

Rohan Jhaveri
830962238

ECE 452

04/23/2017

CQA Assignment - 5

Q1 a.)

1 word = 4 bytes = $4 \times 8 = 32$ bits

$\therefore 16$ bytes = 4 integers
4 bytes

\therefore 4 integers of 32-bits could be stored in a 16-byte cache block

b.) In the given code I P T are accessed continuously. Hence I P T exhibit temporal locality

c.) Since in this case $A[I] \circledcirc [J]$ are itself accessed continuously, hence $A[I] \circledcirc [J]$ also ~~performs~~ exhibits spatial locality.

d) Total number of words =

$$(8 \times 8000) + (8 \times 8000) + 8 - 8 \times 8$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$A[I, J]$ $A[J, I]$ $B[I, 0]$ (common between
 $A[I, J]$ $B[I, I]$)

= 127944 words

= 511776 bytes

~~511776 bytes~~ ~~16 bytes cache block~~
can store ~~511776~~ = 31986

∴ For 511776 bytes, ~~51176~~ = 31986

~~cache~~ cache blocks of 16 bytes
are required.

e) Again I & J are accessed continuously hence it exhibits temporal locality.

f) In MATLAB, since columns are accessed first $A[J, I]$ exhibits spatial locality.

Q2 a) Tag size = 2^2 bits, Index = 5 bits

Address = 32 bits

block size = 2^m words

Also, it requires a 2 bit for

~~32 = 22 + 5~~ byte part

\therefore Tag size = Address - (Index + word part
+ byte part)

$$\therefore 2^2 = 32 - (5 + m + 2)$$

$$\therefore m = 3$$

$$\therefore \text{Block size} = 2^3 = 8 \text{ words/}$$

b) No. of entries = 2^{index}

Address bits used = $2^5 = 32$ entries

c) Total bits for cache =

$$\begin{aligned} & 2^{\text{index}} \times (\text{Block size} \times 32 + 32 - \text{index} - \\ & 5) \\ & = 2^5 \times ((2^3 \times 32 + 32 - 5 - 3) + 1) \\ & = 32 \times (8 \times 32 + 23) \\ & = 279 \times 32 = 8928 \end{aligned}$$

Data storage bits = $2^{\text{index}} \times (\text{Block size} \times 32)$

$$= 2^5 \times (8 \times 32)$$

$$= 256 \times 32 = 8192$$

Total bits for cache = $8928 = 1.089$

Data storage bits = 8192

Key word = 85 = size of block.

| | | | | | | | | | | | | |
|---------|---|---|----|-----|-----|-----|------|----|-----|------|-----|------|
| Address | 0 | 4 | 16 | 132 | 232 | 160 | 1024 | 30 | 140 | 3100 | 180 | 2180 |
| Tag | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 0 | 2 |
| Index | 0 | 0 | 0 | 4 | 7 | 5 | 0 | 0 | 4 | 0 | 5 | 4 |
| Offset | 0 | 4 | 16 | 4 | 8 | 0 | 0 | 30 | 12 | 28 | 20 | 4 |

This is obtained by writing the address in the format as shown below

| Tag | Index | Offset |
|-------|-------|--------|
| 31-10 | 9-5 | 4-0 |

blanks = 8 points when set :

Now,

| | | | | | | | | | | | | |
|----------|------|-----|------|------|------|------|---------|---------|---------|---------|---------|------|
| Address | 0 | 4 | 16 | 132 | 232 | 160 | 1024 | 30 | 140 | 3100 | 180 | 2180 |
| Tag | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 0 | 2 |
| Index | 0 | 0 | 0 | 4 | 7 | 5 | 0 | 0 | 4 | 0 | 5 | 4 |
| Offset | 0 | 4 | 16 | 4 | 8 | 0 | 0 | 30 | 12 | 28 | 20 | 4 |
| Hit/Miss | Miss | Hit | Miss | Miss | Miss | Miss | Miss | Hit | Miss | Hit | Miss | |
| Replace | | | | | | | Replace | Replace | Replace | Replace | Replace | |

d) 4 blocks are replaced.

e) The hit ratio is $\frac{4}{12} = \frac{1}{3}$

f) The final state of the cache would be the latest index & tag access and its data.

