

This picture explains our thought process in designing the 8-way set associative cache controller. Instead of the 4-way terminating at the third layer, we add an additional 4 bits check to account for doubling 4-way to 8-way.

Basically the functionality is the same as 4-way, we check to see if the bits are 0 or 1, and set the bits according to the concatenated values in the brackets.

If there’s a cache hit, we update LRUBits\_Out to the concatenated values in the diagram. If there isn’t a cache hit, we generate a burst fill and update the LRUbits.

|  |  |
| --- | --- |
| Line LRU State Bits [6:0] | Replace data in this block: |
| xxx0x00 | 0 |
| xxx1x00 | 1 |
| xx0xx10 | 2 |
| xx1xx10 | 3 |
| x0xx0x1 | 4 |
| x1xx0x1 | 5 |
| 0xxx1x1 | 6 |
| 1xxx1x1 | 7 |

|  |  |
| --- | --- |
| If a read or write (load) access to a line in this block occurs | Set LRU State Bits [6:0] for the line to |
| 0 | \_ \_ \_ 1 \_ 1 1 |
| 1 | \_ \_ \_ 0 \_ 1 1 |
| 2 | \_ \_ 1 \_ \_ 0 1 |
| 3 | \_ \_ 0 \_ \_ 0 1 |
| 4 | \_ 1 \_ \_ 1 \_ 0 |
| 5 | \_ 0 \_ \_ 1 \_ 0 |
| 6 | 1 \_ \_ \_ 0 \_ 0 |
| 7 | 0 \_ \_ \_ 0 \_ 0 |

Text

Description automatically generated