

Name: HARSH VORA

Roll No: 2021CS10548

(COL 216) Computer Architecture

May 1, 2023

Major Exam

Duration: 120 minutes

(60 marks)

**Beware:** Be concise in your writing. You can use rough sheets for calculations. But you cannot submit any additional sheet for grading on Gradescope. So make sure you are certain when you write something (after rough work, or use a dark pencil). If you cheat, you will surely get an F in this course.

1. Consider a processor with a 16 Kbyte unified L1 cache. The miss rate for this cache is 3% and the hit time is 2 clock cycles. The processor also has an 8 Mbyte, on-chip L2 cache. 95% of the time, data requests to the L2 cache are found. If data is not found in the L2 cache, a request is made to a 4 Gbyte main memory. The time to service a memory request is 100,000 clock cycles. On average, it takes 3.5 clock cycles to process a memory request. How often is data found only in main memory, and not in either of the two caches? [3 marks]

L1 16 KB, 2cc, 3% | 8MB L2 5%

L1 — miss 3% → L2 — miss 5%  $\frac{3}{100} \times \frac{5}{100} = 0.15\%$   
Data found in main memory 0.15% times ANS

In 10,000 instr, 15 instr require main memory  
⇒ every  $\frac{10,000 \times 3.5}{15}$  c.c. =  $\frac{35000}{15} = 2333.3$  c.c. =

every 2333.3 clock cycles, data is needed from mem.

$$2 + \frac{3}{100} \times 2 + \frac{15}{10000} \times 100000 = 3.5$$

Name: HARSH VORA

Roll No: 2021CS10548

2. Each instruction fetch means a reference to the instruction cache and 35% of all instructions reference data memory. Processor A has two 8 Kbyte, L1 caches - one for data and one for instructions. Computer B has a single, unified 16 Kbyte L1 cache that holds both instructions and data. For A, the average miss rate in the L1 instruction cache is 2%, the average miss rate in the L1 data cache is 10%, and the miss penalty for both data and instruction caches is 9 clock cycles. For B, the average miss rate is 3% for the cache as a whole, and the miss penalty is again 9 clock cycles. Which processor has better performance? [3 marks]

A: 8KB : inst 2%  
8KB data 10%

100 inst  $\rightarrow$  2% = 2 inst get inst miss

$\hookrightarrow$  35 mem.  $\rightarrow$  10% = 3.5 inst - mem miss

5.5 inst miss average. per 100

miss rate  $\frac{5.5}{100}$

B: 16KB, 3%

100  $\rightarrow$  3% = 3 inst miss

$\hookrightarrow$  35  $\rightarrow$  3% = 1.05 inst

4.05 miss rate -  
 $\frac{4.05}{100}$

(B) less miss rate - BETTER PERFORMANCE.

Name: HARSH VORA

Roll No: 2021CS10548

3. The following table gives the parameters for a number of different caches. Your task is to fill in the missing fields in the table. Recall that  $m$  is the number of physical address bits,  $C$  is the cache size (number of data bytes),  $B$  is the block size in bytes,  $E$  is the associativity,  $S$  is the number of cache sets,  $t$  is the number of tag bits,  $s$  is the number of set index bits, and  $b$  is the number of block offset bits. [4 marks]

Cache	$m$	$C$	$B$	$E$	$S$	$t$	$s$	$b$
1.	32	<u>2048</u>	8	1	<u><math>2^8 = 256</math></u>	21	8	3
2.	32	2,048	<u>4</u>	<u>4</u>	128	23	7	2
3.	32	1,024	2	8	64	<u>25</u>	<u>6</u>	1
4.	32	1024	<u>32</u>	2	16	23	4	<u>5</u>