The cache entries would be

$\langle \text{Index, tag, data} \rangle = \langle 7, 0, \text{Mem}[224-232] \rangle,$
 $\langle 5, 0, \text{Mem}[160-180] \rangle, \langle 0, 3, \text{Mem}[3072-3100] \rangle,$
 $\langle 4, 2, \text{Mem}[2176-2180] \rangle$

Q3 a) Clock rate for $P_1 = \frac{1}{\text{hit time for } P_1}$

hit time for P_1

$$= \frac{1}{0.66 \text{ ns}} = 1.5156 \text{ Hz}$$

Clock rate for $P_2 = \frac{1}{\text{hit time for } P_2}$

hit time for P_2

$$= \frac{1}{0.90 \text{ ns}} = 1.1111 \text{ Hz}$$

b.) $\text{AMAT}(P_1) = \text{Hit time} + \text{Miss Rate} \times \text{Miss time}$

$$= 0.66 \text{ ns} + 0.08 \times 70 \text{ ns}$$
$$= 6.26 \text{ ns}$$

$\text{AMAT}(P_2) = \text{Hit time} + \text{Miss Rate} \times \text{Miss time}$

$$= 0.90 \text{ ns} + 0.06 \times 70 \text{ ns}$$
$$= 51 \text{ ns}$$

$$c.) P1 \frac{\text{Total cycles}}{\text{miss}} = \frac{(0.36 \times 0.08 \times 70) \times IC}{0.66} = 3.054 IC$$

miss

19.2 ns/inst

$$\text{Total CPI for P1} = 3.054 IC$$

$$CPI = \frac{3.054 IC}{IC} = 3.054$$

$$\text{Total CPI for P1} = 1 + 3.054$$

$$= 4.054$$

$$P2 \text{ miss cycles} = \frac{(0.36 \times 0.06 \times 70) \times IC}{0.66} = 1.68 IC$$

$$CPI = \frac{1.68 IC}{IC} = 1.68$$

$$\text{Total CPI for P2} = 1 + 1.68$$

$$= 2.68$$

d) New AMAT = (L1 Hit Rate + L1 miss Rate ×
 (L2 Hit time + L2 miss rate
 × Mem Access time))

$$\text{AMAT} = 0.66 \times 0.08 \times (5.62 \text{ ns} + 0.95 \times 7.9 \text{ ns})$$

$$\therefore \text{AMAT} = 6.1429 \text{ ns}$$

\therefore AMAT is worse with L2 cache

e) PI miss time with L2 cache = $0.36 \times 0.08 \times (5.62 + 0.95 \times \frac{70}{0.66})$

$$= (0.36 \times 0.08 \times 5.62) + (0.36 \times 0.08 \times 0.95 \times \frac{70}{0.66})$$

$$= 3.06 \text{ IC}$$

~~$\therefore \text{CPI} = \frac{3.06 \text{ IC}}{\text{IC}} = 3.06$~~

~~$\therefore \text{Total CPI with L2 cache} = 1 + 3.06$~~

~~$= 4.06$~~

~~and hence 19.76 clock cycles off~~
 ~~$\therefore \text{CPI} \geq 21.90 \text{ is bad}$~~

f) We don't know C.P.I. for P1 is 4.06 & for P2 is

P1 is 4.06 & for P2 is 2.68

P2 is faster than P1

To make P1 as fast as P2 miss rate should be manipulated

$$CPI_{P_1} = CPI_{P_2}$$
$$0.60 + \text{Miss Rate}_{P_1} \times (5.62 + 0.95 \times 70) = 0.9 + 0.06 \times 70$$

$$\text{Miss Rate}_{P_1} = \frac{0.9 + 0.06 \times 70 - 0.66}{5.62 + 0.95 \times 70}$$

Hence, to make P1 as fast as P2 the miss rate of P1 should be reduced $0.0615 = 6.15\%$

