Δημοκρίτειο Πανεπιστήμιο Θράκης
Τμήμα: Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Καθηγητής Δασυγένη Μηνάς

ΣΥΣΤΗΜΑΤΑ ΑΡΧΕΙΩΝ ΣΕ ΕΝΣΩΜΑΤΩΜΕΝΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Ο ρόλος ενός λειτουργικού συστήματος είναι να παρέχει ένα περιβάλλον όπου διάφορα προγράμματα μπορούν να τρέξουν ταυτόχρονα, με έναν ελάχιστο κίνδυνο ανεπιθύμητης παρέμβασης μεταξύ τους, αλλά με την υποστήριξη για την ασφαλή διανομή δεδομένων. Το λειτουργικό σύστημα πρέπει επίσης να παρέχει μια καθαρή διεπαφή στις εγκαταστάσεις του hardware της μηχανής. Η παρέμβαση μεταξύ των διαδικασιών ελαχιστοποιείται από τη διαχείριση της μνήμης και τα σχέδια προστασίας, τα οποία επιτρέπουν σε κάθε διαδικασία να έχει πρόσβαση μόνο στο δικό της χώρο της μνήμης. Στη διαδικασία δίνεται η δικιά της θέση της μνήμης του συστήματος και όταν ένας διακόπτης διαδικασίας πραγματοποιείται η έκταση της μνήμης μετασχηματίζεται δυναμικά σε αυτή της νέας διαδικασίας, με όλη τη μνήμη που χρησιμοποιείται από την προηγούμενη διαδικασία να αφαιρείται. Αυτό απαιτεί εξελιγμένη υποστήριξη hardware, για να λειτουργήσει αποτελεσματικά. Η διανομή δεδομένων συνεπάγεται ένα «παραθυράκι» στο σχέδιο προστασίας, το οποίο πρέπει να ελέγχεται με μεγάλη προσοχή. Η τυχαία πρόσβαση στις δομές που μοιράζονται μπορεί να οδηγήσει στις πιο δυσνόητες μορφές κακής συμπεριφοράς προγράμματος, έτσι μια πειθαρχημένη προσέγγιση πρέπει να εφαρμοστεί. Η πρόσβαση στις εγκαταστάσεις του hardware συχνά εμπλέκει πολύ χαμηλού επιπέδου χειρισμό bit και παρά να απαιτεί κάθε διαδικασία να το κάνει αυτό ανεξάρτητα, οι λεπτομέρειες αντιμετωπίζονται κεντρικά από το λειτουργικό σύστημα. Οι διαδικασίες μπορούν τότε να έχουν πρόσβαση στις συναρτήσεις εισόδου/εξόδου σε ένα υψηλό επίπεδο μέσω των calls συστημάτων.

Μεθοδολογίες Ανάπτυξης για Ενσωματωμένα Συστήματα

Παρότι δεν υπάρχει «τυπικός τρόπος» ανάπτυξης ενσωματωμένων συστημάτων, υπάρχουν κάποιες γενικές αρχές. Συστήματα τα οποία βασίζονται στην απόλυτη εκμετάλλευση πόρων από μικρής υπολογιστικής ισχύος μικροελεγκτές (π.χ. 8-bit) συχνά αναπτύσσονται σε γλώσσες assembly. Η προσέγγιση αυτή γίνεται όλο και σπανιότερη διότι ακόμη και σε 8-bit μικροελεγκτές υπάρχουν compilers που κάνουν πάρα πολύ καλή δουλειά, και επιτρέπουν την ανάπτυξη σε υψηλές γλώσσες προγραμματισμού όπως η C. Όσο περισσότερο ισχυρός είναι ο χρησιμοποιούμενος επεξεργαστής τόσο η ανάπτυξη μοιάζει με αυτήν σε συμβατικές αρχιτεκτονικές. Οι διεργασίες τρέχουν υποβοηθούμενες από λειτουργικά συστήματα πραγματικού χρόνου, και η δυνατότητα συντήρησης και εξέλιξης του κώδικα οδηγεί ακόμη και σε ανάπτυξη με γλώσσες όπως η C++ ή η Java. Επειδή όμως σε κάποιες από αυτές τις περιπτώσεις ο κώδικας δεν έχει ιδιαίτερα καλά χαρακτηριστικά όπως καλή ταχύτητα

και κατανάλωση ισχύος, κρίσιμα μέρη του προγράμματος συνεχίζουν να αναπτύσσονται σε γλώσσες όπως C και assembly. Αυθαίρετο παράδειγμα είναι ότι ενδεχόμενα σε ένα κινητό τηλέφωνο η επαφή με τον χρήστη μπορεί να είναι σε Java αλλά ένα φίλτρο που κάνει αναίρεση ήχους και θορύβου να είναι αναπτυγμένα σε C.

Περιβάλλοντα Ανάπτυξης

Εν γένει τα περιβάλλοντα ανάπτυξης δίνουν ολοκληρωμένες λύσεις για συγκεκριμένη κατηγορία επεξεργαστών. Αυτό περιλαμβάνει τόσο τα περιβάλλοντα λογισμικού όσο και την υποστήριξη του υπό ανάπτυξη συστήματος με προσομοιωτές και εντός κυκλώματος προσομοιωτές. Τα περιβάλλοντα ανάπτυξης περιλαμβάνουν:

• Cross compilers: οι cross compilers λειτουργούν όπως και οι compilers αλλά βγάζουν κώδικα assembly για διαφορετική αρχιτεκτονική από αυτήν στην οποία τρέχουν οι ίδιοι. Για παράδειγμα μπορεί να έχουμε έναν C compiler που να τρέχει σε αρχιτεκτονική Intel x86 (π.χ. σε PC) και ο οποίος βγάζει κώδικα για αρχιτεκτονική Atmel AVR.

• Cross assemblers: Ίδια με τους cross compilers αλλά παίρνουν σαν είσοδο γλώσσα assembly αρχιτεκτονικής άλλης από αυτήν στην οποία τρέχουν και βγάζουν εκτελέσιμο κώδικα αντίστοιχο με αυτόν της assembly που δέχονται. Για παράδειγμα, όταν σε προσωπικό υπολογιστή PC τρέχουμε assembler για αρχιτεκτονική MIPS, αυτός είναι cross assembler.

• Προσομοιωτές: Αυτοί προσομοιώνουν την λειτουργία κάποιου επεξεργαστή, όπως π.χ. το SPIM
• Εντός κυκλώματος προσομοιωτές (in-circuit emulators). Είναι προσομοιωτές που όχι μόνο δίνουν μία λειτουργική προσομοίωση ενός επεξεργαστή αλλά και αναπαράγουν τα χαρακτηριστικά του σε επίπεδο ακροδεκτών, δίνουν δηλαδή σε ένα συνδετήρα (connector) τα λειτουργικά χαρακτηριστικά κάποιου επεξεργαστή (π.χ. Intel 8086) αλλά με τις επί πλέον ιδιότητες του να βλέπει το σύστημα ανάπτυξης θέματα λειτουργίας όπως τα περιεχόμενα των καταχωρητών και της μνήμης. Επί πλέον ο χρήστης μπορεί να εισάγει στον κώδικα σημεία παρακολούθησης (breakpoints) να δημιουργήσει σήματα αντίστοιχα εξωτερικών σημάτων (όπως διακοπές – interrupts) και να αλλάξει κάποια στοιχεία, π.χ. περιεχόμενα καταχωρητών. Οι εντός κυκλώματος καταχωρητές δεν λειτουργούν πάντοτε στην ίδια μέγιστη ταχύτητα όπως οι επεξεργαστές που προσομοιώνουν.

