Εντολές PIC16F887

Εντολή αφαίρεσης: Η αφαίρεση στο μικροελεγκτή γίνεται με τη βοήθεια του συμπληρώματος ως προς 2 του αφαιρέτη και αναγάγεται σε πρόσθεση.

8. sublw <αριθμητική τιμή>

Η εντολή αφαιρεί τα περιεχόμενα του W από την αριθμητική τιμή που ακολουθεί (literal) και αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Carry flag (C) και το Zero flag (Z).

Αν το αποτέλεσμα είναι θετικό τότε C=1, Z=0

Αν το αποτέλεσμα είναι αρνητικό τότε C=0, Z=0

Αν το αποτέλεσμα είναι μηδέν τότε C=1, Z=1

π.χ.:

αρχική τιμή: W=b'00001010' (=d'10')

sublw d'25'

τελική τιμή: W=b'00001111' (=d'15'), C=1, Z=0

Εντολές PIC16F887

9. subwf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή αφαιρεί τα περιεχόμενα του W από τον καταχωρητή που ακολουθεί και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Carry flag και το Zero flag.
Αν το αποτέλεσμα είναι θετικό τότε C=1, Z=0
Αν το αποτέλεσμα είναι αρνητικό τότε C=0, Z=0
Αν το αποτέλεσμα είναι μηδέν τότε C=1, Z=1
π.χ.:
Αρχικές τιμές:
Reg1=b'00001010' (=d'10'), W=b'00001111' (=d'15')
subwf Reg1,w
τελικές τιμές:
W=b'11111011' (=d'251'), Reg1=d'10', C=0, Z=0

Εντολές PIC16F887

10. andlw <αριθμητική τιμή>
Η εντολή κάνει την πράξη 'λογικό AND' αντίστοιχα στα bit της αριθμητικής τιμής που ακολουθεί και στα bit του καταχωρητή W. Αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010'
εκτέλεση andlw b'00001111'
τελική τιμή: W=b'00001010', Z=0

Εντολές PIC16F887

11. andwf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή κάνει την πράξη 'λογικό AND' αντίστοιχα στα bit του καταχωρητή που ακολουθεί και στα bit του W και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010', Reg1=b'00110011'
εκτέλεση andwf Reg1,f
τελική τιμή: W=b'10001010', Reg1=b'00000010', Z=0

Επεξήγηση: W=b'10001010'
Reg1=b'00110011'
b'00000010'

Εντολές PIC16F887

12. iorlw <αριθμητική τιμή>
Η εντολή κάνει την πράξη 'λογικό OR' αντίστοιχα στα bit της αριθμητικής τιμής που ακολουθεί και στα bit του καταχωρητή W. Αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010'
εκτέλεση iorlw b'00001111'
τελική τιμή: W=b'10001111', Z=0

Εντολές PIC16F887

13. iorwf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή κάνει την πράξη 'λογικό OR' αντίστοιχα στα bit του καταχωρητή που ακολουθεί και στα bit του W και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010', Reg1=b'00110011'
εκτέλεση iorwf Reg1,f
τελική τιμή: W=b'10001010', Reg1=b'10111011', Z=0

Επεξήγηση: W=b'10001010'
Reg1=b'00110011'
b'10111011'

Εντολές PIC16F887

14. xorlw <αριθμητική τιμή>
Η εντολή κάνει την πράξη 'λογικό XOR' αντίστοιχα στα bit της αριθμητικής τιμής που ακολουθεί και στα bit του καταχωρητή W. Αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010'
εκτέλεση xorlw b'00001111'
τελική τιμή: W=b'10000101', Z=0

Εντολές PIC16F887

15. xorwf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή κάνει την πράξη 'λογικό XOR' αντίστοιχα στα bit του καταχωρητή που ακολουθεί και στα bit του W και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: W=b'10001010', Reg1=b'00110011'
εκτέλεση iorwf Reg1,f
τελική τιμή: W=b'10001010', Reg1=b'10111001', Z=0

Επεξήγηση: W=b'10001010'
Reg1=b'00110011'
b'10111001'

