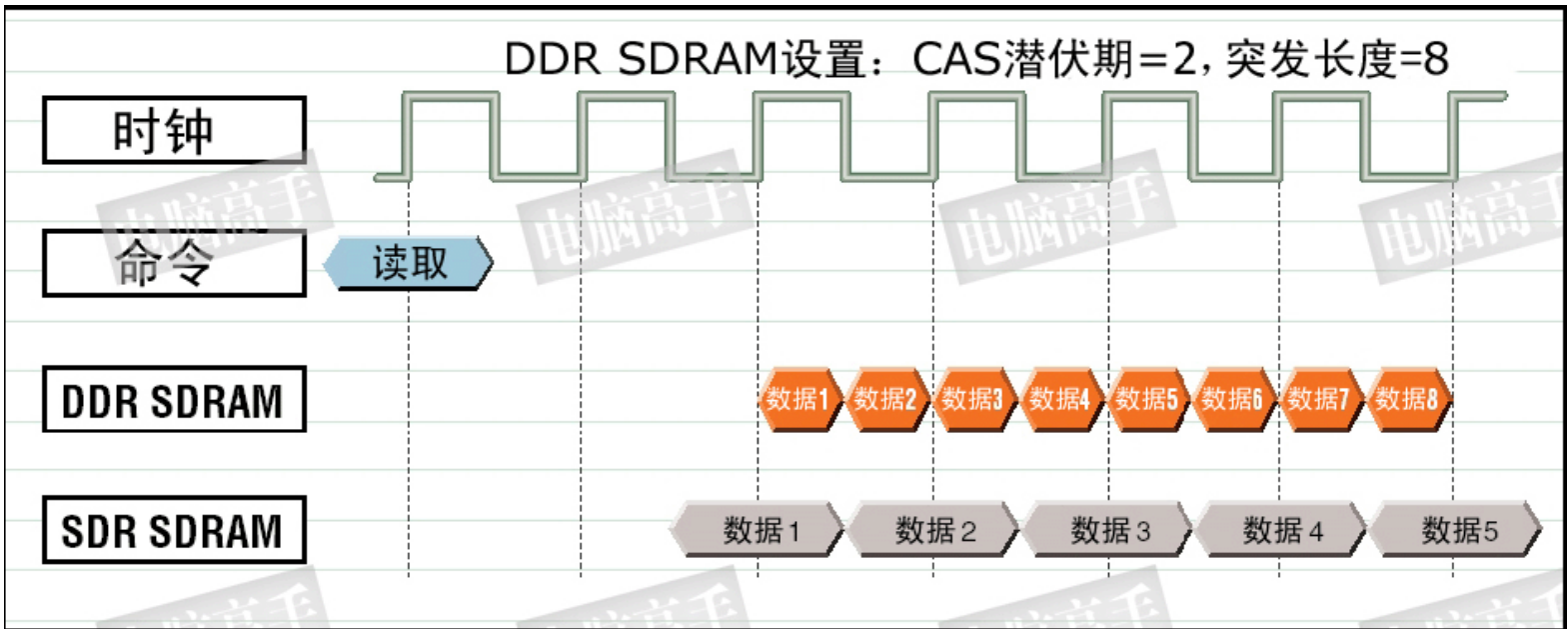


## 第二章 DDR SDRAM 的原理和时序

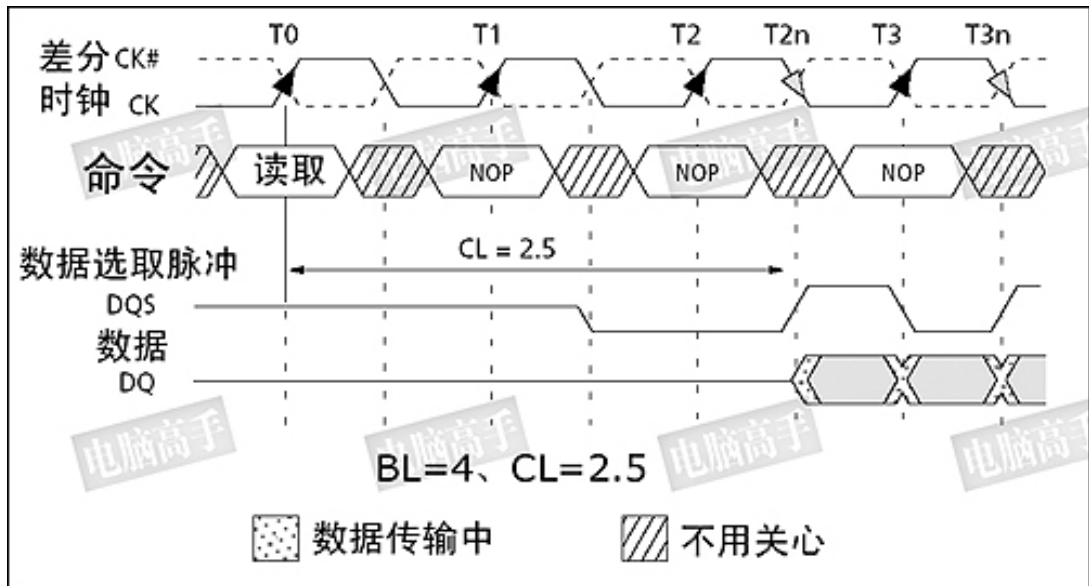
DDR SDRAM 全称为 Double Data Rate SDRAM，中文名为“双倍数据流 SDRAM”。DDR SDRAM 在原有的 SDRAM 的基础上改进而来。也正因为如此，DDR 能够凭借着转产成本优势来打败昔日的对手 RDRAM，成为当今的主流。由于 SDRAM 的结构与操作在前一章已有详细阐述，所以本文只着重讲讲 DDR 的原理和 DDR SDRAM 相对于传统 SDRAM（又称 SDR SDRAM）的不同。



DDR SDRAM可在一个时钟周期内传送两次数据

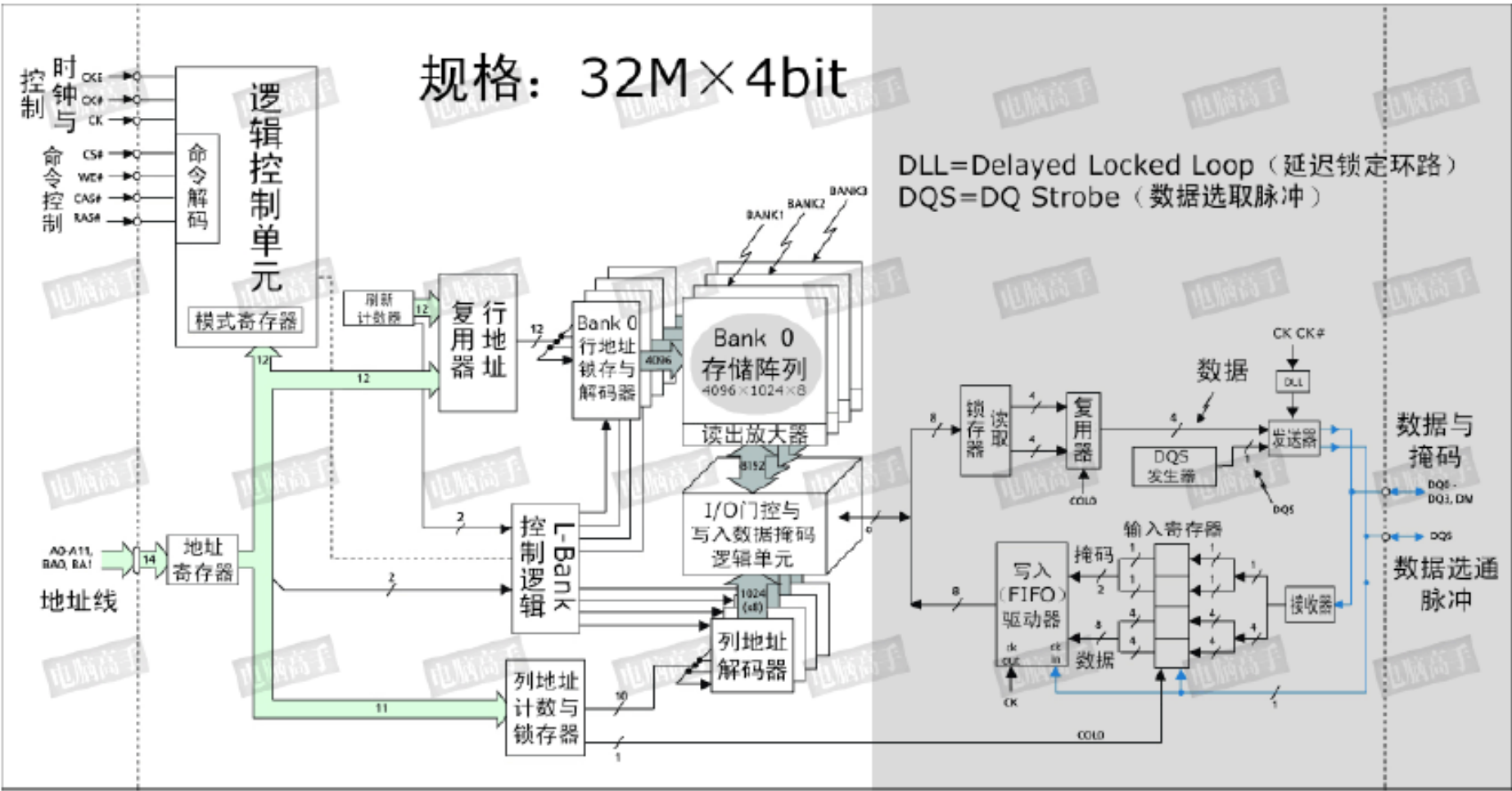
### 2.1 DDR 的基本原理

有很多文章都在探讨 DDR 的原理，但似乎也不得要领，甚至还带出一些错误的观点。首先我们看看一张 DDR 正规的时序图。



DDR SDRAM 读操作时序图

从中可以发现它多了两个信号：CLK#与DQS，CLK#与正常CLK时钟相位相反，形成差分时钟信号。而数据的传输在CLK与CLK#的交叉点进行，可见在CLK的上升与下降沿（此时正好是CLK#的上升沿）都有数据被触发，从而实现DDR。在此，我们可以说通过差分信号达到了DDR的目的，甚至讲CLK#帮助了第二个数据的触发，但这只是对表面现象的简单描述，从严格的定义上讲并不能这么说。之所以能实现DDR，还要从其内部的改进说起。



DDR 内存芯片的内部结构图，注意比较上文中 SDRAM 的结构图

这也是一颗128Mbit的内存芯片，标称规格也与前文的SDRAM一样为32 x 4bit。从图中可以看出来，白色区域内与SDRAM的结构基本相同，但请注意灰色区域，这是与SDRAM的不同之处。

首先就是内部的L-Bank规格。SDRAM中L-Bank存储单元的容量与芯片位宽相同，但在DDR SDRAM中并不是这样，存储单元的容量是芯片位宽的一倍，所以在此不能再套用讲解SDRAM时“芯片位宽=存储单元容量”的公式了。也因此，真正的行、列地址数量也与同规格SDRAM不一样了。

以本芯片为例，在读取时，L-Bank在内部时钟信号的触发下一次传送8bit的数据给读取锁存器，再分成两路4bit数据传给复用器，由后者将它们合并为一路4bit数据流，然后由发送器在DQS的控制下在外部时钟上升与下降沿分两次传输4bit的数据给北桥。这样，如果时钟频率为100MHz，那么在I/O端口处，由于是上下沿触发，那么就是传输频率就是200MHz。

现在大家基本明白DDR SDRAM的工作原理了吧，这种内部存储单元容量（也可以称为芯片内部总线位宽）=2 x 芯片位宽（也可称为芯片I/O总线位宽）的设计，就是所谓的两位预取（2-bit Prefetch），有的公司则贴切的称之为2-n Prefetch（n代表芯片位宽）。

## 2.2 DDR SDRAM 与 SDRAM 的不同

DDR SDRAM 与 SDRAM 的不同主要体现在以下几个方面，如下表。

DDR SDRAM 与 SDRAM 的主要不同对比表

内存类型 比较项	SDRAM	DDR SDRAM
命令		
全页式突发传输	支持	不支持
时钟信号挂起	支持	不支持
读出数据屏蔽	支持	不支持
写入数据屏蔽	支持	支持
功能		
时钟	单一时钟	差分时钟
预取设计	1-bit	2-bit
数据传输率	1/时钟周期	2/时钟周期
CAS 潜伏期	2、3	1.5、2、2.5、3
写入潜伏期	0	可变
突发长度	1、2、4、8、全页	2、4、8
延迟锁定回路	可选	工作时必需
自动刷新间隔周期	固定	弹性设计（最大值与 SDRAM 的固定值相同）
数据选取脉冲	无	必需
封装与电气特性		
封装类型 (≥64Mbit)	TSOP-II 54pin (400mil)	TSOP-II 66pin (400mil)
		CSP 60pin
工作电压	3.3V (LVTTTL 接口)	2.5V (SSTL_2 接口)
模组	168pin DIMM	184pin DIMM

注：LVTTTL=Low Voltage Transistor-Transistor Logic（低电压晶体管-晶体管逻辑电路）、SSTL=Stub Series Terminated Logic（短线串联终止逻辑电路）

提示：TSOP-II 与 CSP
TSOP-II：是指小外形薄型封装（Thin Small Outline Package）的第二种方式，引脚在封装的长边两侧，TSOP-I 的引脚则在短边的两侧。
CSP：是指芯片尺寸封装（Chip Scale Package），其封装尺寸与芯片核心尺寸基本相同，所以称 CSP，其内核面积与封装面积的比例约为 1:1.1，凡是符合这一标准的封装都可称之为 CSP。

DDR SDRAM 与 SDRAM 一样，在开机时也要进行 MRS，不过由于操作功能的增多，DDR SDRAM 在 MRS 之前还多了一 EMRS 阶段（Extended Mode Register Set，扩展模式寄存器设置），这个扩展模式寄存器控制着 DLL 的有效/禁止、输出驱动强度、QFC 有效/无效等。

提示与误区：QFC 的含义与作用
QFC 是指 FET Switch Controller（FET 开关控制），低电平有效。用于借助外部 FET（场效应管）开关控制模组上芯片间的相互隔离，没有读写操作时进入隔离状态，以确保芯片间不受相互干扰。QFC 是一个特选功能，厂商都是在接到芯片买家的指定要求后，才在芯片中加入这一功能，并且需要在模装配时进行相关的设计改动（如增加 VddQ 的上拉电阻），所以 DIY 市场上几乎很少见到支持这一功能的 DDR 内存。而在 2002 年 5 月，JEDEC 最新发布的 DDR 规范中，已经不在有 QFC 的定义，而且即使有 FET 开关也将由北桥控制（此时是用来隔离模组的），因此 QFC 已经成为历史。可是，在很多“深入”性介绍中，都将 QFC 认为是一个必要的过程，这显然是错的。

由于 EMRS 与 MRS 的操作方法与 SDRAM 的 MRS 大同小异，在此就不再列出具体的模式表了，有兴趣的话可查看相关的 DDR 内存资料。下面我们就着重说说 DDR SDRAM 的新设计与新功能。

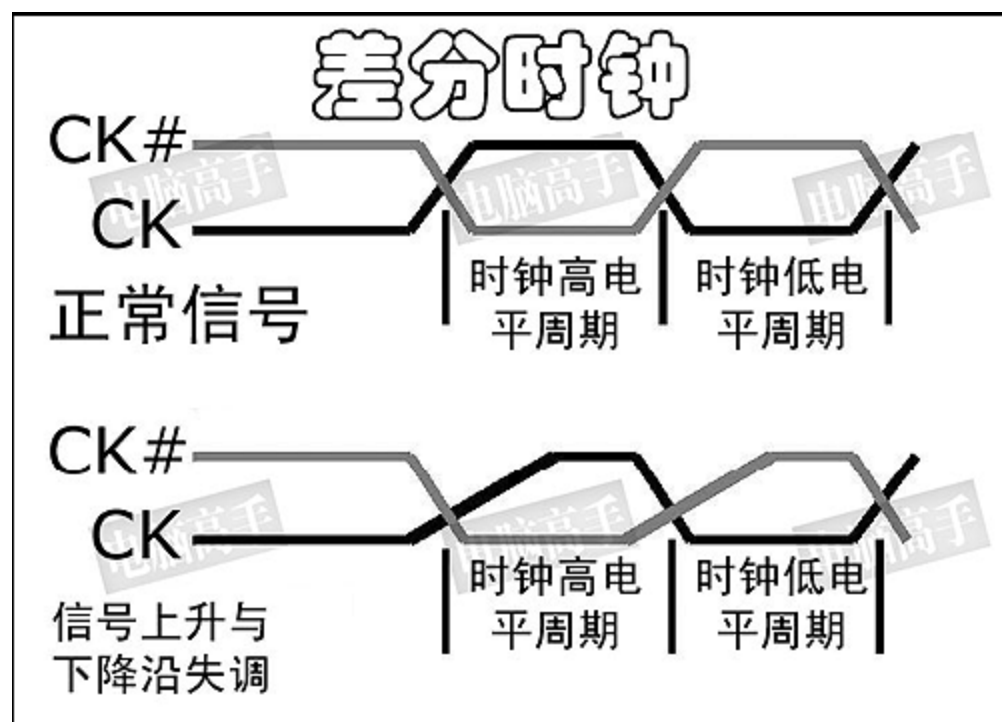
误区：CSP 与 uBGA、WBGA、TinyBGA、FBGA 等是不同的封装技术
实际上，CSP 只是一种封装标准/类型，不涉及具体的封装技术，只要达到它的尺寸标准都可称之为 CSP 封装。近几年出现的 uBGA、WBGA、TinyBGA、FBGA 小型芯片封装技术则是 CSP 的具体表现形式（其实都是 BGA 封装技术的一种），由此可以看出 CSP 并没有固定的封装技术，它自己更不是一个封装技术，只要厂商愿意或有实力，可以开发出更多的符合 CSP 标准的封装技术。

## 2.3 差分时钟

差分时钟（参见上文“DDR SDRAM 读操作时序图”）是 DDR 的一个必要设计，但 CK#的作用，并不能理解为第二个触发时钟（你可以在讲述 DDR 原理时简单地这么比喻），而是起到触发时钟校准的作用。

由于数据是在 CK 的上下沿触发，造成传输周期缩短了一半，因此必须要保证传输周期的稳定以确保数据的正确传输，这就要求 CK 的上下沿间距要有精确的控制。但因为温度、电阻性能的改变等原因，CK 上下沿间距可能发生变化，此时与其反相的 CK#就起到纠正的作用（CK 上升快下降慢，CK#则是上升慢下降快）。而由于上下沿触发的原因，也使 CL=1.5 和 2.5 成为可能，并容易实现。





与 CK 反相的 CK#保证了触发时机的准确性

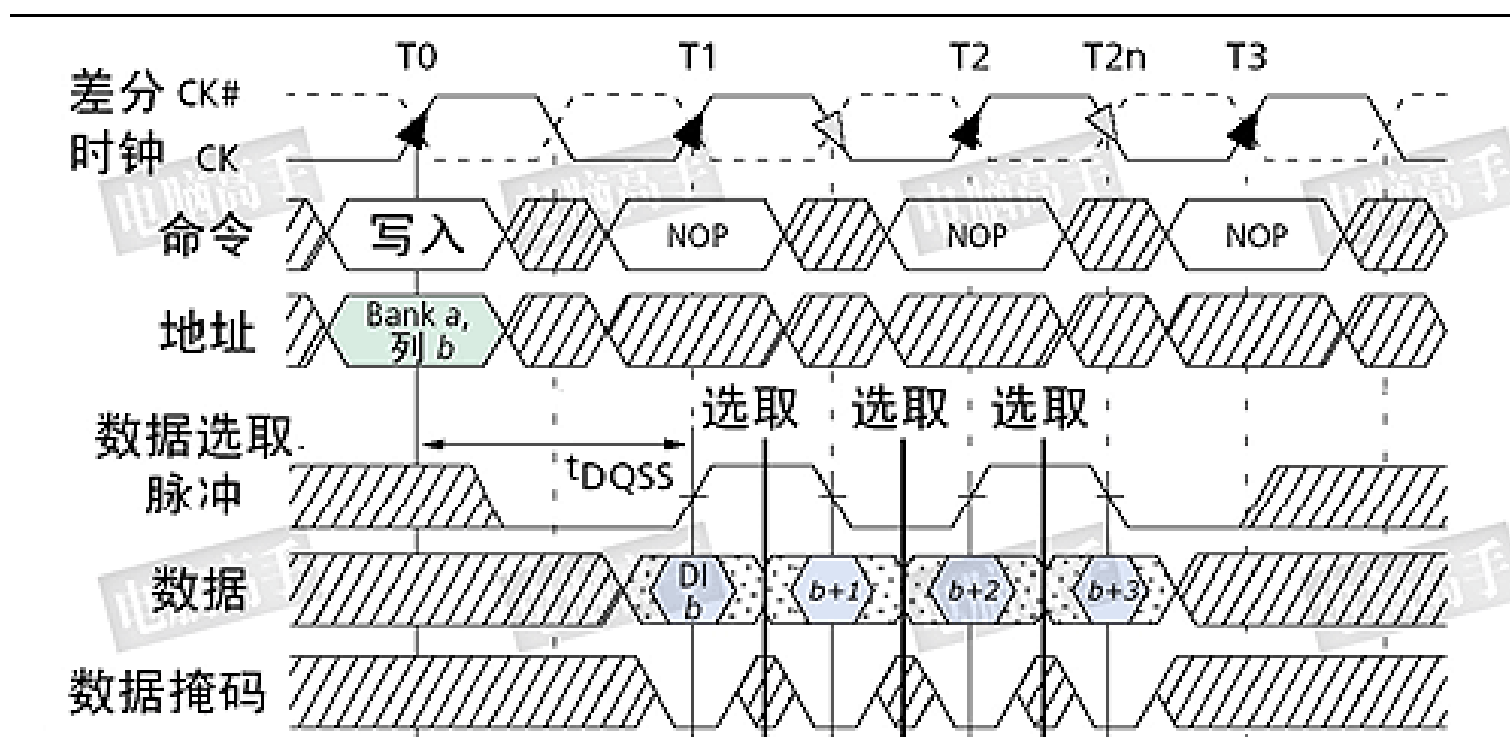
## 2.4 数据选取脉冲（DQS）

DQS 是 DDR SDRAM 中的重要功能，它的功能主要用来在一个时钟周期内准确的区分出每个传输周期，并便于接收方准确接收数据。每一颗芯片都有一个 DQS 信号线，它是双向的，在写入时它用来传送由北桥发来的 DQS 信号，读取时，则由芯片生成 DQS 向北桥发送。完全可以说，它就是数据的同步信号。