Οι δομές δεδομένων που χρησιμοποιούμε στον προγραμματισμό έχουν άμεση σχέση με την υλοποίηση ενσωματωμένων συστημάτων, ειδικά επειδή μεγάλη κατηγορία ενσωματωμένων συστημάτων δεν είναι τίποτα περισσότερο από την υλοποίηση ενός αλγορίθμου σε υλικό. Το κόστος σε χώρο (π.χ. πύλες) και σε χρόνο (π.χ. χρόνος ολοκλήρωσης ενός έργου) είναι

σημαντικό για ένα τέτοιο σύστημα και σαν μηχανικοί έχουμε πληθώρα εναλλακτικών λύσεων και συμβιβασμών, από την επιλογή της τεχνολογίας υλοποίησης μέχρι την επιλογή αλγορίθμου.

Παρότι ενσωματωμένα συστήματα είναι μία ευρύτατη γκάμα εφαρμογών (όπως δείξαμε και στο Κεφάλαιο 1) και δεν έχει νόημα να γενικεύσουμε, εν τούτοις είναι συνηθισμένο για τέτοια συστήματα να έχουν περιορισμούς τόσο στον χώρο (πουενέργειας) και χρόνο (που συνήθως μεταφράζεται σε απόκριση πραγματικού χρόνου). Αυτό σημαίνει πως οι δομές που χρησιμοποιούμε στο υλικό, ακόμη και αν αυτές απεικονίζονται σε επεξεργαστή, έχουν περισσότερους περιορισμούς από ότι σε συνηθισμένα προγραμματιστικά περιβάλλοντα, και αυτοί οι περιορισμοί μπορεί να είναι η έλλειψη ιδεατής μνήμης με παράλληλα αυστηρά όρια στην φυσική μνήμη (ένας 8-bit μικροελεγκτής συχνά έχει διευθυνσιοδότηση μνήμης 64KBytes), καθώς και σε χρόνο απόκρισης.

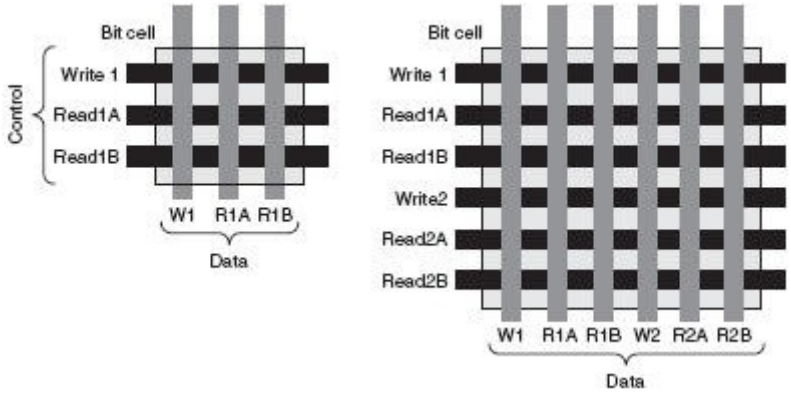
Πρακτικά αυτό σημαίνει πως πρέπει να έχουμε καλή κατανόηση των δομών δεδομένων που χρησιμοποιούμε, ξεκινώντας από την χρήση στοιβών (stacks και heaps). Υλοποίηση δομών δεδομένων όπως λίστες και στοίβες σε υλικό είναι αρκετά συνηθισμένο φαινόμενο, ενώ άλλες δομές (δένδρα, γράφοι, κλπ.) συναντιόνται μόνο σε περισσότερο εξειδικευμένες εφαρμογές και δεν έχουν ακόμη μελετηθεί επαρκώς για αποτελεσματική υλοποίηση σε ενσωματωμένα συστήματα υλικού. Για τους λόγους αυτούς καλή κατανόηση των δυνατοτήτων της τεχνολογίας υλοποίησης είναι απαραίτητη ώστε οι επιλεχθείσες δομές να υποστηρίζονται.

Σχεδιασμός αρχείων καταχωρητων

Πολλές διαστάσεις του σχεδιασμού register αρχείων καθορίζονται από τα κελιά και τις γραμμές των ίδιων των καταχωτητων αρχείων. Αν και το αρχείο καταχωρηρτη είναι συνδεδεμένο με άλλα τμήματα του chip, αυτές οι εσωτερικές διαστάσεις του σχεδιασμού μπορούν να εξεταστούν χωρίς να υπάρχουν ανησυχίες για αλληλεπιδράσεις με άλλα τμήματα του chip. Τα αρχεία καταχωρητων είναι συμπιεσμένα και στα τρανζίστορ και στους αγωγούς, με μια τακτική επαναλαμβανόμενη δομή
Παρ΄ όλα τα απέραντα πλεονεκτήματα από την εξέλιξη της τεχνολογίας, ο σχεδιασμός των αρχείων καταχωρητων παραμένει σημαντικό κομμάτι των ενσωματωμένων συστημάτων. Μαζί με το σύστημα παράκαμψης δικτύου, το αρχείο είναι ένα από τούς ρυθμιστές του ILP ενός πλήρως συνδεδεμένου συμπλέγματος.

Δομή αρχείου καταχωρητη

Στα αρχεία καταχωρητων κάθε πύλη ανάγνωσης και γραφής μπορεί να οδηγήσει ένα αρχείο καταχωτητων να διαβάσει ή να γράψει κατά τη διάρκεια της συχνότητας ρολογιού.
Όσον αφορά την πρόσβαση στο αρχείο του καταχωρητη, οι επεξεργαστές προσπελαύνουν τους καταχωρητες στο φυσικό τους μέγεθος, τυπικά 8,16 ή 32 bits. Κάθε bit του καταχωρητη αντιπροσωπεύεται από ένα *κελί bit*, το οποίο είναι το βασικό δομικό κομμάτι του συνόλου του αρχείου καταχωρητη.



Σχημα 1 Οραγανωση αρχείου καταχώρητη

Υπάρχουν δυο επίπεδα στην οργανωτική μορφή των αρχείων: α) πως τα κελια bit συγκροτούνται και β) πως λειτουργούν τα πράγματα μέσα σε κάθε bit κελί. Για τη δόμηση ενός αρχείου καταχωρητη, τα κελία bit συγκροτούνται ώστε να σχηματίσουν ξεχωριστους καταχωρητες και οι συγκροτούνται ώστε να σχηματίσουν ένα συνολικό register αρχείο.
Τα αρχεία αυτά είναι κατά κανόνα οργανωμένα σαν ένα δυσδιάστατο δικτύωμα αγωγών, π.χ., η οδός ελέγχου τρέχει οριζόντια και η οδός δεδομένων τρέχει κάθετα.

Έστω ένα RISC αρχείο, με 32 καταχωρητες, καθένα από τα οποία είναι 32 bit και έστω ένα αρχείο καταχωρητη υποστηρίζει μόνο μια ανάγνωση ή γραφή κάθε φορά. Σε αυτή τη περίπτωση, οι κάθετοι διοδοι δεδομένων αντιστοιχούν στη θέσεις του bit μέσα στη λέξη, ενώ οι οριζόντιοι οδοί ελέγχου επιλέγουν ένα μοναδικό register εκτός αρχείου. Η υποστήριξη μια από τις 32 γραμμές ελέγχου προκαλεί κάθε κελί bit στη συγκεκριμένη σειρά να τοποθετήσει δεδομένα στη γραμμή δεδομένων, επιτρέποντας την 32 bit τιμή του register να διαβάσει από τις οδούς των δεδομένων.

Στα αρχεία καταχωρητων με πολλαπλές πύλες, υπάρχουν πολλές επιπρόσθετες γραμμές ελέγχου και δεδομένων που συμπίπτουν, αφού οι γραμμές ελέγχου πρέπει να υποστηρίζονται ξεχωριστά και οι γραμμές δεδομένων πρέπει να πραγματοποιούνται ξεχωριστά ώστε να επιτευχθεί η ανεξάρτητη προσπέλαση στα κελία bit (και κατ' επέκταση οι registers).

