

Άσκηση 3

Εντολές AND,OR,NOT,XOR,SHR,SHL,ROR,ROL

Α. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

Οι εντολές αυτής της άσκησης υποστηρίζουν λογικές πράξεις αλλά και διαχείριση περιεχομένου με περιστροφή (rotate) και μετατόπιση (shift). Ακολουθεί σύντομος πίνακας παρουσίασης των εντολών που θα μελετήσουμε στο πειραματικό μέρος.

| Εντολή | Σύνταξη | Παράδειγμα | | | | | | | | | | | | | | | |
|--|---------|---------------|---------------|---|---|---|---|---|---|---|---|---|---|---|---|--|---|
| <p>AND</p> <p>Λογικό AND μεταξύ των ορισμάτων (bit προς bit) και αποθήκευση στο πρώτο όρισμα. Ισχύει ο πίνακας αληθείας της AND μεταξύ των bit</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>F = (A AND B)</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | A | B | F = (A AND B) | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | <p>AND REG, ADDR AND ADDR, REG AND REG, REG AND ADDR, VAL AND REG, VAL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=13 (00001101) BL=07 (00000111)</p> <p>Η εντολή AND AL,BL Εκτελεί τη λογική πράξη</p> <p style="text-align: right;">00001101 (AL)</p> <p>AND</p> <p style="text-align: right;">00000111 (BL)</p> <p style="text-align: center;">-----</p> <p>AL= 00000101 (AL=5)</p> |
| A | B | F = (A AND B) | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | |

| <p>OR</p> <p>Λογικό OR μεταξύ των ορισμάτων (bit προς bit) και αποθήκευση στο πρώτο όρισμα. Ισχύει ο πίνακας αληθείας της OR μεταξύ των bit</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>F = (A OR B)</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | A | B | F = (A OR B) | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | <p>OR REG, ADDR OR ADDR, REG OR REG, REG OR ADDR, VAL OR REG, VAL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=13 (00001101) BL=07 (00000111)</p> <p>Η εντολή OR AL,BL Εκτελεί τη λογική πράξη</p> <p style="text-align: right;">00001101 (AL)</p> <p>OR</p> <p style="text-align: right;">00000111 (BL)</p> <p style="text-align: center;">-----</p> <p>AL= 00001111 (AL=15)</p> |
|--|---|--------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| A | B | F = (A OR B) | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | |

| Εντολή | Σύνταξη | Παράδειγμα | | | | | | | | | | | | | | | |
|--|---------|---------------|---------------|---|---|---|---|---|---|---|---|---|---|---|---|--|--|
| <p>XOR</p> <p>Λογικό XOR μεταξύ των ορισμάτων (bit προς bit) και αποθήκευση στο πρώτο όρισμα. Ισχύει ο πίνακας αληθείας της XOR μεταξύ των bit</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>F = (A XOR B)</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | A | B | F = (A XOR B) | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | <p>XOR REG, ADDR XOR ADDR, REG XOR REG, REG XOR ADDR, VAL XOR REG, VAL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=13 (00001101) BL=07 (00000111)</p> <p>Η εντολή XOR AL,BL Εκτελεί τη λογική πράξη</p> <p style="text-align: right;">00001101 (AL)</p> <p>XOR</p> <p style="text-align: right;">00000111 (BL)</p> <p style="text-align: center;">-----</p> <p>AL= 00001010 (AL=10)</p> |
| A | B | F = (A XOR B) | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | |

| | | | | | | | | |
|---|-------------|-------------|---|---|---|---|---|---|
| <p>NOT</p> <p>Λογικό NOT του περιεχομένου του ορίσματος (αντιστροφή bit προς bit). Ισχύει ο πίνακας αληθείας της NOT.</p> <table border="1" data-bbox="277 436 516 533"> <tr> <td>A</td> <td>F = (NOT A)</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table> | A | F = (NOT A) | 0 | 1 | 1 | 0 | <p>NOT REG NOT ADDR</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση</p> | <p>Έστω AL=5 (00000101)</p> <p>Η εντολή NOT AL Εκτελεί τη λογική πράξη</p> <p>NOT 00000101 (AL)</p> <p>AL = 1111010 (μετά)</p> |
| A | F = (NOT A) | | | | | | | |
| 0 | 1 | | | | | | | |
| 1 | 0 | | | | | | | |

