

Ενότητα 9^η: Απευθείας πρόσβαση στη μνήμη.

Σκοπός Ο σκοπός της ενότητας αυτής είναι να περιγράψει την απευθείας πρόσβαση στη μνήμη, τη λειτουργία της και τις παραμέτρους που τη διέπουν.

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



περιγράφετε τη λειτουργία της απευθείας πρόσβασης στη μνήμη (DMA).



απαριθμείτε τα στάδια σε μια μεταφορά DMA.



παρουσιάζετε την DMA και τα συστήματα ιδεατής και κρυφής μνήμης.



απευθείας πρόσβαση στη μνήμη (DMA)



Η λειτουργία της απευθείας πρόσβασης μνήμης

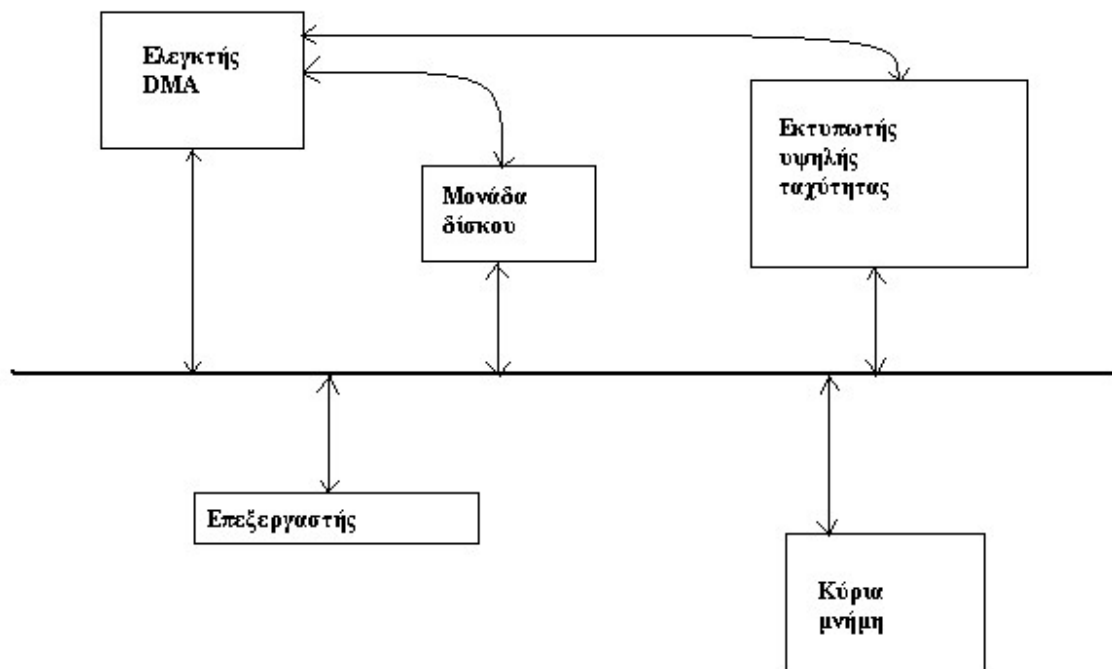
Το I/O οδηγούμενο από διακοπές απαλλάσσει τον επεξεργαστή από την αναμονή για κάθε λειτουργία I/O. Παρόλ'αυτά, αν χρησιμοποιήσουμε αυτή τη μέθοδο για τη μεταφορά δεδομένων από ή προς ένα σκληρό δίσκο, **η επιβάρυνση παραμένει τεράστια**, αφού, στο χρονικό διάστημα που ο δίσκος εκτελεί μεταφορές, καταναλώνεται το 100% της επεξεργαστικής ισχύος. Για συσκευές υψηλού εύρους ζώνης, όπως οι σκληροί δίσκοι, οι μεταφορές συνίστανται κυρίως από σχετικά μεγάλα μπλοκ δεδομένων (εκατοντάδες ως χιλιάδες bytes). Για παράδειγμα, για τη μεταφορά ενός μπλοκ 2048 λέξεων θα απαιτούταν τουλάχιστον 2048 αποθηκεύσεις και 2048 φορτώσεις, ενώ θα είχαμε και την επιπρόσθετη επιβάρυνση λόγω των διακοπών.

