**1. Problem 6.26**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cache | *m* | *C* | *B* | *E* | *S* | *t* | *s* | *b* |
| 1. | 32 | 2048 | 8 | 1 | 256 | 21 | 8 | 3 |
| 2. | 32 | 2048 | 8 | 2 | 128 | 23 | 7 | 2 |
| 3. | 32 | 1024 | 2 | 8 | 64 | 25 | 6 | 1 |
| 4. | 32 | 1024 | 32 | 2 | 16 | 23 | 4 | 5 |

**2. Problem 6.27**

**A. List all of the hex memory addresses that will hit in set 1.**

Both valid. Tag 45 and 38. So it’s all the address that are 0 1000 1010 01xx and 0 0111 0000 01xx.

0x08A4, 0x08A5, 0x08A6, 0x08A7, 0x0704, 0x0705, 0x0706, 0x0707.

**B. List all of the hex memory addresses that will hit in set 6.**

One valid. Tag 91. So it’s all the addresses that are 1 0010 0011 10xx.

0x1238, 0x1239, 0x123A, 0x123B.

**3. Problem 6.29**

**A.**

E = 2, B = 4, S = 4, m = 13, t = 9, s = 2, b = 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CT | CT | CT | CT | CT | CT | CT | CT | CT | CI | CI | CO | CO |
| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

**B.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Operation | Address (hex) | Address (bin) | Tag | Set | Offset | Hit? | Read Value (or unknown) |
| Read | 0x834 | 0 1000 0011 0100 | 83 | 1 | 0 | No | Unknown |
| Write | 0x836 | 0 1000 0011 0110 | 83 | 1 | 2 | Yes | Unknown |
| Read | 0xFFD | 0 1111 1111 1101 | FF | 3 | 1 | Yes | C0 |

**4. Problem 6.38**

**A.** 16x16 = 256

**B.** 32

**C.** 12.5%

**5. Problem 6.39**

**A.** 256

**B.** 64

**C.** 25%

**6. Problem 424-7**

The results for this program are not very consistent between repeated program calls. Below is a chart of the results I obtained when running the cacheperf with different options. It looks like the difference between the cold cache and warm cache is most apparent starting when SQSIZE is 16. From then on it seems to hold at about the same ratio. However, when compiled with the flag ‘-O3’ there seems to be much larger discrepancies that are not consistent. Without using compile flags, it seems that the poor locality code ran in about the same time as the code with good locality. However, when compiled with the ‘-O3’ flag, it seems to take almost 2x as long when SQSIZE is above 17.