Table 1: Cache organization

$$C = B \times E \times S$$

$$s = \log_2 S \quad \left| \quad S = 2^s$$

$$b = \log_2 B \quad \left| \quad B = 2^b$$

$$t = 32 - (s + b)$$

$1) \quad 2^8 = 256$ $\begin{array}{r} 256 \\ \underline{8} \\ 2048 \end{array}$	$3) \quad \begin{array}{r} 32 \\ \underline{-7} \\ 25 \end{array}$	$4) \quad \begin{array}{r} 128 \quad 32 \\ \underline{1024} \\ 304 \end{array}$
$2) \quad \begin{array}{r} 2048 \times 4 \\ \underline{11 \times 128} \end{array}$		

4. A dynamic RAM has a memory cycle time of 64 nsec. It has to be refreshed 100 times per msec and each refresh takes 100 nsec. What percentage of the memory cycle time is used for refreshing? [3 marks]

$$64 \times 10^{-9} \text{ s}, \quad \text{to } \frac{100 \text{ times/sec}}{10^5 \text{ times/sec}}, \quad 10^{-7} \text{ sec}$$

$$\text{refresh: } 10^5 \text{ times/sec} \times 10^{-7} \text{ sec} \rightarrow 10^{-2} \text{ sec per sec}$$

$$10^{-2} \times 64 \text{ nsec} \quad \text{used in refreshing}$$

$$\frac{10^{-2} \times 64}{64} \times 100 = 10^{-6} \% \text{ used}$$

$$\left[ \frac{10^{-2} \times 64 \text{ nsec}}{64 \text{ nsec}} \right] \times 100 = \boxed{1\%}$$

Ans = 1% of cycle time is used for

refreshing.

Name: HARSH VORA

Roll No: 2021CS10548

5. Consider a symmetric shared-memory multiprocessor (3 processors sharing a bus) implementing a snooping cache coherence protocol (MSI). For each of the events below, explain the coherence protocol steps (does the cache flag a hit/miss, what request is placed on the bus, who responds, is a writeback required, etc.) and mention the eventual state of the data block in the caches of each of the 3 processors. Assume that X and Y are not in any of the caches at the start of the sequence, the caches are direct-mapped, and blocks X and Y map to the same set in each cache (X and Y cannot co-exist in a cache at any time). [7 marks]

Request	Cache hit/miss	Request on bus	Who responds/ Write Back happens?	Cache 1 state	Cache 2 state	Cache 3 state
P1: Write X	miss	P1: write miss X, inv all X.	no wrt b k mem responds with X.	M(X)	I(X)	I(X)
P2: Write X	miss	P2: write miss X, inv all X. (invalidate)	P1 may send modified value to P2. P1(X)M → I no wrt b k	I	M(X)	I
P3: Read X	miss	P3: read miss X.	P2 responds with modified value. wrt b k to mem: P2 X	I	S(X)	S(X)
P1: Read X	miss	P1: read miss X.	mem responds with X. P2, P3 may respond no wrt b k	S(X)	S(X)	S(X)
P3: Write X	hit	P3: write hit X. invalidate all X.	all other process invalidate X. no wrt b k.	I(X)	I(X)	M(X)
P3: Read Y	miss	P3: read miss Y.	mem responds with Y. wrt b k X to mem.	I(X)	I(X)	S(Y)
P2: Write Y	miss	P2: write miss Y invalidate all Y: P2.	mem responds with Y value. no wrt b k	I(X)	M(Y)	I(Y)

Table 2: Snoop based Cache Coherence Table

Based on lecture slides where  $M(P1) \rightarrow M(P2) \Rightarrow$  no wrt b k.

