



Ενότητα 2: Η κρυφή μνήμη και η λειτουργία της

Στην ενότητα αυτή θα αναφερθούμε εκτενέστερα στη λειτουργία και την οργάνωση της κρυφής μνήμης. Θα προσδιορίσουμε τις βασικές λειτουργίες που σχετίζονται με την κρυφή μνήμη, όπως ο εντοπισμός, η τοποθέτηση και η αντικατάσταση. Τέλος θα περιγράψουμε τη λειτουργία ανάγνωσης και εγγραφής της κρυφής μνήμης.

Όταν θα έχετε μελετήσει την ενότητα θα μπορείτε:



να περιγράψετε την Κρυφή μνήμη



να αναφέρετε τις βασικές λειτουργίες της κρυφής μνήμης



να εντοπίζετε ένα μπλοκ στην κρυφή μνήμη



να τοποθετείτε ένα μπλοκ στην κρυφή μνήμη



να επιλέγετε το μπλοκ που πρέπει να αντικατασταθεί



να περιγράψετε τη διαδικασία ανάγνωσης και εγγραφής στην κρυφή μνήμη



Λέξεις κλειδιά κρυφή μνήμη, μπλοκ, ανάγνωση, εγγραφή, εντοπισμός, τοποθέτηση, αντικατάσταση, τεχνικές αντικατάστασης, πλήρως συσχετιστική κρυφή μνήμη, κρυφή μνήμη άμεσης αντιστοίχισης, σύνολο-συσχετιστική κρυφή μνήμη, διεύθυνση μπλοκ, ετικέτα, πεδίο δείκτη, μετατόπιση μπλοκ, μετατόπιση byte, ψηφίο εγκυρότητας, τεχνικές εγγραφής.



Η κρυφή μνήμη

▣ Θεωρητικό πλαίσιο λειτουργίας

Κατά τη διάρκεια εκτέλεσης ενός τμήματος προγράμματος οι διαδοχικές προσβάσεις που γίνονται στην κύρια μνήμη πραγματοποιούνται μέσα από ένα μικρό σύνολο γειτονικών διευθύνσεων.

Αυτό συμβαίνει για δύο λόγους

1. Η αποθήκευση των εντολών ενός προγράμματος γίνεται σε διαδοχικές θέσεις της κύριας μνήμης άρα η επόμενη εντολή που θα εκτελεστεί βρίσκεται συνήθως στην επόμενη θέση μνήμης. Το ίδιο συμβαίνει και με τα δεδομένα των προγραμμάτων τα οποία και αυτά, τοποθετούνται σε διαδοχικές θέσεις μνήμης.
2. Κατά την εκτέλεση βρόχων (loops) οι εντολές που περιλαμβάνονται στο σώμα του βρόχου εκτελούνται διαδοχικά πολλές φορές. Άρα γίνεται επαναλαμβανόμενη πρόσβαση του επεξεργαστή στις ίδιες θέσεις μνήμης που περιέχουν τις εντολές του βρόχου.

Η διαπίστωση ότι οι περισσότερες από τις διαδοχικές αναφορές στη μνήμη γίνονται μέσα από ένα σύνολο γειτονικών διευθύνσεων ονομάζεται αρχή της **τοπικότητας της αναφοράς** ή απλά **αρχή της τοπικότητας** και διακρίνεται σε δύο τύπους τη **τοπικότητα χρόνου** και τη **τοπικότητα χώρου**.

Τοπικότητα χρόνου

Αν ένα πρόγραμμα έχει αναφερθεί σε μια διεύθυνση μνήμης, τότε υπάρχει μεγάλη πιθανότητα να αναφερθεί σε σύντομο χρονικό διάστημα στην ίδια διεύθυνση. Αυτό συμβαίνει στην περίπτωση ενός βρόχου.

Τοπικότητα χώρου

Αν ένα πρόγραμμα έχει αναφερθεί σε μια διεύθυνση μνήμης, τότε υπάρχει μεγάλη πιθανότητα να αναφερθεί σύντομα σε γειτονικές διευθύνσεις. Αυτό συμβαίνει λόγω της αποθήκευσης των εντολών και των δεδομένων ενός προγράμματος σε διαδοχικές θέσεις μνήμης.

Αξιοποίηση της αρχής της τοπικότητας από την κρυφή μνήμη

Όταν γίνει αίτηση του επεξεργαστή για ανάγνωση ή εγγραφή σε μια διεύθυνση της κύριας μνήμης και το περιεχόμενο αυτής της θέσης δεν υπάρχει στην κρυφή μνήμη, τότε στην κρυφή μνήμη δεν μεταφέρεται μόνο το περιεχόμενο αυτής της διεύθυνσης αλλά και τα δεδομένα που περιέχονται σε γειτονικές διευθύνσεις της. Με τον τρόπο αυτό η επόμενη αίτηση του επεξεργαστή σε μια διεύθυνση της κύριας μνήμης είναι πολύ πιθανόν να εξυπηρετηθεί από την κρυφή μνήμη γιατί το περιεχόμενο της ζητούμενης διεύθυνσης θα βρεθεί στην κρυφή μνήμη. Το κέρδος στην περίπτωση αυτή είναι η ταχύτερη διεκπεραίωση της εντολής γιατί η κρυφή μνήμη είναι πολύ ταχύτερη από την κύρια μνήμη.

Στην κρυφή μνήμη εκτός από το byte και τη λέξη (word) τα δεδομένα είναι οργανωμένα σε μπλοκ ή γραμμές της κρυφής μνήμης (cache blocks ή cache lines). Το μέγεθος ενός μπλοκ μπορεί να είναι από 8 μέχρι 64 bytes. Ανάλογα με το είδος της εντολής που εκτελεί ο επεξεργαστής μπορεί να γίνει πρόσβαση του επεξεργαστή σε ποσότητα δεδομένων ίση με ένα byte, με μια λέξη ή με ένα μπλοκ. Η μεταφορά όμως των δεδομένων μεταξύ κύριας και κρυφής μνήμης γίνεται μόνο σε μπλοκ. Για το

λόγο αυτό μπορούμε να αναφερόμαστε σε μπλοκ της κύριας μνήμης θεωρώντας ότι και η κύρια μνήμη είναι οργανωμένη σε νοητά μπλοκ αντίστοιχου μεγέθους με αυτά της κρυφής μνήμης.



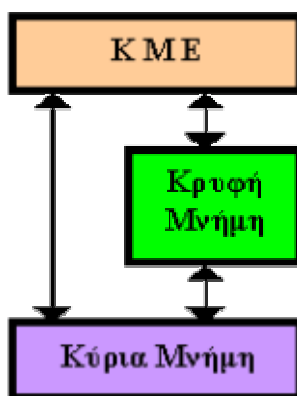
Λόγω της αρχής της τοπικότητας εάν το μέγεθος του μπλοκ της κρυφής μνήμης δεν αποτελείται από αρκετά bytes τότε είναι σαν να μην υπάρχει κρυφή μνήμη.



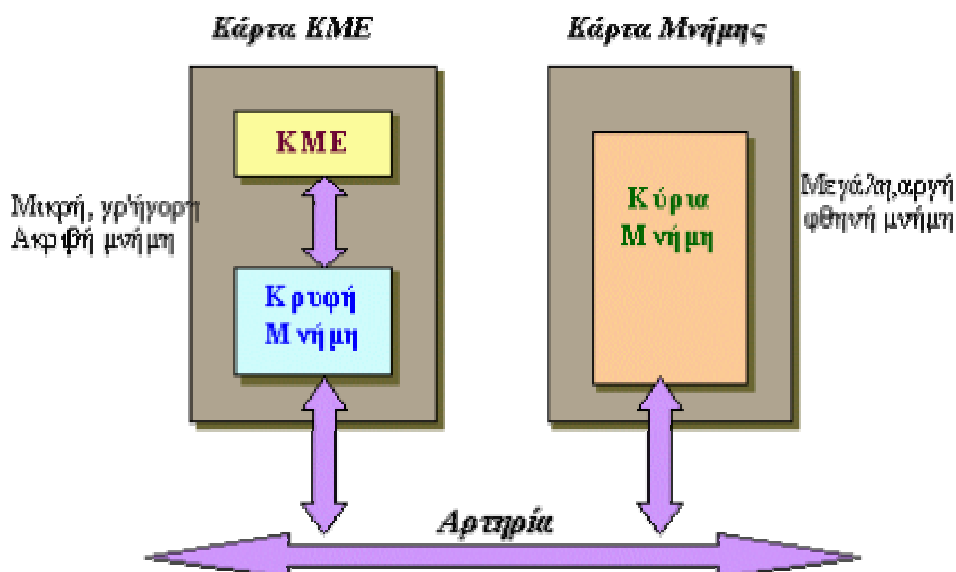
Κρυφή μνήμη δεδομένων - εντολών: Ο επεξεργαστής χρησιμοποιεί δεδομένα και εντολές. Μία κρυφή μνήμη που περιέχει και δεδομένα και εντολές θα προκαλούσε προβλήματα συνωστισμού στην μεταφορά των στοιχείων της προς τον επεξεργαστή. Για παράδειγμα ένας επεξεργαστής που χρησιμοποιεί την τεχνική σωλήνωσης μπορεί να διαβάζει μια εντολή από τη μνήμη και ταυτόχρονα να διαβάζει ή να γράφει δεδομένα μιας άλλης εντολής που βρίσκεται στη σωλήνωση. Η κρυφή μνήμη προκειμένου να αντιμετωπίσει τον κατασκευαστικό κίνδυνο. Η λύση είναι ο διαχωρισμός της κρυφής μνήμης σε κρυφή μνήμη εντολών και σε κρυφή μνήμη δεδομένων. Η ΚΜΕ γνωρίζει πότε αναφέρεται σε μία διεύθυνση δεδομένων και πότε σε μία διεύθυνση εντολών και μπορεί να χρησιμοποιήσει ξεχωριστές θύρες για κάθε αναφορά της.

▣ Η θέση της κρυφής μνήμης

Η κρυφή μνήμη αντιστοιχεί στο δεύτερο επίπεδο της ιεραρχίας μνήμης. Η κρυφή μνήμη τοποθετείται ανάμεσα στην κύρια μνήμη και τον επεξεργαστή (σχήμα 4.2.1), ενώ στους σημερινούς επεξεργαστές βρίσκεται στην κάρτα του επεξεργαστή (σχήμα 4.2.2). Η κρυφή μνήμη κατασκευάζεται από προηγμένα ολοκληρωμένα κυκλώματα, για να εξασφαλίζει μικρούς χρόνους πρόσβασης. Για το λόγο αυτό έχει αυξημένο κόστος.



Σχήμα 4.2.1 - Θέση κρυφής μνήμης: Η κρυφή μνήμη που εξυπηρετεί τον επεξεργαστή τοποθετείται ανάμεσα στην κύρια μνήμη και τον επεξεργαστή



Σχήμα 4.2.2 - Η κρυφή μνήμη στους σύγχρονους επεξεργαστές βρίσκεται στην κάρτα της ΚΜΕ



Η κρυφή μνήμη δεν τοποθετείται μόνο ανάμεσα στον επεξεργαστή και την κύρια μνήμη αλλά και ανάμεσα στην κύρια και δευτερεύουσα μνήμη. Έτσι όλοι οι σύγχρονοι σκληροί δίσκοι (και άλλες περιφερειακές συσκευές) διαθέτουν κρυφή μνήμη.

Στα περισσότερα συστήματα υπολογιστών η κρυφή μνήμη είναι οργανωμένη σε μία πολυεπίπεδη διαβάθμιση. Πιο συγκεκριμένα, υπάρχουν δύο επίπεδα κρυφής μνήμης:

- το 1ο επίπεδο κρυφής μνήμης (Level 1 cache ή L1) ή εσωτερική κρυφή μνήμη ή πρωταρχική κρυφή μνήμη: (συνήθως) βρίσκεται μέσα στην κάρτα του επεξεργαστή και έχει την ίδια ταχύτητα με αυτόν.
- το 2ο επίπεδο κρυφής μνήμης (Level 2 cache ή L2) ή εξωτερική κρυφή μνήμη ή δευτερεύουσα κρυφή μνήμη: (συνήθως) βρίσκεται στη μητρική πλακέτα του συστήματος (motherboard) ή σε ειδική κάρτα.

Η εξωτερική κρυφή μνήμη έχει συνήθως πολύ μεγαλύτερη χωρητικότητα από την εσωτερική. Για παράδειγμα ένα σύστημα υπολογιστή μπορεί να περιέχει 256 KB εσωτερικής κρυφής μνήμης και 256 MB εξωτερικής (χίλιες φορές μεγαλύτερη).



Σε ορισμένες υλοποιήσεις υπολογιστών έχει χρησιμοποιηθεί επιπλέον κρυφή μνήμη τρίτου επιπέδου (L3).



Λειτουργίες της κρυφής μνήμης

Όλες οι λειτουργίες της κρυφής μνήμης εκτελούνται από το υλικό του υπολογιστή. Η κρυφή μνήμη ενεργοποιείται όταν ο επεξεργαστής εκτελεί μια εντολή που απαιτεί πρόσβαση στην κύρια μνήμη. Οι λειτουργίες που εκτελούνται αναλυτικά είναι:

▣ Αίτηση του επεξεργαστή για ανάγνωση από μια θέση κύριας μνήμης

Όταν ο επεξεργαστής εκτελέσει μια εντολή που απαιτεί ανάγνωση από μια διεύθυνση της κύριας μνήμης τότε η αίτηση αυτή απευθύνεται στην κρυφή μνήμη με σκοπό να διαπιστωθεί αν το περιεχόμενο της ζητούμενης διεύθυνσης έχει ήδη μεταφερθεί και υπάρχει σε κάποιο μπλοκ της κρυφής μνήμης. Τότε αρχίζει η **λειτουργία εντοπισμού** η οποία ελέγχει αν σε κάποιο μπλοκ της κρυφής μνήμης έχει μεταφερθεί και περιέχεται η ζητούμενη διεύθυνση. Υπάρχουν δύο περιπτώσεις:

- Αν το μπλοκ βρεθεί στην κρυφή μνήμη τότε αυτό ονομάζεται "επιτυχία της κρυφής μνήμης" (cache hit) και το περιεχόμενο της ζητούμενης διεύθυνσης τοποθετείται στις γραμμές δεδομένων που συνδέουν την κρυφή μνήμη και τον επεξεργαστή και μεταφέρεται απευθείας στον επεξεργαστή (**λειτουργία ανάγνωσης**).
- Αν το μπλοκ δεν βρεθεί στην κρυφή μνήμη τότε αυτό ονομάζεται "αποτυχία της κρυφής μνήμης" (cache miss). Στην περίπτωση αυτή αρχίζει η **λειτουργία τοποθέτησης** κατά την οποία το μπλοκ της κύριας μνήμης που περιέχει τα δεδομένα της ζητούμενης διεύθυνσης καθώς και τα περιεχόμενα των γειτονικών διευθύνσεων, μεταφέρεται σε ένα αντίστοιχο μπλοκ της κρυφής μνήμης, που προσδιορίζεται από κάποιο αλγόριθμο αντιστοίχισης. Ταυτόχρονα το περιεχόμενο της ζητούμενης διεύθυνσης μνήμης τοποθετείται στις γραμμές δεδομένων που συνδέουν την κύρια μνήμη και τον επεξεργαστή και προωθείται στον επεξεργαστή. Επίσης είναι πιθανό το μπλοκ αυτό να αντικαταστήσει κάποιο άλλο μπλοκ που βρίσκεται στην κρυφή μνήμη οπότε πραγματοποιείται η **λειτουργία αντικατάστασης**.

▣ Αίτηση του επεξεργαστή για εγγραφή σε μια θέση κύριας μνήμης

Όταν ο επεξεργαστής εκτελεί μια εντολή που απαιτεί εγγραφή σε μια θέση της κύριας μνήμης τότε ακολουθείται μια διαδικασία αντίστοιχη με την διαδικασία της ανάγνωσης. Δηλαδή η αίτηση απευθύνεται στην κρυφή μνήμη και αρχίζει η **λειτουργία εντοπισμού** η οποία ελέγχει αν το περιεχόμενο της ζητούμενης διεύθυνσης έχει ήδη μεταφερθεί και υπάρχει σε κάποιο μπλοκ της κρυφής μνήμης. Αν υπάρξει επιτυχία της κρυφής μνήμης γίνεται απευθείας η εγγραφή στη θέση αυτή (**λειτουργία εγγραφής**). Αν υπάρξει αποτυχία της κρυφής μνήμης τότε αρχίζει η **λειτουργία τοποθέτησης** η οποία θα μεταφέρει το μπλοκ της κύριας μνήμης που περιέχει τα δεδομένα της ζητούμενης διεύθυνσης καθώς και των γειτονικών διευθύνσεων, σε αντίστοιχο μπλοκ της κρυφής μνήμης, που προσδιορίζεται από κάποιο αλγόριθμο αντιστοίχισης. Είναι πιθανό το μπλοκ αυτό να πρέπει να αντικαταστήσει κάποιο άλλο μπλοκ που βρίσκεται ήδη στην κρυφή μνήμη (**λειτουργία αντικατάστασης**). Στη συνέχεια πραγματοποιείται η **λειτουργία εγγραφής**.

Σύμφωνα με τα ανωτέρω οι λειτουργίες της κρυφής μνήμης είναι:

- i. Η λειτουργία του εντοπισμού
- ii. Η λειτουργία της τοποθέτησης
- iii. Η λειτουργία της αντικατάστασης
- iv. Η λειτουργία της ανάγνωσης
- v. Η λειτουργία της εγγραφής

Η αντιστοίχιση ενός μπλοκ της κύριας μνήμης στην κρυφή μνήμη

Το μέγεθος της κρυφής μνήμης είναι εκατοντάδες φορές μικρότερο από αυτό της κύριας. Οι γνωστοί επεξεργαστές των προσωπικών υπολογιστών Pentium, για παράδειγμα διαθέτουν συνήθως κρυφή μνήμη μεγέθους 512 KB ενώ το μέγιστο μέγεθος της κύριας μνήμης που μπορούν να δεχθούν είναι πάνω από 2 GB. Η αύξηση του μεγέθους της κρυφής μνήμης έχει άμεση επίπτωση στο κόστος υλοποίησης της, για παράδειγμα οι επεξεργαστές Pentium διατίθενται σε ειδικές εκδόσεις με κρυφή μνήμη 2 MB αλλά σε υψηλότερη τιμή.

