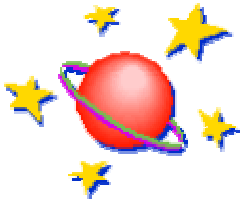


Ενότητα 1η : *Ταξινόμηση αρχιτεκτονικών συνόλων εντολών.*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει τον τρόπο με τον οποίο ταξινομούνται οι αρχιτεκτονικές των υπολογιστών ανάλογα με τον τρόπο εσωτερικής αποθήκευσης και προσπέλασης των πληροφοριών για τους τελεστέους. Επίσης θα εξετάσουμε τη δομή της στοίβας και του συσσωρευτή. Τέλος θα αναφέρουμε τα βασικά χαρακτηριστικά για το διαχωρισμό των αρχιτεκτονικών που χρησιμοποιούν καταχωρητές γενικής χρήσης.**

Προσδοκώμενα Αποτελέσματα **Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση να:**

περιγράφετε από τι αποτελείται μια θέση μνήμης,

διαχωρίζετε τα πεδία μιας εντολής και να εξηγείται τη λειτουργία καθενός από αυτά,

αναγνωρίζετε τον τρόπο με τον οποίο προσδιορίζονται οι πληροφορίες για τη διεύθυνση ή την τιμή των τελεστέων,

κατατάσσετε τα είδη των αρχιτεκτονικών που χρησιμοποιούνται ανάλογα με τον τρόπο εσωτερικής αποθήκευσης των πληροφοριών για τους τελεστέους,

περιγράφετε τον τρόπο λειτουργίας της στοίβας και να εκτελείται τις δύο βασικές λειτουργίες της,

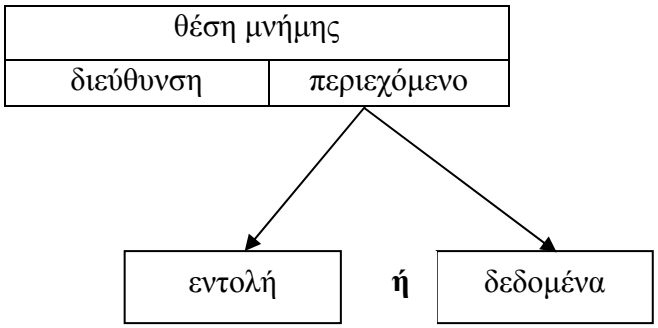
<u>εξηγείτε τον τρόπο λειτουργίας του συσσωρευτή,</u>

<u>διαχωρίζετε τις αρχιτεκτονικές καταχωρητών γενικής χρήσης.</u>

απαριθμητής προγράμματος, πεδίο κωδικού λειτουργίας, τελεστής, πηγαίος τελεστέος, τελεστέος προορισμού, στοίβα, συσσωρευτής, καταχωρητής, κορυφή στοίβας, πυθμένας, δείκτης στοίβας, αρχιτεκτονική καταχωρητή-καταχωρητή, αρχιτεκτονική καταχωρητή-μνήμης

Αρχιτεκτονικές συνόλου εντολών

Πριν από την εκτέλεση ενός προγράμματος, το πρόγραμμα και τα δεδομένα του μεταφέρονται στην κύρια μνήμη. Κάθε θέση μνήμης χαρακτηρίζεται από τη διεύθυνσή της, μέσω της οποίας ο επεξεργαστής και τα προγράμματα μπορούν να αναφέρονται στο περιεχόμενο μιας θέσης μνήμης. Το περιεχόμενο μιας θέσης μνήμης μπορεί να είναι είτε η εντολή ενός προγράμματος, είτε τα δεδομένα που χρειάζονται για την εκτέλεση της εντολής.



Σχήμα 2.1.1 – Η θέση μνήμης χαρακτηρίζεται από τη διεύθυνση και το περιεχόμενό της. Το περιεχόμενο μπορεί να είναι εντολή ή τα δεδομένα εντολής.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Μπορείτε να εξηγήσετε από που μεταφέρονται το πρόγραμμα και τα δεδομένα στην κύρια μνήμη; Να αναφέρετε τουλάχιστον δύο περιπτώσεις στις οποίες συμβαίνει κάτι τέτοιο.

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Το πρόγραμμα και τα δεδομένα του μπορεί να βρίσκονται στο σκληρό δίσκο, επομένως να μεταφέρονται από εκεί στην κύρια μνήμη. Μια άλλη περίπτωση είναι να δημιουργούνται από τον προγραμματιστή, επομένως να εισάγονται κατ' ευθείαν στην κύρια μνήμη.

Με ποιο τρόπο «καταλαβαίνει» η ΚΜΕ ότι το περιεχόμενο μιας θέσεις μνήμης είναι εντολή ή δεδομένα, έτσι ώστε να προχωρήσει στην εκτέλεση της εντολής;
Η ΚΜΕ περιέχει έναν καταχωρητή ειδικής χρήσης που ονομάζεται **απαριθμητής προγράμματος** (program counter - PC) και ο οποίος κρατάει τη διεύθυνση της επόμενης εντολής που πρόκειται να εκτελεστεί και αυξάνεται κάθε φορά που εκτελείται μια εντολή. Στην περίπτωση που η διεύθυνση μιας θέσης της κύριας μνήμης είναι το περιεχόμενο του PC, τότε η ΚΜΕ «καταλαβαίνει» πως το περιεχόμενο της θέσης μνήμης είναι η εντολή που πρόκειται να εκτελεστεί. Στην περίπτωση που η διεύθυνση της θέσης μνήμης δεν προσδιορίζεται από τον PC αλλά από κάποιο πεδίο της εντολής, ή δημιουργείται κατά την εκτέλεση της εντολής, τότε η ΚΜΕ «καταλαβαίνει» πως το περιεχόμενο της θέσης μνήμης είναι τα δεδομένα και όχι η ίδια η εντολή.

Μια θέση στην κύρια μνήμη μπορεί να περιέχει μια κωδικοποιημένη εντολή. Σε αυτή την περίπτωση, το ένα τμήμα της εντολής καθορίζει την λειτουργία που πρόκειται να εκτελεστεί, δηλαδή τον τελεστή και ονομάζεται **πεδίο κωδικού λειτουργίας**. Τα άλλα τμήματα της εντολής καθορίζουν πληροφορίες για τη διεύθυνση των τελεστέων, ή ακόμα και τους ίδιους τους τελεστέους. Κάθε τμήμα της εντολής ονομάζεται πεδίο. Στο παράδειγμα που ακολουθεί απεικονίζεται μια εντολή με 4 πεδία.

Παράδειγμα

ADD	R1	R2	R3
πεδίο κωδικού λειτουργίας	τελεστέος 1	τελεστέος 2	τελεστέος 3

Η εντολή αποτελείται από το πεδίο κωδικού λειτουργίας και από άλλα 3 πεδία τα οποία είναι: τελεστέος 1 (R1), τελεστέος 2 (R2) και τελεστέος 3 (R3) αντίστοιχα. Το πεδίο του κωδικού λειτουργίας καθορίζει τον τελεστή, δηλαδή τη λειτουργία που πρόκειται να εκτελεστεί. Στο παράδειγμα αυτό, ο τελεστής είναι το ADD και η λειτουργία που πρόκειται να εκτελεστεί είναι η πρόσθεση. Το κάθε ένα από τα άλλα τρία πεδία μπορεί να περιέχει πληροφορίες για τη διεύθυνση των τελεστέων ή τις τιμές των τελεστέων.
Κατά την εκτέλεση της εντολής αυτής, γίνεται πρόσθεση των τιμών των τελεστέων R1 και R2 και το αποτέλεσμα αποθηκεύεται στον R3. Σε αυτή την περίπτωση οι R1 και R2 ονομάζονται **πηγαίοι τελεστέοι**, επειδή αυτοί δίνουν τις αρχικές τιμές στις οποίες πρόκειται να εκτελεστεί η πρόσθεση και ο R3 ονομάζεται **τελεστέος προορισμού**, λόγω του ότι σε αυτόν θα αποθηκευτεί το αποτέλεσμα.

