 Δημοκρίτειο Πανεπιστήμιο Θράκης  
Πολυτεχνική Σχολή Ξάνθης  
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ

Σχεδιασμός Ενσωματωμένων Κυκλωμάτων

Εργασία 3η

Βενέτης-Παρασκευάς Παλληκαράς ΑΜ: 56857

Γουδελής Γεώργιος ΑΜ: 56843

Ο στόχος της συγκεκριμένης εργασίας είναι να δημιουργήσουμε «διατάξεις» οι οποίες θα λειτουργούν ως buffer που θα διατηρούν τα δεδομένα για (όσο χρειάζεται ) έτσι ώστε να τα φορτώνω μια φορά από την ram και όχι κάθε φορά που θα τα χρειάζομαι. Με αυτό το τρόπο γλυτώνω όλους τους παραπάνω κύκλους μνήμης άρα και καθυστέρησης. Στον αρχικό κώδικα έχουμε μια cache η οποία περιέχει μόνο τον πίνακα g της συνέλιξης ενώ στα βελτιστοποιημένα έχουμε εισάγει τους buffer επαναχρησιμοποίησης στην διάταξη αυτή.

Στη συνάρτηση conv2D έχω ένα buffer 5 γραμμών, όσο και το μέγεθος Ν της μάσκας του convolution. Αυτόν τον buffer τον χρησιμοποιώ για να φορτώνω την padded\_image. Έτσι δεδομένου ότι στο convolution κανονικά θα χρειαζόμουν να φορτώνω συνεχώς τα ίδια κελιά από τη μνήμη, δεδομένου ότι η συνέλιξη εφαρμόζεται πάνω από μια φορά στο ίδιο κελί, με την χρήση του buffer τα φορτώνω μια φορά και έπειτα επικοινωνώ με τον buffer ο οποίος βρίσκεται στην cache άρα όλο το overhead βρίσκεται μόνο στο πρώτο φόρτωμα από τη μνήμη καθώς η επικοινωνία με την cache είναι σχεδόν άμεσα. Στο τέλος της εφαρμογής του convolution για μια δεδομένη σειρά ανανεώνω του buffer σαν κάθετο κυλιόμενο παράθυρο. Μετακινώ τις σειρές 1-4 του buffer στις σειρές 0-3 του buffer και εφόσον είναι μέσα στο buffer αυτή η διεργασία δεν δαπανώ wait cycles για την επικοινωνία με την RAM. Έπειτα στη 4η σειρά φορτώνω τη νέα γραμμή από τη RAM. Κάθε κελί κανονικά έπρεπε να φορτωθεί ΝxΝ φορές (εξαιρώ γωνίες όπου είναι λιγότερες) ενώ τώρα το φορτώνουμε μόνο μια φορά.

Στα final steps δεν έχει νόημα η επαναχρησιμοποίηση δεδομένων, καθώς το κάθε κελί το χρησιμοποιώ μόνο μια φορά. Άρα είτε το φορτώνω αμέσως, είτε στον buffer/cache είναι σχεδόν το ίδιο, δεδομένου ότι δαπανώ και στα δυο τον ίδιο χρόνο καθυστέρησης από το μνήμη . Εδώ θα βοηθούσε αν είχαμε write με burst σε μια cache όπου θα τραβούσε πιο γρήγορα όλη την σειρά του πίνακα (ή και παραπάνω) από ότι αν τράβαγα ένα ένα τα κελιά. Ομοίως ούτε για την δημιουργία του g θα μπορούσε να χρησιμοποιηθεί τέτοια διάταξη.

Παρακάτω φαίνεται η εφαρμογή της επαναχρησιμοποίησης στους 3 καλυτέρους κώδικες των προηγούμενων εργασιών.