Επειδή η κρυφή μνήμη είναι πολύ μικρότερη από την κύρια θα πρέπει να υπάρχει ένας αλγόριθμος αντιστοίχισης των διευθύνσεων της κύριας μνήμης στην κρυφή μνήμη.

Η μεταφορά δεδομένων από την κύρια μνήμη προς την κρυφή γίνεται σε μπλοκ, συνεπώς η αντιστοίχιση διευθύνσεων της κύριας μνήμης πραγματοποιείται με αλγορίθμους που αντιστοιχούν διευθύνσεις μπλοκ της κύριας μνήμης σε μπλοκ της κρυφής μνήμης.

Υπάρχουν τρεις διαφορετικοί τρόποι αντιστοίχισης ενός μπλοκ της κύριας μνήμης στην κρυφή μνήμη και οι τρεις αυτοί τρόποι αντιστοίχισης καθορίζουν την οργάνωση της κρυφής μνήμης.

- **Κρυφή μνήμη Άμεσης αντιστοίχισης** (direct mapped)

Σε αυτό τον τύπο κρυφής μνήμης κάθε μπλοκ της κύριας μνήμης μπορεί να αντιστοιχηθεί μόνο σε ένα συγκεκριμένο μπλοκ της κρυφής μνήμης. Η διεύθυνση αυτού του συγκεκριμένου μπλοκ της κρυφής μνήμης υπολογίζεται από την διεύθυνση του μπλοκ της κύριας μνήμης σύμφωνα με τον παρακάτω τύπο:

Διεύθυνση μπλοκ κρυφής μνήμης = (διεύθυνση μπλοκ κύριας μνήμης) MOD (αριθμός μπλοκ κρυφής μνήμης)



Υπενθυμίζεται ότι η συνάρτηση MOD υπολογίζει το ακέραιο υπόλοιπο της διαίρεσης δύο αριθμών.

- **Κρυφή μνήμη Πλήρως Συσχετιστική** (fully associative)

Σε αυτό τον τύπο κρυφής μνήμης κάθε μπλοκ της κύριας μνήμης μπορεί να αντιστοιχηθεί σε οποιοδήποτε μπλοκ της κρυφής μνήμης.

- **Κρυφή μνήμη Συνολο-συσχετιστική** (set-associative)

Σε αυτό τον τύπο κρυφής μνήμης τα μπλοκ της κρυφής μνήμης είναι οργανωμένα σε σύνολα που το κάθε ένα από αυτά περιέχει δύο ή περισσότερα μπλοκ. Ένα μπλοκ της κύριας μνήμης αντιστοιχίζεται σε ένα συγκεκριμένο σύνολο της κρυφής μνήμης αλλά στη συνέχεια μπορεί να τοποθετηθεί σε οποιοδήποτε από τα μπλοκ που περιέχει το σύνολο αυτό. Η διεύθυνση του

συγκεκριμένου συνόλου της κρυφής μνήμης υπολογίζεται (όπως και στην άμεση αντιστοίχιση) από την διεύθυνση του μπλοκ της κύριας μνήμης σύμφωνα με τον παρακάτω τύπο:

$$\text{Διεύθυνση συνόλου κρυφής μνήμης} = (\text{διεύθυνση μπλοκ κύριας μνήμης}) \text{ MOD } (\text{αριθμός συνόλων κρυφής μνήμης})$$

Αν υπάρχουν m μπλοκ μέσα σε κάθε σύνολο τότε η κρυφή μνήμη ονομάζεται m -δρόμων συνολο-συσχετιστική, δηλαδή ο αριθμός των δρόμων δηλώνει τον αριθμό των μπλοκ που περιέχει κάθε σύνολο της κρυφής μνήμης.

Η κρυφή μνήμη **άμεσης αντιστοίχισης** έχει χαμηλό κόστος υλοποίησης επειδή τα κυκλώματα εντοπισμού που χρησιμοποιεί ελέγχουν την διεύθυνση ενός μόνο μπλοκ της κρυφής μνήμης, αλλά η απόδοσή της είναι χαμηλή επειδή είναι πιθανό τα μπλοκ της κύριας μνήμης που μεταφέρονται στην κρυφή να αντιστοιχηθούν στο ίδιο μπλοκ της κρυφής με αποτέλεσμα να υπάρχουν πολλές αντικαταστάσεις μπλοκ (καθυστερήσεις).

Η **πλήρως συσχετιστική** μνήμη έχει το υψηλότερο κόστος υλοποίησης επειδή διαθέτει κυκλώματα εντοπισμού, τα οποία ελέγχουν ταυτόχρονα την διεύθυνση όλων των μπλοκ της κρυφής μνήμης. Αυτός ο τύπος κρυφής μνήμης έχει και την καλύτερη απόδοση.

Η **συνολο-συσχετιστική** μνήμη αποτελεί μια ενδιάμεση λύση μεταξύ του χαμηλού κόστους υλοποίησης της μνήμης άμεσης αντιστοίχισης και της απόδοσης που παρουσιάζει η πλήρως συσχετιστική μνήμη. Αποτελεί τον πιο συνηθισμένο τύπο κρυφής μνήμης και η υλοποίησή της γίνεται με 2 ή 4 μπλοκ ανά σύνολο (2 και 4 δρόμων συνολο-συσχετιστική)



Μία κρυφή μνήμη άμεσης αντιστοίχισης με μέγεθος m μπλοκ, μπορεί να θεωρηθεί ότι είναι μια 1 δρόμων συνολο-συσχετιστική μνήμη ή εναλλακτικά ότι περιέχει m σύνολα με 1 μπλοκ ανά σύνολο. Αντίστοιχα, μία πλήρως συσχετιστική κρυφή μνήμη με μέγεθος m μπλοκ μπορεί να θεωρηθεί ότι είναι μια m δρόμων συνολο-συσχετιστική ή εναλλακτικά ότι περιέχει 1 σύνολο με m μπλοκ ανά σύνολο.



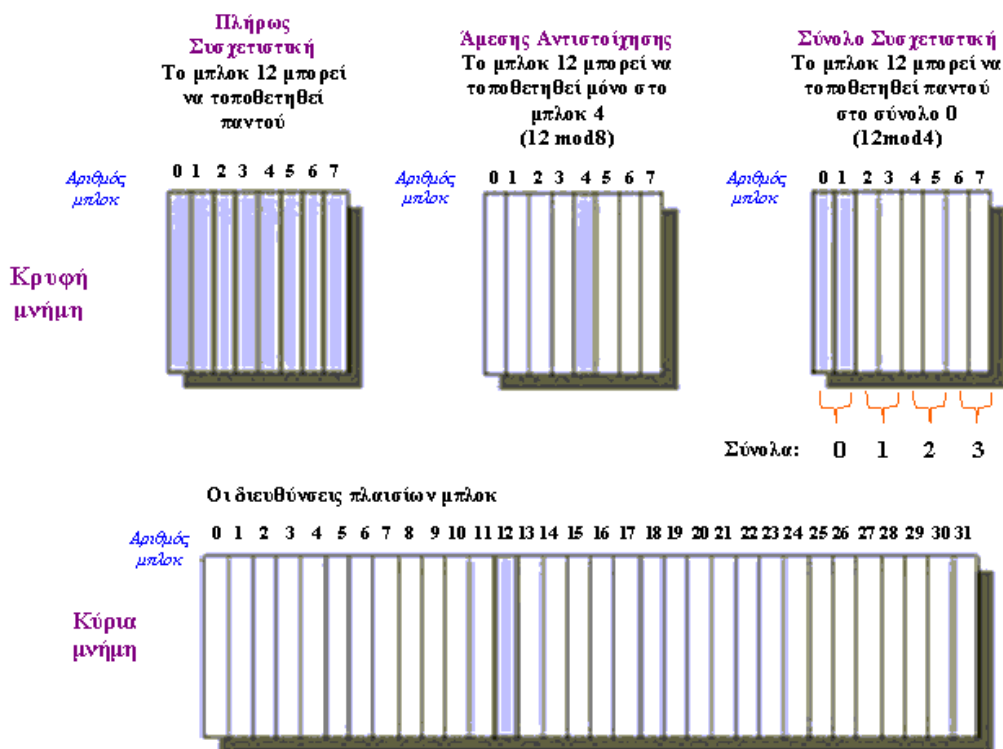
Παράδειγμα 1^ο

Θεωρούμε μία κρυφή μνήμη με 8 μπλοκ και μια κύρια μνήμη που έχει 32 μπλοκ. Η συνολο-συσχετιστική οργάνωση έχει 4 σύνολα με 2 μπλοκ ανά σύνολο. Υποθέτουμε ότι δεν υπάρχει τίποτα στη κρυφή μνήμη και ζητείται η διεύθυνση του μπλοκ της κρυφής μνήμης που προσδιορίζει το μπλοκ 12 της κύριας μνήμης.

Απάντηση

- Στην περίπτωση της πλήρως συσχετιστικής, το μπλοκ 12 της κύριας μνήμης μπορεί να μπει στη θέση οποιουδήποτε από τα 8 μπλοκ της κρυφής μνήμης.
- Με την άμεση αντιστοίχιση, το μπλοκ 12 μπορεί να τοποθετηθεί μόνο στο μπλοκ 4 ($12 \bmod 8$) της κρυφής μνήμης.
- Στην περίπτωση συνολο-συσχετιστικής κρυφής μνήμης το μπλοκ επιτρέπεται να τοποθετηθεί οπουδήποτε μέσα στο σύνολο 0 ($12 \bmod 4$). Με δύο μπλοκ ανά σύνολο, αυτό σημαίνει ότι το μπλοκ 12 μπορεί να τοποθετηθεί ή στο μπλοκ 0 ή στο μπλοκ 1 του συνόλου 0 της κρυφής μνήμης.

Προκύπτει το σχήμα 4.2.3, στο οποίο παριστάνονται οι πιθανές τοποθετήσεις των μπλοκ ανάλογα με το είδος της απεικόνισης για την κρυφή μνήμη:



Σχήμα 4.2.3 – Υποθέτουμε ότι η κρυφή μνήμη είναι άδεια. Οι 3 τύποι οργάνωσης της κρυφής μνήμης φαίνονται από τα αριστερά προς τα δεξιά. Η σύνολο συσχετιστική, συνδυάζει κάποια από τα χαρακτηριστικά της άμεσης αντιστοίχισης και της πλήρως συσχετιστικής. Στην πράξη οι κρυφές μνήμες περιέχουν εκατοντάδες μπλοκ και οι μνήμες κάποια εκατομμύρια από αυτά. Η σύνολο-συσχετιστική οργάνωση έχει 4 σύνολα με 2 μπλοκ ανά σύνολο (2-δρόμων συνολοσυσχετιστική).



Παράδειγμα 2^ο

Θεωρούμε ότι έχουμε έναν υπολογιστή που διαθέτει:

- Κύρια μνήμη με μέγεθος 32 bytes
- Κρυφή μνήμη με μέγεθος 8 bytes
- Το μπλοκ της κρυφής μνήμης έχει μέγεθος 1 byte

Εξετάζουμε τις τεχνικές αντιστοίχισης για το μπλοκ της κύριας μνήμης με διεύθυνση 5.

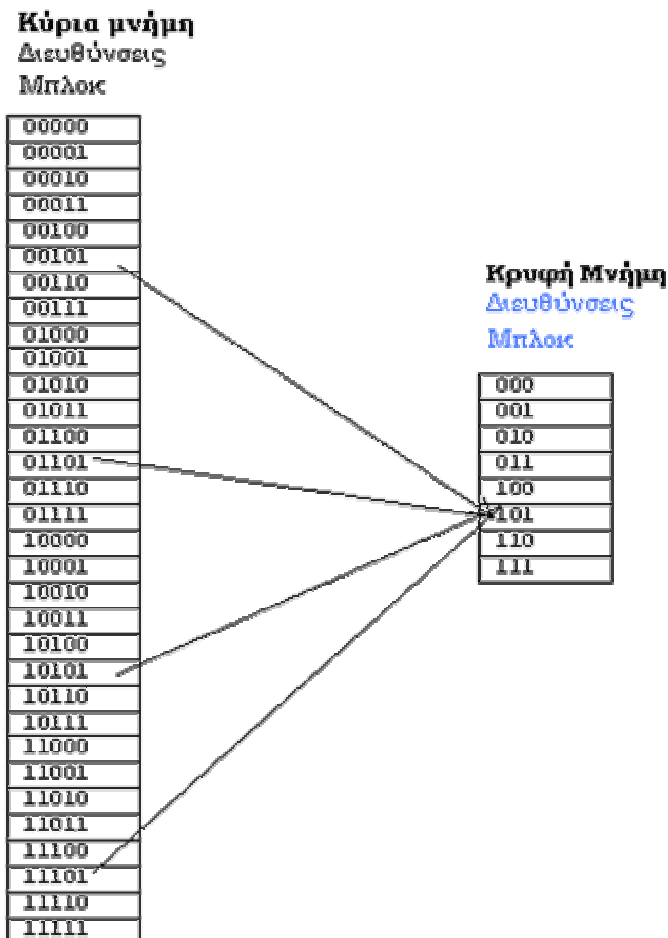
Απάντηση

▪ Κρυφή μνήμη Άμεσης αντιστοίχισης

Το μπλοκ της κύριας μνήμης με διεύθυνση 5 (ή 00101₂ στο δυαδικό), θα αντιστοιχηθεί στο μπλοκ της κρυφής μνήμης με διεύθυνση 5 MOD 8 = 5 (ή 101₂).

Στο ίδιο μπλοκ όμως της κρυφής μνήμης μπορούν να αντιστοιχηθούν και τα μπλοκ της κύριας

μνήμης με διεύθυνση 13 (ή 01101₂), 21 (ή 10101₂), 29 (ή 11101₂) δεδομένου ότι 13 MOD 8 = 5, 21 MOD 8 = 5 και 29 MOD 8 = 5.



Ένα ερώτημα που δημιουργείται είναι πόσα μπλοκ της κύριας μνήμης μπορούν να αντιστοιχηθούν στο ίδιο μπλοκ της κρυφής μνήμης.

Η απάντηση βρίσκεται στη διαφωνία του αριθμού των μπλοκ της κύρια μνήμης με τον αριθμό των μπλοκ της κρυφής μνήμης. Δηλαδή $32/8 = 4$.

Πως όμως ο υπολογιστής θα υπολογίσει από την διεύθυνση του μπλοκ της κύριας μνήμης την διεύθυνση του μπλοκ της κρυφής που θα πρέπει να γίνει η αντιστοίχιση, δεδομένου ότι η διευθύνσεις είναι στο δυαδικό σύστημα;

Παρατηρούμε ότι η δυαδική διεύθυνση 101₂ του μπλοκ της κρυφής μνήμης μπορεί να προσδιοριστεί από τα τρία λιγότερα σημαντικά ψηφία των διευθύνσεων που έχουν τα μπλοκ της κύριας μνήμης (00101₂, 01101₂, 10101₂, 11101₂) τα οποία μπορούν να αντιστοιχηθούν σε αυτό. Δηλαδή στο συγκεκριμένο παράδειγμα τα τρία λιγότερο σημαντικά ψηφία κάθε διεύθυνσης ενός μπλοκ της κύριας μνήμης .. καθορίζουν τη διεύθυνση του μπλοκ της

κρυφής στο οποίο μπορεί να αντιστοιχηθεί. Τα ψηφία αυτά αποτελούν το πεδίο **Δείκτη** της διεύθυνσης.

Ένα επιπλέον πρόβλημα είναι ότι επειδή σε κάθε μπλοκ της κρυφής μνήμης μπορούν να αντιστοιχηθούν πολλά μπλοκ της κύριας, η κρυφή μνήμη πρέπει να έχει την ικανότητα να διαπιστώσει ποιο από τα μπλοκ της κύριας έχει αντιστοιχηθεί. Το πρόβλημα λύνεται με την ύπαρξη ενός πεδίου **ετικέτας** στην διεύθυνση κάθε μπλοκ της κρυφής μνήμης.

Στο συγκεκριμένο παράδειγμα τα δύο περισσότερο σημαντικά ψηφία της διεύθυνσης του μπλοκ της κύριας μνήμης μπορούν να χρησιμοποιηθούν για τον προσδιορισμό του συγκεκριμένου μπλοκ που έχει αντιστοιχηθεί.



Στην περίπτωση που το μέγεθος του μπλοκ της κρυφής μνήμης είναι μεγαλύτερο από ένα byte, τότε τα λιγότερο σημαντικά ψηφία της διεύθυνσης προσδιορίζουν το πεδίο μετατόπισης, δηλαδή τη θέση του byte μέσα στο μπλοκ. Από το υπόλοιπο τμήμα της διεύθυνσης, προσδιορίζεται το πεδίο δείκτη και το πεδίο ετικέτας.

▪ **Κρυφή μνήμη Πλήρως Συσχετιστική (fully associative)**

Σε αυτό το είδος κρυφής μνήμης σε κάθε μπλοκ που περιέχει μπορούν να αντιστοιχηθούν όλα τα

μπλοκ της κύριας μνήμης. Συνεπώς το μπλοκ της κύριας μνήμης με διεύθυνση 5 μπορεί να τοποθετηθεί σε οποιοδήποτε μπλοκ της κρυφής μνήμης.