Προσοχή! Δεν πρέπει να συγχέεται τις έννοιες τελεστής και τελεστέος! Ο τελεστής αφορά στη λειτουργία που πρόκειται να εκτελεστεί και καθορίζεται από το πεδίο του κωδικού λειτουργίας, ενώ ο τελεστέος αφορά στα δεδομένα που είναι απαραίτητα για την εκτέλεση της εντολής. Ο τελεστέος βρίσκεται σε καταχωρητή (είναι δηλαδή το περιεχόμενό του), ή σε θέση μνήμης.

Σε όλους τους υπολογιστές είναι διαφορετικά τα πεδία της εντολής (ή το πεδίο, στην περίπτωση που το πεδίο με τον τελεστέο είναι ένα) που περιέχουν πληροφορίες για τη διεύθυνση ή τις τιμές των τελεστέων. Επομένως το κριτήριο που χρησιμοποιείται για την κατάταξη της εκάστοτε αρχιτεκτονικής που χρησιμοποιούν οι υπολογιστές, είναι ο τρόπος με τον οποίο γίνεται η προσπέλαση των τελεστέων.
Ο τρόπος με τον οποίο προσδιορίζεται η διεύθυνση ή η τιμή των τελεστέων μπορεί να γίνει με δύο διαφορετικούς τρόπους: η διεύθυνση ή η τιμή του τελεστέου να υπονοείται (**υπονοούμενος**-implicit) ή να είναι σαφώς καθορισμένη (**σαφής**-explicit).
Όσον αφορά στον τρόπο αποθήκευσης και προσπέλασης των πληροφοριών για τη διεύθυνση ή την τιμή των τελεστέων, οι κύριες αρχιτεκτονικές είναι:

- η **στοίβα**,
- ο **συσσωρευτής** και
- δύο κατηγορίες αρχιτεκτονικών με καταχωρητές γενικής χρήσης: **Καταχωρητή–μνήμης** και **Καταχωρητή-καταχωρητή**.

Στην περίπτωση της αρχιτεκτονικής Καταχωρητή-καταχωρητή, η προσπέλαση των πληροφοριών για τη διεύθυνση ή τη τιμή των τελεστέων που βρίσκονται στη μνήμη, μπορεί να γίνει μόνο μέσω των εντολών φόρτωσης και των εντολών αποθήκευσης. Στην περίπτωση αυτή, οι τελεστέοι προσπελαύνονται από τη μνήμη και αποθηκεύονται προσωρινά σε κάποιον καταχωρητή, για την εκτέλεση της εντολής. Γι' αυτό το λόγο η αρχιτεκτονική Καταχωρητή-καταχωρητή ονομάζεται και αρχιτεκτονική Φόρτωσης-αποθήκευσης.
Σε αντίθεση, στην αρχιτεκτονική Καταχωρητή-μνήμης, η προσπέλαση των τελεστέων μπορεί να γίνει απ' ευθείας από τη μνήμη.

Παράδειγμα

Με ποιον τρόπο θα εμφανιστεί η σειρά του κώδικα C = A + B στις αρχιτεκτονικές Καταχωρητή-μνήμης και Καταχωρητή-καταχωρητή αντίστοιχα; (Υποθέτουμε ότι οι διευθύνσεις των τελεστέων A, B και C βρίσκονται στην κύρια μνήμη και οι τιμές των A και B δεν μπορούν να καταστραφούν).

Απάντηση:
Ο ακόλουθος πίνακας δείχνει τον τρόπο με τον οποίο εκτελείται η εντολή.

Καταχωρητή-μνήμης	**Καταχωρητή-καταχωρητή**
Φόρτωσε R1,A (η διεύθυνση του A βρίσκεται στη μνήμη, από όπου βρίσκεται η τιμή του A και φορτώνεται στον καταχωρητή R1)	**Φόρτωσε R1, A** (η διεύθυνση του A βρίσκεται στη μνήμη, από όπου βρίσκεται η τιμή του A και φορτώνεται στον καταχωρητή R1)
Πρόσθεσε R1,B (η διεύθυνση του B βρίσκεται στη μνήμη, από όπου βρίσκεται η τιμή του B και προστίθεται στον καταχωρητή R1)	**Φόρτωσε R2, B** (η διεύθυνση του B βρίσκεται στη μνήμη, από όπου βρίσκεται η τιμή του B και φορτώνεται στον καταχωρητή R2)
Αποθήκευσε C, R1 (η διεύθυνση του C βρίσκεται στη μνήμη, όπου φορτώνεται το περιεχόμενο του καταχωρητή R1)	**Πρόσθεσε R3, R1, R2** (γίνεται η πρόσθεση των περιεχομένων των καταχωρητών R1 και R2 και το αποτέλεσμα αποθηκεύεται στον καταχωρητή R3)
	Αποθήκευσε C, R3 (η διεύθυνση του C βρίσκεται στη μνήμη, όπου φορτώνεται το περιεχόμενο του καταχωρητή R3)

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Μπορείτε να προσδιορίσετε τις διαφορές που υπάρχουν ανάμεσα στις αρχιτεκτονικές Καταχωρητή-μνήμης και Καταχωρητή-καταχωρητή; Στην περίπτωση που δυσκολεύεστε, κρίνετε σκόπιμο να επαναλάβετε την ανάγνωση της προηγούμενης παραγράφου, καθώς επίσης και του παραδείγματος.

Ανάλογα με τον τρόπο προσπέλασης των τελεστέων και την αρχιτεκτονική που χρησιμοποιείται σε κάθε υπολογιστή, έχουμε:

Αρχιτεκτονική	**Προσδιορισμός τελεστέων**	**Επεξήγηση**
Στοίβας	Υπονοούμενος	Οι τελεστέοι ωθούνται στην κορυφή της στοίβας, εκτελείται η εντολή και γίνεται εξαγωγή του αποτελέσματος από τη στοίβα. Η διεύθυνση της κορυφής της στοίβας προσδιορίζεται από το δείκτη της στοίβας.

Συσσωρευτή (Motorola 68000)	Υπονοούμενος	Η διεύθυνση του τελεστέου προορισμού υπονοείται ότι είναι η διεύθυνση του συσσωρευτή, ενώ οι διευθύνσεις των πηγαίων τελεστέων προσδιορίζονται.
Καταχωρητή- Καταχωρητή ή Φόρτωσης-αποθήκευσης (DLX)	Σαφής	Οι τελεστέοι προσπελαύνονται από τη μνήμη μόνο μέσω των εντολών φόρτωσης και αποθήκευσης και αποθηκεύονται προσωρινά σε κάποιον καταχωρητή, για την εκτέλεση της εντολής.
Καταχωρητή-μνήμης (MIPS)	Σαφής	Η προσπέλαση των τελεστέων γίνεται κατ' ευθείαν από τη μνήμη.

Αρχιτεκτονική στοίβας

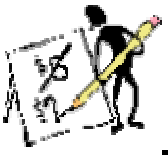
Οι στοίβα είναι μια βασική δομή δεδομένων που χρησιμοποιείται στην αρχιτεκτονική υπολογιστών και στον προγραμματισμό.

