

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

Σχεδιασμός Ενσωματωμένων Κυκλωμάτων

Εργασία 2η

Βενέτης-Παρασκευάς Παλληκαράς ΑΜ: 56857

Γουδελής Γεώργιος ΑΜ: 56843

Στην συγκεκριμένη εργασία εισάγαμε ένα σύστημα τριών επιπέδων, (καταχωρητές)/ cache/RAM/ROM. Στην αρχή πήραμε μετρήσεις χωρίς να υπάρχει η cache, όπου και σαφώς πήραμε τα χειρότερα αποτελέσματα. Αρχικά, δοκιμάσαμε όλα τα αρχεία από την προηγούμενη εργασία και κρατήσαμε τα τρία καλύτερα, τα αποτελέσματα αυτών φαίνονται παρακάτω :

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3\_8 | 5 | 6 |
| Instructions | 360356443 | 351892614 | 351892509 |
| Core\_Cycles | 510692268 | 483981333 | 483981205 |
| S\_Cycles | 404807346 | 392744669 | 392744548 |
| N\_Cycles | 79942559 | 69551806 | 69551793 |
| I\_Cycles | 96349425 | 92091920 | 92091926 |
| C\_Cycles | 0 | 0 | 0 |
| Wait\_States | 449336140 | 366816076 | 366816142 |
| Total | 1030435470 | 921204471 | 921204409 |
| True\_Idles | 38713078 | 38713078 | 38713078 |
| Grand Totals | | | |
| Code | 18468 | 18424 | 18716 |
| RO Data | 470 | 470 | 470 |
| RW Data | 0 | 0 | 0 |
| ZI Data | 1632820 | 1632820 | 1632820 |
| Debug | 14408 | 14436 | 14568 |

Μετά θεωρήσαμε την ιδανική περίπτωση όπου η cache είναι αρκετά μεγάλη για να χωρέσει όλα τα δεδομένα. Τα Wait States εδώ οφείλονται στο «φόρτωμα» της φωτογραφίας και της αποθήκευσης της, αλλά και από τους κύκλους που «χάνονται» για την μεταφορά των δεδομένων των καταχωρητών στο stack, όταν γίνεται κλήση μιας άλλης συνάρτησης (και πιθανών interrupt), αλλά και την επιστροφή τους στους καταχωρητές όταν επιστρέφουμε πίσω. Επίσης, δαπανάται χρόνος από την μεταφορά των δεδομένων από το heap (μνήμη) στους καταχωρητές.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Αρχικό 3\_8 | 3\_8 | Αρχικό 5 | 5 | Αρχικό  6 | 6 |
| Instructions | 360356443 | 360356440 | 351892614 | 351892611 | 351892509 | 351892506 |
| Core\_Cycles | 510692268 | 510692263 | 483981333 | 483981328 | 483981205 | 483981200 |
| S\_Cycles | 404807346 | 404807342 | 392744669 | 392744665 | 392744548 | 392744544 |
| N\_Cycles | 79942559 | 79942558 | 69551806 | 69551805 | 69551793 | 69551792 |
| I\_Cycles | 96349425 | 96349425 | 92091920 | 92091920 | 92091926 | 92091926 |
| C\_Cycles | 0 | 0 | 0 | 0 | 0 | 0 |
| Wait\_States | 449336140 | 29884428 | 366816076 | 273094924 | 366816142 | 273094990 |
| Total | 1030435470 | 879943753 | 921204471 | 827483314 | 921204409 | 827483252 |
| True\_Idles | 38713078 | 38713078 | 38713078 | 38713078 | 38713078 | 38713078 |
| Grand totals | | | | | | |
| Code | 18120 | 18120 | 18424 | 18424 | 18716 | 18716 |
| RO Data | 470 | 470 | 470 | 470 | 470 | 470 |
| RW Data | 0 | 0 | 0 | 0 | 0 | 0 |
| ZI Data | 1632820 | 1632820 | 1632820 | 1632820 | 1632820 | 1632820 |
| Debug | 14344 | 14460 | 14436 | 14444 | 14568 | 14676 |

