计算机体系结构



**cache模拟实验报告**

|  |  |
| --- | --- |
| 学 院： | 计算机与信息技术学院 |
| 专 业： | 计算机科学与技术 |
| 指导教师： | 吴为民 |
| 班 级： | 计科1602 |
| 姓 名： | 麻锦涛 |
| 学 号： | 16281262 |

2019年 6月 8日

1. **实验目的**

**熟悉 cache 系统的运行过程。**

1. **实验内容**

1）在 C++环境下运行 cache 系统模拟程序 cache2018.c，也可在命令行环境下直接运行可执行文件 cache2018.exe。

**介绍：cache2018.c 是我编的一个小程序，用来展示 4-路组相连 cache 的运行结果。可参见书 4-35 页上的图 4.19。为便于实验，采用了与书图上不同的数字：**

**Tag 位数为 3，而不是 22.**

**Index 位数采用了 3，而不是 8。因此组数为 23=8.**

**字节偏移位数为 2。一个块的大小是一个字，即 4 个字节.**

**总的地址空间的位数为 8，而不是书图中的 32 位。因此，存储器中总的块数为 2（8-2）=64. Cache 中的总的块数为：组数×相联数=8×4=32，是存储器中总的块数的一半。运行后，程序随机产生出 40 次访问的存储器块号（在 0-63 之间），并计算命中次数、不中次数、以及替换次数。程序采用的是最近最少使用(LRU)策略。**

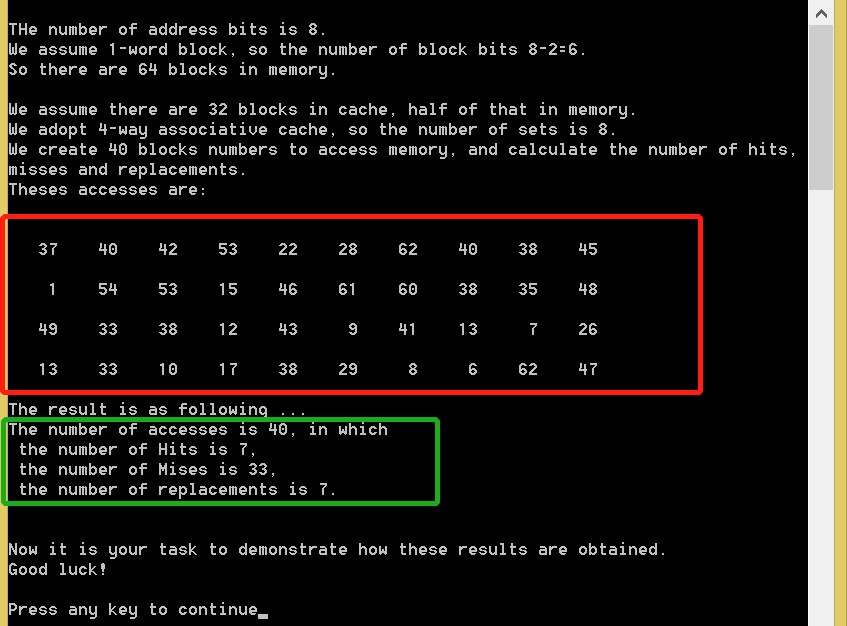
2）cache 运行情况跟踪。

请根据软件随机产生的 40 个连续访问的块号，一步一步地跟踪 cache 系统的执行过程，记录各个地址来了以后，cache 的变化情况。最后，核实一下你的最终结果是否与程序给出的结果相同。

**3. 实验过程及结果**

**1）在 C++环境下运行 cache 系统模拟程序 cache2018.cpp,运行结果如下：**

红色框内为随机产生的连续访问的块号，绿色框内为计算结果包括：命中次数、缺失次数和置换次数。

 执行过程如下：组相连**:** 组号 **=** 块号 **mod cache** 中的组数

**2）cache 运行情况跟踪。**

1. 当 CPU 要取存储器的第 37 块时：

由 37 mod 8 = 5 可知该块需要在 cache 的第 5 组中查找，此时 tag = 100, 因为初始时 cache 为空，所以 cache 中所有的块的 valid 位均为无效，发生缺失，将存储器中的37掉入cache中的任意一路的第五组的位置即可，如下表所示，将其调入第 0 路：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 |  |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 |  |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 |  |  |  |  |
| 5 | 101 | 37 |  |  |  |
| 6 | 110 |  |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **1** | 0 | 0 |