Η στοίβα είναι μια λίστα από δεδομένα, τα οποία είναι συνήθως λέξεις ή bytes, με τη διαφορά πως μόνο ένα στοιχείο δεδομένων μπορεί να προστίθεται ή να αφαιρείται από τη μία άκρη της στοίβας κάθε φορά. Η άκρη αυτή της στοίβας ονομάζεται κορυφή της στοίβας, ενώ η άλλη άκρη ονομάζεται πυθμένας τη στοίβας. Ολόκληρη η δομή συνήθως αναφέρεται σαν "στοίβα ώθησης προς τα κάτω" (pushdown stack).

Για να κατανοήσετε καλύτερα τη λογική που ακολουθείται στη δημιουργία και λειτουργία της στοίβας, μπορείτε να φανταστείτε ένα σωρό από δίσκους σε ένα εστιατόριο self-service. Οι πελάτες παίρνουν δίσκους από την κορυφή της στοίβας και οι καθαροί δίσκοι τοποθετούνται στην κορυφή της στοίβας. Με τον ίδιο τρόπο εισάγονται και εξάγονται τα δεδομένα από τη στοίβα. Δηλαδή, μόνο η μία άκρη της στοίβας χρησιμοποιείται για την εισαγωγή και εξαγωγή των δεδομένων.

Η στοίβα αποκαλείται επίσης στοίβα τελευταίο μέσα-πρώτο έξω (last-in first-out, LIFO). Η στοίβα ονομάζεται έτσι για να περιγράψει τον τύπο του μηχανισμού εισόδου-εξόδου που χρησιμοποιείται στη στοίβα: το τελευταίο στοιχείο δεδομένων που εισάγεται στη στοίβα είναι και το πρώτο το οποίο εξάγεται όταν αρχίζει η εξαγωγή των δεδομένων. Οι όροι **ώθηση** (push) και **εξαγωγή** (pop) χρησιμοποιούνται για να περιγράψουνε την τοποθέτηση ενός νέου στοιχείου δεδομένων στη στοίβα και την απομάκρυνση ενός στοιχείου δεδομένων από τη στοίβα, αντίστοιχα.

Μία στοίβα μπορεί να αποθηκευτεί στην κύρια μνήμη ενός υπολογιστή, όπου τα διαδοχικά στοιχεία τη στοίβας βρίσκονται σε διαδοχικές θέσεις μνήμης. Το πρώτο στοιχείο που εισάγεται στη στοίβα αποθηκεύεται στη διεύθυνση μνήμης **«πυθμένας»** (BOTTOM) και τα διαδοχικά στοιχεία τοποθετούνται σε περιοχές της μνήμης με μικρότερη ή μεγαλύτερη διεύθυνση, ανάλογα με τη σχεδίαση της στοίβας. Το στοιχείο που εισάγεται τελευταίο σε μία στοίβα και αποτελεί την κορυφή της στοίβας, ονομάζεται **«στοιχείο κορυφής»** (TOP).



Παράδειγμα

Έστω ότι σε μια δομή στοίβας πρόκειται να αποθηκευτούν διαδοχικά τρία στοιχεία με τιμές -50, +5 και -150 αντίστοιχα και κάθε στοιχείο καταλαμβάνει 4 bytes. Εάν ο πυθμένας βρίσκεται στη διεύθυνση 1000, σε ποιες διευθύνσεις θα αποθηκευτούν αυτά τα τρία στοιχεία στην περίπτωση που:

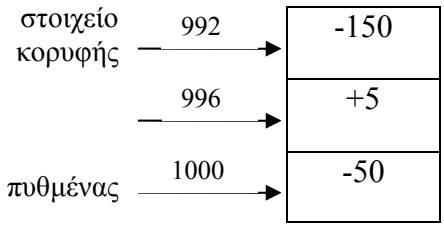
α) τα στοιχεία αποθηκεύονται διαδοχικά σε θέσεις μνήμης με μικρότερη διεύθυνση,

β) τα στοιχεία αποθηκεύονται διαδοχικά σε θέσεις μνήμης με μεγαλύτερη διεύθυνση;

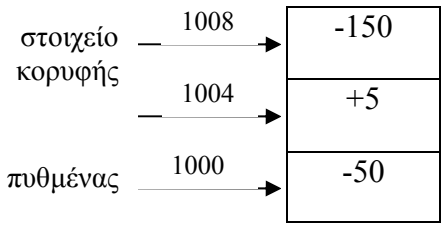
Απάντηση:

Το στοιχείο -50, το οποίο εισάγεται πρώτο στη στοίβα, αποθηκεύεται στον πυθμένα της στοίβας, δηλαδή στη διεύθυνση 1000. Τα επόμενα στοιχεία εισάγονται στη στοίβα ανάλογα με το αν η διεύθυνση του δείκτη στοίβας μειώνεται ή αυξάνεται, ως εξής:

α) Στην περίπτωση που τα στοιχεία αποθηκεύονται διαδοχικά σε θέσεις μνήμης με μικρότερη διεύθυνση, το στοιχείο με τιμή +5 θα αποθηκευτεί στη διεύθυνση 996 και το στοιχείο με τιμή -150 θα αποθηκευτεί στη διεύθυνση 992. Η τελική μορφή της στοίβας απεικονίζεται στο σχήμα που ακολουθεί.



β) Στην περίπτωση που τα στοιχεία αποθηκεύονται διαδοχικά σε θέσεις μνήμης με μεγαλύτερη διεύθυνση, το στοιχείο με τιμή +5 θα αποθηκευτεί στη διεύθυνση 1004 και το στοιχείο με τιμή -150 θα αποθηκευτεί στη διεύθυνση 1008. Η τελική μορφή της στοίβας απεικονίζεται στο σχήμα που ακολουθεί.



Πως υλοποιείται η δομή στοίβας σε έναν υπολογιστή

Όταν σε έναν υπολογιστή χρησιμοποιείται η αρχιτεκτονική στοίβας, είναι απαραίτητος ένας καταχωρητής ειδικής χρήσης που χρησιμοποιείται σαν **δείκτης της στοίβας** (καταχωρητής SP,stack pointer). Ο καταχωρητής αυτός περιέχει την εκάστοτε διεύθυνση του στοιχείου κορυφής της στοίβας. Η τιμή του δείκτη στοίβας αυξάνεται ή μειώνεται ανάλογα με τη σχεδίαση της στοίβας (τα διαδοχικά στοιχεία τοποθετούνται σε περιοχές της μνήμης με μεγαλύτερη ή μικρότερη διεύθυνση αντίστοιχα) και ανάλογα με τη λειτουργία που επιτελείται στη στοίβα κάθε φορά.

Οι δύο βασικές λειτουργίες που μπορούν να εκτελεστούν σε μια στοίβα είναι η ώθηση και η εξαγωγή, οι οποίες μπορούν να υλοποιηθούν ακολουθώντας τα παρακάτω βήματα (υποθέτουμε πως τα στοιχεία τοποθετούνται διαδοχικά σε θέσεις μνήμης με μεγαλύτερη διεύθυνση και κάθε στοιχείο καταλαμβάνει 4 bytes):

Ώθηση:

- Αύξηση της τιμής (του περιεχομένου) του δείκτη της στοίβας κατά 4.
- Ώθηση του επόμενου στοιχείου, στην κορυφή της στοίβας. Ο δείκτης στοίβας δείχνει το νέο στοιχείο κορυφής.