Μετά στην μικρή cache έχουμε την χειρότερη περίπτωση, όπου δεν χωράνε όλα τα δεδομένα μας και θεωρούμε κακή πολιτική LRU η οποία πιθανόν να οδηγεί σε φαινόμενα τύπου thrashing. Άρα, το κάθε στοιχείο του πινάκα θα πρέπει να φορτωθεί τουλάχιστον μια φορά. Αυτούς τους έξτρα wait states/cycles τους προσομοιώνουμε αφήνοντας τα δεδομένα μας όπως και πριν στην cache, αλλά «φορτώνοντας» τα δεδομένα σε πίνακες temp πάνω στην RAM. Αυτά τα χαμένα wait state/cycles για το «φόρτωμα» των temp προσομοιώνουν τους χαμένους κύκλους που θα χάναμε αν «φορτώναμε» τα δεδομένα μας από την RAM στους καταχωρητές/cache, αλλά και το αντίστροφο αν «φορτώναμε» από τους καταχωρητές/cache στην RAM.

Τέλος, στην 3η περίπτωση (medium cache) χρησιμοποιούμε μια πιο ρεαλιστική προσέγγιση, όπου και πάλι προσομοιώνουμε το «φόρτωμα» των δεδομένων από την RAM στους καταχωρητές/cache, αλλά και το αντίστροφο με την προηγούμενη λογική, όπου «φορτώναμε» τους temp πινάκες που είχαμε στην RAM. Χρησιμοποιούμε την περίπτωση όπου το μέγεθος του block είναι 32/16/8/4 KB, άρα για τον πίνακα image χρειαζόμαστε 13/26/52/104 «φορτώσεις», για το padded image χρειαζόμαστε 14/28/56/112 «φορτώσεις» και για το convout 25/50/100/200, κάθε φορά που τους χρησιμοποιούμε στους ανάλογους αλγορίθμους. Εδώ κάνουμε την θεώρηση ότι από την στιγμή που θα «φορτωθεί» το ανάλογο block θα έχουμε αρκετό μέγεθος στην cache, ώστε να μην χρειαστεί να αντικατασταθεί αλλά ούτε και να «επαναφορτωθεί» (thrashing), αλλά παραμένει μέχρι το τέλος της χρησιμότητας του από τον αλγόριθμο.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3\_8 | 32 block | 16 block | 8 block | 4 block |
| Instructions | 362088401 | 362089321 | 362091161 | 362094841 |
| Core\_Cycles | 513648700 | 513650220 | 513653260 | 513659340 |
| S\_Cycles | 406539249 | 406540119 | 406541859 | 406545339 |
| N\_Cycles | 81167201 | 81167711 | 81168731 | 81170771 |
| I\_Cycles | 96349402 | 96349632 | 96350092 | 96351012 |
| C\_Cycles | 0 | 0 | 0 | 0 |
| Wait\_States | 303742418 | 303744098 | 303747458 | 303754178 |
| Total | 887798270 | 887801560 | 887808140 | 887821300 |
| True\_Idles | 38713078 | 38713078 | 38713078 | 38713078 |
| Grand totals | | | | |
| Code | 18388 | 18388 | 18388 | 18388 |
| RO Data | 470 | 470 | 470 | 470 |
| RW Data | 0 | 0 | 0 | 0 |
| ZI Data | 3265340 | 3265340 | 3265340 | 3265340 |
| Debug | 14736 | 14736 | 14736 | 14736 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 32 block | 16 block | 8 block | 4 block |
| Instructions | 353117991 | 353118927 | 353120795 | 353124531 |
| Core\_Cycles | 486431786 | 486433354 | 486436478 | 486442726 |
| S\_Cycles | 393969990 | 393970877 | 393972645 | 393976181 |
| N\_Cycles | 70776749 | 70777275 | 70778323 | 70780419 |
| I\_Cycles | 92092199 | 92092444 | 92092932 | 92093908 |
| C\_Cycles | 0 | 0 | 0 | 0 |
| Wait\_States | 277992914 | 277994594 | 277997954 | 278004674 |
| Total | 834831852 | 83485190 | 834841854 | 834855182 |
| True\_Idles | 38713078 | 38713078 | 38713078 | 38713078 |
| Grand totals | | | | |
| Code | 18704 | 18704 | 18704 | 18704 |
| RO Data | 470 | 470 | 470 | 470 |
| RW Data | 0 | 0 | 0 | 0 |
| ZI Data | 3265340 | 3265340 | 3265340 | 3265340 |
| Debug | 14736 | 14736 | 14736 | 14736 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 32 block | 16 block | 8 block | 4 block |
| Instructions | 353117835 | 353118769 | 353120637 | 353124373 |
| Core\_Cycles | 486431554 | 486433116 | 486436240 | 486442488 |
| S\_Cycles | 393969815 | 393970699 | 393972467 | 393976003 |
| N\_Cycles | 70776711 | 70777235 | 70778283 | 70780379 |
| I\_Cycles | 92092180 | 92092424 | 92092912 | 92093888 |
| C\_Cycles | 0 | 0 | 0 | 0 |
| Wait\_States | 277992674 | 277994354 | 277997714 | 278004434 |
| Total | 834831380 | 834834712 | 834841376 | 834854704 |
| True\_Idles | 38713078 | 38713078 | 38713078 | 38713078 |
| Grand totals | | | | |
| Code | 18984 | 18984 | 18984 | 18984 |
| RO Data | 470 | 470 | 470 | 470 |
| RW Data | 0 | 0 | 0 | 0 |
| ZI Data | 3265340 | 3265340 | 3265340 | 3265340 |
| Debug | 14960 | 14960 | 14960 | 14960 |