1. 当 CPU 要取存储器的第 40 块时：

由 40 mod 8 = 0 可知该块需要在第 0 组中查找，此时 tag = 101，因为当前第 0 组内没有数据块调入，所以 4 路的 valid 位均为无效，发生缺失，所以将第 40 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 |  |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 |  |  |  |  |
| 5 | 101 | 37 |  |  |  |
| 6 | 110 |  |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **2** | 0 | 0 |

1. 当 CPU 要取存储器的第 42 块时：

由 42 mod 8 = 2 可知该块需要在第 2 组中查找，此时 tag = 101，因为当前第 2 组内没有数据块调入，所以 4 路的 valid 位均为无效，发生缺失，所以将第 42 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 |  |  |  |  |
| 5 | 101 | 37 |  |  |  |
| 6 | 110 |  |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **3** | 0 | 0 |

1. 当 CPU 要取存储器的第 53 块时：

由 53 mod 8 = 5 可知该块需要在第 5 组中查找，此时 tag = 110，当前第 5 组内第一路的 valid 为有效，但是其 tag 为 100，与当前查找的 tag 不相同，所以未找到，发生缺失，所以将第 53 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 |  |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 |  |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **4** | 0 | 0 |

1. 当 CPU 要取存储器的第 22 块时：

由 22 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 010，因为当前第 2 组的 4 路 cache 的 valid 位均为无效，发生缺失，所以将第 22 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 |  |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 | 22 |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **5** | 0 | 0 |

1. 当 CPU 要取存储器的第 28 块时：

由 28 mod 8 = 4 可知该块需要在第 4 组中查找，tag = 011，因为当前第 4 组的

4 路 cache 的 valid 均为无效，发生缺失，所以将第 28 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 | 22 |  |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **6** | 0 | 0 |

1. 当 CPU 要取存储器的第 62 块时：

由 62 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 111，因为第 6 组中只有第 0 路的位置 valid 有效，其 tag = 010，与当前查找的 tag 不相等，因此发生缺失，将第 62 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 | 22 | 62 |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **7** | 0 | 0 |

1. 当 CPU 要取存储器的第 40 块时：

由 40 mod 8 = 0 可知该块需要在第 0 组中查找，tag = 101，因为第 0 组中只有第 0 路的位置 valid 有效，其 tag = 101，与当前查找的 tag 相等，所以命中，因为当前第 0 组中只有一块数据块，所以无需调整用于 LRU 置换的指针，下表中红色标记的为 Hit 的块 40：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 | 22 | 62 |  |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **7** | 1 | 0 |

1. 当 CPU 要取存储器的第 38 块时：

由 38 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 100，因为第 6 组中只有第 1 路和第 2 路的 valid 有效，tag 的值分别为 010 和 111，没有与当前查找的 tag 相同的块，所以缺失，将第 38 块调入第 2 路中，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 |  |  |
| 6 | 110 | 22 | 62 | 38 |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **8** | 1 | 0 |

1. 当 CPU 要取存储器的第 45 块时：

由 45 mod 8 = 5 可知块需要在第 5 组中查找，tag = 101，因为第 5 组中只有第

1 路和第 2 路的 valid 有效，tag 的值分别为 100 和 110，没有与当前查找的 tag 相同的块，所以缺失，将第 45 块调入第 2 路中，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 |  |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **9** | 1 | 0 |

1. 当 CPU 要取存储器的第 1 块时：

由 1 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 000，因为第 0 组中 4 路 cache 的 valid 位均为无效，发生缺失，将第 1 块调入第 0 路中，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 |  |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **10** | 1 | 0 |

1. 当 CPU 要取存储器的第 54 块时：

由 54 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 110，因为第 6 组的 0、

1、2 路的 valid 位有效，tag 值分别为 010、111 和 100，没有与当前查找的块相同的 tag，所以发生缺失，将第 54 块调入第 3 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 | 54 |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **11** | 1 | 0 |