Για να **απαλλαγεί ο επεξεργαστής** όσο το δυνατόν περισσότερο, οι σχεδιαστές υπολογιστών ανακάλυψαν ένα μηχανισμό όπου για τη μεταφορά των δεδομένων απευθείας προς ή από τη μνήμη υπεύθυνος είναι ο ελεγκτής της συσκευής και δεν εμπλέκεται ο επεξεργαστής. Αυτός ο μηχανισμός ονομάζεται **άμεση προσπέλαση μνήμης** (direct memory access – **DMA**). Ο μηχανισμός διακοπών χρησιμοποιείται και τώρα από τη συσκευή για να επικοινωνεί με τον επεξεργαστή, αλλά μόνο όταν η μεταφορά I/O ολοκληρώνεται ή όταν προκύπτει κάποιο λάθος.



Η DMA λοιπόν υλοποιείται με έναν εξειδικευμένο επεξεργαστή – ελεγκτή- ο οποίος μεταφέρει δεδομένα μεταξύ της μνήμης και μιας συσκευής εισόδου/εξόδου, ενώ την ίδια στιγμή ο επεξεργαστής ασχολείται με άλλες διεργασίες. Γι'αυτό το λόγο βρίσκεται εκτός του επεξεργαστή και είναι κύριος (master) στην αρτηρία, κατευθύνοντας τις αναγνώσεις και τις εγγραφές ανάμεσα στη συσκευή και τη μνήμη.

Στο παρακάτω σχήμα βλέπουμε ένα υπολογιστικό σύστημα με δυο συσκευές εισόδου/εξόδου που χρησιμοποιούν ένα ελεγκτή DMA για να επικοινωνήσουν με τη μνήμη. Η οργάνωση αρτηρίας είναι απλοποιημένη.



Σχήμα 5.9.1 - Ένας DMA ελεγκτής δύο καναλιών, δηλαδή ένας ελεγκτής που χρησιμοποιείται ταυτόχρονα για δύο συσκευές I/O , στο συγκεκριμένο παράδειγμα για μια μονάδα δίσκου και ένα εκτυπωτή υψηλής ταχύτητας.

▣ Στάδια σε μία μεταφορά DMA

Υπάρχουν τρία στάδια σε μια μεταφορά DMA:

1. Ο επεξεργαστής **προετοιμάζει** τη DMA παρέχοντας την ταυτότητα της συσκευής, τη λειτουργία που θα επιτελέσει στη συσκευή, τη διεύθυνση μνήμης που αποτελεί την πηγή ή τον προορισμό των δεδομένων που πρόκειται να μεταφερθούν και το πλήθος των bytes.
2. Η DMA **ξεκινά τη λειτουργία στη συσκευή** και εμπλέκεται σε ένα σχήμα διαιτησίας για να γίνει κάτοχος της αρτηρίας. Όταν τα δεδομένα είναι διαθέσιμα (από τη συσκευή ή τη μνήμη) τα μεταφέρει. Η συσκευή DMA παρέχει τη διεύθυνση μνήμης για την ανάγνωση ή την εγγραφή. Αν η αίτηση απαιτεί περισσότερες από μία μεταφορές αρτηρίας, η μονάδα DMA παράγει την επόμενη διεύθυνση μνήμης και ξεκινά την επόμενη μεταφορά. **Χρησιμοποιώντας αυτό το μηχανισμό, μία μονάδα DMA μπορεί να μεταφέρει έναν ολόκληρο τομέα δίσκου, που μπορεί να έχει μέγεθος χιλιάδες bytes, χωρίς να ενοχλήσει τον επεξεργαστή. Πολλοί ελεγκτές DMA περιλαμβάνουν κάποια ενδιάμεση αποθήκευση (buffering) για να μπορούν να αντιμετωπίσουν με ευελιξία τις όποιες καθυστερήσεις είτε κατά τη μεταφορά, είτε εκείνες που προκύπτουν όσο ο ελεγκτής περιμένει να γίνει κύριος της αρτηρίας.**

3. Από τη στιγμή που η μεταφορά DMA έχει ολοκληρωθεί, ο ελεγκτής στέλνει ένα σήμα διακοπής στον επεξεργαστή, ο οποίος είναι σε θέση τότε να προσδιορίσει αν η όλη λειτουργία ολοκληρώθηκε με επιτυχία εξετάζοντας τη συσκευή DMA ή τη μνήμη.