在读取时，DQS 与数据信号同时生成（也是在 CK 与 CK#的交叉点）。而 DDR 内存中的 CL 也就是从 CAS 发出到 DQS 生成的间隔，数据真正出现在数据 I/O 总线上相对于 DQS 触发的时间间隔被称为 tAC。注意，这与 SDRAM 中的 tAC 的不同。实际上，DQS 生成时，芯片内部的预取已经完毕了，tAC 是指上文结构图中灰色部分的数据输出时间，由于预取的原因，实际的数据传出可能会提前于 DQS 发生（数据提前于 DQS 传出）。由于是并行传输，DDR 内存对 tAC 也有一定的要求，对于 DDR266，tAC 的允许范围是 $\pm 0.75\text{ns}$ ，对于 DDR333，则是 $\pm 0.7\text{ns}$ ，有关它们的时序图示见前文，其中 CL 里包含了一段 DQS 的导入期。

DQS 在读取时与数据同步传输，那么接收时也是以 DQS 的上下沿为准吗？不，如果以 DQS 的上下沿区分数据周期的危险很大。由于芯片有预取的操作，所以输出时的同步很难控制，只能限制在一定的时间内，数据在各 I/O 端口的出现时间可能有快有慢，会与 DQS 有一定的间隔，这也就是为什么要有一个 tAC 规定的原因。而在接收方，一切必须保证同步接收，不能有 tAC 之类的偏差。这样

在写入时，芯片不再自己生成 DQS，而以发送方传来的 DQS 为基准，并相应延后一定的时间，在 DQS 的中部为数据周期的选取分割点（在读取时分割点就是上下沿），从这里分隔开两个传输周期。这样做的好处是，由于各数据信号都会有一个逻辑电平保持周期，即使发送时不同步，在 DQS 上下沿时都处于保持周期中，此时数据接收触发的准确性无疑是最高的。



在写入时，以 DQS 的高/低电平期中部为数据周期分割点，而不是上/下沿，但数据的接收触发仍为 DQS 的上/下沿。

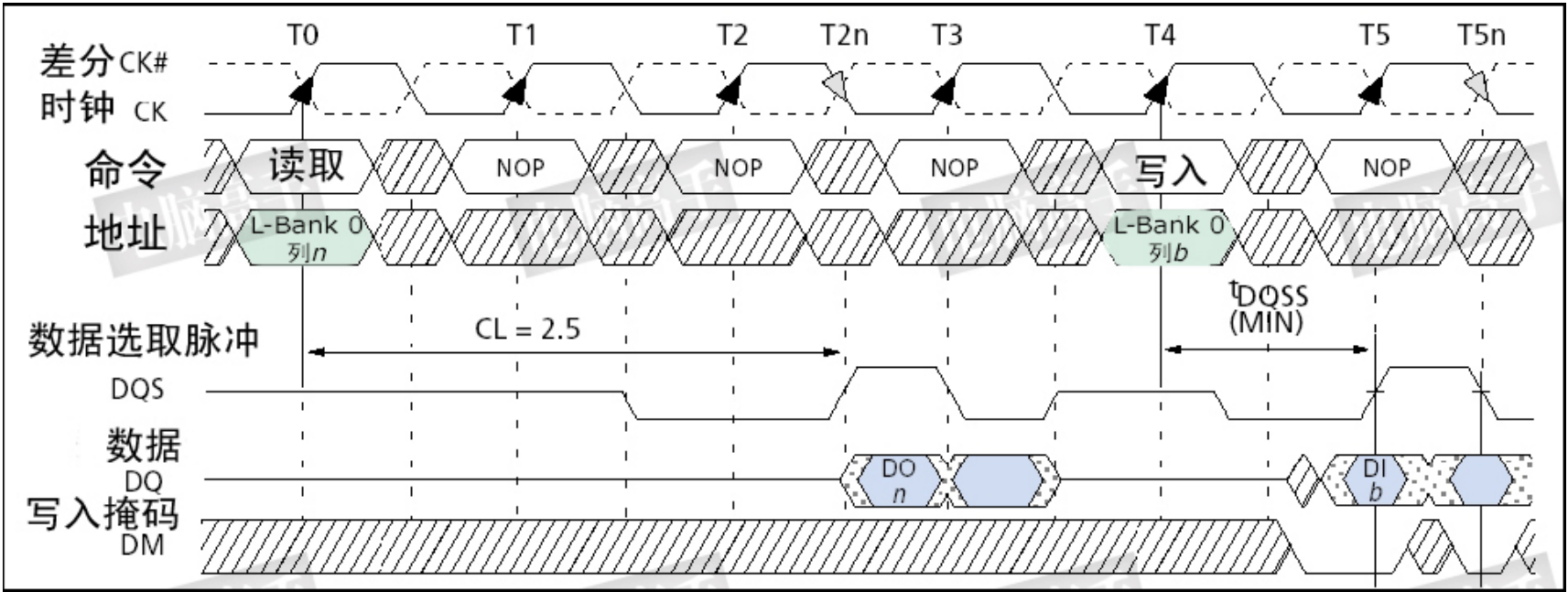
## 2.5 写入延迟

在上面的 DQS 写入时序图中，可以发现写入延迟已经不是 0 了，在发出写入命令后，DQS 与写入数据要等一段时间才会送达。这个周期被称为 DQS 相对于写入命令的延迟时间（ $t_{DQSS}$ ，WRITE Command to the first corresponding rising edge of DQS）。

为什么要有这样的延迟设计呢？原因也在于同步，毕竟一个时钟周期两次传送，需要很高的控制精度，它必须要等接收方做好充分的准备才行。 $t_{DQSS}$  是 DDR 内存写入操作的一个重要参数，太短的话恐怕接受有误，太长则会造成总线空闲。 $t_{DQSS}$  最短不能小于 0.75 个时钟周期，最长不能超过 1.25 个时钟周期。

有人可能会说，如果这样，DQS 不就与芯片内的时钟不同步了吗？对，正常情况下， $t_{DQSS}$  是一个时钟周期，但写入时接受方的时钟只用来控制命令信号的同步，而数据的接受则完全依靠 DQS 进行

同步，所以 DQS 与时钟不同步也无所谓。不过，tDQSS 产生了一个不利影响——读后写操作延迟的增加，如果 CL=2.5，还要在 tDQSS 基础上加入半个时钟周期，因为命令都要在 CK 的上升沿发出。



当 CL=2.5 时，读后写的延迟将为 tDQSS+0.5 个时钟周期（图中 BL=2）

另外，DDR 内存的数据真正写入由于要经过更多步骤的处理，所以写回时间（tWR）也明显延长，一般在 3 个时钟周期左右，而在 DDR-II 规范中更是将 tWR 列为模式寄存器的一项，可见它的重要性。

**误区：DDR SDRAM 各种延迟与潜伏期的单位时间减半**

一些文章认为，DDR 使数据传输率加倍，那么与之相关的延迟与潜伏期的单位时间也减半，比如 DDR-266 内存，tRCD、CL、tRP 的单位周期为 3.75ns，比 PC133 内存少了一半。这是严重的概念性错误，从我们列举的时序图中可以看出，tRCD、CL、tRP 是以时钟信号来界定的，不能用传输周期去表示，否则 CL=2.5 的参考基准是什么？对于 DDR-266，时钟频率是 133MHz，时钟周期仍是 7.5，和 PC133 的标准一样。那些文章的作者显然是将时钟周期与传输周期弄混了

## 2.6 突发长度与写入掩码

在 DDR SDRAM 中，突发长度只有 2、4、8 三种选择，没有了随机存取的操作（突发长度为 1）和全页式突发。这是为什么呢？因为 L-Bank 一次就存取两倍于芯片位宽的数据，所以芯片至少也要进行两次传输才可以，否则内部多出来的数据怎么处理？而全页式突发事实证明在 PC 内存中是很难用得上的，所以被取消也不稀奇。

但是，突发长度的定义也与 SDRAM 的不一样了（见本章节最前那幅 DDR 简示图），它不再指所连续寻址的存储单元数量，而是指连续的传输周期数，每次是一个芯片位宽的数据。

对于突发写入，如果其中有不存入的数据，仍可以运用 DM 信号进行屏蔽。DM 信号和数据信号同时发出，接收方在 DQS 的上升与下降沿来判断 DM 的状态，如果 DM 为高电平，那么之前从 DQS 中部选取的数据就被屏蔽了。有人可能会觉得，DM 是输入信号，意味着芯片不能发出 DM 信号给北桥作为屏蔽读取数据的参考。其实，该读哪个数据也是由北桥芯片决定的，所以芯片也无需参与北桥的工作，哪个数据是有用的就留给北桥自己去选吧。

## 2.7 延迟锁定回路（DLL）

DDR SDRAM 对时钟的精确性有着很高的要求，而 DDR SDRAM 有两个时钟，一个是外部的总线时钟，一个是内部的工作时钟，在理论上 DDR SDRAM 这两个时钟应该是同步的，但由于种种原因，如温度、电压波动而产生延迟使两者很难同步，更何况时钟频率本身也有不稳定的情况（SDRAM 也有内部时钟，不过因为它的工作/传输频率较低，所以内外同步问题并不突出）。

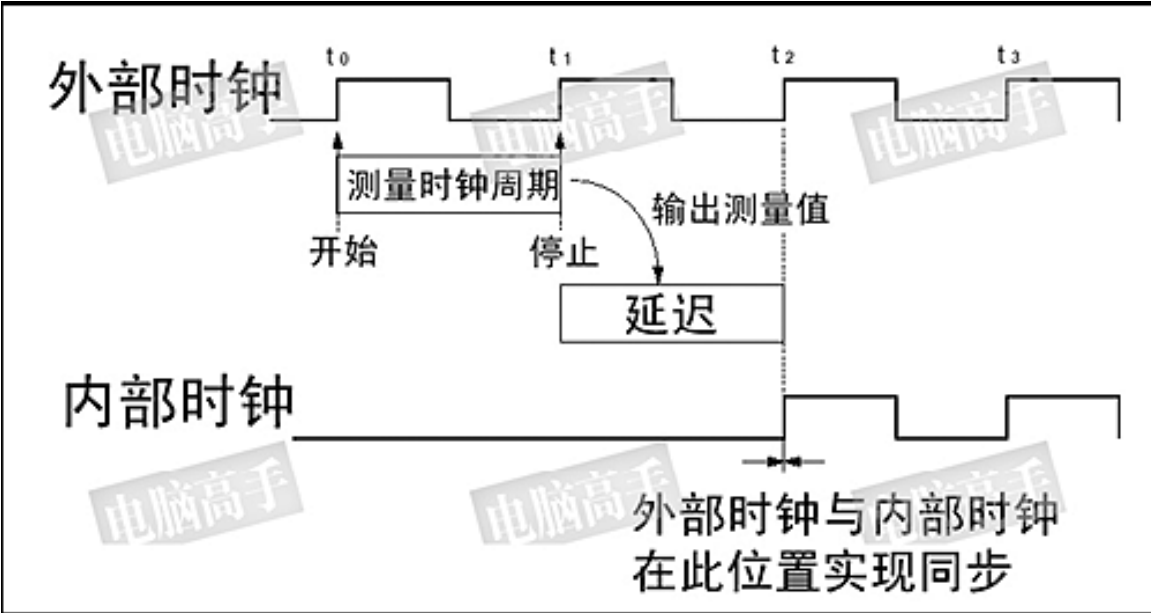
DDR SDRAM 的  $t_{AC}$  就是因为内部时钟与外部时钟有偏差而引起的，它很可能造成因数据不同步而产生错误的恶果。实际上，不同步就是一种正/负延迟，如果延迟不可避免，那么若是设定一个延迟值，如一个时钟周期，那么内外时钟的上升与下降沿还是同步的。鉴于外部时钟周期也不会绝对统一，所以需要根据外部时钟动态修正内部时钟的延迟来实现与外部时钟的同步，这就是 DLL 的任务。

DLL 不同于主板上的 PLL，它不涉及频率与电压转换，而是生成一个延迟量给内部时钟。目前 DLL 有两种实现方法，一个是时钟频率测量法（CFM，Clock Frequency Measurement），一个是时钟比较法（CC，Clock Comparator）。

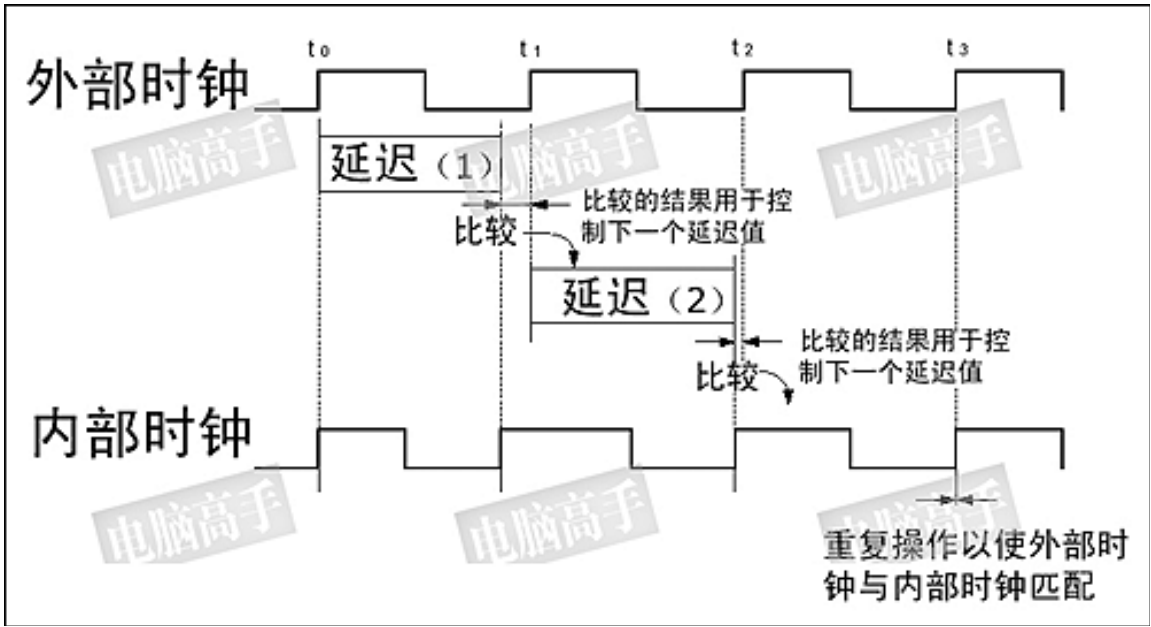
CFM 是测量外部时钟的频率周期，然后以此周期为延迟值控制内部时钟，这样内外时钟正好就相差了一个时钟周期，从而实现同步。DLL 就这样反复测量反复控制延迟值，使内部时钟与外部时钟保持同步。

CC 的方法则是比较内外部时钟的长短，如果内部时钟周期短了，就将所少的延迟加到下一个内部时钟周期里，然后再与外部时钟做比较，若是内部时钟周期长了，就将多出的延迟从下一个内部时钟中刨除，如此往复，最终使内外时钟同步。





CFM 式 DLL 工作示意图



CC 式 DLL 工作示意图

CFM 与 CC 各有优缺点，CFM 的校正速度快，仅用两个时钟周期，但容易受到噪音干扰，并且如果测量失误，则内部的延迟就永远错下去了。CC 的优点则是更稳定可靠，如果比较失败，延迟受影响的只是一个数据（而且不会太严重），不会涉及到后面的延迟修正，但它的修正时间要比 CFM 长。DLL 功能在 DDR SDRAM 中可以被禁止，但仅限于除错与评估操作，正常工作状态是自动有效的。

误区：DLL 是实现 DDR 传输的关键
“DDR 内存通过内部的 DLL 延时锁相环提供精确的时钟定位，这样就可以在每个时钟周期的上升沿和下降沿传输数据”。“DDR 使用了 DLL 来提供一个数据滤波信号 DQS 来选取数据”。
以上是较为流行的对 DDR 工作原理的两种解释，现在大家能看出错误所在吗？两者都把 DLL 的功能夸大了，DLL 只是一个重要的辅助校准设计，与能否实现双沿触发没有关系，它只是保证数据的输出尽量与外部时钟同步。后者则是概念性错误，从 DDR 内存结构图可看出，DQS 不是 DLL 生成的，只是由 DLL 保证其与外部时钟的同步，DQS 由单独的 DQS 发生器生成。

### 第三章 DDR-II 的原理和新技术

作为 DDR 的接班人，DDR-II 在规范制定之初就引起了广泛的关注，进入 2002 年，三星、Elpida、Hynix、Micron 等都相继发布了 DDR-II 芯片（最早由三星在 5 月 28 日发布），让人觉得 DDR-II 突然和我们近了。可是，DDR-II 规范却一直没有正式公开，在 JEDEC 上仍只有一篇 ATi 技术人员写的，在目前看来有些内容都已过时的简要介绍。

原来，DDR-II 标准到 2002 年 10 月完成度也没有达到 100%（厂商透露大约为 95%），而上述厂商所推出的芯片也在不断的修改中，预计正式的规范将在明年第一季度推出。不过，DDR-II 的主体设计已经完成，不会有大的改动，所以通过这些“试验性”芯片，我们仍可掌握 DDR-II 的主要信息。

DDR-II 相对于 DDR 的主要改进如下表

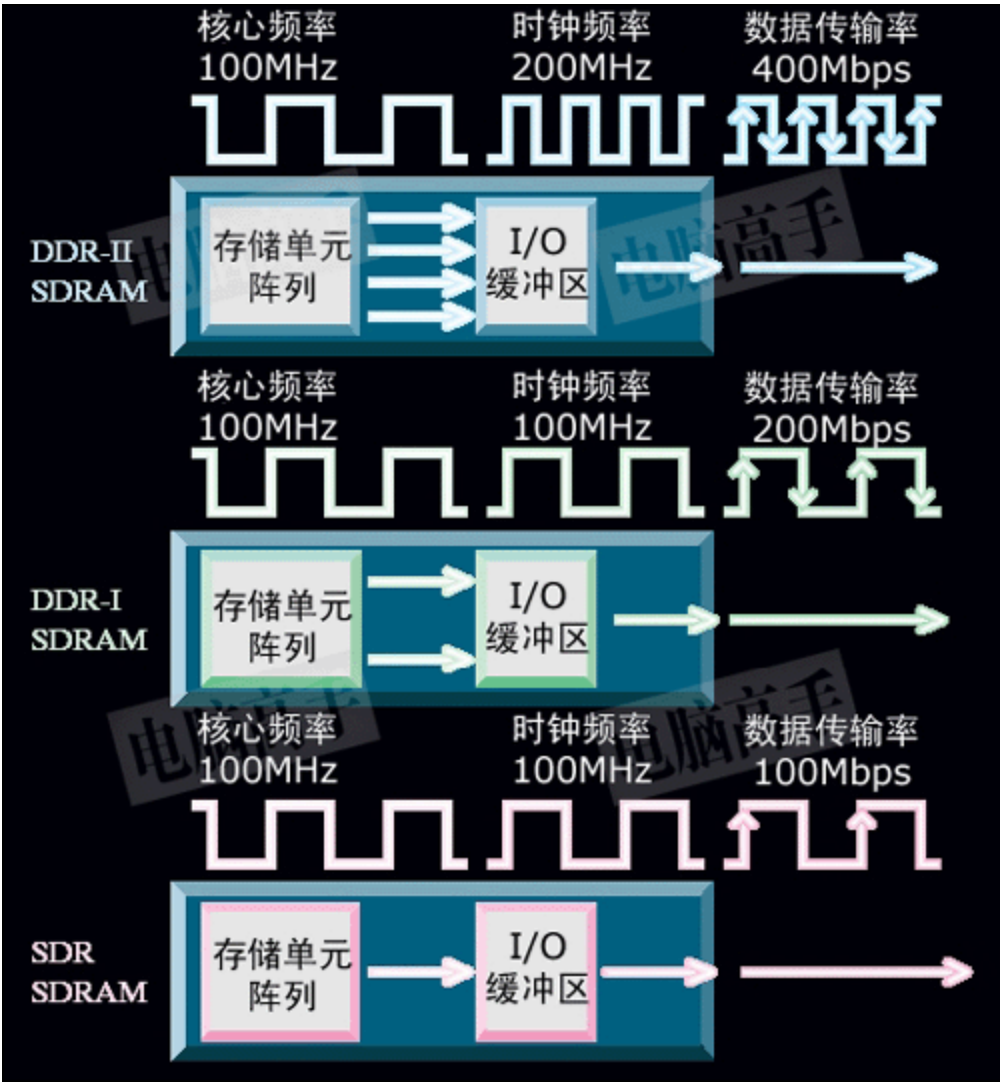
项目		DDR-II SDRAM	DDR SDRAM
基本设计	时钟频率（MHz）	200/266/333	100/133/166/200
	数据传输率（Mbps）	400/533/667	200/266/333/400
	预取设计	4bit	2bit
	突发长度	4/8	2/4/8
	L-Bank 数量	最多 8 个	最多 4 个
	CL 值	3、4、5	1.5、2、2.5、3
	数据选取脉冲	差分数据选取脉冲	单数据选取脉冲
电气性能与封装	工作电压	1.8V	2.5V
	接口标准	SSTL_18	SSTL_2
	消耗功率	304mW（最大） 533Mbps	418mW（最大） 266Mbps
	封装	CSP（FBGA）无铅封装， 60/64/68/84/92pin	TSOP-II（66Pin）、 CSP（60pin）
	模组标准	240pin DIMM	184pin DIMM
	系统最高 P-Bank 数量	4	8
功能	命令集	同 DDR	-
	基本时序定义	同 DDR	-
	新功能	ODT、OCD 调校、Posted CAS、AL	无