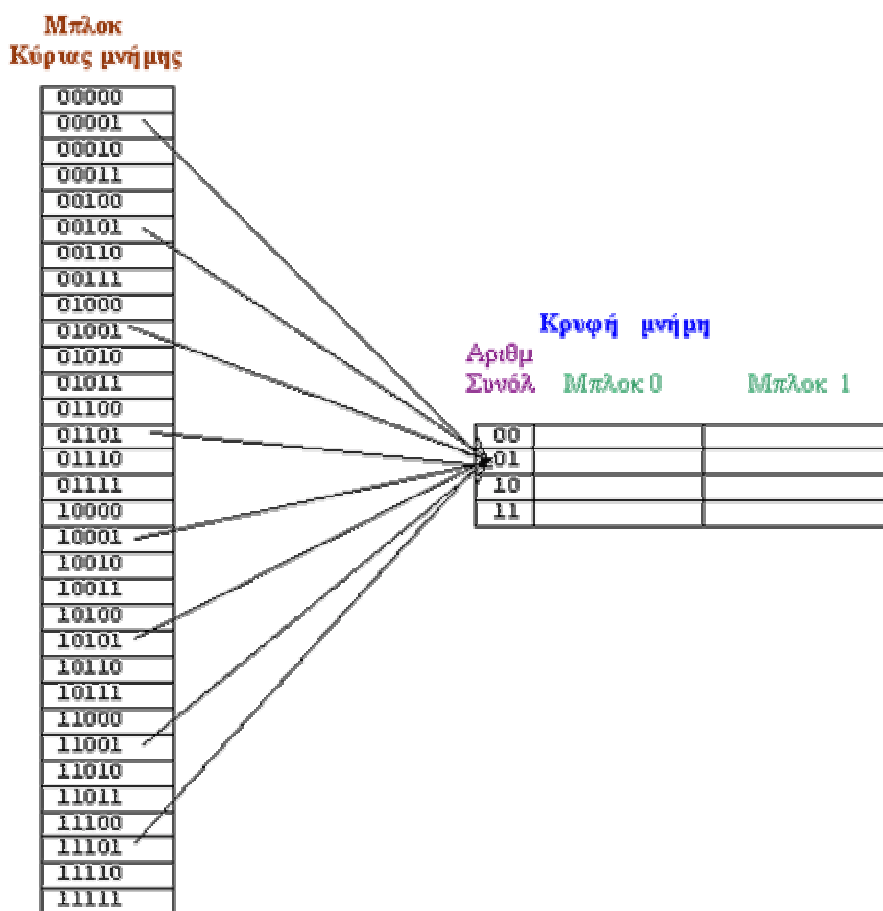
▪ **Κρυφή μνήμη Συνολο-συσχετιστική 2- δρόμων**

Η συνολο-συσχετιστική κρυφή μνήμη 2 δρόμων είναι οργανωμένη σε σύνολα και το κάθε σύνολο περιέχει δύο μπλοκ.

Συνεπώς στην κρυφή μνήμη του παραδείγματός μας υπάρχουν $8/2 = 4$ σύνολα .

Το μπλοκ της κύριας μνήμης με διεύθυνση 5 (ή 00101_2) θα αντιστοιχηθεί στο σύνολο της κρυφής μνήμης με διεύθυνση $5 \text{ MOD } 4 = 1$ (ή 01_2). Μέσα στο σύνολο αυτό μπορεί να αντιστοιχηθεί σε οποιοδήποτε από τα δύο μπλοκ που διαθέτει το σύνολο.

Στο ίδιο σύνολο όμως της κρυφής μνήμης μπορούν να αντιστοιχηθούν και τα μπλοκ της κύριας μνήμης με διεύθυνση 1 (ή 00001_2), 9 (ή 01001_2), 13 (ή 01101_2), 17 (ή 10001_2), 21 (ή 10101_2), 25 (ή 11001_2) και 29 (ή 11101_2) δεδομένου ότι η διαίρεση των αριθμών 1, 5, 9, 13, 17, 21, 25, και 29 με το 4 μας δίνουν υπόλοιπο 1.



Παρατηρούμε ότι η δυαδική διεύθυνση 01 του συνόλου της κρυφής μνήμης μπορεί να προσδιοριστεί από τα δύο λιγότερα σημαντικά ψηφία των διευθύνσεων που έχουν τα μπλοκ της κύριας μνήμης ($00001_2, 00101_2, 01001_2, 01101_2, 10001_2, 10101_2, 11001_2, 11101_2$) που μπορούν να αντιστοιχηθούν σε αυτό το σύνολο. Τα δύο αυτά ψηφία αποτελούν το πεδίο του δείκτη. Τα τρία περισσότερο σημαντικά ψηφία κάθε διεύθυνσης μπορούν να χρησιμοποιηθούν για τον προσδιορισμό του συγκεκριμένου μπλοκ που έχει αντιστοιχηθεί και αποτελούν το πεδίο ετικέτας. 🏷️

▣ Η διεύθυνση των μπλοκ στην κρυφή μνήμη

Κάθε μπλοκ της κρυφής μνήμης διαθέτει μια διεύθυνση. Ένα τμήμα της διεύθυνσης αυτής χρησιμοποιείται για την διευθυνσιοδότηση (αρίθμηση) των μπλοκ που διαθέτει η κρυφή μνήμη. Το τμήμα αυτό της διεύθυνσης ενός μπλοκ ονομάζεται **πεδίο δείκτη**. Στην περίπτωση που τα μπλοκ είναι οργανωμένα σε σύνολα (συνολο-συσχετιστική κρυφή μνήμη) το πεδίο δείκτη παρέχει την αρίθμηση των συνόλων. Το μέγεθός του πεδίου δείκτη εξαρτάται από το πλήθος των μπλοκ ή των συνόλων που διαθέτει η κρυφή μνήμη.

Για να μπορεί να διαπιστωθεί πιο συγκεκριμένο μπλοκ της κύριας μνήμης έχει ήδη μεταφερθεί σε ένα μπλοκ της κρυφής μνήμης, η διεύθυνση κάθε μπλοκ της κρυφής μνήμης περιέχει επιπλέον ένα πεδίο **ετικέτας**.



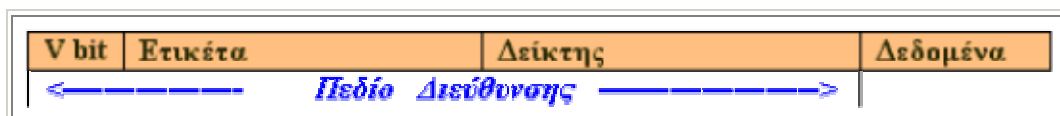
Το πεδίο ετικέτας χρησιμοποιείται κατά τη λειτουργία του εντοπισμού (συγκρίνεται) και ενημερώνεται κατά τη λειτουργία της τοποθέτησης με την ετικέτα του νέου μπλοκ που τοποθετείται στη θέση αυτή.

Στην κρυφή μνήμη όμως μπορούν να υπάρχουν μπλοκ από προηγούμενα προγράμματα που έχει τελειώσει η εκτέλεσή τους και να έχουν την ίδια ετικέτα με αυτή που αναζητείται. Συνεπώς η κρυφή μνήμη πρέπει να έχει την ικανότητα να διαπιστώσει αν τα μπλοκ που περιέχονται ήδη σε αυτή ανήκουν στο πρόγραμμα που εκτελείται ή σε παλαιότερα προγράμματα. Το πρόβλημα αυτό λύνεται με την ύπαρξη ενός **ψηφίου εγκυρότητας** (*valid bit*).



Το ψηφίο εγκυρότητας (Valid bit) ελέγχεται κατά τη λειτουργία του εντοπισμού και όταν έχει τεθεί (έχει τιμή 1) τότε το μπλοκ αυτό ανήκει στα τρέχον πρόγραμμα διαφορετικά (όταν έχει τιμή 0) το μπλοκ ανήκει σε παλαιότερο πρόγραμμα. Το ψηφίο αυτό τίθεται κατά τη λειτουργία της τοποθέτησης. Κατά την εκκίνηση ενός νέου προγράμματος το πεδίο εγκυρότητας όλων των μπλοκ της κρυφής μνήμης μηδενίζεται.

Συνεπώς κάθε μπλοκ της κρυφής μνήμης έχει την παρακάτω γενική δομή:



Τα τρία πρώτα πεδία αποτελούν το πεδίο της διεύθυνσης του μπλοκ



Παράδειγμα 3^ο

Συνεχίζοντας το παράδειγμα με τον υπολογιστή που διαθέτει κύρια μνήμη 32 bytes, κρυφή μνήμη 8 bytes και μέγεθος μπλοκ 1 byte, για τη δομή ενός μπλοκ της κρυφής μνήμης έχουμε:

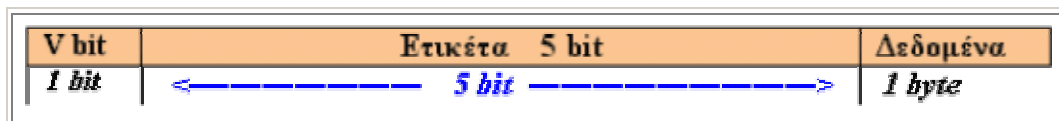
Κρυφή Μνήμη άμεσης αντιστοίχισης

Τα μπλοκ της κρυφής μνήμης θα έχουν την παρακάτω δομή :



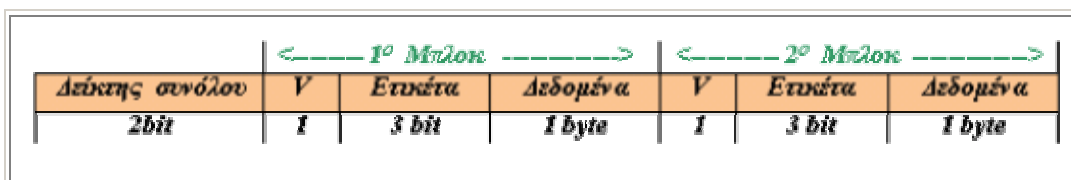
Κρυφή μνήμη πλήρως συσχετιστική

Τα μπλοκ της κρυφής μνήμης θα έχουν την παρακάτω δομή :



Συνολο-συσχετιστική κρυφή μνήμη 2 δρόμων

Σε αυτό το είδος της κρυφής μνήμης τα μπλοκ της κρυφής μνήμης είναι οργανωμένα σε σύνολα και το κάθε σύνολο περιέχει δύο μπλοκ. Συνεπώς τα μπλοκ αυτής της κρυφής μνήμης θα έχουν την παρακάτω δομή :



Η λειτουργία του εντοπισμού

Η λειτουργία του εντοπισμού, όπως έχει αναφερθεί, αρχίζει όταν ο επεξεργαστής εκτελεί εντολή που απαιτεί πρόσβαση στο περιεχόμενο μιας διεύθυνσης της κύριας μνήμης και η αίτηση αυτή μεταφέρεται στην κρυφή μνήμη. Η λειτουργία του εντοπισμού σκοπό έχει να διαπιστώσει αν το μπλοκ της κύριας μνήμης που περιέχει τη ζητούμενη διεύθυνση, έχει ήδη μεταφερθεί σε ένα αντίστοιχο μπλοκ της κρυφής και συνεπώς το περιεχόμενο της ζητούμενης διεύθυνσης υπάρχει ήδη μέσα στην κρυφή μνήμη. Τα βήματα του εντοπισμού είναι:

1. Τμηματοποίηση

Η ζητούμενη διεύθυνση της κύριας μνήμης χωρίζεται σε τμήματα αρχίζοντας από το λιγότερο σημαντικό bit της διεύθυνσης.

- i. Αρχικά χωρίζεται το τμήμα της **μετατόπισης** το οποίο χρησιμοποιείται μόνο σε περίπτωση επιτυχίας για την επιλογή του περιεχομένου της αντίστοιχης διεύθυνσης μέσα από το μπλοκ π.χ για την επιλογή του 5^{ου} byte μέσα από ένα μπλοκ με μέγεθος 8 byte. Ο αριθμός των bit που αποτελούν αυτό το πεδίο εξαρτάται από το μέγεθος του μπλοκ.



Το υπόλοιπο τμήμα της διεύθυνσης ανάλογα με το είδος της κρυφής μνήμης μπορεί να διαιρεθεί σε δύο τμήματα το τμήμα **ετικέτας** και το τμήμα του **δείκτη**.



- ii. Στη συνέχεια χωρίζεται το τμήμα **δείκτη** το οποίο χρησιμοποιείται για την επιλογή του συνόλου(ή του μπλοκ) της κρυφής μνήμης που περιέχει τα μπλοκ που πρέπει να ελεγχθούν. Το μέγεθος του πεδίου αυτού εξαρτάται από τον αριθμό των συνόλων/μπλοκ της κρυφής μνήμης.
 - iii. Τα υπόλοιπα πλέον σημαντικά ψηφία της διεύθυνσης αποτελούν το τμήμα ετικέτας.
2. **Επιλογή του συνόλου ή του μπλοκ**
Από το πεδίο δείκτη υπολογίζεται το σύνολο ή το μπλοκ της κρυφής μνήμης.

3. Σύγκριση

Το πεδίο ετικέτας του συγκεκριμένου μπλοκ που έχει επιλεγεί από το πεδίο δείκτη της διεύθυνσης συγκρίνεται με το αντίστοιχο τμήμα ετικέτας της διεύθυνσης της κύριας μνήμης για να διαπιστωθεί αν είναι ίσα. Ειδικότερα για τα διάφορα είδη κρυφής μνήμης ισχύει:

- Κρυφή μνήμη **πλήρως συσχετιστική**: Ο έλεγχος της **ετικέτας** γίνεται ταυτόχρονα σε όλα τα μπλοκ της κρυφής μνήμης.
- Κρυφή μνήμη **συνολο-συσχετιστική**:. Ο έλεγχος της **ετικέτας** γίνεται ταυτόχρονα σε όλα τα μπλοκ ενός συνόλου.

Ταυτόχρονα με τον έλεγχο του πεδίου ετικέτας ενός μπλοκ ελέγχεται και το πεδίο εγκυρότητας του

➡ Αποτέλεσμα εντοπισμού

Αν για το μπλοκ (ή για τα μπλοκ του συνόλου) που καθορίζεται, βρεθεί ότι η ετικέτα του είναι ίση με το πεδίο ετικέτας που προκύπτει από την τμηματοποίηση της ζητούμενης διεύθυνσης της κύριας μνήμης και το ψηφίο εγκυρότητάς του μπλοκ είναι ίσο με ένα ($V=1$) τότε έχουμε **επιτυχία** της κρυφής μνήμης, διαφορετικά αν το ψηφίο εγκυρότητας είναι μηδέν ($V=0$) ή τα πεδία ετικέτας διαφέρουν τότε έχουμε **αποτυχία** της κρυφής μνήμης.

Αν το αποτέλεσμα του εντοπισμού είναι επιτυχία τότε αρχίζει η λειτουργία της **ανάγνωσης**, ενώ αν είναι αποτυχία αρχίζει η λειτουργία της **τοποθέτησης**.



Παράδειγμα

Υποθέτουμε ότι διαθέτουμε ένα υπολογιστή με τα παρακάτω στοιχεία: η κύρια μνήμη έχει μέγεθος 1 MB (2^{20} bytes), η κρυφή μνήμη έχει μέγεθος 1 KB (2^{10}) και το μέγεθος των μπλοκ που περιέχει, είναι 8 bytes (2^3 bytes). Αυτό σημαίνει ότι οι διευθύνσεις της κύριας μνήμης έχουν μήκος 20 bit. Η κρυφή μνήμη αποτελείται από $2^{10}/2^3 = 2^7$ μπλοκ ή 128 μπλοκ. Να προσδιορίσετε τα βήματα του εντοπισμού για κάθε μία από τις παρακάτω οργανώσεις της κρυφής μνήμης για αίτηση του επεξεργαστή από τη διεύθυνση κύριας μνήμης $(5F012)_{16}$.