LRU Policy

| Q 4(a) Address of memory Block Accessed | Hit or miss | Evicted Block | Contents of Cache Block After Reference | | | |
|--|-------------|------------------|--|---------|---------|---------|
| | | | Set 0 | Set 0 | Set 1 | Set 1 |
| 0 | miss | | Mem[0] | | | |
| 2 | miss | | Mem[0] | | | |
| 4 | miss | | Mem[0] | Mem[4] | Mem[2] | |
| 8 | miss | | Mem[0] | Mem[4] | Mem[2] | Mem[8] |
| 10 | miss | Mem[0] | Mem[10] | Mem[4] | Mem[2] | Mem[8] |
| 12 | miss | Mem[2] | Mem[10] | Mem[4] | Mem[2] | Mem[8] |
| 14 | miss | Mem[4] | Mem[10] | Mem[4] | Mem[2] | Mem[8] |
| 16 | miss | Mem[8] | Mem[10] | Mem[4] | Mem[2] | Mem[8] |
| 18 | miss | Mem[10] | Mem[0] | Mem[14] | Mem[12] | Mem[16] |

~~0~~ miss, ~~evict~~ cache = ~~full~~ (3)

MRU policy

Contents of Cache Block

| b.) | Address | Hit or miss | Evicted Block | Set 0 | Set 1 | Set 2 | Set 3 |
|-----|---------|-------------|---------------|--------|--------|--------|---------|
| | 0 | miss | | Mem[0] | | | |
| | 2 | miss | | Mem[0] | Mem[2] | | |
| | 4 | miss | | Mem[0] | Mem[4] | Mem[2] | |
| | 8 | miss | | Mem[0] | Mem[4] | Mem[2] | Mem[8] |
| | 10 | miss | Mem[8] | Mem[0] | Mem[4] | Mem[2] | Mem[10] |
| | 12 | miss | Mem[10] | Mem[0] | Mem[4] | Mem[2] | Mem[12] |
| | 14 | miss | Mem[12] | Mem[0] | Mem[4] | Mem[2] | Mem[14] |
| | 16 | miss | Mem[14] | Mem[0] | Mem[4] | Mem[2] | Mem[16] |
| | 0 | hit | | Mem[0] | Mem[4] | Mem[2] | Mem[16] |
| | 2 | hit | | | | | |

c.) Using random replacement policy

there can be 1 hit or 0 hit,
no but not more than 1 hit.

1 hit is possible if we have all fails in which we will get a hit when 0 is accessed second time.

d) MRU should be used as it gives evicts the most recently used address to get hit for best access. It's a hit is possible if we follow this optimal policy.

e) Dynamic replacement policy would be optimal as the hardware would decide which address to evict. It is difficult to implement this as it is sometimes not possible to predict it from prior which address should be evicted. Hence, sometimes it is better to have some replacement policy and optimize it, to maximize hit rate.

f.) As I would know whether or not to cache a particular address, I would only cache if it will be used frequently otherwise not. In this case, there will be space in cache for frequently used data and hence miss rate would be minimized.

Q5 a) Page Size = $4 \text{ KiB} = 2^{12}$

Page table entries needed = $43 - 12 = 31$
∴ 31 page table entries are needed.

Required Physical memory = $2^{31} \times 4 \text{ bytes}$
 $= 2^{33} \text{ bytes}$
 $= 8 \text{ GiB}$

b) No. of PTE's = $2^{32} = 2^{19}$

Size of Page Table = $5 \times 2^{19} \times 4$
 $= 20 \times 2^{19}$

But only half of the available memory can be used.

∴ Size of Page Table = $20 \times 2^{19} \times 1$
 $= 10 \times 2^{19} = 5 \times 2^{20}$
 $= 5 \text{ MiB}$

c) The valid bit is set to 0 or 1 when the page is present on the disk and not on TLB cache or RAM.

Hence, in this case when entry 2's page would be replaced and paged out to disk, is when its valid bit be set to zero.

d) There will be a miss in the TLB. So, it will check whether Page Table has the Physical address for the given Virtual address. If there is, then it will update the TLB. If a software can fetch the TLB entries from prior, then in software managed-TLB it would be faster than hardware-managed TLB.

e.) Since page 200 is read-only, so there will be an interrupt when an instruction writes to virtual address page 200