

STEP 3-LAB 2

Το κόστος της υλοποίησης μιας μνήμης εξαρτάται από τους εξής παράγοντες(σύμφωνα με την εκφώνηση):

L1 instruction cache size

L1 instruction cache associativity

L1 data cache size

L1 data cache associativity

L2 cache size

L2 cache associativity

Μέγεθος cache line

Τι γνωρίζουμε(είτε από βιβλιογραφία είτε από τα δεδομένα);

α)Γνωρίζουμε από την βιβλιογραφία ότι η L1 instruction cache size είναι 2 φορές πιο γρήγορα από την L1 data cache size.

β)Επίσης γνωρίζουμε ότι η L1 instruction cache είναι 3-5 φορές γρηγορότερη από την L2 cache.

γ)Επίσης, ξέρουμε ότι όσο πιο μεγάλο το assoc τόσο πιο μεγάλη η πολυπλοκότητα, άρα τόσες περισσότερες λογικές πύλες.

δ)Επίσης γνωρίζουμε ότι το cache line size χρειάζεται πολλούς πολυπλέκτες και συνδέσεις με L2 και RAM μνήμες.

-ΠΑΡΑΔΟΧΕΣ:

κάνουμε τις εξής παραδοχές σχετικά με το κόστος.

α) $\text{κοστος} = \sqrt{\text{μεγεθος μνήμης}} * \text{μοναδα κοστους}$.

Δηλαδή αν μια μνήμη είναι 3 φορές πιο γρήγορα τότε είναι 9 φορές πιο ακριβή.

β) Η σχέση μεταξύ του associativity και του κόστους είναι γραμμική. Δηλαδή αν είναι για $\text{assoc}=1$ 40 μονάδες κόστους για $\text{assoc}=2$ έχουμε 80 μονάδες κόστους.

Οπότε με αυτά τα δεδομένα και αυτές τις παραδοχές μπορούμε να βγάλουμε μία συνάρτηση κόστους.

MK = η αυθαίρετη μονάδα κόστους.

$$\text{κόστος} = \text{L1Ins_cache} * \text{MK}^4 + \text{L1Dat} * \text{MK}^3 + \text{L2} * \text{MK} + \text{L1Ins_Assoc} * \text{MK} + \text{L1Dat_Assoc} * \text{MK} + \text{L2} * \text{Assoc} + \text{Cache_Lsize} * \text{MK}.$$

Προφανώς για την instruction cache χρειάζεται να πληρώσουμε το παραπάνω.

Για το specmark specbzip:

Παρατηρούμε από το αρχείο results.txt ότι το καλύτερο CPI είναι όταν έχουμε

L1ins = 32kB assoc=2, L1dat = 64kB assoc=2,
L2 = 2MB assoc=8, cache_line_size=128

Επομένως το κόστος σύμφωνα με την συνάρτηση κόστους είναι(για MK=2):

ΚΟΣΤΟΣ = $32*16+64*8+400+2*2+2*2+2*8+128*2 = 1704$ MK. Συνολικά το L1 είναι περίπου 3 φορές πιο ακριβό από το L2 στην προκειμένη περίπτωση.

Για το specmark specmcf:

Παρατηρούμε ότι έχουμε το καλύτερο CPI σε τρεις περιπτώσεις:

A)Για L1inst = 64KB assoc=2, L1dat=128KB assoc=2,
L2=2MB assoc=8 cache_line_size=64

ΚΟΣΤΟΣ =

$64*16+128*8+400+2*2+2*2+2*8+64*2=2600$ MK

B)L1INS =128KB assoc=2,L1DATA=64kB assoc=2,
L2=2MB assoc=8, cache_line_size=64

ΚΟΣΤΟΣ=3624MK

Γ)L1INS=32KB assoc=4, L1D = 64KB ASSOC=2

L2=2MB ASSOC=8, CACHE_LINE_SIZE=64

ΚΟΣΤΟΣ=2604MK

Η ΚΑΛΥΤΕΡΗ ΕΠΙΛΟΓΗ ΕΙΝΑΙ Η Α)

Για το specmark spechmmer:

Η καλύτερη επιλογή είναι σαν του specbzip

L1ins = 32kB assoc=2, L1dat = 64kB assoc=2,

L2 = 2MB assoc=8, cache_line_size=128

με κόστος ΚΟΣΤΟΣ= 1704MK

Για το specmark specsjeng:

Πάλι η καλύτερη επιλογή είναι σαν του specbzip ή του spechmmer

L1ins = 32kB assoc=2, L1dat = 64kB assoc=2,

L2 = 2MB assoc=8, cache_line_size=128

Με κόστος ΚΟΣΤΟΣ=1704MK

Για το specmark speclibm:

Επίσης η καλύτερη επιλογή είναι το

L1ins = 32kB assoc=2, L1dat = 64kB assoc=2,

L2 = 2MB assoc=8, cache_line_size=128

με κοστος ΚΟΣΤΟΣ=1704MK.

Παρατηρούμε ότι σε όλα τα προγράμματα εκτός από 1 το cache_line_size επηρεάζει αρκετά το CPI του προγράμματος, και μάλιστα το κάνει αρκετά καλύτερο. Επίσης επηρεάζει πολύ θετικά και τα miss rates.

Μάλιστα ίσως το cache line size να επηρεάζει και πολύ περισσότερο την τιμή κόστους από την παραδοχή την δική μας. Ίσως δηλαδή η συνάρτηση κόστους να είναι τελείως διαφορετική λόγω του cache line size, να αυξάνεται πχ εκθετικά όταν αυξάνεται το cache line size.

Αν ισχύει κάτι τέτοιο τότε σίγουρα δεν θα είναι η βέλτιστη λύση οικονομικά, αν δηλαδή το cache line size είναι πολύ ακριβό.

Επόμενως μπορούμε να καταφύγουμε στην δεύτερη λύση που μπορεί να μην είναι τόσο γρήγορη όσο όταν το cache_line_size=128, αλλά σίγουρα είναι οικονομικότερη και έχει καλό cpi.

Αυτή η λύση είναι οι εξής:

```
L1i-64kB-assoc=2      L1d-128kB-assoc=2  
L2-2MB-assoc8.
```

Παρατηρούμε ότι για όλα τα specmarks αυτή η λύση στέκεται τίμια. Έχει καλό CPI, και μάλιστα για το specmark specmcf έχει και καλύτερη από τι αν cache line size=128.