

6. (1) 32位为4字节

64K - $4 \times m$

$m = 16K$ 它的寻址范围是 16K

其存储容量按字编址是 16K字，按字节编址是 64K
字编址其寻址范围是从 0 到 16383

(2) 按字节编址，其寻址范围是 64K

小端方式

高字节—字节地址—低字节				字地址
3	2	1	0	0
7	6	5	4	4
...	8
65535	65534	65533	65532	65528

(3) 按字编址：

优点：地址空间小，因为一个地址对应一个存储字，所需地址线数量少，硬件设计相对简单，CPU访问主存时，一次就能获取一个完整的存储字

缺点：无法访问单个字节，灵活性较差，不便处理字节级的数据。

按字节编址：

优点：灵活性较高，对于不同长度的数据都能灵活寻址和操作

缺点：地址空间大，需要更多的地址线来表示地址，硬件设计相对复杂

$$7. (1) 2^{20} \times 4 = 4MB = 4096K$$

$$(2) N = 1M \times 32 / (512K \times 8) = 8$$

(3)

地址线 20 位，低 19 位作为芯片地址输入；
最高 1 位用于芯片选择

12.

$$t = 8000 \mu s \div 8192 = 0.9765625 \mu s \approx 0.9 \mu s$$

刷新需要 8K 个刷新周期

19. (1) Hello 的 ASCII 码为 72, 101, 108, 111

H: 48H E: 65H L: 6CH O: 6FH

100: 64H 200: C8H 300: 012CH

400: 0190H

(1)

字节地址 3

2

1

0

字地址

...				
6CH	6CH	65H	48H	00BD4010H
00H	00H	00H	6FH	00B04014H
00H	C8H	00H	64H	00B04018H
		01H	2CH	00B0401CH
...				

高字节

←

低字节

(2)

字节地址 3

2

2

1

0

字地址

...				
6CH	6CH	65H	48H	00BD4010H
2CH	00H	C8H	6FH	00B04014H
	01H	90H	01H	00B04018H
				00B0401CH
...				

高字节

←

低字节