Αρχεια καταχωρητων, Τεχνολογια, και Συμπλεγματα

Για οποιαδήποτε γενιά της VLSI τεχνολογίας, υπάρχει πάντα ένα σημείο από την άποψη του αριθμού των πυλών των αρχείων καταχωρητων το οποίο μπορεί να πραγματοποιηθεί χωρίς τη δημιουργία κρίσιμων οδών για την προσπέλαση του αρχείου καταχωρητη. Με άλλα λόγια, όταν γίνεται ο σχεδιασμός ενός αρχείου καταχωρητη δεν είναι αναγκαίο να ελαχιστοποιείται ο αριθμός των πυλών.

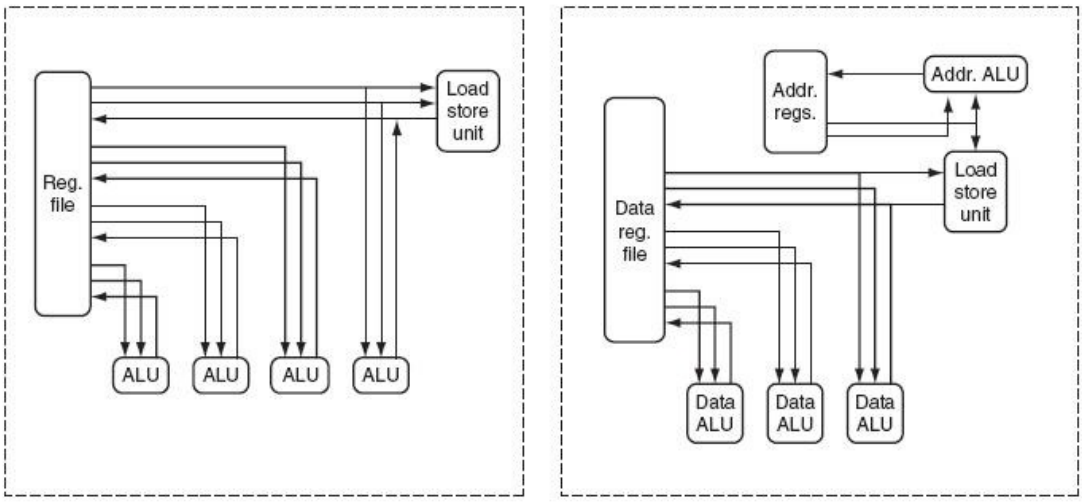
Για παράδειγμα, σε μία τυπική 0.18μ διαδικασία, μπορούν πολύ εύκολα να κατασκευαστούν γύρω στις 12 πύλες (8 για ανάγνωση + 4 για γραφή) και πιθανώς 15 με 20 πύλες με μια μικρή ταχύτητα συμβιβασμού. Ένας τέτοιος σχεδιασμός θα μπορούσε να υποστηρίξει πέντε με εφτά εκτελεστικές μονάδες με ταυτόχρονο συντελεστή προσπέλασης. Πέρα από το προαναφερθέν σημείο, πρόσθετες πύλες απαιτούν υψηλότερα οριακά κόστη (κατανάλωση δύναμης, πολυπλοκότητα σχεδιασμού), ενώ κάτω από το σημείο δεν χρισημοποιείται όλη η τεχνολογία που προσφέρεται. Παρ' 'όλα αυτά, οι εφαρμογές μπορούν να έχουν περισσότερα διαθέσιμα ILP από τις τεχνολογικές υποστηρίξεις του σημείου. Για να ξεπεραστούν οι περιορισμοί που θέτονται από το εύρος των register αρχείων, πρέπει να γίνει ο clustered σχεδιασμός. Κάθε σύμπλεγμα έχει το δικό του register αρχείο, σύστημα παράκαμψης δικτύου και λειτουργικές μονάδες. Το εύρος της οδού δεδομένων του συμπλέγματος θα βελτιστοποιηθούν για να ταιριάξουν με την υποκείμενη VLSI τεχνολογία. Το σύμπλεγμα μειώνει το μέγεθος του register αρχείου. Για παράδειγμα, διαιρώντας ένα μεγάλο αρχείο καταχωρητων σε δυο συμπλέγματα προκύπτουν δυο μικρά αρχεία καταχωρητων, το καθ' ένα από τα οποία έχει περίπου το ένα τέταρτο του μεγέθους του πρωτότυπου αρχείο. Παρ' όλα αυτά, ο σχεδιασμός συμπέγματος επιτρέπει στον πυρήνα να επιτύχει περισσότερα ILP στο κόστος της ανώτερης διαχείρισης του συμπλέγματος στο ISA, μικροαρχιτεκτοχνική (σύνδεση εννδοσυμπλέγματος) και επίπεδα μεταγλωττιστή. Αν και μειώνοντας τη register πίεση μέσα σε ένα σύμπλεγμα, το αρχιτεκτονικά ορατό σύμπλεγμα προσθέτει μια υψηλή register πίεση για την απομόνωση της register προσπέλασης ή τη ρήτη μετακίνηση των εντολών. Οι δίαυλοι μεταξύ των συμπλεγμάτων που εφαρμόζουν από μόνοι τους ή έπειτα από εντολή αντίγραφα είναι δαπανηροί πόροι και απαιτούν πύλες ανάγνωσης και γραφής και στις δυο πλευρές. Αυτό επιβάλλει ένα μικροαρχιτεκτονικό «φόρο» στο μέγεθος του register αρχείου ακόμα και μέσα σε ένα σύμπλεγμα και η έλλειψη πόρων απαιτεί ο compiler να κάνει χρήση τέτοιων κινήσεων με προσοχή.

Ξεχωριστες διευθυνσεις και δεδομενα αρχειων καταχωρητων

Ξεχωριστές διευθύνσεις και δεδομένα καταχωρητων έχουν πλεονεκτήματα των οποίων τα κόστη είναι μόνο εμφανή σε άλλες περιοχές, όπως η αρχιτεκτονική (κωδικοποίηση εντολών) και η compilation (πρόωρη τοποθέτηση απόφασης). Οι περισσότεροι VLIW σχεδιασμοί ακολουθούν τη φιλοσοφία του συνδυασμού του υπολογισμού ακεραίων και διευθετούν τις λειτουργίες σε ένα γενικού σκοπού χώρο. Από μικροαρχιτεκτονική άποψη, το ξερό τρανζίστορ που παρακινεί ξεχωριστές διευθύνσεις registers και τις μονάδες παραγωγής διευθύνσεων (AGUs) δεν χρησιμοποιείται πλέον και λίγες αρχιτεκτονικές διατηρούν το διαχωρισμό διεύθυνσης/δεδομένων. Παρόλα αυτά, μικρότερες διευθύνσεις και συμπλέγματα δεδομένων απαιτούν μικρότερα συστήματα παράκαμψης δικτύου, το οποίο μπορεί να οδηγήσει σε ευκολότερο routing και κοντύτερους αγωγούς – κλειδί στις σύγχρονές μικροαρχιτεκτονικές ανυσηχίες.

Οι ξεχωριστές διευθύνσεις και τα δεδομένα registers αυξάνουν την πολυπλοκότητα, αλλά εξοικονομούν hardware. Η περιοχή ενός register αρχείου αυξάνεται με το τετράγωνο του αριθμού των πυλών. Τα register αρχεία διευθύνσεων είναι πιθανόν να απαιτήσουν λιγότερες πύλες από ότι τα register αρχεία δεδομένων. Παρ'όλα αυτά, αν διαιρέσουμε ένα εννοποιημενο αρχειο καταχωρητων και αναθέσουμε ένα μέρος του σε ένα ξεχωριστό σύνολο διευθύνσεων registers με λιγότερες πύλες εξοικονομείται silicon περιοχή. Επιπροσθέτως, για ένα μικρότερο σύνολο register (όπως είναι οι registers διευθύνσεων) είναι ευκολότερο να κατασκευαστούν πιο περίπλοκοι μηχανισμοί, όπως είναι post-increments, rotations, and special address arithmetic units.