Απάντηση

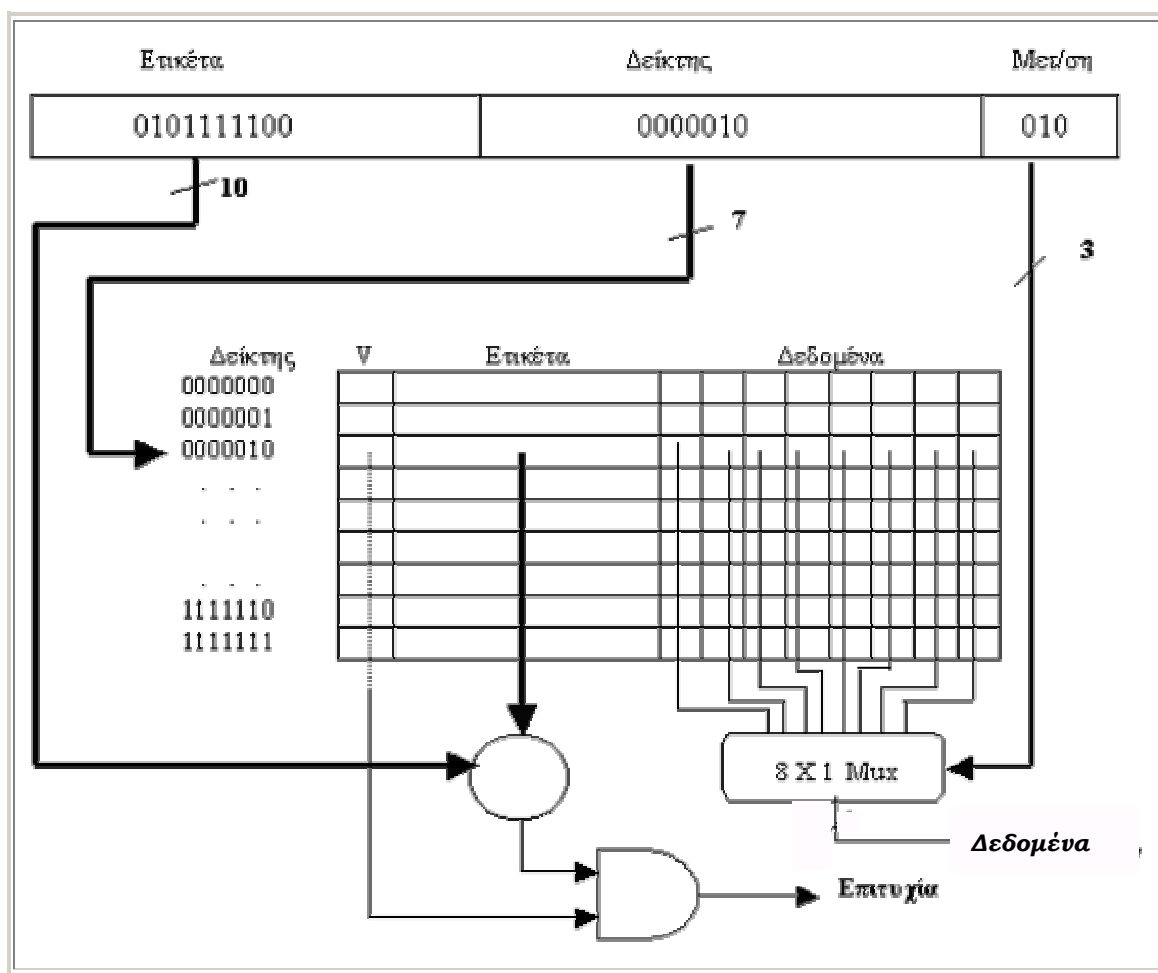
➔ **Κρυφή Μνήμη άμεσης αντιστοίχισης**

Η τμηματοποίηση μιας διεύθυνσης της κύριας μνήμης αρχίζει από τα λιγότερο σημαντικά ψηφία της διεύθυνσης και σύμφωνα με τα παραπάνω γίνεται ως εξής:

Από τα 20 bits μιας διεύθυνσης μνήμης τα 3 λιγότερο σημαντικά αποτελούν το πεδίο μετατόπισης και καθορίζουν την διεύθυνση του byte μέσα στο μπλοκ της κρυφής μνήμης. Τα επόμενα 7 αποτελούν το πεδίο δείκτη και καθορίζουν το μπλοκ της κρυφής μνήμης που αντιστοιχεί στη διεύθυνση αυτή ενώ τα υπόλοιπα 10 bits (20-7-3) καθορίζουν την ετικέτα της διεύθυνσης αυτής.

Η λειτουργία του Εντοπισμού για μια αίτηση ανάγνωσης του επεξεργαστή από τη διεύθυνση κύριας μνήμης $(5F012)_{16}$ η οποία αντιστοιχεί στην δυαδική διεύθυνση $(0101\ 1111\ 00\ 00\ 0001\ 0010)_2$ γίνεται με τα παρακάτω βήματα:

1. **Τμηματοποίηση της διεύθυνσης**



2. Επιλογή του μπλοκ με διεύθυνση 0000010_2
3. Σύγκριση της ετικέτας του μπλοκ με την ετικέτα 0101111100_2 και έλεγχος του V.

Αποτέλεσμα του Εντοπισμού

Εάν η ετικέτα του μπλοκ $(0000010)_2$ της κρυφής μνήμης ταυτίζεται με την ετικέτα $(0101111100)_2$ της διεύθυνσης της κύριας μνήμης και το ψηφίο εγκυρότητας έχει τεθεί ($V=1$) τότε έχουμε επιτυχία διαφορετικά έχουμε αποτυχία της κρυφής μνήμης.

➤ **Κρυφή Μνήμη πλήρως συσχετιστική**

1. Τμηματοποίηση της διεύθυνσης

Για την τμηματοποίηση μιας διεύθυνσης της κύριας μνήμης με μήκος 20 bits επιλέγονται τα 3 λιγότερα σημαντικά bits, τα οποία αποτελούν το πεδίο μετατόπισης για τον καθορισμό του byte μέσα στο μπλοκ της κρυφής μνήμης, ενώ τα υπόλοιπα 17 bit αποτελούν το πεδίο ετικέτας.



2. Δεν γίνεται επιλογή μπλοκ δεδομένου ότι σ' αυτήν την οργάνωση η αντιστοίχιση γίνεται σε οποιοδήποτε ελεύθερο μπλοκ της κρυφής μνήμης
3. Γίνεται ταυτόχρονα σύγκριση της ετικέτας όλων των μπλοκ της κρυφής μνήμης με την ετικέτα $(01011111000000010)_2$ της διεύθυνσης της κύριας μνήμης και έλεγχος του ψηφίου εγκυρότητας.

Αποτέλεσμα του Εντοπισμού

Εάν η ετικέτα κάποιου μπλοκ της κρυφής μνήμης ταυτίζεται με την ετικέτα $(01011111000000010)_2$ της διεύθυνσης της κύριας μνήμης και το ψηφίο εγκυρότητας έχει τεθεί ($V=1$) τότε έχουμε επιτυχία διαφορετικά έχουμε αποτυχία της κρυφής μνήμης.

➤ **Σύνολο-Συσχετιστική Κρυφή Μνήμη 2 δρόμων**

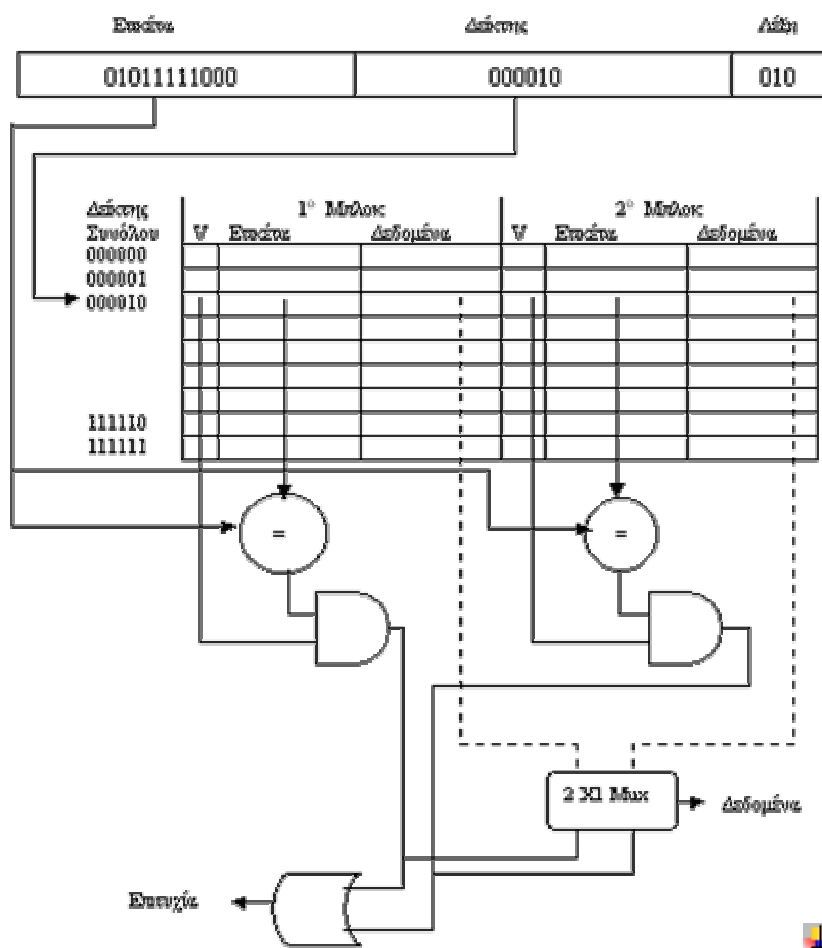
1. Τμηματοποίηση της διεύθυνσης

Στην συνολο-συσχετιστική μνήμη 2 δρόμων η κρυφή μνήμη είναι οργανωμένη σε σύνολα και το κάθε σύνολο περιέχει δύο μπλοκ. Συνεπώς στην κρυφή μνήμη του παραδείγματός μας υπάρχουν 2^6 (64) σύνολα. Από τα 20 bits μιας διεύθυνσης μνήμης τα 3 λιγότερο σημαντικά αποτελούν το πεδίο μετατόπισης και καθορίζουν την διεύθυνση του byte μέσα στο μπλοκ της κρυφής μνήμης. Τα επόμενα 6 bits καθορίζουν το σύνολο της κρυφής μνήμης που αντιστοιχεί στη διεύθυνση αυτή ενώ τα υπόλοιπα 11 bits $(20-6-3)$ καθορίζουν την ετικέτα της διεύθυνσης αυτής (βλέπε ακόλουθο σχήμα).

2. Επιλογή του συνόλου με διεύθυνση $(000010)_2$
3. Ελέγχεται ταυτόχρονα η ετικέτα στα δύο μπλοκ του συνόλου $(000010)_2$ της κρυφής μνήμης και το ψηφίο εγκυρότητας.

Αποτέλεσμα του Εντοπισμού

Εάν η ετικέτα κάποιου από τα δύο μπλοκ του συνόλου $(000010)_2$ της κρυφής μνήμης ταυτίζεται με την ετικέτα $(01011111000)_2$ της διεύθυνσης της κύριας μνήμης και το ψηφίο εγκυρότητας έχει τεθεί ($V=1$) τότε έχουμε επιτυχία διαφορετικά έχουμε αποτυχία της κρυφής μνήμης.



Η λειτουργία της τοποθέτησης και της αντικατάστασης

☐ Η λειτουργία της τοποθέτησης

Η λειτουργία της τοποθέτησης αρχίζει όταν το αποτέλεσμα του εντοπισμού είναι αποτυχία κρυφής μνήμης, η οποία οφείλεται στο ότι το μπλοκ που έχει ήδη επιλεγεί κατά τη λειτουργία του εντοπισμού δεν ανήκει στο τρέχον πρόγραμμα (δηλαδή όταν $V=0$). Η λειτουργία της τοποθέτησης πραγματοποιείται με τα παρακάτω βήματα:

1. Το μπλοκ της κύριας μνήμης που περιέχει τη ζητούμενη διεύθυνση μεταφέρεται στην κρυφή μνήμη στη διεύθυνση του μπλοκ που έχει ήδη επιλεγεί κατά τη λειτουργία του εντοπισμού. Ταυτόχρονα, εάν ο επεξεργαστής έχει ζητήσει να διαβάσει το περιεχόμενο της ζητούμενης διεύθυνσης, τότε αυτό τοποθετείται στις γραμμές δεδομένων που συνδέουν την κύρια μνήμη με τον επεξεργαστή.

2. Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας του μπλοκ της κρυφής μνήμης από το αντίστοιχο πεδίο του μπλοκ της κύριας μνήμης που μεταφέρεται και τίθεται η τιμή του ψηφίου εγκυρότητας ($V=1$).



Όταν το αποτέλεσμα του εντοπισμού είναι αποτυχία κρυφής μνήμης, η οποία οφείλεται στο ότι το μπλοκ που έχει ήδη επιλεγεί κατά τη λειτουργία του εντοπισμού ανήκει στο τρέχον πρόγραμμα (δηλαδή όταν $V=1$), τότε αντί για τοποθέτηση γίνεται αντικατάσταση.

Η λειτουργία της αντικατάστασης

Η λειτουργία της αντικατάστασης αρχίζει όταν το αποτέλεσμα του εντοπισμού είναι αποτυχία κρυφής μνήμης, η οποία οφείλεται στο ότι το πεδίο ετικέτας του μπλοκ της κρυφής μνήμης δεν ταυτίζεται με εκείνο της κύριας μνήμης, ενώ το μπλοκ που έχει ήδη επιλεγεί κατά τη λειτουργία του εντοπισμού, ανήκει στο τρέχον πρόγραμμα (δηλαδή όταν $V=1$). Η λειτουργία της αντικατάστασης πραγματοποιείται με τα παρακάτω βήματα:

1. Επιλογή του προς αντικατάσταση μπλοκ

Η επιλογή του μπλοκ της κρυφής μνήμης που θα αντικατασταθεί εξαρτάται από το είδος της κρυφής μνήμης. Συγκεκριμένα:

- Αν η κρυφή μνήμη είναι άμεσης αντιστοίχισης τότε το μπλοκ της κύριας μνήμης αντιστοιχείται μόνο σε ένα μπλοκ της κρυφής μνήμης
- Αν η κρυφή μνήμη είναι πλήρως συσχετιστική ή συνολο-συσχετιστική τότε υπάρχουν πολλά μπλοκ που μπορούν να επιλεγούν για αντικατάσταση. Στην περίπτωση αυτή οι τεχνικές αντικατάστασης είναι οι ακόλουθες:
 - i. **Τυχαία (Random):** Τα υπονήφια μπλοκ επιλέγονται τυχαία για να έχουμε όσο το δυνατό ομοιόμορφη κατανομή.
 - ii. **Λιγότερο πρόσφατα χρησιμοποιημένο (LRU - Least Recently Used):** Αντικαθίσταται κάθε φορά το μπλοκ που έχει το μεγαλύτερο χρόνο παραμονής στην κρυφή μνήμη χωρίς να γίνει αίτηση του επεξεργαστή σε αυτό, για ανάγνωση ή εγγραφή.
 - iii. **Πρώτη Μέσα Πρώτη Έξω (FIFO - First In First Out):** Η αντικατάσταση γίνεται με βάση το χρόνο παραμονής. Δηλαδή αντικαθίσταται το μπλοκ που έχει το μεγαλύτερο χρόνο παραμονής στην κρυφή μνήμη. Ουσιαστικά η τεχνική FIFO αντικαθιστά τα μπλοκ της κρυφής μνήμης διαδοχικά με τη σειρά σύμφωνα με τη φυσική τους θέση.

2. Ακολουθούν τα βήματα της τοποθέτησης (όπως έχουν ήδη παρουσιαστεί)



Κατά τη λειτουργία της αντικατάστασης, εάν το μπλοκ που απομακρύνεται από την κρυφή μνήμη έχει τροποποιηθεί λόγω εγγραφής νέων δεδομένων σε αυτό, μετά από προηγούμενη αίτηση εγγραφής του επεξεργαστή, χωρίς να έχει ενημερωθεί η κύρια μνήμη (τεχνική επανεγγραφής), τότε το μπλοκ αυτό μεταφέρεται στην κύρια μνήμη πριν να αντικατασταθεί (βλέπε λειτουργία εγγραφής)



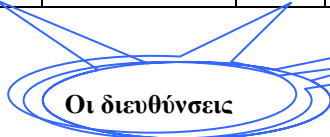
Παράδειγμα 1ο

Δίνεται μια σειρά από διευθύνσεις λέξεων κύριας μνήμης 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4,43, 5, 6, 9, 17, σε μία κύρια μνήμη η οποία έχει μέγεθος 64 λέξεων. Θεωρώντας μια κρυφή μνήμη άμεσης αντιστοίχισης με 16 μπλοκ μιας λέξης, τα οποία είναι αρχικά άδεια, χαρακτηρίστε κάθε αναφορά στη λίστα ως επιτυχία ή αποτυχία της κρυφής μνήμης και δείξτε το τελικό περιεχόμενο της κρυφής μνήμης.

Απάντηση

Αρχικά αναπαριστούμε τις διευθύνσεις της κύριας μνήμης με δυναδικούς αριθμούς των 6 bit, δεδομένου ότι το μέγεθος της κύρια μνήμης είναι 64 λέξεις:

1	000001	20	010100	9	001001	5	000101
4	000100	17	010001	11	001011	6	000110
8	001000	19	010011	4	000100	9	001001
5	000101	56	111000	43	101011	17	010001



Μια διεύθυνση X δείχνει στο μπλοκ ($X \bmod 16$) της κρυφής μνήμης, όπου $16=2^4$ είναι ο αριθμός των μπλοκ στην κρυφή μνήμη. Συνεπώς, τα 4 τελευταία ψηφία κάθε διεύθυνσης ορίζουν σε ποιο μπλοκ της κρυφής μνήμης αναφέρονται και αποτελούν το πεδίο του δείκτη (index field). Για παράδειγμα, η διεύθυνση 010011_2 (η διεύθυνση 19 της κύριας μνήμης) αντιστοιχεί στο μπλοκ $(010011_2) \bmod 16 = (19 \bmod 16) = 0011_2$ (η διεύθυνση 3 της κρυφής μνήμης). Όσον αφορά το πεδίο ετικέτας (tag field), αυτό ορίζεται από τα 2 αριστερότερα ψηφία της διεύθυνσης. Συγκεκριμένα, στο προηγούμενο παράδειγμα το πεδίο ετικέτας είναι το 01.

