Ερώτηση 1

Δεν έχει απαντηθεί ακόμα

Βαθμολογήθηκε στα 1,00

Επισήμανση ερώτησης

Α.Π.Θ. **Αρχιτεκτονικές Προηγμένων Υπολογιστών** Ι. Παπαευσταθίου

τμήμα Η.Μ.Μ.Υ. **1η εξεταστική περίοδος** 11/2/2021

Σύνολο μονάδων 100. Απαντήστε **σύντομα** και **περιεκτικά όλες** τις ερωτήσεις. Αν χρειαστεί να κάνετε κάποιες υποθέσεις για να προχωρήσετε στην λύση, γράψτε καθαρά **όλες** αυτές τις υποθέσεις. Δείξτε και την διαδικασία της λύσης μαζί με το τελικό αποτέλεσμα. Καλή επιτυχία!

Κάθε 10 λεπτά **υποχρεωτικά** πατήστε **αποθήκευση** και μετά επιστρέψτε στο γράψιμο σας (όχι οριστική υποβολή!)

Ερώτηση 1 : (**25** Μονάδες)

(Α) Δώστε ένα κοινό στοιχείο που έχουν οι VLIW και superscalar επεξεργαστές

(Β) Είναι πιο εύκολο να υλοποιήσουμε σε υλικό (hardware) έναν επεξεργαστή VLIW ή έναν superscalar που να υποστηρίζουν ταυτόχρονα τον ίδιο αριθμός απλών εντολών. Δώστε 2 διαφορετικά μεταξύ τους επιχειρήματα που να δικαιολογούν την απάντηση σας

(Γ) Από τους παραπάνω δύο επεξεργαστές, ανεξάρτητα από το κόστος τους σε υλικό, ποιος θα είναι πιο αποδοτικός ; Δώστε 2 διαφορετικά μεταξύ τους επιχειρήματα που να δικαιολογούν την απάντηση σας

(Δ) Σε ένα πολυεπεξεργαστικό σύστημα με ιδιωτικές (private) L1 caches και μία κοινόχρηστη L2 εάν τα διαμοιραζόμενα δεδομένα διαβάζονται από όλους τους επεξεργαστές/πυρήνες σε κάθε κύκλο ρολογιού αλλά γράφονται μόνο 1 φορά κάθε 1000 κύκλους από έναν και μόνο επεξεργαστή/πυρήνα, πιο αποδοτικό θα είναι ένα σύστημα cache coherency που βασίζεται σε update ή κάποιο που βασίζεται σε invalidate και γιατί ;

(Ε) Τι είδους παραλληλισμό σύμφωνα με την κατάταξη του Flynn εκμεταλλεύονται οι κάρτες γραφικών (GPUs) ; Οι vector επεξεργαστές ; Οι superscalar επεξεργαστές ;

Ερώτηση 2 : (**40** Μονάδες)

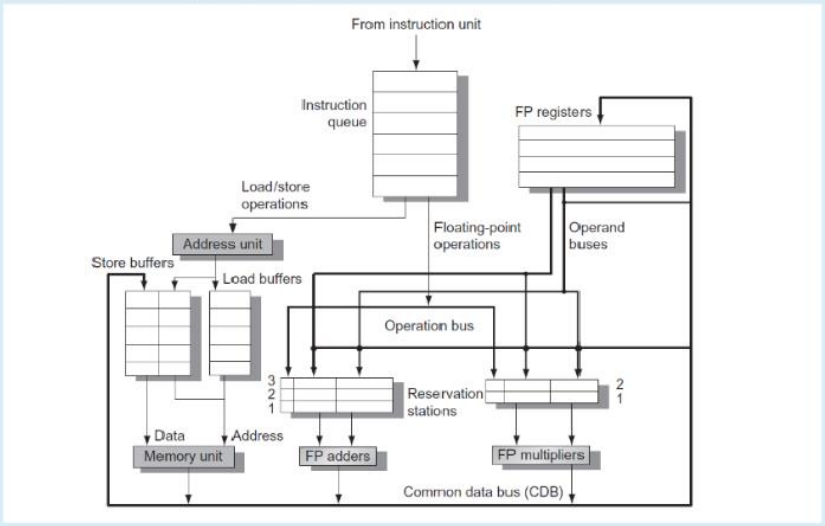
Έστω η υλοποίηση του αλγορίθμου Tomasulo που παρουσιάζεται παρακάτω. Έστω ότι οι καθυστερήσεις των δομικών μονάδων είναι οι εξής: Οι load/store εντολές χρειάζονται 1 κύκλο, οι add/sub (που εκτελούνται στην δομή FP adders) 4 κύκλους, οι multiply 8 κύκλους και οι div 16 κύκλους (και οι δύο εκτελούνται στην δομή FP multipliers).

