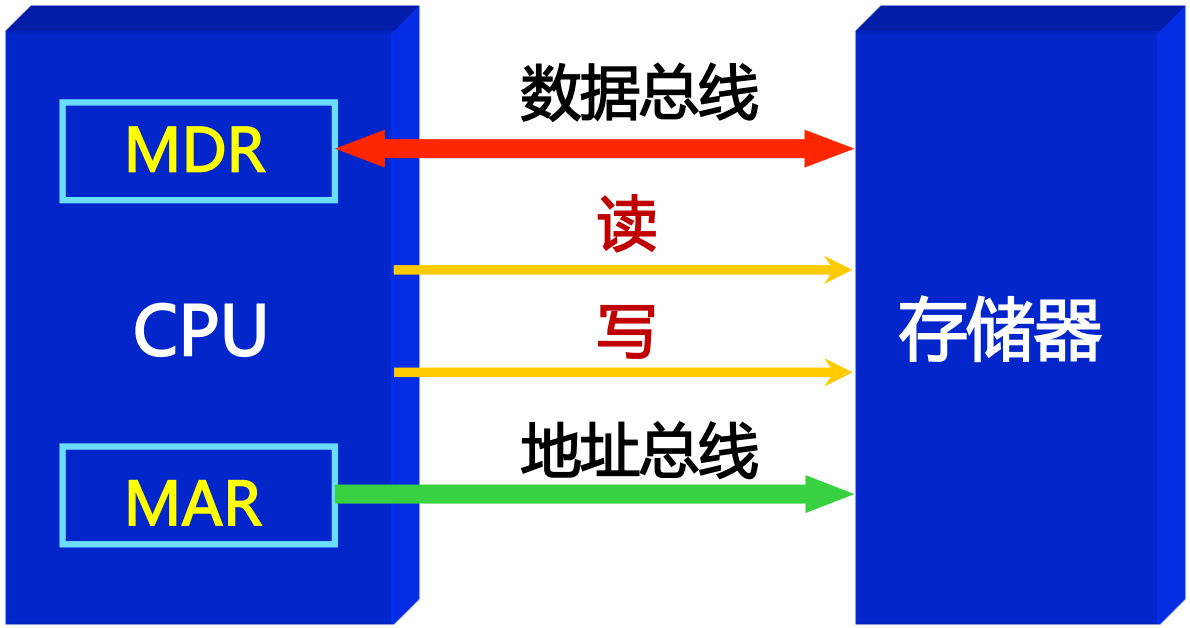
计算机原理

5.2 主存储器(II)

国防科技大学计算机学院　刘 芳

5.2.2存储器与CPU的连接

5.2.2 存储器与CPU的连接



总线连接方式

- 地址线的连接
- 数据线的连接
- 控制线的连接

总线连接方式

- 地址线的连接
- 数据线的连接
- 控制线的连接

- CPU地址线数决定了整个主存空间的寻址范围

  CPU地址线数>存储芯片地址引脚线

- 通常将CPU地址线的低位和存储芯片地址线相连，高位用作字扩展时的片选信号的译码

总线连接方式

- 地址线的连接
- 数据线的连接
- 控制线的连接

- CPU数据线数决定了一次可读写的最大数据宽度

  CPU数据线数>存储芯片数据引脚线

- 通常将CPU数据线连到多个位扩展的芯片中，使扩展后的位数与CPU数据线数相等

5.2.2 存储器与CPU的连接

总线连接方式

- 地址线的连接
- 数据线的连接
- 控制线的连接

- 若CPU读/写命令线和存储芯片的读/写控制线是一根，且电平信号一致，则可直接相连
- 若CPU读/写命令线分开，则需分别进行连接

CPU中的访存信号线$\overline{\text{MREQ}}$用来确定是访问主存还是I/O端口

（$\overline{\text{MREQ}}$信号为低电平时才选择存储芯片）

CPU和主存之间的两种通信方式

异步方式过程(需握手信号)

写操作过程类似

读操作

- CPU送地址到地址线，主存进行地址译码
- CPU发读命令，然后等待存储器发回“完成”信号
- 主存收到读命令后开始读数，完成后发“完成”信号给CPU
- CPU接收到“完成”信号，从数据线取数

CPU和主存之间的两种通信方式

同步方式的特点

- CPU和主存由统一时钟信号控制，无需应答信号
- 主存总是在确定的时间内准备好数据
- CPU送出地址和读命令后，总是在确定的时间取数据
- 存储器芯片必须支持同步方式

存储器芯片和CPU的连接举例

主存空间的划分

主存空间包括ROM和RAM区

- ROM区用来存放系统程序、标准子程序等，选ROM芯片构造；
- RAM区用来存放用户程序，选RAM芯片构造

选择存储芯片的类型和数量时，须先确定ROM区和RAM区的地址范围

存储器芯片和CPU的连接举例

- 例：设CPU有16根地址线，8根数据线，并用$\overline{\text{MREQ}}$作访存控制信号，用WR作读/写控制信号；
- 现有下列存储芯片：1K×4位RAM、4K×8位RAM、8K×8位RAM，2K×8位ROM、4K×8位ROM、8K×8位ROM及74LS138(3-8译码器)和各种门电路；
- 要求主存地址空间满足如下条件：7000H～77FFH为系统程序区；7800H～7BFFH为用户程序区；
- 试合理选择上述存储芯片，并画出CPU与存储器的连接图。

存储器芯片和CPU的连接举例

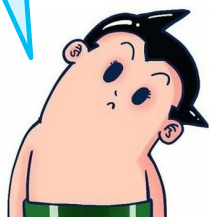
- 例：设CPU有16根地址线，8根数据线，并用$\overline{MREQ}$作访存控制信号，用WR作读/写控制信号；
- 现有下列存储芯片：1K×4位RAM、4K×8位RAM、8K×8位RAM，2K×8位ROM、4K×8位ROM、8K×8位ROM及74LS138(3-8译码器)和各种门电路；
- 要求主存地址空间满足如下条件：7000H～77FFH为系统程序区；7800H～7BFFH为用户程序区；
- 试合理选择上述存储芯片，并画出CPU与存储器的连

存储器芯片和CPU的连接举例

(1) 写出对应的二进制地址码，确定总容量

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_7	…	A_4	A_3	…	A_0	
0	1	1	1	0	0	0	0	0	0	0	0	0	0	7000H
⋮														~
0	1	1	1	0	1	1	1	1	1	1	1	1	1	77FFH
0	1	1	1	1	0	0	0	0	0	0	0	0	0	7800H
⋮														~
0	1	1	1	1	0	1	1	1	1	1	1	1	1	7BFFH

系统程序区和用户程序区的空间大小分别为多少？



存储器芯片和CPU的连接举例

(1) 写出对应的二进制地址码，确定总容量

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_7	…	A_4	A_3	…	A_0	
0	1	1	1	0	0	0	…	0	…	0	0	…	0	ROM 2K×8位
⋮														
0	1	1	1	0	1	1	…	1	…	1	1	…	1	
0	1	1	1	1	0	0	…	0	…	0	0	…	0	RAM 1K×8位
⋮														
0	1	1	1	1	0	1	…	1	…	1	1	…	1	

(2) 确定芯片的类型及数量

存储器芯片和CPU的连接举例

(1) 写出对应的二进制地址码，确定总容量

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_7	…	A_4	A_3	…	A_0	
0	1	1	1	0	0	0	…	0	…	0	0	…	0	ROM 2K×8位
⋮														
0	1	1	1	0	1	1	…	1	…	1	1	…	1	
0	1	1	1	1	0	0	…	0	…	0	0	…	0	RAM 1K×8位
⋮														
0	1	1	1	1	0	1	…	1	…	1	1	…	1	

1片 2K×8位

2片1K×4位

(2) 确定芯片的类型及数量

现有下列存储芯片：

1K × 4位RAM;

4K × 8位RAM;

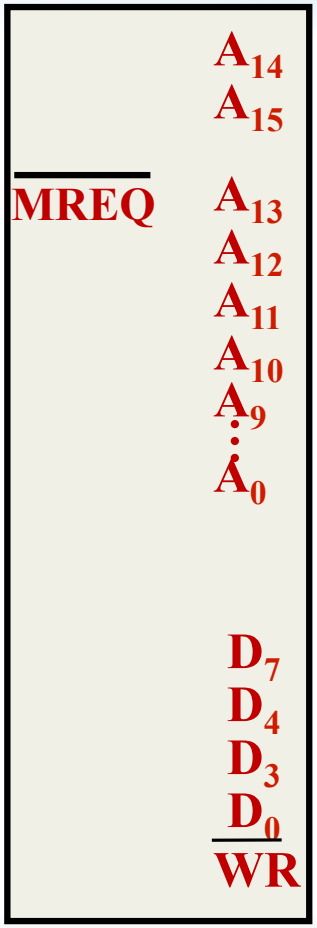
8K × 8位RAM;