Εντολές PIC16F887

16. incf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή αυξάνει κατά 1 τα περιεχόμενα του καταχωρητή που ακολουθεί και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: Reg1=d'40'
incf Reg1,f
τελική τιμή: Reg1=d'41', Z=0

αρχική τιμή: Reg1=d'255'
incf Reg1,f
τελική τιμή: Reg1=d'0', Z=1

Εντολές PIC16F887

17. incfsz <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή αυξάνει κατά 1 τα περιεχόμενα του καταχωρητή που ακολουθεί και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επιπλέον σε περίπτωση που το αποτέλεσμα είναι μηδέν τότε παρακάμπτει την επόμενη εντολή.
π.χ.: αρχική τιμή: Reg1=d'255'

```
    incfsz Reg1,f
    εντολή 1
    εντολή 2
```

Τελική τιμή: Reg1=d'0'
Η επόμενη εντολή που θα εκτελεστεί θα είναι η 'εντολή 2' (η 'εντολή 1' θα παρακαμφθεί)

Εντολές PIC16F887

18. decf <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή μειώνει κατά 1 τα περιεχόμενα του καταχωρητή που ακολουθεί και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επηρεάζει το Zero flag.
π.χ.:
αρχική τιμή: Reg1=d'40'
decf Reg1,f
τελική τιμή: Reg1=d'39', Z=0

αρχική τιμή: Reg1=d'0'
decf Reg1,f
τελική τιμή: Reg1=d'255', Z=0

Εντολές PIC16F887

19. decfsz <όνομα καταχωρητή>,a
a=f ή a=w
Η εντολή μειώνει κατά 1 τα περιεχόμενα του καταχωρητή που ακολουθεί και αν a=f αποθηκεύει το αποτέλεσμα στον καταχωρητή ενώ αν a=w αποθηκεύει το αποτέλεσμα στον W. Επιπλέον σε περίπτωση που το αποτέλεσμα είναι μηδέν τότε παρακάμπτει την επόμενη εντολή.
π.χ.: αρχική τιμή: Reg1=d'1'

```
    decfsz Reg1,f
    εντολή 1
    εντολή 2
```

Τελική τιμή: Reg1=d'0'
Η επόμενη εντολή που θα εκτελεστεί θα είναι η 'εντολή 2' (η 'εντολή 1' θα παρακαμφθεί)

Παραδείγματα – Ασκήσεις

1. Να υπολογιστεί η παράσταση A=23+15-12-10+8=24 και να αποθηκευτεί το αποτέλεσμα στον καταχωρητή Reg1 ο οποίος να οριστεί στη θέση μνήμης h'20'.

ΛΥΣΗ:

```
Reg1    equ    h'20'
movlw d'23'      (W <= d'23')
addlw d'15'      (W <= d'23'+d'15'=d'38')
movwf Reg1       (Reg1 <= W, άρα Reg1<=d'38')
movlw d'12'      (W <= d'12')
subwf Reg1,f     (Reg1 <= Reg1-W, άρα Reg1<=d'26')
movlw d'10'      (W <= d'10')
subwf Reg1,w     (W <= Reg1-W, άρα W<=d'16')
addlw d'8'       (W <= W+d'8', άρα W<=d'24')
movwf Reg1       (Reg1 <= W, άρα Reg1<=d'24')
```

Παραδείγματα – Ασκήσεις

2. Να υπολογιστεί η παράσταση A=5-22-50+30=-37 (ή b'11011011'=h'DB'=d'219' συμπλήρωμα ως προς 2 του 37) και να αποθηκευτεί το αποτέλεσμα στον καταχωρητή Reg1 ο οποίος να οριστεί στη θέση μνήμης h'25'.

ΛΥΣΗ:

```
Reg1    equ    h'25'
movlw d'5'     (W <= d'5')
movwf Reg1     (Reg1 <= d'5')
movlw d'22'    (W <= d'22')
subwf Reg1,f   (Reg1<=d'5'-d'22'=h'EF'=d'239')
movlw d'50'    (W <= d'50')
subwf Reg1,f   (Reg1 <= Reg1-W, άρα Reg1<=h'BC')
movlw d'30'    (W <= d'30')
addwf Reg1,f   (Reg1 <= Reg1+W, άρα Reg1<=h'DB')
```

Παραδείγματα – Ασκήσεις

3. Να μηδενιστούν τα τέσσερα σημαντικότερα bit του καταχωρητή Reg1 ο οποίος να οριστεί στη θέση μνήμης h'30'. Τα τέσσερα λιγότερο σημαντικά να παραμείνουν αμετάβλητα. Στη συνέχεια να γίνει '1' το LSB του καταχωρητή Reg1.

