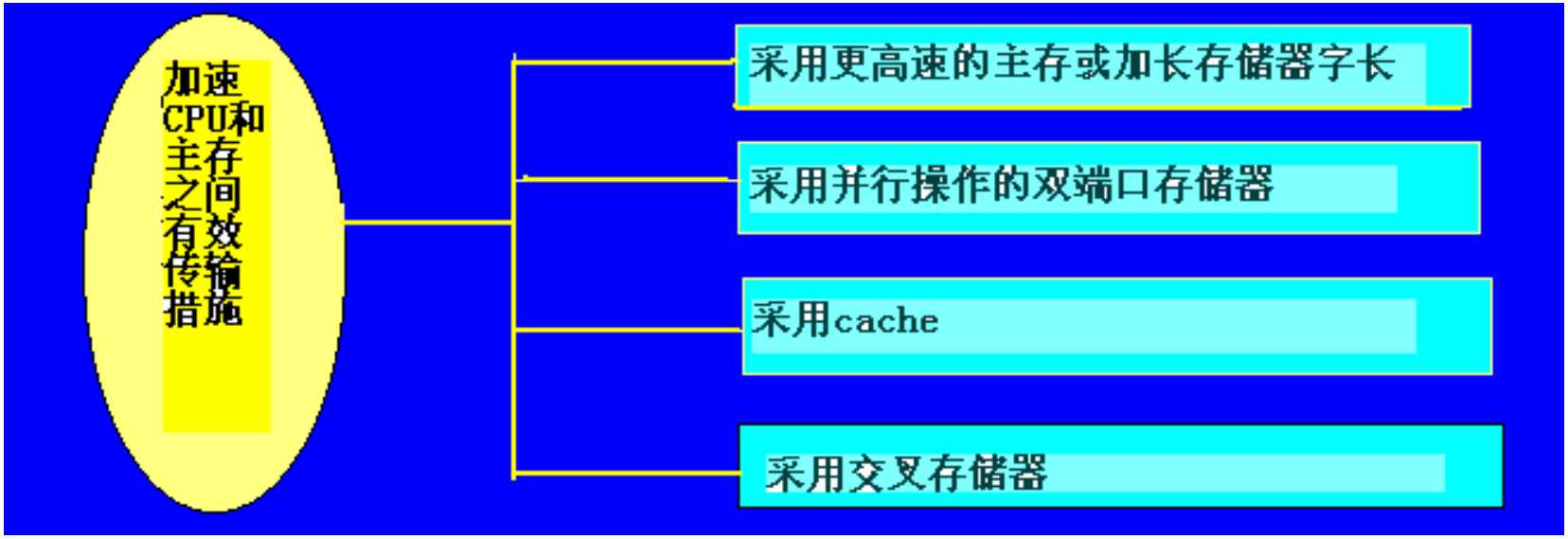
3.5 并行存储器

- 由于CPU和主存储器之间在速度上是不匹配的，这种情况便成为限制高速计算机设计的主要问题。为了提高CPU和主存之间的数据传输率，除了主存采用更高速的技术来缩短读出时间外，还可以采用并行技术的存储器。



