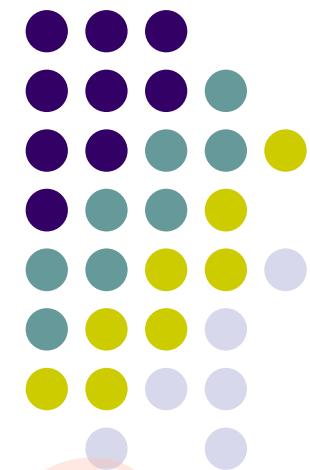


嵌入式系统设计与应用

第四章 基于总线的计算机系统

西安交通大学电信学院
孙宏滨





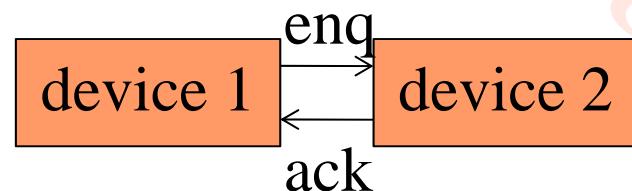
1 CPU总线

- 计算机系统包括微处理器（CPU）、输入输出设备（I/O Devices）和存储器组成。
- **总线**是CPU与存储器和设备通信的机制，定义它们之间通信的协议。
- **协议**（protocol）定义了设备之间如何通信。
- 握手保证当两个设备想通信时，一个准备发送，另一个准备好接收。
- 握手协议使用一对线进行握手：enq（表示查询enquire）ack（表示应答acknowledge）



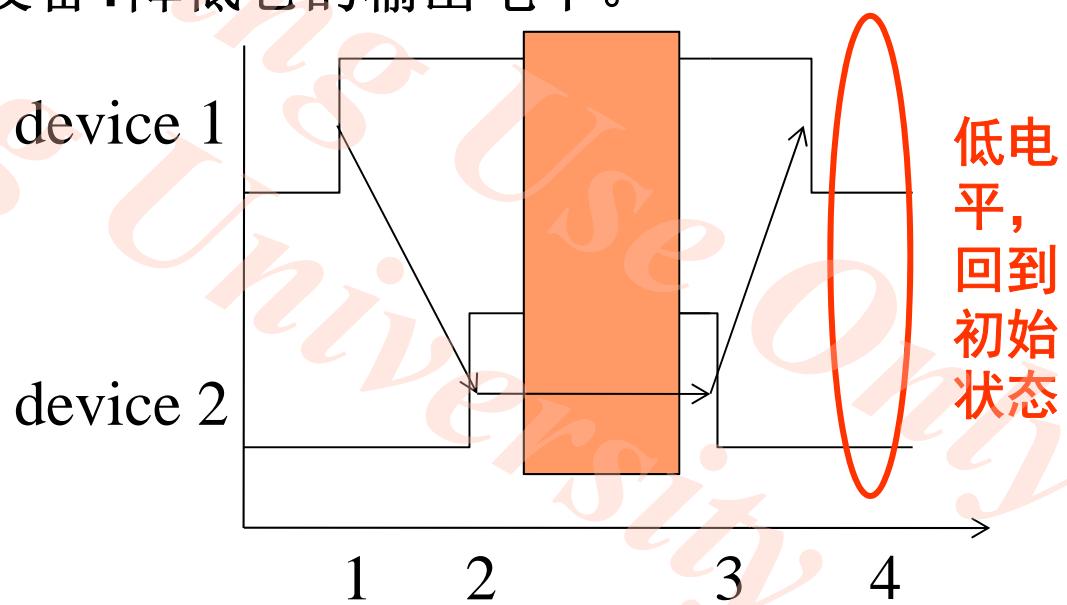
四周期握手 (four-cycle handshake)

- 设备1升高输出电平发查询信号，以告诉设备2应准备监听数据。
- 当设备2准备好接收数据时，它使应答信号为高电平。这时设备1和设备2已准备好发送或接收。
- 数据传送完毕，设备2使应答信号为低电平，表示已经接收完数据。
- 检测到应答信号变低，设备1降低它的输出电平。



