Αρχιτεκτονική Υπολογιστών

Ενότητα 8: Ολισθήσεις – Περιστροφές

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.ece.uowm.gr/mdasyg

Άδειες Χρήσης

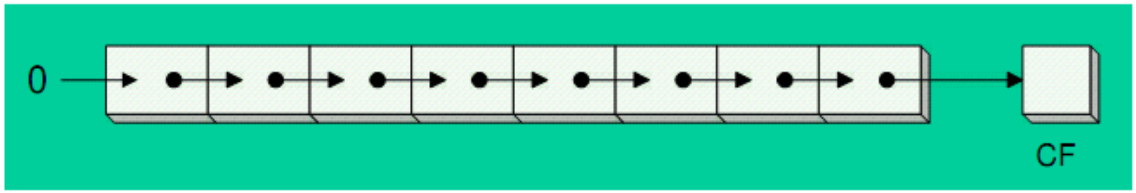
- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

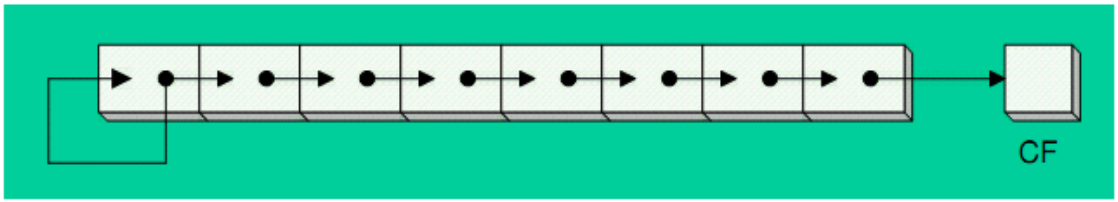
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Λογική/αριθμητική ολίσθηση

Η λογική ολίσθηση συμπληρώνει τη νέα θέση bit με 0

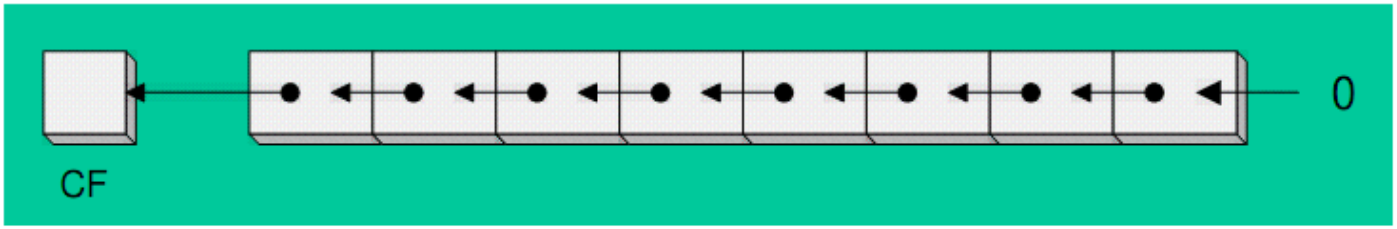


Η αριθμητική ολίσθηση συμπληρώνει τη νέα θέση bit με ήδη υπάρχον περιεχόμενο



SHift Left (SHL)

- Η SHL ή SAL μετακινεί κάθε bit του τελεστή μια θέση προς τα αριστερά προς την κατεύθυνση του σημαντικότερου bit. Το σημαντικότερο bit μετακινείται στην CF και στο χαμηλότερο bit εισέρχεται ένα 0.
- Ισοδυναμεί με ταχύτατο πολλαπλασιασμό του τελεστή επί 2.