2K × 8位ROM;

4K × 8位ROM;

8K × 8位ROM

5.2.2 存储器与CPU的连接



2K ×8位 ROM

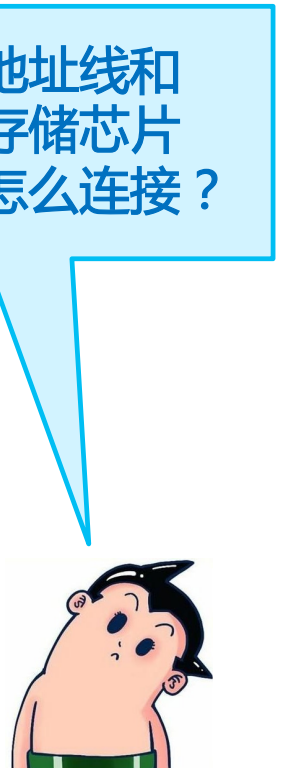
1K ×4位 RAM

1K ×4位 RAM

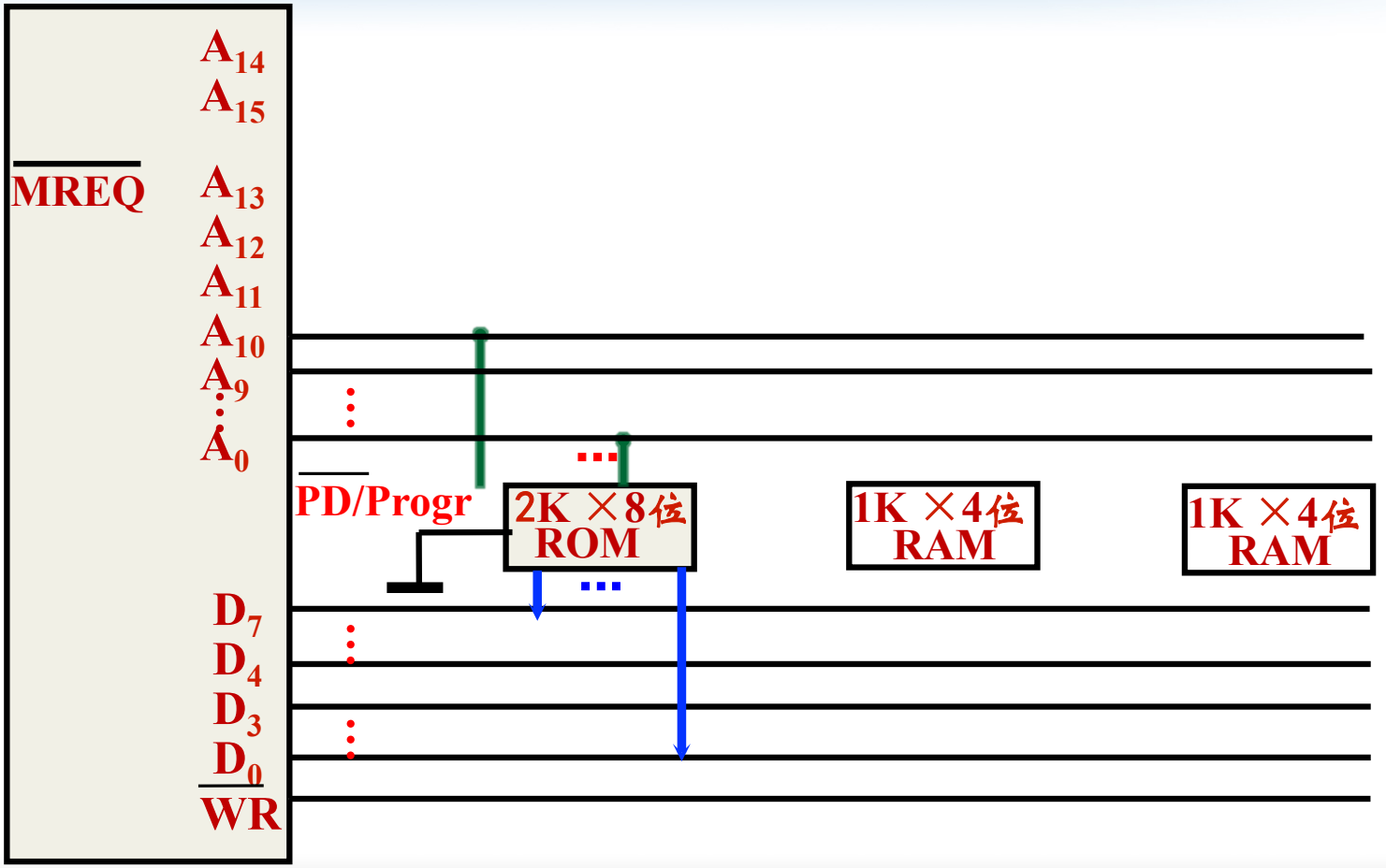
(3) 分配地址线

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_7	…	A_4	A_3	…	A_0	
0	1	1	1	0	0	0	0	0	0	0	0	0	0	1片ROM 2K×8位
⋮														
0	1	1	1	0	1	1	1	1	1	1	1	1	1	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	2片RAM 1K×4位
⋮														
0	1	1	1	1	0	1	1	1	1	1	1	1	1	