Πολλαπλές συσκευές DMA

Μπορεί να υπάρχουν πολλαπλές συσκευές DMA σε ένα υπολογιστικό σύστημα. Για παράδειγμα, σε ένα σύστημα με μία μόνο αρτηρία επεξεργαστή-μνήμης και πολλές αρτηρίες I/O, κάθε ελεγκτής αρτηρίας I/O πολλές φορές περιέχει έναν επεξεργαστή DMA που διαχειρίζεται κάθε μεταφορά ανάμεσα σε μία συσκευή που βρίσκεται στην αρτηρία I/O και τη μνήμη.

Ας δούμε τι ποσοστό του επεξεργαστή καταναλώνεται χρησιμοποιώντας DMA για να αντιμετωπίσουμε το παράδειγμά μας με το σκληρό δίσκο.



Παράδειγμα

Υποθέστε ότι έχουμε τον ίδιο επεξεργαστή και τον ίδιο σκληρό δίσκο όπως στο παράδειγμα της ενότητας 7. Υποθέστε πως η αρχική ρύθμιση (set-up) μίας μεταφοράς DMA απαιτεί 1000 κύκλους ρολογιού για τον επεξεργαστή και υποθέστε πως η διαχείριση της διακοπής κατά την ολοκλήρωση της DMA απαιτεί 500 κύκλους. Ο σκληρός δίσκος έχει ένα ρυθμό μεταφοράς δεδομένων των 2 MB/sec και χρησιμοποιεί DMA. Αν η μέση μεταφορά από το δίσκο είναι 4 KB, τι ποσοστό του επεξεργαστή των 50 MHz καταναλώνεται αν ο δίσκος εκτελεί μεταφορές για το 100% του χρόνου; Αγνοήστε οποιαδήποτε επίδραση από το συναγωνισμό αρτηρίας ανάμεσα στον επεξεργαστή και τον ελεγκτή DMA.

Απάντηση:

Κάθε μεταφορά DMA απαιτεί

$$\frac{4 \text{ KB}}{2 \frac{\text{MB}}{\text{second}}} = 2 \times 10^{-3} \text{ seconds}$$

Έτσι αν ο δίσκος εκτελεί συνεχώς μεταφορές, απαιτούνται:

$$\frac{1000 + 500 \frac{\text{κύκλοι}}{\text{μεταφορά}}}{2 \times 10^{-3} \frac{\text{seconds}}{\text{μεταφορά}}} = 750 \times 10^{-3} \frac{\text{κύκλοι ρολογιού}}{\text{second}}$$

Αφού ο επεξεργαστής λειτουργεί στα 50 MHz:

$$\text{Ποσοστό του επεξεργαστή που καταναλώθηκε} = \frac{750 \times 10^3}{50 \times 10^6} = 15 \times 10^{-3} = 1.5\%$$



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Τι συμπεράσματα προκύπτουν από το προηγούμενο παράδειγμα όσον αφορά τη χρήση DMA για τη διασύνδεση του σκληρού δίσκου; Να συγκρίνετε την απάντησή σας με την παράγραφο που ακολουθεί.



Αντίθετα με το I/O εξέτασης ή το I/O οδηγούμενο από διακοπές, η DMA μπορεί να χρησιμοποιηθεί για τη διασύνδεση ενός σκληρού δίσκου χωρίς την κατανάλωση όλων των κύκλων του επεξεργαστή για μια απλή λειτουργία εισόδου/εξόδου. Επιπλέον, λογικά ο δίσκος δε θα μεταφέρει δεδομένα την περισσότερη ώρα και η τιμή που υπολογίστηκε στο παραπάνω παράδειγμα θα είναι σημαντικά χαμηλότερη. Φυσικά, αν ταυτόχρονα και ο επεξεργαστής συναγωνίζεται για να χρησιμοποιήσει τη μνήμη, η εξυπηρέτηση του θα καθυστερεί όταν η μνήμη είναι απασχολημένη εκτελώντας μία μεταφορά DMA, και υπάρχει και η πιθανότητα να συμβεί σύγκρουση (collision). Χρησιμοποιώντας κρυφές μνήμες (caches), ο επεξεργαστής μπορεί να αποφύγει την ανάγκη πρόσβασης στη μνήμη τον περισσότερο καιρό, αφήνοντας έτσι το περισσότερο εύρος ζώνης της μνήμης διαθέσιμο για χρήση από τις συσκευές I/O.