Ταχύς πολλαπλασιασμός

- Η ολίσθηση μίας θέσης ισοδυναμεί με ταχύτατο πολλαπλασιασμό του τελεστή επί 2.

```
mov dl,5   ; dl = 00000101 = 5
shl dl,1   ; dl = 00001010 = 10
```

- Η ολίσθηση **n** bit ισοδυναμεί με πολλαπλασιασμό του τελεστή επί 2^n.

```
mov dl,5
shl dl,2                    ; DL = 20
```

Σύνταξη εντολής ολίσθησης

- Οι εντολές συντάσσονται με δυο τρόπους. Εφόσον η ολίσθηση είναι μια ο δεύτερος τελεστής είναι 1, αλλιώς ο δεύτερος τελεστής είναι ο καταχωρητής CL ο οποίος και καθορίζει το αριθμό των ολισθήσεων. π.χ. με την εντολή:

```
SHL AX,1
```

- Πολλαπλασιάζεται το περιεχόμενο του AX επί 2.
- Ενώ με τις εντολές:

```
MOV CL,4
SHL AX,CL
```

- Πολλαπλασιάζεται το περιεχόμενο του AX επί 16.

Δείτε επίσης τη διαφάνεια:

Σύνταξη εντολών ολίσθησης περιστροφής & emu8086

Σύνταξη εντολών ολίσθησης περιστροφής & emu8086

- Όλες οι εντολές ολίσθησης περιστροφής θα έχουν στη δεύτερη παράμετρο είτε την τιμή 1 (για μια ενέργεια μόνο) είτε έναν καταχωρητή που φέρει έναν αριθμό (για πολλαπλές ενέργειες).
- Η παρακάτω εντολή λοιπόν στην 8086 είναι άτοπη και λάθος: **`shl ax,3`**
- Εντούτοις, ο emu8086 προκειμένου να διευκολύνει τον προγραμματιστή τις δέχεται αυτές τις εντολές, αλλά στον κώδικα που εξάγει τις αντικαθιστά με πολλαπλές εντολές με 2η παράμετρο 1. Δηλαδή, η παραπάνω εντολή γίνεται στο machine code:

```
shl ax,1
shl ax,1
shl ax,1
```

Σύνταξη της εντολής ολίσθησης

- Οι εντολές ολίσθησης, περιστροφής συντάσσονται με 2 τρόπους:
- Είτε να θέσουμε άμεσα την τιμή του αριθμού ολισθήσεων/περιστροφών αν είναι 1:

```
shl al,1
```

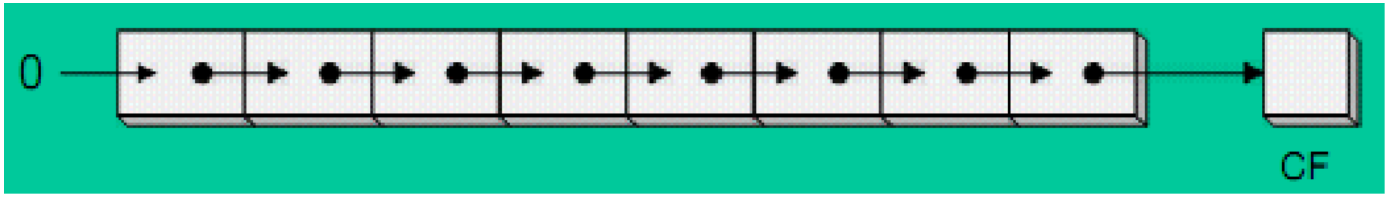
- Είτε με έμμεσο τρόπο, δηλαδή να δώσουμε την τιμή ολισθήσεων/περιστροφών σε έναν καταχωρητή, ο οποίος θα χρησιμοποιηθεί ως 2η παράμετρο.

```
mov cl,4
shl  al,cl
```

Δείτε επίσης τη διαφάνεια:
Σύνταξη εντολών ολίσθησης περιστροφής & emu8086

SHift Right (SHR)

- Η SHR μετακινεί κάθε bit του τελεστή μια θέση προς τα δεξιά. Το μη σημαντικότερο bit *(υπ' αριθμ. 0)* μετακινείται στην CF και στο υψηλότερο bit εισέρχεται ένα 0. Ισοδυναμεί με ταχύτατη μη προσημασμένη διαίρεση του τελεστή δια 2.