Χρονοπρογραμματείστε την εκτέλεση δύο επαναλήψεων του παρακάτω loop (χωρις την jump φυσικα) σε έναν επεξεργαστή που χρησιμοποιεί αυτή την δομή Tomasulo αναφέροντας για κάθε εντολής σε ποιον κύκλο κάνει issue, execution complete και Write result.

```
Loop :    LD F2, 100(R1)
          MULTD  F4, F6, F2
          LD F8, 100(R2)
          ADDD   F10, F8, F4
          DIV F12, F10, F8
          LD F14, 100(R3)
          SUB F16, F14, F12
          MULTD  F18, F8, F2
          J loop
```

Σημείωση : μπορείτε από τον editor του elearning να βάλετε έναν πίνακα (δεξί click, εισαγωγή/επεξεργασία πίνακα) που να έχει τόσες γραμμές όσες οι εντολές για 2 επαναλήψεις, και 4 στήλες (εντολή, issue, execution complete, Write result) ώστε να βάλετε εκεί μέσα τους κύκλους που ζητούνται.

Εκτός του πίνακα περιγράψτε περιληπτικά το σκεπτικό σας.



Ερώτηση 2 : (35 Μονάδες) Έστω το παρακάτω κομμάτι κώδικα όπου οι αριθμοί είναι floating point

```
for (i =0 ; i < 100.000.000.000; i++)
{
        c[i] = a[i] * b[i] – d[i] * e[i] + g[i] * f[i];
        f[i] = a[i] * e [i] – d[i] * g[i] - b[i] * f[i];
h [i] = d[i] - e[i]
k[i] =  a[i] / e [i] + d[i] * g[i]
}
```

1) Υπολογίστε πόσος είναι ο μικρότερος δυνατός χρόνος που θεωρητικά θα εκτελεστεί ο παραπάνω κώδικας σε (α) έναν Intel Core i7 920, (β) σε ένα NEC SX-9 και (γ) σε μία Nvidia GTX280 σύμφωνα με τα roofline μοντέλα που σας δίνονται παρακάτω

2) Μετατρέψτε το πρόγραμμα σε assembly χρησιμοποιώντας το παρακάτω vector instruction set

3) Ο NEC SX-9 λειτουργεί στα 3GHz, έχει μέγιστο μήκος διανύσματος (vector length) 32 και convoy 8 εντολών, ενώ υποστηρίζει chaining μεταξύ εντολών πρόσβασης μνήμης και αριθμητικών πράξεων. Όλες οι μονάδες (load/store, πολλαπλασιασμού/διαίρεσης και πρόσθεσης/αφαίρεσης) χρειάζονται αρχικά 4 κύκλους για να «γεμίσουν» και να αρχίσουν την εκτέλεση. Κατόπιν όλα λειτουργούν σε έναν κύκλο ρολογιού (fully pipelined).

Πόσα chimes περιέχει ο κώδικας σας. Με βάση και τις αρχικές καθυστερήσεις που αναφέρονται παραπάνω και έστω ότι υπάρχει μόνο μια μνήμη για τα load και store (άρα κάθε memory access χρειάζεται έναν κύκλο), σε πόσους κύκλους υπολογίζεται η κάθε τετράδα τιμών c,f, h, k ; Σε πόσο χρόνο θα ολοκληρωθεί το πρόγραμμα σας ; Πως συγκρίνεται αυτός ο χρόνος σε σχέση με αυτό που βγάλατε για το roofline model ; Αν υπάρχει κάποια διαφορά που οφείλεται αυτή ;