提纲

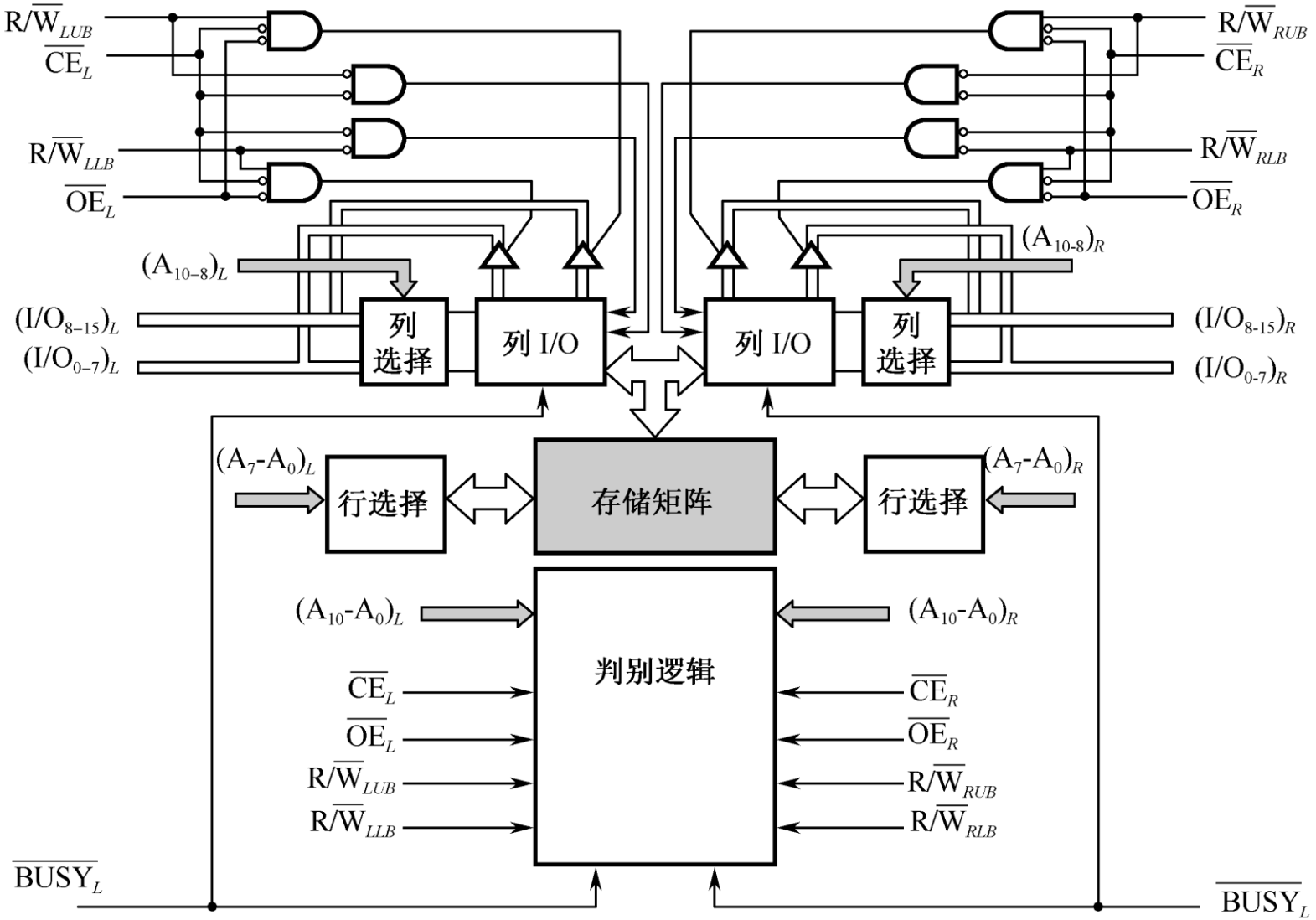
3.5.1 双端口存储器

1、双端口存储器的逻辑结构

- 由于同一个存储器具有两组相互独立的读写控制电路而得名。由于进行并行的独立操作，因而是一种高速工作的存储器，在科研和工程中非常有用。

3.5.1 双端口存储器

- 举例说明，双端口存储器IDT7133（2K×16的SRAM）的逻辑框图



3.5.1 双端口存储器

2、无冲突读写控制

- 两个端口的地址不相同时，在两个端口上进行读写操作，一定不会发生冲突。当任一端口被选中驱动时，就可对整个存储器进行存取，每一个端口都有自己的片选控制和输出驱动控制
- 无冲突的读/写条件（1：高电平，0：低电平，X：任意，Z：高阻）

左端口或右端口						功　能
R/$\overline{W}$lb	R/$\overline{W}$ub	$\overline{CE}$	$\overline{OE}$	I/O0-7	I/O8-15	
×	×	1	1	Z	Z	端口不用
0	0	0	×	数据入	数据入	低位和高位字节数据写入存储器
0	1	0	0	数据入	数据出	低位字节数据写入存储器，存储器中数据输出至高位字节
1	0	0	0	数据出	数据入	存储器中数据输出至低位字节，高位字节数据写入存储器
0	1	0	1	数据入	Z	低位字节写入存储器
1	0	0	1	Z	数据入	高位字节写入存储器
1	1	0	0	数据出	数据出	存储器中数据输出至低位字节号高位字节
1	1	0	1	Z	Z	高阻抗输出