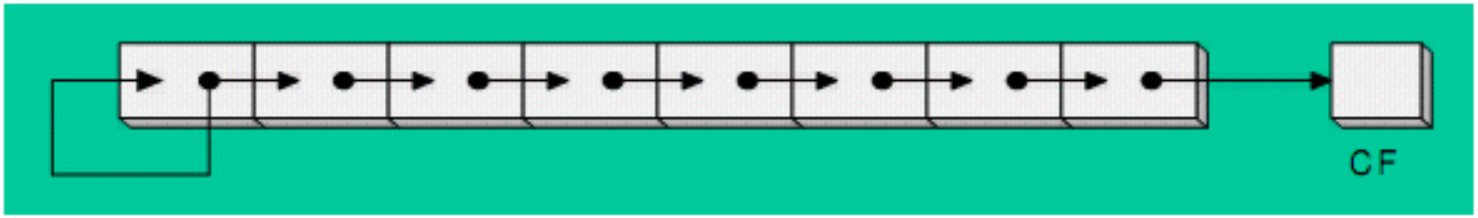
- Η ολίσθηση n bit ισοδυναμεί με διαίρεση του τελεστή δια 2^n.

```
mov dl,80 01010000
shr dl,1                    ; DL = 40
                              00101000
shr dl,2                    ; DL = 10
                              00001010
```

Στο παράδειγμα οι εντολές
είναι συνεχόμενες.

Shift Arithmetic Right (SAR)

- Η SAR είναι παρόμοια προς την προηγούμενη εντολή δηλαδή μετακινεί κάθε bit του τελεστή μια θέση προς τα δεξιά αλλά το σημαντικότερο bit αφού μετακινηθεί προς τα δεξιά επιστρέφει στην θέση του.



Παραδείγματα (1/3)

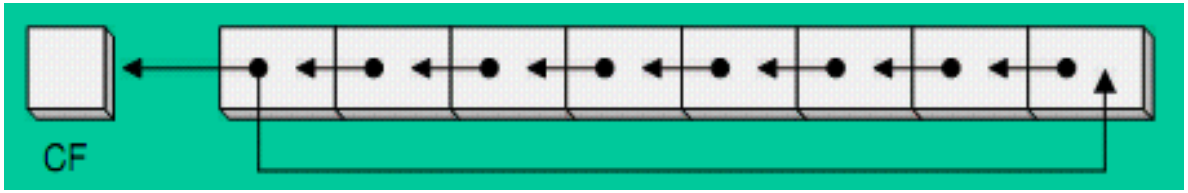
	Αρχική Τιμή στον AL	Τιμή που έχει ο AL μετά την πράξη
`mov al,6Bh`	01101011	
`shr al,1`		a. 35h 00110101
`shl al,3`		b. A8h 10101000
`mov al,8Ch`	10001100	
`sar al,1`		c. C6h 11000110
`sar al,3`		d. F8h 11111000

Στο παράδειγμα

οι εντολές είναι συνεχόμενες.

ROtate Left (ROL)

Η ROL είναι παρόμοια με την SHL με την διαφορά ότι το σημαντικότερο bit μεταφέρεται στο λιγότερο σημαντικό bit καθώς και στην CF.



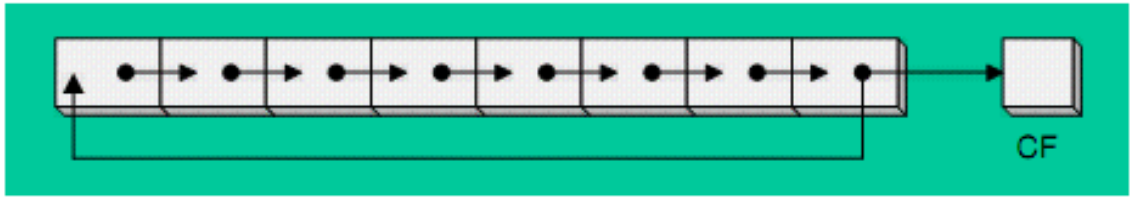

```
mov al,11110000b
rol al,1                    ; AL = 11100001b

mov dl,3Fh                  ;  00111111
rol dl,4                    ; DL = F3h 11110011
```

Στο παράδειγμα οι εντολές είναι συνεχόμενες.