由于 DDR-II 相对 DDR-I 的设计变动并不大，因此很多操作就不在此详细介绍了，本文重点阐述 DDR-II 的一些重要变化。

### 3.1 DDR-II 内存结构

DDR-II 内存的预取设计是 4bit，通过 DDR 的讲述，大家现在应该知道是什么意思了吧。上文已经说过，SDRAM 有两个时钟，一个是内部时钟，一个是外部时钟。

在 SDRAM 与 DDR 时代，这两个时钟频率是相同的，但在 DDR-II 内存中，内部时钟变成了外部时钟的一半。以 DDR-II 400 为例，数据传输频率为 400MHz（对于每个数据引脚，则是 400Mbps/pin），外部时钟频率为 200MHz，内部时钟频率为 100MHz。因为内部一次传输的数据就可供外部接口传输 4 次，虽然以 DDR 方式传输，但数据传输频率的基准——外部时钟频率仍要是内部时钟的两倍才行。



DDR-II、DDR 与 SDRAM 的操作时钟比较

所以，当预取容量超过接口一次 DDR 的传输量时，内部时钟必须降低（除非数据传输不是 DDR 方式，而是一个时钟周期 4 次）。如果内部时钟也达到 200MHz，那外部时钟也要达到 400MHz，这会使成本有大幅度提高。因此，DDR-II 虽然实现了 4-bit 预取，但在实际效能上，与 DDR 是一样的。

在上面那幅比较图中，可以看出厂商们的一种误导，它虽然表示出在相同的核心频率下，DDR-II 达到了两倍于 DDR 的带宽，但前提是 DDR-II 的外部时钟频率也是 DDR 和 SDRAM 的两倍。在 DDR 的时钟频率已经达到 166/200MHz 的今天，再用 100MHz 去比较，显然意义不大。这点也请大家们注意识别，上图更多的是说明 DDR-II 内外时钟的差异。毕竟内部时钟由外部决定，所以外部时钟才是比较的根本基准。

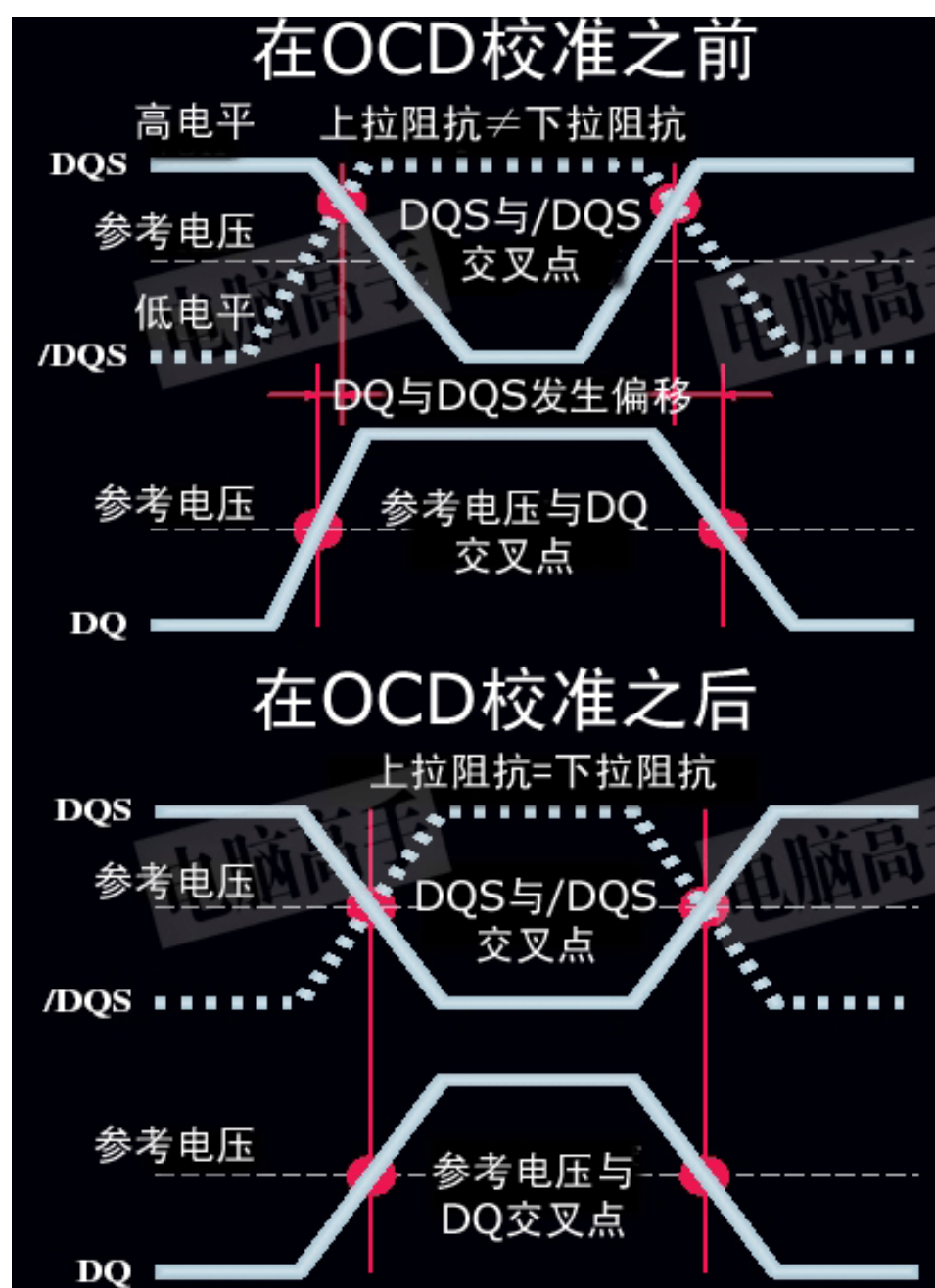
总之，现在大家要明确认识，在外部时钟频率相同的情况下，DDR-II 与 DDR 的带宽一样。

## 3.2 DDR-II 的新操作与新时序设计

### ● 3.2.1 片外驱动调校 (OCD, Off-Chip Driver)

DDR-II 内存在开机时也会有初始化过程，同时在 EMRS 中加入了新设置选项，由于大同小异，在此就不多说了。在 EMRS 阶段，DDR-II 加入了可选的 OCD 功能。

OCD 的主要用意在于调整 I/O 接口端的电压，来补偿上拉与下拉电阻值。目的是让 DQS 与 DQ 数据信号之间的偏差降低到最小。调校期间，分别测试 DQS 高电平/DQ 高电平，与 DQS 低电平/DQ 高电平时的同步情况，如果不满足要求，则通过设定突发长度的地址线来传送上拉/下拉电阻等级（加一档或减一档），直到测试合格才退出 OCD 操作。



OCD 的作用在于调整 DQS 与 DQ 之间的同步，以确保信号的完整与可靠性

不过，据一些厂商的技术人员介绍，一般情况下有 DQS#（差分 DQS 时）就基本可以保证同步的准确性，而且 OCD 的调整对其他操作也有一定影响，因此在普通台式机上不需要用 OCD 功能，它一般只会出现在高端产品中，如对数据完整性非常敏感的服务器等。

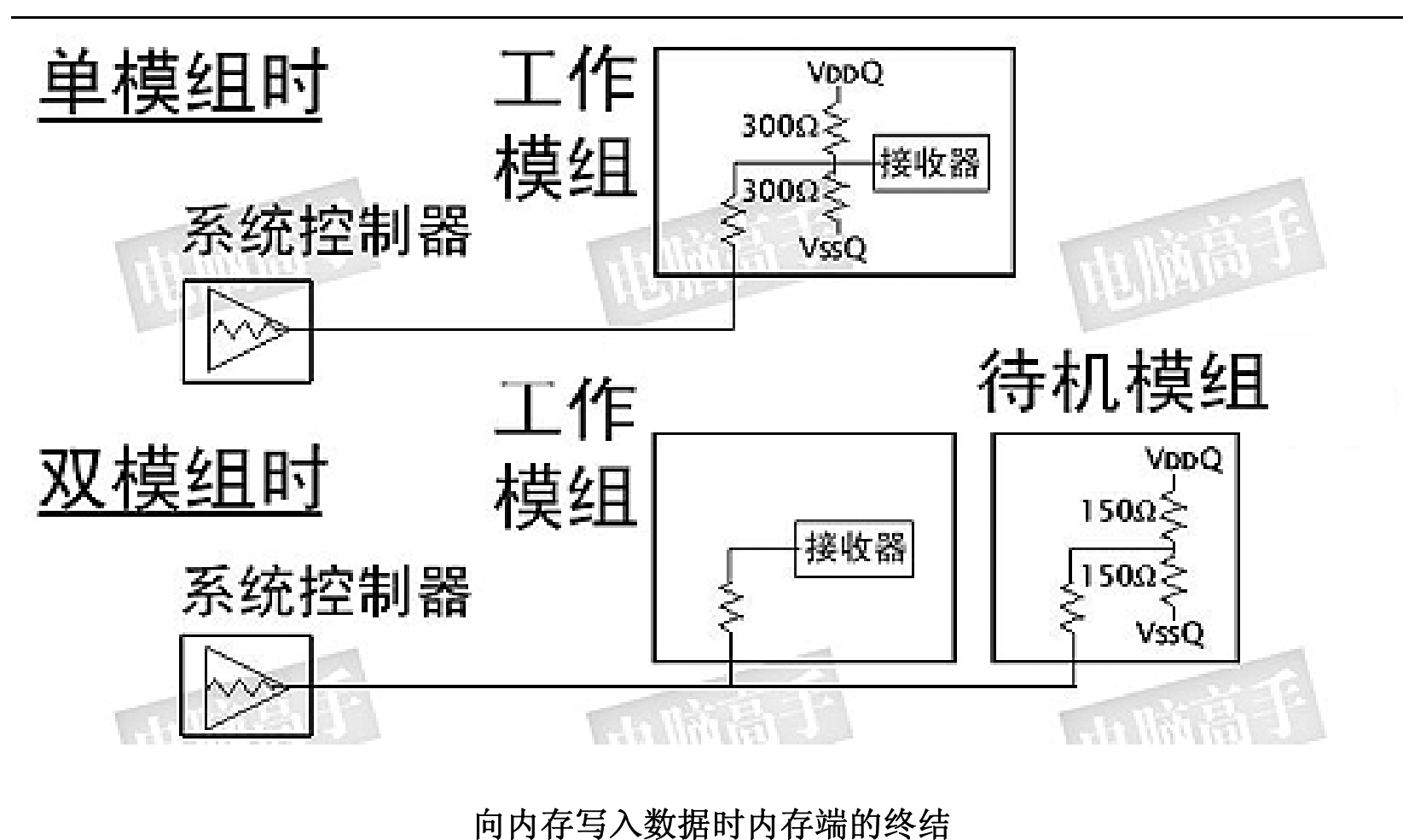
### ● 3.2.2 片内终结 (ODT, On-Die Termination)

所谓的终结，就是让信号被电路的终端吸收掉，而不会在电路上形成反射，造成对后面信号的影响。

在 DDR 时代，控制与数据信号的终结在主板上完成，每块 DDR 主板在 DIMM 槽的旁边都会有一个终结电压岛的设计，它主要由一排终结电阻构成。长期以来，这个电压岛一直是 DDR 主板设计上的一个难点。而 ODT 的出现，则将这个难点消灭了。

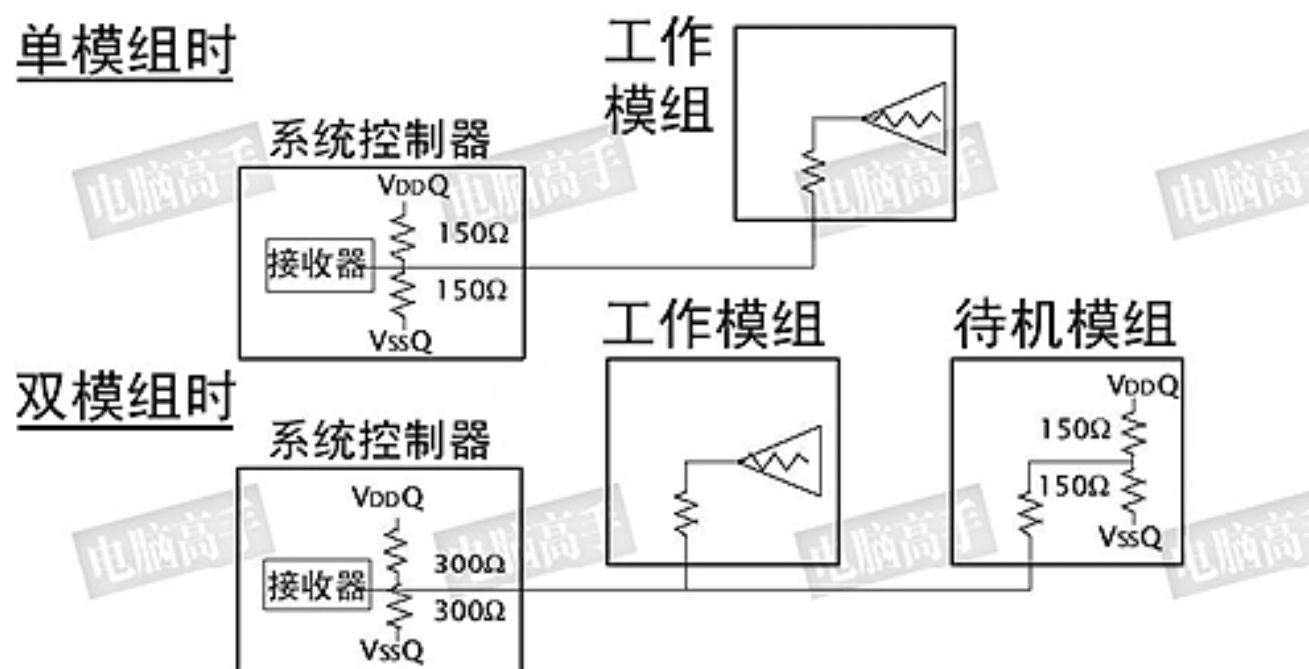
顾名思义，ODT 就是将终结电阻移植到了芯片内部，主板上不再有终结电路。ODT 的功能与禁止由北桥芯片控制，ODT 所终结的信号包括 DQS、RDQS（为 8bit 位宽芯片增设的专用 DQS 读取信号，主要用来简化一个模组中同时使用 4 与 8bit 位宽芯片时的控制设计）、DQ、DM 等。需要不需要该芯片进行终结由北桥控制。

那么具体的终结操作如何实现呢？首先要确定系统中有几条模组，并因此来决定终结的等效电阻值，有 150 和 75  $\Omega$  两档，这一切由北桥在开机进行 EMRS 时进行设置。



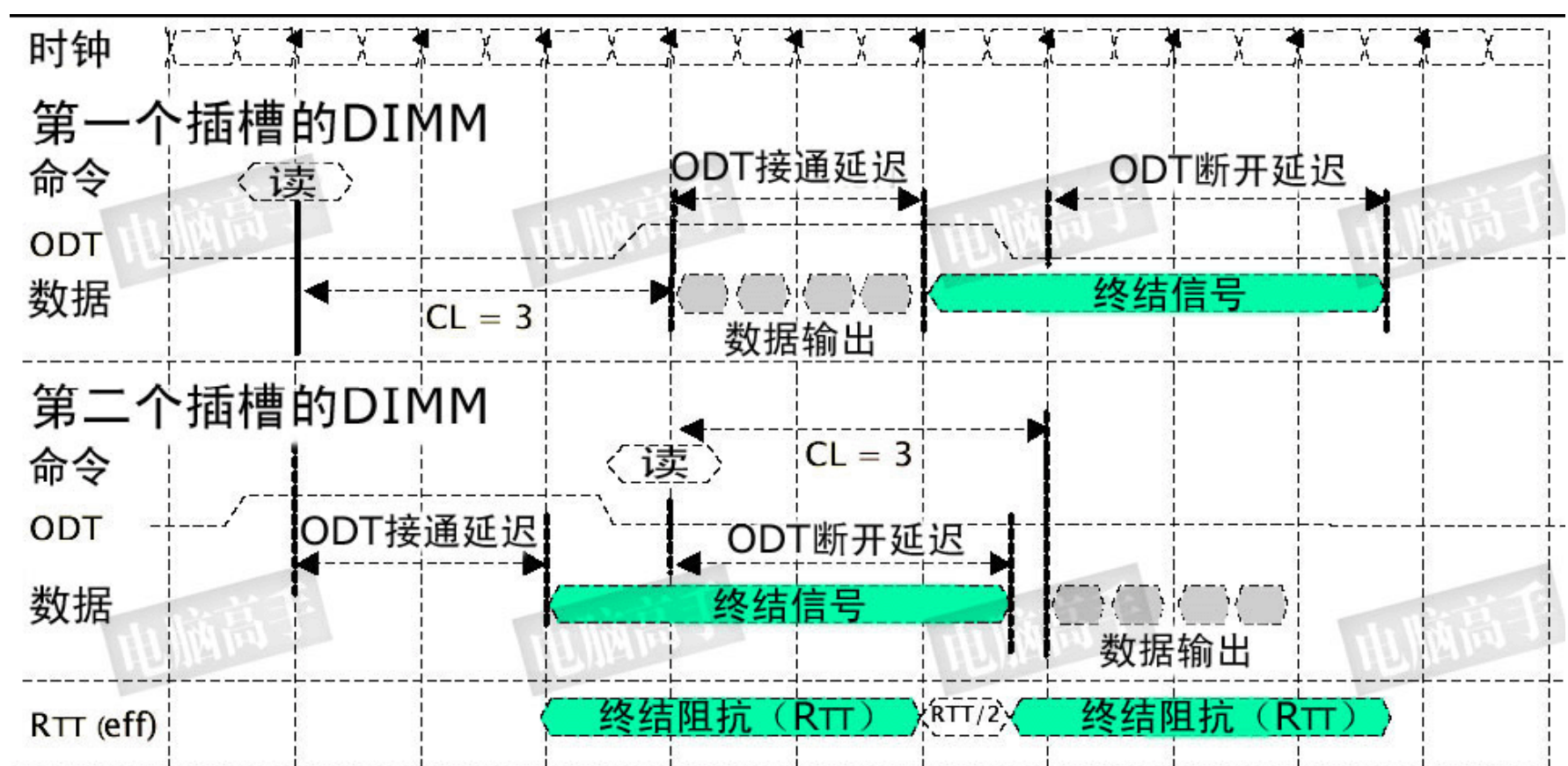
在向内存写入时，如果只有一条 DIMM，那么这条 DIMM 就自己进行终结，终结电阻等效为 150  $\Omega$ 。如果为两条 DIMM，一条工作时，另一条负责终结，但等效电阻为 75  $\Omega$ 。