A_{10}~A_0 接 2K × 8位 ROM 的地址线



5.2.2 存储器与CPU的连接

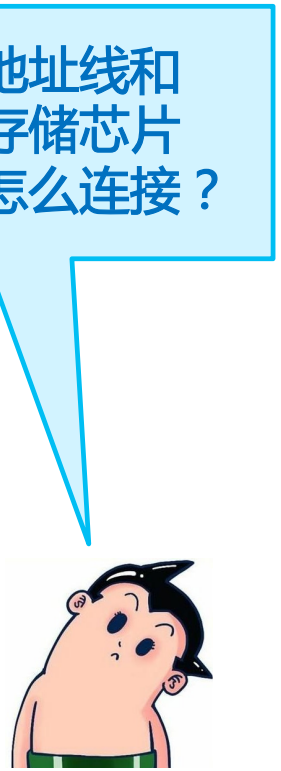


(3) 分配地址线

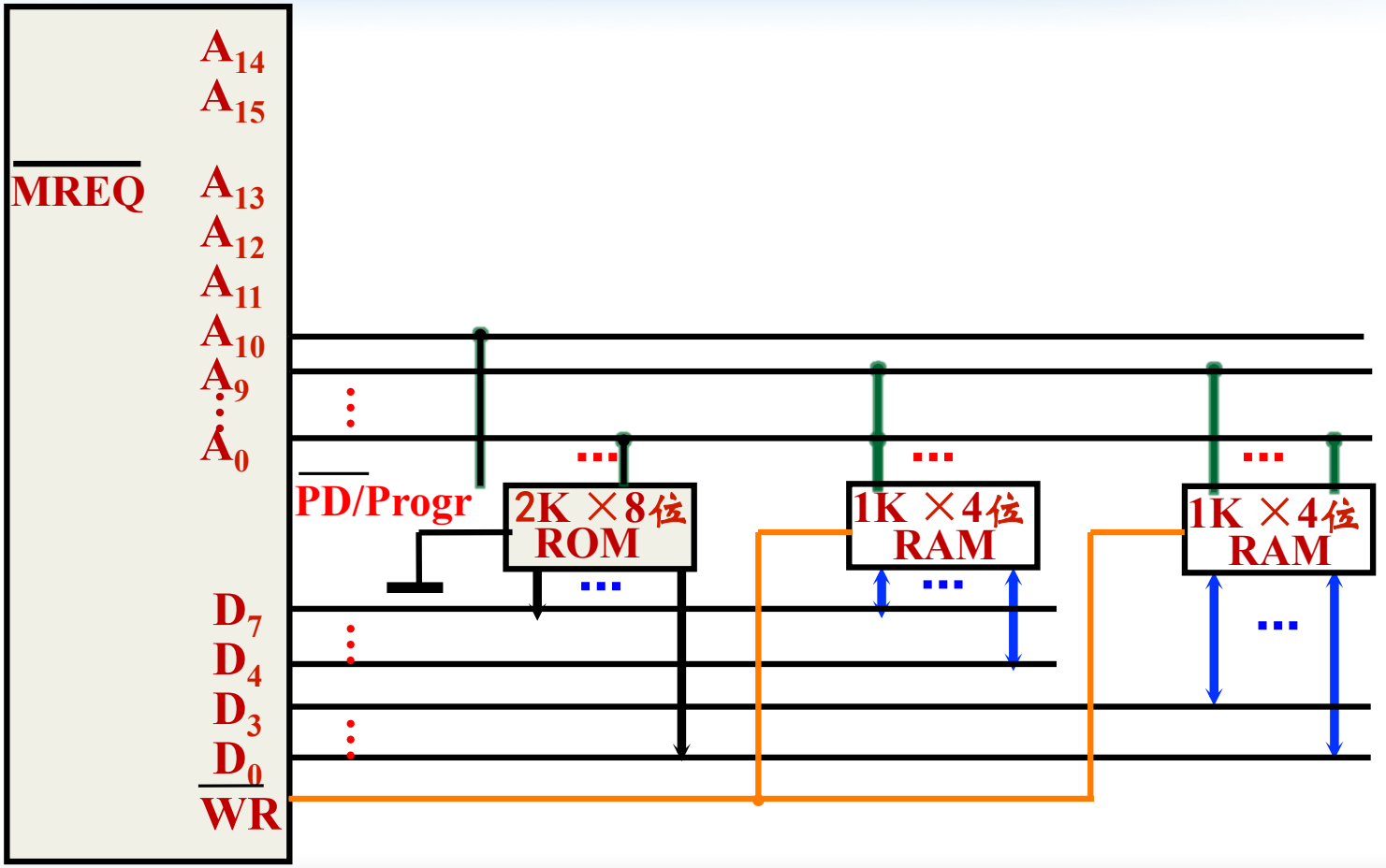
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	…	A_7	…	A_4	A_3	…	A_0	
0	1	1	1	0	0	0	0	0	0	0	0	0	0	1片ROM 2K×8位
⋮														
0	1	1	1	0	1	1	1	1	1	1	1	1	1	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	2片RAM 1K×4位
⋮														
0	1	1	1	1	0	1	1	1	1	1	1	1	1	

