缓存确定一个请求是否命中，然后抽取出被请求的字的过程，分为三步：1）组选择；2）行匹配；3）字抽取。

1. 直接映射高速缓存中的组选择

在这一步中，高速缓存从 w 的地址中间抽取出 s 个组索引位。这些位被解释成一个对应于一个组号的无符号整数。换句话来说，如果我们把高速缓存看成是一个关于组的一维数组，那么这些组索引位就是一个到这个数组的索引。图 6-28 展示了直接映射高速缓存的组选择是如何工作的。在这个例子中，组索引位 00001_2 被解释为一个选择组 1 的整数索引。

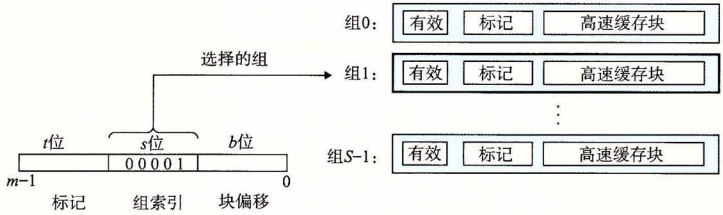

图 6-28　直接映射高速缓存中的组选择

2. 直接映射高速缓存中的行匹配

在上一步中我们已经选择了某个组 i，接下来的一步就要确定是否有字 w 的一个副本存储在组 i 包含的一个高速缓存行中。在直接映射高速缓存中这很容易，而且很快，这是因为每个组只有一行。当且仅当设置了有效位，而且高速缓存行中的标记与 w 的地址中的标记相匹配时，这一行中包含 w 的一个副本。

图 6-29 展示了直接映射高速缓存中行匹配是如何工作的。在这个例子中，选中的组中只有一个高速缓存行。这个行的有效位设置了，所以我们知道标记和块中的位是有意义的。因为这个高速缓存行中的标记位与地址中的标记位相匹配，所以我们知道我们想要的那个字的一个副本确实存储在这个行中。换句话说，我们得到一个缓存命中。另一方面，如果有效位没有设置，或者标记不相匹配，那么我们就得到一个缓存不命中。

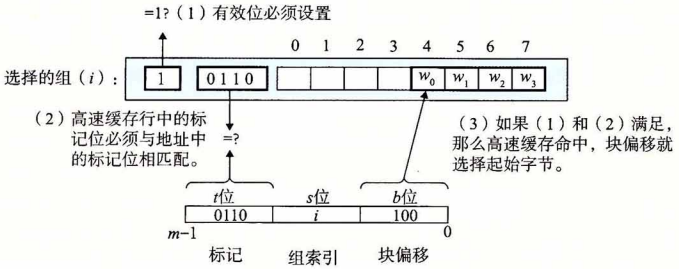

图 6-29　直接映射高速缓存中的行匹配和字选择。在高速缓存块中，w_0 表示字 w 的低位字节，w_1 是下一个字节，依此类推

3. 直接映射高速缓存中的字选择

一旦命中，我们知道 w 就在这个块中的某个地方。最后一步确定所需要的字在块中是从哪里开始的。如图 6-29 所示，块偏移位提供了所需要的字的第一个字节的偏移。就像我们把高速缓存看成一个行的数组一样，我们把块看成一个字节的数组，而字节偏移是到