从内存读取数据时控制器端的终结

在从内存读出时，终结操作也将在北桥内进行，如果有两条 DIMM，不工作的那一条将会终结信号在另一方向的余波，等效电阻也因 DIMM 的数量而有两种设置



两个 DIMM 在交错工作中的 ODT 情况

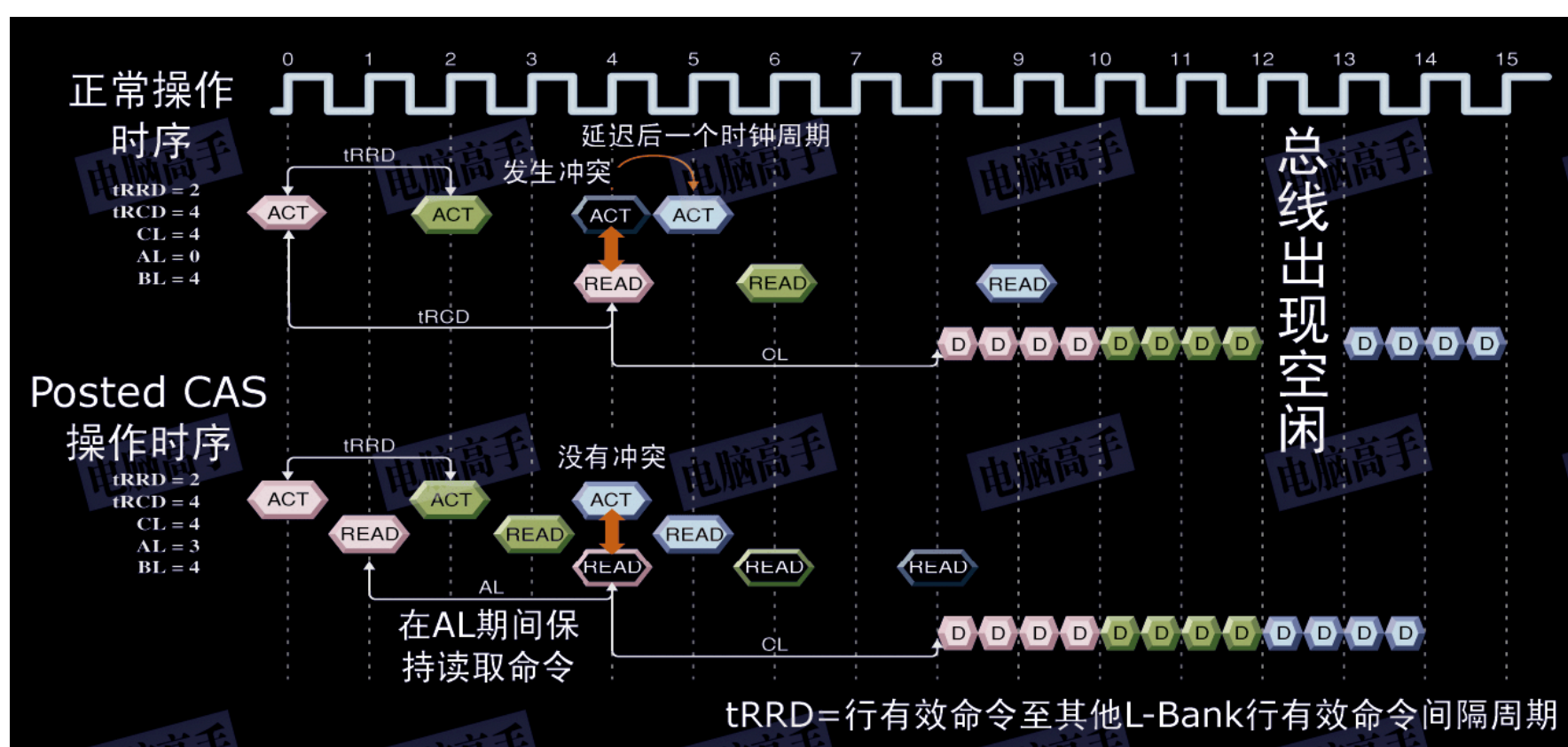
上图为两个 DIMM 在交错工作中的 ODT 情况，第一个模组工作时，第二个模组进行终结操作，等第二个模组工作时，第一个模组进行终结操作

现在我们应该基本了解了 ODT 的功能，它在很大程度上减少了内存芯片在读取时的 I/O 功率消耗，并简化了主板的设计，降低了主板成本。而且 ODT 也要比主板终结更及时有效，从而也成为了提高信号质量的重要功能，这有助于降低日后 DDR-II 进一步提速的难度。但是，由于为了确保信号的有效终结，终结操作期将会比数据传输期稍长，从而多占用一个时钟周期的时间而造成总线空闲。不过，有些厂商的技术人员称，通过精确设置  $t_{DQSS}$ ，可以避免出现总线空闲。

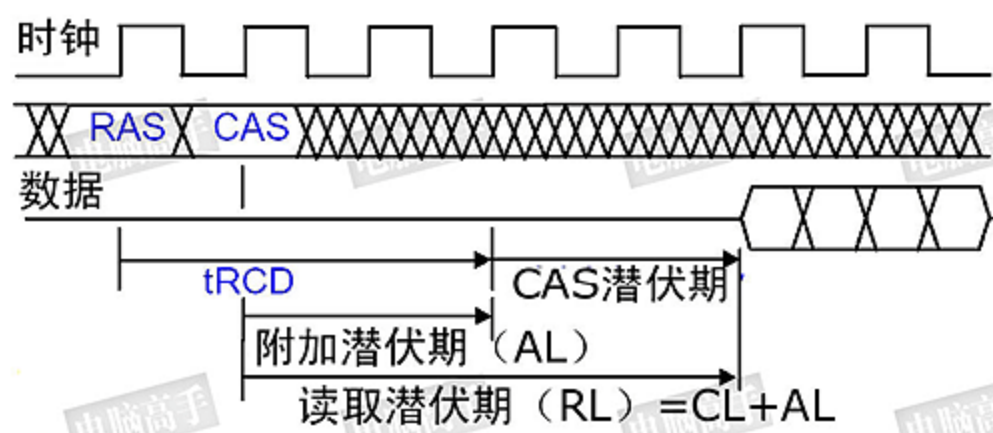
### ● 3.2.3 前置 CAS、附加潜伏期与写入潜伏期

前置 CAS (Posted CAS) 是为了解决 DDR 内存中指令冲突而设计的功能。它允许 CAS 信号紧随 RAS 发送，相对于以往的 DDR 等于将 CAS 前置了。

这样，地址线可以立刻空出来，便于后面的行有效命令发出，避免造成命令冲突而被迫延后的情况发生，但读/写操作并没有因此而提前，仍有要保证有足够的延迟/潜伏期，为此，DDR-II 引入了附加潜伏期的概念 (AL, Additive Latency)，与 CL 一样，单位为时钟周期数。AL+CL 被定义为读取潜伏期 (RL, Read Latency)，相应的，DDR-II 还对写入潜伏期 (WL, Write Latency) 制定了标准，WL 是指从写入命令发出到第一笔数据输入的潜伏期，不要将它和 tDQSS 弄混了，后者是指 DQS 而不是数据。按规定，WL=RL-1，即 AL+CL-1。



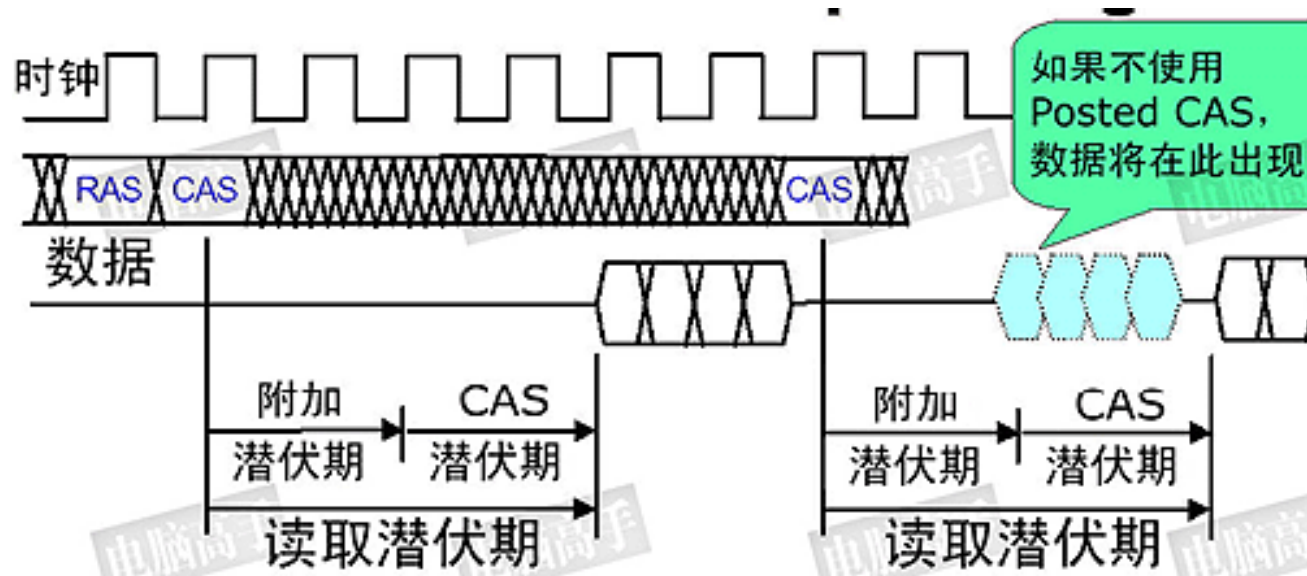
在没有前置 CAS 功能时，对其他 L-Bank 的寻址操作可能会因当前行的 CAS 命令占用地址线而延后，并使数据 I/O 总线出现空闲，当使用前置 CAS 后，消除了命令冲突并使数据 I/O 总线的利用率提高。



设置 Posted-CAS 后，必须附加潜伏期以保证应有延迟，此时读取潜伏期 (RL) 就等于 AL+CL，从中可以看出 AL 的值为 CL+tRCD-1



DDR-II 中 CL 最低值为 3, 最高为 5, 并且不再有 x.5 的设计, 而 AL 值则为 0-4。当 AL 设为 0 时, 前置 CAS 无效, 即为传统 DDR 模式的操作。不过前置 CAS 在解决命令冲突的时候也带来了新的问题——在背靠背式读取时, 仍将经过 AL+CL 的潜伏期才能读取数据, 比传统的只有 CL 相比, 读取的延迟反而增加了。因此, AL=0 是默认设置, 只有在那些读写命令非常频繁的操作场合, 才建议启动前置 CAS 功能 (如服务器等), 对于台式机用户, 前置 CAS 的优点不足以抵消其带来的不利影响。



由于有了 AL, 在同一行中进行再读取时, 在 CL 的基础上仍将增加 AL 造成的延迟, 从而影响了性能。

### 3.3 DDR-II 未来发展

#### ● 3.3.1 DDR-II 的发展计划

虽然目前多家厂商都推出了 DDR-II 内存芯片, 但从 DDR 官方组织 JEDEC 方面得到的信息表明, 距离 DDR-II 内存大规模上市还很遥远, 2004 年才会是 DDR-II 普通的阶段。而由于三星、南亚与 Micron 公司的大力推广, 这期间 JEDEC 很可能会接受 DDR-400 标准, 目前的争执主要在于能否在 DDR-I 的体系下保证 DDR-400 的可靠性。对此 (成为 JEDEC 正式标准), 三星与南亚公司都表示了很强的信心。

笔者认为, DDR-400 应该会获得认可, 毕竟市场上是有需要的, 而让市场去等一年的时间迎接 DDR-II 400 似乎并不现实。不过, 多通道技术在 DDR 领域里的普及, 可能也会改变 JEDEC 对认证 DDR-400 的想法, 但关键要看多通道的性价比能不能填补这一空档, 否则 DDR-400 就是一个最佳的选择。

三星是 DDR-400 的主推厂商, 但请注意 DDR-400 的电压变化, 它可能是引起兼容性问题的根源之

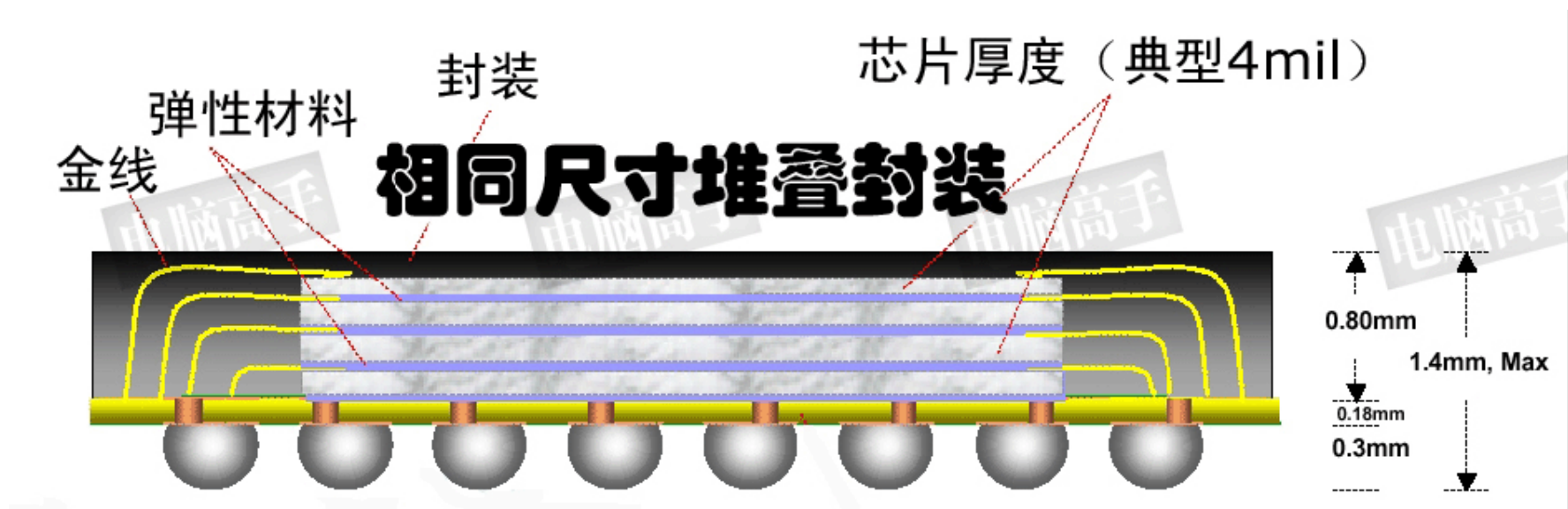
一也由于多通道的出现，市场上对 DDR-II 的渴望也并不大，毕竟它与同频的 DDR-I 内存的带宽一样。而从上文可以看出，DDR-II 相对于 DDR-I 的不同设计很多都集中在了如何在更高的工作（时钟）频率下保证数据的可靠。只有当 DDR-II 依靠自身的特有功能与设计来获得更高的时钟频率时，再配合多通道，才会真正拉开与 DDR-I 的距离，那时也就是 DDR-II 普及的开始。但笔者预测 DDR-II 400 将像 DDR-200 一样，注定是一个一出生就过时的标准，DDR-II 至少要从 533 开始流行。不过在目前情况下，我们还不必太在意 DDR-II 的进展情况，说句实话，它离我们还很远。今天的介绍只是让大家对其有一个大概的了解。

### ● 3.3.2 DDR-II 时代的封装技术

可以肯定的是 TSOP-II 将在 DDR-II 时代彻底退出内存封装市场。并且将会出现改良型的 CSP——WLP（Wafer Level Packaging，晶圆级封装），它是比 CSP 更为贴近芯片尺寸的封装方法，由于在晶圆上就做好了封装布线，因此在可靠性方面达到了更高的水平。不过，外在的模样仍与现在的 CSP 封装差不多，WLP 更多的改进是在其内部。

另外值得一提的是为了应付更高容量的需求而采用的 SiP 封装技术，它是 System-in-a-Package 的缩写，有时又称之为 Stacked Pakage，可以看作是一种集成封装技术。

它将多枚内存芯片核心堆叠在一起，然后统一封装成一颗芯片，在有限的面积内通过充分利用空间达到容量倍增的目的。SiP 并不是内存中专用的封装技术，原来是用于多种不同功能的芯片统一封装（如一颗嵌入式 CPU+DRAM 芯片）。



目前的 SiP 技术可以在 CSP 的基础上最多堆叠 4 枚内存芯片。

## 第四章 Rambus DRAM 的原理

谈起 DDR SDRAM 与 Rambus DRAM（简称 RDRAM）之间的恩怨，很多人现在还是津津乐道。的确，上一世纪末的内存大战虽胜负已分，但至今仍余波未平。在主流市场 DDR SDRAM 成为王者，RDRAM 则沦为“高端贵族”。

Rambus 公司于 1990 年 3 月成立，之后不久就有了 Rambus 的核心专利——RSL（Rambus Signaling Level，Rambus 发信电平技术）。Rambus 内存最早出现于 1995 年 12 月，那时它与任天堂 64（Nintendo64）游戏机一起发售，但名声不大。

从 1996 年 12 月开始，Rambus 与 Intel 合作开发，准备将 Rambus 推广到 PC 领域。到 Rambus 内存真正亮相于 PC 市场时已经是 1999 年 11 月了。

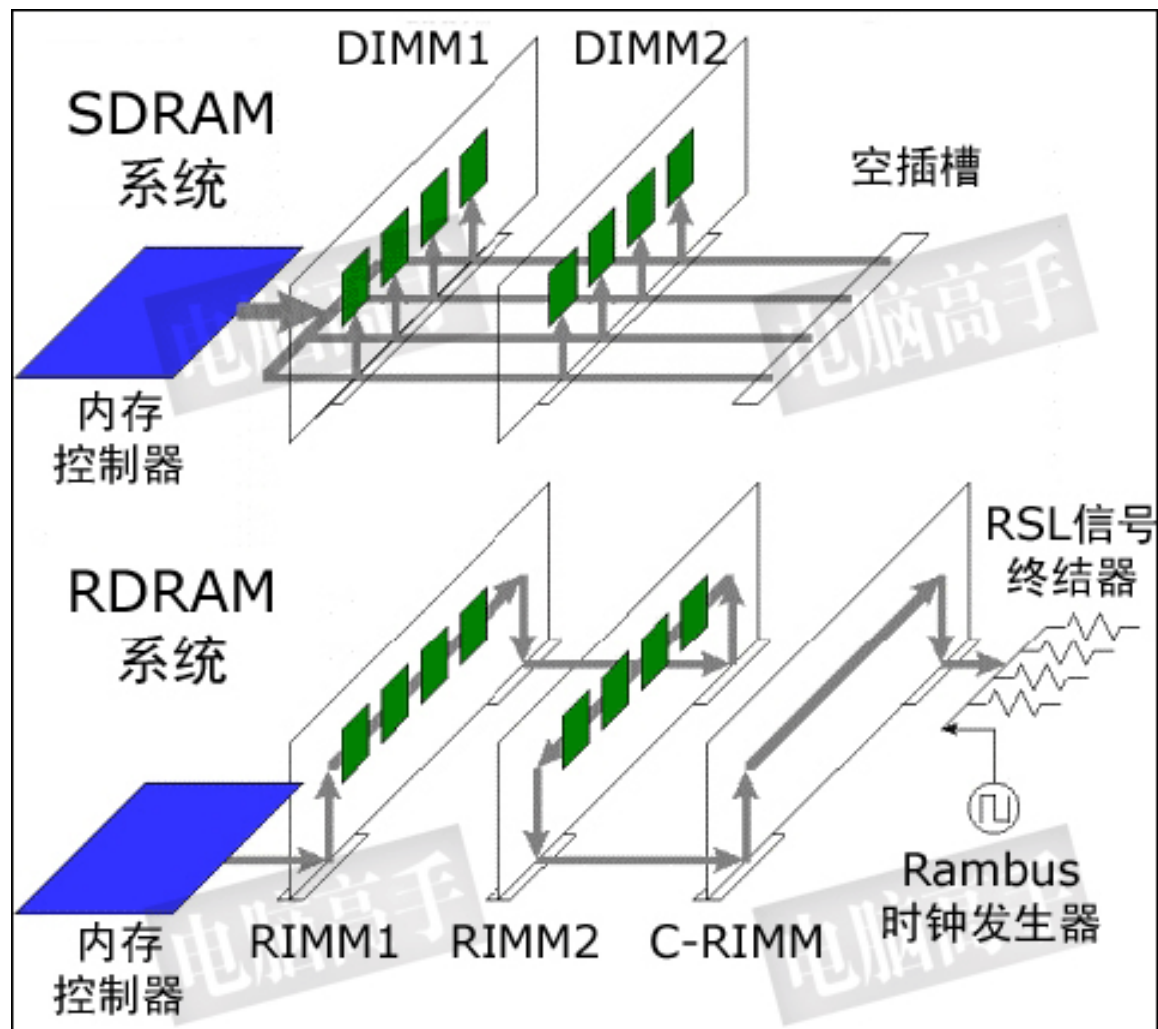
### 4.1 RDRAM 简介

RDRAM 与 DDR SDRAM 一样，也是一种采用双沿触发技术的内存，但它在结构、控制体系方面相对于传统 SDRAM 有着不小的变化，首先我们来看看它与 SDRAM 之间的简单比较。