1. 当 CPU 要取存储器的第 53 块时：

由 53 mod 8 = 5 可知该块需要在第 5 组中查找，tag = 110，因为第 5 组中 0、

1、2 路的 cache 的 valid 位有效，tag 值分别为：100、110 和 101，存在与查找的 tag 值相等的块，所以命中，下表中红色标记的为 Hit 的块 53：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 | 54 |
| 7 | 111 |  |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **11** | 2 | 0 |

1. 当 CPU 要取存储器的第 15 块时：

由 15 mod 8 = 7 可知该块需要在第 7 组中查找，tag = 001,因为第 7 组的 4 路 cache 的 valid 位无效，所以发生缺失，将第 15 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 |  |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **12** | 2 | 0 |

1. 当 CPU 要存储器的第 46 块时：

由 46 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 101，因为第 6 组中 4 路 cache 均有效，tag 值分别为 010、111、100 和 110，没有与查找的块的 tag 相同，但因为第 6 组的 cache 已满，所以需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 22 换出，将块 46 调入第 6 组第 0 路的位置，并且调整用于 LRU 的指针，第 6 组对应的指针为：62->38->54->46 （队首元素为最近最久未使用的块）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 | 48 |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 22 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 | 48 |  |  |
| 5 | 101 | 37 | 53 | 45 |  |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **13** | 2 | 1 |

1. 当 CPU 要取存储器的第 61 块时：

由 61 mod 8 = 5 可知该块需要在第 5 组中查找，tag = 111，因为第 5 组中 0、1、 2 路 cache 均有效，tag 值分别为 100、110 和 101，没有与查找的块的 tag 相同，所以发生缺失，因此将第 61 块调入第 3 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 | 48 |  |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **14** | 2 | 1 |

1. 当 CPU 要取存储器的第 60 块时：

由 60 mod 8 = 4 可知该块需要在第 4 组中查找，tag = 111，因为第 4 组的 0、1 路的 valid 均为有效，tag 值分别为：011、110，没有与查找的块相同的 tag，所以发生缺失，因此将第 60 块调入第 2 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **15** | 2 | 1 |

1. 当 CPU 要取存储器的第 38 块时：

由 38 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 100，因为第六组中 4 路 cache 的 valid 位均有效，tag 值分别为 101、111、100 和 110，能够找到与查找的块相同的 tag，因此命中，下表中已用红色标注，并且调整用于 LRU 的指针，第 6 组对应的指针为：62->54->46->38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 |  |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **15** | 3 | 1 |

1. 当 CPU 要取存储器的第 35 块时：

由 35 mod 8 = 3 可知该块需要在第三组中查找，tag = 100，第三组中 4 路 valid 均无效，所以发生缺失，将第 35 块调入第 0 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 |  |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **16** | 3 | 1 |

1. 当 CPU 要取存储器的第 48 块时：

由 48 mod 8 = 0 可知该块需要在第 0 组中查找，tag = 110，第 0 组中只有第 0 路的 valid 有效，tag = 101，与查找块的 tag 不相等，所以发生缺失，将第 48 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 |  |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **17** | 3 | 1 |

1. 当 CPU 要取存储器的第 49 块时：

由 49 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 110，第 1 组中只有第 0 路的 valid 有效，tag 值为 000，与查找块的 tag 值不相同，所以发生缺失，将第 49 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 |  |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **18** | 3 | 1 |

1. 当 CPU 要取存储器的第 33 块时：

由 33 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 100，第一组中第 0、1 路 valid 位有效，tag 值分别为 000 和 110，与查找块的 tag 值不相等，所以发生缺失，将第 33 块调入第 2 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **19** | 3 | 1 |

1. 当 CPU 要取存储器的第 38 块时：

由 38 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 100，因为第六组中 4 路 cache 的 valid 位均有效，tag 值分别为 101、111、100 和 110，能够找到与查找的块相同的 tag，因此命中，下表中已用红色标注，因为当前的用于 LRU 的第 6 组对应的指针为：62->54->46->38，因此不需要调整。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 |  |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **19** | 4 | 1 |