Εξαγωγή:

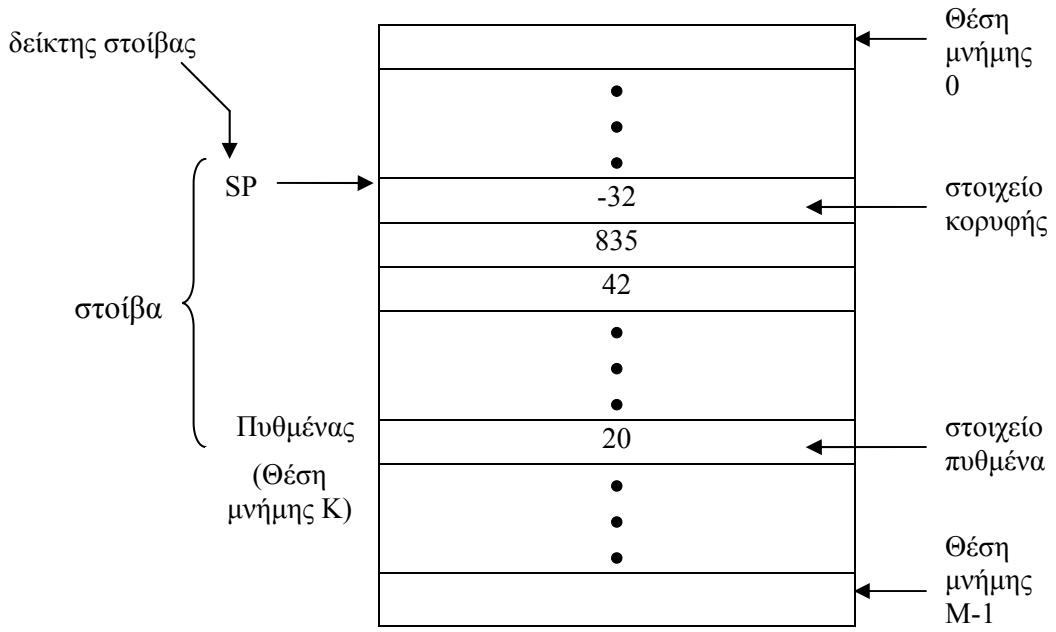
- Εξαγωγή του πρώτου στοιχείου που υπάρχει στην κορυφή της στοίβας.
- Μείωση της τιμής (του περιεχομένου) του δείκτη στοίβας κατά 4. Ο δείκτης στοίβας δείχνει στο προηγούμενο στοιχείο, το οποίο είναι τώρα το νέο στοιχείο κορυφής.

Δημιουργία στοίβας στη μνήμη από τον προγραμματιστή.

Έστω ότι η κύρια μνήμη αποτελείται από Μ θέσεις. Για τη δημιουργία της στοίβας, ο προγραμματιστής πρέπει αρχικά να ορίσει τη διεύθυνση του πυθμένα της στοίβας (έστω Κ), καθώς επίσης και το ανώτατο όριο στο οποίο μπορεί να φτάσει η στοίβα, δηλαδή τη μεγαλύτερη διεύθυνση στην οποίο μπορεί να βρίσκεται το στοιχείο κορυφής. Οι θέσεις της κύριας μνήμης οι οποίες καταχωρούνται για τη δημιουργία της στοίβας, πρέπει να είναι λιγότερες από Μ (Κ<Μ).

Στη συνέχεια, με τη χρήση εντολών σχετικές με τη λειτουργία της ώθησης, ο προγραμματιστής εισάγει τα στοιχεία στη στοίβα και αυξάνει ή μειώνει τον δείκτη στοίβας ανάλογα με τον τρόπο με τον οποίο σχεδιάζει τη στοίβα.

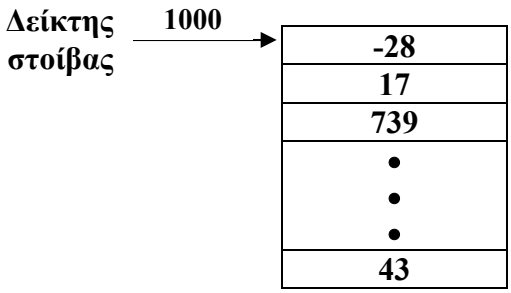
Σε πολλές περιπτώσεις που χρησιμοποιούνται οι στοίβες στον προγραμματισμό, είναι απαραίτητο να αποφεύγεται η εξαγωγή ενός στοιχείου από μία άδεια στοίβα, ή να ωθείται ένα νέο στοιχείο σε μία γεμάτη στοίβα. Για την αποφυγή τέτοιων προβλημάτων, υπάρχουν ειδικές εντολές που χρησιμοποιούν οι προγραμματιστές, οι οποίες ελέγχουν τη διεύθυνση του δείκτη της στοίβας, πριν την ώθηση ή την εξαγωγή ενός στοιχείου προς ή από τη στοίβα αντίστοιχα.



Σχήμα 2.1.2 – Μία στοίβα στην κύρια μνήμη. Έστω ότι η κύρια μνήμη αποτελείται από Μ θέσεις. Η στοίβα δημιουργείται σε συνεχόμενες θέσεις μνήμης, οι οποίες πρέπει να είναι λιγότερες από Μ. Στο σχήμα, ο πυθμένας της στοίβας βρίσκεται στη διεύθυνση Κ και περιέχει το στοιχείο 20, ενώ ο δείκτης στοίβας δείχνει το στοιχείο κορυφής που είναι το -32.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Έστω η στοίβα του παρακάτω σχήματος στην οποία ο δείκτης στοίβας βρίσκεται στη διεύθυνση 1000 και δείχνει το στοιχείο με τιμή -28.



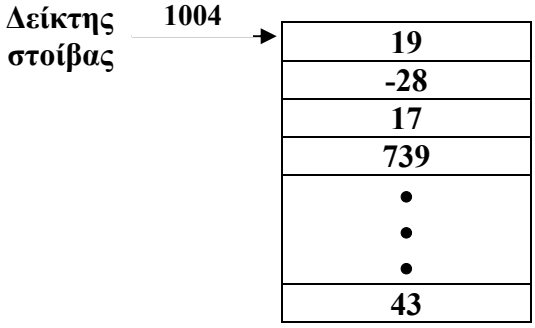
- **Να εξηγήσετε τον τρόπο με τον οποίο θα εκτελεστεί:**
  i) ώθηση στη στοίβα του στοιχείου με τιμή 19,
  ii) εξαγωγή από τη στοίβα του στοιχείου κορυφής.
- **Ποια θα είναι η διεύθυνση του δείκτη στοίβας σε κάθε μία από αυτές τις περιπτώσεις; Να θεωρήσετε ότι κάθε στοιχείο καταλαμβάνει 4 bytes και πως τα στοιχεία τοποθετούνται διαδοχικά σε θέσεις μνήμης με μεγαλύτερη διεύθυνση.**
- **Να σχεδιάσετε τη μορφή που θα έχει η νέα στοίβα μετά την ώθηση και την εξαγωγή στοιχείων αντίστοιχα.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

i) Ώθηση:

- **Αύξηση της τιμής (του περιεχομένου) του δείκτη της στοίβας κατά 4. Η νέα τιμή του δείκτη στοίβας είναι το 1004.**
- **Ώθηση του επόμενου στοιχείου, που έχει τιμή 19, στην κορυφή της στοίβας. Ο δείκτης δείχνει το νέο στοιχείο κορυφής, με τιμή 19.**

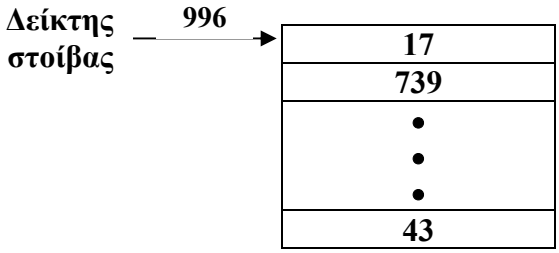
Στο παρακάτω σχήμα απεικονίζεται η στοίβα μετά την ώθηση στην αρχική στοίβα, του νέου στοιχείου με τιμή 19.

