CSCI 524 – Computer Architecture
Homework 6

Due: November 10, 2016

Alexander Powell

1. a) The table below shows whether each address in the reference string is a cache hit or miss.

Value	Binary	Hit/Miss	Block
1	0001	Hit	0001
18	10010	Miss	0010
2	0010	Miss	0010
3	0011	Hit	0011
4	0100	Hit	0100
20	10100	Miss	0100
5	0101	Hit	0101
21	10101	Miss	0101
33	100001	Miss	0001
34	100010	Miss	0010
1	0001	Miss	0001
4	0100	Miss	0100

So, there were 4 cache hits and 8 misses so the cache hit rate is $\frac{1}{3}$. This table shows the contents of the cache after the last reference.

Index	0000	0001	0010	0011	0100	0101	0110	0111
Value	0	1	34	3	4	21	6	7
Index	1000	1001	1010	1011	1100	1101	1110	1111
Value	8	9	10	11	12	13	14	15

b) The table below shows whether each address in the reference string is a cache hit or miss.

Value	Binary	Hit/Miss	Block
1	0001	Miss	000
18	10010	Miss	001
2	0010	Miss	001
3	0011	Hit	001
4	0100	Miss	010
20	10100	Miss	010
5	0101	Miss	010
21	10101	Miss	010
33	100001	Miss	000
34	100010	Miss	001
1	0001	Miss	000
4	0100	Miss	010

There was only 1 cache hit and 11 misses, so the hit rate is $\frac{1}{12}$. After the last reference the cache looks like the following (a hyphen denotes an empty cache block):

Index	000	001	010	011
Value	0, 1	34, 35	20, 21	-
Index	100	101	110	111
Value	-	-	-	-

c) The table below shows whether each address in the reference string is a cache hit or miss.

Value	Binary	Hit/Miss	Block
1	0001	Miss	000
18	10010	Miss	001
2	0010	Miss	001
3	0011	Hit	001
4	0100	Miss	010
20	10100	Miss	010
5	0101	Miss	010
21	10101	Miss	010
33	100001	Miss	000
34	100010	Miss	001
1	0001	Miss	000
4	0100	Miss	010

There were 5 cache hits and 7 misses, so the hit rate is $\frac{5}{12}$. After the last reference the cache looks like the following (a hyphen denotes an empty cache block):

Index	000 000	001 001	010 010	011 011
Value	0, 1 32, 33	34, 35 2, 3	4, 5 20, 21	- -
Index	100 100	101 101	110 110	111 111
Value	-	-	-	-

2. a) Clock rate is equal to $\frac{1}{\text{Hit time}}$, so in this case we have $\frac{1}{0.5 \text{ nsec}} = 2$ GHz.

b) The average memory access time is equal to the hit time plus the miss rate multiplied by the miss penalty. By pluggin in the given values, we get $0.5 + 0.08 \times \frac{100}{.5} = 16.5$ nsec.
The actual CPI can be calculated as $1 + \frac{1}{0.5} \times (0.08 \times (100 - 0.5) \times 0.35) = 6.57$.

c) Again, the average memory access time is equal to the hit time plus the miss rate multiplied by the miss penalty but we need to take into account both the L1 and L2 caches. The AMAT is calculated as $0.5 + 0.08 \times (5 + 0.95 * (100 - 5)) = 8.12$.
Again, to calculate the actual CPI we have $1 + 0.35 \times 0.08 \times \frac{100}{5} = 7.65$. 1+ 1/.5 * .35*.08*95.25

