Assignment 1

Στεφανίδης Ιωάννης

AEM: 9587

1)

Από το αρχείο **starter_se.py** μπορούμε να δούμε τα εξής:

- CPU type: διαλέγεται από ένα array ανάλογα την παράμετρο `--cpu` άρα **MinorCPU**
- Clock speed: **1Ghz**
- **»»βασικες μοναδες??««**
- Caches: με την επιλογή minor cpu έχουμε και L1 cache και L2 cache
- Μνήμη: με την εντολή που τρέχουμε δεν δίνουμε την παράμετρο `--mem-size` άρα παίρνει την default τιμή που είναι **2Gb**
- Components voltage: default->3.3V
- CPU claster voltage: default->1.2V
- CPU freq: default->4Ghz
- Number of cores: default->1

2)

a)

- config.ini:20 -> mem_mode=**timing** -> Memory mode: Timing
- config.ini:44 -> clock=**1000** -> Clock Speed: 1Ghz
- config.ini:58 -> clock=**250** -> CPU freq: 4Ghz
- config.ini:65 -> type=**MinorCPU** -> CPU Type: MinorCPU
- config.ini:113 -> numThreads=**1** -> CPU cores: 1
- config.ini:1339 -> voltage=**1.2** -> CPU claster voltage: 1.2V
- config.ini:1652 -> voltage=**3.3** -> Components voltage: 3.3V

b) Committed instractions: **5149** -> stats.txt:14 `system.cpu_cluster.cpus.committedInsts`

c) Προσπελάσεις L2 **480** -> stats.txt:481 `system.cpu\_cluster.l2.overall\_accesses::total`

3)

In-order CPUs:

MinorCPU: allows visualisation of an instruction's position in the pipeline through the MinorTrace/minorview.py format/tool

TimingSimpleCPU: It stalls on cache accesses and waits for the memory system to respond prior to proceeding.

AtomicSimpleCPU It uses the latency estimates from the atomic accesses to estimate overall cache access time.

Χωρίς παραμέτρους

stats	MinorCPU	TimingSimpleCPU
sim_sec	0.000037	0.000042
committedInsts	10600	10506
numCycles	74022	84878
cpi	6.983208	-

–mem-type=DDR4_2400_8x8

stats	MinorCPU	TimingSimpleCPU
sim_sec	0.000036	0.000042
committedInsts	10600	10506
numCycles	72202	83266
cpi	6.811509	-

–sys-clock=500000000

stats	MinorCPU	TimingSimpleCPU
sim_sec	0.000045	0.000050
committedInsts	10600	10506
numCycles	89510	99630

- Ο **MinorCPU** είναι λογικό να τρέξει πιο γρηγορότερα αφού έχει pipeline ενώ ο **TimingSimpleCPU** δεν έχει και περιμένει την προσπέλαση μνήμης να τελειώσει πριν συνεχίσει.
- Επίσης παρατηρούμε ότι ο αριθμός των κύκλων του **MinorCPU** είναι μικρότερος από αυτόν του **TimingSimpleCPU** και αυτό γιατί με timing memory access χρειάζονται περισσότεροι κύκλοι.
- Με το να αλλάξουμε τον τύπο της μνήμης από DDR3_1600_8x8 σε DDR4_2400_8x8 βλέπουμε μια μικρή βελτίωση στον χρόνο εκτέλεσης αλλά και μείωση στους κύκλους που απαιτούνται.
- Αλλάζοντας την συχνότητα από 1Ghz σε 0.5Ghz παρατηρούμε ότι οι χρόνοι εκτέλεσης αυξήθηκαν όπως και οι κύκλοι που απαιτούνται.

Assignment Review

Η εργασία πιστεύω είναι καλή ευκαιρία να έρθει ο μαθητής σε επαφή με πολλά χρήσιμα εργαλεία όπως το terminal, git, markdown. Εγώ είχα εμπειρία με linux και δεν δυσκολεύτηκα ιδιαίτερα να κάνω το setup για να μπορώ να τρέξω simulations με το gem5, για έναν νέο χρήστη όμως ένα μικρό λάθος να γίνει και να του πετάξει error μπορεί να τον κάνει να τα παρατήσει. Το error όμως μπορεί να μην έχει σχέση με την εργασία αλλά με το λειτουργικό του ή με το λάθος setup του VM, που αν είναι νέος σε αυτά δεν θα καταλαβαίνει και τι του λέει το error.

Ένα λάθος που εντόπισα στην εκφώνηση είναι στο 3a όπου το se.py δεν παίρνει positional arguments και χρειάζεται -c πριν το πρόγραμμα που είναι να τρέξει.

Το πρόβλημα που αντιμετώπισα εγώ ήταν ότι επειδή δεν έτρεχα τα outdated Ubuntu αλλά Arch δεν είχα την δυνατότητα να τρέξω την εντολή `gcc-arm-linux-gnueabihf`. Αυτό που έκανα είναι να κατεβάσω το `docker` και μέσω του dockcross να κάνω compile για armv7. Έτσι είδα ότι είχε γίνει compile το hello test, αλλά πιστεύω είναι και πιο δύσκολος τρόπος για να το βάλετε στο pdf.

Suggestion

Να κάνετε setup ένα VM που να έχει ότι χρειάζεται ο χρήστης και να ανεβάσετε κάπου το image του. Το manual setup πιστεύω είναι καλύτερο.. μπορούν να υπάρχουν και οι 2 τρόποι στο pdf την εργασίας.