3.5.1 双端口存储器

3、有冲突读写控制

- 当两个端口同时存取存储器同一存储单元时，便发生读写冲突。为解决此问题，特设置了BUSY标志。由片上的判断逻辑决定对哪个端口优先进行读写操作，而暂时关闭另一个被延迟的端口
- 判断方式有两种：
  - CE判断：如果地址匹配且在CE之前有效，片上的控制逻辑在CE_L和CE_R之间进行判断来选择端口
  - 地址有效判断：如果CE在地址匹配之前变低，片上的控制逻辑在左、右地址间进行判断来选择端口

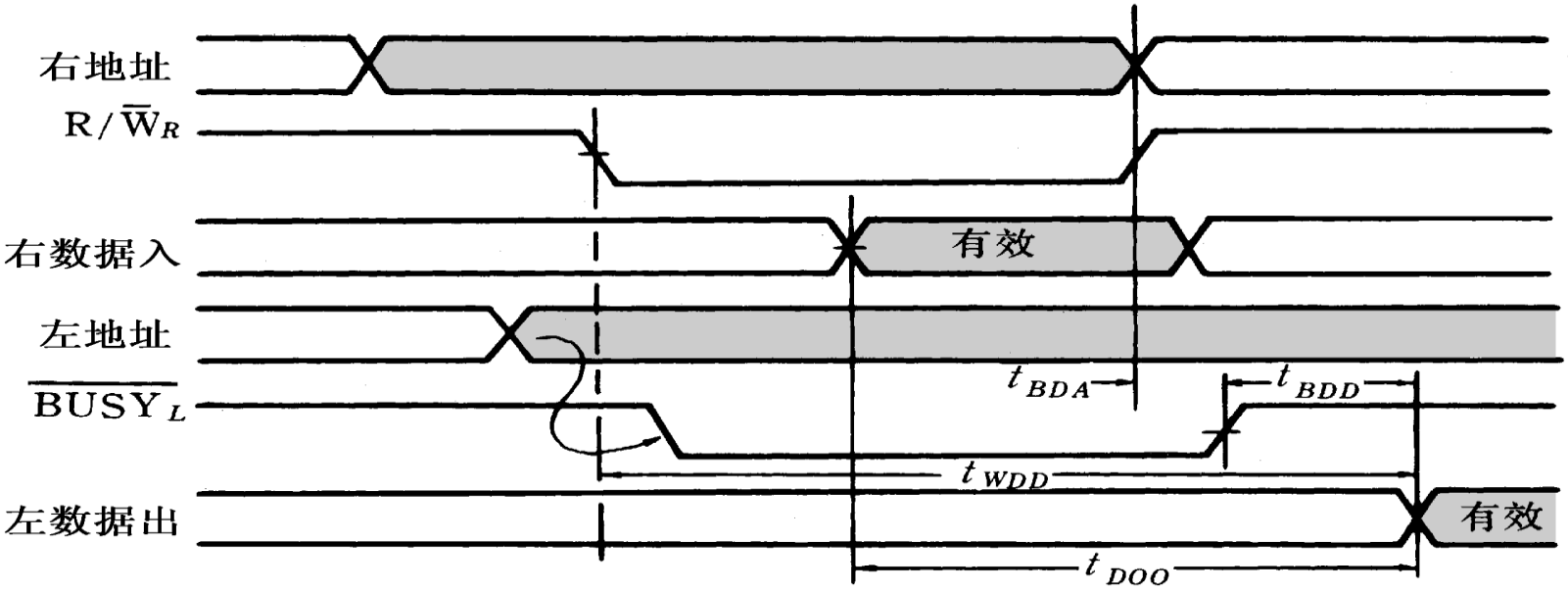
3.5.1 双端口存储器

- LV5R：左地址有效先于右地址50ns；RV5L：右地址有效先于左地址50ns；Same：左右地址均在50ns内匹配
- LL5R：$\overline{CE_L}$ 变低先于 $\overline{CE_R}$ 50ns；RL5L：$\overline{CE_R}$ 变低先于 $\overline{CE_L}$ 50ns；LW5R：$\overline{CE_L}$ 和 $\overline{CE_R}$ 均在50ns内变低

