

1

(1) 存取时间：读或者写操作所用的时间

存取周期：两次访问存储单元的最小时间间隔

(2) 存储器带宽：单位时间内访问的存储量

计算： $64/(200 \times 10^{-9}) = 320 \text{ Mb/s}$

2

按字编址：字长为 32 位，因此，按字编址，寻址范围就是 $64\text{K} \times 8 / 32 = 16\text{K}$ 字

按字节编址：容量为 64KB，即按字节编址，寻址范围就是 64K 字节

每个字包含 4 个字节，主存字地址和字节地址的分配情况：

字地址	字节地址			
0	0	1	2	3
4	4	5	6	7
.....
65532	65532	65533	65534	65535

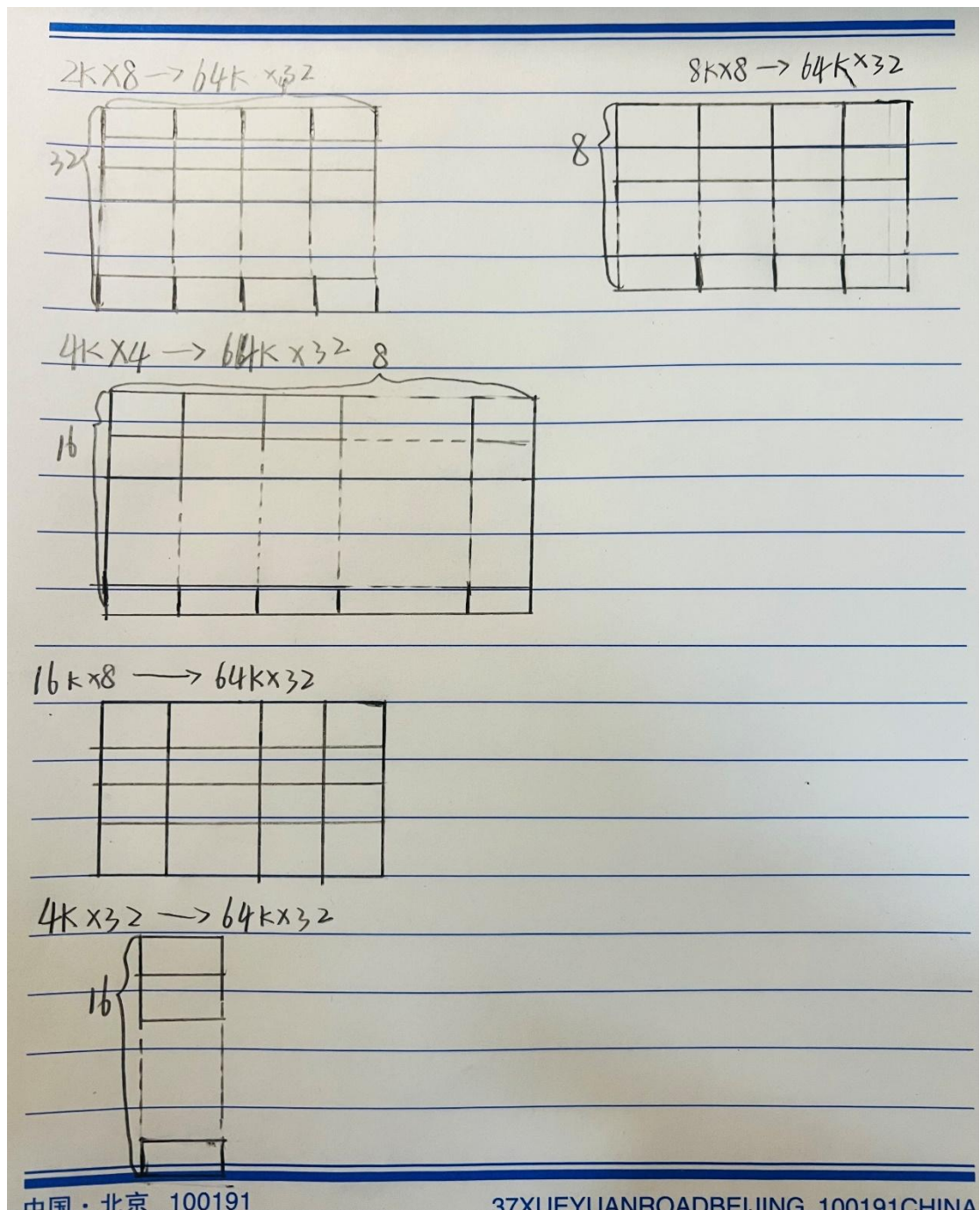
3

地址线： $64\text{K} = 2^{16}$ ，于是地址线为 16 根

数据线为 32 根

$2^x \times 2^x = 64\text{K} \rightarrow X = 8$

对应的行选择线 $2^8 = 256$ ，列选择线 $2^8 = 256$



芯片图

(2) $256K \times 8 = 2^{21}$

(3) $256K = 2^{18} \rightarrow$ 行地址为 9 位 \rightarrow 刷新地址计数器为 9 位

(4)