A_{10}~A_0 接 2K × 8位 ROM 的地址线

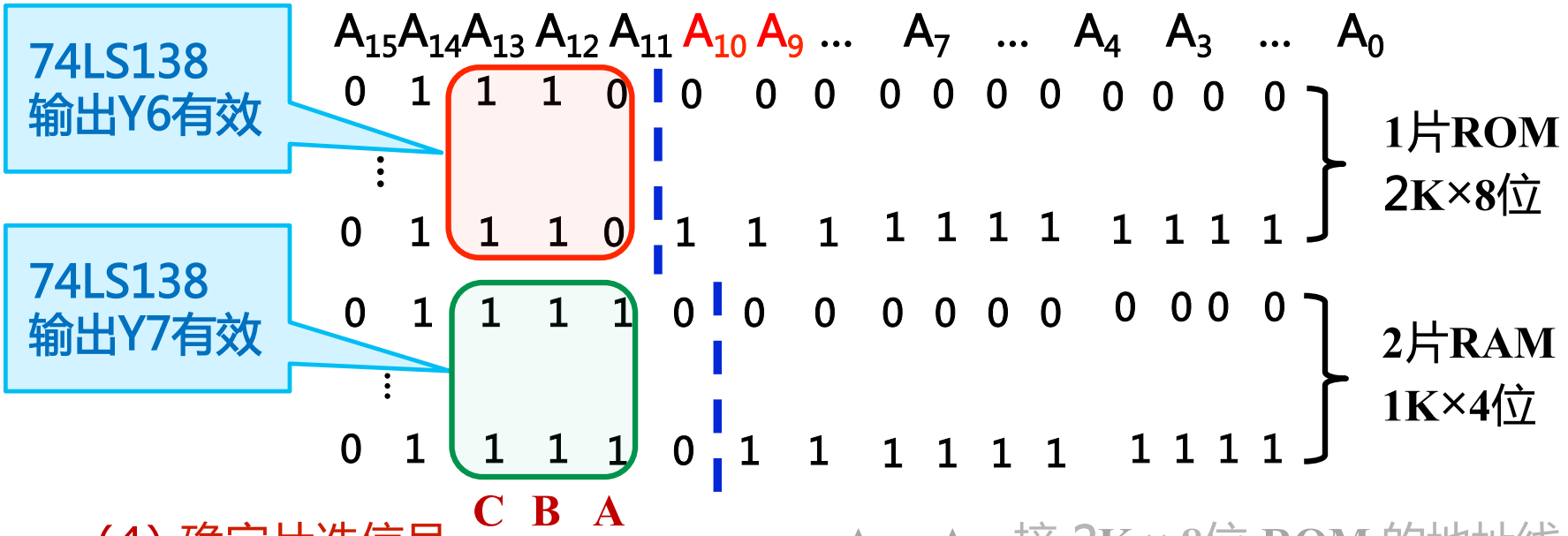
A_9 ~ A_0 接 1K ×4位 RAM 的地址线



5.2.2 存储器与CPU的连接



(3) 分配地址线

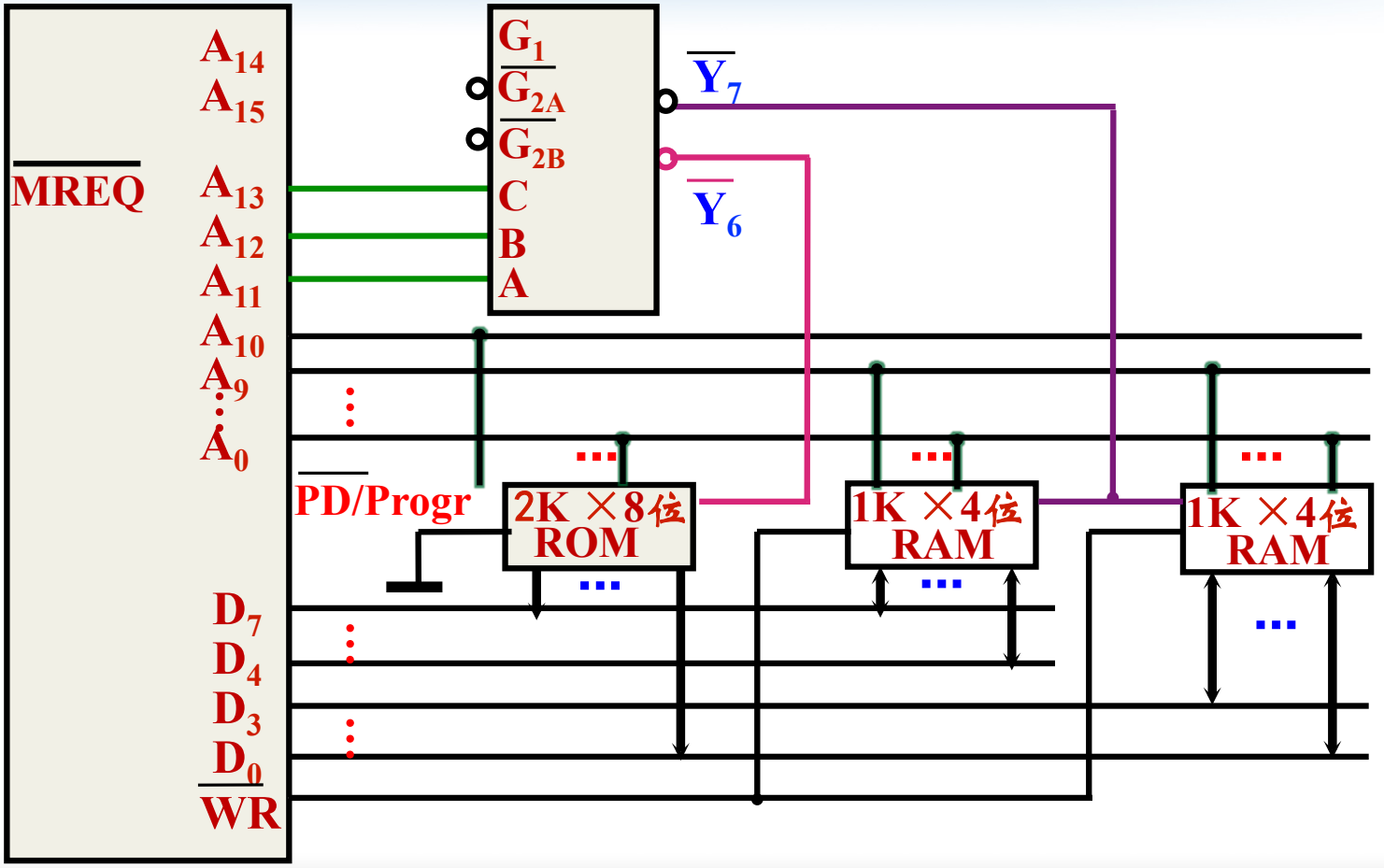


(4) 确定片选信号

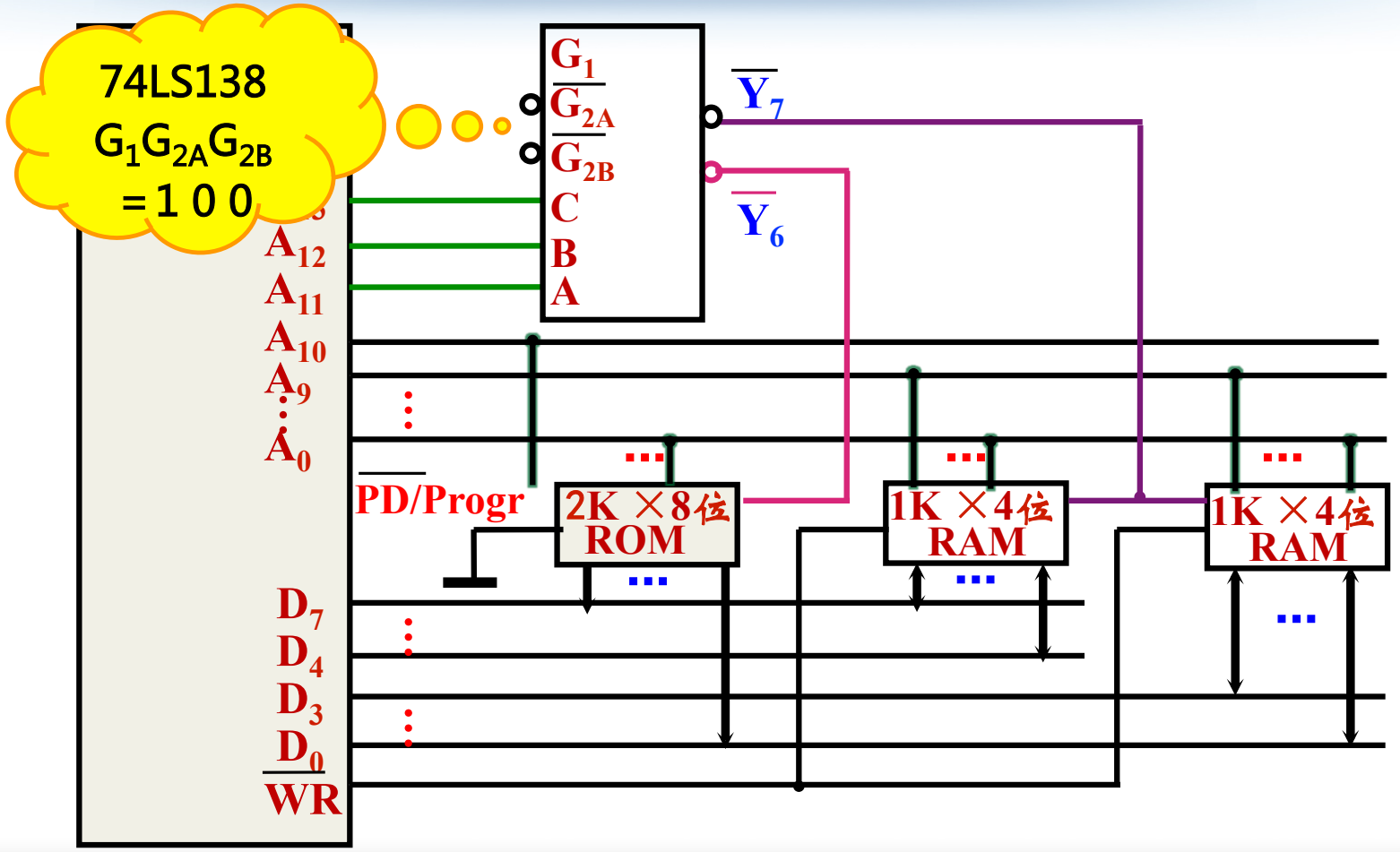
A_{10}~A_0 接 2K × 8位 ROM 的地址线

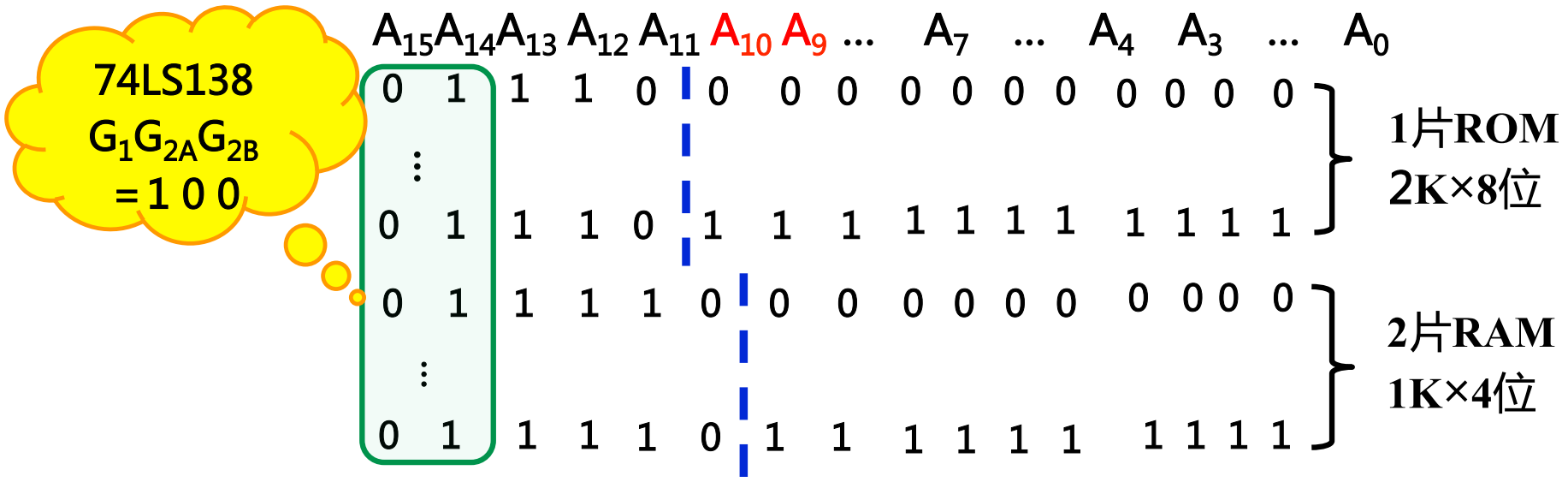
A_9 ~A_0 接 1K ×4位 RAM 的地址线

(5)画出CPU与存储器的连接图