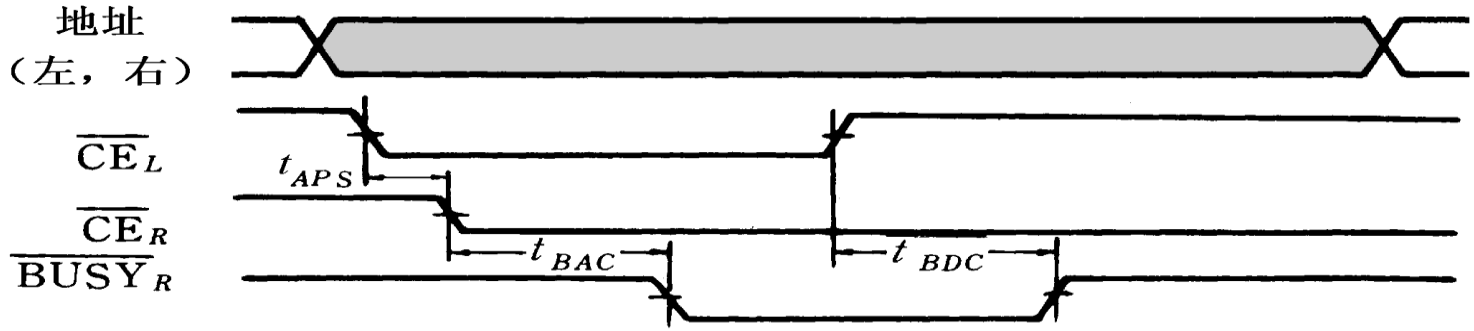
第6行应该是$(A_0\text{-}A_{10})L$ 第11行LL5R

左端口		右端口		标志		功能	说明
$\overline{CEL}$	(A0-A10)L	$\overline{CER}$	(A0-A10)R	$\overline{BUSYL}$	$\overline{BUSYR}$		
1	×	1	×	1	1	无冲突	
0	Any	1	×	1	1	无冲突	
1	×	0	Any	1	1	无冲突	
0	≠(A0-A10)R	0	≠(A0-A10)R	1	1	无冲突	
0	LV5R	0	LV5R	1	0	左端口取胜	
0	RV5L	0	RV5L	0	1	右端口取胜	$\overline{CE}$在地址匹配之前变低的地址判断
0	Same	0	Same	1	0	消除判断	
0	Same	0	Same	0	1	消除判断	
RL5R	=(A0-A10)R	LL5R	=(A0-A10)L	1	0	左端口取胜	
RL5L	=(A0-A10)R	RL5L	=(A0-A10)L	0	1	右端口取胜	
LW5R	=(A0-A10)R	LW5R	=(A0-A10)L	1	0	消除判断	地址匹配在$\overline{CE}$之前的$\overline{CE}$判断
LW5R	=(A0-A10)R	LW5R	=(A0-A10)L	0	1	消除判断	