Name: HARSH VORA

Roll No: 2021CS10548

6. You are given the following code to analyze:

```

1 int x[2][128];
2 int i; int sum = 0;
3
4 for (i = 0; i < 128; i++) {
5     sum += x[0][i] * x[1][i];
6 }
    
```

Assume we execute this under the following conditions: (a)  $\text{sizeof(int)} = 4$ . (b) Array  $x$  begins at memory address  $0x0$  and is stored in row-major order. (c) In each case below, the cache is initially empty. (d) The only memory accesses are to the entries of the array  $x$ . All other variables are stored in registers. Given these assumptions, estimate the miss rates for the following cases: [10 marks]

- A. Case 1: Assume the cache is 512 bytes, direct-mapped, with 16-byte cache blocks. What is the miss rate?
- B. Case 2: What is the miss rate if we double the cache size to 1,024 bytes?
- C. Case 3: Now assume the cache is 512 bytes, two-way set associative using an LRU replacement policy, with 16-byte cache blocks. What is the cache miss rate?
- D. For case 3, will a larger cache size help to reduce the miss rate? Why or why not?
- E. For case 3, will a larger block size help to reduce the miss rate? Why or why not?

size of array =  
 $2 \times 128 \times 4 \text{ bytes} = 1024 \text{ bytes}$   
 $= 256 \text{ words.}$   
 mem space b/w  $x[0][i]$  &  $x[1][i]$   
 $= 128 \times 4 \text{ bytes} = 512 \text{ bytes} = 128 \text{ words}$

A)  $x[0][0]$  stored in set 0  
 $x[1][0]$  in set  $\frac{512}{16} \bmod (512/16) \equiv 0$   
 $\Rightarrow x[0][i]$  is always overwritten by  $x[1][i] \Rightarrow$  ALL MISSES  
 miss rate 100%

B) Direct map 1024 bytes 16 byte sets  
 $\Rightarrow 64 \text{ sets.}$   
 if  $x[0][i]$  is mapped to set  $s$   
 $x[1][i]$  is mapped to  $\frac{512}{16} \bmod \frac{1024}{16} + s$   
 $= \text{set}(32+s)$   
 $\Rightarrow$  no overlap. 1 line stores 4 words  
 2 miss then 6 hits always.  
 $\rightarrow (\frac{1}{4}) = \text{miss rate} \rightarrow 25\%$

C)  $512 / (16 \times 2) = 16 \text{ sets.}$   
 $x[0][i]$  is stored in set  $s$ .  
 $\rightarrow x[1][i]$  in set  $(s + (\frac{512}{16}) \bmod 16) \equiv \text{set } s$ .  
 But it's stored in 2nd way.  
 $x[0][i], x[1][i]$  miss  $\rightarrow$  2 miss  
 then  $x[0][i+1], x[1][i+1]$  }  $\leftarrow$  6 hits  
 $\begin{matrix} i+2 & i+2 \\ i+3 & i+3 \end{matrix}$   
 miss rate =  $\frac{2}{8} = \frac{1}{4} = 25\%$

D) NO,  $\because$  there is no conflict miss until the whole line is used, call data in line used from cache and there is no capacity miss until whole line is used. one miss that happens in every 4 iterations = compulsory miss due to block size.

E) YES block stores  $x$  words.  
 1st compulsory miss to get block, remaining  $(x-1)$  words fetched, not replaced until completely used  $\Rightarrow$  miss rate =  $\frac{1}{x}$ .  
 $x$  increases (block size  $\uparrow$ ) miss rate  $\downarrow$ .

Name: HARSH VORA

Roll No: 2021CS10548

Name: HARSH VORA

Roll No: 2021CS10548

7. You are writing a new 3D game. You are currently working on a function to blank the screen buffer before drawing the next frame. The screen you are working with is a  $640 \times 480$  array of pixels. The machine you are working on has a 32 KB direct-mapped cache with 8-byte lines. The C structures you are using are as follows:

```

1 struct pixel{
2     char r;
3     char g;
4     char b;
5     char a;
6 };
7
8 struct pixel buffer[480][640];
9 int i, j;
10 char *cptr;
11 int *iptr;
12 }
    
```

Assume the following: (a)  $\text{sizeof(char)} = 1$  and  $\text{sizeof(int)} = 4$  (b) buffer begins at memory address 0. The cache is initially empty. (c) The only memory accesses are to the entries of the array buffer. Variables i, j, cptr, and iptr are stored in registers.