(5)画出CPU与存储器的连接图

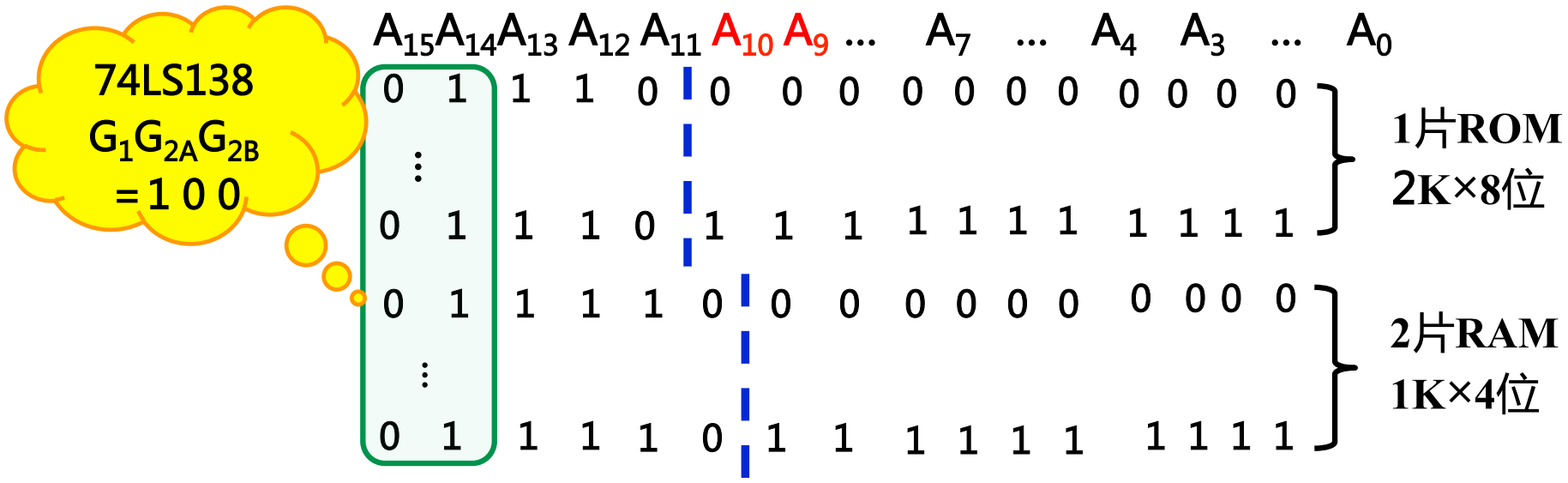




(4) 确定片选信号

A_{10}~A_0 接 2K × 8位 ROM 的地址线

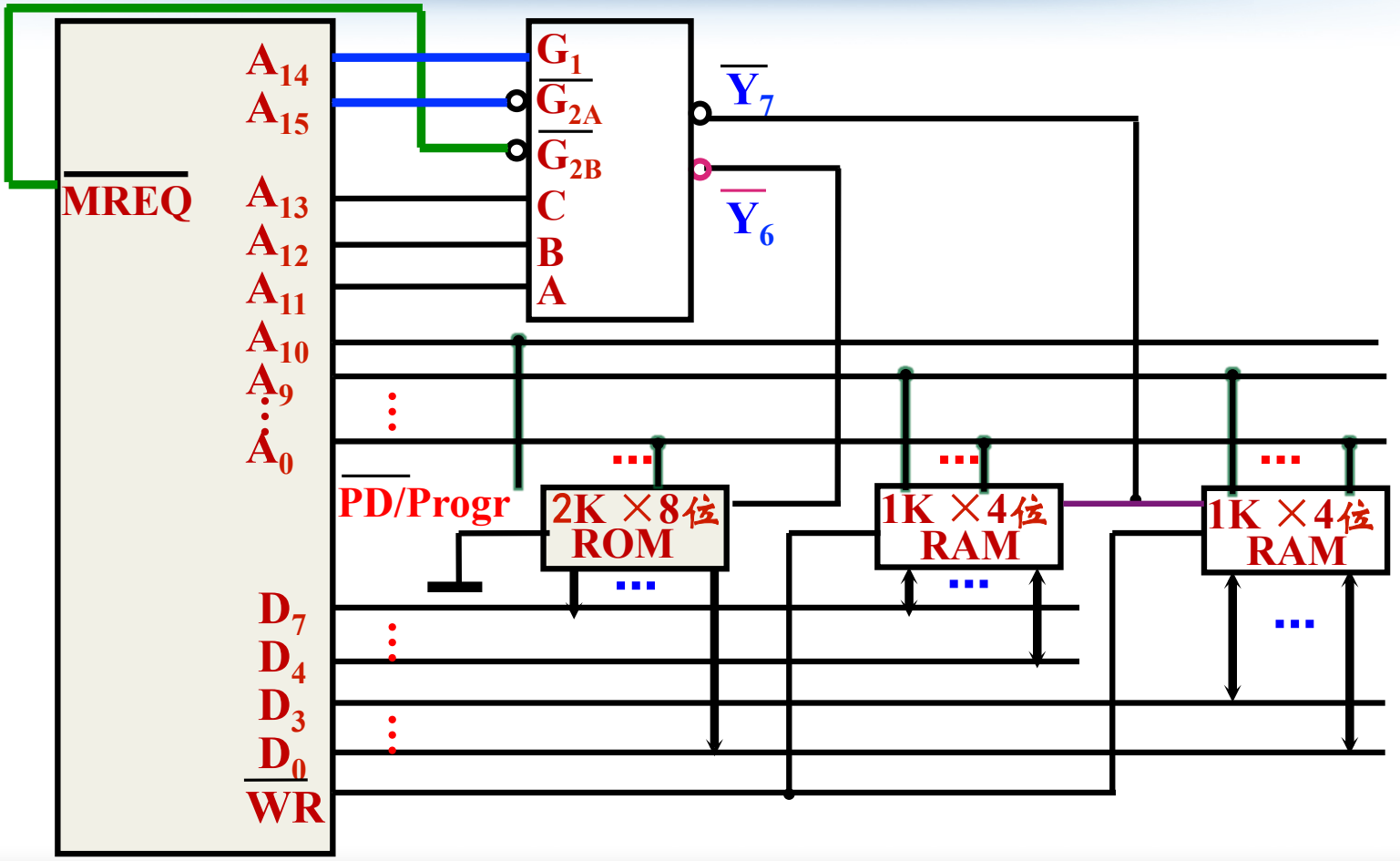
A_9 ~A_0 接 1K ×4位 RAM 的地址线



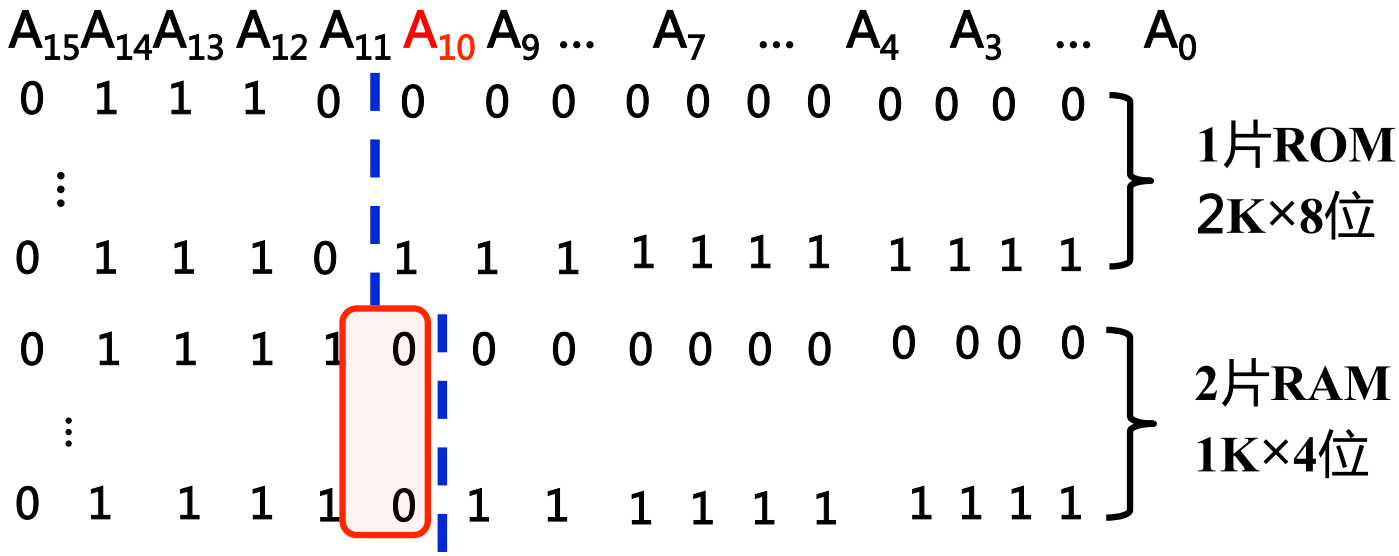
(4) 确定片选信号

CPU用$\overline{MREQ}$作访存控制信号

5.2.2 存储器与CPU的连接

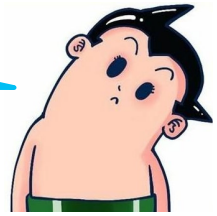


5.2.2 存储器与CPU的连接