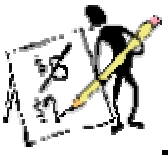


ii) Εξαγωγή:

- **Εξαγωγή του πρώτου στοιχείου που υπάρχει στην κορυφή της στοίβας, το οποίο είναι το -28.**
- **Μείωση της τιμής (του περιεχομένου) του δείκτη στοίβας κατά 4. Η νέα τιμή του δείκτη στοίβας είναι το 996. Ο δείκτης στοίβας δείχνει στο προηγούμενο στοιχείο, το οποίο έχει τιμή 17 και είναι τώρα το νέο στοιχείο κορυφής.**

Στο παρακάτω σχήμα απεικονίζεται η στοίβα μετά την εξαγωγή από την αρχική στοίβα, του στοιχείου κορυφής με τιμή -28.



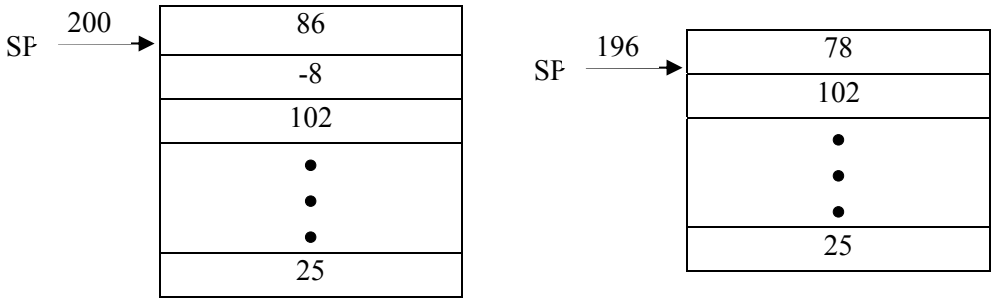


Παράδειγμα

Μία από τις λειτουργίες της στοίβας, είναι το να «κρατάει» στοιχεία τα οποία βρίσκονται σε διαδοχικές διευθύνσεις, και να εκτελεί αριθμητικές πράξεις σε αυτά τα δύο στοιχεία. Μία συχνή αριθμητική πράξη στις στοίβες είναι αυτή που εξάγει τα δύο πρώτα στοιχεία από τη στοίβα (το στοιχείο κορυφής και το αμέσως προηγούμενο), εκτελεί πρόσθεση (αφαίρεση, πολλαπλασιασμό, κλπ.) σε αυτά και στη συνέχεια ωθεί το αποτέλεσμα στην κορυφή της στοίβας.
Το αποτέλεσμα της πράξης ADD, η οποία εκτελείται στη στοίβα του σχήματος 2.1.3.(α), απεικονίζεται στο σχήμα 2.1.3.(β). Τα βήματα που ακολουθούνται για την εκτέλεση της πράξης ADD στη στοίβα, είναι:

ADD

- Εξαγωγή του στοιχείου κορυφής που βρίσκεται στη διεύθυνση που δείχνει ο δείκτης στοίβας (200), δηλαδή του στοιχείου με τιμή 86.
- Μείωση της τιμής του δείκτη στοίβας κατά 4, έτσι ώστε να δείχνει το προηγούμενο στοιχείο που είναι το -8 και βρίσκεται στη διεύθυνση 196.
- Εξαγωγή του στοιχείου κορυφής που βρίσκεται στη διεύθυνση που δείχνει ο δείκτης στοίβας (196), δηλαδή του στοιχείου με τιμή 86.
- Μείωση της τιμής του δείκτη στοίβας κατά 4, έτσι ώστε να δείχνει το προηγούμενο στοιχείο που είναι 102 και βρίσκεται στη διεύθυνση 192.
- Πρόσθεση των τιμών των δύο στοιχείων (το 86 και το -8).
- Αύξηση της τιμής του δείκτη στοίβας κατά 4, έτσι ώστε να δείχνει το στη διεύθυνση 196.
- Ώθηση του αποτελέσματος, το οποίο είναι το 78, στη θέση που δείχνει ο δείκτης στοίβας, δηλαδή στη διεύθυνση 196.



(α) Η αρχική στοίβα

(β) Η στοίβα μετά την εκτέλεση της πράξης ADD, στην αρχική στοίβα

Σχήμα 2.1.3 – Η αρχική στοίβα (α) και η στοίβα μετά την εκτέλεση της πράξης ADD (β).

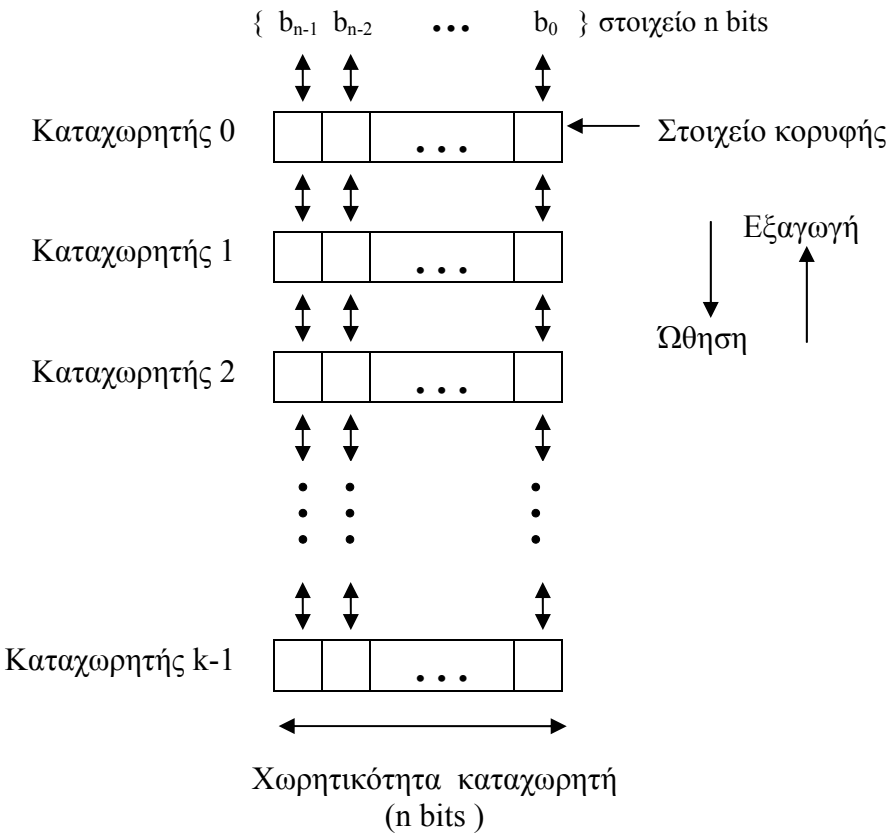
Ένας εναλλακτικός τρόπος για την υλοποίηση της στοίβας σε έναν υπολογιστή, είναι η χρήση ενός συνόλου καταχωρητών γενικής χρήσης. Υπάρχουν δύο τρόποι υλοποίησης της στοίβας:

- **Οι καταχωρητές να είναι συνδεδεμένοι παράλληλα.**
- **Η χρήση καταχωρητών ολίσθησης.**

Στη συνέχεια θα αναλύσουμε τον τρόπο με τον οποίο υλοποιείται η στοίβα στις δύο αυτές περιπτώσεις.