内存 指标	DDR SDRAM（非 ECC）				RDRAM（单通道）			
模组传输位宽	非 ECC	64bit			非 ECC	16/32/64bit		
	ECC	72bit			ECC	18/36/72bit		
芯片位宽	4、8、16、32bit				16bit/18bit（ECC）			
芯片封装	TSOP-II（66pin）、CSP（60pin）				CSP（54、74、62、92pin）			
刷新周期	64ms				32ms			
内存模组	184pin（22pin 接地）				184pin（72pin 接地）			
工作电压	2.5V				2.5V			
时钟频率	100MHz	133MHz	166MHz	200MHz	300MHz	355.5MHz	400MHz	533MHz
数据传输频率	200MHz	266MHz	333MHz	400MHz	600MHz	711MHz	800MHz	1066MHz
传输带宽	1.6GB/s	2.1GB/s	2.7GB/s	3.2GB/s	1.2GB/s	1.4GB/s	1.6GB/s	2.1GB/s
模组标准	PC1600	PC2100	PC2700	PC3200	RIMM600	RIMM700	RIMM800	RIMM2100
芯片标准	DDR-200	DDR-266	DDR-333	DDR-400	PC600	PC700	PC800	PC1066

RDRAM 与 DDR SDRAM 比较





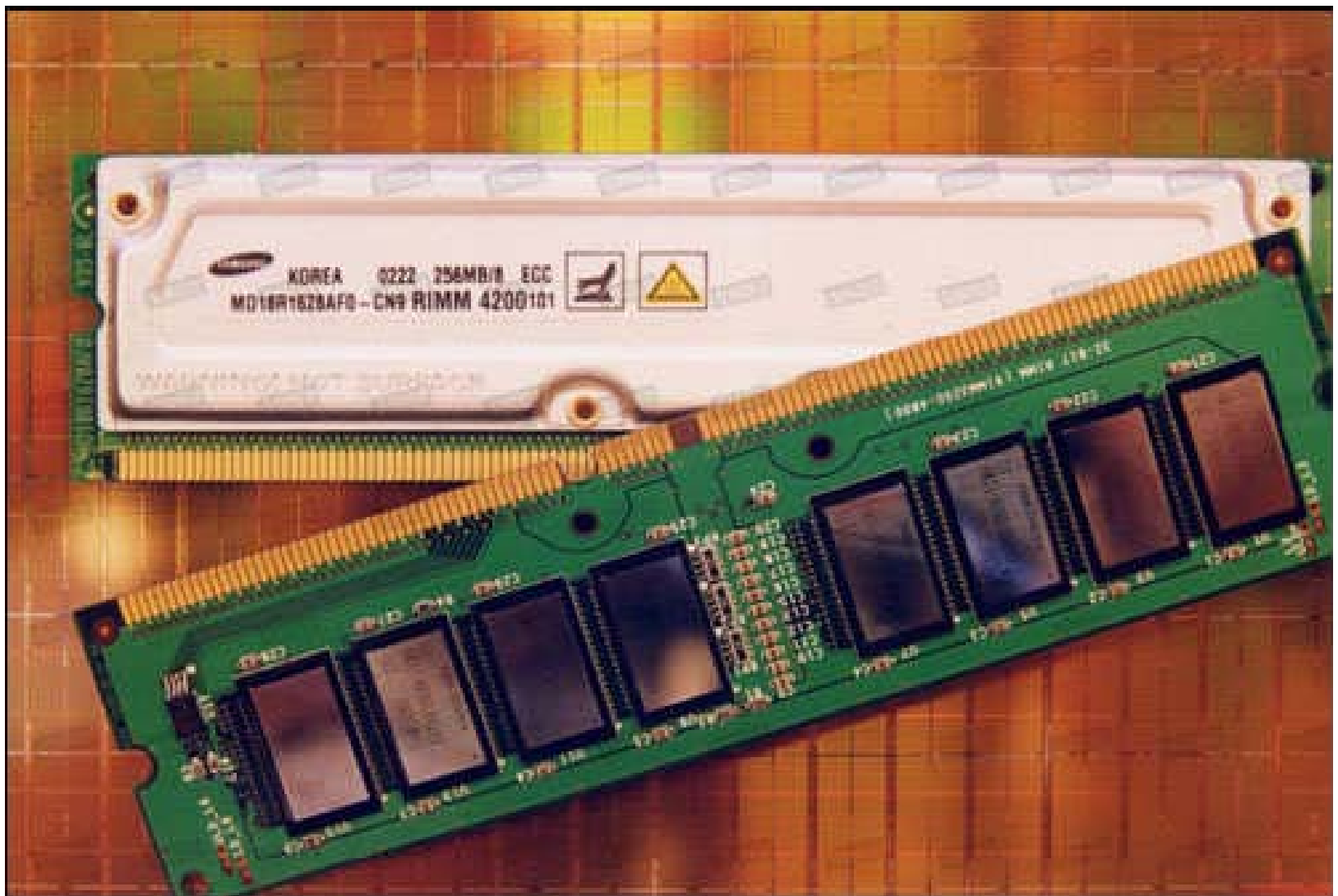
RDRAM 与传统 SDRAM 的架构比较

从架构比较图中，可以看出 RDRAM 在工作方式上与 SDRAM 有了很大不同。SDRAM 需要多颗芯片并联组成 P-Bank 与北桥沟通，而在 RDRAM 架构中每个芯片就是一个单独工作的读写单元，芯片的位宽就是与北桥接口的位宽，所以如果想用 ECC，就要用专门的 ECC 型芯片，也因此有了 16/18bit 的两种规格。而芯片的位宽就是一个 RDRAM 通道的位宽（本文以 16bit 芯片为例进行介绍）。

为了达到更高的容量，在一个通道中将多颗 RDRAM 芯片串起来，形成 RIMM（RambusInterface Memory Module，Rambus 接口内存模组），如果主板允许，完全可以设计一个超长的内存插槽与模组，但现实中肯定不能这么做，所以在主板上 Rambus 又把模组串起来组成通道。

由于是串联的形式，所以要求起始端与终结端形成一个完整的通路，而 RIMM 就是这个通路的串联器，因此 Rambus 要求所有的插槽必须插满，如果没有 RIMM 则用 C-RIMM（Continuity RIMM，RIMM 续连器）代替，以达到联通 RSL 信号并行终结器的目的。

工作时，RDRAM 每次寻址一颗芯片，所需要的数据则由通道数据总线传送到北桥，而不像 SDRAM 那样由所在模组直接通过 DIMM 接口传向北桥，也因此 RIMM 的引脚定义几乎是左右对称的。由于位宽的降低，为保证高带宽，RDRAM 使用了更高的时钟频率（这就意味着它不可能与系统时钟同步，所以只能叫 RDRAM 而不是 RSDRAM），芯片的工作频率明显高于 SDRAM/DDR，这样芯片的工作热量也急剧上升，为此 Rambus 在官方规范中规定 RIMM 必须配备散热片，从而成了现在这个样子。



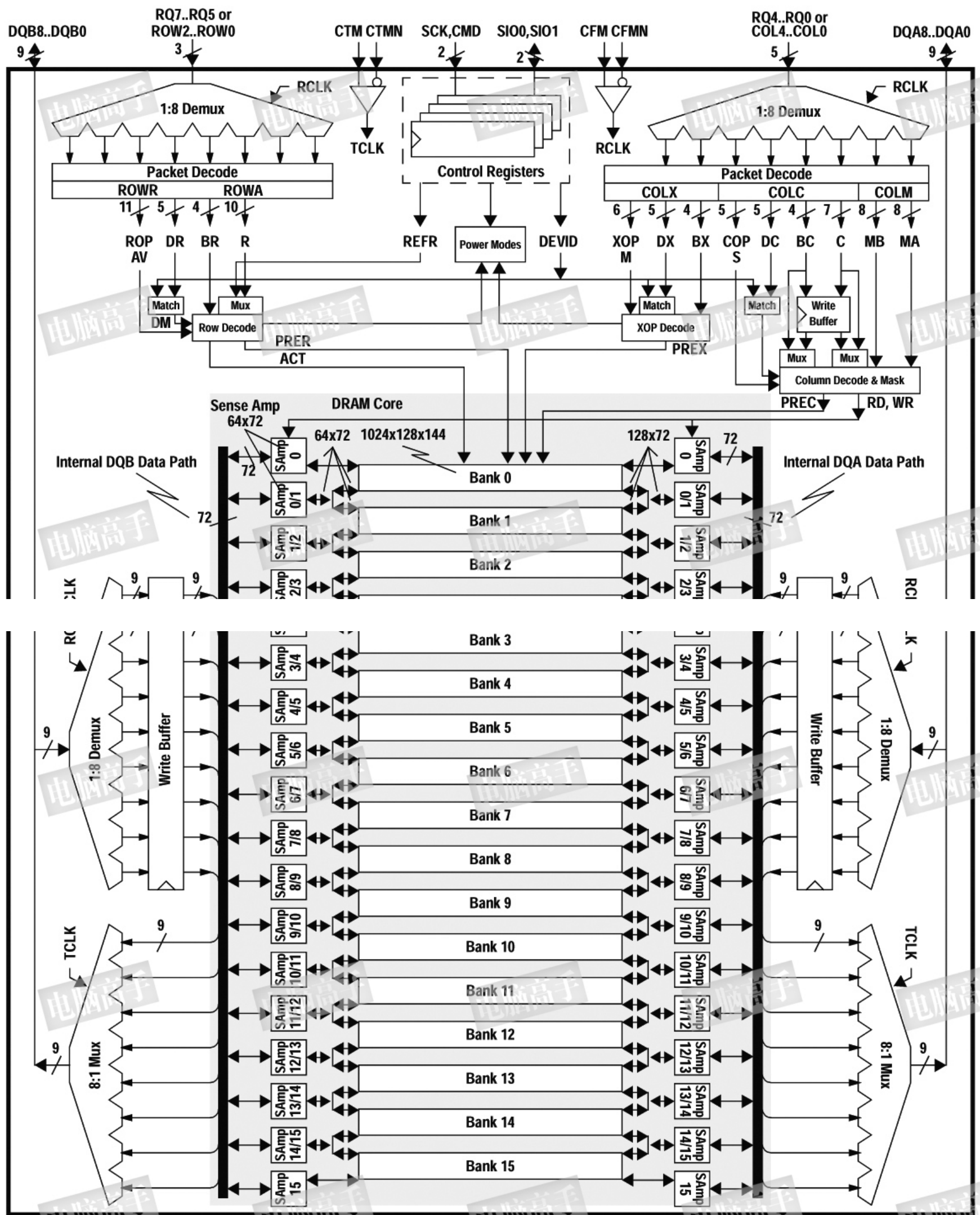
32bit 位宽 PC1066 芯片标准的 RIMM，它是目前 PC 领域中性能最高的 RDRAM 产品

## 4.2 RDRAM 的结构简介

### ● 4.2.1 RDRAM 的 L-Bank 结构

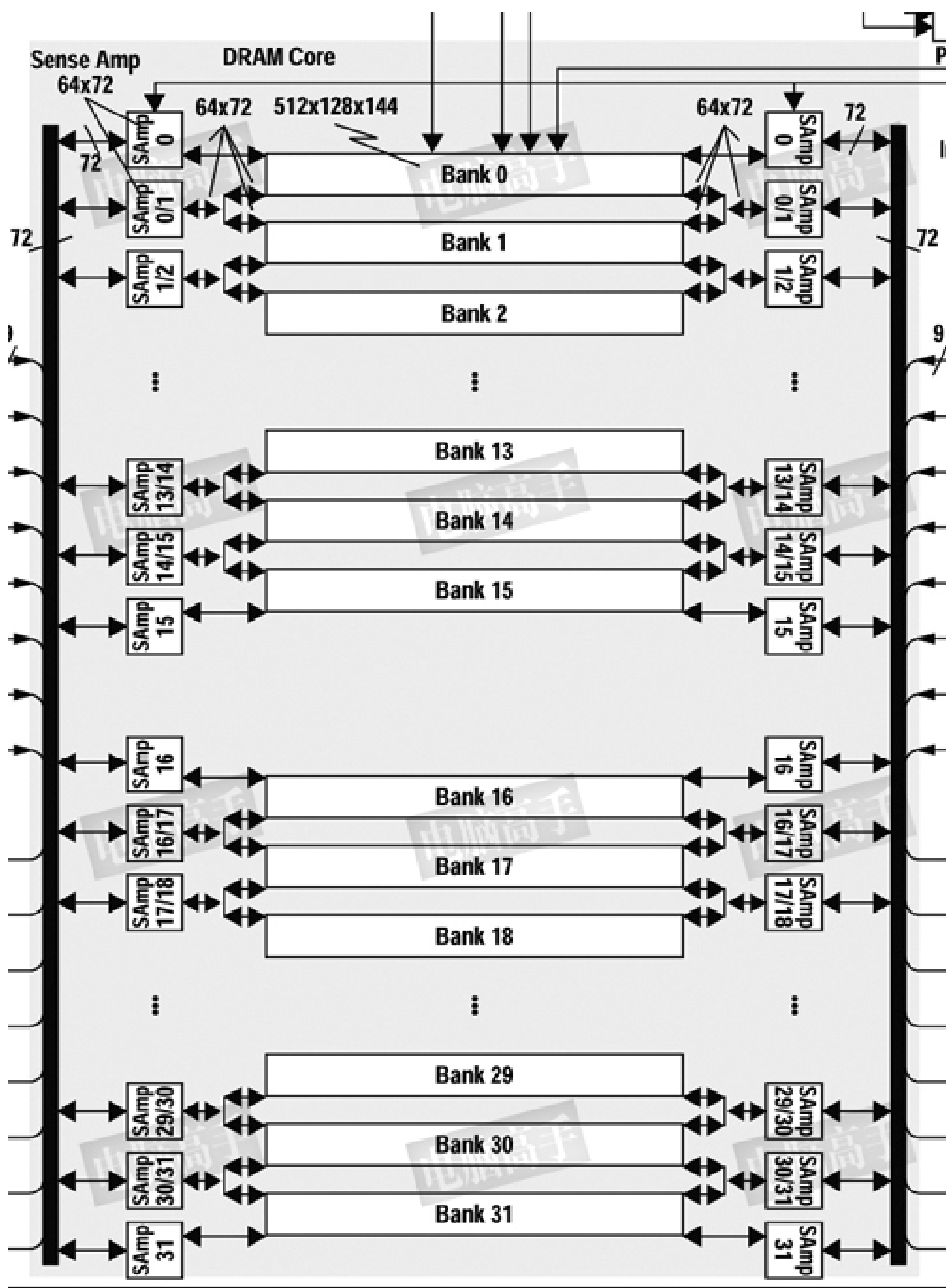
RDRAM 的内部仍主要由 L-Bank 构成，但它的设计与 SDRAM 家族有很大的不同。首先，每个 L-Bank 有两个数据通道 A 和 B，各为 8bit 位宽（ECC 型号为 9bit，这种设计就是 DirectDRAM 较以前 RDRAM 的不同），每个端口都配有 S-AMP。根据 L-Bank 数量与 S-AMP 的分配方式不同，目前 RDRAM 共有三种内核结构，分别是 32s、16d 与 4i。

较早时，RDRAM 的设计是 16d，所谓的 d 是指 Double（双），即除了 0 与 15 号 L-Bank，其余相邻的 L-Bank 每个数据通道（A 和 B）共用一个 S-AMP。



16d L-Bank 结构

后来分别向高端和低端领域发展了 32s 与 4i 技术。前者的 s 代表 Split，它将原来的 16d 内核分割为两个部分，各为 16d 结构，0、15、16、31 号 L-Bank 的每个数据通道各自独占一个 S-AMP。4i 则与传统的 SDRAM 相似，i 代表 Independent（独立），只有 4 个 L-Bank，各 L-Bank 的每个数据通道有单独的 S-AMP。



32s L-Bank 结构





上文已经讲过，L-Bank 数越多，造成 L-Bank 寻址冲突的机率就越小，但理论上 L-Bank 越多，所用的 S-AMP 也就越多，RDRAM 内存核心加工与面积控制的难度就越大，因此 32s 与 16d 都采用了共享 S-AMP 的设计。但即使这样，RDRAM 的生产成本仍被限制在较高的水平上，在早期这成为了 RDRAM 难以普及的重要原因。而 4i 就是为解决这一问题而出现的方案，成本更低，但性能也较前两者降低了。

另外，由于共享 S-AMP 的设计，除了个别独有 S-AMP 的 L-Bank，其他的 L-Bank 每次预充电操作也都是成双成对的。为此，在逻辑控制上，RDRAM 的操作要尽量避免相邻 L-Bank 前后进行，否则也会降低 RDRAM 的实际效率。

## ● 4.2.2 RDRAM 的主要特点

目前 RDRAM 主要有两个容量规格——128Mbit 和 256Mbit。L-Bank 中存储单元的容量也并不等于 RDRAM 的接口位宽，而是它的 8 倍，因此可以说 RDRAM 是一种 8bit 预取设计，这是它最主要的特点。对于 16bit 芯片，其存储单元的容量为 128bit，这些数据分别从通道 A 和 B 传输至 L-Bank，也就是说 L-Bank 两端的 S-AMP 一次各负责 72bit 数据的传输。由于预取为 8bit，所以 RDRAM 的突发长度也固定为 8，因为如果再高，对于 PC 应用将不太适合。

不过需要特别注意的是，一个字节的的数据不是由数据通道中的 8 条数据线进行并排传输，而是一个字节由一条数据线进行 8 次传输，这一点也与 SDRAM 不同，它意味着北桥在进行数据读/写时，必须要等 8 个周期之后才能完成，中途不能停止。也就是说，读取时目前的北桥（如 850）一次接收 128bit（16 字节）的数据，然后再转换为两个 64bit 数据分两次向 CPU 传送。

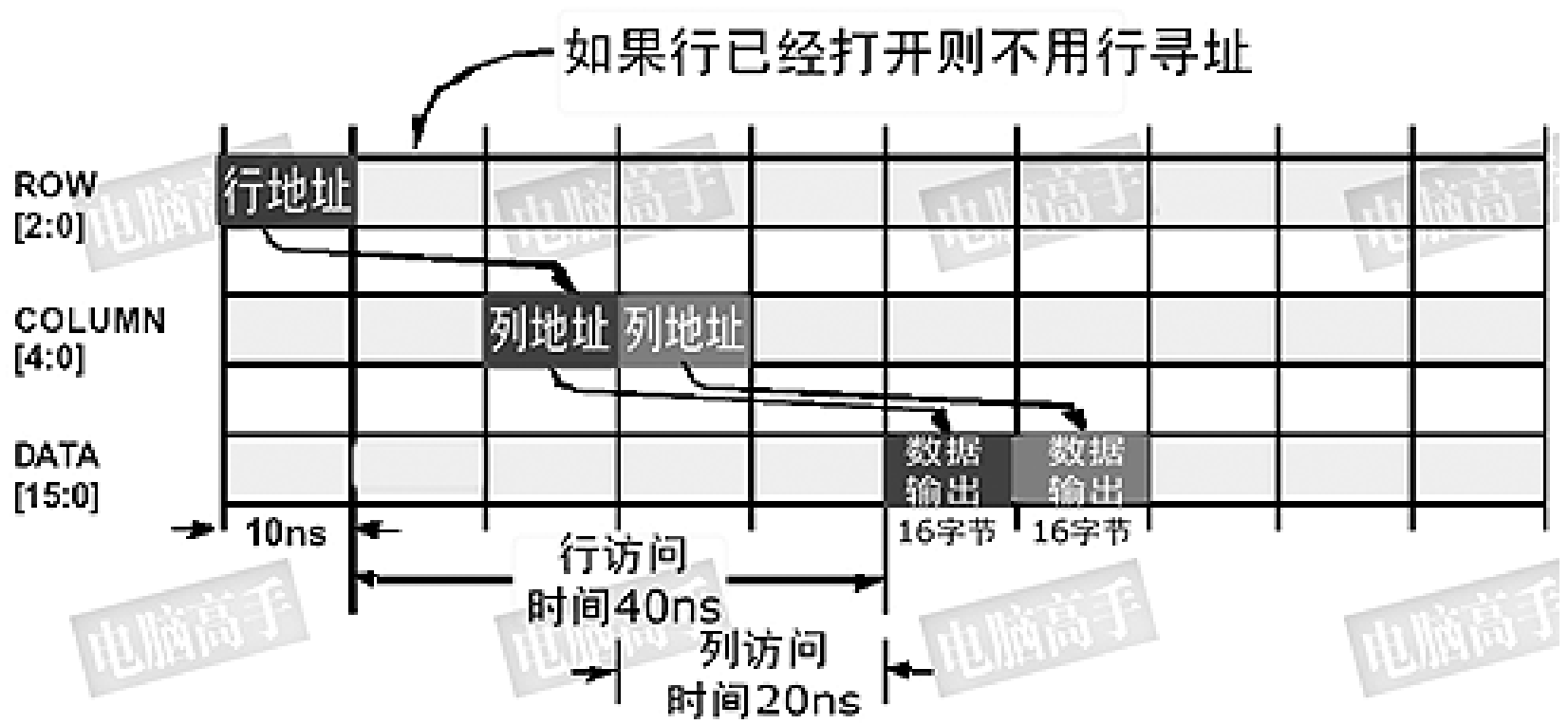
由于 RDRAM 的存储单元容量很大，所以 RDRAM 的行列地址线也大为减少，以 256Mbit 的 4i 结构的 RDRAM 为例，行地址为 12bit（4096），列地址为 7bit（128）。如果是 32s 结构的，由于 L-Bank 地址的增多，行列地址要更少（分别是 9 和 7bit）。而且 RDRAM 的行列地址线是独立的，但是 RDRAM 的行与列地址线各自只有 3 条和 5 条，显然不够用，Rambus 又是怎么搞定的呢？这就涉及到 RDRAM 具体的操作设计了。