To clock που χρησιμοποιήσαμε είναι 50 MHz. Επίσης δεδομένου του προβλήματος που είχαμε με το scatter χρησιμοποιήσαμε μια μεγάλη cache και μνήμη γενικότερα αλλά στην πραγματικότητα δεν εκμεταλλευτήκαμε όλο το μέγεθος της. Επίσης προσομοιώσαμε την μεταφορά από τη RAM στη cache κάνοντας το ανάποδο δηλαδή μεταφέραμε δεδομένα από τη cache στη RAM.

Πιο συγκεκριμένα τα μεγέθη της μνήμης που χρησιμοποιήσαμε (θεωρητικά) για τα ανάλογα optimization είναι :

Από το make κρίθηκε ότι ήταν απαραίτητο να έχουμε μια μνήμη ROM 0x4AE6 B, για να είμαστε καλυμμένοι για μελλοντική χρήση θα μπορούσαμε να της δώσουμε το διπλάσιοι χώρο από αυτό που απαιτεί το πρόγραμμα. Άρα βάζουμε την μνήμη ROM μας 0x8000.

Στην χειρότερη περίπτωση αν θέλουμε να έχουμε όλους μας τους πινάκες στην RAM ταυτόχρονα, θέλουμε περίπου 1.6 ΜΒ. Άρα, η μνήμη RAM μας θα μπορούσε να έχει μέγεθος 0x7A1200, που αντίστοιχη σε 8 MB.

Τώρα για την μνήμη cache, θα ήταν χρήσιμο να έχει τόσο μέγεθος, ώστε να μπορεί να χωράει ταυτόχρονα την 5x5 μάσκα μας, καθώς και 5 block από την image (int) και 5 block από την convout. Τα παραπάνω μεγέθη αντιστοιχούν σε μια στιγμιαία εφαρμογή την μάσκας μας και υπολογισμού του convolution. Άρα η cache μας θα πρέπει να έχει μέγεθος τουλάχιστον 480 ΚΒ, αφήνοντας και λίγο περιθώριο θεωρούμε ότι η cache μας θα έχει μέγεθος είτε 500 ΚΒ 0x7A120 είτε 1 MB (0xF4240)αν και δεν είναι ρεαλιστικό καθώς το μέγεθος της είναι συγκρίσιμο με την RAM.

Τελικώς, συμφώνα με όλα τα παραπάνω το 5 μας δίνει τα καλύτερα αποτελέσματα σε σύγκριση με τα 3\_8 και 6.