HW #6

21700034 KYH

1. In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 32-bit integer.

```
for(I=0;I<8;I++)
    for(J=0;J<8,000;J++)
        A[I][J]=B[I][0]+A[J][I];
```

1-1. How many 32-bit integers can be stored in a 16-byte cache block?

Sol) 1 Byte = 8 bits

=> 16-Byte = 16×8 = 128 bits

$128\text{-bit}/32\text{-bit} = \underline{4}$

Ans) 4 integer numbers can be stored in a 16-byte cache block

1-2. References to which variables exhibit temporal locality?

*Temporal Locality

: reuse of specific data, and/or resources, within a relatively small time duration

Sol) I & J & B[I][0] are much likely to be accessed again soon

Ans) I, J, and B[I][0] variables exhibit temporal locality.

1-3. References to which variables exhibit spatial locality?

*Spatial Locality

: use of data elements within relatively close storage location.

Sol) A[I][J] is much likely to be accessed again soon (Sequentially)

A[J][I] is not sequentially accessed

Ans) A[I][J] variable exhibits spatial locality.

3. For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache

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

3-1. What is the cache block size (in words)?

*Cache Block
: basic unit for the cache storage.

1 words
= 4 bytes

sol) offset Byte : 5

$2^5 = 32$ bytes = Size of block

Ans)
32/4 = 8 words is the cache block size

3-2. How many entries does the cache have?

sol) index number : 5, $2^5 = 32$ bytes

direct-mapped cache => each set has only one entry.

Ans) 32 entries

3-3. What is the ratio between total bits required for such a cache implementation over the data storage bits?

sol) number of bits = Valid bits + tag bits + data bits
(in 1 entry) = 1 + 22 + (32 × 8) = 279

number of entries cache have = 2^5 = 32 entries

279 × 32 = 8928 bits (total bits)

data storage bits = 32 × (32 × 8) = 8192

Ans) The ratio between total bits required for such a cache implementation over the data storage bits is
8928/8192 = 1.08984375

Starting from power on, the following byte-addressed cache references are recorded.

Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

3-4. How many blocks are replaced?

Sol)

Byte Addr	Block Addr	Cache Index	H or M
0	0	0	Miss
4	0	0	Hit
16	0	0	Hit
132	4	4	Miss
232	7	7	Miss
160	5	5	Miss
1024	32	0	Miss ✓
30	0	0	Miss ✓
140	4	4	Hit
3100	96	0	Miss ✓
180	5	5	Hit
2180	68	4	Miss ✓

Ans) 4 blocks (1024, 30, 3100, 2180) are replaced.

3-5. What is the hit ratio?

Hit Ratio = number of hits / number of cache reference × 100(%)

Sol) $= 4/12 = 1/3 = 0.3333\ldots \times 100(\%) = 33.3333\ldots(\%)$

Ans) Hit Ratio is 33.3%.

6. In this exercise, We will look at the different ways Capacity affects overall performance. In general, Cache access time is proportional to Capacity. Assume that main memory accesses take 70 ns and that memory accesses are 36% of all instructions. The following table Shows data for L1 Caches attached to each of two processors, P1 and P2.

	L1 Size	L1 Miss Rate	L1 Hit Rate
P1	2KiB	8.0%	0.66ns
P2	4KiB	6.0%	0.90ns

6-1. Assuming that the L1 hit time determines the cycle times for P1 and P2, What are their respective Clock rates?

sol) Cycle time = L1 hit time

Clock rate = 1/Cycle time

$P1 = 1/0.66\text{ns} = 1.\dot{5}\dot{1}$
$P2 = 1/0.90\text{ns} = 1.\dot{1}$) Clock rate

Ans) P1 clock rate = 1.51 GHz, P2 Clock rate = 1.11GHz

6-2. What is the Average Memory Access Time for P1 and P2?

Average Memory Access Time = Hit time + (Miss rate × Memory Access time)

Sol)

$P1 = 0.66 + (0.08 \times 70\text{ns}) = 6.26\text{ ns}$

$P2 = 0.90 + (0.06 \times 70\text{ns}) = 5.1\text{ ns}$

Ans) Average Memory Access Time for P1 = 6.26ns
" " " " P2 = 5.1 ns

6-3. Assuming a base CPI of 1.0 without any memory stalls. What is the total CPI for P1 and P2? Which processor is faster?

Sol)

P1's CPI $= 1.0 + 0.08 \times (0.36 \times 70ns) \div 0.66ns = 4.0545...$ $\frac{4.05}{2.68}$

P2's .. $= 1.0 + 0.06 \times (0.36 \times 70ns) \div 0.9ns = 2.68$ $= 1.5111...$

Ans) P2 is faster than P1 by 1.5.

6-4. What is the AMAT for P1 with the addition of an L2 Cache? Is the AMAT better or worse with the L2 Cache?

In Multi-layered Cache,

AMAT = L1 Hit Time + L1 Miss Time × (L2 Hit Time + L2 Size + L2 Miss Rate)

Sol)

P1 AMAT $= 0.66ns + 0.08 \times (5.62ns + 0.95 \times 70ns) = 6.4296(ns)$

P2 .. $= 0.9ns + 0.06 \times (5.62ns + 0.95 \times 70ns) = 5.2272(ns)$

Ans)

AMAT Worse with the L2 Cache.

6-5. Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

$$CPI = \text{Memory access ratio} \times \frac{AMAT}{\text{Hit time}} + (1 - \text{Memory access ratio})$$

Sol)

Total CPI (P1 with L2 Cache) $= 0.36 \times \frac{6.4296}{0.66} + (1 - 0.36) = 4.1470...$

Ans)

Total CPI for P1 withe L2 Cache = 4.15

6-6. Which Processor is faster, now that P1 has an L2 Cache? If P1 is faster, What miss rate would P2 need in its L1 Cache to match P1's performance?

- If P2 is faster, What miss rate would P1 need in its L1 Cache to match P2's performance?

Sol)

AMAT (P1, L2 Cache) = 6.43ns

AMAT (P2) = 5.1 ns

Ans)

P2 is faster than P1

Sol)

$$\text{AMAT}(\text{new P1}) = \underset{ns}{0.66} + \underset{(\text{new P1})}{\text{Miss rate}} \times (5.62 + 0.95 \times 70\text{ns}) = 5.1$$

$$\underset{(\text{new P1})}{\text{Miss rate}} = (5.1 - 0.66) \div (5.62 + 0.95 \times 70\text{ns})$$

$$= 0.0615640599\ldots \qquad 0.062 \times 100 = 6.2\%$$

Ans)

6.2% miss rate for new P1 need

12. In this exercise, we will examine Space/time Optimizations for page tables. The following list provides parameters of a virtual memory System.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size(byte)
43	16 GiB	4KiB	4

12-1. For a Single-level page table, how many page table entries (PTEs) are needed? How much Physical memory is needed for storing the page table?

Sol)

Page Size = 4 KiB = 2^{12} bytes

Number of page table entries $= \dfrac{\overset{\text{Virtual Address}}{2^{43}}}{\underset{\text{Page Size}}{2^{12}}} = 2^{31}$

PTE Size = 4 = 2^2 bytes

Physical memory (needed) $= 2^{31} \times 2^2 = 2^{33}$ (bytes)

Ans)

2^{33} bytes Physical memory is needed.