Τέλος, ένας τρόπος για να γνωρίζουμε αν ένα μπλοκ κρυφής μνήμης περιέχει ή όχι έγκυρες πληροφορίες, δηλαδή εάν τα περιεχόμενά του ανήκουν στο πρόγραμμα που εκτελείται, είναι να προσθέσουμε ένα ψηφίο εγκυρότητας (Valid bit). Έτσι, λοιπόν, οι καταστάσεις της κρυφής μνήμης, με τη σειρά των διευθύνσεων της άσκησης, είναι οι εξής :

Δείκτης	V	ετικέτα	Δεδομένα
0000	N		
0001	N		
0010	N		
0011	N		
0100	N		
0101	N		
0110	N		

Δείκτης	V	Ετικέτα	Δεδομένα
0000	N		
0001	Y	00	Mem(000001)
0010	N		
0011	N		
0100	Y	00	Mem(000100)
0101	Y	00	Mem(000101)
0110	N		

0111	N		
1000	N		
1001	N		
1010	N		
1011	N		
1100	N		
1101	N		
1110	N		
1111	N		

Πίνακας Α

0111	N		
1000	Y	00	Mem(001000)
1001	N		
1010	N		
1011	N		
1100	N		
1101	N		
1110	N		
1111	N		

Πίνακας Β

Ο Πίνακας Α δείχνει την αρχική κατάσταση της κρυφής μνήμης, ενώ ο Πίνακας Β δείχνει την κατάσταση της κρυφής μνήμης μετά τη διαδοχική αναφορά των διευθύνσεων 1, 4, 8, 5 κατά τις οποίες έχουμε αποτυχία, αφού η κρυφή μνήμη είναι άδεια. Σειρά έχει η διεύθυνση 010100₂ (η διεύθυνση 20 της κύριας μνήμης) κατά την οποία έχουμε και πάλι αποτυχία κρυφής μνήμης, με τη διαφορά ότι τα δεδομένα Mem(010100) της συγκεκριμένης διεύθυνσης πρέπει να αντικαταστήσουν τα δεδομένα Mem(000100) που βρίσκονται στο μπλοκ 0100 της κρυφής μνήμης και ταυτόχρονα να ενημερώσουν το πεδίο ετικέτας που τώρα γίνεται 01. Έτσι η κατάσταση της κρυφής μνήμης είναι η ακόλουθη (βλέπε πίνακα Γ):

Το bit εγκυρότητας έχει τεθεί, άρα το μπλοκ έχει εγκυρη πληροφορία

Δείκτης	V	Ετικέτα	Δεδομένα
0000	N		
0001	Y	00	Mem(000001)
0010	N		
0011	N		
0100	Y	01	Mem(010100)
0101	Y	00	Mem(000101)
0110	N		
0111	N		
1000	Y	00	Mem(001000)
1001	N		
1010	N		
1011	N		
1100	N		
1101	N		
1110	N		
1111	N		

Πίνακας Γ

Ακολουθούν οι διευθύνσεις τις κύριας μνήμης 17, 19, 56, 9, 11 στις οποίες έχουμε αποτυχίες κρυφής μνήμης. Πιο συγκεκριμένα τα δεδομένα Mem(010001) της διεύθυνσης 17 αντικαθιστούν

τα ήδη υπάρχοντα Mem(000001) στο μπλοκ 0001 της κρυφής μνήμης, ενώ τα Mem(111000) της διεύθυνσης 56 αντικαθιστούν τα Mem(001001) της διεύθυνσης 8 στο μπλοκ 1000, ενώ ταυτόχρονα γίνεται ενημέρωση του πεδίου ετικέτας. Μετά την 11 έρχεται η διεύθυνση 4 κατά την οποία έχουμε αποτυχία, τα δεδομένα της οποίας Mem(000100) αντικαθιστούν τα Mem(010100) της 20 στο μπλοκ 0100. Ακολουθεί η 43, κατά την οποία έχουμε αποτυχία, τα δεδομένα της οποίας Mem(101011) αντικαθιστούν τα Mem(001011) της 11 στο μπλοκ 1011. Τέλος, έχουμε διαδοχικά τις διευθύνσεις 5(επιτυχία), 6(αποτυχία), 9(επιτυχία), 17(επιτυχία). Συνεπώς, η τελική κατάσταση της κρυφής μνήμης είναι η ακόλουθη :

Δείκτης	V	Ετικέτα	Δεδομένα
0000	N		
0001	Y	01	Mem(010001)
0010	N		
0011	Y	01	Mem(010011)
0100	Y	00	Mem(000100)
0101	Y	00	Mem(000101)
0110	Y	00	Mem(000110)
0111	N		
1000	Y	11	Mem(111000)
1001	Y	00	Mem(001001)
1010	N		
1011	Y	10	Mem(101011)
1100	N		
1101	N		
1110	N		
1111	N		

Πίνακας Δ



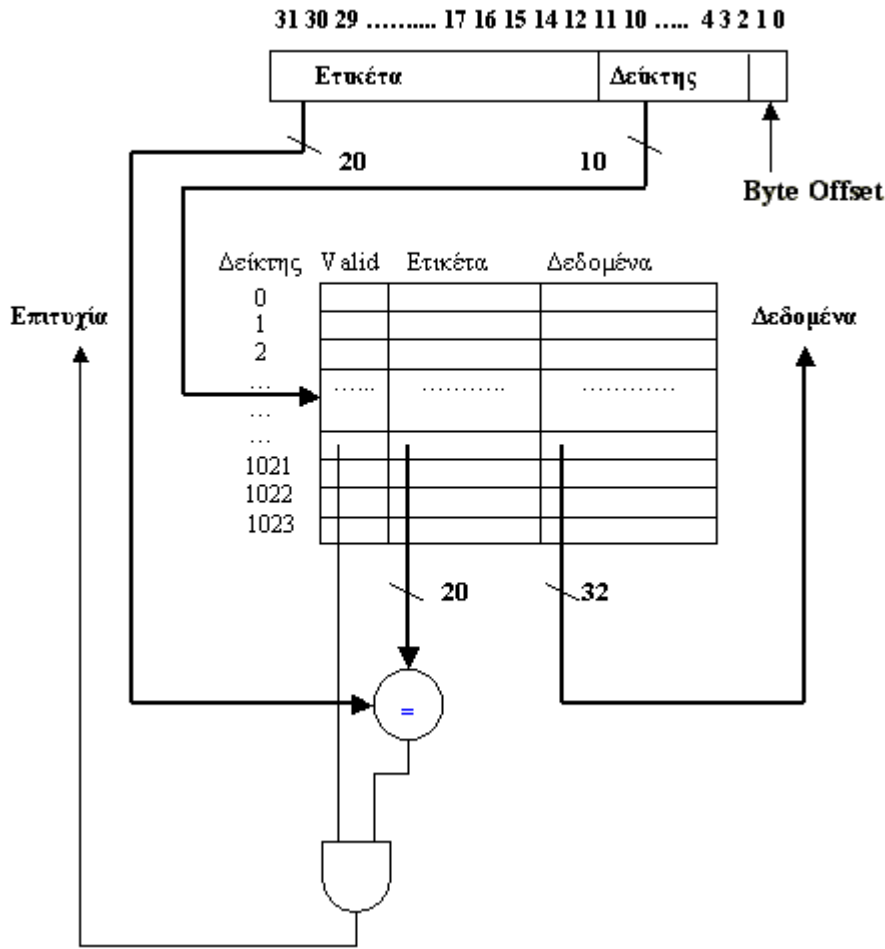
Γενικά, θεωρώντας μία διεύθυνση byte 32 bits (MIPS υπολογιστής), μία άμεσης αντιστοίχισης κρυφή μνήμη με μέγεθος 2^n λέξεις και με μπλοκ μιας λέξης απαιτεί πεδίο ετικέτας με μέγεθος $32 - (n + 2)$ bits, όπου 2 είναι τα bits που χρησιμοποιούνται για τη μετατόπιση byte και n τα bits για το δείκτη. Ο συνολικός αριθμός των bits σε μια άμεσης αντιστοίχισης κρυφή μνήμη είναι : $2^n \times (\text{μέγεθος μπλοκ} + \text{μέγεθος ετικέτας} + \text{μέγεθος πεδίου εγκυρότητας})$

Αφού το μπλοκ έχει μέγεθος 1 λέξη (32 bits) και μέγεθος διεύθυνσης 32 bits, ο αριθμός των bits στην κρυφή μνήμη είναι $2^n \times (32 + (32 - n - 2) + 1) = 2^n \times (63 - n)$.

Αν έχω μία κρυφή μνήμη με 64 Kbytes δεδομένα, τότε 64 Kbytes είναι 16 Kwords δηλαδή 2^{14} words (2^{14} λέξεις). Αρα, το συνολικό μέγεθος της κρυφής μνήμης είναι:

$$2^{14} \times (32 + (32 - 14 - 2) + 1) = 2^{14} \times 49 = 784 \times 2^{10} = 784 \text{ Kbits} = 98 \text{ KB}$$

Διεύθυνση



Η παραπάνω κρυφή μνήμη έχει 2^{10} ή 1024 λέξεις και μέγεθος μπλοκ 1 λέξη. Για το δείκτη χρησιμοποιούνται 10 bits, ενώ μένουν $32 - 10 - 2 = 20$ bits για την ετικέτα. Αν η ετικέτα και τα 20 αριστερά bits της διεύθυνσης είναι ίσα και το bit εγκυρότητας έχει τεθεί 1 ή Yes, τότε έχουμε επιτυχία. Σε διαφορετική περίπτωση έχουμε αποτυχία κρυφής μνήμης. 🚦



Παράδειγμα 2ο

Δίνεται μια σειρά από διευθύνσεις μπλοκ κύριας μνήμης: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4,43, 5, 6, 9, 17. Θεωρώντας μια κρυφή μνήμη άμεσης αντιστοίχισης με μέγιστο μέγεθος 16 λέξεις και μέγεθος μπλοκ 4 λέξεις, δείξτε τις επιτυχίες και αποτυχίες κρυφής μνήμης, καθώς και την τελική κατάσταση της κρυφής μνήμης;

Απάντηση

Ο γενικός τύπος υπολογισμού του μπλοκ της κρυφής μνήμης, στο οποίο θα τοποθετηθεί μια διεύθυνση είναι ο εξής:

$$\left(\frac{\text{Διεύθυνση Λέξης}}{\text{Λέξεις ανά μπλοκ}} \right) \bmod (\text{Αριθμός των μπλοκ της κρυφής μνήμης})$$

Αφού η κρυφή έχει μέγεθος 16 λέξεις και τα μπλοκ είναι 4 λέξεων, τότε ο αριθμός των μπλοκ στην κρυφή μνήμη είναι 4.

Συνεπώς ο αριθμός του μπλοκ που θα τοποθετηθεί κάθε διεύθυνση είναι :

$$\begin{aligned}
 [1/4] \bmod 4 = 0 \bmod 4 = 0, \text{ δηλαδή το } 00 \\
 [4/4] \bmod 4 = 1 \bmod 4 = 1, \text{ δηλαδή το } 01 \\
 [8/4] \bmod 4 = 2 \bmod 4 = 2, \text{ δηλαδή το } 10 \\
 [5/4] \bmod 4 = 1 \bmod 4 = 1, \text{ δηλαδή το } 01 \\
 [20/4] \bmod 4 = 5 \bmod 4 = 1, \text{ δηλαδή το } 01 \\
 [17/4] \bmod 4 = 4 \bmod 4 = 0, \text{ δηλαδή το } 00 \\
 [19/4] \bmod 4 = 4 \bmod 4 = 0, \text{ δηλαδή το } 00 \\
 [56/4] \bmod 4 = 14 \bmod 4 = 2, \text{ δηλαδή το } 10 \\
 [11/4] \bmod 4 = 2 \bmod 4 = 2, \text{ δηλαδή το } 10 \\
 [43/4] \bmod 4 = 10 \bmod 4 = 2, \text{ δηλαδή το } 10 \\
 [6/4] \bmod 4 = 1 \bmod 4 = 1, \text{ δηλαδή το } 01
 \end{aligned}$$

Η διεύθυνση όμως του μπλοκ δεν προσδιορίζει και τη θέση της λέξης, στην οποία γίνεται αναφορά, μέσα στο μπλοκ.

Ένας πλήρης τρόπος υπολογισμού προκύπτει από την τμηματοποίηση της διεύθυνσης. Υπενθυμίζεται ότι επειδή η μεγαλύτερη αναφορά σε διεύθυνση λέξης είναι το 56 για την αναπαράσταση των διευθύνσεων απαιτούνται 6 bit. Επειδή η κρυφή μνήμη είναι τεσσάρων λέξεων, σε κάθε μπλοκ μπορούν να μπουν έως 4 διαφορετικά δεδομένα. Συνεπώς τα 2 δεξιότερα ψηφία της διεύθυνσης ορίζουν σε ποια από τις 4 θέσεις των λέξεων αναφέρεται η διεύθυνση (**πεδίο μετατόπισης**). Επειδή δεν γίνεται αναφορά σε bytes δεν υπολογίζεται πεδίο για byte offset. Τα επόμενα 2 ψηφία προς τα αριστερά δηλαδή το 4^ο, 3^ο από δεξιά αποτελούν το **πεδίο δείκτη** κάθε διεύθυνσης. Για παράδειγμα, η διεύθυνση 101011₂ αναφέρεται στη θέση 4 (11) του δεύτερου (10) μπλοκ της κρυφής μνήμης. Τα 2 αριστερότερα ψηφία της διεύθυνσης 10 αποτελούν το **πεδίο ετικέτας**.

Σύμφωνα με τη λίστα των διευθύνσεων, αρχικά γίνεται αναφορά στην διεύθυνση 1 (ή 000001₂). Κατά την λειτουργία εντοπισμού αυτής της διεύθυνσης στην κρυφή μνήμη υπολογίζεται ότι θα πρέπει να ερευνηθεί το πρώτο μπλοκ (00). Στη διαδικασία αυτή ο εντοπισμός σημειώνει αποτυχία γιατί το πεδίο εγκυρότητας του πρώτου μπλοκ είναι 0. Η λέξη με διεύθυνση 1 θα τοποθετηθεί στο πρώτο μπλοκ (00) και στη θέση 01 της κρυφής μνήμης. Επειδή το μπλοκ είναι τεσσάρων λέξεων μαζί με αυτήν θα τοποθετηθούν και οι γειτονικές λέξεις με διευθύνσεις 0,2,3. Το πεδίο ετικέτας παίρνει την τιμή 00 και το ψηφίο εγκυρότητας γίνεται 1 (τίθεται).

Ακολουθεί η αναφορά στην διεύθυνση 4 (ή 000100₂). Σύμφωνα με την παραπάνω περιγραφόμενη διαδικασία εντοπισμού σημειώνεται αποτυχία και ακολουθεί η τοποθέτηση της λέξης αυτή στο δεύτερο μπλοκ (01) στη θέση 00. Στο πεδίο ετικέτας τίθεται η τιμή 00 και στο πεδίο εγκυρότητας το 1. Μαζί με αυτή τοποθετούνται και οι γειτονικές λέξεις της κύριας μνήμης με διευθύνσεις 5,6,7. Ακολουθεί η αναφορά στην διεύθυνση 8 (ή 001000₂). Σύμφωνα με την παραπάνω περιγραφόμενη διαδικασία υπάρχει αποτυχία και ακολουθεί η τοποθέτηση της λέξης στο τρίτο μπλοκ (10) στη θέση 00. Το πεδίο ετικέτας τίθεται 00 και το ψηφίο εγκυρότητας 1. Μαζί με τη λέξη αυτή τοποθετούνται και οι γειτονικές λέξεις της κύριας μνήμης με διευθύνσεις 9,10,11.

Κατά την αναφορά στην διεύθυνση 5 (ή 000101₂) η λειτουργία εντοπισμού προκαλεί επιτυχία γιατί εντοπίζεται η ετικέτα 00 και το ψηφίο επικύρωσης του δεύτερου (01) μπλοκ είναι 1. Συνεπώς τα περιεχόμενα της κρυφής μνήμης δεν μεταβάλλονται.

Η κατάσταση, λοιπόν, της κρυφής μνήμης, μετά την αναφορά στις τρεις αυτές διευθύνσεις είναι η εξής (πίνακας 1):

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	00	Mem(0)	Mem(1)	Mem(2)	Mem(3)
01	1	00	Mem(4)	Mem(5)	Mem(6)	Mem(7)
10	1	00	Mem(8)	Mem(9)	Mem(10)	Mem(11)
11	0	00				

Πίνακας 1

Ακολουθεί αναφορά στην διεύθυνση 20 (ή 010100₂) και ο εντοπισμός σημειώνει αποτυχία επειδή στο δεύτερο μπλοκ δεν εντοπίζεται το πεδίο ετικέτας 01. Η λέξη θα πρέπει να τοποθετηθεί στο δεύτερο μπλοκ (01) στην πρώτη θέση (00) (μαζί με τις γειτονικές λέξεις με διευθύνσεις 21,22,23) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος. Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με τη τιμή 01. Το ψηφίο εγκυρότητας παραμένει 1.

Ακολουθεί αναφορά στην διεύθυνση 17 (ή 010001₂) και ο εντοπισμός σημειώνει αποτυχία επειδή στο δεύτερο μπλοκ δεν εντοπίζεται το πεδίο ετικέτας 01. Η λέξη θα πρέπει να τοποθετηθεί μαζί με τις γειτονικές λέξεις της κύριας μνήμης με διευθύνσεις 16,18,19 στο πρώτο μπλοκ (01) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος. Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με την τιμή 01. Το ψηφίο εγκυρότητας παραμένει 1

Κατά την αναφορά στην διεύθυνση 19 (ή 010011₂) η λειτουργία εντοπισμού σημειώνει επιτυχία γιατί στο πρώτο μπλοκ (00) εντοπίζεται η ετικέτα 01 και το ψηφίο επικύρωσης είναι 1. Συνεπώς τα περιεχόμενα της κρυφής μνήμης δεν μεταβάλλονται.

Η κατάσταση, λοιπόν, της κρυφής μνήμης, μετά τις αναφορές 20, 17, 19 εμφανίζεται στον πίνακα 2:

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	01	Mem(16)	Mem(17)	Mem(18)	Mem(19)
01	1	01	Mem(20)	Mem(21)	Mem(22)	Mem(23)
10	1	00	Mem(8)	Mem(9)	Mem(10)	Mem(11)
11	0	00				

Πίνακας 2

Ακολουθεί αναφορά στην διεύθυνση 56 (ή 111000₂) και ο εντοπισμός σημειώνει αποτυχία επειδή στο τρίτο μπλοκ δεν εντοπίζεται το πεδίο ετικέτας 11. Η λέξη θα πρέπει να τοποθετηθεί στο τρίτο μπλοκ (10) στη θέση 00 (μαζί με τις λέξεις με διευθύνσεις 57,58,59) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος. Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με την τιμή 11. Το ψηφίο εγκυρότητας παραμένει 1.