## 4.3 RDRAM 的具体操作与相关技术

### ● 4.3.1 初始化与命令包

RDRAM 也有一个控制寄存器，在开机初始化过程中用来对 RDRAM 芯片进行配置，有的信息由北桥动态写入（如芯片地址、自刷新模式等），有的则是出厂时就设置好不能更改的（如刷新计数、生产商信息、支持的协议版本等）。在初始化之后，RDRAM 才能进入正常的工作状态。

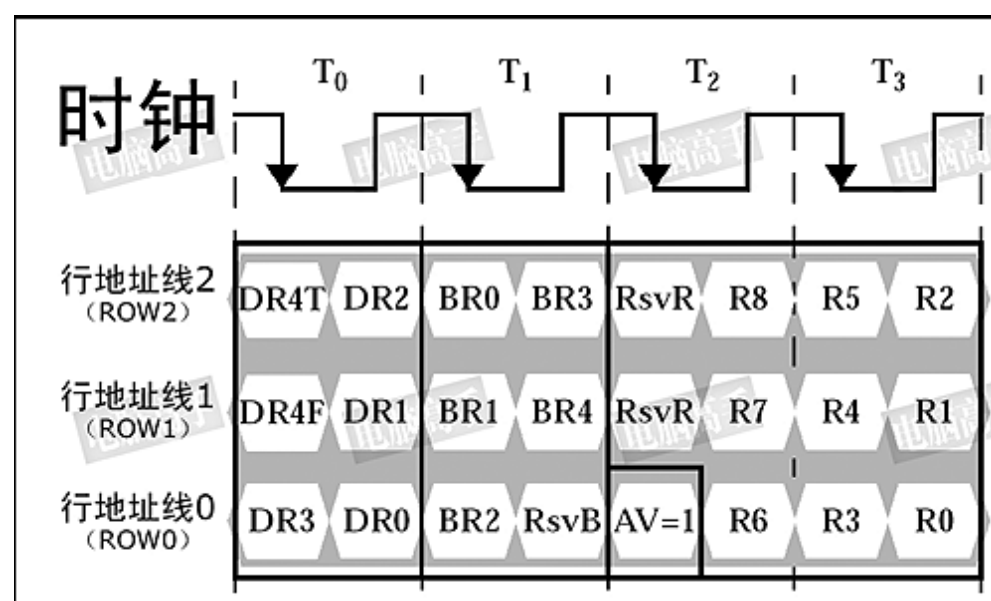
RDRAM 的读写操作过程与 SDRAM 基本是一样的，也要进行片选、L-Bank 定址、行/列寻址等操作（此时的行就是指 RDRAM 内存系统中的页），但由于它的每次操作只针对一颗芯片，所以具体操作起来有很大不同，这主要体现在“命令包”的方式上。



RDRAM 读取时序图，以 PC800 为例，400MHz 时钟频率

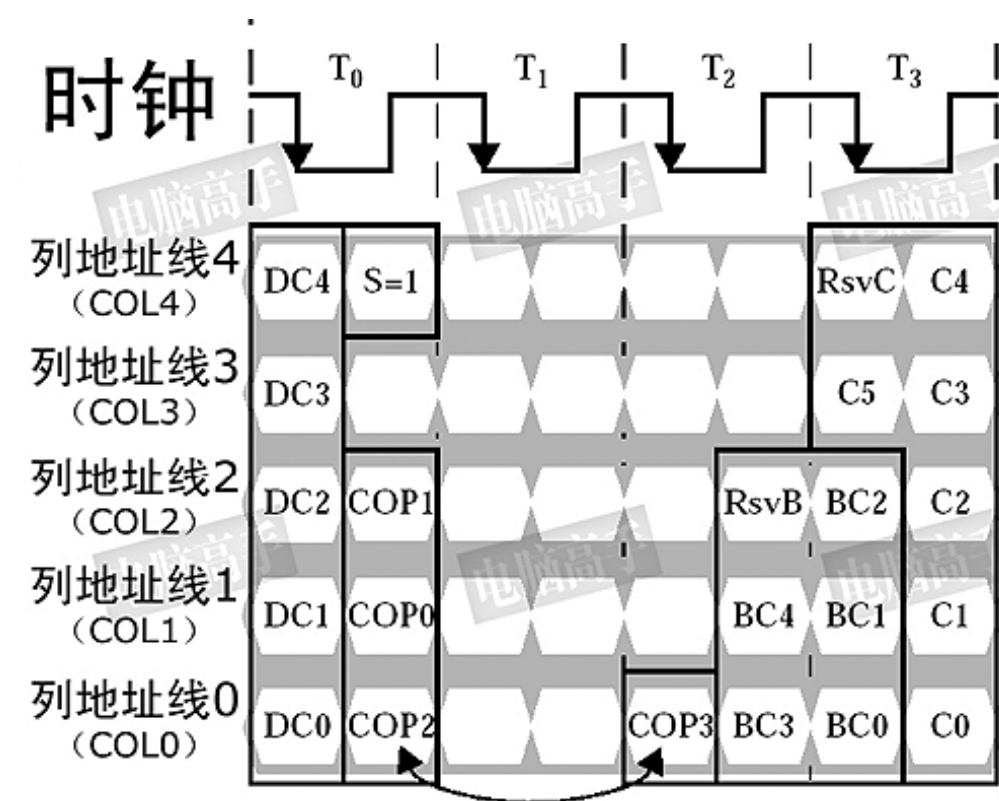
在上图中，我们可以看到行寻址命令与列寻址（读）命令并没有同时发出，而且各自占用了 10ns 的时间。我们算一算，对于 PC800，10ns 相当于 8 个传输周期。难道是传输有延迟？

从行列地址的设计，就能猜到这是一个命令包形式的操作。所谓的命令包，就是将一组命令集合在一起，统一发出。在 RDRAM 中，行命令包与列命令包都分为两种，一种是正常的读/写操作命令，一种是芯片操作命令（如数据掩码、预充电、刷新、电源管理等）。现在我们就看看行与列读/写命令包都包含哪些信息。



行读/写命令包的信息组成

DR4T、DR4F	命令包的广播类型（是对所有芯片还是单一芯片），如果是单一芯片，可用来当作最高芯片地址
DR3-DR0	芯片地址
BR4-BR0	L-Bank 地址
R0-R8	行地址
AV	区分是操作命令还是读取命令，如果是操作命令，R0-R8 将成为操作控制代码
Rsv (X)	可忽略或保留用 bit



列读/写命令包的信息组成

DC4-DC0	芯片地址
COP0-COP3	指定具体命令（是读还是写）
BC4-DC0	L-Bank 地址
C5-C0	列地址
S	允许操作控制位（1 为允许读写）
Rsv (X)	可忽略或保留用 bit

至于操作命令包就不在此多说了，因为构成的形式基本就是这样，每次用 8 个传输周期进行命令发送。而且由于 RDRAM 的命令代码很多，也比较复杂，在本专题中也不用一一列出，关键在于让大家明白 RDRAM 的寻址是怎么一回事即可，剩下的具体代码定义，如果有兴趣大家可以自行研究。

<b>提示：RDRAM 的 32 颗芯片寻址限制</b>
从行列命令包中，大家能发现它的芯片地址只有 5bit，最多只能寻址 32 颗芯片。这就是那个“臭名昭著”的每个 RDRAM 通道只能容纳 32 颗芯片限制的根本原因。也就是说不管通道中用了几条 RIMM，芯片的总数不能超过 32 颗，如果有 3 根 RIMM 插槽，但使用了两条 16 颗芯片的 RIMM，那么第三根插槽就只能插 C-RIMM。以目前的设计，RDRAM 芯片最大容量为 64MB (512Mbit-4i)，这样一个通道最多只有 2GB 的容量。

### ● 4. 3. 2 操作时序计算

通过上面的时序图，我们可以发现 RDRAM 计算时序的方法与 SDRAM 家族不一样，这在比较两者间时序效率时有着关键的影响。

Rambus 的时序规定与 FPE/EDO 内存时一样，在读取时延用了 tRAC、tCAC 的定义，前者是行访问周期 (RAC, RAS Access Cycle/Delay)，后者是列访问周期 (CAC, CAS Access Cycle/Delay)，你可以把它等同于 SDRAM 中的 CL，但决不能在 RDRAM 中引入 CL 这个概念。

在写入时则将 tCAC 替换为 tCWD (CAS to Write Delay)。它们的单位都是时钟周期，对于 PC800，一个时钟周期就是 2.5ns，对于 PC1066 就是 1.876ns 了。显然，时钟频率越高，延迟周期就越短。

但是这些时序是从命令包发送完毕开始计算，SDRAM 则是在命令发送同时开始计算。因此，在计算 RDRAM 的操作延迟时，命令包本身占用的时间也必须要考虑进来。

### ● 4. 3. 3 写入延迟与掩码操作

RDRAM 为写入设置了专用的延迟 tCWD，这并不是被迫而是有意设计的。RDRAM 不需要 DQS 之类的

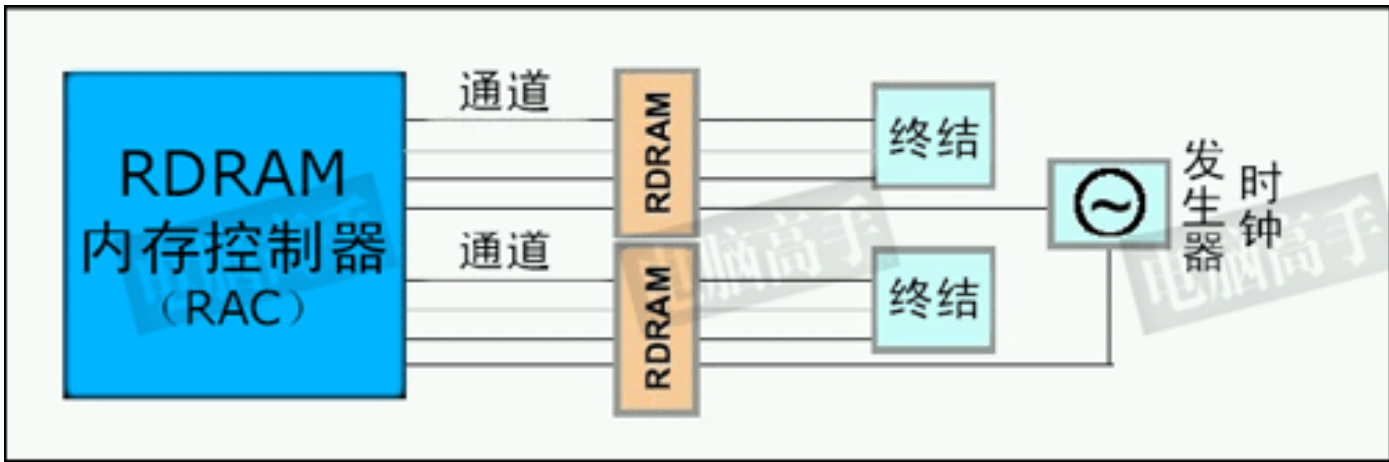
信号进行同步操作，数据是可以立即接受的，但出于总线利用率的考虑，RDRAM 加入了写入延迟，它略短于 tCAC。在具体操作中，芯片上没有引脚控制写入允许/禁止，一切的命令在命令包中进行定义，所以读命令可以在写过程中发出，经过 tCAC 后有效。这样在写后读操作中，除了 tCAC 与 tCWD 之间的差距外（估计是留给写回的时间），几乎没有任何停顿，而不像 SDR/DDR SDRAM 中有较大延迟。

在写入过程中，数据都是先存在写入缓冲区中，这个操作的目的在于等待掩码的控制。RDRAM 的数据掩码只对写入有效，当收到掩码命令后，RDRAM 将指定的引脚数据从缓冲区中删除，之后再进行真正的写入。

● 4.3.4 多通道技术与多通道模组

PC800 的速度在当时可算是 RDRAM 的一极限，但它的 1.6GB/s 带宽并不能满足高端应用的需要，而且 DDR 一方主推的产品是 P2100 的 DDR-266，为此 RDRAM 利用双通道技术来弥补带宽上的不足。简单的说，它就像一个用于内存的 RAID，两个通道的数据在 RAC 一端进行分割（写）与合并（读），两个通道的 RIMM 缺一不可并要求结构完全一致，因为寻址信号是一样的，必须适用于两个 RIMM，这也就意味着两个 RIMM 的存储轨迹也是一样。

但是，数据的寻址延迟并没有变化，只是连续传输率提高了一倍达到 3.2GB/s（两个 PC800 通道），而且总的内存容量也增加了一倍。时至今日，虽然 RDRAM 使用窄位宽设计，但毕竟不是串行的方式，提升频率也越来越困难，最新的 PC1066 标准也只达到 2.1GB/s 的带宽，此时双通道设计几乎成为 RDRAM 的标配。可以说没有双通道技术的支持，RDRAM 是很难走到今天的。

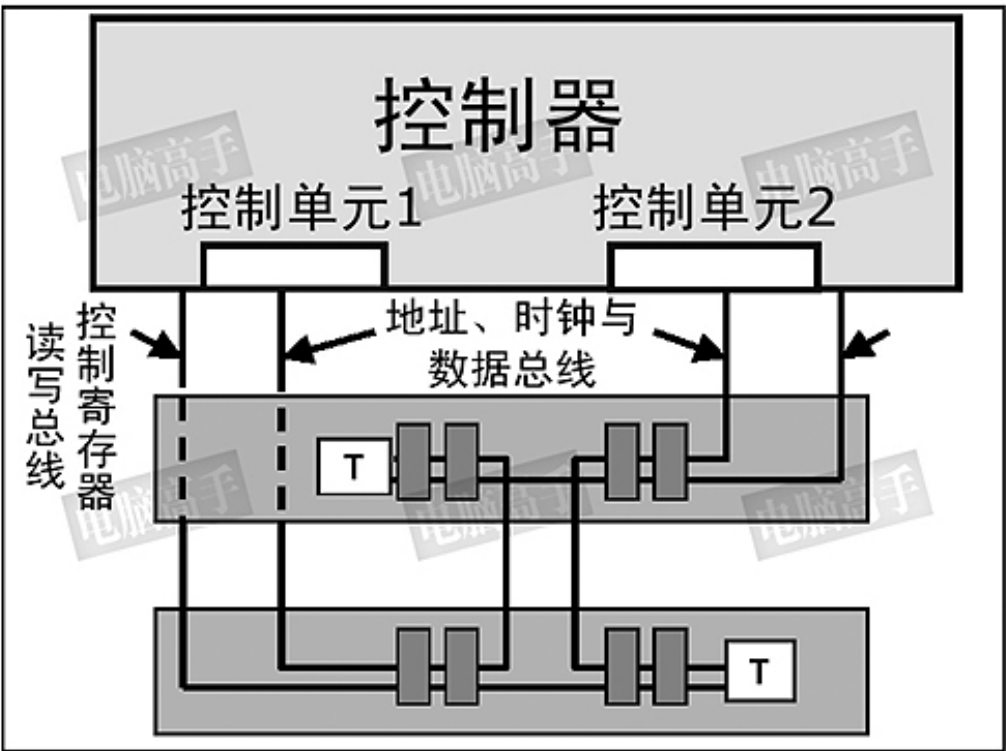


RDRAM 双通道结构

以前，双通道技术是以两条 RIMM 来实现，在双通道已经是 RDRAM 标准设计的今天，这种设计的弊病很明显，比如客户的购置成本、主板的布线设计等。为此，在一些内存厂商的支持下，RDRAM 出



现了多通道模组设计，其主体思路就是将每个通道的信号终结电路移植到模组上来，在一个模组上实现多通道传输。



32bit 的 RIMM 设计，每个通道的终结器做在了模组上

目前 PC 市场上 32bit RIMM 逐渐开始流行并终将取代传统的双通道设计，对于 64bit RIMM，由于是 4 通道设计，得需要 4 通道北桥芯片的支持，所以目前不可能在台式机领域里普及。

模组类型	16 bit RIMMs		32 bit RIMMs		64 bit RIMMs	
模组名称	RIMM1600	RIMM2100	RIMM3200	RIMM4200	RIMM6400	RIMM8500
RDRAM芯片数据传输频率	800 MHz	1066 MHz	800 MHz	1066 MHz	800 MHz	1066 MHz
模组数据位宽	16 or 18 bits	16 or 18 bits	32 or 36 bits	32 or 36 bits	64 or 72 bits	64 or 72 bits
模组带宽 (MB/s)	1600	2133	3200	4266	6400	8532
模组上最少芯片数量	1	1	2	2	4	4
模组上最多芯片数量	16	16	16	16	16	16
地址总线数量	1	1	2	2	1	1
共享地址总线	No	No	No	No	Yes	Yes
局部数据终结	No	No	Yes	Yes	Yes	Yes
字节屏蔽支持	Yes	Yes	Yes	Yes	No	No
ECC模组位宽	18	18	36	36	72	72
模组引脚数量	168	168	232	232	326	326
数据总线阻抗	28 Ohms	28 Ohms	40 Ohms	40 Ohms	40 Ohms	40 Ohms
模组工作电压	2.5 Volts	2.5 Volts	2.5 Volts	2.5 Volts	1.8 Volts	1.8 Volts
模组终结电压	none	none	1.8 Volts	1.8 Volts	1.5 -1.8 Volts	1.5 -1.8 Volts
内同步写数据	No	No	No	No	Yes	Yes

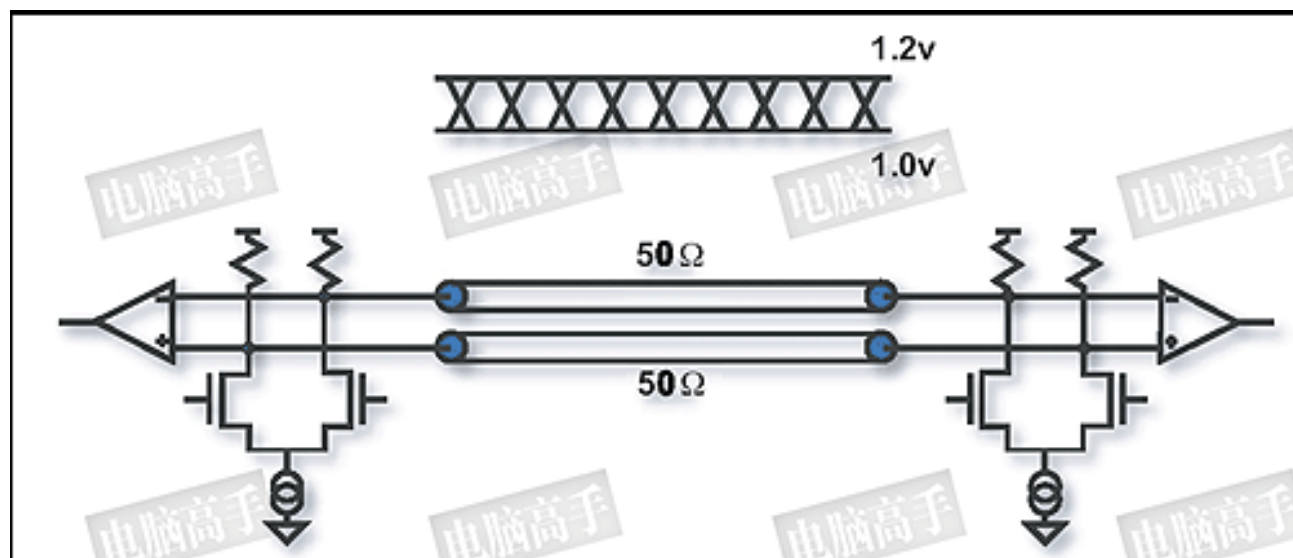
不同规格的 RIMM 间比较

误区：32bit RIMM 需要特殊的双通道芯片组支持

在 Intel 850E 主板上市时，很多人都发现有的主板使用了 32bit RIMM 设计，并由此断定 Intel 850E 新增了重要的功能，即支持 32bit RIMM。这是一个很常见的错误认识。32bit RIMM 相对于传统的双通道主板，只涉及物理联接形式的改变，而不涉及逻辑控制方面的变动，所以只要是支持双通道的芯片组，都可以由主板厂商通过专门的布线设计来支持 32bit RIMM，因此不要过分迷信 32bit RIMM 的技术含量，850 芯片组在理论上完全可以设计出支持 32bit RIMM 的主板，就看主板厂商愿不愿意了。