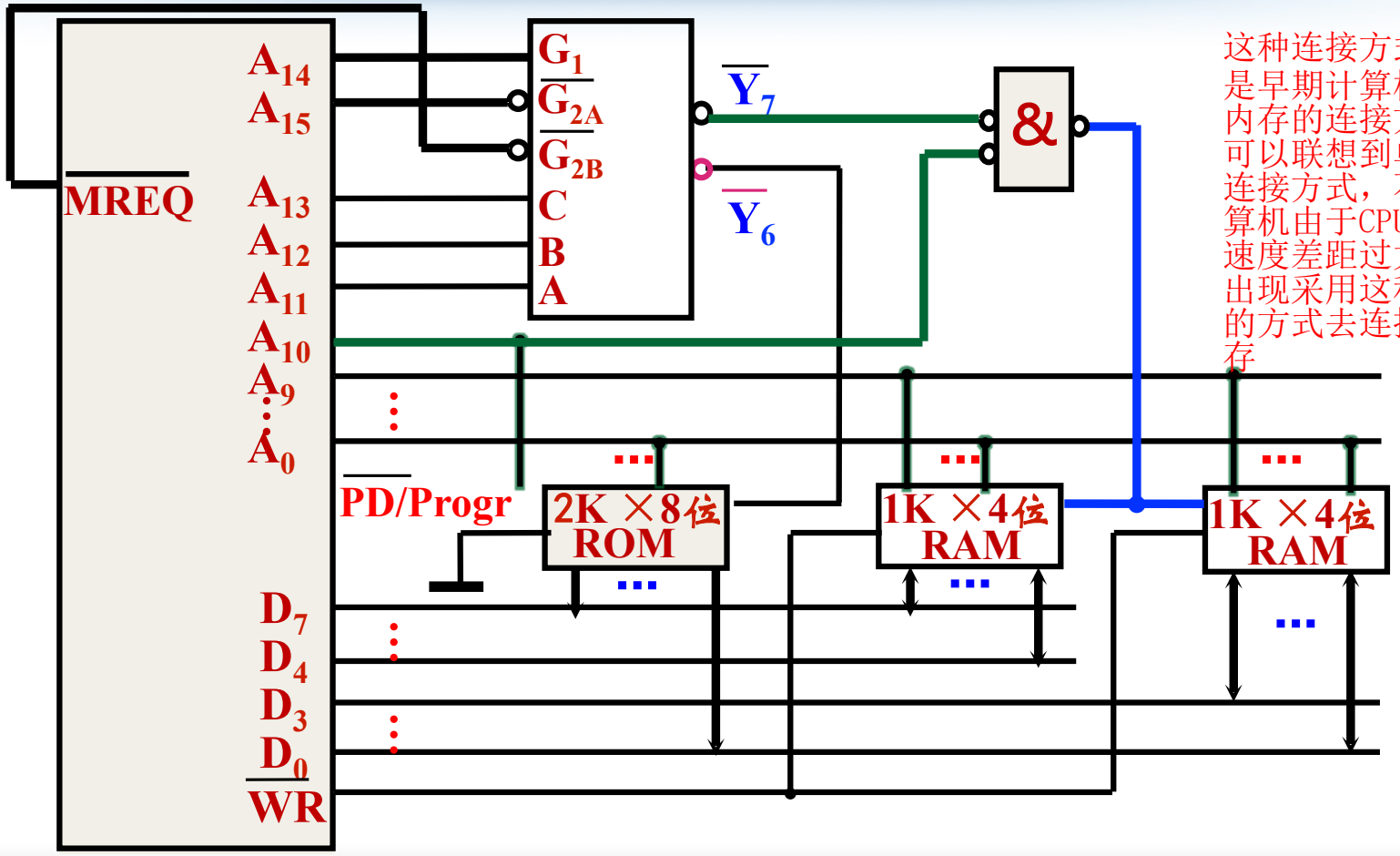


(4) 确定片选信号

当CPU的地址A_{10}为“1”时，会怎样？

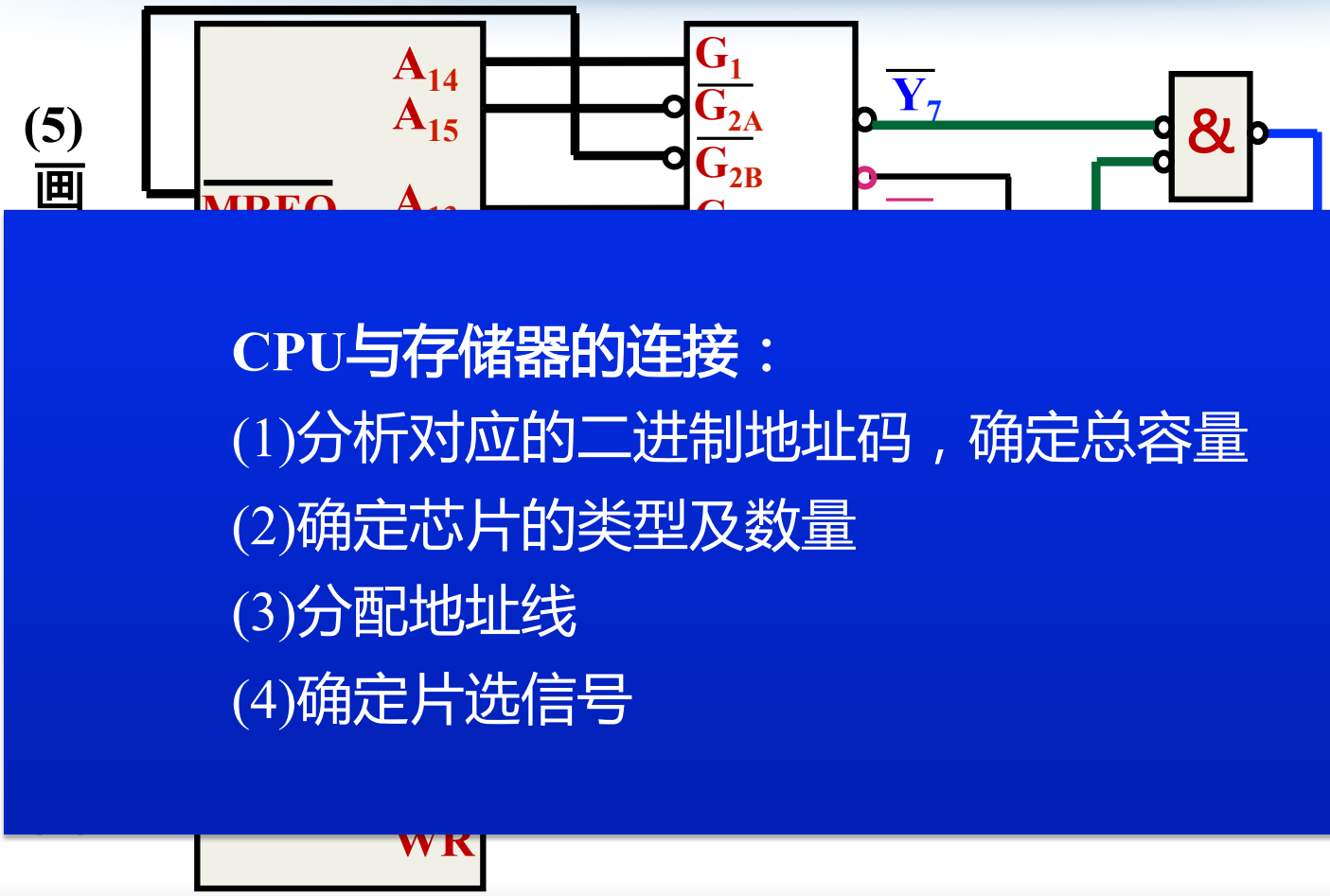


(5)画出CPU与存储器的连接图



这种连接方式可以认为是早期计算机中CPU和内存的连接方式；或者可以联想到单片机中的连接方式，不过现代计算机由于CPU和主存的速度差距过大已经不会出现采用这种直接相连的方式去连接CPU和内存

(5)画



CPU与存储器的连接：

(1)分析对应的二进制地址码，确定总容量

(2)确定芯片的类型及数量

(3)分配地址线

(4)确定片选信号

5.2.3 DRAM的刷新

Why & How?