Η εικόνα δείχνει 2 πιθανές VLIW οργανώσεις που χρησιμοποιούν εννοποιημενα ή ξεχωριστά αρχεία καταχωρητων δεδομένων / διευθύνσεων. Παρομοίως, τα DSPs πλεονεκτούν ως προς από το διαχωρισμό AGU από το κύριο ALU (το οποίο μπορεί να χειριστεί fixed-point δεδομένα καλύτερα από ότι ακέραιους).



Σχημα 2. Διεύθυνση και αρχεία καταχωρητων

Χωρίς να αναφερθούν οι ιδιαιτερότητες μιας target τεχνολογίας είναι δύσκολο να προσδιοριστεί η ποσότητα των οικονομιών του hardware ενός ξεχωριστού συνόλου διευθύνσεων καταχωτητων. Τα μειονεκτήματα τέτοιων σχεδιασμών είναι πολλά και τα τεχνολογικές τάσεις δίνουν έμφαση σε ένα μέλλον στο οποίο αυτοί οι τύποι επιλογών θα έχουν λίγες συνέπειες στο κόστος, δεδομένου του σημείου στο σχεδιασμό των πολλαπλών πυλών των register αρχείων.

Αρχιτεκτονική ARM MMU

Μια MMU εκτελεί δύο αρχικές λειτουργίες:
• Μεταφράζει τις εικονικές διευθύνσεις σε φυσικές διευθύνσεις.
• Ελέγχει τις άδειες πρόσβασης στη μνήμη, ματαιώνοντας τις παράνομες προσβάσεις.

Η αρχιτεκτονική ARM MMU χρησιμοποιεί πίνακα σελίδων 2- επιπέδων και ένα TLB που αποθηκεύει τις πρόσφατα χρησιμοποιημένες μεταφράσεις σελίδων. Σε περιπτώσεις που ο επεξεργαστής έχει ξεχωριστές κρυφές μνήμες οδηγιών και στοιχείων είναι πιθανό επίσης να έχει χωριστά την οδηγία και τα στοιχεία TLB.

Κοκκοποίηση μνήμης (Memory granularity)

Η χαρτογράφηση μνήμης εκτελείται σε διάφορες διαφορετικές κοκκοποιήσεις από τον ίδιο
βασικό μηχανισμό. Οι μονάδες που μπορούν να χρησιμοποιηθούν είναι:

• **Τομείς.** Αυτοί είναι μπλόκ μνήμης 1 Mbyte.
• **Μεγάλες σελίδες.** Αυτές είναι μπλόκ μνήμης 64 Kbyte, και μέσα σε μια μεγάλη σελίδα
ο έλεγχος πρόσβασης εφαρμόζεται σε κομμάτια υποσελίδων 16 Kbyte**.**
• **Μικρές σελίδες.** Αυτές είναι μπλόκ μνήμης 4 Kbyte, και μέσα σε μια μεγάλη σελίδα
ο έλεγχος πρόσβασης εφαρμόζεται σε κομμάτια υποσελίδων 1 Kbyte.
• **Μικροσκοπικές σελίδες.** Μερικές από τις τελευταίες CPU υποστηρίζουν μικροσκοπικές σελίδες 1 Kbyte.

Η κανονική κοκκοποίηση είναι η μικρή σελίδα 4 kbyte. Οι μεγάλα σελίδες και οι τομείς υπάρχουν

Για να επιτρέπουν τη χαρτογράφηση μεγάλων περιοχών δεδομένων με μια ενιαία είσοδο TLB. Αναγκάζοντας μια μεγάλη περιοχή στοιχείων για να

χαρτογραφηθεί με μικρές σελίδες μπορεί, υπό ορισμένες συνθήκες, να οδηγήσει σε ανεπάρκεια του TLB.

Περιοχές

Οι περιοχές είναι ένα ασυνήθιστο χαρακτηριστικό γνώρισμα της αρχιτεκτονικής ARM MMU. Μια περιοχή είναι ομάδα τομέων ή/και σελίδων που έχουν ιδιαίτερες άδειες πρόσβασης. Αυτό επιτρέπει σε έναν αριθμό διαφορετικών διαδικασιών να τρέξουν με τους ίδιους πίνακες μεταφράσεων διατηρώντας ταυτόχρονα κάποια προστασία ο ένας από τον άλλον. Δίνει έναν πιό ελαφρύ μηχανισμό διακοπτών διαδικασίας από αυτόν που θα έδινε αν κάθε διαδικασία πρέπει να έχει τους δικούς της πίνακες μεταφράσεων.

Ο έλεγχος πρόσβασης είναι βασισμένος σε δύο είδη των προγραμμάτων:

• **Οι πελάτες** είναι χρήστες των περιοχών και πρέπει να παρατηρήσουν τις άδειες πρόσβασης των μεμονωμένων τομέων και των σελίδων που αποτελούν την περιοχή.

• **Οι διευθυντές** είναι οι ελεγκτές της περιοχής και μπορούν να παρακάμψουν τις άδειες πρόσβασης των μεμονωμένων τμημάτων ή των σελίδων.

.

Value	Status	Description
00	Καμία πρόσβαση περιοχών	Οποιαδήποτε πρόσβαση θα παραγάγει ένα ελάττωμα
01	Πελάτης	Τα κομμάτια άδειας σελίδων και τμημάτων ελέγχονται
10	Κατειλημμένη	Μην τη χρησιμοποιήείται
11	Διευθυντής	Τα κομμάτια άδειας σελίδων και τμημάτων δεν ελέγχονται

Πίνακας 1 Κομμάτια ελέγχου πρόσβασης περιοχών

Σε οποιοδήποτε χρόνο ένα πρόγραμμα μπορεί να είναι πελάτης μερικών περιοχών, διευθυντής σε μερικές άλλες περιοχές και δεν έχουν καμία πρόσβαση καθόλου στις υπόλοιπες περιοχές. Αυτό ελέγχεται από το CP15 με τον καταχωρητή 3 όποιος περιέχει δύο μπιτ για κάθε μια από τις 16 περιοχές περιγράφοντας τη θέση του τρέχοντος προγράμματος όσον αφορά κάθε περιοχή . Η ερμηνεία των δύο μπιτ δίνεται στον πίνακα. Η

σχέση ενός προγράμματος σε όλες τις περιοχές μπορεί να αλλάξει με το γράψιμο μιας ενιαίας νέας αξίας στον CP15 στον καταχωρητή 3.