ΛΥΣΗ:

```
Reg1    equ    h'30'
movlw b'00001111'      (W <= h'0F')
andwf Reg1,f           (Reg1 <= b'0000UUUU')
movlw b'00000001'      (W <= h'01')
iorwf Reg1,f           (Reg1 <= b'0000UUU1')
```

Σημ.: U=Unchanged

Παραδείγματα – Ασκήσεις

4. Να φορτωθεί στον καταχωρητή Reg1 ο αριθμός h'AA', να υπολογιστεί το συμπλήρωμα ως προς δύο του αριθμού αυτού και να αποθηκευτεί επίσης στον καταχωρητή Reg1 ο οποίος να οριστεί στη θέση μνήμης h'40'.

ΛΥΣΗ:

```
Reg1    equ    h'40'
movlw h'AA'          (W <= h'AA')
movwf Reg1           (Reg1 <= W, άρα Reg1<=h'AA')
movlw b'11111111'    (W <= h'FF')
xorwf Reg1,f         (Reg1 XOR W, άρα Reg1<=h'55')
incf Reg1,f          (Reg1 <= Reg1+1, άρα Reg1=h'56')
```

Παραδείγματα – Ασκήσεις

5. Να γίνει η πρόσθεση A=1+2+3+4+...+20+21 και να αποθηκευτεί το αποτέλεσμα στον καταχωρητή Reg1 ο οποίος να οριστεί στη θέση μνήμης h'60'.

ΛΥΣΗ 1η:

```
Reg1    equ    h'60'
  movlw d'21'    (W <= h'21')
  movwf Reg1     (Reg1 <=d'21')
  movlw d'0'     (W <= d'0')
LOOP
  addwf Reg1,w   (W <= W + Reg1)
  decfsz Reg1,f  (Reg1 <= Reg1 - 1)
  goto LOOP
```

Παραδείγματα – Ασκήσεις

ΛΥΣΗ 2η:

```
Reg1     equ     h'60'
  movlw d'1'      (W <= d'1)
  movwf Reg1      (Reg1 <= d'1')
  movlw d'0'      (W <= d'0')
  addwf Reg1,w    (W <= W + Reg1)
  incf Reg1,f     (Reg1 <= Reg1 + 1 = d'2')
  addwf Reg1,w    (W <= W + Reg1)
  incf Reg1,f     (Reg1 <= Reg1 + 1 = d'3')
  addwf Reg1,w    (W <= W + Reg1)
```

...άλλες 17 φορές αυτές οι δύο εντολές...

```
  incf Reg1,f     (Reg1 <= Reg1 + 1)
  addwf Reg1,w    (W <= W + Reg1)
```

Ποιος τρόπος λύσης εκ των 1ης, 2ης είναι καλύτερος και γιατί;

Παραδείγματα – Ασκήσεις

Η σωστή απάντηση είναι: "ΕΞΑΡΤΑΤΑΙ!"

ΑΝΑΛΥΣΗ:

Η 1η λύση απαιτεί 6 εντολές άρα και 6 θέσεις μνήμης προγράμματος ενώ εκτελείται σε 4*21+3=87 παλμούς ρολογιού.

Η 2η λύση απαιτεί 44 εντολές άρα και 44 θέσεις μνήμης προγράμματος ενώ εκτελείται σε 44 παλμούς ρολογιού.

Υπενθυμίζεται ότι όλες οι εντολές στον PIC16F887 εκτελούνται σε 1 παλμό ρολογιού πλην των εντολών διακλάδωσης που εκτελούνται σε 2 παλμούς.

Αν προέχει η εξοικονόμηση μνήμης προγράμματος τότε καλύτερη λύση είναι η 1η ενώ αν προέχει η ταχύτητα εκτέλεσης καλύτερη λύση είναι η 2η.