Επεξεργαστές I/O

Για να ελαττώσουμε περαιτέρω την ανάγκη να διακόπτουμε τον επεξεργαστή και να τον απασχολούμε με τη διαχείριση μίας αίτησης I/O που μπορεί στην πραγματικότητα να συνεπάγεται την εκτέλεση αρκετών λειτουργιών, μπορούμε να κάνουμε τον ελεγκτή I/O να λειτουργεί πιο έξυπνα. Οι έξυπνοι ελεγκτές ονομάζονται συχνά **επεξεργαστές I/O** (I/O processors, όπως επίσης και ελεγκτές I/O – I/O controllers ή ελεγκτές καναλιών – channel controllers).

Αυτοί οι εξειδικευμένοι επεξεργαστές εκτελούν βασικά μία σειρά από λειτουργίες I/O, που ονομάζεται πρόγραμμα I/O (I/O program). Το πρόγραμμα μπορεί να είναι αποθηκευμένο στον επεξεργαστή I/O ή μπορεί να είναι αποθηκευμένο στη μνήμη και να ανακτάται από τον επεξεργαστή I/O.

Πριν την ενεργοποίηση ενός επεξεργαστή I/O, το λειτουργικό σύστημα συνήθως προετοιμάζει ένα πρόγραμμα I/O που δηλώνει τις λειτουργίες I/O που πρέπει να γίνουν, όπως επίσης και το μέγεθος και τη διεύθυνση μεταφοράς για τυχόν αναγνώσεις ή εγγραφές. Στη συνέχεια ο επεξεργαστής I/O διαβάζει το πρόγραμμα I/O, εκτελεί τις λειτουργίες που περιγράφονται σε αυτό, και διακόπτει τον επεξεργαστή, μόνο όταν όλο το πρόγραμμα έχει τελειώσει.



Οι επεξεργαστές DMA είναι βασικά επεξεργαστές ειδικού σκοπού (συνήθως αποτελούνται από ένα μόνο κύκλωμα και δεν είναι προγραμματιζόμενοι), ενώ οι επεξεργαστές I/O υλοποιούνται συχνά με μικροεπεξεργαστές γενικού σκοπού οι οποίοι εκτελούν ένα εξειδικευμένο πρόγραμμα I/O.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

- Να εξηγήσετε τι είναι η απευθείας πρόσβαση μνήμης (DMA) και πως αυτή υλοποιείται.
- Να απαριθμήσετε τα στάδια στα οποία χωρίζεται μια μεταφορά DMA.
- Να συγκρίνετε τις απαντήσεις σας με την υποενότητα που μόλις διαβάσατε: «Η λειτουργία της απευθείας πρόσβασης μνήμης».



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να συγκρίνετε τους επεξεργαστές I/O με τους ελεγκτές DMA. Ποια είναι η βασική διαφορά ανάμεσα σε αυτούς τους δύο τύπους επεξεργαστών;



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Οι επεξεργαστές DMA είναι βασικά επεξεργαστές ειδικού σκοπού (συνήθως αποτελούνται από ένα μόνο κύκλωμα και δεν είναι προγραμματιζόμενοι), ενώ οι επεξεργαστές I/O υλοποιούνται συχνά με μικροεπεξεργαστές γενικού σκοπού οι οποίοι εκτελούν ένα εξειδικευμένο πρόγραμμα I/O.