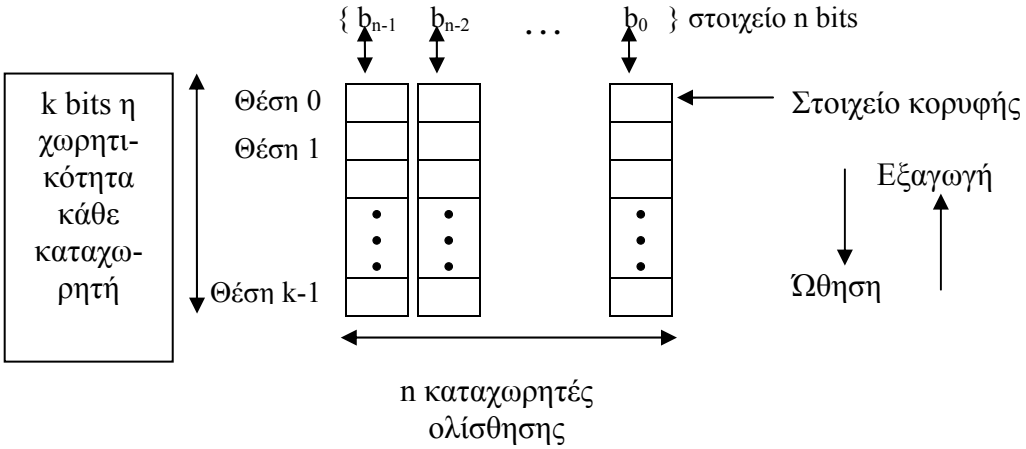
Οι καταχωρητές να είναι συνδεδεμένοι παράλληλα.

Ας υποθέσουμε πως η στοίβα αποθηκεύει στοιχεία των n δυαδικών ψηφίων (bits) και πως η χωρητικότητα της στοίβας είναι k στοιχεία. Η στοίβα δηλαδή αποτελείται από k καταχωρητές, όπου το στοιχείο κορυφής της στοίβας βρίσκεται στον καταχωρητή 0. Οι καταχωρητές είναι συνδεδεμένοι με τέτοιο τρόπο, ώστε η ώθηση ενός στοιχείου στη στοίβα μεταφέρει το περιεχόμενο όλων των καταχωρητών μία θέση προς τα κάτω. Δηλαδή, μετά την ώθηση το περιεχόμενο του καταχωρητή i μεταφέρεται στον καταχωρητή i+1 και το νέο στοιχείο που εισάγεται στη στοίβα γίνεται το περιεχόμενο του καταχωρητή 0. Με ανάλογο τρόπο, στην εξαγωγή ενός στοιχείου από τη στοίβα, το περιεχόμενο όλων των καταχωρητών μεταφέρεται μία θέση προς τα πάνω. Δηλαδή το περιεχόμενο του καταχωρητή i μεταφέρεται στον καταχωρητή i-1 και το στοιχείο εξάγεται από τον καταχωρητή 0.



Σχήμα 2.1.4 – Η υλοποίηση της στοίβας με χρήση καταχωρητών συνδεδεμένους παράλληλα.

Οι καταχωρητών ολίσθησης.
Οι καταχωρητές έχουν χωρητικότητα k bits και κάθε στοιχείο αποτελείται από n bits. Σε αυτή την περίπτωση, το στοιχείο i αποτελείται από τα bits που βρίσκονται στη θέση i όλων των καταχωρητών. Δηλαδή το στοιχείο κορυφής αποτελείται από τα bits που βρίσκονται στη θέση 0 όλων των καταχωρητών. Οι καταχωρητές μπορούν να ολισθαίνουν το περιεχόμενό τους και προς τις δύο κατευθύνσεις, κατά ένα bit κάθε φορά. Δηλαδή, μετά την ώθηση ενός στοιχείου στη στοίβα, το στοιχείο i, ολισθαίνει στη θέση i+1 σε όλους τους καταχωρητές. Με ανάλογο τρόπο, στην εξαγωγή ενός στοιχείου από τη στοίβα, η θέση 0 όλων των καταχωρητών ολισθαίνει έξω από τους καταχωρητές, καθώς το στοιχείο κορυφής που βρίσκεται εκεί, εξάγεται από τη στοίβα.



Σχήμα 2.1.5 – Η υλοποίηση της στοίβας με χρήση καταχωρητών ολίσθησης.

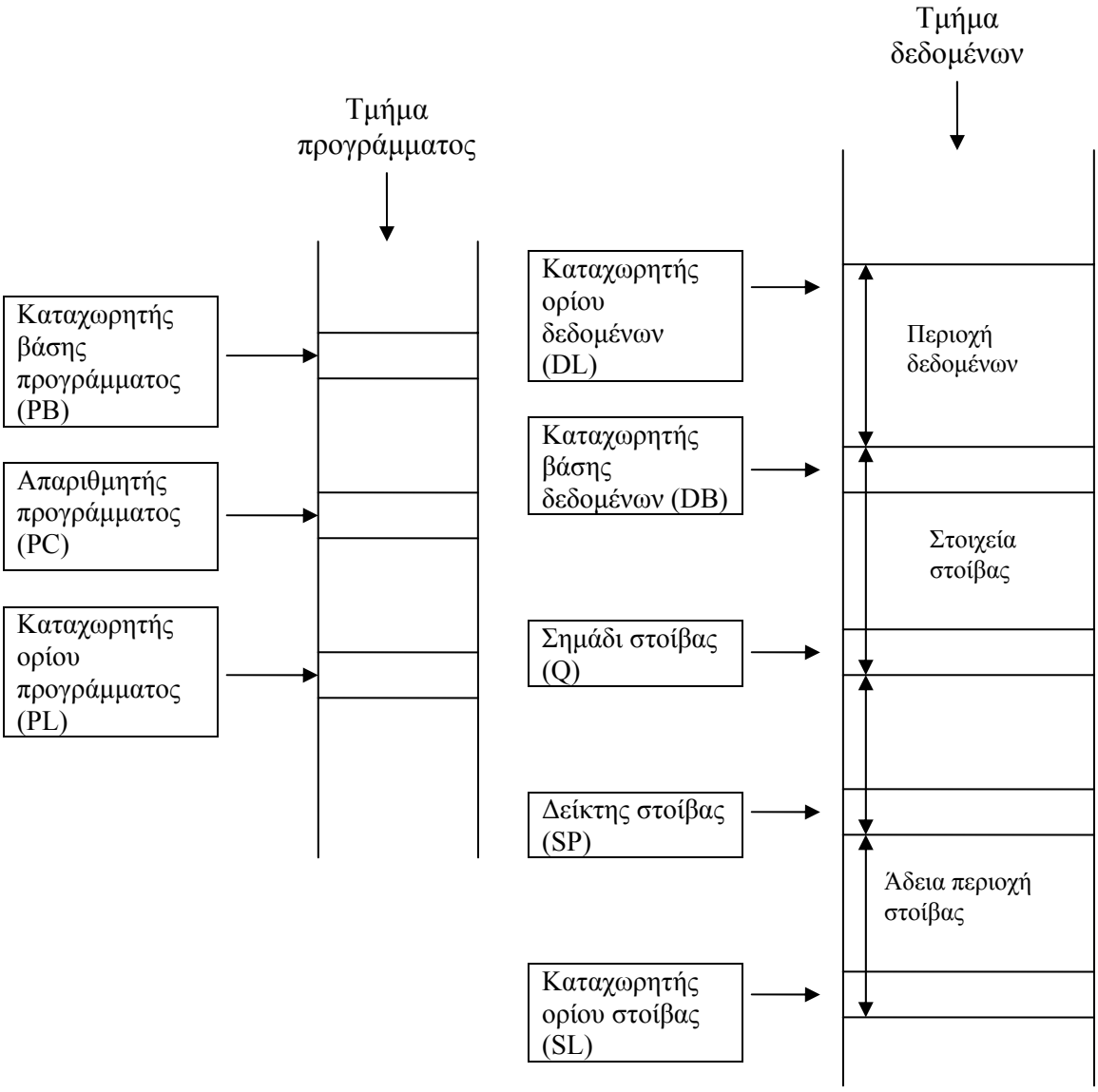
Το βασικό μειονέκτημα του τρόπου υλοποίησης της στοίβας με τη χρήση καταχωρητών, είναι το κόστος του υλικού που χρησιμοποιείται. Επιπλέον, το βάθος της στοίβας μπορεί να είναι πολύ μεγάλο (k>1000), με αποτέλεσμα να χρησιμοποιούνται μικροί καταχωρητές (μικρό n) στην περίπτωση της παράλληλης σύνδεσης των καταχωρητών και λίγοι καταχωρητές, στην περίπτωση των καταχωρητών ολίσθησης.