3.5.1 双端口存储器



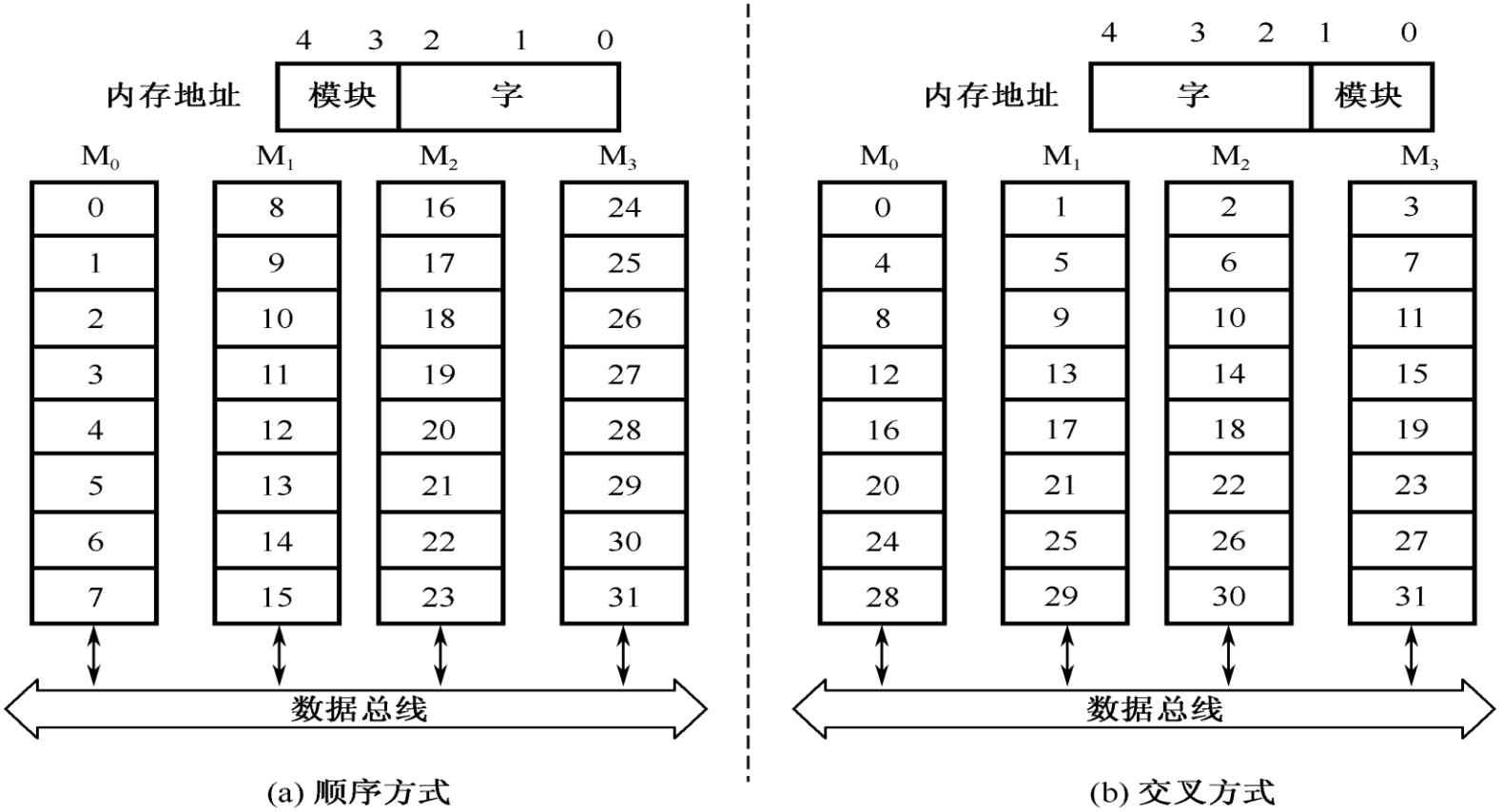
(a) 未发生冲突时读写时序波形（左读右写）



(b) 用$\overline{CE}$判断的冲突周期时序波形（$\overline{CE}_L$先有效）

3.5.2 多模块交叉存储器：

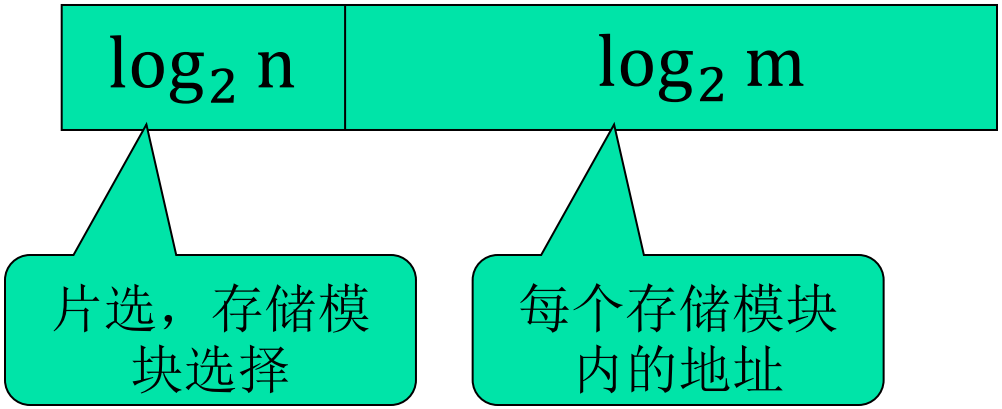
- 一个由若干个模块组成的主存储器是线性编址的
- 这些地址在各模块中如何安排，有两种方式：一种是顺序方式，一种是交叉方式



