

1.Direct mapped caches

这个缓存 keyi 存储多少字节数据？存储多少字？按照下图地址分解成“Tag bits, Index bits, Offset bits”填入下表

CPU Cache	Index Number	Offset				Tag bits	Index bits	Offset bits	Total
		3	2	1	0				
	0								32
	1								

Index bits=log₂ (Number of index rows) Offset bits=log₂ (Number of offsets columns)

8 bytes 2 words

Tag	Index	Offset bits	Total
29	1	2	32

2. 有一个容量 8192KiB 的缓存，每块存 128B 数据，地址 0xFEEDF00D 的标识位、索引位、偏移位各为多少？

FE	ED	F0	0D
1111 1110	1110 1101	1111 0000	0000 1101

Tag: **111111101 (0x1FD)** Index: **1101101111100000 (0xDBE0)** Offset: **0001101 (0x0D)**

3.缓存每行包含数据块、标识(tag)和一位有效位（valid），补全下表数据.

Address size (bits)	Cache size	Block size	Tag bits	Index bits	Offset bits	Bits per row
16	4KiB	4B				
32	32KiB	16B				
32			16	12		
64	2048KiB			14		1068

Address size (bits)	Cache size	Block size	Tag bits	Index bits	Offset bits	Bits per row
16	4KiB	4B	4	10	2	32+4+1
32	32KiB	16B	17	11	4	128+17+1
32	64KiB	16B	16	12	4	128+16+1
64	2048KiB	128B	43	14	7	1068

4. If 80386 used Direct Cache Mapping :

以字节编址 32 位的内存 MM = 4GB (2^{32}), 缓存容量(Cache Size = 64KB (2^{16})), 字块大小 4 个字 (Block Size = ($16=2^4$) bytes = 4 words), 问内存以块划分, 一共多少块, 缓存一共有多少行, 每行包含数据块、标识(tag) 和一位有效位 (valid), 一共多少位?

内存块数 Number of blocks in MM = $2^{32} / 2^4 = 2^{28}$

缓存行数 Number of Cache Block Frames = $2^{16} / 2^4 = 2^{12} = 4096$, index 位=12

index 位=> 12 Block field bits, tag=32-12-4=16

4*32+16+1=145 位

5. 考虑下列两组代码完成同样的计算.哪个代码对缓存性能更好,为什么?

Code 1:

```
for (j = 0; j < 4; j++)  
  
  for(i=0;i<4;i++)  
    x[i][j]=10*y[i][j];
```

Code 2:

```
for (i = 0; i < 4; i++)  
  for(j=0;j<4;j++)  
    x[i][j]=10*y[i][j];
```