Name: Jude Onyia

Student ID: V00947095

Course: ECE 596C

Due Date: July 15, 2020

Assignment 5: Non – Programming Exercise

8.21)

$Block\ size = 64\ bytes = 2^6\ bytes$

$Therefore: \quad Block\ Offset = 6\ bits$

$Since\ 4-way\ set\ associative,\ each\ set = 4\ blocks = 4(64\ bytes) = 256\ bytes$

$Since\ capacity\ of\ cache = 32KB,\ the\ number\ of\ sets\ in\ cache = 32KB \times \frac{1\ set}{256\ bytes}$

$= 2^5 \times 2^{10} \times \frac{1\ set}{2^8} = 2^7\ sets = 128\ sets$

$Therefore: \quad Index = 7\ bits$

$Since\ the\ number\ of\ bits\ in\ an\ address\ is\ 32,\ Tag = 32 - 7 - 6 = 19\ bits$

$Tag = 19\ bits,\ \ Index = 7\ bits,\ Block\ Offset = 6\ bits$

8.20 a)

$Since\ index = 8\ bits\ \ and\ \ cache\ is\ 2-way\ associative,$

$Number\ of\ blocks\ in\ cache = Number\ of\ sets \times Number\ of\ blocks\ in\ a\ set$

$= 2^8 sets \times 2\ blocks\ per\ set = 512\ blocks$

8.20 b)

$Block\ addess = tag + index = 14\ bits + 8\ bits = 22\ bits$

$Therefore,\ number\ of\ blocks\ in\ memory = 2^{22} = 4194304\ blocks$

8.20 c)

$Address = 557A02_{16}$

$Tag = 0101\ 0101\ 0111\ 10_2$

$Index = 10\ 0000\ 00_2$

$Offset = 10_2$

This byte was present in the cache and its value is $C2_{16}$

8.20 d)

$Address = FFFFFF_{16}$

$Tag = 1111\ 1111\ 1111\ 11_2$

$Index = 11\ 1111\ 11_2$

$Offset = 11_2$

This byte was not present in the cache, no tags at index $11\ 1111\ 11_2$ matches its tag.

8.22)

Matrix a of size 1024×1024 is aligned on a 64-byte boundary and the block size of the cache is 64 bytes, therefore, we do not need to worry about additional misses caused by alignment. An object of type double requires 8 bytes of storage, therefore each block of the cache can store: $64\ bytes \times \frac{1\ double}{8\ bytes} = 8\ elements\ of\ type\ double.$

Code fragment A accesses matrix a in a cache efficient way, walking through the rows of matrix a. Since the language uses row major, the number of cache misses that occurs during the execution is: $\frac{Number\ of\ elements\ in\ matrix}{Number\ of\ elements\ a\ block\ can\ hold} = \frac{1024 \times 1024}{8} = 2^{17} = 131072\ cache\ misses.$

Code fragment B does not access the matrix in a cache efficient way. Fragment B walks through the columns of matrix a, requiring large strides in memory as it iterates. However, the capacity of the cache is 8KB. This results in $number\ of\ blocks = \frac{capacity}{block\ size} = \frac{8K\ bytes}{64\ bytes\ per\ block} = 128\ blocks$. Also, since the size of the column of the matrix is $1024\ elements \times \frac{1\ block}{8\ elements} = 128\ blocks$, the column of the matrix can indeed fit in the cache. Since the column of the matrix can fit in the cache, as the program goes down the first column (causing cache misses), the next column will result in cache hits because the blocks are still in the cache. Therefore, the number of cache misses that occurs during execution is: $\frac{Number\ of\ elements\ in\ matrix}{Number\ of\ elements\ a\ block\ can\ hold} = \frac{1024 \times 1024}{8} = 2^{17} = 131072\ cache\ misses.$

8.27)

The system has $page\ size = 1KB = 2^{10} bytes$, therefore, $Page\ offset = 10\ bits$. Since, virtual address is 24 bits, the virtual page number has $24 - 10 = 14\ bits$. The number of virtual pages is $2^{14} = 16384\ pages$. Since, the physical address is 16 bits, the physical page number has $16 - 10 = 6\ bits$. The number of physical pages is $2^6 = 64\ pages$.

8.28 a)

Virtual page number has 14 bits

Physical page number has 6 bits

Page offset has 10 bits

In virtual address $0000\ 0000\ 0000\ 0000\ 0000\ 0000_2$, the virtual page number $0000\ 0000\ 0000\ 00_2$ is not in the page table. The protection check will result in an access violation.

8.28 b)

Virtual address $0000\ 0000\ 0000\ 1100\ 1100\ 1100_2$ maps to the physical address: $0010\ 1000\ 1100\ 1100_2$. The protection check will allow the write because this page is writable.

8.28 c)

Virtual address $1111\ 1111\ 1111\ 1100\ 0000\ 0000_2$ maps to the physical address: $0011\ 0000\ 0000\ 0000_2$. The protection check will result in an access violation because this page is not executable, so it cannot be fetched for execution.