集中刷新：在刷新周期的某一时间段集中刷新所有行，因此，刷新的间隔时间即为刷新周期，一般取 2ms

分散刷新：刷新分散到每个存取周期，每个存取周期刷新一行。

故刷新的间隔时间为 $512 \times 0.25\mu s = 128\mu s$

异步刷新：只要保证在一个刷新周期内将存储芯片所有行刷新一遍即可。因此，刷新的间隔时间仍为刷新周期，一般取 2ms

5

1K \times 4 芯片组成 64K \times 8 存储器，需要进行字扩展和位扩展

字扩展： $64K/1K = 64$

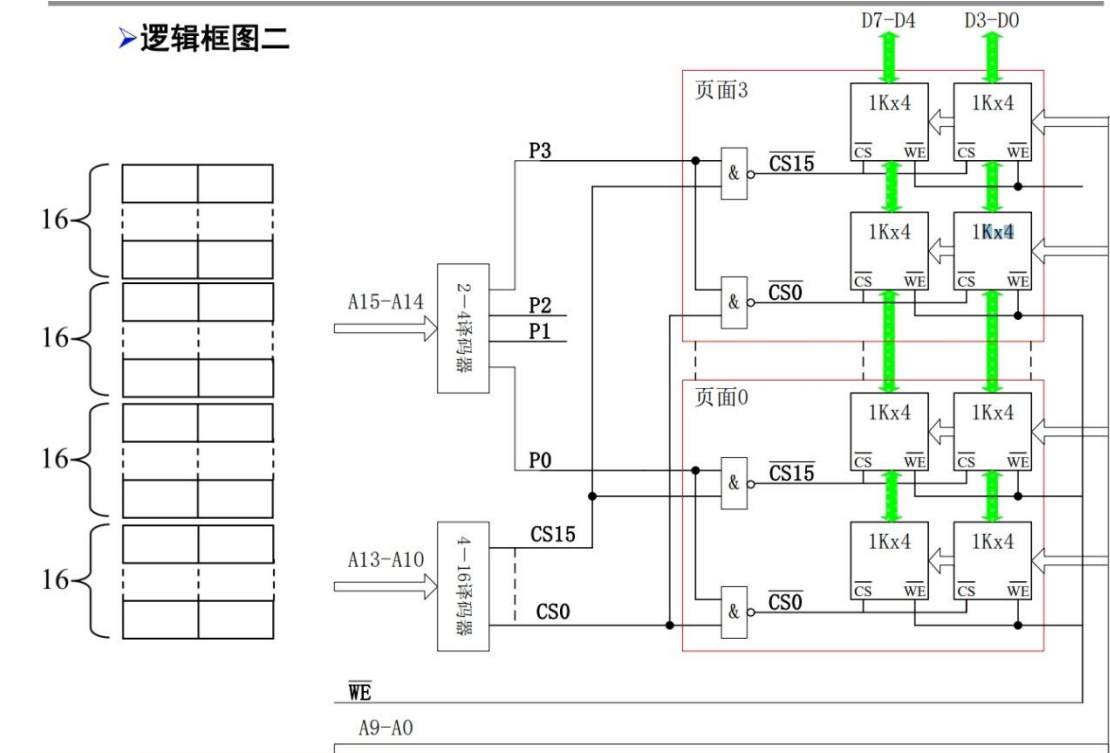
位扩展： $8/4 = 2$

芯片数： $(64K \times 8) / (1K \times 4) = 128$

将 64K 字空间分为 4 个页面

整个存储器分成 4 个 16K \times 8 的小存储器

➤逻辑框图二



逻辑框图

6

该芯片的存储基元总数=64K×16 位=1024K=220(个)