1. 当 CPU 要取存储器的第 12 块时：

由 12 mod 8 = 4 可知该块需要在第 4 组中查找，tag = 001，第 4 组中的第 0、

1、2 路的 valid 位有效，tag 值分别为 100、110 和 111，没有找到与查找块相同的 tag，因此发生缺失，将第 12 块调入第 3 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 |  |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **20** | 4 | 1 |

1. 当 CPU 要取存储器的第 43 块时：

由 43 mod 8 = 3 可知该块需要在第 3 组中查找，tag = 101，第 3 组中只有第 0 路的 valid 位是有效的，tag 值为 100，与查找块的 tag 值不想等，所以发生缺失，将第 43 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 |  |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **21** | 4 | 1 |

1. 当 CPU 要取存储器的第 9 块时：

由 9 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 001，第 1 组中的第 0、1、 2 路的 valid 位有效，tag 值分别为 000、110、100，均与查找块的 tag 值不相同，所以发生缺失，将第 9 块调入第 3 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **22** | 4 | 1 |

1. 当 CPU 要取存储器的第 41 块时：

由 41 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 101，第 1 组中 4 路 cache 的 valid 位均有效，tag 值分别为：000、110、100 和 001，与查找块的 tag 值均不相同，且因为 4 路 cache 已满，需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 1 换出，将块 41 调入第 1 组第 0 路的位置，并且调整用于 LRU 的指针，第 1 组对应的指针为：49->33->9->41.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 1 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **23** | 4 | 2 |

1. 当 CPU 要取存储器的第 13 块时：由 13 mod 8 = 5 可知该块需要在第 5 组中查找，tag = 001，第 5 组中 4 路 cache 的 valid 位均有效，tag 值分别为：100、110、101 和 111，与查找块的 tag 值均不相同，且因为 4 路 cache 已满，需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 37 换出，将块 13 调入第 5 组第 0 路的位置，并且调整用于 LRU 的指针，第 5 组对应的指针为：53->45->61->13.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 37 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 |  |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **24** | 4 | 3 |

1. 当 CPU 要取存储器的第 7 块时：

由 7 mod 8 = 7 可知该块需要在第 7 组中查找，tag = 000，第 7 组中只有第 0 路的 valid 位有效，tag 值为 001，与查找块的 tag 值不同，所以发生缺失，将第 7 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 |  |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **25** | 4 | 3 |

1. 当 CPU 要取存储器的第 26 块时：

由 26 mod 8 = 2 可知该块需要在第 2 组中查找，tag = 010，由于第 2 组中 valid 有效的第 0 路的 tag = 101，与查找块的 tag 值不相同，所以发生缺失，将第

26 块调入第 1 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 | 26 |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **26** | 4 | 3 |

1. 当 CPU 要取存储器的第 13 块时：

由 13 mod 8 = 5 可知该块在第 5 组 cache 中查找，tag = 001，由于第 5 组的第

0 路的 valid 位有效且 tag = 001，所以命中，并且需要修改第 5 组对应的用于

LRU 的指针，变为：53->45->61->13.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 | 26 |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **26** | 5 | 3 |

1. 当 CPU 要取存储器的第 33 块时：

由 33 mod 8 = 1 可知该块需要在第 1 组中查找，由于第 1 组 valid 有效的 tag 值分别为 101、110、100 和 001，所以含有与查找块 tag 相同的值，所以命中，并且需要修改第 1 组对应用于 LRU 的指针，变为：49->9->41->33.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 | 26 |  |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **26** | 6 | 3 |

1. 当 CPU 要取存储器的第 10 块时：

由 10 mod 8 = 2 可知该块需要在第 2 组中查找，由于第 2 组中 valid 有效的 tag 值分别为 101 和 011，没有与查找块 tag 相同的值，所以发生缺失，将第 10 块调入第 2 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **27** | 6 | 3 |

1. 当 CPU 要取存储器的第 17 块时：

由 17 mod 8 = 1 可知该块需要在第 1 组中查找，tag = 010，因为第 1 组中 4 路 cache 均有效，tag 值分别为 101、110、100 和 001，没有与查找的块的 tag 相同，但因为第 1 组的 cache 已满，所以需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 49 换出，将块 17 调入第 1 组第 1 路的位置，并且调整用于 LRU 的指针，第 1 组对应的指针为：9->41->33->17.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 49 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **28** | 6 | 4 |