Η κατάσταση της κρυφής μνήμης μετά την αναφορά στη διεύθυνση 56 εμφανίζεται στον πίνακα 3:

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	01	Mem(16)	Mem(17)	Mem(18)	Mem(19)
01	1	01	Mem(20)	Mem(21)	Mem(22)	Mem(23)
10	1	11	Mem(56)	Mem(57)	Mem(58)	Mem(59)
11	0	00				

Πίνακας 3

Κατά την αναφορά στη διεύθυνση 9 (ή 001001₂) η διαδικασία εντοπισμού σημειώνει αποτυχία γιατί δεν εντοπίζει το σωστό πεδίο ετικέτας . Η λέξη θα πρέπει να τοποθετηθεί (μαζί με τις λέξεις με διευθύνσεις 8,10,11) στο τρίτο μπλοκ (10) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος (λειτουργία αντικατάστασης). Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με την τιμή 00. Το ψηφίο εγκυρότητας παραμένει 1.

Ακολουθεί αναφορά στη διεύθυνση 11 (ή 001011₂) και ο εντοπισμός σημειώνει επιτυχία αφού εντοπίζει στο τρίτο μπλοκ την ετικέτα 00 και το πεδίο επικύρωσης του μπλοκ είναι 1.

Κατά την αναφορά στην διεύθυνση 4 (ή 000100₂) η διαδικασία εντοπισμού σημειώνει αποτυχία γιατί στο δεύτερο μπλοκ δεν εντοπίζει το σωστό πεδίο ετικέτας 00. Ακολουθεί η διαδικασία της τοποθέτησης κατά την οποία τοποθετείται η λέξη με διεύθυνση 4 στο μπλοκ δύο (01) θέση 00. Μαζί με αυτή τοποθετούνται και οι λέξεις με διευθύνσεις 5,6,7 και ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με την τιμή 00. Το ψηφίο εγκυρότητας παραμένει 1. Η τοποθέτηση προκαλεί την αντικατάσταση του ήδη υπάρχοντος.

Η κατάσταση της κρυφής μνήμης, μετά τις αναφορές στις διευθύνσεις 9, 11 και 4 εμφανίζεται στον πίνακα 4:

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	01	Mem(16)	Mem(17)	Mem(18)	Mem(19)
01	1	00	Mem(4)	Mem(5)	Mem(6)	Mem(7)
10	1	00	Mem(8)	Mem(9)	Mem(10)	Mem(11)
11	0	00				

Πίνακας 4

Κατά την αναφορά στη διεύθυνση 43 (ή 101011₂) η διαδικασία εντοπισμού σημειώνει αποτυχία γιατί δεν εντοπίζει το σωστό πεδίο ετικέτας 10. Η λέξη θα πρέπει να τοποθετηθεί (μαζί με τις λέξεις με διευθύνσεις 40,41,42) στο τρίτο μπλοκ (10) στη θέση 4 (11) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος. Ταυτόχρονα ενημερώνεται το πεδίο ετικέτας με την τιμή 10. Το ψηφίο εγκυρότητας παραμένει 1.

Ακολουθεί αναφορά στη διεύθυνση 5 (ή 000101₂) με επιτυχία αφού η λέξη εντοπίζεται στο δεύτερο μπλοκ (01) του οποίου το πεδίο ετικέτας είναι 00 και το ψηφίο εγκυρότητας είναι 1. Όμοια η αναφορά στη διεύθυνση 6 (ή 000110₂) είναι επιτυχής.

Η κατάσταση της κρυφής μνήμης μετά τις αναφορές στις διευθύνσεις 43, 5, 6 εμφανίζεται στον πίνακα 5:

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	01	Mem(16)	Mem(17)	Mem(18)	Mem(19)
01	1	00	Mem(4)	Mem(5)	Mem(6)	Mem(7)
10	1	10	Mem(40)	Mem(41)	Mem(42)	Mem(43)
11	0	00				

Πίνακας 5

Ακολουθεί ξανά αναφορά στη λέξη με διεύθυνση 9 (ή 001001₂) και η λειτουργία εντοπισμού σημειώνει αποτυχία γιατί στο τρίτο μπλοκ δεν εντοπίζεται η ετικέτα 00. Η λέξη θα πρέπει να τοποθετηθεί στο τρίτο μπλοκ (10) στη θέση 2 (01) (μαζί με τις λέξεις με διευθύνσεις 8,10,11) και αυτό προκαλεί την αντικατάσταση του ήδη υπάρχοντος. Ταυτόχρονα ενημερώνεται το πεδίο

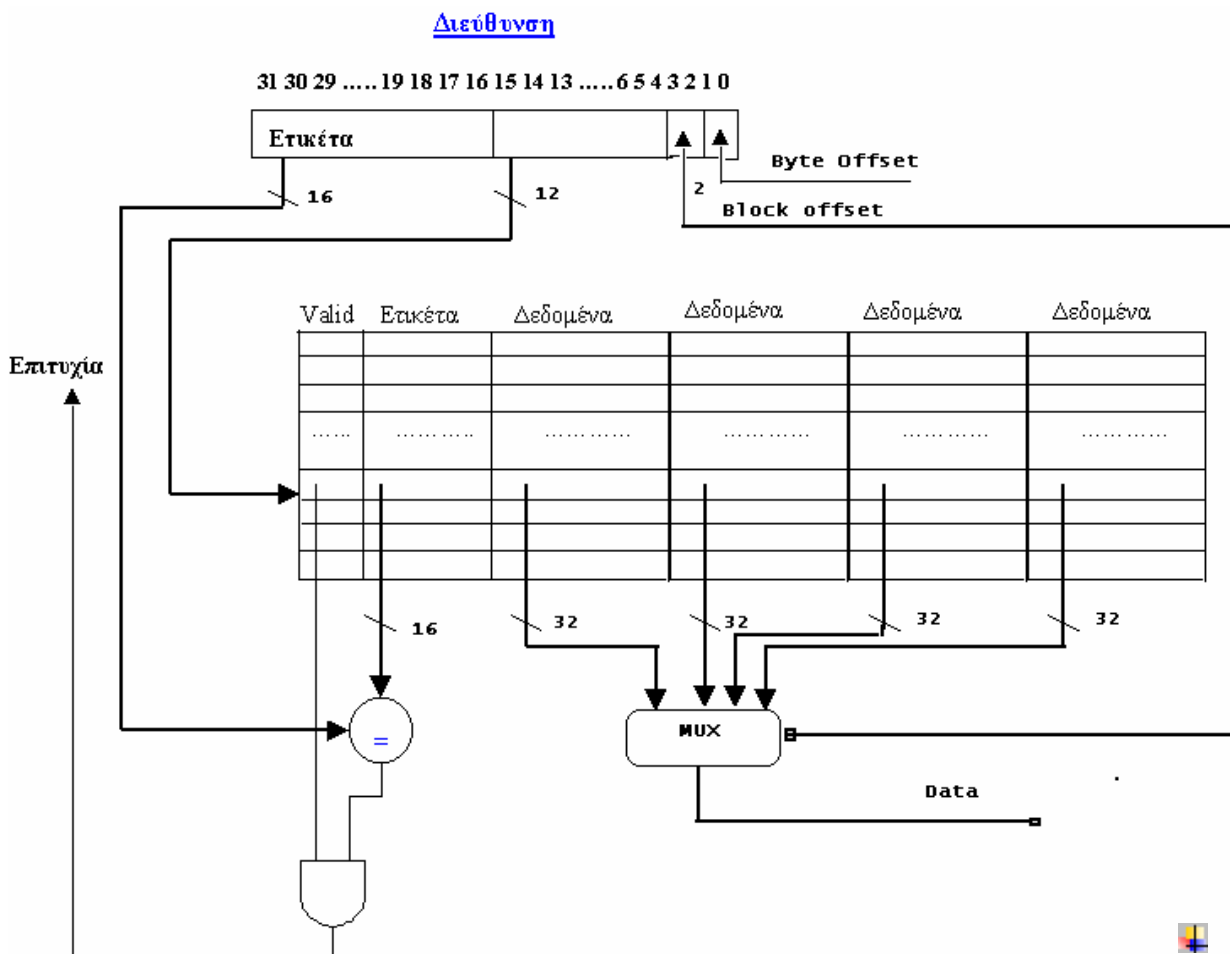
ετικέτας με την τιμή 00. . Το ψηφίο εγκυρότητας παραμένει 1.

Η τελευταία αναφορά γίνεται στη λέξη με διεύθυνση 17) η λειτουργία εντοπισμού σημειώνει επιτυχία γιατί στο πρώτο μπλοκ εντοπίζεται η ετικέτα 01 και το ψηφίο επικύρωσης είναι 1. Τα περιεχόμενα της κρυφής μνήμης δεν μεταβάλλονται. Συνεπώς τελική κατάσταση της κρυφής μνήμης είναι αυτή που παρουσιάζει ο πίνακας 6:

Δείκτης	V	ετικέτα	Δεδ.1	Δεδ.2	Δεδ.3	Δεδ.4
00	1	01	Mem(16)	Mem(17)	Mem(18)	Mem(19)
01	1	00	Mem(4)	Mem(5)	Mem(6)	Mem(7)
10	1	00	Mem(8)	Mem(9)	Mem(10)	Mem(11)
11						

Πίνακας 6 – Τελική κατάσταση κρυφής μνήμης

Το ακόλουθο σχήμα υλοποιεί την πορεία επίλυσης του παραδείγματος για μια κύρια μνήμη με διευθύνσεις των 32 bits και κρυφή μνήμη άμεσης αντιστοίχισης με 2^{12} μπλοκ και μέγεθος μπλοκ 4 λέξεων (16 bytes).





Παράδειγμα 3^ο

Έστω ότι υπάρχουν τρεις μικρές κρυφές μνήμες, η κάθε μια από τις οποίες αποτελείται από τέσσερα μπλοκ μιας λέξης. Η πρώτη κρυφή μνήμη είναι πλήρως συσχετιστική, η δεύτερη είναι 2-δρόμων συνολο-συσχετιστική, ενώ η τρίτη είναι άμεσης αντιστοίχισης. Αν θεωρήσουμε ότι η τεχνική αντικατάστασης που χρησιμοποιείται είναι η LRU, υπολογίστε τον αριθμό των αποτυχιών σε κάθε οργάνωση της κρυφής μνήμης, αν δοθούν οι ακόλουθες διευθύνσεις μπλοκ: 0,8,0,6,8.

Απάντηση

Η ευκολότερη περίπτωση είναι εκείνη της κρυφής μνήμης **άμεσης αντιστοίχισης**. Πρώτα από όλα πρέπει να υπολογίσουμε ποιο μπλοκ της κρυφής μνήμης θα αντιπροσωπεύσει καθένα από τα δοθέντα μπλοκ της κύριας μνήμης. Το μπλοκ της κρυφής μνήμης υπολογίζεται συνήθως από τον τύπο: $(\text{διεύθυνση του μπλοκ}) \bmod (\text{αριθμός των μπλοκ της κρυφής μνήμης})$

Διεύθυνση μπλοκ	Μπλοκ κρυφής
0	$(0 \bmod 4) = 0$
4	$(4 \bmod 4) = 0$
6	$(6 \bmod 4) = 2$
8	$(8 \bmod 4) = 0$

Τώρα μπορούμε να συμπληρώσουμε την κατάσταση της κρυφής μνήμης μετά από κάθε αναφορά στη μνήμη. Αρχικά ζητάμε να επεξεργαστούν τα δεδομένα που βρίσκονται στη διεύθυνση της μνήμης. Επειδή η κρυφή μνήμη είναι κενή έχουμε μια αποτυχία. Έτσι δημιουργούμε τον παρακάτω πίνακα :

Περιεχόμενα της κρυφής μνήμης μετά από αναφορά

Διεύθυνση μπλοκ κύριας	Αποτυχία/ Επιτυχία	0	1	2	3
0	Αποτυχία	Memory [0]			
8	Αποτυχία	Memory [8]			
0	Αποτυχία	Memory [0]			
6	Αποτυχία	Memory [0]		Memory [6]	
8	Αποτυχία	Memory [8]			

Η **κρυφή μνήμη άμεσης αντιστοίχισης** παράγει πέντε αποτυχίες. Οι περισσότερες αποτυχίες συμβαίνουν γιατί θεωρούμε ότι είναι αρχικά κενή. Στην περίπτωση όμως των δεδομένων που βρίσκονται στη διεύθυνση 8 της κύριας μνήμης, η δεύτερη αναφορά οδηγεί και πάλι σε αποτυχία γιατί αυτά έχουν απομακρυνθεί από την κρυφή μνήμη με την εφαρμογή του αλγόριθμου LRU. Στη δεύτερη περίπτωση η **σύνολο-συσχετιστική κρυφή μνήμη** έχει δύο σύνολα, κάθε ένα από τα οποία έχει δύο στοιχεία. Υπολογίζουμε και πάλι το μπλοκ της κρυφής μνήμης, που θα καταλάβει κάθε μπλοκ της κύριας μνήμης στο οποίο αναφερόμαστε.

Διεύθυνση μπλοκ	Μπλοκ κρυφής
0	$(0 \text{ modulo } 2) = 0$
4	$(4 \text{ modulo } 2) = 0$
6	$(6 \text{ modulo } 2) = 0$
8	$(8 \text{ modulo } 2) = 0$

Το σύνολο επιλέγεται με την αντιστοίχιση:

$$(\text{διεύθυνση του μπλοκ}) \bmod (\text{πλήθος συνόλων στη κρυφή μνήμη}).$$

Επειδή πρέπει να αποφασίσουμε ποια εγγραφή μέσα σε ένα σύνολο θα αντικατασταθεί σε μια αποτυχία, χρειαζόμαστε κάποιους κανόνες αντικατάστασης. Οι σύνολο-συσχετιστικές κρυφές μνήμες χρησιμοποιούν συνήθως LRU αλγόριθμο αντικατάστασης. Έτσι δημιουργείται ο παρακάτω πίνακας:

Περιεχόμενα της κρυφής μνήμης μετά από αναφορά

Διεύθυνση μπλοκ κύριας	Αποτυχία/Επιτυχία	Σύνολο 0	Σύνολο 0	Σύνολο 1	Σύνολο 1
0	Αποτυχία	Memory [0]			
8	Αποτυχία	Memory [0]	Memory [8]		
0	Επιτυχία	Memory [0]	Memory [8]		
6	Αποτυχία	Memory [0]	Memory [6]		
8	Αποτυχία	Memory [0]	Memory [8]		

Η **2-δρόμων σύνολο-συσχετιστική κρυφή μνήμη** έχει συνολικά τέσσερις αποτυχίες, μία λιγότερη από την κρυφή μνήμη άμεσης αντιστοίχισης.

Η **πλήρως συσχετιστική κρυφή μνήμη** έχει 4 μπλοκ σε ένα υποτιθέμενο σύνολο και κάθε μπλοκ της μνήμης μπορεί να τοποθετηθεί σε οποιοδήποτε μπλοκ κρυφής μνήμης. Έτσι δημιουργούμε τον παρακάτω πίνακα που δείχνει την κατάσταση της κρυφής μνήμης.

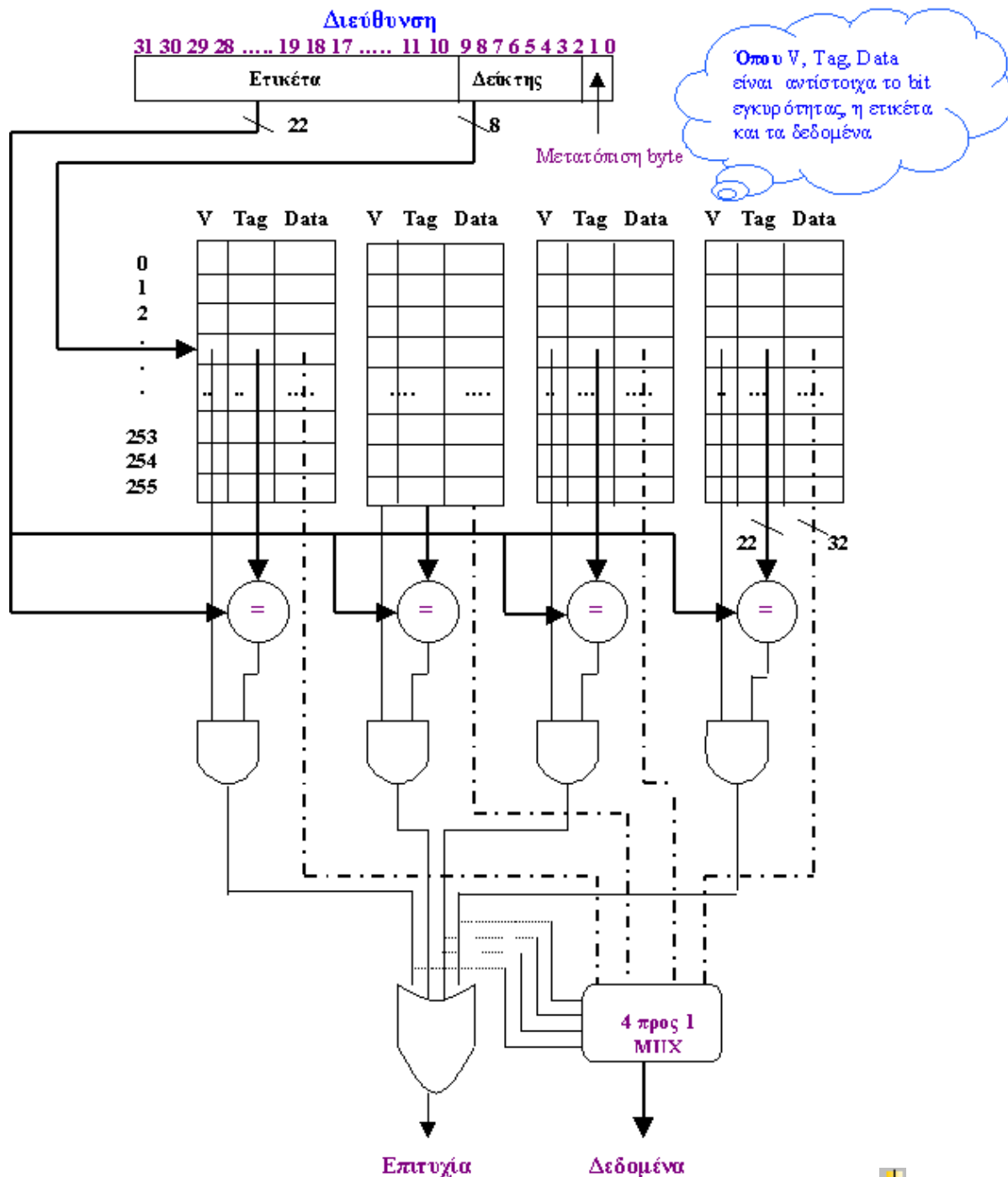
Η πλήρως συσχετιστική κρυφή μνήμη έχει φανερά την καλύτερη απόδοση, με μόνο τρεις αποτυχίες. Παρατηρούμε ότι αν είχαμε οκτώ μπλοκ στην κρυφή μνήμη, τότε δεν θα χρειαζόταν αντικατάσταση στην 2-δρόμων συνολο-συσχετιστική κρυφή μνήμη και θα είχαμε τον ίδιο αριθμό αποτυχιών με την πλήρως συσχετιστική κρυφή μνήμη. Ομοίως, αν είχαμε 16 μπλοκ και οι τρεις κρυφές μνήμες θα είχαν τον ίδιο αριθμό αποτυχιών.