如要满足地址线 and 数据线总和最小，应尽量把存储元安排在字向，因为地址位数和字数成 2 的幂的关系，可较好地压缩线数

设地址为 n 位，数据为 b 位，则： $2^n \times b = 220$ ，即： $b = 2^{20-n}$

(n, b) 的组合有 (20, 1)，(19, 2)，(18, 4)，(17, 8)，……

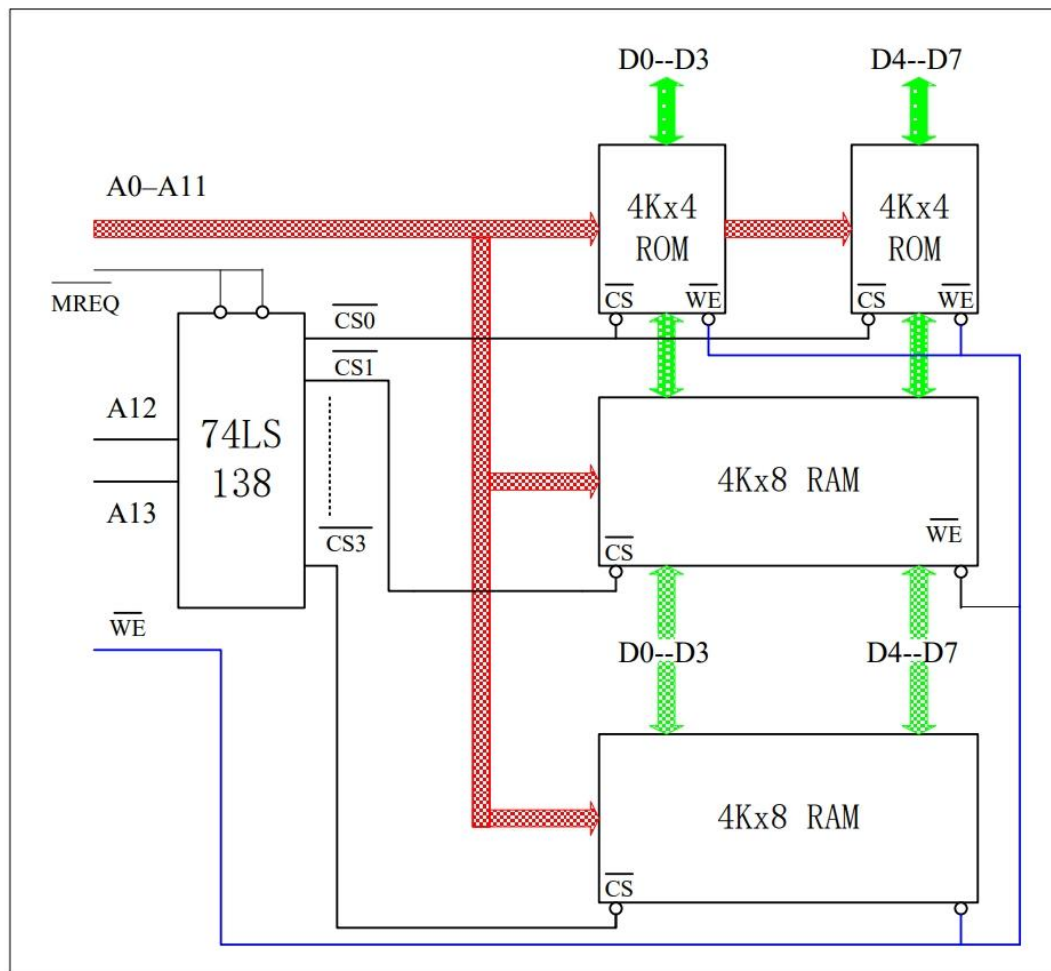
n+b 最小的组合为 (20, 1) 和 (19, 2)，所以有两种解答。

7

- 1、 由于地址码为 16 位，主存按字节编址，故系统的最大主存空间应为 216B，即 64KB。
- 2、 其中，最高 16KB 为系统的 BIOS 程序区，确定应选择 8K×8 位的 ROM 芯片 2 片；其余为用户程序区，可知用户程序区应为 64KB-16KB=48KB，则应选择 8K×8 位的 SRAM 芯片 6 片。
- 3、 主存地址 4 位；应采用 2-4 译码器

8

使用 38 译码器



片选逻辑

共需 4 个片选

0000H~0FFFH ROM(2 片 4K×4) 2 片选连接在一起

1000H~1FFFH RAM(4K×8) 独立片选

2000H~2FFFH RAM(4K×8) 独立片选

3000H~3FFFH RAM(4K×8) 独立片选

8 片 $8K \times 8$ 位的 RAM 芯片与 CPU 相连

主存储器总容量为 $64K \times 8$

CPU 的数据线为 8 位，地址线为 16 位

8 片 RAM 只能是字扩展

每片 RAM 占用 8KB 的地址空间

每片 RAM 的片选信号均为独立

存储器芯片地址空间如表所示

74LS138 译码器

CPU 地址 A15, A14, A13 对应 74LS138 的 A2, A1, A0

Y0#...Y7#对应 8 个 RAM 芯片的片选

(3)

CPU 地址 A15, A14, A13 对应 74LS138 的 A2, A1, A0

Y0#...Y7#对应 8 个 RAM 芯片的片选

以 A000H 为起始地址的存储芯片始终被选中

Y5#在写入操作过程中，恒为低电平

(4)

只能正确访问地址中 A13=1 的 RAM 芯片 1、3、5、7，而访问不到

地址中 A13=0 的 RAM 芯片 0、2、4、6