(A) What percentage of writes in the following code will hit in the cache? [4 marks]

```

1 for (j = 639; j >= 0; j--) {
2     for (i = 479; i >= 0; i--) {
3         buffer[i][j].r = 0;
4         buffer[i][j].g = 0;
5         buffer[i][j].b = 0;
6         buffer[i][j].a = 0;
7     }
8 }
    
```

$2^2$   
 $2^5 \times 3 \times 5$   
 $2^7 \times 5$

(B) What percentage of writes in the following code will hit in the cache? [3 marks]

```

1 char *cptr = (char *) buffer;
2 for (; cptr < (((char *) buffer) + 640 * 480 * 4); cptr++)
3     *cptr = 0;
    
```

(C) What percentage of writes in the following code will hit in the cache? [3 marks]

```

1 int *iptr = (int *)buffer;
2 for (; iptr < ((int *)buffer + 640*480); iptr++)
3     *iptr = 0;
4 }
    
```

A) pixel struct size = 4 bytes.

cache: 32KB, 8 Byte lines  
~~lines~~  $\frac{32 \times 1024}{8} = 4096$  lines

(size =  $2^{15}$  KB)

size of matrix  
 $= 4 \times 480 \times 640$   
 $= 2^{14} \times 3 \times 2^5 \text{ B} > 2^{15} \text{ B.}$

when at  $\text{buffer}[i][j].r$ , whole word is called into cache  $\Rightarrow$  g, b, a will all hit.  $\rightarrow$  3 hits every 4 lines writes

$\text{buffer}[i][j] \rightarrow \text{buffer}[i+1][j]$  - gap =  $640 \times 4$  bytes =  $\frac{640 \times 4}{8} \pmod{4096}$  lines

set s  $\rightarrow$  set (s+320)

$\text{buffer}[i+k][j] \rightarrow$  set  $s + \underbrace{(320k) \% 4096}$  this = 0 when  $k = 64$ .

i goes from 48 to 479  $\rightarrow$  0.  
 $\rightarrow$   $\text{buffer}[i][j]$  line overwritten by  $\text{buffer}[i-64][j]$  line.  
 $\rightarrow$  new miss when accessing  $\text{buffer}[i][j+1]$ . later.  
 miss

so only 3 hits in 4 writes  $\rightarrow$   $\frac{75\%}{\text{hit}}$

Name: HARSH VORA

Roll No: 2021CS10548

B) cache line has 8 Bytes.

• (cptr)  $\rightarrow$  1 Byte, cptr++  $\rightarrow$  address increases by 1 Byte.  
(stride 1 access)

when initially cptr is written, (miss),  
line brought into cache.

next 7 writes in same line  $\rightarrow$  hits

then write in next line -

$$\frac{7}{8} \text{ write hits} \rightarrow \frac{700}{8} = \boxed{87.5\% \text{ write hits}}$$

C) stride 1 access  $\rightarrow$  2 ints per line

address  $\uparrow$  by 4 every int iptr++.

1 miss (initial int of each line) }  
2<sup>nd</sup> hit (2<sup>nd</sup> int of that line) }

$$\boxed{50\% \text{ write hits}}$$

then new line accessed (miss).

Name: HARSH VORA

Roll No: 2021CS10548

8. Choose the correct answer or write short answers to the following questions. [20 marks]

(a) Consider the IEEE-754 single precision floating point numbers  $P = 0xC1800000$  and  $Q = 0x3F5C2EF4$ . Which one of the following corresponds to the product of these numbers represented in the IEEE-754 single precision format? [3 marks]

