
DAA          ; A ← 4X (Dec)
ADD C        ; A ← 6X (Hex)
DAA          ; A ← 6X (Dec)
ADD B        ; 10X+Y+6X = 16X+Y
DAA          ; Dec
HLT
```

Λόγω της DAA που ακολουθεί μπορούμε να θεωρήσουμε ότι ο B περιέχει τον αρχικό αριθμό σε δεκαδική μορφή (10X+Y)