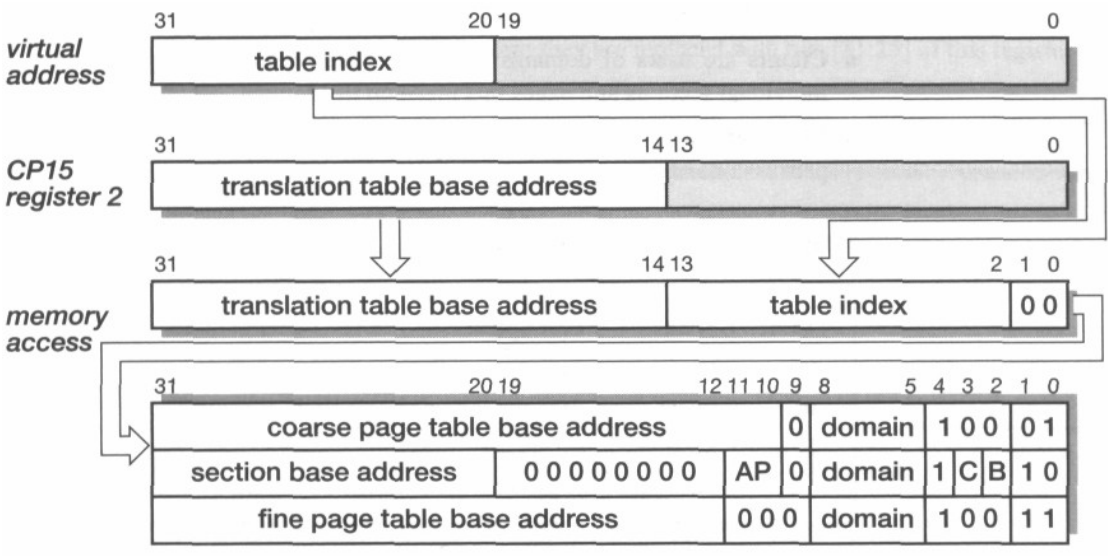
Διαδικασία μεταφράσεων

Η μετάφραση μιας νέας εικονικής διεύθυνσης αρχίζει πάντα με μια ευρύτητα. (Αγνοούμε για τώρα το TLB, όποια είναι μόνο μια κρυφή μνήμη για να επιταχύνει τη διαδικασία που περιγράφεται κατωτέρω.) Αυτό χρησιμοποιεί τη διεύθυνση βάσεων μεταφράσεων που κρατιέται μέσα στο CP15 στον καταχωρητή 2. Τα κομμάτια [31:14] του καταλόγου βάσεων μεταφράσεων συνδέονται με τα κομμάτια [31:20] της εικονικής διεύθυνσης για να διαμορφώσουν μια διεύθυνση μνήμης που χρησιμοποιείται για να έχει πρόσβαση στον περιγραφέα 1ου - επιπέδου, όπως φαίνεται στο σχήμα 11.3

Ο περιγραφέας 1ου επιπέδου μπορεί να είναι ή ένας τομών ή ένας δείκτης σε έναν πίνακα σελίδων δεύτερου επιπέδου ανάλογα με τα δύο κατώτατα μπιτ. Το '01 "δείχνει έναν δείκτη σε έναν χονδροειδή πίνακα σελίδων δεύτερου επιπέδου; το '10 " δείχνει έναν περιγραφέα τομέων; το '11 "δείχνει έναν δείκτη σε έναν λεπτό πίνακα σελίδων δεύτερου επιπέδου (υποστηρίζεται μόνο από ορισμένους CPU), ενώ το '00 "πρέπει να χρησιμοποιηθεί για να δείξει έναν περιγραφέα που προκαλεί ένα ελάττωμα μεταφράσεων.

Μετάφραση τομέων

Όταν ο περιγραφέας 1ου επιπέδου δείχνει ότι η εικονική διεύθυνση μεταφράζει σε ένα τομέα, η περιοχή ("περιοχή" στον περιγραφέα τμημάτων) ελέγχεται και, εάν η τρέχουσα διαδικασία είναι πελάτης της περιοχής, οι άδειες πρόσβασης (οι άδειες πρόσβασης ("AP" στον περιγραφέα τομέων) ελέγχονται επίσης. Εάν η πρόσβαση είναι επιτρεπόμενη, η διεύθυνση μνήμης διαμορφώνεται με τη σύνδεση των bit [31:20] από τον περιγραφέα τμημάτων με τα bit [19:0] ths εικονικής διεύθυνσης. Αυτή η διεύθυνση χρησιμοποιείται για να έχει πρόσβαση στα στοιχεία της μνήμης. Η πλήρης ακολουθία μεταφράσεων τομέων παρουσιάζεται στο σχήμα 11.4. Η διαδικασία των bits άδειας πρόσβασης (AP) περιγράφεται στις "άδειες πρόσβασης, και η διαδικασία bufferable (B) και cacheable (C) των bits περιγράφεται ενότητα «έλεγχος κρυφής μνήμης και buffer εγγραφής».

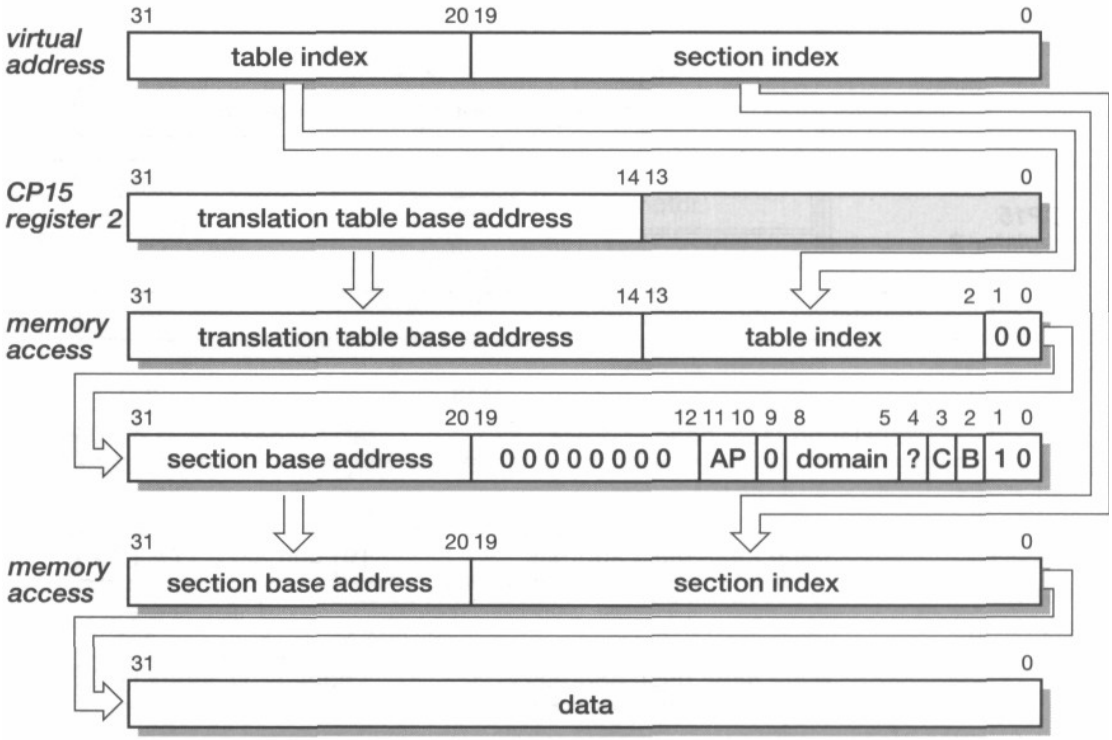


Σχημα 3 Πρωτου επιπεδου μεταφραση

Μετάφραση σελίδων

Όταν ο περιγραφέας 1ου επιπέδου δείχνει ότι η εικονική διεύθυνση μεταφράζει σε μια σελίδα, μια περαιτέρω πρόσβαση απαιτείται σε ένα πίνακα σελίδων 2ου επιπέδου. Η διεύθυνση ενός χονδροειδούς περιγραφέα σελίδων δεύτερου επιπέδου διαμορφώνεται με τη σύνδεση των bits [31:10] του περιγραφέα 1ου επιπέδου με τα bits [19:12] της εικονικής διεύθυνσης. Η διεύθυνση ενός λεπτού περιγραφέα σελίδων δεύτερου επιπέδου διαμορφώνεται με τη σύνδεση των bits [31:12] του περιγραφέα 1ου επιπέδου με τα bits [19:10] της εικονικής διεύθυνσης. Ο χονδροειδής περιγραφέας σελίδων δεύτερου επιπέδου μπορεί να είναι ένας μεγάλος (kbyte 64) περιγραφέας σελίδων ή ένας μικρός (kbyte 4) περιγραφέας σελίδων, ανάλογα με τα δύο κατώτατα bit. '01 "δείχνει μια μεγάλη σελίδα '10 "δείχνει μια μικρή σελίδα. Οι υπόλοιπες τιμές είναι παγιδευμένες, και το '00 "πρέπει να χρησιμοποιηθεί για να παραγάγει ένα ελάττωμα μεταφράσεων;το '11 "δεν πρέπει να χρησιμοποιηθεί. Ένας λεπτός περιγραφέας σελίδων δεύτερου επιπέδου μπορεί επίσης να είναι ένας μικροσκοπικός (1 kbyte) περιγραφέας σελίδων, που υποδεικνύεται από "11" στα δύο κατώτατα bits, ή μπορεί να είναι ένας μεγάλος ή μικρός περιγραφέας σελίδων όπως ανωτέρω .

