5.3

5.3.1

Cache's capacity $= 32 \cdot 2^{10} = 2^{15}$ bytes

One block $= 2 \cdot 2^{3} = 2^{4}$ bytes

Number of blocks $= \frac{2^{15}}{2^{4}} = 2^{11}$

Tag	Num of Blocks (Index)	byte offset
49	11	4

Total 64 bits

Total number of bits $= (1 + 49 + 128) \cdot 2^{11} = 178 \times 2^{11}$

(1: Valid bit; 49: Tag; 128: two-words data; 2^{11}: blocks number)

5.3.2

Cache's capacity $= 64 \cdot 2^{10} = 2^{16}$ bytes

One block $= 2^{4} \cdot 2^{3} = 2^{7}$ bytes

Number of blocks $= \frac{2^{16}}{2^{7}} = 2^{9}$

Tag	Num of Blocks	byte offset
48	9	7

64 bits

Total number of bits $= (1 + 48 + 16 \times 64) \cdot 2^{9} = 1073 \times 2^{9}$

(1: Valid bit; 48: Tag; 16×64: 16-words data; 2^{9}: blocks number)

How much bigger? $\frac{1073 \times 2^{9}}{178 \times 2^{11}} \approx 1.507 < 2$

5.3.3

由于5.3.2中的Cache每个block为16个words，因此虽然该Cache拥有更大的存储空间，但是它在一个block中对word的寻址，以及在单个block传输过程中可能有更大的时间延迟。因此5.3.2中的Cache可能性能表现不如前一个Cache。

5.3.4

32 KiB two-way set associate cache ②　　5.3.1 ①

Tag	index	byte offset
50	10	4

64 bits

Read request:

	①:		②: Index	
0x 61	miss	Index 0x6 [0x61]	miss 0x6	[0x61]
0x 10061	miss	0x6 [0x10061]	miss 0x6	[0x61 / 0x10061]
0x 61	miss	0x6 [0x61]	hit 0x6	[0x61 / 0x10061]

三者 index 均相等

5. 4.

64 bits

[63:54]	[53:44]	m	offset

total blocks = 2^{20+m}

possible

共有 2^{10} 种运算结果 ⇒ 1024 Blocks

对于任意一个 Index ⇒ $2^{10} \cdot 2^{m} = 2^{10+m}$ Blocks

这样每个 Index 映射到的内存中 Block Address 都是平均的。

Tag 变为 m + 10 bits