1. 当 CPU 要取存储器的第 38 块时：

由 38 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 100，由于第 6 组中 valid 有效的 tag 值分别为 101、111、100 和 110，所以命中，并且修改第 6 组对应

的用于 LRU 的指针，变为：62->54->46->38.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **28** | 7 | 4 |

1. 当 CPU 要取存储器的第 29 块时：

由 29 mod 8 = 5 可知该块需要在第 5 组中查找，tag = 011，因为第 5 组中 4 路 cache 均有效，tag 值分别为 001、110、101 和 111，没有与查找的块的 tag 相同，但因为第 5 组的 cache 已满，所以需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 53 换出，将块 29 调入第 5 组第 1 路的位置，并且调整用于 LRU 的指针，第 5 组对应的指针为：45->61->13->29.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 53 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 |  |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **29** | 7 | 5 |

1. 当 CPU 要取存储器的第 8 块时：

由 8 mod 8 = 0 可知该块需要在第 0 组中查找，tag = 000，由于第 0 组中 valid 有效的两路 cache 的 tag 值均不等于 000，所以发生缺失，第 8 块调入第 2 路的位置，如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **30** | 7 | 5 |

1. 当 CPU 要取存储器的第 6 块时：

由 6 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 000，因为第 6 组中 4 路 cache 均有效，tag 值分别为 101、111、100 和 110，没有与查找的块的 tag 相同，但因为第 6 组的 cache 已满，所以需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 62 换出，将块 6 调入第 6 组第 1 路的位置，并且调整用于 LRU 的指针，第 6 组对应的指针为：54->46->38->6.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 62 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 6 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **31** | 7 | 6 |

1. 当 CPU 要取存储器的第 62 块时：

由 62 mod 8 = 6 可知该块需要在第 6 组中查找，tag = 111，因为第 6 组中 4 路 cache 均有效，tag 值分别为 101、000、100 和 110，没有与查找的块的 tag 相同，但因为第 6 组的 cache 已满，所以需要发生置换，根据 LRU 的原则，选择将最近最久未使用的块 54 换出，将块 62 调入第 6 组第 3 路的位置，并且调整用于 LRU 的指针，第 6 组对应的指针为：46->38->6->62.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 6 | 38 | 54 |
| 7 | 111 | 15 | 7 |  |  |

变为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 6 | 38 | 62 |
| 7 | 111 | 15 | 7 |  |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **32** | 7 | 7 |

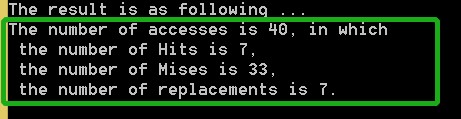
1. 当 CPU 要取存储器的第 47 块时：由 47 mod 8 = 7 可知该块需要在第 7 组中查找，tag = 101，第 7 组中 0、1 路的 valid 有效，tag 值为 001 和 000，查找块的 tag 值不相同，所以发生缺失，将第 47 块调入第 2 路的位置如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号 | index | 第0路 | 第1路 | 第2路 | 第3路 |
| 0 | 000 | 40 | 48 | 8 |  |
| 1 | 001 | 41 | 17 | 33 | 9 |
| 2 | 010 | 42 | 26 | 10 |  |
| 3 | 011 | 35 | 43 |  |  |
| 4 | 100 | 28 | 48 | 60 | 12 |
| 5 | 101 | 13 | 29 | 45 | 61 |
| 6 | 110 | 46 | 6 | 38 | 62 |
| 7 | 111 | 15 | 7 | 47 |  |

此时

|  |  |  |
| --- | --- | --- |
| 缺失次数 | 命中次数 | 置换次数 |
| **33** | 7 | 7 |
|  |  |  |
| 缺失次数 | 命中次数 | 置换次数 |
| **33** | 7 | 7 |

**实验结果：**



**当40次访问结束后，命中次数为7、缺失次数为33、置换次数为7；与程序计算结果一致。**

**本学期的体系结构实验就到此结束了，六次实验中都让我收获颇多，非常感谢吴为民对我们专业班同学的指导！**