Περιεχόμενα της κρυφής μνήμης μετά από αναφορά

Διεύθυνση Μπλοκ	Αποτυχία/Επιτυχία	1 ^ο μπλοκ	2 ^ο μπλοκ	3 ^ο μπλοκ	4 ^ο μπλοκ
0	Αποτυχία	Memory [0]			
8	Αποτυχία	Memory [0]	Memory [8]		
0	Επιτυχία	Memory [0]	Memory [8]		
6	Αποτυχία	Memory [0]	Memory [8]	Memory [6]	
8	Επιτυχία	Memory [0]	Memory [8]	Memory [6]	

Το ακόλουθο σχήμα είναι μία συνολο-συσχετιστική κρυφή μνήμη 4-δρόμων που αποτελείται από 4 συγκριτές και 1 πολυπλέκτη. Οι συγκριτές καθορίζουν ποιο από τις 4 ετικέτες ταιριάζουν με την

ετικέτα της διεύθυνσης. Η έξοδος των συγκριτών χρησιμοποιείται για την επιλογή των δεδομένων από ενός εκ των 4 δεικτοδοτημένων συνόλων, μέσω ενός πολυπλέκτη.





Η λειτουργία της Ανάγνωσης και της Εγγραφής

Όταν ο επεξεργαστής εκτελέσει μια εντολή που απαιτεί ανάγνωση από ή εγγραφή σε μια διεύθυνση της κύριας μνήμης τότε η αίτηση αυτή απευθύνεται στην κρυφή μνήμη, όπου αρχίζει η λειτουργία του εντοπισμού. Έχει παρατηρηθεί ότι οι αιτήσεις για ανάγνωση είναι πολύ περισσότερες από τις αιτήσεις εγγραφής, επειδή η εκτέλεση όλων των εντολών αρχίζει με την ανάκληση της εντολής, η οποία είναι αίτηση ανάγνωσης. Στη συνέχεια κατά την εκτέλεση της εντολής, μόνον οι εντολές φόρτωσης ή αποθήκευσης περιέχουν αιτήσεις ανάγνωσης ή εγγραφής αντιστοίχα (επεξεργαστής DLX). Η αίτηση ανάγνωσης που είναι ή πιο συχνή περίπτωση είναι και η απλούστερη στην υλοποίηση.

➡ Η ανάγνωση στην κρυφή μνήμη

Κατά τη λειτουργία του εντοπισμού η ανάγνωση των δεδομένων μιας διεύθυνσης, που περιέχεται σε ένα μπλοκ, γίνεται ταυτόχρονα με τη σύγκριση της ετικέτας του και τον έλεγχο του ψηφίου εγκυρότητας. Κατά συνέπεια, η ανάγνωση του μπλοκ ξεκινάει από την στιγμή που είναι διαθέσιμη η διεύθυνσή του, δηλαδή μετά το βήμα της τμηματοποίησης. Αν το αποτέλεσμα του εντοπισμού είναι επιτυχία, τότε τα ζητούμενα δεδομένα τοποθετούνται στην αρτηρία που συνδέει την κρυφή μνήμη με την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ), διαφορετικά αγνοούμε ό,τι διαβάσαμε.

➡ Η εγγραφή στην κρυφή μνήμη

Κατά τη λειτουργία εγγραφής δεδομένων στην κρυφή μνήμη η διαδικασία διαφοροποιείται αισθητά συγκριτικά με εκείνη της ανάγνωσης.

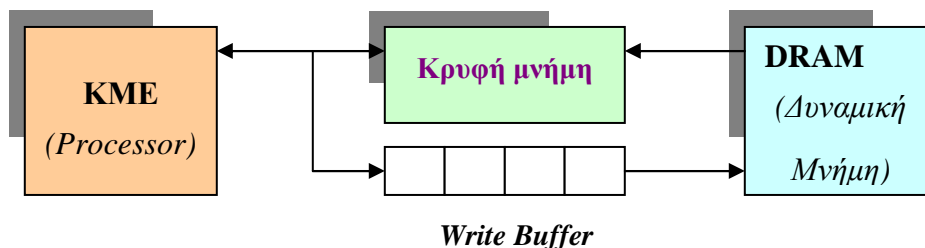
- **Αν το αποτέλεσμα του εντοπισμού είναι επιτυχία**, τότε μόνο μπορεί να τροποποιηθεί το μπλοκ. Ο έλεγχος της ετικέτας ή του ψηφίου εγκυρότητας V , δεν γίνεται παράλληλα, έτσι η λειτουργία της εγγραφής είναι πιο χρονοβόρα. Πιο συγκεκριμένα διακρίνουμε δύο τεχνικές εγγραφής σε περίπτωση επιτυχίας κρυφής μνήμης:
 - i. **Διεγγραφή (write through)** : Η πληροφορία εγγράφεται στο μπλοκ που βρίσκεται στην κρυφή μνήμη καθώς και στο αντίστοιχο μπλοκ της κύριας μνήμης.
 - ii. **Επανεγγραφή (write back)** : Η πληροφορία εγγράφεται μόνο στο μπλοκ της κρυφής μνήμης. Το τροποποιημένο μπλοκ της κρυφής μνήμης εγγράφεται στην κύρια μνήμη μόνο όταν αυτό αντικατασταθεί. Η τροποποίηση του μπλοκ συνοδεύεται με την ενημέρωση ενός ψηφίου κατάστασης, που καλείται dirty bit ($D=1$). Αν η ένδειξη είναι clear bit ($D=0$), δηλώνει ότι το μπλοκ είναι καθαρό, δηλαδή δεν τροποποιήθηκε στην κρυφή μνήμη, οπότε δε χρειάζεται να επανεγγραφεί.
- **Αν το αποτέλεσμα του εντοπισμού είναι αποτυχία** τότε έχουμε δύο εναλλακτικές επιλογές (τεχνικές εγγραφής):
 1. **Ανάθεσηεγγραφής**: το μπλοκ μεταφέρεται από την κύρια μνήμη στην κρυφή (λειτουργία τοποθέτησης ή αντικατάστασης) και στη συνέχεια ακολουθούν οι ενέργειες της επιτυχίας εγγραφής (διεγγραφή ή επανεγγραφή).
 2. **Ανάθεση μη εγγραφής**: το μπλοκ τροποποιείται μόνο στην κύρια μνήμη και όχι στην κρυφή.



Η τεχνική επανεγγραφής συνήθως συνδυάζεται με την ανάθεση εγγραφής ενώ η διεγγραφή με την ανάθεση μη εγγραφής.

➤ Καταχωρητής εγγραφής

Εάν χρησιμοποιούμε την τεχνική της διεγγραφής, τότε η κεντρική μονάδα επεξεργασίας πρέπει να περιμένει την ολοκλήρωση των εγγραφών. Για τη μείωση των καθυστερήσεων των εγγραφών χρησιμοποιείται μια ενδιάμεση μνήμη, ένας καταχωρητής εγγραφής, που επιτρέπει στον επεξεργαστή να συνεχίσει παράλληλα με την ενημέρωση της μνήμης (σχήμα 4.2.4). Η λύση αυτή δεν μηδενίζει ωστόσο τις καθυστερήσεις στις εγγραφές .



Σχήμα 4.2.4 - Η Λειτουργία της διεγγραφής με καταχωρητή εγγραφής

▣ Σύγκριση των μεθόδων εγγραφής στην κρυφή μνήμη

Καθεμία από τις τεχνικές εγγραφής παρουσιάζει τα δικά της πλεονεκτήματα. Πιο συγκεκριμένα:

Επανεγγραφή:

1. Οι αιτήσεις του επεξεργαστή για εγγραφή μεμονωμένων λέξεων ικανοποιούνται με την ταχύτητα της κρυφής μνήμης.
2. Πολλαπλές εγγραφές σ' ένα μπλοκ, απαιτούν μόνο μία εγγραφή στη κύρια μνήμη.
3. Επειδή κάποιες από τις εγγραφές δεν γίνονται άμεσα στην κύρια μνήμη η τεχνική αυτή απαιτεί μικρότερο εύρος ζώνης κύριας μνήμης
4. Όταν το μπλοκ επανεγράφεται, το σύστημα χρησιμοποιεί αποτελεσματικά την κύρια μνήμη δεδομένου ότι εγγράφεται ολόκληρο το μπλοκ.

Διεγγραφή:

1. Μια αποτυχία ανάγνωσης δεν μπορεί ποτέ να οδηγήσει (λόγω πιθανής αντικατάστασης μπλοκ) σε εγγραφή στην κύρια μνήμη.
2. Η τεχνική αυτή υλοποιείται ευκολότερα από την επανεγγραφή (δεν απαιτεί τη χρήση Dirty-bit).
3. Η κύρια μνήμη είναι πάντοτε ενημερωμένη και σύμφωνη με τα περιεχόμενα της κρυφής μνήμης.

Δεν είναι εύκολο να απαντηθεί το ερώτημα ποια από τις δύο τεχνικές εγγραφής είναι η καλύτερη. Η επανεγγραφή απαιτεί μικρότερο εύρος ζώνης κύριας μνήμη και μειώνει την κυκλοφορία των δεδομένων αλλά η άμεση ενημέρωση των περιεχομένων της κύριας μνήμης που προσφέρει η διεγγραφή είναι απαραίτητη σε ορισμένες διεργασίες I/O καθώς και σε περιβάλλον πολυεπεξεργασίας. Η βέλτιστη τεχνική μπορεί να βρεθεί με τεχνικές προσομοίωσης μόνο για

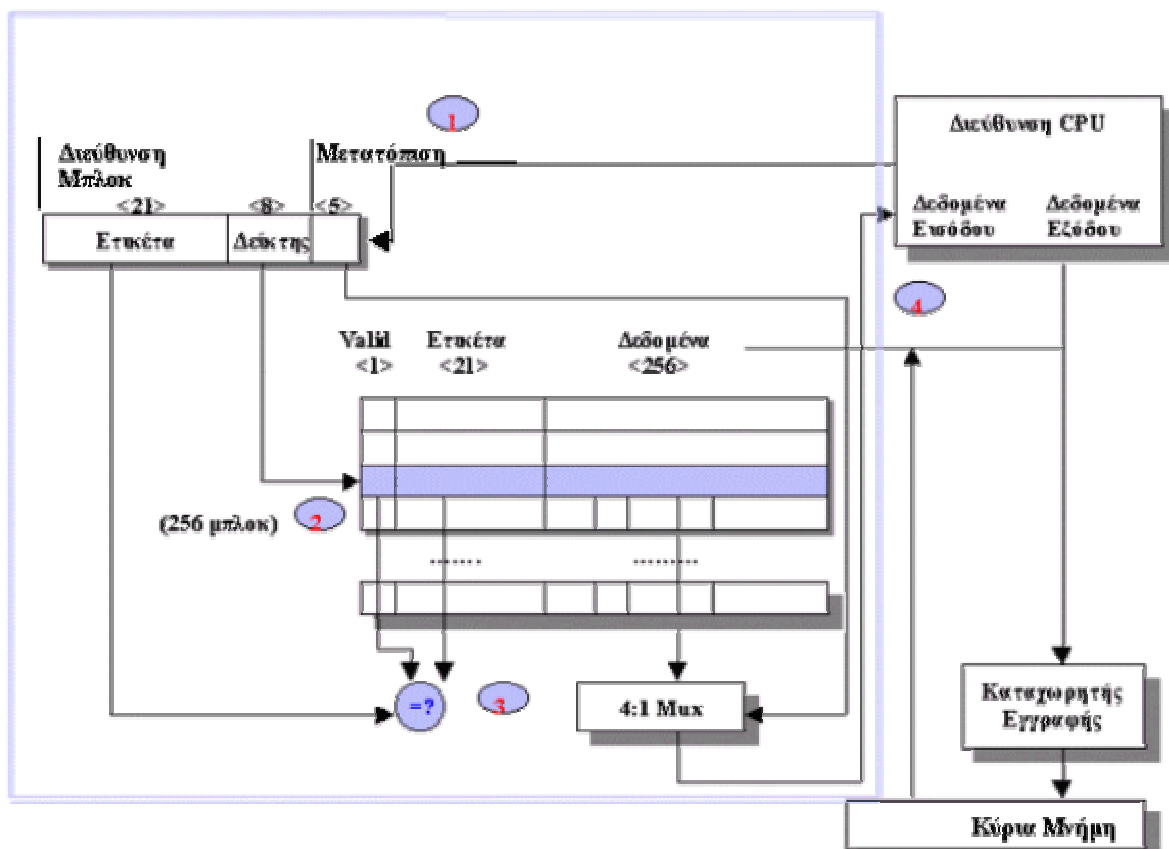
συγκεκριμένη αρχιτεκτονική του υπολογιστή και για συγκεκριμένη χρήση προγραμμάτων.



Παράδειγμα – “Η λειτουργία εγγραφής στον επεξεργαστή Alpha AXP 21064”

Ο επεξεργαστής Alpha AXP 21064 έχει κρυφή μνήμη 8 KB και μέγεθος μπλοκ 32 bytes, και χρησιμοποιεί την τεχνική άμεσης αντιστοίχισης. Σε περίπτωση επιτυχίας χρησιμοποιεί την τεχνική διεγγραφής με καταχωρητή εγγραφής μεγέθους 4 μπλοκ και σε περίπτωση αποτυχίας εγγραφής την τεχνική της ανάθεσης μη εγγραφής. Ο μικροεπεξεργαστής χρησιμοποιεί διευθύνσεις κύριας μνήμης μεγέθους 34-bits.

Ο επεξεργαστής εκτελεί μια εντολή η οποία απαιτεί εγγραφή σε μια διεύθυνση της κύριας μνήμης. Μετά από την λειτουργία του εντοπισμού η οποία εμφανίζεται στα βήματα 1,2 και 3 του σχήματος 4.2.5 υπάρχουν οι εξής δύο περιπτώσεις:



Σχήμα 4.2.5 - Η οργάνωση των δεδομένων της κρυφής μνήμης για τον μικροεπεξεργαστή Alpha AXP 21064.

1. Περίπτωση επιτυχίας της κρυφής μνήμης

Τα δεδομένα γράφονται στο μπλοκ της κρυφής μνήμης. Επιπρόσθετα επειδή χρησιμοποιείται η τεχνική της διεγγραφής τα δεδομένα στέλνονται στον καταχωρητή εγγραφής για να εγγραφούν και

στην κύρια μνήμη.

- Εάν ο καταχωρητής είναι άδειος, τότε τα δεδομένα και ολόκληρη η διεύθυνση αποθηκεύονται σε αυτόν και η εγγραφή σταματά εκεί, όσον αφορά το ρόλο της ΚΜΕ. Η ΚΜΕ συνεχίζει τη λειτουργία της, ενώ ο καταχωρητής εγγραφής ετοιμάζεται να γράψει τη λέξη στη μνήμη.
- Εάν ο καταχωρητής δεν είναι άδειος και περιέχει ένα ή περισσότερα τροποποιημένα μπλοκ, συγκρίνεται η διεύθυνση του νέου μπλοκ με τη διεύθυνση των μπλοκ που υπάρχουν στον καταχωρητή. Εάν ταιριάζει με κάποια από τις διευθύνσεις αυτές, τότε ακολουθεί μία συνένωση των δεδομένων των δύο μπλοκ που ονομάζεται συγχώνευση εγγραφής. Η διαδικασία αυτή γίνεται για λόγους βελτιστοποίησης, αφού διαφορετικά 4 αποθηκεύσεις διαδοχικών διευθύνσεων θα γέμιζαν τον καταχωρητή εγγραφής, ακόμα κι αν οι 4 διευθύνσεις αναφέρονται στο ίδιο μπλοκ της κύριας μνήμης. Στην περίπτωση που ο καταχωρητής εγγραφής είναι πλήρης και καμία διεύθυνση από τα μπλοκ που περιέχει δεν ταιριάζει με τη διεύθυνση του νέου μπλοκ τότε, η κρυφή μνήμη (και η ΚΜΕ) είναι αναγκασμένη να περιμένει να ελευθερωθεί μία θέση στον καταχωρητή εγγραφής.

2. Περίπτωση αποτυχίας της κρυφής μνήμης

Επειδή χρησιμοποιείται η τεχνική ανάθεσης μη εγγραφής η ΚΜΕ παρακάμπτει την κρυφή μνήμη και γράφει κατευθείαν στην κύρια μνήμη χωρίς να επηρεάζει τη κρυφή μνήμη.