Μια μικρή διεύθυνση βάσεων σελίδων κρατιέται στα bits [31:12] από τον περιγραφέα σελίδων. Τα bits [11:4] περιέχουν δύο bits άδειας πρόσβασης ('APO-3') για κάθε μια από τις τέσσερις υποσελίδες, όπου μια υποσελίδα είναι το ένα τέταρτο του μεγέθους της σελίδας. Τα bits [3:2] περιέχουν τα 'bufferable' και 'cacheable' bits. (Τα bits που έχουν '?' έχουν συγκεκριμένες χρήσεις.)



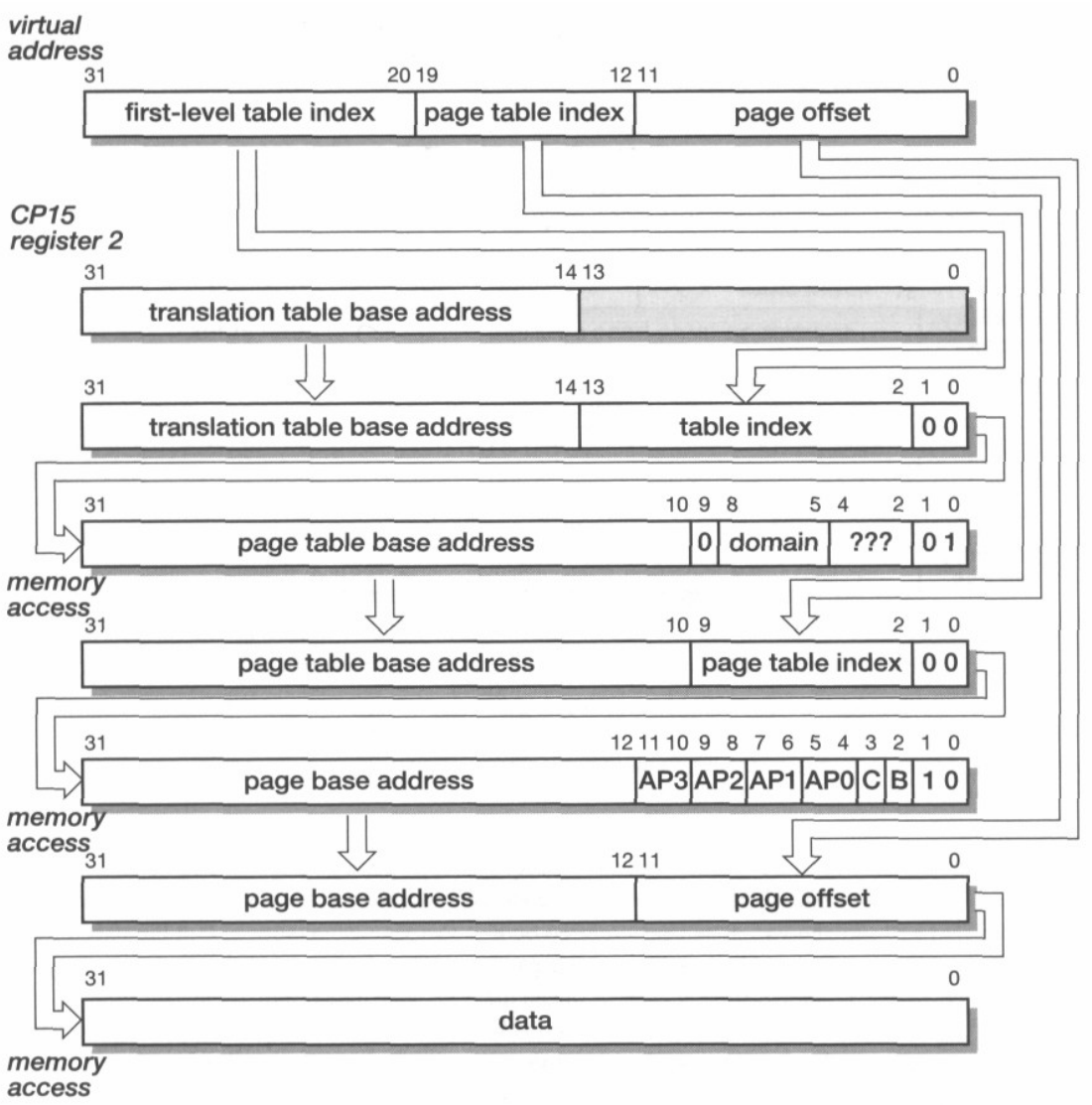
Σχημα 4 Ακολουθία μεταφράσεων τομέων.

Η γενική ακολουθία μεταφράσεων για μια μικρή σελίδα παρουσιάζεται στο σχήμα Σχημα 4 . Η ακολουθία μεταφράσεων για μια μεγάλη σελίδα είναι παρόμοια εκτός από τα bits [15:12] της εικονικής διεύθυνσης, τα οποία χρησιμοποιούνται και στον επιτραπέζιο δείκτη των σελίδων. Κάθε επιτραπέζια είσοδος σελίδων για μια μεγάλη σελίδα πρέπει επομένως να αντιγραφεί 16 φορές στον πίνακα σελίδων για κάθε τιμή αυτών των bit στον επιτραπέζιο δείκτη σελίδων.

Το μικροσκοπικό σχέδιο μεταφράσεων σελίδων είναι επίσης παρόμοιο, αλλά πρέπει να αρχίσει από έναν λεπτό περιγραφέα 1ου επιπέδου. Οι μικροσκοπικές σελίδες δεν υποστηρίζουν υποσελίδες και επομένως υπάρχει μόνο ένα σύνολο αδειών πρόσβασης στον περιγραφέα δεύτερου επιπέδου

Άδειες πρόσβασης

Τα AP bits για κάθε τμήμα ή υποσελίδα χρησιμοποιούνται μαζί με τις πληροφορίες περιοχών στον περιγραφέα 1ου επιπέδου, οι πληροφορίες ελέγχου περιοχών του CP15 στον καταχωρητή 3, τα bit ελέγχου S και R του CP15 του καταχωρητή 1 και η κατάσταση χρηστών/εποπτών του πεξεργαστή καθορίζει εάν επιτρέπει ανάγνωση ή γραφή σε κάποια θέση. Η διαδικασία που ελέγχει την άδεια συνεχίζει ως ακολούθως:



Σχημα 5 Μικρή ακολουθία μεταφράσεων σελίδων.