Παρακάτω φαίνονται τα στατιστικά του κώδικα 3\_8 πριν και μετά την βελτιστοποίηση:

|  |  |  |
| --- | --- | --- |
|  | 3\_8 clean | 3\_8 optimized |
| Instructions | 360356445 | 361119532 |
| Core\_Cycles | 510692272 | 512192185 |
| S\_Cycles | 404807348 | 405073301 |
| N\_Cycles | 79942560 | 81077065 |
| I\_Cycles | 96349426 | 96282147 |
| C\_Cycles | 0 | 0 |
| Wait\_States | 413853340 | 385898380 |
| Total | 994952674 | 968330893 |
| True\_Idles | 38713078 | 38597410 |
| Grand Totals | | |
| Code | 18124 | 18268 |
| RO Data | 470 | 470 |
| RW Data | 0 | 0 |
| ZI Data | 1632820 | 1639940 |
| Debug | 14292 | 14400 |

Παρακάτω φαίνονται τα στατιστικά του κώδικα 5 πριν και μετά την βελτιστοποίηση:

|  |  |  |
| --- | --- | --- |
|  | 5  clean | 5  optimized |
| Instructions | 351892613 | 352757076 |
| Core\_Cycles | 483981334 | 486799135 |
| S\_Cycles | 392744668 | 393111997 |
| N\_Cycles | 69551807 | 71294568 |
| I\_Cycles | 92091921 | 92632898 |
| C\_Cycles | 0 | 0 |
| Wait\_States | 331333276 | 303378316 |
| Total | 885721672 | 860417779 |
| True\_Idles | 38713078 | 38597410 |
| Grand Totals | | |
| Code | 18428 | 18576 |
| RO Data | 470 | 470 |
| RW Data | 0 | 0 |
| ZI Data | 1632820 | 1639940 |
| Debug | 14288 | 14436 |

Παρακάτω φαίνονται τα στατιστικά του κώδικα 6 πριν και μετά την βελτιστοποίηση:

|  |  |  |
| --- | --- | --- |
|  | 6  clean | 6  optimized |
| Instructions | 351892509 | 352756972 |
| Core\_Cycles | 483981203 | 486799004 |
| S\_Cycles | 392744547 | 393111876 |
| N\_Cycles | 69551792 | 71294553 |
| I\_Cycles | 92091926 | 92632903 |
| C\_Cycles | 0 | 0 |
| Wait\_States | 331333342 | 303378382 |
| Total | 885721607 | 860417714 |
| True\_Idles | 38713078 | 38597410 |
| Grand Totals | | |
| Code | 18716 | 18864 |
| RO Data | 470 | 470 |
| RW Data | 0 | 0 |
| ZI Data | 1632820 | 1639940 |
| Debug | 14520 | 14660 |

Σύγκριση των επιμέρους βελτιστοποιημένων κωδίκων μεταξύ τους:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3\_8 | 5 | 6 |
| Instructions | 361119532 | 352757076 | 352756972 |
| Core\_Cycles | 512192185 | 486799135 | 486799004 |
| S\_Cycles | 405073301 | 393111997 | 393111876 |
| N\_Cycles | 81077065 | 71294568 | 71294553 |
| I\_Cycles | 96282147 | 92632898 | 92632903 |
| C\_Cycles | 0 | 0 | 0 |
| Wait\_States | 385898380 | 303378316 | 303378382 |
| Total | 968330893 | 860417779 | 860417714 |
| True\_Idles | 38597410 | 38597410 | 38597410 |

Από τα παραπάνω προκύπτει ότι τα καλύτερα αποτελέσματα είναι αυτά του 5 με μικρή διαφορά από αυτά του 6 όμως. Καλυτέρα αποτελέσματα θα μπορούσαμε να είχαμε αν υπήρχε μια ολοκληρωμένη διάταξη cache όπως αυτή που υλοποιήσαμε στη προηγούμενη εργασία καθώς θα εκμεταλλευόμασταν την λειτουργία burst και θα μπορούσαμε να είχαμε περισσότερες πληροφορίες «πιο κοντά» στον επεξεργαστή.

Δεδομένου του μικρού μεγέθους cache που χρειαζόμαστε στην συγκεκριμένη εργασία αλλάζουμε την διευθυνσιοδότηση της μνήμης και χρησιμοποιούμε μικρότερη cache που να χωράει πλέον μόνο τον πίνακα g και της 5 γραμμές του buffer καθώς και λίγο κενό χώρο για καλύτερη ευθυγράμμιση των διευθύνσεων. Πιο συγκεκριμένα :

ROM 0x0 0x7BC000

{

ROM 0x0 0x8000

{

\*.o ( +RO )

}

DRAM 0x8000 0x07B0000

{

\* ( +RW )

\* ( +ZI )

}

SRAM 0x7B8000 0x4000

{

\* ( cache )

}

}