### ● 4.3.5 黄石技术

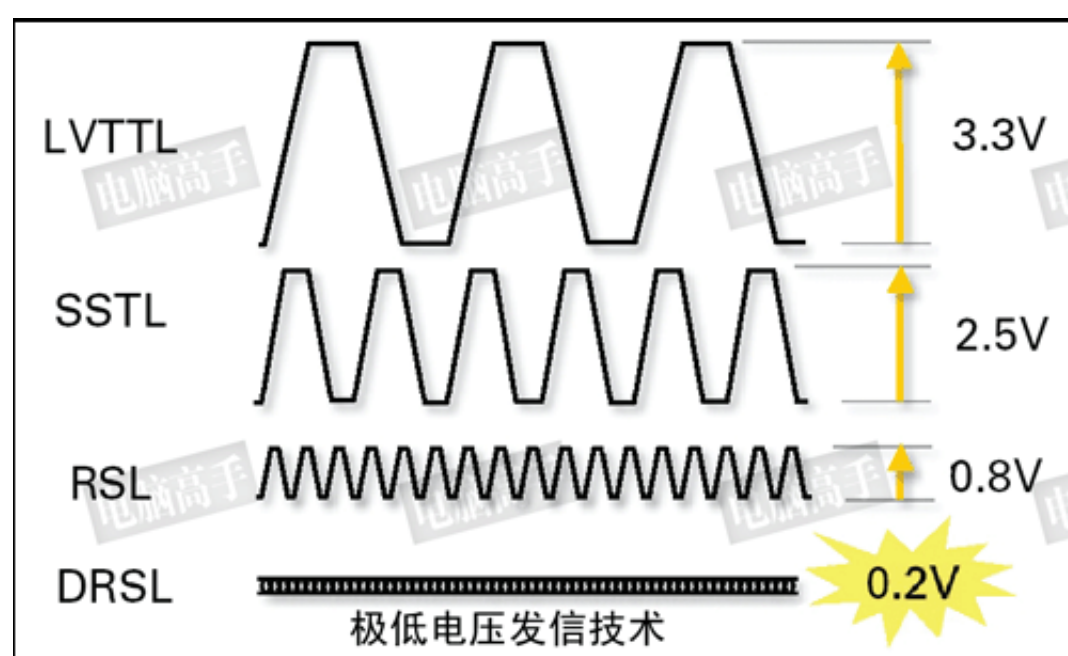
黄石（Yellowstone）是 Rambus 为了适应未来带宽的需要而开发的信号与数据传输技术，其主要的技术特点有四个：



黄石技术的物理系统结构

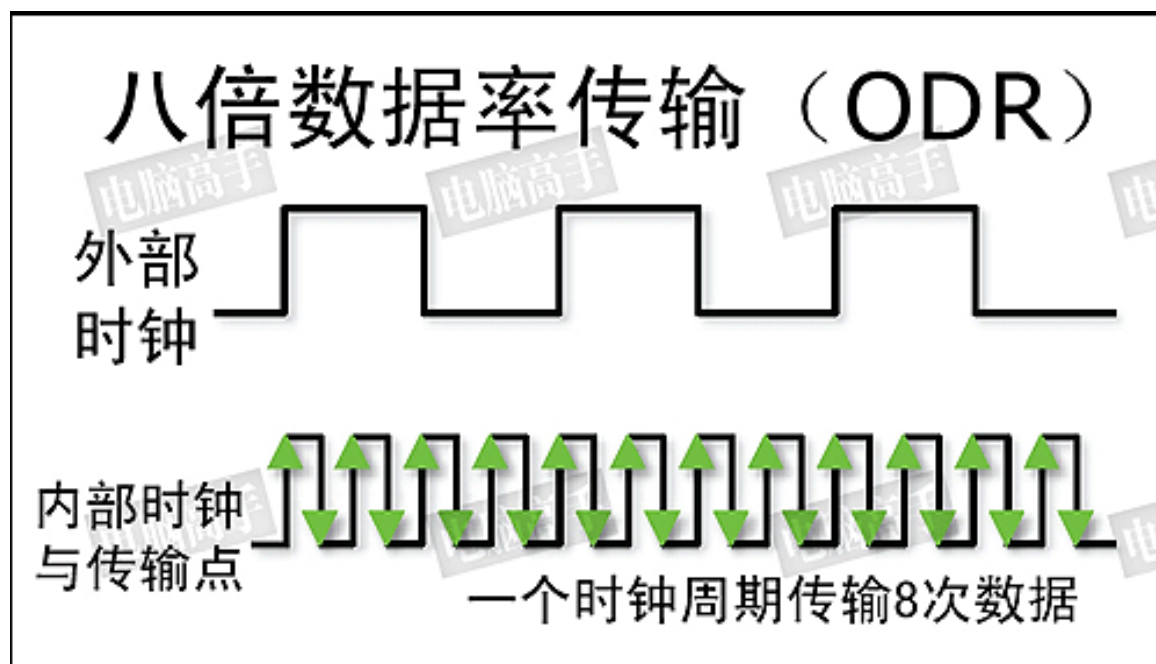
A、3.2GHz 传输频率，未来可高达 6.4GHz，按 16bit 位宽计算，带宽可达 6.4GB/s，双通道应用则为 12.8GB/s。

B、极低电压的差分 RSL 信号（DRSL），降低电源消耗并保证信号质量与制造成本。信号电压差值只有 200mV，是目前电压差最小的内存信号技术。



DRSL 发信技术与其他内存接口发信技术的比较

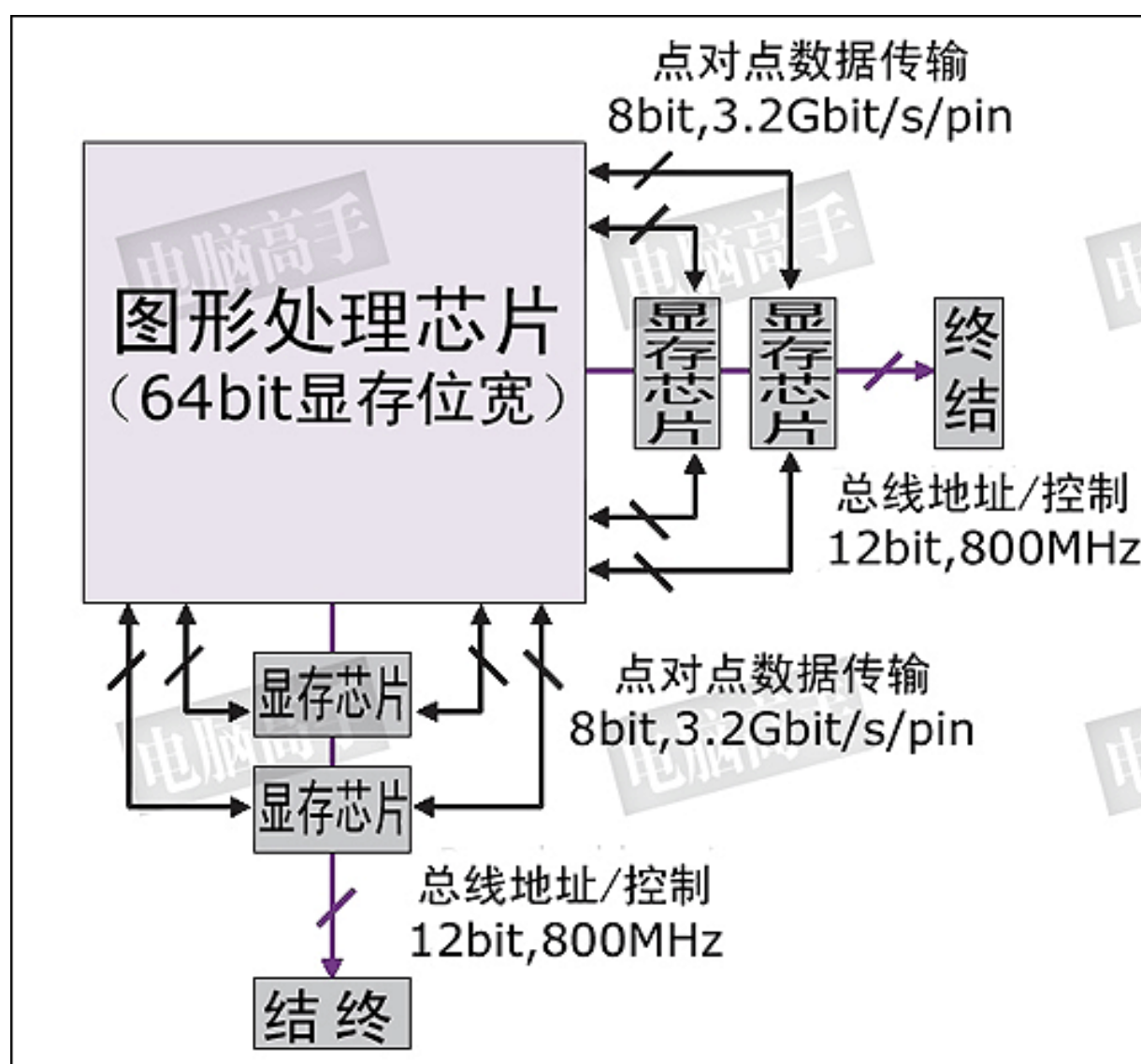
C、八倍数据流技术（ODR，Octal Data Rate）。目前采用黄石技术的 RDRAM，时钟频率仍是 400MHz，但芯片内部通过专用的锁相回路（PLL）将其转换为 1.6GHz 的内部时钟，然后在此基础上使用 DDR 技术，从而能在一个时钟周期内传输 8 次数据。数据传输频率也因此达到了 3.2GHz。



ODR 操作示意图

D、固定相位技术 (FlexPhase)，使内存生产者不再费力的去调校 PCB 的设计以减少延迟/潜伏期对数据/时钟间同步的影响。固定相位技术使信号本身就具备了数据/时钟同步与自校准能力，从而使外围有关时序跟踪的设计与布线变得非常简单，并有助于提高同步性，提高总线利用率。

黄石技术于 2001 年 10 月 2 日正式发布，但是虽然它有这样那样的优点，但从最近的资料中显示，Rambus 主要将其定位于消费电子、网络、通信和图形设备市场。对于目前的桌面 PC 市场，黄石在近期应用的可能性并不大。



Rambus 展示的用于显卡的点对点黄石 RDRAM 显存方案



4.4 延迟与总线利用率的比较

仍以 PC800 为例，由于 tRAC 已经包括了从行选通至数据输出的所有延迟，与是否双通道无关，所以我们只需将它再加上命令包的占用时间即可算出 RDRAM 一次访问所需要的时间。

PC800 的 tRAC 基本都是 40ns（16 个时钟周期），加上命令包占用的 4 个时钟周期 10ns，总共耗时为 50ns。而这是在行关闭的情况下，没有计算预充电的时间 tRP，它一般为 12 个时钟周期（加上命令包时间），即 30ns，共计 80ns。显然，时钟频率越高，延迟就会越短。

下面就来比较一下读取操作时 RDRAM 与 DDR SDRAM 的延迟。

内存类型 时序比较	RDRAM		DDR SDRAM		
	PC800	PC1066	266B (CL=2.5)	333 (CL=2.5)	400 (CL=3)
时钟周期	2.5ns	1.876ns	7.5ns	6ns	5ns
需要预充电时的延迟	80ns	54.512ns	63.75ns	51ns	45ns
页面已关闭时的延迟	50ns	39.504ns	41.25ns	36ns	30ns
页面已打开时的延迟	30ns	22.512ns	18.75ns	15ns	15ns

注：DDR SDRAM 的 tRP 与 tRCD 都按 3 计算，RDRAM 方面，tCAC=8、tRP=8，PC800 的 tRAC 按 40ns 计算，PC1066 按 32ns 计算

读取操作时 RDRAM 与 DDR SDRAM 的延迟比较表

从对比表中可以看出，RDRAM 相对于 DDR SDRAM 在首次寻址时的确存在较大的延迟，即使是最新的 PC1066，在与 DDR-333 的比较中也不占优势。不过，借助于双通道的设计，RDRAM 在高数据量传输应用中的优势还是比较明显的。另外，在总线的利用率方面 RDRAM 的设计也居领先地位，这为保证它的总体效率提供了坚实的保障。

内存类型	PC100	PC133	DDR266	RDRAM PC800
时钟频率 (MHz)	100	133	133	400
数据传输率 (MHz)	100	133	266	800
系统数据总线位宽	64-bit	64-bit	64-bit	16-bit
峰值带宽 (MB/s)	800	1067	2133	1600
总线利用率	62%	59%	42%	74%
最大实际带宽 (MB/s)	494	631	897	1190

注：SDRAM 与 DDR SDRAM 的 L-Bank 数按 4 个计算，RDRAM 按 16 个计算

各内存的总线效率比较

这个对比表是东芝公司经过反复实验而得出的结论，它是通过一些典型的操作（如写-读-读），结合不同页命中情况下的时序，以及刷新对内存操作的影响等分析而得出的。

由于 DDR SDRAM 在 L-Bank 数量上占劣势，所以出现 L-Bank 寻址冲突的可能性要大为提高，而且在写后读操作中，RDRAM 的延迟也明显小于 SDRAM 家族，因此虽然 PC800 的峰值带宽不如 DDR-266，但综合效率要更好。这可以解释为什么在一些测试中，RDRAM 明显比 DDR 领先的原因。不过，在以零散数据为主的操作中，RDRAM 的固定传输周期以及高延迟就成为了性能的障碍。

从前面的分析可以看出，SDRAM/DDR 在数据控制上的灵活性要比 RDRAM 高，首次访问的延迟也更短，因此在某些操作中，即使带宽比 RDRAM 系统小，性能仍不见得落后。比如 845D/E 在某些应用测试中，完全可以与双通道 PC1066 一较高低。而 Intel 决定在高端服务器领域使用 DDR 芯片组，也基本是出于这个考虑，因为在服务器的操作中，零散型存取操作所占比例很大。相反，若大规模连续存取操作占比例很大（如视频与音频工作站），那么可能就要考虑 RDRAM 了。

## 4.5 未来竞争展望

目前随着多通道技术在 DDR 上的普及，RDRAM 在带宽上的优势也变得不明显了。所以，RDRAM 如果不及时提高单通道的性能，很快会被强大的 DDR 家族赶出台式机领域。但 RDRAM 的时钟频率已经很高了，再向上提高已经很难，不少 RDRAM 厂商都表示，800MHz 时钟频率可能将是 RDRAM 的一个巨大门槛，即使能超过，成本可能也是惊人的，要知道目前 533/400MHz 的 RIMM 就已使用了 8 层 PCB，800MHz 时 PCB 成本将很难控制。这也是为什么 RDRAM 急于推出 32bit 与 64bit RIMM 的原因，毕竟内存这种高带宽应用设备，还是需要一定位宽的保证。而且高位宽的同步性也不像想象中的那么难以控制，DQS 的设计就很大程度地解决了这一问题，所以，DDR 可以借助较少的转产成本，较低的 PCB 成本（即使是 DDR-II 也是 6 层设计），成为 PC 内存的首选产品。

现在再去争论 RDRAM 与 DDR 谁胜谁败已经没有意义，RDRAM 已经很难再在主流市场重振雄风。这主要不是它的技术限制，而是早期的市场动作与成本的压力造成的。虽然现在 4i 芯片开始起步，但支持这种结构的芯片组还很难找到（至少 850E 不支持）。在 820 时代，RDRAM 由于成本而没有打开市场，现在可以通过降低成本来提高竞争力，但 DDR 一方也有了多通道技术。Rambus 也因此明智地将黄石定位于专用/定制市场。这样，在今后很长一段时期里我们只有看 DDR 的独角戏了。



# 第五章 内存模组介绍

内存模组是内存在 PC 系统中的最终体现形式，所以在本专题的最后，我们来简要谈谈内存模的类型和未来的发展情况。不过，本章节只介绍 DIMM，而不涉及 RIMM（其实两者的很多概念是相通的）。目前经常见到的模组主要有五种：

- 1、Unbuffered DIMM：无缓冲型模组，这是我们平时所用到的标准 DIMM，分有 ECC 和无 ECC 两种，简称 Unb-DIMM。
- 2、Regustered DIMM：寄存型模组，这是高端服务器所使用的 DIMM，分有 ECC 和无 ECC 两种，但市场上几乎都是 ECC 的，简称 Reg-DIMM。
- 3、SO-DIMM：Small Outline DIMM，小外型 DIMM，笔记本电脑中所使用的 DIMM，分 ECC 和无 ECC 两种，DDR-II 时代仅有无 ECC 的型号。
- 4、Micro-DIMM：微型 DIMM，供小型笔记本电脑或手持式设备使用的 DIMM。
- 5、Mini-DIMM：DDR-II 时代新出现的模组类型，它是 Regustered DIMM 的缩小版本，用于刀片式服务器等对体积要求苛刻的高端领域。

内存类型	SDRAM	DDR SDRAM	DDR-II SDRAM
Unbuffered DIMM	168pin（可 ECC）	184pin（可 ECC）	240pin（可 ECC）
Registered DIMM	168pin（可 ECC）	184pin（可 ECC）	240pin（可 ECC）
SO-DIMM	144pin（可 ECC）	200pin（可 ECC）	200pin（无 ECC）
Micro-DIMM	144pin（无 ECC）	172pin（无 ECC）	200pin（无 ECC）
Mini-DIMM	无	无	244pin（仅 ECC）
DIMM 的 PCB 层数	4/6	6	6

各类型内存 DIMM 对比表



三星公司 DDR-333 标准的 SO-DIMM，容量高达 512MB

## 5.1 Unb 与 Reg-DIMM 的区别

Unb 与 Reg-DIMM 的最大区别在于模组上是否有寄存器。在高容量模组上，内存芯片数量很多，而且在需要大容量内存的工作场合，内存模组的安插数量也是很多的，这使命令与寻址信号的稳定性受到了严峻考验。很多芯片组的资料中都说明只有使用 Reg-DIMM 才能达到标称的最高内存容量，从这点就能猜到寄存器的作用——稳定命令/地址信号，隔离外部干扰。

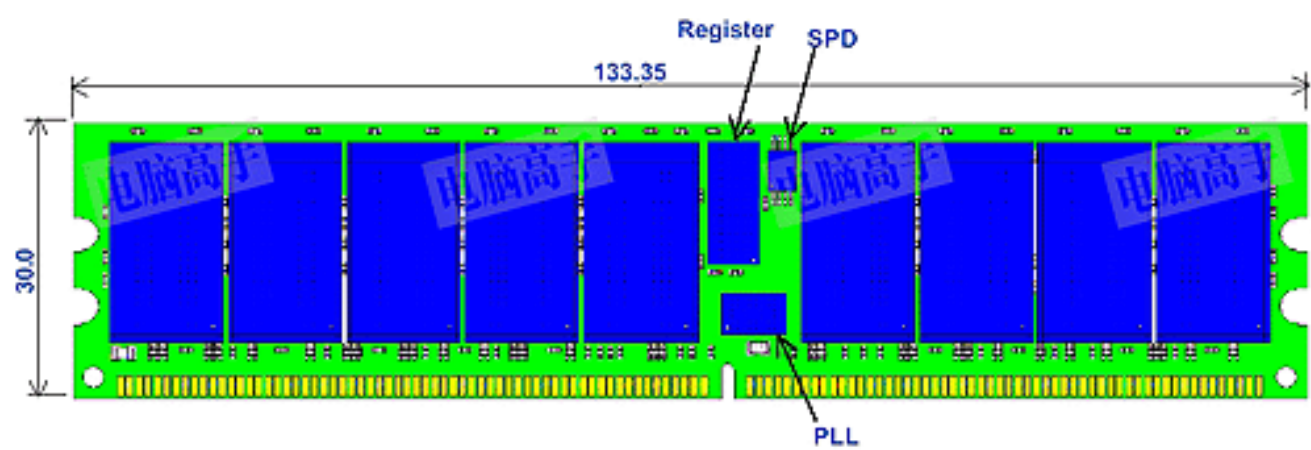


Reg-DIMM 工作示意图，命令与地址信号通过寄存器中继传输至内存芯片

在工作时，命令地址信号会先送入寄存器进行“净化”并进入锁存状态，然后再发送至内存芯片，芯片中的数据则不经过寄存器而直接传向北桥。由于要经过中继传输，所以内存操作的时序也会因此而增加一个时钟周期，这是它所带来的一个弊端，但在高端应用中，内存系统的稳定可靠的重要性远在性能之上，所以 Reg-DIMM 一般只用于高端市场，并且需要芯片组的支持才行（主要是 Reg 所引起的时序变化）。

而在高端设备中，ECC 基本都是必须的，因此市场上的 Reg-DIMM 也都无一例外的是 ECC 型模组，虽然也有无 ECC 的 Reg-DIMM 设计标准。

另外，为了保证内存工作时钟的稳定，Reg-DIMM 上还要有一颗 PLL 对时钟信号对主板发来的时钟信号进行跟踪/锁定。在 SDRAM 时代，这并不是必须的设计，但到了 DDR 时代，由于对时钟的敏感性，PLL 成为了必备元件。



DDR 内存模组的结构图，寄存器与 PLL 是它相对于 Unb-DIMM 的最大不同

现在再回头看看 Unb-DIMM，就很明白了。它关键就少了寄存器，但为什么不称之为

Unregistered-DIMM 呢？其实，Buffered 与 Registered 是 Reg-DIMM 的两种工作模式，前者在 Reg-DIMM 上并不常用，它是以时钟异步方式工作的，输出信号的再驱动不与时钟同步，Registered 模式下输入信号的再驱动则与时钟同步。显然，Buffered 模式下的性能要更低一些。不过，从原理上讲 Registered 模式也是一种缓冲操作，只是与时钟同步而已。在 SDRAM 的 Reg-DIMM 上，Buffered 与 Registered 模式通过 REGE 信号控制，但到了 DDR SDRAM-DIMM 时代，可能由于性能的原因 Buffered 模式被取消了。

