Assignment 9

Alex Clemmer
Student number: u0458675

Problem 1:

Spatial locality. If things are close together, and you have a lot of space in cache, clearly you will not miss as often.

Problem 2:

5.4.1

For **(a)** the cache line size is 4 words, with $21 + 1$ valid bit, $+4words = 150$ bits to hold the whole line. For **(b)**, it's 8, with $19 + 1$ valid bit $+8words = 276$ bits to hold the whole line.

5.4.2

(a) has 64 entries, and **(b)** has 128 entries.

5.4.3

For **(a)**, I would guess that $1 + (\frac{\frac{22}{8}}{16}) \approx 1.172$. Similarly, for **(b)**, we have $1 + (\frac{\frac{20}{8}}{32}) \approx 1.078$

Problem 3:

5.3.2

Assume that each of these addresses is in binary. In order, their addresses are:
`1, 10000110, 11010100, 1, 10000111, 11010101, 10100010, 10100001, 10, 101100, 101001, 11011101.`

The tag is just a binary address right-shifted 3 bits. The index can be obtained by shifting the binary address 1 bit to the left and then modding that by 8.

As for misses and hits, the sequence should go: MMMHHHMMMMM.

In the case of **(b)**:
`00000110, 11010110, 10101111, 11010110, 00000110, 01010100,`
`01000001, 10110000k 01000000, 01101001, 01010101, 11010111.`

The tag of course is just the address shifted right three bits, and the index (again) is a right shift 1 and then a mod 8.

The misses again demand their own section: MMMHMMMHHMHM.

5.3.3

The best miss rate for **(a)** would be C1(1 hit), C2 (3 hits), C4 (2 hits). Given these conditions, I suppose you want me to find the best design in terms of stall time, in which case: C1 $25 \cdot 11 + 2 \cdot 12 = 299$; C2 $25 \cdot 9 + 3 \cdot 12 = 261$; C3 $25 \cdot 10 + 4 \cdot 12 = 298$.

The best miss rate for **(b)** would be C1 (1hit), C2 (4 hits), C3 (4 hits). Given that, the stall times should be: C1 $25 \cdot 11 + 2 \cdot 12 = 299$; C2 $25 \cdot 8 + 3 \cdot 12 = 236$; C3 $25 \cdot 8 + 5 \cdot 12 = 260$.

Problem 4:

5.6.4

The optimal block size given **(a)** should be 16 bytes. For **(b)**, it should be 8 bytes.

5.6.5

The optimal block size given **(a)** should be 32 bytes. For **(b)** it should be 8 bytes.

Problem 5:

So the memory miss is 125 cycles / (1/3) = 375 clock cycles. Given the base CPI time of 2:

First level cache would be $2 + 375 \cdot 5\% = 20.75$, or 39.5 in the case of doubling and 11.375 in the case of halving.

Second-level direct-mapped cache will be $2 + 15 \cdot 5\% + 375 \cdot 3\% = 14$ or 25.25, or 8.375.

Second-level eight-way associative works out to be $2 + 25 \cdot 5\% + 375 \cdot 1.8\% = 10$ or 16.75, or 6.625.

Problem 6:

Given **(a)**, we have a virtual address of 32 bits, physical memory of 4GB with page size of 8 KB, and a page table entry size of 4 bytes. So for a single level table, we would need $32 - 13 = 19$ bits or 512K entries. Thus the page table should be $512K \cdot 4$ bytes $= 2MB$.

Given **(b)**, the virtual address is 64 bits, the physical memory is 4GB, page size is 4 KB, and PTE is 8 bytes. That means that a single-level table should be $64 - 12 = 52$ bits or 2^{52} entries, which seems wrong but isn't. The page table physical memory should thus be $2^{52} \cdot 8 = 2^{55}$ bytes.