Christopher Daffron CS 530 HW #2

1-2.4) a) The overall size of the second-level cache is 2 MB. This is shown on the graph because the 2MB array is the smallest array that is part of the top plateau. The block size of the cache is 128B because that is the stride size where the top plateau starts.

b) The miss penalty of the second level cache is 100 ns. This is the read time of the top plateau.

c) The associativity of the cache is 8-way because the read time of the large array drops from miss time to hit time when the stride is $\frac{\text{array-size}}{8}$.

d) The main memory size is 512 MB because the 512MB array has such high read times that it's not even on the graph.

2-2.8) a) direct mapped: 0.8625 ns

2-way associative: 1.1206 ns 4-way associative: 1.3714 ns

as associativity goes up, so does access time.

b) 16K: 1.2688 ns 32K: 1.3489 ns 64K: 1.3714 ns

as size goes up, so does access time

3-B.1) a) AMAT = 1 + (0.05)(105) = 6.25 cycles

b) assume hit rate = $\frac{64K}{256M} = 0.00025$

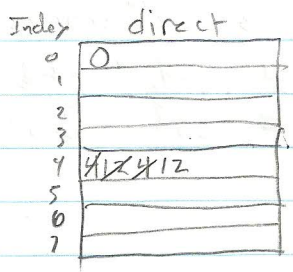
So miss rate = 1 − hit rate = 1 − 0.00025 = 0.99975

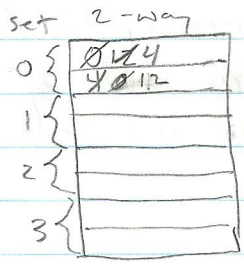
AMAT = 1 + 0.99975(105) = 105.974 cycles

c) If the cache is disabled, AMAT = 100 cycles because of main memory. The principle of locality is very important. Without it, cache would not help any, it would actually hurt performance.

4-B.9) both caches have 8 lines (though size doesn't matter for this to work, the address pattern would just have to be scaled) Repeating the following address pattern would cause the direct mapped to have a 66% miss rate compared to 100% for a direct mapped, as shown below.

Address	direct	2-way
0	M	M
4	M	M
12	M	M
0	H	M
4	M	M
12	M	M
⋮	⋮	⋮





5-B.12)

V Page #	TLB	Pg Table
1	m	F
5	H	X
9	m	F
14	m	F
10	m	H
6	m	H
15	m	H
12	m	H
7	m	H
2	m	F

6) a) $\frac{8\text{ kB}}{4\text{ bytes/word}} = 2048\text{ words}$ $\frac{2048\text{ words}}{8\text{ words/block}} = 256\text{ lines}$

b) Assuming 32 bit addresses, byte addressed

2 bits - byte offset 3 bits - block offset 8 bits - index

so tag = 32 - 2 - 3 - 8 = 19 bits

c) 256 tags = 256 × 19 = 4864 bits

256 valid bits, 256 dirty bits, 65536 data bits

Total = 4864 + 256 + 256 + 65536 = 70912 bits

7) d) $\frac{8\text{ kB}}{4\text{ bytes/word}} = \frac{2048\text{ words}}{8\text{ words/block}} = 256\text{ lines}$ (in 64 sets)

e) 2 bits - byte offset 3 bits - block offset 6 bits - set index

tag = 32 - 2 - 3 - 6 = 21 bits

f) 256 tags = 256(21) = 5376 bits

256 valid bits, 256 dirty bits, 65536 data bits

assuming random replacement, so no LRU bits

Total = 5376 + 256 + 256 + 65536 = 71424 bits

8) Pages = 4 kB so page offset = 12 bits

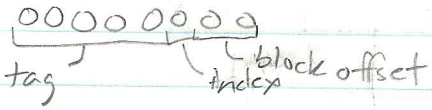
virtual pg # = 20 bits so 2^{20} entries and physical address = 24 bits

physical address = 12 bits pg # + 12 bits pg offset

so total = $(2^{20})(12) = 12582912$ bits

9)

Address	Result
4	m
7	H
15	m
11	m
13	H
2	m
9	H
32	m
45	m
1	m
9	m
7	m
8	H
52	m
21	m
12	m

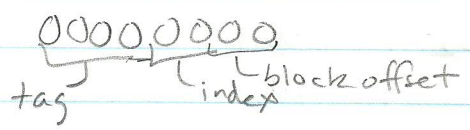


Final State:

Set	Index	Tag	Data (Words)
0	0	00000	0, 1, 2, 3
	1	00001	8, 9, 10, 11
1	2	00011	12, 13, 14, 15
	3	00010	20, 21, 22, 23

10)

Address	Result
4	m
7	H
15	m
11	m
13	H
2	m
9	H
32	m
45	m
1	m
9	H
7	H
8	H
52	m
21	m
12	m



Final State:

Index	Tag	Data (words)
0	0000	0,1,2,3
1	0001	20,21,22,23
2	0000	8,9,10,11
3	0000	12,13,14,15