3.5.2 多模块交叉存储器：

- 假设有n个存储模块，每个存储模块的容量为m个存储单元

1、顺序方式



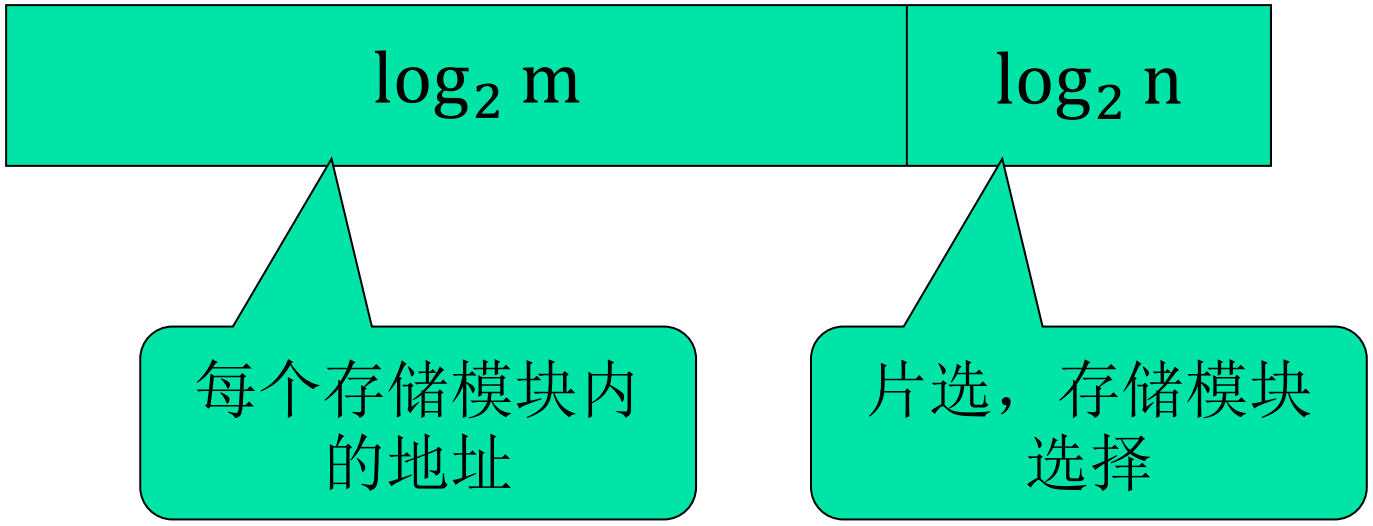
3.5.2 多模块交叉存储器：

- [例]M0 - M3共四个模块，则每个模块8个字
- 顺序方式： M0：0—7
  M1：8 - 15
  M2：16 - 23
  M3：24 - 31
- 5位地址组织如下： X X X X X
- 高位选模块，低位选块内地址
- 特点：某个模块进行存取时，其他模块不工作，优点是某一模块出现故障时，其他模块可以照常工作，通过增添模块来扩充存储器容量比较方便；缺点是各模块串行工作，存储器的带宽受到了限制。