3. The tables below show the performance results of the six benchmarks on the five different machines.

bzip2

	M1	M2	M3	M4	M5
sim IPC	0.2852	0.2674	0.4306	0.6521	0.6521
il1 hits/misses	2000362/571	2000369/571	2000407/480	2000404/480	2000404/480
dl1 hits/misses	414598/296557	414598/296557	414600/296557	488333/222829	488333/222829
ul2 hits/misses	339576/201147	339588/201135	33921/201155	226995/201251	226995/201251
itlb hits/misses	2000916/17	2000923/17	2000870/17	2000867/17	2000867/17
dtlb hits/misses	666080/45077	660080/45077	666082/45077	666087/45077	666087/45077

equake

	M1	M2	M3	M4	M5
sim IPC	0.6489	0.4785	0.7894	0.7439	0.7439
il1 hits/misses	2093198/180319	2093881/180319	2102027/21086	2102861/21086	2102861/21086
dl1 hits/misses	747261/16447	747262/16447	755807/7441	755041/6138	755027/6152
ul2 hits/misses	204515/3359	204515/3359	30948/3359	25926/3359	25950/3359
itlb hits/misses	2273505/12	2274188/12	2123101/12	2123935/12	2123935/12
dtlb hits/misses	772065/58	772066/58	773896/58	773894/58	773894/58

hmmer

	M1	M2	M3	M4	M5
sim IPC	0.4751	0.4584	0.4558	0.4353	0.4353
il1 hits/misses	2011833/9127	2012044/9127	2013122/4545	2012352/4545	2012423/4545
dl1 hits/misses	925733/32138	925733/32138	929051/29605	929751/28946	929741/28964
ul2 hits/misses	28189/14633	28189/14633	20121/14633	19240/14633	19270/14633
itlb hits/misses	2020928/32	2021139/32	2017635/32	2016865/32	2016936/32
dtlb hits/misses	957644/227	957644/227	958430/227	958471/227	958479/227

mcf

	M1	M2	M3	M4	M5
sim IPC	0.8603	0.7122	0.9757	0.9519	0.9518
il1 hits/misses	2047813/82138	2047819/82138	2054079/10974	2053117/10974	2053117/10974
dl1 hits/misses	987690/118862	987690/118862	991832/116141	992026/115947	991997/115976
ul2 hits/misses	259383/58218	259386/58215	183425/58214	180166/58214	180211/58214
itlb hits/misses	2129939/12	2129945/12	2065041/12	2064079/12	2064079/12
dtlb hits/misses	1105631/921	1105632/921	1108013/921	1108493/921	1108493/921

milc

	M1	M2	M3	M4	M5
sim IPC	0.8687	0.8848	0.8826	0.8785	0.8784
il1 hits/misses	2011304/4450	2011307/4450	2011505/4441	2011505/4441	2011505/4441
dl1 hits/misses	992523/137636	992523/137637	992818/137340	992927/137234	992908/137253
ul2 hits/misses	209064/69762	209865/68961	208677/58214	205457/68961	205485/68961
itlb hits/misses	20155740/14	2015743/14	2015932/14	2015932/14	2015932/14
dtlb hits/misses	1128141/2018	1128141/2018	1128140/2018	1128143/2018	1128143/2018

sjeng

	M1	M2	M3	M4	M5
sim IPC	0.9311	1.0131	1.0137	0.9415	0.9415
il1 hits/misses	2304395/241	2304397/241	2264719/221	2254268/221	225468/221
dl1 hits/misses	732115/37162	732115/37162	739869/29542	741759/27652	741759/27652
ul2 hits/misses	50599/12410	52149/10860	40217/10860	34377/10860	34377/10860
itlb hits/misses	2304625/11	2304627/11	2264929/11	2254478/11	2254478/11
dtlb hits/misses	769096/181	769096/181	769230/181	769230/181	769230/181

When examining the integer benchmarks, like bzip2 and mcf, generally the number of instructions per cycle increased as we made changes to the caches. However, in some of the experiments, these changes didn't necessarily help. For example, in the equake benchmark, quadrupling the size of the unified L2 cache caused a performance decrease. This is due to the higher latencies associated with a larger cache.

It should be noted, however, that performance was gained back in machine 3. Basically, we should find an optimal point where improvements are made from using a cache with larger storage ability, but whose latencies do not cause performance drops. Also, generally no changes were observed between machine 4 and 5, where we changed the replacement policy from LRU to FIFO. In conclusion, we see that by increasing the cache size, the hit rates generally increase as well, although this sometimes comes with the added effect of a higher latency.