(a)  $0x404C2EF4$

(b)  $0x405C2EF4$

(c)  $0xC15C2EF4$

(d)  $0xC14C2EF4$

$$\begin{aligned}
 P &= \underline{1} \ \underline{10000011} \ \underline{00000000} = -2^4 \times 1 \\
 Q &= \underline{0} \ \underline{0111110101} = +2^{-1} \times \text{frac} \\
 \text{prod} &= -2^{+3} \times \text{frac} \\
 &= \underline{0} \ \underline{1000001010101} \\
 &= 0xC15C2EF4
 \end{aligned}$$

(b) Consider a 3-stage pipelined processor having a delay of 10 nanosecs, 20 nanosecs, and 14 nanosecs for the first, second, and the third stages, respectively. Assume that there is no other delay and the processor does not suffer from any pipeline hazards. Also assume that one instruction is fetched every cycle. The total execution cycle time for executing 100 instructions on this processor is 2040 nanosecs. [2 marks]

$$\begin{array}{c}
 \text{Clock cycle} = 20 \text{ ns} \quad \begin{array}{ccc} 1 & 2 & 3 \\ & 1 & 2 & 3 \\ & & 1 & 2 & 3 \end{array} \\
 \left| \begin{array}{l} 2 + 1 \times 100 = 102 \text{ cycles} \\ = 102 \times 20 \text{ ns} = 2040 \end{array} \right.
 \end{array}$$

(c) "False sharing occurs only if a cache block contains multiple words" - True or False? Why? [2 marks]

**TRUE** Multi processor - coherence protocol - cache block - one processor modifies one word, invalidating whole line, but other processor wanted to access another word of the same line, which was falsely invalidated even though it was valid. - false sharing. If block has one word, only if it is changed, it will become invalid. It can not be falsely invalidated.

(d) In a MSI coherence protocol, when is a cache controller forced to write back a block, B? [3 marks]

- 1) when another processor requests read access for the same block, which is in modified state in processor.
- 2) when another block is read by the same processor, and the block index is same as the earlier block which was in modified state. Then earlier block is written back then evicted.

Name: HARSH VORA

Roll No: 2021CS10548

- (e) Consider a processor with 64 registers and an instruction set of size twelve. Each instruction has five distinct fields, namely, opcode, two source register identifiers, one destination register identifier, and twelve-bit immediate value. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 100 instructions, the amount of memory (in bytes) consumed by the program text is 500. [3 marks]

inst set 12  $\rightarrow$  opcode has 4 bits

64 registers  $\rightarrow$  reg id = 6 bit

2 source

1 dest

immed

= 12 bit

= 6 bit

= 12 bit

( $8 = 2^3 < 12 < 2^4 = 16$ )

inst size = 34 bit

but must be in BYTES

$\rightarrow$  5 Bytes ( $32 < 34 < 40$ )

100 inst =  $5 \times 100 = 500$  Bytes.

- (f) Consider a 3 GHz processor with a three-stage pipeline and stage latencies  $\tau_1$ ,  $\tau_2$  and  $\tau_3$  such that  $\tau_1 = 3\tau_2/4 = 2\tau_3$ . If the longest pipeline stage is split into two pipeline stages of equal latency, the new frequency is 4 GHz, ignoring delays in the pipeline registers. [3 marks]

say  $\tau_2 = 4k \Rightarrow \tau_1 = 3k, \tau_3 = 1.5k$

CC =  $4k \quad \frac{1}{4k} = 3 \times 10^9$  Hz

new: times:  $3k, 2k, 2k, 1.5k \rightarrow$  CC =  $3k$ .

freq =  $\frac{1}{3k} = 3 \times 10^9 \times \frac{4}{3} = 4 \times 10^9$  Hz = 4 GHz

- (g) Tick all that apply. Concepts taught in class to improve program performance are: [2 marks]

(a) pipelining  (b) branch prediction (c) pipeline stalls  (d) caching (e) cache coherence

- (h) Tick all that apply. Concepts taught in class to maintain program correctness are: [2 marks]

(a) pipelining (b) branch prediction  (c) pipeline stalls (d) caching  (e) cache coherence