5.2.3 DRAM的刷新

为什么要刷新？

DRAM存储位元的特点

- 靠电容上的电荷存储效应记忆信息，虽然有MOS高电阻(10^{12}～$10^{15}\Omega$)，仍会泄漏电荷
- 为保证所存信息的正确性，需要用充电的方法及时使所有位元的电容上电荷恢复到泄漏前的状态

必须设置专门的刷新逻辑

什么是刷新？

按一定时间间隔为记忆电容补充电荷的过程

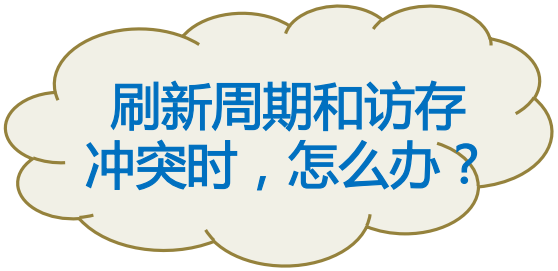
5.2.3 DRAM的刷新

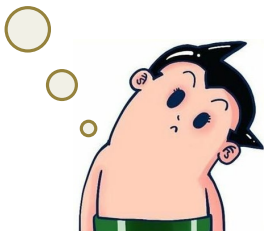
为什么要刷新？

它们是一致的，存储器的刷新就是对其内部所有存储芯片同时进行刷新

什么是刷新？

- 定时刷新
- 刷新优于访存，但不能打断访存
- 刷新期间不允许访存





DRAM刷新的有关参数

1、信息保持时间T_{ref}

从信息以电荷形式存入电容，到电荷经过一段时间泄漏，读放仍能鉴别出原存信息的时间

DRAM刷新的有关参数

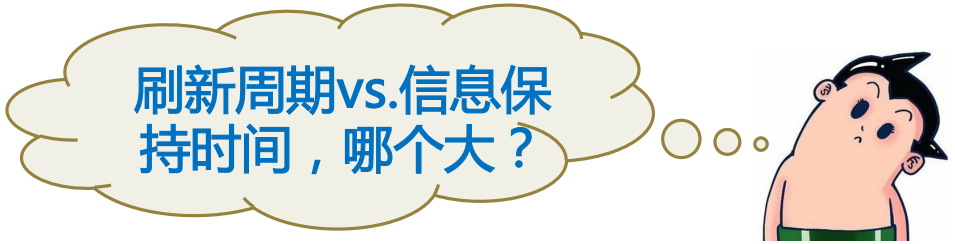
2、刷新周期T_{rc}(refresh cycle)

对同一存储位元连续两次刷新，仍能保证鉴别出原存信息的最大允许间隔时间，即在T_{rc}内必须对每个单元刷新一遍

一般为ms级，亦称为刷新间隔时间

- 在T_{rc}时间内，应刷新存储芯片中的所有存储位
- 在DRAM芯片的主要性能参数中，都给出T_{rc}。通常为2ms、4ms、8ms

$$T_{rc} < T_{ref}$$



DRAM刷新的有关参数

3、刷新操作周期T_{roc} (refresh operating cycle)

刷新一行存储位元即一次刷新操作所需时间。通常和存储周期t_{RC}/t_{WC}相同

存储器的刷新实现方法：

- 周而复始的选取存储矩阵的各行(即二维选址中的一维)进行刷新，同时刷新同一行的所有存储单元
- 在刷新周期内选取完存储矩阵的所有行

DRAM刷新的有关参数

4、刷新操作周期数N_r (Number of refresh operating cycle)

存储芯片所有位元刷新一遍所需的刷新操作周期个数。它与芯片的内部结构有关。即每次可以刷新一行， Nr决定与存储矩阵的行数

例：DRAM存储芯片MCM4027，其字位结构为4K×1位，存储矩阵为64行×64列。每次刷新操作刷新存储矩阵的一行，刷新操作周期数是多少？

每次刷新一行(64个存储位元)，需刷新操作周期数$N_r=64$

DRAM芯片的3个重要参数：刷新周期、刷新操作周期、刷新操作周期数

分类标准

- 按-CAS、-RAS引脚所加时序分类
- 按刷新操作控制方式分类 (同步、异步、半同步刷新)
- 按刷新操作周期的分配方式分类

分类标准

按刷新操作周期的分配方式分类

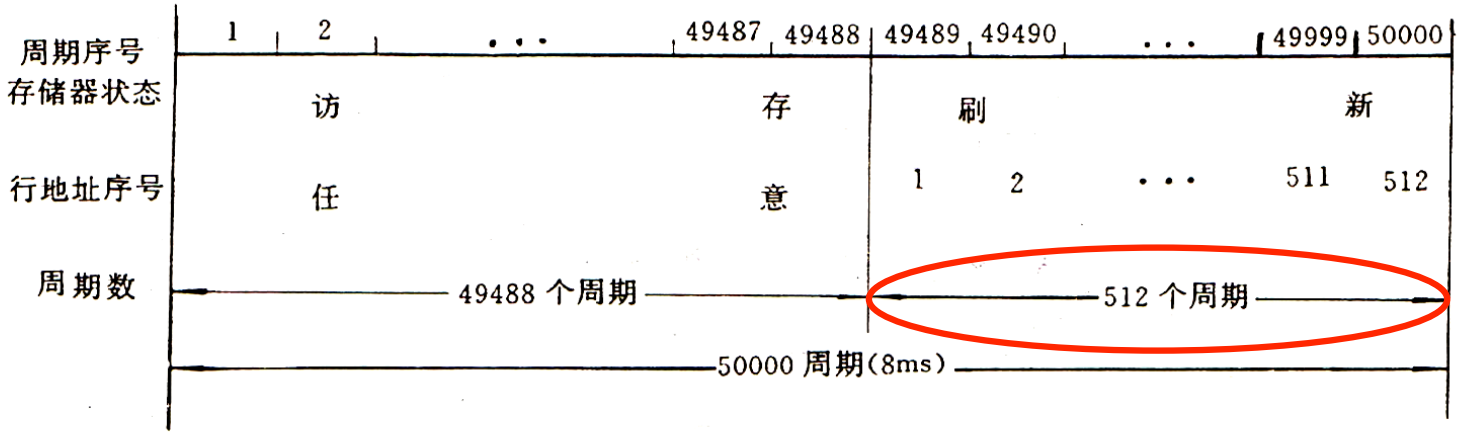
- 集中式刷新
- 分散式刷新
- 透明式刷新

以MCM511000A芯片为例：
刷新周期：8ms；刷新操作周期：160ns；刷新周期数：512

5.2.3 DRAM的刷新

集中式刷新

从刷新周期Trc中抽出最后512个访存周期作为刷新操作周期集中进行刷新。亦称批刷新



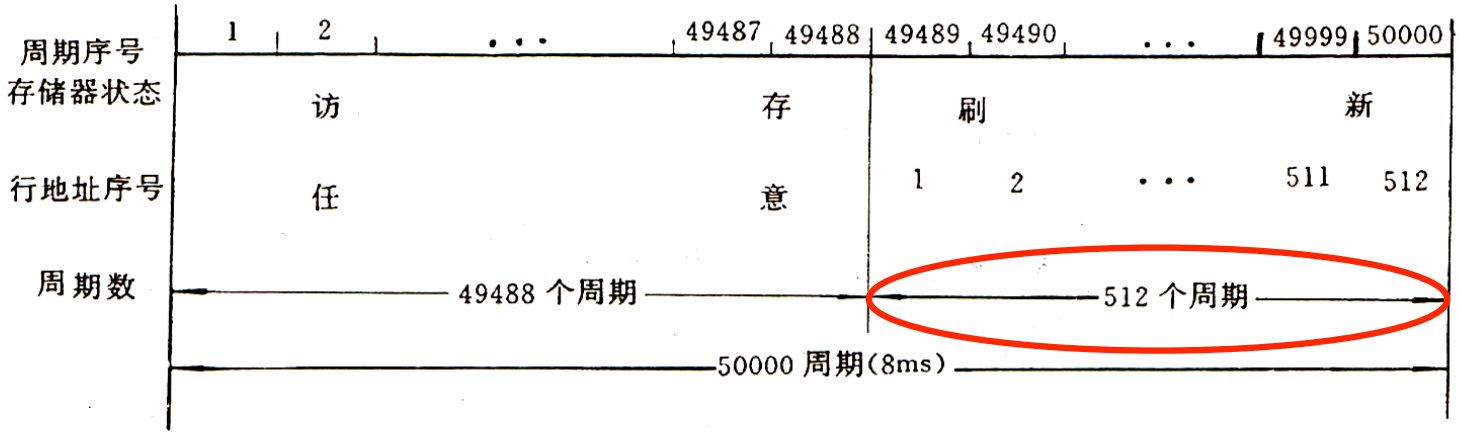
(a) 集中式刷新操作周期分配图

- 以MCM511000A为例，8ms包含50000个访存周期:前49488用于访存,后512用于刷新
- 刷新操作周期所占比例:512/50000=1.024%