3.5.2 多模块交叉存储器：

2、交叉方式

- （可以实现多模块流水式并行存取）



3.5.2 多模块交叉存储器：

- [例]M0 - M3共四个模块，则每个模块8个字
- 交叉方式：

  M0：0，4,...除以4余数为0

  M1：1，5,...除以4余数为1

  M2：2，6,...除以4余数为2

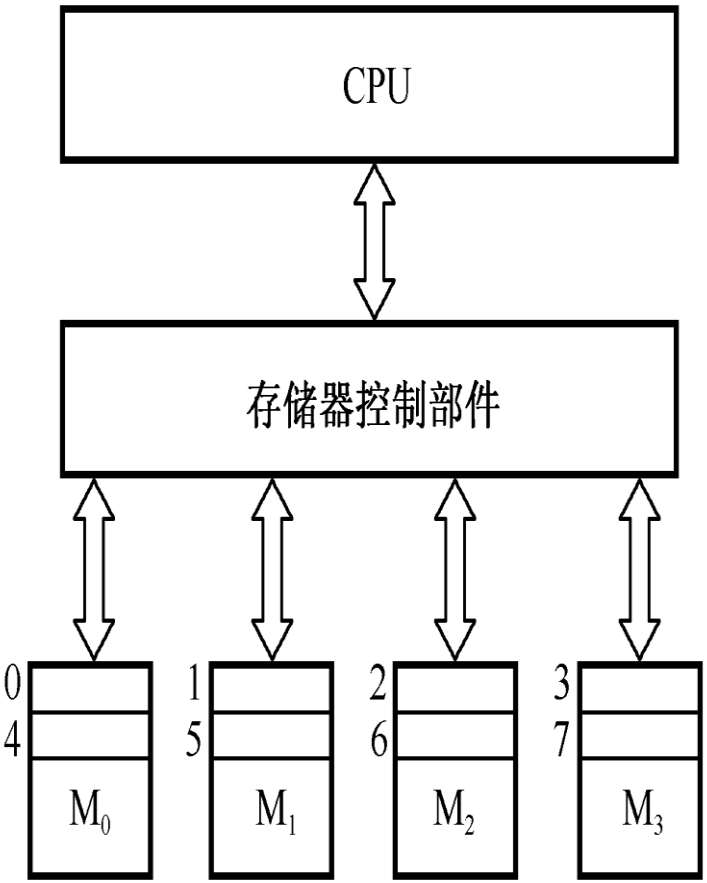
  M3：3，7,...除以4余数为3

- 5位地址组织如下： X X X　X X
- 高位选块内地址，低位选模块
- 特点：连续地址分布在相邻的不同模块内，同一个模块内的地址都是不连续的。优点是对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽。使用场合为成批数据读取

3.5.2 多模块交叉存储器：

3、多模块交叉存储器的基本结构

- 右图为四模块交叉存储器结构框图。主存被分成4个相互独立、容量相同的模块M0，M1，M2，M3，每个模块都有自己的读写控制电路、地址寄存器和数据寄存器，各自以等同的方式与CPU传送信息
- 在理想情况下，如果程序段或数据块都是连续地在主存中存取，那么将大大提高主存的访问速度。
- 由存储器控制部件控制它们分时使用数据总线进行信息传递

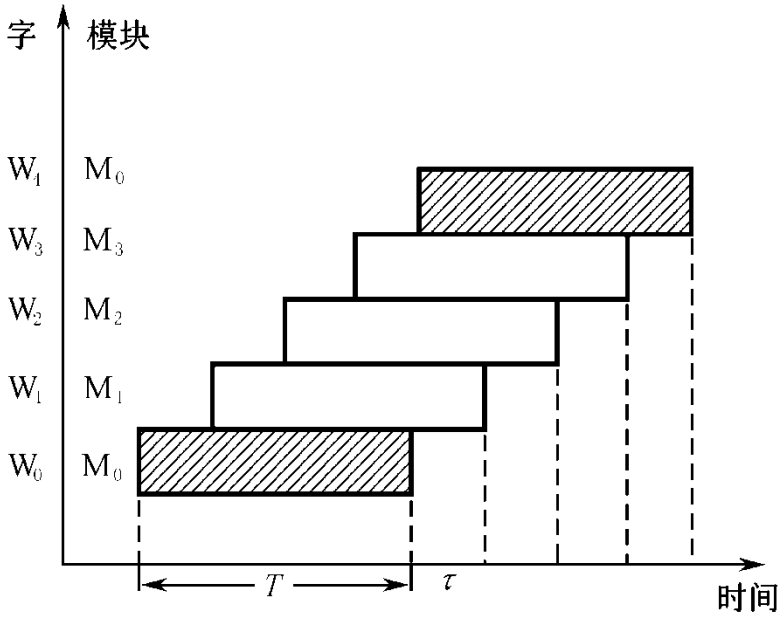


3.5.2 多模块交叉存储器：

定量分析

- 通常在一个存储器周期内，m个存储模块必须分时启动，各个存储体的启动间隔为 $\tau = T/m$ （m为交叉存取度）
- 则整个存储器的存取速度有望提高m倍

$$t_{顺序} = mT$$
$$t_{交叉} = T + (m-1)\tau$$



3.5.2 多模块交叉存储器：

- 例5 设存储器容量为32字，字长64位，模块数m=4，分别用顺序方式和交叉方式进行组织。存储周期T=200ns，数据总线宽度为64位，总线传送周期=50ns。若连续读出4个字，问顺序存储器和交叉存储器的带宽各是多少?

