


的数组内容。这样对高速缓存的使用效率很低。相比较而言，以中间位作为索引，相邻的块总是映射到不同的高速缓存行。在这里的情况中，高速缓存能够存放整个大小为 C 的数组片，这里 C 是高速缓存的大小。

 **练习题 6.11** 假想一个高速缓存，用地址的高 s 位做组索引，那么内存块连续的片(chunk)会被映射到同一个高速缓存组。

- A. 每个这样的连续的数组片中有多少个块？
- B. 考虑下面的代码，它运行在一个高速缓存形式为 $(S, E, B, m) = (512, 1, 32, 32)$ 的系统上：

```
int array[4096];
for (i = 0; i < 4096; i++)
    sum += array[i];
```

在任意时刻，存储在高速缓存中的数组块的最大数量为多少？

6.4.3 组相联高速缓存

直接映射高速缓存中冲突不命中造成的问题源于每个组只有一行(或者，按照我们的术语来描述就是 $E=1$)这个限制。组相联高速缓存(set associative cache)放松了这条限制，所以每个组都保存有多于一个的高速缓存行。一个 $1 < E < C/B$ 的高速缓存通常称为 E 路组相联高速缓存。在下一节中，我们会讨论 $E=C/B$ 这种特殊情况。图 6-32 展示了一个 2 路组相联高速缓存的结构。

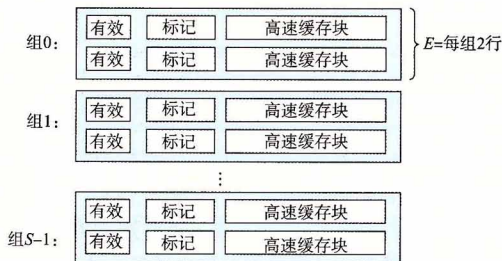


图 6-32 组相联高速缓存($1 < E < C/B$)。在一个组相联高速缓存中，每个组包含多于一个行。
这里的特例是一个 2 路组相联高速缓存

1. 组相联高速缓存中的组选择

它的组选择与直接映射高速缓存的组选择一样，组索引位标识组。图 6-33 总结了这个原理。

2. 组相联高速缓存中的行匹配和字选择

组相联高速缓存中的行匹配比直接映射高速缓存中的更复杂，因为它必须检查多个行的标记位和有效位，以确定所请求的字是否在集合中。传统的内存是一个值的数组，以地址作为输入，并返回存储在那个地址的值。另一方面，相联存储器是一个(key, value)对的数组，以 key 为输入，返回与输入的 key 相匹配的(key, value)对中的 value 值。因此，我们可以把组相联高速缓存中的每个组都看成一个小的相联存储器，key 是标记和有效位，而 value 就是块的内容。