Η δομή της στοίβας του υπολογιστή HP3000

Ένας υπολογιστής που χρησιμοποιεί την αρχιτεκτονική στοίβας είναι ο HP3000. Ο HP3000 αποθηκεύει τις εντολές και τα δεδομένα ενός προγράμματος σε διαφορετικά τμήματα της κύριας μνήμης. Τα τμήματα καθορίζονται από τρεις διαφορετικούς καταχωρητές. Ο καταχωρητής βάσης προγράμματος (program base register - PB) και

ο καταχωρητής ορίου προγράμματος (program limit register - PL), καθορίζουν την περιοχή της κύριας μνήμης που χρησιμοποιείται από το πρόγραμμα, ενώ ο απαριθμητής προγράμματος (PC) δείχνει στην επόμενη εντολή που πρόκειται να εκτελεστεί. Κάθε ένας από αυτούς τους τρεις καταχωρητής περιέχει μια διεύθυνση των 16 ψηφίων.

Η περιοχή που περιέχει τα δεδομένα χωρίζεται σε δύο τμήματα: τη στοίβα και την περιοχή των δεδομένων. Το περιεχόμενο του DB καθορίζει τη διεύθυνση στην οποία αρχίζει η στοίβα. Η στοίβα μεγαλώνει προς μεγαλύτερες διευθύνσεις. Δηλαδή, αν για παράδειγμα το στοιχείο κορυφής βρίσκεται στη διεύθυνση 1000, τότε το επόμενο στοιχείο που ωθείται στη στοίβα, θα βρίσκεται στη θέση 1004. Η διεύθυνση του στοιχείου κορυφής καθορίζεται από τον δείκτη στοίβας SP. Το ανώτατο όριο που μπορεί να φτάσει η στοίβα, δηλαδή η μεγαλύτερη διεύθυνση στη οποία μπορεί να εισαχθεί στοιχείο, καθορίζεται από τον καταχωρητή ορίου στοίβας (stack limit register-SL). Στην περίπτωση που το περιεχόμενο του SP είναι ίσο με το περιεχόμενο του SL, η στοίβα είναι γεμάτη. Στο σχήμα 2.1.6 απεικονίζονται τα τμήματα του προγράμματος και των δεδομένων, όπως είναι οργανωμένα στον υπολογιστή HP3000.



Σχήμα 2.1.6 – Τα τμήματα του προγράμματος και των δεδομένων όπως είναι οργανωμένα στον υπολογιστή HP3000.

Οι περισσότερες εντολές του HP3000 έχουν μέγεθος 16 ψηφίων. Συνολικά υπάρχουν 13 κατηγορίες εντολών. Στη συνέχεια αναφέρουμε ενδεικτικά κάποιες εντολές στις οποίες ο ένας τελεστέος βρίσκεται στη στοίβα και τη λειτουργία που αυτές επιτελούν.

Εντολή	**Λειτουργία**
LOAD A	Ώθηση του στοιχείου A από τη μνήμη στη στοίβα.
ADDM B	Πρόσθεση της τιμής του στοιχείου, το οποίο βρίσκεται στην κύρια μνήμη στη θέση B, με την τιμή του στοιχείου κορυφής της στοίβας και αντικατάσταση του στοιχείου κορυφής με το αποτέλεσμα.
ADD	Πρόσθεση του στοιχείου κορυφής με το επόμενο στοιχείο της στοίβας και αντικατάσταση του στοιχείου κορυφής από το αποτέλεσμα.

STOR C	Αποθήκευση του στοιχείου κορυφής της στοίβας στη θέση διεύθυνση D της κύριας μνήμης.

Στην εντολή ADD τελεστέος υπονοείται ότι είναι το στοιχείο κορυφής της στοίβας. Επίσης και στις υπόλοιπες εντολές μόνο ένας από τους δύο τελεστέους καθορίζεται από την εντολή, ενώ ο άλλος τελεστέος υπονοείται ότι είναι το στοιχείο κορυφής της στοίβας. Επομένως όταν χρησιμοποιείται αρχιτεκτονική στοίβας, όπως στην περίπτωση του HP3000, η διεύθυνση στην οποία βρίσκονται οι τελεστέοι υπονοείται.

Αρχιτεκτονική συσσωρευτή

Ο **συσσωρευτής** είναι ένας καταχωρητής ειδικής χρήσης και χρησιμοποιείται στην αρχιτεκτονική υπολογιστών με τον παρακάτω τρόπο: κατά την εκτέλεση μιας εντολής ο ένας από τους δύο τελεστέους είναι το περιεχόμενο του συσσωρευτή και ο άλλος τελεστέος καθορίζεται από την εντολή.

Παράδειγμα

Έστω η εντολή: ADD A

Κατά την εκτέλεση της εντολής αυτής, το περιεχόμενο του συσσωρευτή είναι ο πρώτος τελεστέος και το στοιχείο που βρίσκεται στη θέση μνήμης με διεύθυνση Α, είναι ο δεύτερος τελεστέος. Γίνεται πρόσθεση των δύο αυτών στοιχείων και το αποτέλεσμα γίνεται το νέο περιεχόμενο του συσσωρευτή.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Να περιγράψετε τον τρόπο με τον οποίο εκτελείται η πράξη C ← [A] + [B] σε έναν υπολογιστή που χρησιμοποιεί την αρχιτεκτονική συσσωρευτή.
(Υπόδειξη: μπορείτε να χρησιμοποιήσετε την παρακάτω ακολουθία εντολών:

Load A
Add B
Store C)

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Κατά την εκτέλεση της ακολουθίας των εντολών έχουμε:
Load A: Το περιεχόμενο της θέσης μνήμης με διεύθυνση Α αποθηκεύεται στον συσσωρευτή.
Add B: Εκτελείται πρόσθεση ανάμεσα στο περιεχόμενο του συσσωρευτή και στο στοιχείο που βρίσκεται στη θέση μνήμης με διεύθυνση Β. Το αποτέλεσμα αποθηκεύεται στο συσσωρευτή.
Store C: Το περιεχόμενο του συσσωρευτή αποθηκεύεται στη θέση μνήμης με διεύθυνση C.

Αρχιτεκτονική καταχωρητών γενικής χρήσης

Οι **καταχωρητές** χρησιμοποιούνται για τη φύλαξη των μεταβλητών και πλεονεκτούν έναντι της μνήμης για τους εξής λόγους:

1. Το φορτίο στη μνήμη μειώνεται (αιτήσεις για φόρτωση ή αποθήκευση)
2. Το πρόγραμμα επιταχύνεται (αφού οι καταχωρητές είναι ταχύτεροι από τη μνήμη).
3. Η πυκνότητα κώδικα αυξάνεται (αφού ένας καταχωρητής μπορεί να προσδιοριστεί με λιγότερα ψηφία από ότι μία θέση μνήμης).

Η προσπάθεια κατανομής μεταβλητών στους καταχωρητές δεν θα είναι αποδοτική αν είναι μικρός ο αριθμός των **καταχωρητών γενικής χρήσης** (όλοι οι καταχωρητές εκτός αυτών που έχουν δεσμευθεί από το σύστημα για την εκτέλεση κάποιας συγκεκριμένης λειτουργίας και ονομάζονται καταχωρητές ειδικής χρήσης). Οι περισσότεροι μεταγλωττιστές δεσμεύουν κάποιους καταχωρητές για την αξιολόγηση εκφράσεων, κάποιους άλλους για το πέρασμα παραμέτρων και επιτρέπουν στους υπόλοιπους να χρησιμοποιηθούν για τη φύλαξη των μεταβλητών.