DMA και το σύστημα μνήμης

Όταν ενσωματώνεται σε ένα σύστημα I/O η DMA, η σχέση ανάμεσα στο σύστημα μνήμης και τον επεξεργαστή αλλάζει. Χωρίς DMA, όλες οι προσβάσεις στο σύστημα μνήμης προέρχονται από τον επεξεργαστή και συνεπώς προχωρούν μέσω της μετάφρασης των διευθύνσεων και της πρόσβασης στην κρυφή μνήμη, ακριβώς σαν να είναι ο επεξεργαστής αυτός που παράγει τις αναφορές. Με τη χρήση DMA, ακολουθείται ένα άλλο μονοπάτι προς το σύστημα μνήμης, και το οποίο δεν περιλαμβάνει το μηχανισμό μετάφρασης διευθύνσεων ή την ιεραρχία της κρυφής μνήμης. Αυτή η διαφοροποίηση δημιουργεί ορισμένα προβλήματα τόσο με τα συστήματα ιδεατής μνήμης (virtual memory systems) όσο και με τα συστήματα με κρυφές μνήμες. Αυτά τα προβλήματα συνήθως επιλύονται με ένα συνδυασμό τεχνικών υλικού και την υποστήριξη του λογισμικού.

Τα συστήματα που θα γνωρίσουμε είναι:

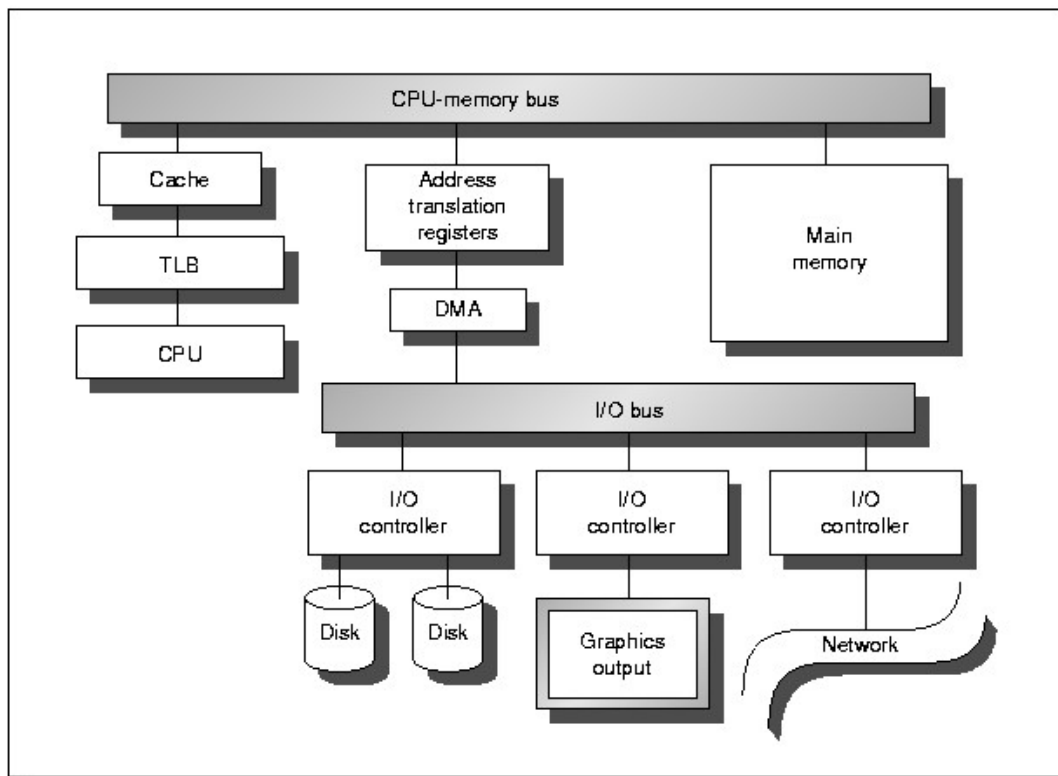
- ❑ DMA και ιδεατή μνήμη
- ❑ DMA και κρυφές μνήμες



Η DMA και η Ιδεατή μνήμη

Σε ένα σύστημα με ιδεατή μνήμη, η DMA πρέπει να εργάζεται με ιδεατές ή με φυσικές διευθύνσεις; Το προφανές πρόβλημα με τις ιδεατές διευθύνσεις είναι ότι η μονάδα DMA θα χρειάζεται να μεταφράζει τις ιδεατές διευθύνσεις σε φυσικές. Αυτό επιτυγχάνεται αν συμπεριλάβουμε στο μηχανισμό DMA κατάλληλους καταχωρητές-μεταφραστές διευθύνσεων, στους οποίους απεικονίζονται οι ιδεατές διευθύνσεις σε φυσικές. Το Σχήμα 5.9.2 απεικονίζει ένα τέτοιο σχήμα.

