Activity 1 (Direct-Map Cache)

1. Cache is our RAM Memory that CPU uses to quickly access information. Cache is smaller than and faster reading/writing speed than main memory.

4) 16 -> log2(16) -> 4 bits of offset (**OFFSET)**

Log2(block\_size) is the function to calc offset

LW/SW has an impractical use of a blocksize of 1 byte. A word is (2 bytes) which is not large enough to even contain these values.

(rows per block)

5) 64 -> log2(64) = 6 bits for **INDEX**

Log2(num\_blocks\_) is the function to calc INDEX

6) num\_blocks \* block\_size gives total size of cache

7)left over bits from the address -> tag

There are multiple memory locations that information can be located in cache, we need tag to find it in cache

8)address 20

128B of cache

8B block size

2^20 bits = the size of main memory (A)

Log2(block\_size) -> log2(8) = 3 bits offset (B)

Num\_blocks = 128 / 8 = 16 bits (C)

Log2(num\_blocks) = log2(16) = 4 bits index (D)

20 – 3 – 4 = 13 bits for tag (E) (also the remaining bits )

8B block size = 64 bits of data

|V | d | tag | data | (F)

|1 | 1 | 13 | 8 bytes |

1 + 1 + 13 + 8 = 79 bits per row

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Valid | Dirty | Data |
| 13 | 1 | 1 | 64 bits |
| 13 | 1 | 1 | 64bits |

Activity 2 (N-Way set associative cache)

1. If two different blocks map to a same location, and continue to reference eachother, the two blocks information will continually be swapped between the two mapped locations. Also, multiple tag bits could be the same and this changing the information of wanted memory locations due to all sharing the same tag bits

4)

Address 20 bits

128B of 2-way cache

8B block size

1. 2^20 bits -> size of main memory
2. Log2(8) = 3 bits offset
3. 128/ 8 = 16 bits blocks in cache
4. Num\_sets = 16/ 2 = 8 sets
5. Log2(num\_sets) = log2(8) = 3 bits index
6. 20 – 3 – 3 = 14 bits for tag

- - - - - - - - - - - - - - -- - - - - - - - - -- - - - - - -

| V | D | tag. | data | | 1 | 1 | 14. | 8 bytes |

- - - - - - - - - - - - - - -- - - - - - - - - -- - - - - - -

| V | D | tag. | data | | 1 | 1 | 14. | 8 bytes |

- - - - - - - - - - - - - - -- - - - - - - - - -- - - - - - -

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valid | Dirty | Tag | Data |  | Valid | Dirty | Tag | Data |
| 1 | 1 | 14 | 64 bytes |  | 1 | 1 | 14 | 64 btyes |
| Valid | Dirty | Tag | Data |  | Valid | Dirty | Tag | Data |
| 1 | 1 | 14 | 64 bytes |  | 1 | 1 | 14 | 64 btyes |

1 + 1 + 14 + 8 = 80/160 bits per row

64

14

2

80

Assignment

1)My CPU has 3 levels of caches. In my CPU, L1 is separated for data and instructions.

2) L1 has a maximum size of 32KB and L2 cache has a maximum size of 256KB. My L3 cache has a maximum size of 3 MB

3) For all cache levels, the line size ( block size) is 64B

4) Majority of the cache are direct mapped , but my L3 cache is labeled as complex. It also states that there is the possibility that my machine can have 4-way and 8-way set associative cache possible.

5) tag bits -> 33- 1 – 5 = 27 bits tag

- - - - - - - - - - - - - - - - - - - - - - - - - - -- - - - - - - - - - -

Index bits -> 2 bits of index

NUM\_SETS = 32Kb/ 8 = 4 sets

Log2(4) = 2 bits of index

- - - - - - - - - - - - - - - - - - - - - - - - - - -- - - - - - - - - - -

Offset bits -> log2(64) = 5 bits offset