| Εντολή | Σύνταξη | Παράδειγμα | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---|----|----|----|----|----|---|---|---|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| <p>SHR</p> <p>Μετατόπιση (του πρώτου ορίσματος) ΔΕΞΙΑ κατά N θέσεις (το N καθορίζεται από το δεύτερο όρισμα)</p> <p>Κατά τη μετατόπιση: (α) μετατοπίζονται όλα τα bit δεξιά, με το δεξιότερο να αποθηκεύεται στο CF (bit c του καταχωρητή κατάστασης) (β) ένα μηδενικό εισάγεται στην αριστερότερη θέση</p> | <p>SHR ADDR, VAL SHR REG, VAL SHR ADDR, CL SHR REG, CL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=181 (10110101)</p> <table border="1" data-bbox="971 688 1409 751"> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=αριστερότερο bit b0=δεξιότερο bit</p> <p>SHR AL,1 Αποτέλεσμα</p> <table border="1" data-bbox="971 968 1409 1031"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=έχει συμπληρωθεί με μηδενικό το προηγούμενο b0 που «χάθηκε» είναι αποθηκευμένο στο CF (τώρα c=1)</p> | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|----|----|----|----|----|---|---|---|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| <p>SHL</p> <p>Μετατόπιση (του πρώτου ορίσματος) ΑΡΙΣΤΕΡΑ κατά N θέσεις (το N καθορίζεται από το δεύτερο όρισμα)</p> <p>Κατά τη μετατόπιση: (α) μετατοπίζονται όλα τα bit αριστερά, με το αριστερότερο να αποθηκεύεται στο CF (bit c του καταχωρητή κατάστασης) (β) ένα μηδενικό εισάγεται στην δεξιότερη θέση</p> | <p>SHL ADDR, VAL SHL REG, VAL SHL ADDR, CL SHL REG, CL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=181 (10110101)</p> <table border="1" data-bbox="971 1274 1409 1337"> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=αριστερότερο bit b0=δεξιότερο bit</p> <p>SHL AL,1 Αποτέλεσμα</p> <table border="1" data-bbox="971 1554 1409 1617"> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b0=έχει συμπληρωθεί με μηδενικό το προηγούμενο b7 που «χάθηκε» είναι αποθηκευμένο στο CF (τώρα c=1)</p> | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Εντολή | Σύνταξη | Παράδειγμα | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|--|----|----|----|----|----|---|---|---|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| <p>ROR</p> <p>Περιστροφή (του πρώτου ορίσματος) ΔΕΞΙΑ κατά N θέσεις (το N καθορίζεται από το δεύτερο όρισμα)</p> <p>Μετατοπίζονται όλα τα bit δεξιά, με το δεξιότερο να αποθηκεύεται στο CF (bit c του καταχωρητή κατάστασης), ενώ ταυτόχρονα εισάγεται στην αριστερότερη θέση</p> | <p>ROR ADDR, VAL ROR REG, VAL ROR ADDR, CL ROR REG, CL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=180 (10110100)</p> <table border="1"> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=αριστερότερο bit b0=δεξιότερο bit</p> <p>ROR AL,1 Αποτέλεσμα</p> <table border="1"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=έχει συμπληρωθεί με το προηγούμενο b0 που «χάθηκε» και είναι επίσης αποθηκευμένο στο CF (τώρα c=0)</p> | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>ROL</p> <p>Περιστροφή (του πρώτου ορίσματος) ΑΡΙΣΤΕΡΑ κατά N θέσεις (το N καθορίζεται από το δεύτερο όρισμα)</p> <p>Μετατοπίζονται όλα τα bit αριστερά, με το αριστερότερο να αποθηκεύεται στο CF (bit c του καταχωρητή κατάστασης), ενώ ταυτόχρονα εισάγεται στην δεξιότερη θέση</p> | <p>ROL ADDR, VAL ROL REG, VAL ROL ADDR, CL ROL REG, CL</p> <p>REG=Καταχωρητής ADDR=Διεύθυνση VAL=Αριθμητική τιμή</p> | <p>Έστω AL=180 (10110100)</p> <table border="1"> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b7=αριστερότερο bit b0=δεξιότερο bit</p> <p>ROL AL,1 Αποτέλεσμα</p> <table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> </table> <p>b0=έχει συμπληρωθεί με το προηγούμενο b7 που «χάθηκε» και είναι επίσης αποθηκευμένο στο CF (τώρα c=1)</p> | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

B. ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

1. Δοκιμάστε το ακόλουθο πρόγραμμα:

```
mov al,1fh
mov bl,2eh
and al,bl
```

- α) Μετατρέψτε στο δυαδικό σύστημα τους αριθμούς 1f και 2e
- β) Μετατρέψτε στο δυαδικό σύστημα το τελικό περιεχόμενο του al
- γ) Επαληθεύστε το περιεχόμενο του al
- δ) Ποια είναι η λειτουργία του;

2. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov ax,0 α) Ποιο είναι το αποτέλεσμα που παράγει ;
and ds:[0000],ax β) Πώς παράγεται το ίδιο αποτέλεσμα χωρίς τη χρήση λογικής πράξης ;

3. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,00101111b Σχολιάστε το αποτέλεσμα και το νόημα των εντολών and σε σχέση με την
and al,00001000b τιμή του al πριν την εκτέλεσή τους (εντολές and)
mov al,00100111b
and al,00001000b * Το σύμβολο «b» σημαίνει ότι ο αριθμός είναι εκφρασμένος στο δυαδικό
mov al,00110111b σύστημα
and al,11101111b

4. Σε ένα σύστημα ελέγχου εργοστασίου, ο καταχωρητής al ελέγχει την κατάσταση 8 αντλιών (1=σε λειτουργία, 0=εκτός λειτουργίας). Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,10110011b ;τυχαία κατάσταση αντλιών α) Ποια είναι η τελική τιμή του al;
and al,11011101b β) Ποια είναι η νέα κατάσταση των αντλιών και πώς το σχολιάζετε;

5. Σε ένα σύστημα ελέγχου εργοστασίου, ο καταχωρητής al ελέγχει την κατάσταση 8 αντλιών (1=σε λειτουργία, 0=εκτός λειτουργίας). Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,00110110b ;τυχαία κατάσταση αντλιών α) Ποια είναι η τελική τιμή του al;
or al,10000000b β) Ποια είναι η νέα κατάσταση των αντλιών και πώς το σχολιάζετε;

6. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov ax,0ffffh α) Τι αποτέλεσμα παράγει ;
or ds:[0002],ax β) Πώς θα μπορούσε να επιτευχθεί το ίδιο αποτέλεσμα χωρίς τη λογική πράξη or ;

7. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,11 ;τυχαία τιμή
mov bl,8 ;τυχαία τιμή
and al,bl
mov cl,al
mov al,11
or al,bl
mov dl,al
xor cl,dl

Ποια λογική συνάρτηση υλοποιεί;

8. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,30 α) Ποια αριθμητική πράξη υλοποιεί και ποιο είναι το τελικό αποτέλεσμα ;
mov bl,36

not al
add al,1
add bl,al

β) Ποια αριθμητική πράξη θα υλοποιηθεί αν $al=36$ και $bl=30$; και ποιες άλλες πράξεις θα πρέπει να συμπληρωθούν για να είναι ολοκληρωμένο το αποτέλεσμα;

9. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,2 Σχολιάστε τα αποτελέσματα που παράγει και βρείτε τη λειτουργία της εντολής shl ως
shl al,1 προς το πρακτικό αποτέλεσμα
shl al,1
shl al,1
mov al,2
shl al,2
shl al,2

10. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,128 Σχολιάστε τα αποτελέσματα που παράγει και βρείτε τη λειτουργία της εντολής shr ως
shr al,1 προς το πρακτικό αποτέλεσμα
shr al,1
shr al,1
mov al,160
shr al,1
shr al,1
shr al,1

11. Δοκιμάστε το ακόλουθο πρόγραμμα:

mov al,160 Σχολιάστε τα αποτελέσματα που παράγει και βρείτε τη λειτουργία των εντολών rol και
rol al,1 rol ως προς το πρακτικό αποτέλεσμα
rol al,1
rol al,1
rol al,1
rol al,1
rol al,1
rol al,1
mov al,160
ror al,1
ror al,1
ror al,1
ror al,1