- 解：顺序存储器和交叉存储器连续读出m=4个字的信息总量都是：

$$q=64b\times4=256b$$

顺序存储器和交叉存储器连续读出4个字所需的时间分别是：

$t2=mT=4\times200ns=800ns=8\times10^{-7}s$

$t1=T+(m-1)x\tau=200ns+150ns=350ns=35\times10^{-7}s$

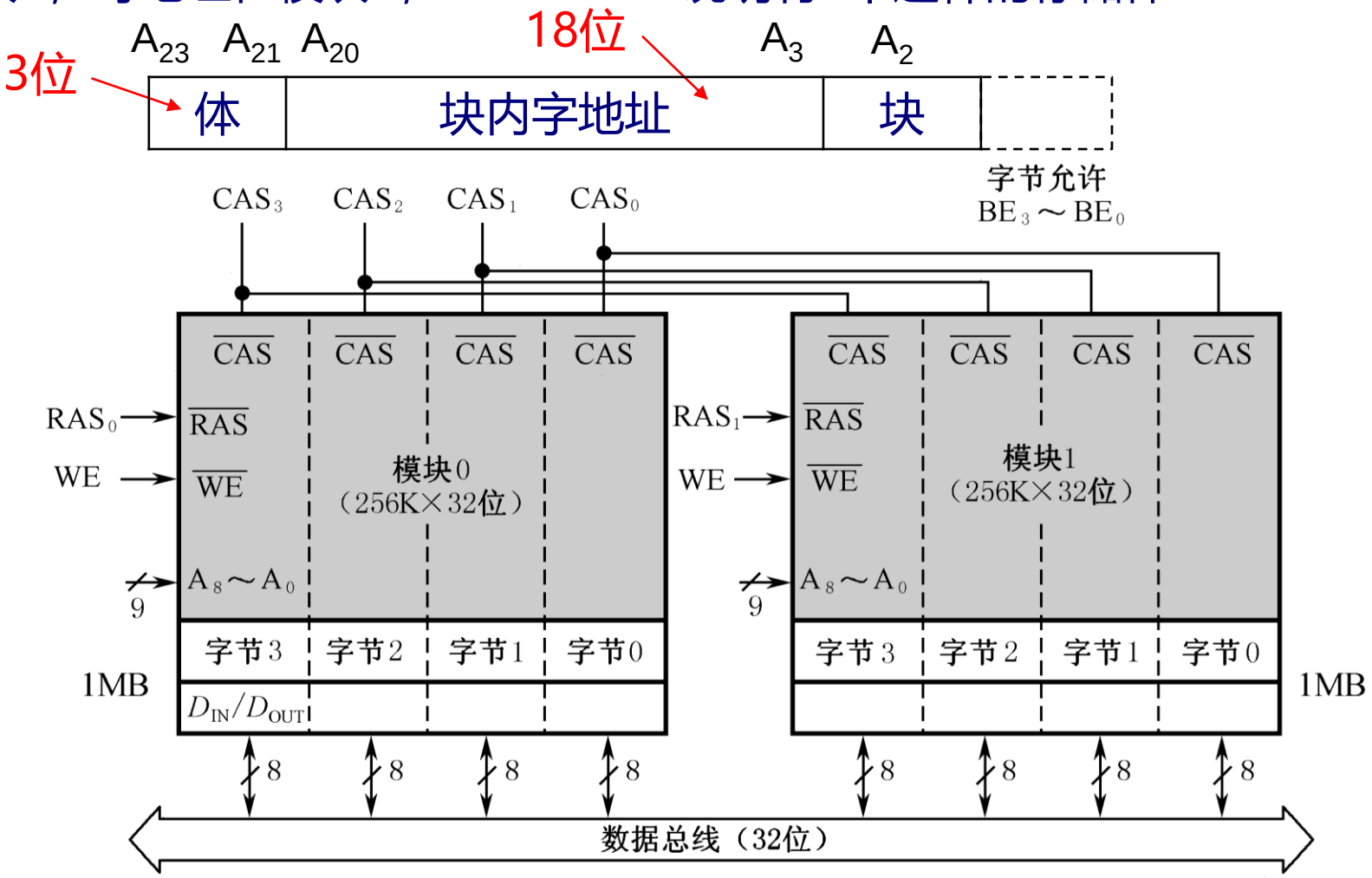
顺序存储器和交叉存储器的带宽分别是：

$W2=q/t2=256b\div(8\times10^{-7})s=320Mb/s$

$W1=q/t1=256b\div(35\times10^{-7})s=730Mb/s$

3.5.2 多模块交叉存储器：

- 二模块交叉存储器举例，A20—A3的18位地址用于模块中256K个存储字的选择；A2用模块选择，连续的存储字交错分布在两个模块上，偶地址在模块0，奇地址在模块1； A23—A21说明有8个这样的存储体



3.5.2 多模块交叉存储器：

- 刷新周期是在RAS有效下输入刷新地址，此地址指示的一行所有存储元全部被再生

注意刷新与重写的区别。

破坏性读出后重写，以恢复原来的信息。

非破坏性读出的动态M，需补充电荷以保持原来的信息。