ROtate Right (ROR)

Η ROR είναι παρόμοια με την SHR με την διαφορά ότι το ολιγότερο σημαντικό bit μεταφέρεται στο σημαντικότερο bit καθώς και στην CF.




```
mov al,11110000b
ror al,1                    ; AL = 01111000b

mov dl,3Fh ;00111111
ror dl,4                    ; DL = F3h  11110011
```

Στο παράδειγμα οι εντολές είναι συνεχόμενες.

Παραδείγματα (2/3)

	Αρχική Τιμή στον AL	Τιμή που έχει ο AL μετά την πράξη

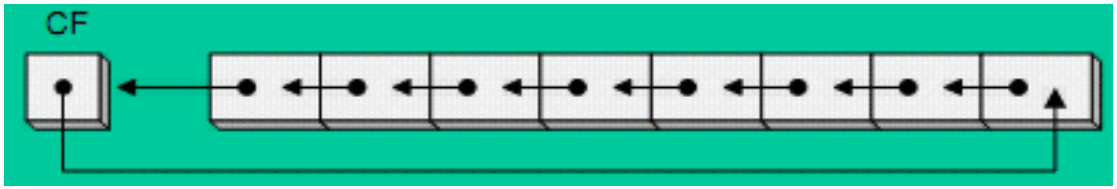
`mov al,6Bh`
`ror al,1`
`rol al,3`
	`01101011`	a. `B5h 10110101`
		b. `ADh 10101101`

Στο παράδειγμα

οι εντολές είναι συνεχόμενες.

Rotate with Carry Left (RCL)

Στην εντολή αυτή κατά την περιστροφή συμμετέχει και η CF. Δηλαδή κατά την RCL τα bits ολισθαίνουν προς τα αριστερά, το σημαντικότερο bit μεταφέρεται στην CF και το περιεχόμενο της CF μεταφέρεται στο υπ' αριθμ. 0 bit.

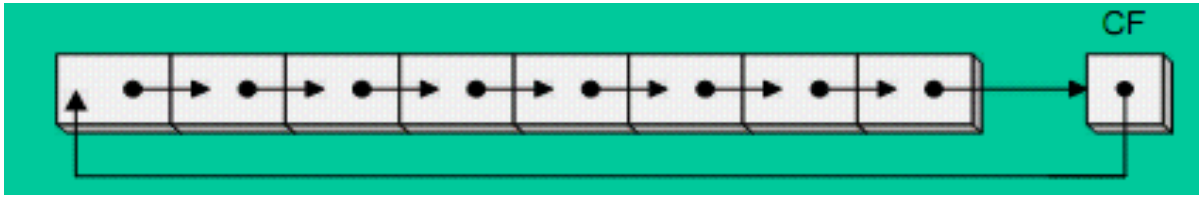



```
clc                     ; CF = 0
mov bl,88h              ; CF=0, BL = 10001000b
rcl bl,1                ; CF=1, BL = 00010000b
rcl bl,1                ; CF=0, BL = 00100001b
```

Στο παράδειγμα οι εντολές είναι συνεχόμενες.

Rotate Carry Right

Στην εντολή αυτή κατά την περιστροφή συμμετέχει και η CF. Δηλαδή κατά την RCR τα bits ολισθαίνουν προς τα δεξιά, το λιγότερο σημαντικό bit μεταφέρεται στην CF και το περιεχόμενο της CF μεταφέρεται στο σημαντικότερο bit.




```
stc                     ; CF = 1
mov ah,10h              ; CF = 1 , AH = 00010000b
rcr ah,1                ; CF = 0 , AH = 10001000b
```

Στο παράδειγμα οι εντολές είναι συνεχόμενες.

Παραδείγματα (3/3)

	Αρχική Τιμή στον AL		Τιμή που έχει ο AL μετά την πράξη	
`stc` `mov al,6Bh`	`; 01101011`			
`rcr al,1`		a.	`B5h 10110101`	`CF =1`
`rcl al,3`		b.	`AEh 10101110`	`CF =1`

Στο παράδειγμα οι εντολές είναι συνεχόμενες.