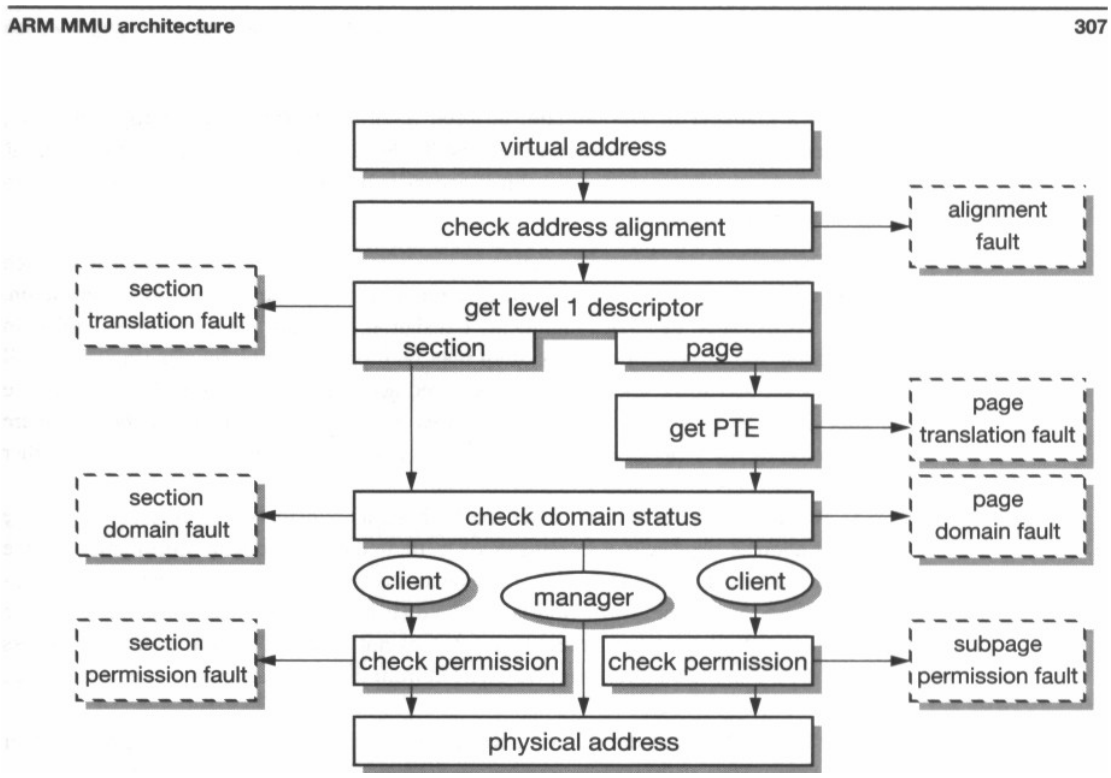
1. Εάν ο έλεγχος ευθυγράμμισης επιτρέπεται (το bit 1 του CP15 του καταχωρητή 1 ενεργοποιείται) ελέγξτε την ευθυγράμμιση διευθύνσεων και δείξε σφάλμα αν είναι μη- ευθυγραμμισμένη(δηλαδή εάν μια λέξη δεν ευθυγραμμίζεται σε ένα όριο 4 ψηφιολέξεων ή μισή λέξη δεν ευθυγραμμίζεται σε ένα όριο 2 ψηφιολέξεων).

2. Προσδιορίστε την περιοχή της εξετασμένης θέσης από τα bits [8:5] περιγραφέα 1^{ου} επιπέδων.

3. Έλενξε στο CP15 τον καταχωρητή 3, τον κατάλογος ελέγχου πρόσβασης περιοχών, εάν η παρούσα διαδικασία είναι πελάτης ή διευθυντής αυτής της περιοχής. Εάν δεν ισχύει κανένα από τα παραπάνω, δήλωσε σφάλμα.

4. Εάν είναι διευθυντής αυτής της περιοχής, προχωρά αγνοώντας τις άδειες πρόσβασης. Εάν είναι πελάτης, ελέγξε τις άδειες πρόσβασης σε σχέση με

χρησιμοποιώντας τα bits S και R από τον καταχωρητή 1 του CP15. Δήλωσε σφάλμα εάν η πρόσβαση δεν επιτρέπεται, διαφορετικά συνέχισε.



Σχημα 6 **Ελενγχος αδειας προσβασης**

Αδεια προσβασης			
AP	S	R	Supervisor User
00	0	0	No access No access
00	1	0	Read only No access
00	0	1	Read only Read only
00	1	1	Do not use
01	-	-	Read/write No access
10	-	-	Read/write Read only
11	-	-	Read/write Read/ write

Ο έλεγχος αδειοδότητησης φαίνεται στο σχήμα 6 , το οποίο παρουσιάζει διάφορα ελαττώματα που μπορούν να παραχθούν κατά τη διάρκεια μιας μετάφρασης διευθύνσεων. Το MMU μπορεί να παραγάγει τα ελαττώματα ευθυγράμμισης, μεταφράσεων, περιοχών και άδειας. Επιπλέον, το εξωτερικό σύστημα μνήμης μπορεί να σφάλει στις ευρύτητες γραμμών κρυφής μνήμης (αν και δεν υποστηρίζουν όλες οι CPUs αυτό),σε προσβάσεις χωρίς κρυφή μνήμη ή χωρίς buffer και επιτραπέζιες προσβάσεις μεταφράσεων. Αυτά τα ελαττώματα όλα καλούνται

ματαίωση(abort) και αντιμετωπίζονται από τον επεξεργαστή ανάλογα με το εάν η πρόσβαση ήταν για μια οδηγία ή για τα στοιχεία.

Ένα ελάττωμα σε μια πρόσβαση στοιχείων προκαλεί τον καταχωρητή σφαλμάτων (CP15 καταχωρητής 5) και τον καταχωρητή διευθύνσεων σφαλμάτων (CP15 καταχωρητής 6) να ενημερωθούν για να παρέχουν τις πληροφορίες για την αιτία και τη θέση του σφάλματος. Ένα σφάλμα σε μια πρόσβαση οδηγίας προκαλεί μόνο μια εξαίρεση εάν και όταν εκτελείται η οδηγία και δεν ενημερώνει τη θέση ελαττωμάτων και δεν εξετάζει τους καταλόγους. Η διεύθυνση σφαλμάτων μπορεί να συναχθεί από την επιστροφής διεύθυνση στον κατάλογο συνδέσεων.

‘Ελεγχος κρυφής μνήμης και buffer εγγραφής

Τα bits C και B στους τομείς και τους περιγραφείς σελίδων 2^{ου} επιπέδου ελέγχει εάν τα στοιχεία στο τμήμα ή τη σελίδα μπορούν να αντιγραφούν σε μια κρυφή μνήμη ή να γραφούν στη μνήμη μέσω της εγγραφής σε ένα buffer.

Εκεί που η κρυφή μνήμη χρησιμοποιεί μέθοδο εγγραφής, το C ελέγχει εάν τα στοιχεία είναι ή όχι αποθηκεύσιμα σε μια κρυφή μνήμη και ο B ελέγχει εάν ή όχι μπορούν να αποθηκευθούν

Εξωτερικά σφάλματα

Σημειώστε ότι ο επεξεργαστής δεν μπορεί να επανέλθει από τα εξωτερικά σφάλματα, επειδή ώσπου να επισημανθεί το σφάλμα, ο επεξεργαστής μπορεί να είχε εκτελέσει διάφορες εντολές και είναι επομένως ανίκανος να ανακτήσει την κατάσταση που ήταν για να διορθώσει το σφάλμα. Όπου η αποκατάσταση απαιτείται πρέπει να χρησιμοποιηθούν εγγραφές χωρίς buffer.

Στις χαρακτηριστικές εφαρμογές ARM δεν υπάρχει καμία πιθανότητα ανάκτησης εξωτερικών σφαλμάτων, έτσι αυτό δεν είναι ένα ζήτημα.

Καταχωρητές CP15 προστασίας της μονάδας

Η δομή του καταχωρητή προστασίας της μονάδας διευκρινίζεται στον πίνακα. Οι καταχωρητές διαβάζονται και γράφονται χρησιμοποιώντας την εντολή CP15 που φαίνεται |στο σχημα 7, με CRn να προσδιορίζει την είσοδο στον καταχωρητή. Πιο αναλυτικά, οι λειτουργίες του καταχωρητή είναι οι ακόλουθες:

Καταχωρητής	Σκοπός
0	ID καταχωρητή
1	Διαμόρφωση
2	Έλεγχος προσωρινής μνήμης
3	Έλεγχος οδηγού γραμμής (Buffer)
5	Άδεια εισόδου
6	Βάση και μέγεθος
7	Λειτουργία προσωρινής μνήμης
9	Τερματισμός προσωρινής μνήμης
15	Δοκιμή
4,8,10-14	Δεν χρησιμοποιούνται

Σχημα 7 Δομή του καταχωρητή προστασίας της μονάδας

31 28	27 24	23 21	20	19 16	15 12	11 8	7 5	4	3 0
cond	1110	000	L	CRn	Rd	1111	Cop2	1	CRm

Σχημα 8 *CP15 εντολές μεταφοράς καταχωρητή*

Ο Καταχωρητής 0 (ο οποίος είναι ανάγνωσης μόνο) επιστρέφει πληροφορίες στη συσκευή αναγνώρισης.