Το κύριο πρόβλημα με τις φυσικές διευθύνσεις είναι πως το μέγεθος των δεδομένων που μεταφέρονται δε μπορεί εύκολα να υπερβεί τα όρια μιας σελίδας μνήμης. Αν μία αίτηση I/O ξεπερνούσε το όριο μιας σελίδας, τότε οι θέσεις μνήμης στις οποίες θα μεταφερόταν τα δεδομένα από τη συσκευή δε θα ήταν συνεχόμενες στη φυσική μνήμη. Αυτό συμβαίνει επειδή οι θέσεις μνήμης θα αντιστοιχούσαν σε περισσότερες από μια ιδεατές σελίδες, καθεμιά από τις οποίες θα μπορούσε να απεικονιστεί σε οποιαδήποτε φυσική σελίδα, και φυσικές αυτές το πιθανότερο είναι να μην είναι γειτονικές μεταξύ τους. Κατά συνέπεια, αν χρησιμοποιούμε φυσικές διευθύνσεις, θα πρέπει να περιορίσουμε όλες τις μεταφορές DMA να παραμείνουν μέσα σε μία σελίδα.



Σχήμα 5.9.2 - Η Ιδεατή DMA απαιτεί έναν καταχωρητή για κάθε σελίδα που θα μεταφερθεί στον ελεγκτή DMA, δείχνοντας τα bits προστασίας και την φυσική σελίδα που αντιστοιχεί σε κάθε ιδεατή σελίδα.

■ Μέθοδοι για να ξεπεράσουμε το όριο σελίδας

Μία μέθοδος για να επιτρέψουμε στο σύστημα να εκκινεί μεταφορές DMA που διαπερνούν τα όρια σελίδων είναι να κάνουμε τη **DMA να λειτουργεί με ιδεατές διευθύνσεις**. Σε ένα τέτοιο σύστημα, η μονάδα DMA έχει ένα μικρό πλήθος εισόδων απεικόνισης (map entries) που παρέχουν ιδεατή προς φυσική απεικόνιση (virtual-to-physical mapping) για μία μεταφορά (σχήμα 9.2). Το λειτουργικό σύστημα παρέχει την απεικόνιση όταν ξεκινά το I/O. Χρησιμοποιώντας αυτή την απεικόνιση, η μονάδα DMA δε χρειάζεται να ανησυχεί για τη θέση των ιδεατών σελίδων που εμπλέκονται στη μεταφορά.

Μία άλλη τεχνική είναι **το λειτουργικό σύστημα να σπάσει τη μεταφορά DMA σε μία σειρά μεταφορών, καθεμιά από τις οποίες να περιορίζεται σε μία μόνο φυσική σελίδα**. Οι μεταφορές τότε συναρμολογούνται και οδηγούνται σε έναν επεξεργαστή I/O ή μία έξυπνη μονάδα DMA που εκτελεί την όλη ακολουθία μεταφορών· εναλλακτικά, το λειτουργικό σύστημα μπορεί να ζητήσει τις μεταφορές ξεχωριστά.



Οποιαδήποτε μέθοδος κι αν χρησιμοποιείται, το λειτουργικό σύστημα πρέπει να συνεργάζεται χωρίς να μετακινεί σελίδες εδώ κι εκεί, όση ώρα εξελίσσεται μία μεταφορά DMA που περιλαμβάνει αυτή τη σελίδα.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Πιστεύετε ότι συμφέρει η χρησιμοποίηση φυσικών ή ιδεατών διευθύνσεων μνήμης όσον αφορά την πρόσβαση σε αυτή μέσω DMA. Να αιτιολογήσετε την απάντησή σας.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Η απάντηση βρίσκεται στην κρίση του καθενός και στις απαιτήσεις μας όσον αφορά το λόγο απόδοση/κόστος. Με τη χρήση ιδεατών διευθύνσεων λύνουμε κάθε άλλο πρόβλημα, για την υλοποίηση αυτού του σχήματος χρειάζονται όμως ειδικοί καταχωρητές-μεταφραστές διευθύνσεων, γεγονός που αυξάνει το κόστος του συστήματος. Η χρήση των φυσικών διευθύνσεων είναι απλούστερη, έχει όμως το πρόβλημα ότι οι μεταφορές δεδομένων δεν μπορούν να είναι μεγαλύτερες από μια σελίδα μνήμης. Η επίλυση αυτού του προβλήματος αυξάνει και την πολυπλοκότητα του σχήματος με φυσικές διευθύνσεις.