在 Unb-DIMM 上，没有寄存器也就没了这个 Buffer，但它仍可具备 ECC 功能。这里需要强调的是，ECC 与 Registered 是两码事，前者是在逻辑上保证数据的安全，后者是在物理上保证内存系统的稳定工作。

提示：什么是 ECC？

ECC 的全称是错误检测与校正（Error Checking and Correction），他通过纠错码技术（也叫 ECC，Error Correction Code）来检查并纠正数据中的错误。所使用的 ECC 基本都是汉明码，它可以纠正一位数据的错误（详见本刊 2000 年第 12 期《让我们谈谈 RAID（二）》一文）。ECC 是额外生成的数据，在目前系统中，对于 64bit 的 DIMM，用 8bit 的汉明码来进行负责纠错，这样 ECC 型 DIMM 的位宽就从 64bit 变成了 72bit。  
ECC 的生成与纠错由北桥负责，在向内存写入数据的同时，北桥生成 ECC 通过 DIMM 上专用的引脚传给专用 ECC 芯片进行存储，而不是将 ECC 打散插入原有数据再统一存储。读取时，由 ECC 芯片随数据发送相应的 ECC 代码通过专用的引脚传给北桥并进行校验。

数据芯片

数据芯片

数据芯片

数据芯片

ECC 校验数据芯片

数据芯片

数据芯片

数据芯片

数据芯片

ECC 型 Unb-DIMM 示意图，其中的专用 ECC 芯片就用来存储 ECC 数据，在无 ECC 型 DIMM 上这颗芯片被取消



德国 Infineon 公司推出的容量高达 2GB 的 PC2100 Reg-DIMM



## 5.2 DIMM 引脚的基本设计

讲完 Unb-DIMM 与 Reg-DIMM 的不同之后，现在来看看 DIMM 引脚上的不同。其实，从内存芯片的引脚上就能推断出一些 DIMM 的引脚，因为芯片最终要通过 DIMM 来与主板打交道的。

首先，DIMM 肯定要有 64 个引脚用来数据的传输，而且要有 Ax 地址线、L-Bank 地址线、片选、数据掩码、电源、RAS、CAS… …等信号，另外，ECC 型与 Reg 型 DIMM 要有额外的标定引脚，下面我就以 SDRAM 和 DDR SDRAM 为例，分 Unb-DIMM 和 Reg-DIMM 来介绍一下 DIMM 都包含有哪些的引脚。

SDRAM-DIMM 与 DDR SDRAM-DIMM 引脚数量及定义（“-”表示没有此引脚）

类别	引脚名称	定义	SDRAM DIMM				DDR SDRAM-DIMM			
			Unb	ECC	Reg	ECC	Unb	ECC	Reg	ECC
电源	Vref	参考电压	-	-	-	-	1	1	1	1
	VddID	Vdd 标识	-	-	-	-	1	1	1	1
	Vdd	电源	17	17	17	17	9	9	9	9
	VddQ	芯片 I/O 电源	-	-	-	-	16	16	16	16
	Vss	接地	18	18	18	18	22	22	22	22
地址/命令	CS	片选（注 1）	4	4	4	4	2	2	2	2
	Ax	地址线	14	14	14	14	14	14	14	14
	BA（注 2）	L-Bank 地址线	2	2	2	2	3	3	3	3
	RAS	行选通脉冲	1	1	1	1	1	1	1	1
	CAS	列选通脉冲	1	1	1	1	1	1	1	1
	WE	写允许	1	1	1	1	1	1	1	1
时钟	CK	时钟	4	4	4	4	3	3	1	1
	CK#	反相时钟	-	-	-	-	3	3	1	1
	CKE	时钟有效	2	2	2	2	2	2	2	2
	SA	SPD 地址线	3	3	3	3	3	3	3	3
	WP	SPD 写保护	1	1	1	1	-	-	-	-

SPD	SDA	SPD 数据 I/O	1	1	1	1	1	1	1	1
	SCL	SPD 时钟	1	1	1	1	1	1	1	1
	VddSPD	SPD 电源	-	-	-	-	1	1	1	1
数据与掩码	DQ	数据 I/O	64	64	64	64	64	64	64	64
	DQS	数据选取脉冲	-	-	-	-	8	9	8	9
	DQM	数据 I/O 掩码	8	8	8	8	-	-	-	-
	DM	数据写入掩码	-	-	-	-	8	9	8	9
	CB	ECC 校验数据	-	8	-	8	-	8	-	8
寄存型专用	Reg	寄存型模组	-	-	1	1	-	-	-	-
	Reset	寄存器复位	-	-	-	-	-	-	1	1
保留	NC（注 3）	未使用的引脚	26	18	25	17	19	9	22	12
引脚数量（Pin）			168				184			

注：1、DDR SDRAM-DIMM 中的 CS 信号数量最高也可达 4 个，但目前基本都是 2 个 CS 信号设计，如果为 4 个，NC 的数量就要减少两个

2、DDR SDRAM-DIMM 中定义了 3 个 L-Bank 地址线，但目前只采用两个的设计，用到 3 个 L-Bank 地址线的将是 DDR-II 内存

3、使用引脚是指在设计中预留的引脚，供未来的一些设计使用

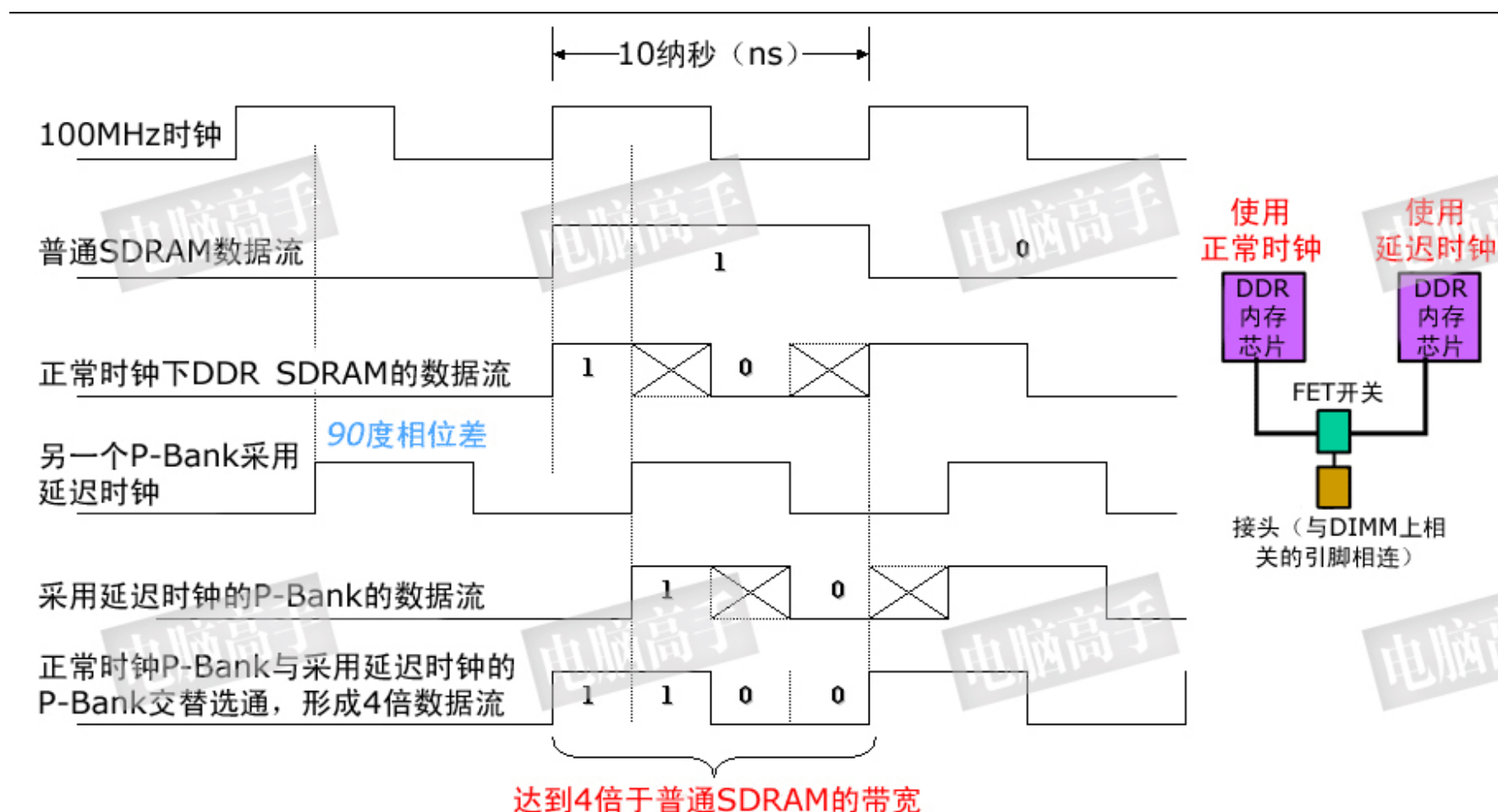
从上面的引脚信号列表中，大家应该能了解到 DIMM 的大体情况了。其中很多信号定义是不是非常熟悉？从中可以看到，在 DDR SDRAM 时代已经为 8 个 L-Bank 做好了准备，但业界显然没有利用到它，不光是内存厂商，DDR 芯片组中似乎没有支持 8 个 L-Bank 的设计。

还有就是 CS 信号，从 SDRAM 到 DDR，都有 4 个 CS 的设计，但目前的 DIMM 还都是双 P-Bank 的设计，不同的是，SDRAM-DIMM 上，4 个 CS 是必须的，两个 CS 对应一个 P-Bank 芯片集，但到了 DDR 时代，可能是技术与工艺的进步，一个 CS 就控制了一个 P-Bank。总之，当我们了解了芯片的引脚设计后，对 DIMM 的引脚组成也就不再陌生。有兴趣的读者，可以自行深入研究。

提示：什么是 SPD？它是怎么工作的
<p>SPD（Serial Presence Detect），笔者翻译为“配置（存在位）串行探测”，而不是“连续存在探测”，如果单从字意上理解，后者的翻译并没有问题，但从其真正用意与工作方式来看，前者更准确一些。为什么呢？下面具体说说。</p> <p>SPD 是一组关于内存模块的配置信息，如 P-Bank 数量、电压、行地址/列地址数量、位宽、各种主要操作时序（如 CL、tRCD、tRP 等）……它们存放在一个容量为 256 字节的 EEPROM（Electrically Erasable Programmable Read Only Memory，电擦除可编程只读存储器）中。实际上，包括 RIMM 在内，SPD 的有效信息只用了 128 个字节。一般的，一个字节至少对应一种参数，有的参数需要多个字节来表述（如产品续列号，生产商在 JEDEC 组织中的代码）。其中，一个字节中的每个 bit 都可能用来表示这一参数的具体数值。由于 SPD 的信息很多，在此就不一一列出了，有兴趣的读者可以参阅相关文档。</p> <p>SPD 的信息由模组生产商写入，主要用途就是协助北桥芯片精确调整内存的物理/时序参数，以达到最佳的使用效果。如果在 BIOS 中将内存设置选项定为“By SPD”，那么在开机时，北桥会根据 SPD 中的参数信息来自动配置相应的内存时序与控制寄存器，避免人为出现调校错误而引起故障。当然，对于 DIYer 来说，也可以自由调整时序与控制参数（物理参数仍要借助 SPD 或北桥自己检测来确定）。</p> <p>那么 SPD 的信息是怎么传送的呢？从 DIMM 的引脚列表中，我们知道了有关 SPD 的引脚信号，与检测操作有关的分别是三个地址线、一个数据线和一个时钟。SPD 所使用的 EEPROM 的时钟频率大都为 100KHz，SPD 的数据通过一条数据线进行串行式交换，这就是所谓“串行探测”的含义，理论上数据的交换速度为 100Kbps，即 12500 字节/秒，读取 128 个字节约需 0.01 秒。在操作中，地址线提供 SPD 芯片地址（因为每个 DIMM 上都有一个 SPD，目前常见的 SPD 地址线为 3 个，也就意味着可访问最多 8 个 DIMM 上的 SPD 芯片），然后通过数据线用 8 个 SPD 时钟周期串行传输字节地址，之后指定字节的数据再从数据线串行传出，并由系统识别，然后用来配置相关寄存器。</p> <p>显然，当我们了解了 SPD 的具体信息后，完全可以通过 SPD 的信息格式与读写操作的方式入手，来编一个 SPD 识别软件，有关编程基础的朋友不妨试试。</p>

## 5.3 QBM 型 DIMM

之所以在前文没有介绍四倍带宽内存（QBM，Quad Band Memory），就是因为不是针对芯片的技术，而针对 DIMM 的技术。它诞生于 DDR 时代，是 Kentron 公司为了解决 DDR 带宽提供困难而提出的设计方案。主要的思路就是让 DIMM 上的两个 P-Bank 交错工作，而交错的时钟周期为原始时钟的  $1/4$ ，即相位相差  $90^\circ$ 。



QBM 的工作时序图

上图是 QBM 的工作时序图，第二个 P-Bank 的工作时钟与第一个 P-Bank 相差  $90^\circ$  ( $1/4$  周期)，这样在第一个 P-Bank 时钟的高/低电平中部就是第二个 P-Bank 的触发点，两者都是 DDR 传输，从而在一个时钟周期内完成 4 次数据触发，实现四倍带宽。为了控制两个 P-Bank 中同一位置的芯片交错工作，模组上要为每组芯片（在 QBM 模组上，一个 P-Bank 位于一侧，两个 P-Bank 中位置相对的芯片为一组）设置一个开关，以控制不同 P-Bank 间的通断。并且还要为延迟  $1/4$  周期的 P-Bank 提供一个 PLL 以保证相位差的准确性。

QBM 的设计是非常巧妙的，经过对现有的 DDR 模组的改装，配合新的芯片组即可将带宽提高一倍，有点类似于 32bit RIMM，在一个模组上实现了双通道的功能，只是 QBM 不是双通道并发，而是双通道交错，通过更高的传输频率实现高带宽。但是新增加的开关与 PLL 元件将增加一定的成本，不过与其所能提供的带宽相比，还是比较划算的。



	QBM533 (DDR I-266)	DDR I-333	DDR I-400	DDR II-400/533
	512MB	512MB	512MB	512MB
带宽	✓ PC4200	PC2700	PC3200	PC3200/4200
PCB成本	\$2	\$2	\$2	目前不可用
内存芯片成本	✓ \$104 (\$6.50x16)	\$128 (\$8x16)	\$160 (\$10x16)	目前不可用
STD元件	\$3	\$3	\$3	目前不可用
开关IC	\$12 (\$1.50 x 8)	N/A	N/A	N/A
PLL	\$2	N/A	N/A	N/A
装配/检测	\$5	\$5	\$5	N/A
模组成本	✓ \$128	\$138	\$170 可用性有限	目前不可用 最早要到2004年

Kentron 公司给出的 QBM 与其他内存方案的成本比较表，从中可以看出 QBM 有较高的性价比

但是，开关元件的同步性对于 QBM 是个考验，时钟频率越高，对开关的控制精度就越高。目前，有不少大牌的模组厂商（如 Infineon）都在论证 QBM 的可行性与可靠性，据部分厂商透露，在使用 DDR-333 或之前标准时，QBM 的表现良好，但到了 DDR-400，QBM 的可靠性就会降低，如果克服这一个问题，那么延迟又会大幅提高。所以，QBM 目前的可行标准是 QBM533(DDR-266)和 QBM667(DDR-333)。VIA 在 P4X800 中将要支持的标准也是 QBM533，虽然不能使用 DDR-400，但它的 5.4GB/s 带宽(QBM667)在目前仍是无敌的。

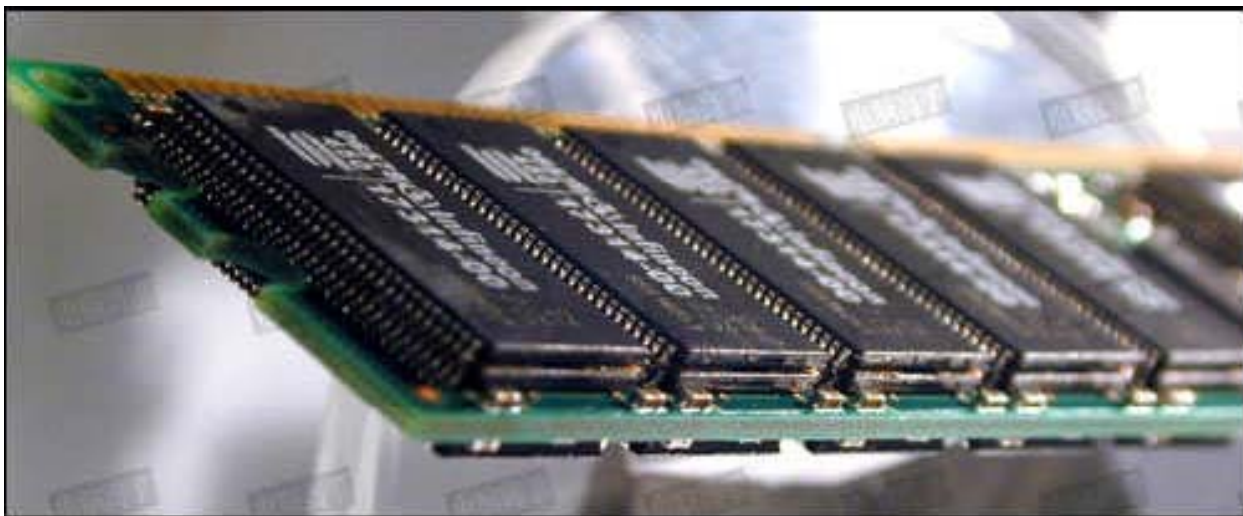
不过，由于 QBM 是针对模组的技术，所以理论上 QBM 可适用于任何 DIMM，包括 SDRAM 和 DDR-II 的 DIMM，Kentron 也有此计划研制 QBM 型 DDR-II DIMM，以保持 QBM 的生命力。另外，Kentron 已将 QBM 标准上报 JEDEC 审批，目前还不知能否通过。很多模组厂商也都在观望，毕竟 QBM 转产是很容易的，就看市场情况了。所以，QBM 虽然设计巧妙，但得到的支持并不强劲，以 Kentron 及 QBM 联盟的生产能力，显然不足以完成普及任务，一切就看 P4X800 的市场效果了。

### 5.4 模组的堆叠装配

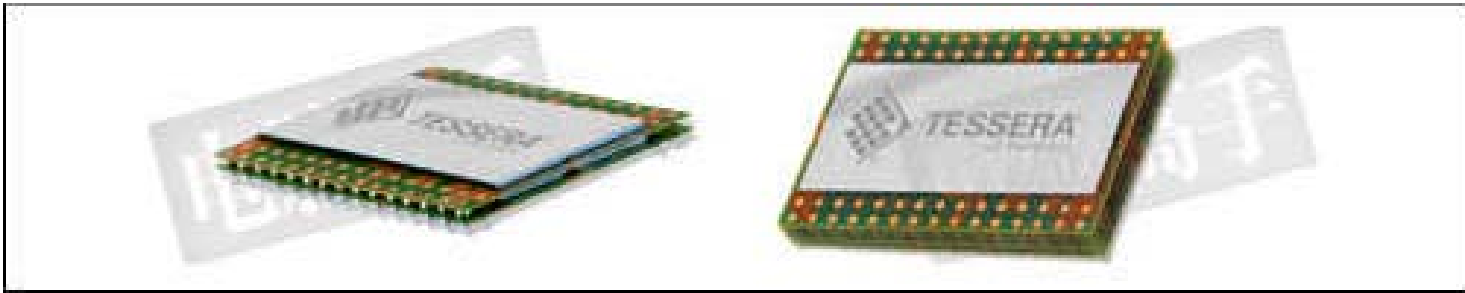
当内存芯片容量无法迅速提高的时候，高容量模组如何设计就体现了厂商间的真正实力，由于高容量模组针对的是高端应用市场，所以谁能在容量上有所突破就意味着滚滚商机。就模组而言，芯片

基本是固定的，所以芯片堆叠装配（Stack Assembly）技术就是增加容量的首选。

这方面除了 Elpida、Kentron、Kingston 等公司较早以前提出的 TCP、FEMMA、EPOC 等堆叠形式外，著名的封装技术开发商 Tessera 公司（它在 1990 年因研制出 CSP 封装而闻名于世）近期宣布了他们的 4 枚芯片堆叠装配的模组技术（TCP 与 EPOC 都是两芯片堆叠）——  $\mu$  Z Package，当然，芯片本身的封装也要有相应的调整。而 Infineon 公司也推出了普通 TSOP-II 技术的双芯片堆叠装配技术。显然，模组厂商都想利用有限的空间（毕竟在主板上插槽之间的距离是有限的）尽量提高装配容量，若再配合 SiP 封装形式的内存芯片，DIMM 的扩容就如虎添翼了。



Infineon 的采用 TSOP-II 堆叠封装的模组，容量高达 2GB



Tessera 公司为高容量模组开发的 4 枚芯片堆叠装配技术  $\mu$  Z Package