31 24	23 16	15 4	3 0
implementer	architecture	past number (BCD)	revision

Τα ψηφία [3:0] περιέχουν ένα επαναλαμβανόμενο αριθμό, τα ψηφία [15:4] περιέχουν ένα τριψήφιο μέρος αριθμού σε δυαδικό-κωδικοποιημένο δεκαδικό σύστημα, τα ψηφία[23:16] περιέχουν την αρχιτεκτονική εκδοχή (0 για εκδοχή 3, 1 για εκδοχή 4, 2 για εκδοχή 4T, 4 για εκδοχή 5T) και τα ψηφία [31:24] περιέχουν τον κώδικα ASCII ενός υλοποιητή χαρακτηριστικών (ASCII ‘A’=4116 υποδηλώνει περιορισμένο ARM, ‘D’=4416 υποδηλώνει ψηφιακό, κ.τ.λ.). Κάποια CPUs δεν ακολουθούν με ακρίβεια την παραπάνω διάταξη του καταχωρητή 0 και τα καινούρια CPUs έχουν ένα δεύτερο καταχωρητή 0

(προσβάσιμο, αλλάζοντας τον τομέα COP2 σε εντολή MRC) που δίνει λεπτομέρειες για την οργάνωση της προσωρινής μνήμης.

Ο **Καταχωρητής 1** (ο οποίος είναι και εγγραφής και ανάγνωσης) περιέχει κάποια ψηφία του ελέγχου πληροφοριών, τα οποία ενεργοποιούν τις λειτουργίες του συστήματος και ελέγχουν τις παραμέτρους του.

31	30	29 28	27	26 25	24	23 14	13	12	11 18	7	6 4	3	2	1	0
iA	nf	Bnk	F	Lck	S	0000000000	V	I	0000	B	000	W	C	0	M

Όλα τα bits είναι απενεργοποιημένα κατά την επαναφορά συστήματος. Αν διαδοχικά ενεργοποιηθεί το σύστημα, το M ενεργοποιεί την προστασια της μονάδας, το C ενεργοποιεί τα δεδομένα ή την ενεργοποιημένη μνήμη, το W ενεργοποιεί το buffer εγγραφής, το B αλλάζει από μικρό σε μεγάλο τελικό byte διαδοχής, το I ενεργοποιεί την προσωρινή μνήμη των εντολών όταν αυτή είναι χωρισμένη από την προσωρινή μνήμη δεδομένων, το V προκαλεί τα ανύσματα εξαίρεσης να κινηθούν δίπλα στην κορυφή του χώρου διευθύνσεων, τα S, Lck, F και Bnk χρησιμοποιούνται για να ελέγξουν την προσωρινή μνήμη (στον ARM740T) και τα nf και iA ελέγχουν διαφόρους μηχανισμούς του ρολογιού παλμών (στους ARM 940T). Θα πρέπει να σημειωθεί ότι όλα τα bit δεν συμμετέχουν σε όλες τις εφαρμογές.

Ο **Καταχωρητής 2** (ο οποίος είναι και εγγραφής και ανάγνωσης) ελέγχει την ικανότητα της προσωρινής μνήμης των 8 ιδιαιτέρων (ή μεμονωμένων) περιοχών προστασίας.

31 8	7	6	5	4	3	2	1	0
00000000000000000000000	7	6	5	4	3	2	1	0

Το bit 0 ενεργοποιεί την προσωρινή μνήμη για να φορτώσει μέσα στην περιοχή 0, το bit 1 ομοίως στην περιοχή 1, κ.τ.λ.. το ARM 9470T έχει ξεχωριστή προστασία μονάδας στις θύρες εντολών και δεδομένων και το COP2 χρησιμοποιείται για να προσδιορίσει ποια μονάδα έχει προσπελαστεί:
COP2=0 δίνει προσπέλαση της μονάδας προστασίας στη θύρα δεδομένων.
COP2=1 δίνει προσπέλαση της μονάδας προστασίας στη θύρα εντολών.

Ο **καταχωρητής 3** (ο οποίος είναι και εγγραφής και ανάγνωσης) προσδιορίζει αν πρέπει ή δεν πρέπει να χρησιμοποιηθεί το buffer εγγραφής για κάθε περιοχή προστασίας. Το μοντέλο του είναι ίδιο με αυτό του καταχωρητή 2 αλλά όπως η θύρα εντολών του ARM940T είναι μόνο για ανάγνωση, το buffer εγγραφής μπορεί να ενεργοποιηθεί για τη θύρα δεδομένων και έτσι το COP2 πρέπει πάντα να είναι μηδέν.

Ο **καταχωρητή 5** (ο οποίος είναι και εγγραφής και ανάγνωσης) προσδιορίζει την έγκριση προσπέλασης για κάθε περιοχή προστασίας.

31 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
0000000000000000000000000	ap7	ap6	ap5	ap4	ap3	ap2	ap1	ap0

Η έγκριση προσπέλασης περιλαμβάνει (με 00) μη προσπέλαση, με (01) μόνο προνομιούχα λειτουργία, με (10) προνομιούχα πλήρης προσπέλαση και μόνο ανάγνωση από τη χρήση και τέλος με (11) πλήρη προσπέλαση. Ξανά ο ARM940T χρησιμοποιεί το τομέα COP2 για να διαφοροποιήσει τη μονάδα προστασίας εντολής (1) και δεδομένου (0).

Ο **καταχωρητής 6** (ο οποίος είναι και εγγραφής και ανάγνωσης) προσδιορίζει την αρχική διεύθυνση και το μέγεθος κάθε μιας από τι 8 περιοχές.

31 12	11 6	5 1	0
Region base address	000000	size	E

Η διεύθυνση βάσης πρέπει να είανια πολλαπλάσιο το μεγέθους της.. Η είδική περιοχή καθορίζεται στο τομέα CRm, η οποία πρέπει να οριστεί από το 0 ως το 7. για να επεξεργατεί ο πυρήνας αρχιτεκτονικής Harvard όπως ο ARM940T υπάρχουν διαφορετικές περιοχές καταχωρητών για την εντολή και τις θύρες μνήμης δεδομένων και το COP2 προσδιορίζει ποια θύρα μνήμης προκύπτει να διευθυνσιοδοτηθεί όπως περιγράφηκε παραπάνω για τον καταχωρητή 2.

Ο **καταχωρητής 7** ελέγχει διάφορες λειτουργίες της προσωρινής μνήμης και η λειτουργία του είναι διαφορετική για τν ARM940T και για τον ARM740T.
Ο **καταχωρητής 9** χρησιμοποιείται στον ARM940T για να τερματισει περιοχές της προσωρινής μνήμης (0 ARM740T χρησιμοποιεί κάποια bits του καταχωρήτή 1 γι΄αυτό το σκοπό).

Ο **καταχωρητής 15** χρησιμοποιείται στον ARM940T για να τροποποιήσει τον αλγόριθμο διαχωρισμού της προσωρινής μνήμης από τυχαίο σε συγκεκριμένο. Αυτό προτείνεται για τη χρήση μόνο κατά τη διάρκεια δοκιμής παραγωγής μικροεπεξεργαστή πυριτίου.

Βιβλιογραφία

1. Addison Wesley - Steve Furber - ARM System-on-Chip Architecture (2nd Edition)

2. Real-Time and Embedded Computing Systems and Applications (Springer-2005) Files

3. Πηγές από internet