Ανακεφαλαιώνοντας λοιπόν



Η λειτουργία της κρυφής μνήμης στηρίζεται στην αντιγραφή μέρους από τα περιεχόμενα της κύριας μνήμης. Αξιοποιώντας την τοπικότητα της αναφοράς των προγραμμάτων, οι περισσότερες αναφορές στη μνήμη γίνονται μέσα από ένα περιορισμένο σύνολο διεθύνσεων με πολύ μικρή καθυστέρηση.



Οι βασικές λειτουργίες αφορούν την αναζήτηση, την τοποθέτηση, τον εντοπισμό, την ανάγνωση, την εγγραφή και την διεγγραφή των μπλοκ στην κρυφή μνήμη. Σημαντικές έννοιες είναι η επιτυχία και η αποτυχία της κρυφής μνήμης.



Η θέση που θα τοποθετηθεί ένα μπλοκ στην κρυφή μνήμη εξαρτάται από τον αλγόριθμο αντιστοίχισης της κρυφής μνήμης (άμεση αντιστοίχιση, πλήρως συσχετιστική και συνολο-συσχετιστική αντιστοίχιση). Η διεύθυνση της κρυφής μνήμης τμηματοποιείται στη διεύθυνση του μπλοκ και στη μετατόπιση του μπλοκ. Επιπλέον η διεύθυνση είναι δυνατόν να διαιρεθεί στο πεδίο ετικέτας και στο πεδίο του δείκτη. Το πεδίο ετικέτας κάθε μπλοκ της κρυφής μνήμης ελέγχεται για να διαπιστωθεί εάν ταιριάζει με τη διεύθυνση του μπλοκ απ'την ΚΜΕ.



Για να εντοπίσουμε το ζητούμενο μπλοκ, συγκρίνουμε την ετικέτα διεύθυνσης με την διεύθυνση του μπλοκ απ'την ΚΜΕ. Κατά κανόνα, όλες οι πιθανές ετικέτες εξετάζονται παράλληλα για ταχύτερα αποτελέσματα. Ένα μπλοκ μπορεί να βρίσκεται στην κρυφή μνήμη (επιτυχία) ή όχι (αποτυχία).



Για να επιλέξουμε το μπλοκ που θα αντικατασταθεί, χρησιμοποιούμε τις μεθόδους της “Τυχαίας αντικατάστασης”, της “Λιγότερο πρόσφατα χρησιμοποιημένο” και της “Πρώτο μέσα πρώτο έξω”.



Κατά τη λειτουργία του εντοπισμού η ανάγνωση των δεδομένων μιας διεύθυνσης, που περιέχεται σε ένα μπλοκ, γίνεται ταυτόχρονα με τη σύγκριση της ετικέτας του και τον έλεγχο του ψηφίου εγκυρότητας. Κατά συνέπεια, η ανάγνωση του μπλοκ ξεκινάει από την στιγμή που είναι διαθέσιμη η διεύθυνσή του, δηλαδή μετά το βήμα της τμηματοποίησης. Αν το αποτέλεσμα του εντοπισμού είναι επιτυχία, τότε τα ζητούμενα δεδομένα τοποθετούνται στην αρτηρία που συνδέει την κρυφή μνήμη με την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ), διαφορετικά αγνοούμε ό,τι



διαβάσαμε. Κατά τη λειτουργία εγγραφής δεδομένων στην κρυφή μνήμη η διαδικασία διαφοροποιείται αισθητά συγκριτικά με εκείνη της ανάγνωσης. Αν το αποτέλεσμα του εντοπισμού είναι επιτυχία, τότε μόνο μπορεί να τροποποιηθεί το μπλοκ. Ο έλεγχος της ετικέτας ή του ψηφίου εγκυρότητας V , δεν γίνεται παράλληλα, έτσι η λειτουργία της εγγραφής είναι πιο χρονοβόρα. Σε αυτή την περίπτωση, διακρίνουμε δύο τεχνικές εγγραφής: την *Διεγγραφή* (*write thought*) και την *Επανεγγραφή* (*write back*). Αν το αποτέλεσμα του εντοπισμού είναι αποτυχία τότε έχουμε δύο εναλλακτικές επιλογές (τεχνικές εγγραφής): την *Ανάθεση εγγραφής* και την *Ανάθεση μη εγγραφής*.



Ενότητα 2

Ερωτήσεις Επισκόπησης

Ερώτηση 1 Τι είναι η κρυφή μνήμη και από τι κατασκευάζεται;

Ερώτηση 2 Τι σημαίνουν οι εκφράσεις τοπικότητα του χρόνου και τοπικότητα του χώρου;

Ερώτηση 3 Πότε έχουμε επιτυχία / αποτυχία σε μια κρυφή μνήμη;

Ερώτηση 4 Ποιους αλγορίθμους γνωρίζετε για την αντιστοίχιση ενός μπλοκ της κύριας μνήμης στην κρυφή;

Ερώτηση 5 Σε ποια τμήματα διαιρείται μία διεύθυνση κύριας μνήμης κατά τη διαδικασία τμηματοποίησης της από την CPU και σε τι χρησιμεύει το καθένα;

Ερώτηση 6 Τι γνωρίζετε για τη λειτουργία του εντοπισμού στην κρυφή μνήμη;

Ερώτηση 7 Ποιες μεθόδους χρησιμοποιούμε για την αντικατάσταση μπλοκ στην κρυφή μνήμη και ποια από αυτές υλοποιείται πιο εύκολα;

Ερώτηση 8 Εάν σε μία εντολή εγγραφής υπάρχει επιτυχία της κρυφής μνήμης, ποιες τεχνικές εγγραφής χρησιμοποιούνται σε αυτή;

Ερώτηση 9 Μελετώντας τα δεδομένα του πίνακα με τους ρυθμούς αποτυχίας για κρυφές μνήμες με εντολές, δεδομένα και μεικτές κρυφές μνήμες διαφορετικών μεγεθών (διαθέτουν κρυφή μνήμη για δεδομένα και για εντολές), τι συμπεραίνετε; Ποια είναι πιο σημαντική από πλευράς ρυθμού αποτυχίας; Τα δεδομένα του πίνακα αναφέρονται σε κρυφή μνήμη άμεσης αντιστοίχισης με μέγεθος μπλοκ 32 bytes και ποσοστό αναφορών σε εντολές 75%.

Μέγεθος	Κρυφή μνήμη Εντολών	Κρυφή μνήμη Δεδομένων	Μεικτή Κρυφή μνήμη
1 KB	3,06%	24,61%	13,34%
2KB	2,26%	20,57%	9,78%
4KB	1,78%	15,94%	7,24%
8KB	1,10%	10,19%	4,57%
16 KB	0,64%	6,47%	2,87%
32 KB	0,39%	4,82%	1,99%
64 KB	0,15%	3,77%	1,35%
128 KB	0,02%	2,88%	0,95%



Ενότητα 2

Απαντήσεις Ερωτήσεων Επισκόπησης

1. Η κρυφή μνήμη είναι μία μικρή σε μέγεθος μνήμη, πολύ υψηλής ταχύτητας που τοποθετείται ανάμεσα στα άλλα επίπεδα μνήμης με σκοπό να βελτιώσει τη ταχύτητα μεταφοράς των δεδομένων. Κατασκευάζεται από προηγμένα κυκλώματα, για να εξασφαλίζει μικρούς χρόνους πρόσβασης. Το τίμημα φυσικά είναι το υψηλό της κόστος. Η λειτουργία της κρυφής μνήμης στηρίζεται στην αντιγραφή μέρους από τα περιεχόμενα της κύριας μνήμης. Αξιοποιώντας την τοπικότητα της αναφοράς των προγραμμάτων, οι περισσότερες αναφορές στη μνήμη γίνονται μέσα από ένα περιορισμένο σύνολο διευθύνσεων με πολύ μικρή καθυστέρηση.

2. **Τοπικότητα χρόνου:** Αν ένα πρόγραμμα έχει αναφερθεί σε μια διεύθυνση μνήμης, τότε υπάρχει μεγάλη πιθανότητα να αναφερθεί σε σύντομο χρονικό διάστημα στην ίδια διεύθυνση. Αυτό συμβαίνει στην περίπτωση ενός βρόχου. **Τοπικότητα χώρου:** Αν ένα πρόγραμμα έχει αναφερθεί σε μια διεύθυνση μνήμης, τότε υπάρχει μεγάλη πιθανότητα να αναφερθεί σύντομα σε γειτονικές διευθύνσεις. Αυτό συμβαίνει λόγω της αποθήκευσης των εντολών και των δεδομένων ενός προγράμματος σε διαδοχικές θέσεις μνήμης.

3. Όταν ο επεξεργαστής εκτελέσει μια εντολή που απαιτεί ανάγνωση από μια διεύθυνση της κύριας

μνήμης τότε η αίτηση αυτή απευθύνεται στην κρυφή μνήμη με σκοπό να διαπιστωθεί αν το περιεχόμενο της ζητούμενης διεύθυνσης έχει ήδη μεταφερθεί και υπάρχει σε κάποιο μπλοκ της κρυφής μνήμης. Αν το μπλοκ βρεθεί στην κρυφή μνήμη τότε αυτό ονομάζεται "επιτυχία της κρυφής μνήμης" (cache hit). Αν το μπλοκ δεν βρεθεί στην κρυφή μνήμη τότε αυτό ονομάζεται "αποτυχία της κρυφής μνήμης" (cache miss).

4. Υπάρχουν τρεις διαφορετικοί τρόποι αντιστοίχισης ενός μπλοκ της κύριας μνήμης στην κρυφή μνήμη, οι οποίοι καθορίζουν την οργάνωση της. Ειδικότερα:

- **Άμεση αντιστοίχιση:** Ένα μπλοκ της κύριας μνήμης μπορεί να αντιστοιχηθεί μόνο σε ένα μπλοκ της κρυφής μνήμης. Η αντιστοίχιση γίνεται με τον τύπο: (διεύθυνση του μπλοκ) MOD (αριθμός των μπλοκ στην κρυφή μνήμη).
- **Πλήρως Συσχετιστική:** Ένα μπλοκ της κύριας μνήμης μπορεί να αντιστοιχηθεί σε οπουδήποτε μπλοκ της κρυφής μνήμης.
- **Συνολο-συσχετιστική:** Σε αυτό τον τύπο κρυφής μνήμης τα μπλοκ της κρυφής μνήμης είναι οργανωμένα σε σύνολα που το κάθε ένα από αυτά περιέχει δύο ή περισσότερα μπλοκ. Ένα μπλοκ της κύριας μνήμης αντιστοιχίζεται σε ένα συγκεκριμένο σύνολο της κρυφής μνήμης αλλά στη συνέχεια μπορεί να τοποθετηθεί σε οποιοδήποτε από τα μπλοκ που περιέχει το σύνολο αυτό. Η διεύθυνση του συγκεκριμένου συνόλου της κρυφής μνήμης υπολογίζεται (όπως και στην άμεση αντιστοίχιση) από την διεύθυνση του μπλοκ της κύριας μνήμης σύμφωνα με τον παρακάτω τύπο: Διεύθυνση συνόλου κρυφής μνήμης = (διεύθυνση μπλοκ κύριας μνήμης) MOD (αριθμός συνόλων κρυφής μνήμης)

5. Η ζητούμενη διεύθυνση της κύριας μνήμης χωρίζεται σε τμήματα αρχίζοντας από το λιγότερο σημαντικό bit της διεύθυνσης. Αρχικά χωρίζεται το τμήμα της **μετατόπισης** το οποίο χρησιμοποιείται μόνο σε περίπτωση επιτυχίας για την επιλογή του περιεχομένου της αντίστοιχης διεύθυνσης μέσα από το μπλοκ. Ο αριθμός των bit που αποτελούν αυτό το πεδίο εξαρτάται από το μέγεθος του μπλοκ. Το υπόλοιπο τμήμα της διεύθυνσης ανάλογα με το είδος της κρυφής μνήμης μπορεί να διαιρεθεί σε δύο τμήματα το τμήμα **ετικέτας** και το τμήμα του **δείκτη**. Στη συνέχεια χωρίζεται το τμήμα **δείκτη** το οποίο χρησιμοποιείται για την επιλογή του συνόλου (ή του μπλοκ) της κρυφής μνήμης που περιέχει τα μπλοκ που πρέπει να ελεγχθούν. Το μέγεθος του πεδίου αυτού εξαρτάται από τον αριθμό των συνόλων/μπλοκ της κρυφής μνήμης. Τα υπόλοιπα πλέον σημαντικά ψηφία της διεύθυνσης αποτελούν το τμήμα ετικέτας.

6. Η λειτουργία του εντοπισμού, όπως έχει αναφερθεί, αρχίζει όταν ο επεξεργαστής εκτελεί εντολή που απαιτεί πρόσβαση στο περιεχόμενο μιας διεύθυνσης της κύριας μνήμης και η αίτηση αυτή μεταφέρεται στην κρυφή μνήμη. Η λειτουργία του εντοπισμού σκοπό έχει να διαπιστώσει αν το μπλοκ της κύριας μνήμης που περιέχει τη ζητούμενη διεύθυνση, έχει ήδη μεταφερθεί σε ένα αντίστοιχο μπλοκ της κρυφής και συνεπώς το περιεχόμενο της ζητούμενης διεύθυνσης υπάρχει ήδη μέσα στην κρυφή μνήμη. Τα βήματα του εντοπισμού είναι: Η τμηματοποίηση, η επιλογή του συνόλου ή του μπλοκ και η σύγκριση.

7. Χρησιμοποιούμε την μέθοδο της τυχαίας αντικατάστασης, όπου τα υποψήφια μπλοκ επιλέγονται τυχαία για να έχουμε όσο το δυνατό ομοιόμορφη κατανομή, τη μέθοδο "Λιγότερο πρόσφατα χρησιμοποιημένο", όπου το καλύτερο υποψήφιο μπλοκ για απομάκρυνση είναι το λιγότερο πρόσφατα χρησιμοποιημένο και τη μέθοδο FIFO - "Πρώτο μέσα Πρώτο έξω", όπου τα μπλοκ αντικαθίστανται διαδοχικά με τη σειρά, σύμφωνα με την φυσική τους θέση. Η μέθοδος της τυχαίας

αντικατάστασης είναι πιο απλή και έχει λιγότερες απαιτήσεις από το υλικό. Καθώς ο αριθμός των μπλοκ που καταγράφονται αυξάνει, η δεύτερη και η τρίτη στρατηγική γίνονται υπερβολικά δαπανηρές.

8. Πιο συγκεκριμένα διακρίνουμε δύο τεχνικές εγγραφής σε περίπτωση επιτυχίας κρυφής μνήμης:

- **Διεγγραφή (write through)** : Η πληροφορία εγγράφεται στο μπλοκ που βρίσκεται στην κρυφή μνήμη καθώς και στο αντίστοιχο μπλοκ της κύριας μνήμης.
- **Επανεγγραφή (write back)** : Η πληροφορία εγγράφεται μόνο στο μπλοκ της κρυφής μνήμης. Το τροποποιημένο μπλοκ της κρυφής μνήμης εγγράφεται στην κύρια μνήμη μόνο όταν αυτό αντικατασταθεί. Η τροποποίηση του μπλοκ συνοδεύεται με την ενεμύρωση ενός ψηφίου κατάστασης, που καλείται dirty bit (D=1). Αν η ένδειξη είναι clear bit (D=0), δηλώνει ότι το μπλοκ είναι καθαρό, δηλαδή δεν τροποποιήθηκε στην κρυφή μνήμη, οπότε δε χρειάζεται να επανεγγραφεί.

9. Μία πρώτη παρατήρηση είναι ότι η κρυφή μνήμη εντολών έχει μικρότερους ρυθμούς αποτυχίας συγκριτικά με την κρυφή μνήμη δεδομένων. Ο διαχωρισμός εντολών και δεδομένων αφαιρεί τις αποτυχίες εξαιτίας των συγκρούσεων μεταξύ των μπλοκ με τα δεδομένα και τις εντολές, αλλά και σταθεροποιεί το μέγεθος που αντιστοιχεί στην καθεμιά από την κρυφή μνήμη. Με μία πιο προσεκτική ματιά, διαπιστώνουμε ότι μία μεικτή κρυφή μνήμη απαιτεί το συνολικό μέγεθος, μέγεθος κρυφής μνήμης δεδομένων και μέγεθος κρυφής μνήμης εντολών, για να έχει το ίδιο αποτέλεσμα. Για παράδειγμα, μία ξεχωριστή κρυφή μνήμη 1KB και μία 1KB κρυφή μνήμη εντολών, απαιτούν μία 2KB μεικτή κρυφή μνήμη για να είναι η σύγκριση σωστή. Αν υπολογίσουμε και τον μέσο ρυθμό αποτυχίας για κάθε τύπο κρυφής μνήμης σε συνδυασμό με το ποσοστό αναφορών στη μνήμη, τα συμπεράσματά μας αναδεικνύουν και πάλι τις αποκλίσεις. Το πλέον σίγουρο είναι ότι επηρεάζουν την απόδοση του συστήματος.

Συγκρίνουμε τη μεικτή μνήμη 2KB με τη κρυφή μνήμη δεδομένων και εντολών μεγέθους 1KB

Μέγεθος	Κρυφή μνήμη Εντολών	Κρυφή μνήμη Δεδομένων	Μεικτή Κρυφή μνήμη
1 KB	3,06%	24,61%	13,34%
2KB	2,26%	20,57%	9,78%
4KB	1,78%	15,94%	7,24%
8KB	1,10%	10,19%	4,57%
16 KB	0,64%	6,47%	2,87%
32 KB	0,39%	4,82%	1,99%
64 KB	0,15%	3,77%	1,35%
128 KB	0,02%	2,88%	0,95%