Παράδειγμα Δημιουργίας διψήφιου αριθμού (1/2)

Η πράξη της ολίσθησης επιτρέπει την τοποθέτηση 2 ψηφίων του δεκαεξαδικού συστήματος (το κάθε ψηφίο είναι 4 bit) σε έναν καταχωρητή. Η γενική δομή είναι

1. Εισαγωγή ύστερα από έλεγχο του πρώτου ψηφίου και αποθήκευση καθαρού αριθμού στον CH.
2. Εισαγωγή ύστερα από έλεγχο του δεύτερου ψηφίου και αποθήκευση καθαρού αριθμού στον CL 3)Ολίσθηση τεσσάρων bit προς τα αριστερά του πρώτου ψηφίου (του CL).
3. Ολίσθηση του συνολικού καταχωρητή CX προς τα δεξιά τέσσερα bit.

Παράδειγμα Δημιουργίας διψήφιου αριθμού (2/2)

```
mov ah,08h
int 21h
.....          ;συγκρίσεις εγκυρότητας
sub al,30h     ;μετατροπή σε καθαρό αριθμό
mov CH,AL

mov ah,08h
int 21h
.....          ;συγκρίσεις εγκυρότητας
               ;μετατροπή σε καθαρό αριθμό
sub al,30h     ;μετατροπή σε καθαρό αριθμό
mov CL,AL

shl CL,4
shr CX,4       ;ο αριθμός βρίσκεται στο CL
               ;[τα 4 πρώτα bit του πρώτου
               ; και τα 4 επόμενα του δεύτερου]
```

Παράδειγμα Δημιουργίας διψήφιου αριθμού

Η πράξη της ολίσθησης επιτρέπει την τοποθέτηση 2 ψηφίων του δεκαεξαδικού συστήματος (το κάθε ψηφίο είναι 4 bit) σε έναν καταχωρητή.

Η γενική δομή είναι:

1. Εισαγωγή ύστερα από έλεγχο του πρώτου ψηφίου και αποθήκευση καθαρού αριθμού στον CH
2. Εισαγωγή ύστερα από έλεγχο του δεύτερου ψηφίου και αποθήκευση καθαρού αριθμού στον CL.
3. Ολίσθηση τεσσάρων bit προς τα αριστερά του πρώτου ψηφίου (του CL) 3)Ολίσθηση του συνολικού καταχωρητή CX προς τα δεξιά τέσσερα bit.

```
π.χ.
mov ah,08h
int 21h
.....          ;συγκρίσεις εγκυρότητας
sub al,30h     ;μετατροπή σε καθαρό αριθμό
mov CH,AL

mov ah,08h
int 21h
.....          ;συγκρίσεις εγκυρότητας
                ;μετατροπή σε καθαρό αριθμό
sub al,30h     ;μετατροπή σε καθαρό αριθμό
mov CL,AL

shl CL,4
shr CX,4       ;ο αριθμός βρίσκεται στο CL
                ;[τα 4 πρώτα bit του πρώτου και τα 4 επόμενα του
δεύτερου]
```

Συγκρίσεις χαρακτήρων

Μερικές φορές χρειάζεται να γίνει σύγκριση με κάποιο χαρακτήρα που έχει εισάγει ο χρήστης, οπότε είναι σε ASCII.

Η σύγκριση μπορεί να γίνει με την αναγραφή της τιμής του ASCII στην εντολή cmp,όπως cmp AL,48 το οποίο έχει το μειονέκτημα της εύρεσης της κωδικοποίησης ASCII από έναν πίνακα ή της λανθασμένης κωδικοποίησης αν ο χρήστης δε το θυμάται σωστά.

Η σύγκριση μπορεί να γίνει με την αναγραφή του χαρακτήρα μέσα σε μονά ή διπλά εισαγωγικά, όπως cmp AL,'1‘ το οποίο είναι πολύ πιο εύκολο να το γράψει κάποιος, δε χρειάζεται πίνακες ASCII και είναι πιο ευκολονόητο.

Να σημειωθεί ότι και στις 2 περιπτώσεις θα δημιουργηθεί ο ίδιος κώδικας μηχανής από τον assembler. Ό 2ος τρόπος όμως έχει πάρα πολλά πλεονεκτήματα και κανένα μειονέκτημα.

Τέλος Ενότητας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης