Canvas Group Name: group\_58 Student 1: Kevin Le, kevle@ucsc.edu Student 2: Matthew Wark, Mtwark@ucsc.edu

CMPE110 HA4: Cache Detective

Due Date: Thursday 05/10/18

1. Consider a 16-bit processor with a directly mapped cache with 8 Byte cache blocks, and 8 entries. Fill out the bitfields in the address map below showing how the address bits are used to lookup a value in the cache. **(2 Points)**

2^k = 8 entries

k = 3

2^n = 8 cache blocks

n = 3

**k bits n bits**

|  |  |  |
| --- | --- | --- |
| Tag | Index | Offset |

15 5 2 0

1. Assume the following address sequence is used to access the cache: 4033, 3077, 1416, 3529, 4033, 3077, 3076, 3200, 4264, 2222, 3333. Determine the hit and miss rate. Determine the hit and miss rate. Use the same cache block size and number of entries as 1). Show the hits and misses by filling out the table (for the whole sequence) below -

**(4 Points)**

* Divide Decimal byte address by 8 bytes because we’re dealing with an 8 byte cache block.
* <https://www.youtube.com/watch?v=RqKeEIbcnS8>
* Check tag and index to see if each unique combination matches each other once we go down the line.
* Offset doesn’t matter in this example?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decimal byte address | Binary address of block reference | Tag | Index AKA Assigned cache block (where found (hit) or placed (miss)) | Hit or miss in cache |
| 4033 | 504= 111111000 | 111 | 111 | Miss |
| 3077 | 384= 110000000 | 110 | 000 | Miss |
| 1416 | 177= 010110001 | 010 | 110 | Miss |
| 3529 | 441= 110111001 | 110 | 111 - replaced tag 111 in first row with tag 110 from this row | Miss |
| 4033 | 504= 111111000 | 111 | 111 | Miss |
| 3077 | 384= 110000000 | 110 | 000 | Hit |
| 3076 | 384= 110000000 | 110 | 000 | Hit |
| 3200 | 400= 110010000 | 110 | 010 | Miss |
| 4264 | 533= 1000010101 | 000 | 010 | Miss |
| 2222 | 277= 100010101 | 100 | 010 | Miss |
| 3333 | 416= 110100000 | 110 | 100 | Miss |

Hit Rate - 2/11 = 1/6 = 18.1818%

Miss Rate-9/11 = 5/6 = 81.8181%

1. Consider the access sequence form 2) and draw the state of the cache at the end of that sequence, filling out the table below. **(4 Points)**

* If valid bit is 0, we put data into cache block -> then change it to 1
  + Valid bit = 1 = occupied
  + Valid bit = 0 = simply put a new value in
* In determining valid or not, we are going down the list of index from #2 and going down that line.
  + When we finish index 111, we put valid = 1 in this table.
  + Keep going down the line and overwrite data if needed.
  + All values here represent the last time the value was referenced

|  |  |  |  |
| --- | --- | --- | --- |
| Index | valid | Tag | Data |
| 000 | 1 | 0000000110 | Mem 3076 |
| 001 | 0 |  |  |
| 010 | 1 | 0000000100 | Mem 2222 |
| 011 | 0 |  |  |
| 100 | 1 | 0000000110 | Mem 3333 |
| 101 | 0 |  |  |
| 110 | 1 | 0000000010 | Mem 1416 |
| 111 | 1 | 0000000111 | Mem 4033 |

1. Assume the address sequence from 2) with a 16 entry, 32B block, fully associative cache. What is the Hit and Miss rate now? **(4 Points)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Decimal Byte Address** | **Binary Block Address** | **Cache Block**  **(From 0-15)** | **Hit or Miss** |
| 4033 | 126=1111110 | 0000 | Miss |
| 3077 | 96= 1100000 | 0001 | Miss |
| 1416 | 44= 101100 | 0010 | Miss |
| 3529 | 110= 1101110 | 0011 | Miss |
| 4033 | 126= 1111110 | 0000 | Hit |
| 3077 | 96= 1100000 | 0001 | Hit |
| 3076 | 96= 1100000 | 0001 | Hit |
| 3200 | 100= 1100100 | 0100 | Miss |
| 4264 | 133= 10000101 | 0101 | Miss |
| 2222 | 69= 1000101 | 0110 | Miss |
| 3333 | 104= 1101000 | 0111 | Miss |

Hit Rate: 3/11 = 27.2727% Miss Rate: 8/11 = 72.7272%

1. a) Consider a processor with two levels of cache with the following properties: L1 Hit rate: 90%, L1 Access (hit): 4 cycles, L2 Hit Rate: 80%, L2 Access (hit): 12 cycles, DRAM Hit Rate: 100%, DRAM Access (hit): 110 cycles. Compute the AMAT. **(3 Points)**

Hit Time = Hit rate \* Access Time(Cycles)

AMAT - Average Memory Access Time = Hit Time + Miss Rate \* Miss Penalty

* The Miss Rate will vary depending on what level cache you are accessing. If you miss in L1 cache, you need to check in L2 cache, therefore however long it takes you to get your data from the L2 cache is the miss penalty for L1. If it’s not in L2 cache, you need to go to L3, and then to DRAM, so for each level of memory, the Miss Penalty is dependent on the AMAT of the next level of memory.

L1 Hit Time = L1 Hit rate \* L1 Access Time = .9 \* 4 = **3.6**

L2 Hit Time = L2 Hit rate \* (L1 Access Time + L2 Access Time) = .8 \* (4 + 12) = **12.8**

DRAM Hit Time = DRAM Hit Rate \* (L1 Access Time + L2 Access Time + DRAM Access Time) = 1 \* (4 + 12 + 110) = **126**

Miss Penalty = Current level access time + Access Time of next level

AMAT = L1 Hit Time + (L1 Miss \* (L2 Hit Time + (L2 Miss \* (DRAM Hit Time + DRAM Miss \* DRAM Pen))))

AMAT = 3.6 + (.1 \* (12.8 + (.2 \* (126 + 0))))

= 3.6 + (.1 \* (12.8 + 25.2)

= 3.6 + 3.8

= 7.4

= **8 cycles**

b) Consider a processor that has a CPI of 1 for all instructions besides loads. Loads have the AMAT, calculated above. Also, consider that 20% of the instructions being executed are loads. What is the average CPI of the processor? **(1 Point)**

Avg CPI =Typical CPI + Average time spent loading

Typical CPI = 1

Average time spent loading = AMAT \* How often we have to go further than L1 cache

= AMAT \* L1 Miss

= 8 \* .1

Avg CPI = 1 + .8 = **1.8 CPI**



