Name Answers

CE-1921-11 – Dr. Durant – Quiz 9
Spring 2017, Week 10

1. (1 point) ***Define spatial*** locality and ***explain*** how a cache might exploit it.
2. (1 point) A 2048 B direct-mapped cache is divided into 32 blocks. ***Calculate*** how many ~~***set***~~ offset ***bits*** there are.
3. (2 points) Continuing, there are 32 address lines. ***Show*** how the address is broken down into set, offset, and tag bits. (-1/2) xchg set/offset
4. (2 points) ***Calculate*** an ***example*** of 2 read addresses used consecutively that will cause the first read data to be evicted from the cache. (-1) if same tag, but rest ok
5. (1 point) ***Show*** how the address format will change if the cache is instead organized as a ***4-way*** set associative cache, but nothing else changes.
6. (2 points) ***Confirm by calculation*** that the 2 addresses above are still in the same set. Explain why this must be the case.
7. (1 point) ***Explain*** how the ~~2~~ 4-way set associative cache avoids eviction of the first data read.

① after accessing a program/data location, it is likely that nearby locations will be accessed soon due to the way programs & data are organized. Therefore, having a moderate/large blocksize will "pre-cache" I/O likely to be used soon.

② $B = \frac{C}{N} = \frac{2048\ B}{32} = \frac{2^{11}}{2^5} B = 2^6 B = 64\ B$ (blocksize)

bits $= \log_2 |$addressable values$| = \log_2 64 = \boxed{6}$

③ set bits $= \log_2 |N| = 5$

tag	set	offset
31 … 11	10 … 6	5 … 0

④

t	s	o		
1	4	0	→ 1_00100_000000	→ 0x00000900
2	4	0	→ 10_00100_000000	0x00001100
different	same	doesn't matter		

⑤ S≠N, $S = \frac{N}{4\text{ ways}} = \frac{32}{4} = 8$ sets → 3 set bits

tag	set	offset
31 … 9	8 … 6	5 … 0

⑥

		tag	set	offset
0x0900 →	0000	100	0000	0000
0x1100 →	0001	0001	0000	0000

both

Truncated equal #s, so the truncated results are equal

⑦ It puts the data 2 from the 2nd tag in the 2nd way of the same set.

Name Answer

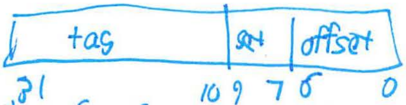
CE-1921-21 – Dr. Durant – Quiz 9
Spring 2017, Week 10

1. (1 point) ***Define temporal*** locality and ***explain*** how a cache might exploit it.
2. (1 point) A 1024 B direct-mapped cache is divided into 8 blocks. ***Calculate*** how many ***set bits*** there are.
3. (2 points) Continuing, there are 32 address lines. ***Show*** how the address is broken down into set, offset, and tag bits.
4. (2 points) ***Calculate*** an ***example*** of 2 read addresses used consecutively that will cause the first read data to be evicted from the cache.
5. (1 point) ***Show*** how the address format will change if the cache is instead organized as a 2-way set associative cache, but nothing else changes.
6. (2 points) ***Confirm by calculation*** that the 2 addresses above are still in the same set. Explain why this must be the case.
7. (1 point) ***Explain*** how the set associative cache avoids eviction of the first data read.

① Once an instruction/data location is accessed, it is likely to be accessed again ~~soon~~ due to loops & how data structures work. So, it should be kept in the cache to speed future access.

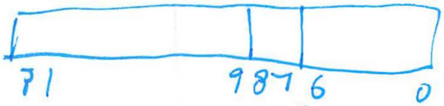
② 8 blocks → 8 sets → $\log_2 8 = 3$ set bits

③ $B = \frac{C}{N} = \frac{1024\ B}{8} = 128\ B$ blocksize → $\log_2 128 = 7$ offset bits



④ T S O
3 5 0 → 11_101_0000000 → 0x00000E80
4 5 0 → 100_101_0000000 → 0x00001280
↑ different ↑ same ↑ don't care

⑤ $S = \frac{N}{2 \leftarrow \text{ways}} = \frac{8}{2} = 4$ sets → 2 set bits



⑥ set offset
0x0E80 = 0000 1110 1000 0000
0x1280 = 0001 0010 1000 0000
both

In the set, we are truncating 2, equal 3-bit #s, so the results are equal.

⑦ It puts data for the 2nd tag in the 2nd way of the same set.