Δύο βασικά χαρακτηριστικά χωρίζουν τις αρχιτεκτονικές καταχωρητών γενικής χρήσης και αφορούν στον αριθμό και τη φύση των τελεστέων για μία αριθμητική και λογική εντολή:

1. **Το πλήθος των τελεστέων σε μία εντολή.**
   Τύπος εντολής με τρεις τελεστέους: η εντολή περιλαμβάνει δύο πηγαίους τελεστέους και έναν τελεστέο προορισμού.
   Τύπος εντολής με δύο τελεστέους: η εντολή περιλαμβάνει δύο πηγαίους τελεστέους, εκ των οποίων ο ένας είναι και τελεστέος προορισμού.
2. **Το πλήθος των τελεστέων που μπορεί να είναι διευθύνσεις μνήμης στις αριθμητικές και λογικές εντολές.**
   Ο αριθμός τελεστέων που μπορεί να είναι διευθύνσεις μνήμης ποικίλλει από κανέναν έως τρεις.

Μετά το 1980 έχει επικρατήσει ο σχεδιασμός υπολογιστών με καταχωρητές γενικής χρήσης για δύο λόγους:

- **Ταχύτητα (οι καταχωρητές είναι γρηγορότεροι από τη μνήμη).**
- **Ευκολία χρήσης και αποδοτικότητα (χρησιμοποιούνται ευκολότερα από το μεταγλωττιστή και από τους τύπους εσωτερικής αποθήκευσης).**

Παρόλο που υπάρχουν 7 δυνατοί συνδυασμοί για να κατηγοριοποιήσουμε τους υπολογιστές ανάλογα με τη φύση και τον αριθμό των τελεστέων που χρησιμοποιούν, οι δύο μόνο από αυτούς χρησιμοποιούνται σε όλους σχεδόν τους υπολογιστές. Τα πλεονεκτήματα και τα μειονεκτήματα καθενός από τους παραπάνω συνδυασμούς φαίνονται στον παρακάτω πίνακα:

ΤΥΠΟΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΚΑΤΑΧΩΡΗΤΩΝ ΓΕΝΙΚΗΣ ΧΡΗΣΗΣ	ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΜΕΙΟΝΕΚΤΗΜΑΤΑ
Καταχωρητή – καταχωρητή (ή Φόρτωσης-αποθήκευσης)	Απλός, σταθερό μήκος κωδικοποίησης εντολής. Απλό μοντέλο παραγωγής κώδικα. Οι εντολές απαιτούν ίδιο χρόνο για να εκτελεστούν.	Μεγαλύτερο πλήθος εντολών. Μερικές εντολές είναι μικρές και μπορεί να υπάρξει σπατάλη των bits κωδικοποίησης.
Καταχωρητή – μνήμης	Μπορεί να γίνει προσπέλαση των δεδομένων, χωρίς να έχει γίνει πρώτα ανάκληση. Ο τύπος των εντολών τείνει να γίνει ευκολότερος για κωδικοποίηση και έχει καλή περιεκτικότητα.	Οι τελεστέοι δεν είναι ισοδύναμοι αφού ένας πηγαίος τελεστέος σε μία δυαδική πράξη καταστρέφεται. Η κωδικοποίηση ενός αριθμού καταχωρητή και μιας διεύθυνσης μνήμης σε κάθε εντολή ίσως περιορίσει τον αριθμό των καταχωρητών. Ο χρόνος που χρειάζεται για την εκτέλεση των εντολών ποικίλλει ανάλογα με τη θέση των τελεστέων.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Ποιοι είναι οι λόγοι για τους οποίους οι καταχωρητές γενικής χρήσης έχουν επικρατήσει στην σχεδίαση των υπολογιστών; Να επιλέξετε μία ή περισσότερες από τις παρακάτω απαντήσεις και να αιτιολογήσετε την απάντησή σας.

Α. Ταχύτητα
Β. Πλεονεκτούν έναντι της μνήμης
Γ. Έχουν μεγαλύτερο πλήθος εντολών
Δ. Προσφέρουν ευκολία χρήσης και είναι πιο αποδοτικοί
Ε. Ο αριθμός και η φύση των τελεστέων ποικίλλει από εντολή σε εντολή

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Οι λόγοι για τους οποίους οι καταχωρητές γενικής χρήσης έχουν επικρατήσει στην σχεδίαση των υπολογιστών είναι:

Α. Ταχύτητα: είναι ταχύτεροι από τη μνήμη.

Β. Πλεονεκτούν έναντι της μνήμης: οι αιτήσεις για φόρτωση και αποθήκευση είναι λιγότερες με αποτέλεσμα να χρησιμοποιείται λιγότερο η μνήμη, οι καταχωρητές είναι ταχύτεροι από τη μνήμη, ένας καταχωρητής μπορεί να προσδιοριστεί με λιγότερα ψηφία από ότι μία θέση μνήμης (βελτιώνεται η πυκνότητα του κώδικα του προγράμματος).

Δ. Προσφέρουν ευκολία χρήσης και είναι πιο αποδοτικοί (χρησιμοποιούνται ευκολότερα από το μεταγλωττιστή και από τους τύπους εσωτερικής αποθήκευσης)

Ανακεφαλαιώνοντας λοιπόν …

Η θέση μνήμης χαρακτηρίζεται από τη διεύθυνση και το περιεχόμενό της. Το περιεχόμενο μπορεί να είναι εντολή ή τα δεδομένα εντολής.

Μια κωδικοποιημένη εντολή χωρίζεται σε πεδία. Ένα από τα πεδία πεδίο καθορίζει την λειτουργία που πρόκειται να εκτελεστεί και ονομάζεται πεδίο κωδικού λειτουργίας. Τα άλλα πεδία της εντολής καθορίζουν πληροφορίες για τη διεύθυνση των τελεστέων, ή ακόμα και τους ίδιους τους τελεστέους.

Ο τρόπος με τον οποίο προσδιορίζεται η διεύθυνση ή η τιμή των τελεστέων μπορεί να γίνει με δύο διαφορετικούς τρόπους: η διεύθυνση ή η τιμή του τελεστέου να υπονοείται (υπονοούμενος-implicit) ή να είναι σαφώς καθορισμένη (σαφής-explicit).

Όσον αφορά στον τρόπο αποθήκευσης και προσπέλασης των πληροφοριών για τη διεύθυνση ή την τιμή των τελεστέων, οι κύριες αρχιτεκτονικές είναι: η στοίβα, ο συσσωρευτής και δύο κατηγορίες αρχιτεκτονικών με καταχωρητές γενικής χρήσης: Καταχωρητή–μνήμης και Καταχωρητή-καταχωρητή.

Η στοίβα είναι μια λίστα από δεδομένα, τα οποία είναι συνήθως λέξεις ή bytes. Ένα στοιχείο δεδομένων μπορεί να προστίθεται ή να αφαιρείται μόνο από την κορυφή της στοίβας κάθε φορά.

Οι δύο βασικές λειτουργίες που μπορούν να εκτελεστούν σε μια στοίβα είναι η ώθηση και η εξαγωγή.

Ο συσσωρευτής είναι ένας καταχωρητής ειδικής χρήσης που χρησιμοποιείται στην αρχιτεκτονική υπολογιστών.

Οι καταχωρητές χρησιμοποιούνται για τη φύλαξη των μεταβλητών και πλεονεκτούν έναντι της μνήμης.