Η DMA και οι κρυφές μνήμες

Οι δυσκολίες της ύπαρξης DMA σε συστήματα με κρυφές μνήμες προκύπτουν επειδή μπορεί να υπάρχουν δύο αντίγραφα ενός στοιχείου δεδομένων: ένα στην κρυφή μνήμη και ένα στην κύρια μνήμη. Επειδή ο επεξεργαστής DMA προωθεί τις αιτήσεις μνήμης απευθείας σε αυτή και όχι μέσω της κρυφής μνήμης, η τιμή μίας θέσης μνήμης που βλέπει η μονάδα DMA και αυτή που βλέπει ο επεξεργαστής μπορεί να διαφέρουν.

Θεωρήστε μία ανάγνωση από το δίσκο που η μονάδα DMA τοποθετεί απευθείας στη μνήμη. Αν μερικές από τις θέσεις μνήμης στις οποίες γράφει η DMA βρίσκονται στην κρυφή μνήμη, ο επεξεργαστής θα λάβει την παλιά τιμή όταν εκτελέσει μία ανάγνωση. Παρόμοια, αν η κρυφή μνήμη είναι τύπου εγγραφής write-back, η DMA μπορεί να διαβάσει μία τιμή απευθείας από την κύρια μνήμη, ενώ μια νεότερη τιμή βρίσκεται ήδη στην κρυφή μνήμη και δεν έχει γραφτεί πίσω στη μνήμη. Αυτό το πρόβλημα ονομάζεται πρόβλημα χαλασμένων δεδομένων (stale data problem) ή πρόβλημα συμφωνίας (coherency problem).

Αντιμετώπιση του προβλήματος

Το πρόβλημα αποφεύγεται χρησιμοποιώντας κάποια από τρεις κύριες τεχνικές.

➤ Μία προσέγγιση είναι η **δρομολόγηση της δραστηριότητας εισόδου/εξόδου μέσα από την κρυφή μνήμη**. Αυτό εξασφαλίζει ότι οι αναγνώσεις βλέπουν την πιο πρόσφατη τιμή ενώ οι εγγραφές ενημερώνουν τυχόν δεδομένα στην κρυφή μνήμη. Η δρομολόγηση όλου του I/O μέσα από την κρυφή μνήμη είναι ακριβή και πιθανώς έχει μεγάλη αρνητική επίδραση στην απόδοση του επεξεργαστή, αφού τα δεδομένα I/O σπάνια χρησιμοποιούνται αμέσως και για χάρη τους μπορεί να απομακρυνθούν από την κρυφή μνήμη χρήσιμα δεδομένα που χρειάζεται ένα εκτελούμενο πρόγραμμα, και συνεπώς και ο επεξεργαστής.

➤ Μία δεύτερη επιλογή είναι το **λειτουργικό σύστημα να ακυρώνει κατά βούληση την κρυφή μνήμη** για μία ανάγνωση I/O ή να επιβάλλει να συμβαίνουν υστεροεγγραφές για μία εγγραφή I/O (αυτό συχνά ονομάζεται εκκένωση –flushing- της κρυφής μνήμης). Αυτή η προσέγγιση δεν έχει μειονεκτήματα στο υλικό και είναι πιθανά πιο αποδοτική, αν το λογισμικό μπορεί να εκτελεί τη λειτουργία εύκολα και αποδοτικά. Επειδή αυτή η εκκένωση μεγάλων τμημάτων της κρυφής μνήμης χρειάζεται να συμβαίνει μόνο όταν υπάρχει πρόσβαση μπλοκ DMA, θα είναι σχετικά σπάνια.

➤ Η τρίτη προσέγγιση αποτελεί την παροχή ενός **μηχανισμού υλικού για την επιλεκτική εκκένωση (ή ακύρωση) των εισόδων της κρυφής μνήμης** (ενώ στην προηγούμενη επιλογή αυτή τη διαδικασία την εκπλήρωνε το λογισμικό). Η ακύρωση μέσω υλικού έτσι ώστε να εξασφαλίζεται η σύμπτωση μνήμης - κρυφής μνήμης είναι συνηθισμένη σε πολυεπεξεργαστικά συστήματα και η ίδια τεχνική μπορεί να χρησιμοποιηθεί για την είσοδο/έξοδο.