5.2.3 DRAM的刷新

集中式刷新

从刷新周期Trc中抽出最后512个访存周期作为刷新操作周期集中进行刷新。亦称批刷新



(a) 集中式刷新操作周期分配图

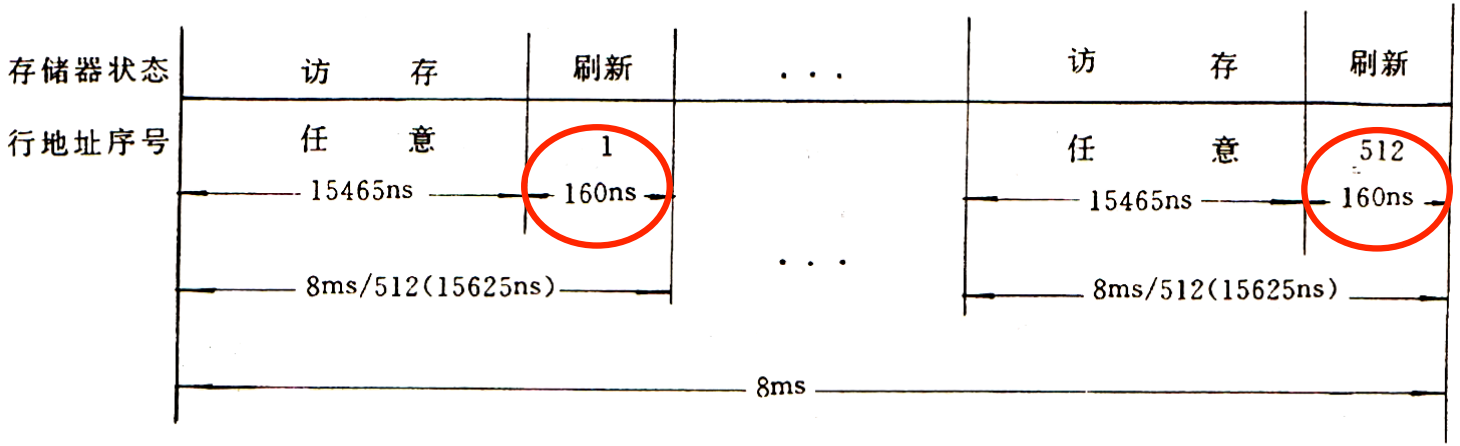
缺点：集中式刷新使98.976%的时间用于访存，这期间存储器的效能得以充分发挥，但有1.024%的时间，即在8ms中有81.92μs不允许访存，CPU要处于等待状态，影响了计算机的工作效率

优点：控制逻辑简单，设计容易实现

5.2.3 DRAM的刷新

分散式刷新

把集中到一起的不允许访存的刷新时间分散开，每个等分的最后一个访存周期用作刷新操作周期，以完成一行存储位元的刷新，其余时间则用于访存



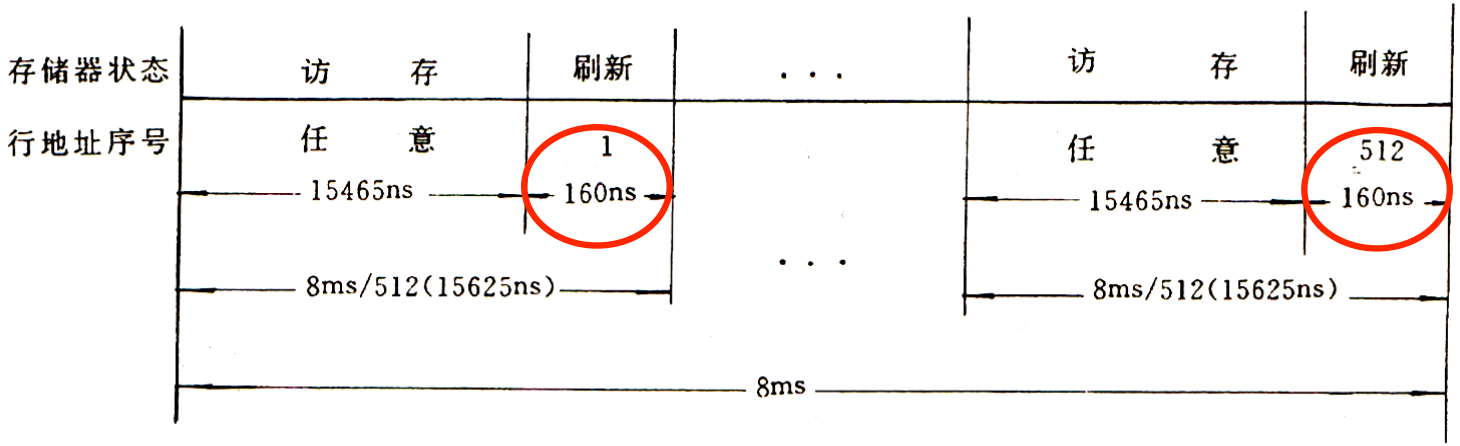
(b) 分散式刷新操作周期分配图

- 以MCM511000A为例，把8ms按存储位元的行数(即行地址译码线数)分成512等分
- 刷新操作周期所占比例:512/50000=1.024%

5.2.3 DRAM的刷新

分散式刷新

把集中到一起的不允许访存的刷新时间分散开，每个等分的最后一个访存周期用作刷新操作周期，以完成一行存储位元的刷新，其余时间则用于访存



(b) 分散式刷新操作周期分配图

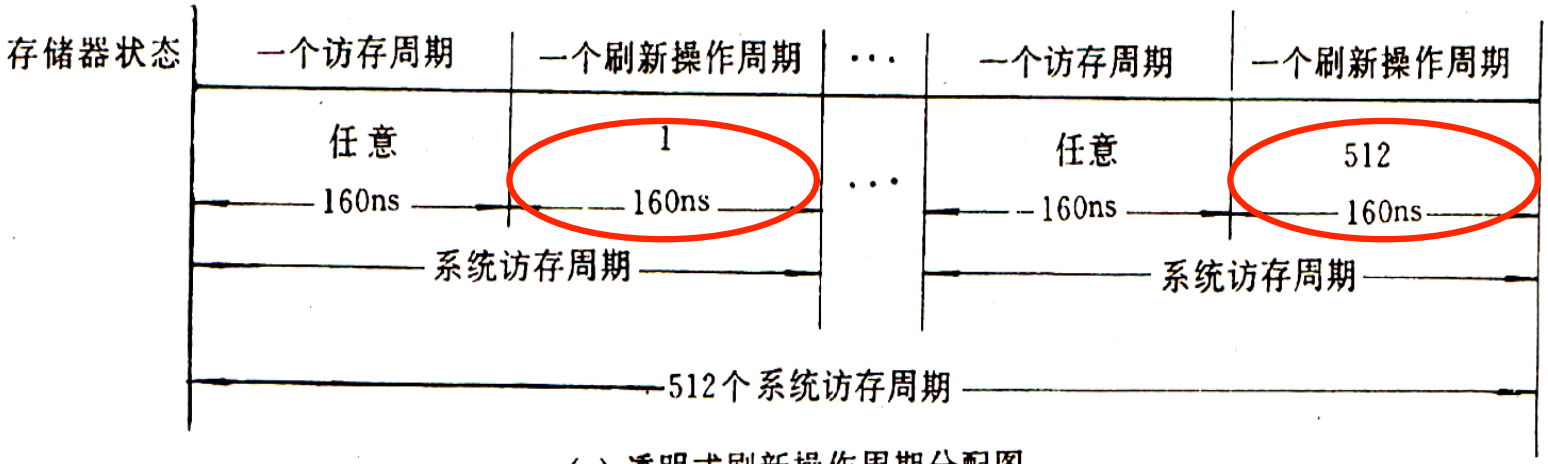
优点：提高了计算机的工作效率

缺点：控制逻辑复杂，设计不易实现

5.2.3 DRAM的刷新

透明式刷新

设一个系统的访存周期是存储器实际访存周期的两倍，并令系统访存周期的前半周期用于访存，后半周期用于刷新



(c) 透明式刷新操作周期分配图

优点：控制简单、设计容易，不需增加多少器材

缺点：存储器的效能仅利用50%，仅用于低速系统