Έχουμε ρίξει μία ματιά σε τρεις διαφορετικές μεθόδους για τη μεταφορά δεδομένων ανάμεσα σε μία συσκευή I/O και τη μνήμη. Μεταβαίνοντας από τη μέθοδο της εξέτασης σε αυτή της οδηγούμενης από διακοπές και τέλος στη διασύνδεση DMA, **μετατοπίζουμε το βάρος της διαχείρισης μίας λειτουργίας I/O από τον επεξεργαστή σε ένα σταδιακά εξυπνότερο ελεγκτή I/O**.



Αυτές οι μέθοδοι έχουν το πλεονέκτημα της αποδέσμευσης κύκλων επεξεργαστή. Τα μειονεκτήματά τους είναι ότι αυξάνουν το κόστος του συστήματος I/O. Λόγω των παραπάνω, ένα δεδομένο υπολογιστικό σύστημα μπορεί να επιλέξει ποιο σημείο μέσα σε αυτό το φάσμα είναι κατάλληλο για το σύστημα εισόδου/εξόδου που διαθέτει.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Να απαριθμήσετε τις τεχνικές που χρησιμοποιούνται για να αντιμετωπίσουμε το πρόβλημα των χαλασμένων δεδομένων. Να συγκρίνετε την απάντησή σας με την παράγραφο: «Αντιμετώπιση του προβλήματος».



Ανακεφαλαιώνοντας λοιπόν...



Η απευθείας πρόσβαση στη μνήμη (DMA) χρησιμοποιείται για να απαλλάξει τον επεξεργαστή από τη συνεχή συμμετοχή του στις λειτουργίες εισόδου/εξόδου.



Η DMA υλοποιείται με έναν ελεγκτή, ο οποίος μεταφέρει δεδομένα μεταξύ της μνήμης και μιας συσκευής εισόδου/εξόδου, ενώ την ίδια στιγμή ο επεξεργαστής ασχολείται με άλλες διεργασίες.



Τρία στάδια πραγματοποιούνται σε μια μεταφορά DMA. Αρχικά ο επεξεργαστής αρχικοποιεί τον ελεγκτή DMA δίνοντάς του πληροφορίες σχετικές με τη μεταφορά. Στη συνέχεια ο ελεγκτής γίνεται κύριος της αρτηρίας και μεταφέρει τα δεδομένα απευθείας, χωρίς τη διαμεσολάβηση του επεξεργαστή. Τέλος ο ελεγκτής θέτει ένα σήμα διακοπής στον επεξεργαστή για να τον ειδοποιήσει ότι η μεταφορά ολοκληρώθηκε.



Η αποτελεσματικότητα του μηχανισμού DMA αυξάνεται περαιτέρω με την ταυτόχρονη χρησιμοποίηση των επεξεργαστών I/O , οι οποίοι αποτελούν ειδικευμένους επεξεργαστές που εκτελούν ένα συγκεκριμένο πρόγραμμα, απαλλάσσοντας περαιτέρω τον επεξεργαστή.



Δίλημμα αποτελεί η χρησιμοποίηση φυσικών ή ιδεατών διευθύνσεων στο μηχανισμό DMA. Οι ιδεατές διευθύνσεις μας αναγκάζουν να τις μετατρέπουμε συνεχώς σε φυσικές, ενώ με τις φυσικές πρέπει να χρησιμοποιήσουμε ειδικές τεχνικές για να μπορούμε να μεταφέρουμε ποσότητα δεδομένων μεγαλύτερη από το μέγεθος μιας σελίδας μνήμης.



Επειδή η DMA προσπελαίνει απευθείας τη μνήμη ανεξάρτητα από τον επεξεργαστή μπορούν να υπάρχουν προβλήματα συνέπειας με την κρυφή μνήμη. Τα δεδομένα που βλέπει ο ελεγκτής DMA στη μνήμη μπορεί να έχουν άλλη τιμή από αυτή που έχουν αν βρίσκονται στην κρυφή μνήμη.