四周期握手

Four-cycle handshake

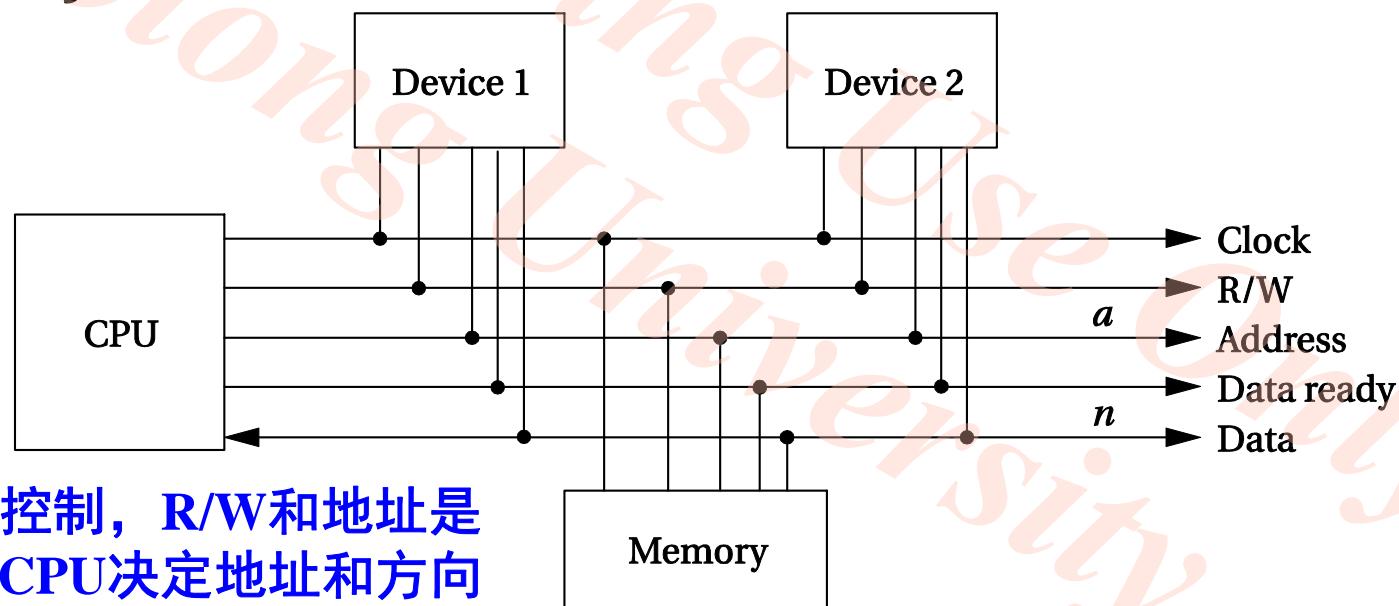




典型总线结构

- **Clock**: 提供总线组件各部分同步。
- 当总线读时，**R/W**为1；当总线写时，**R/W**为0。
- **Address**是一个 a 位信号束（**bundle**），提供访问地址。
- **Data**是一个 n 位信号束，可从**CPU**得到数据或传送数据。
- **Data ready**当数据束上的值有效时发信号。

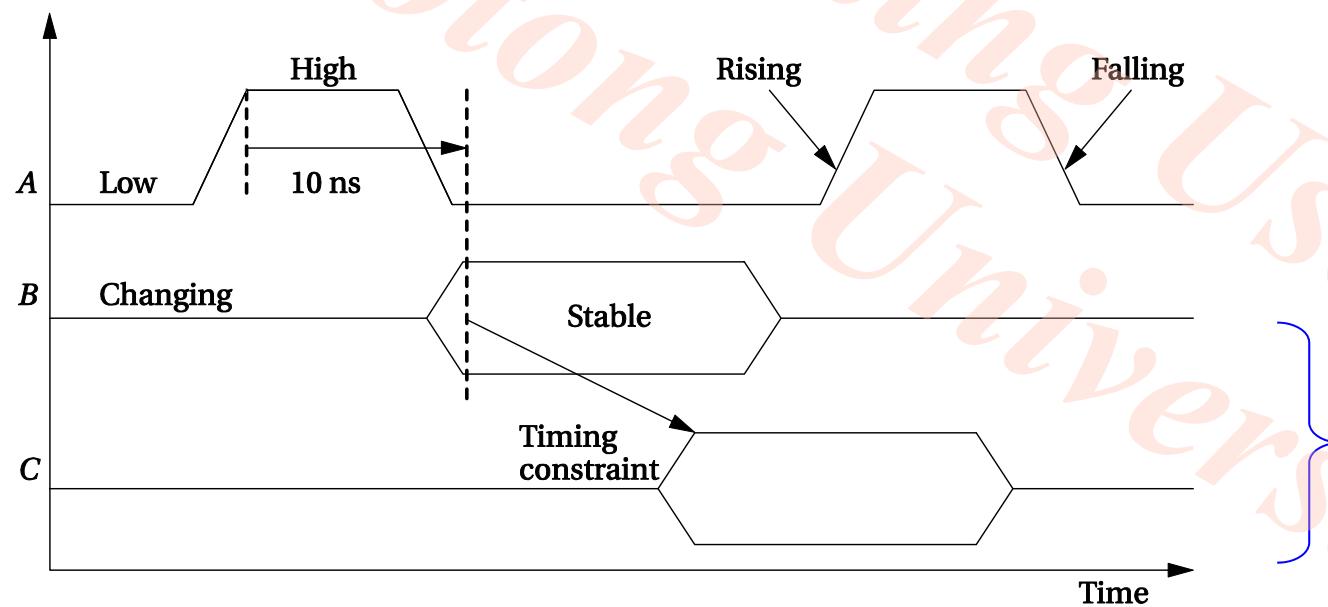
一个支持
读和写的
典型总线
结构





时序图

- 时序图表示总线上的信号如何随时间变化。
- 为了保证该信号在正确时间达到正确值，时序图经常显示定时约束（**timing constraint**）。
- 定时约束既定义顺序也定义时间差。

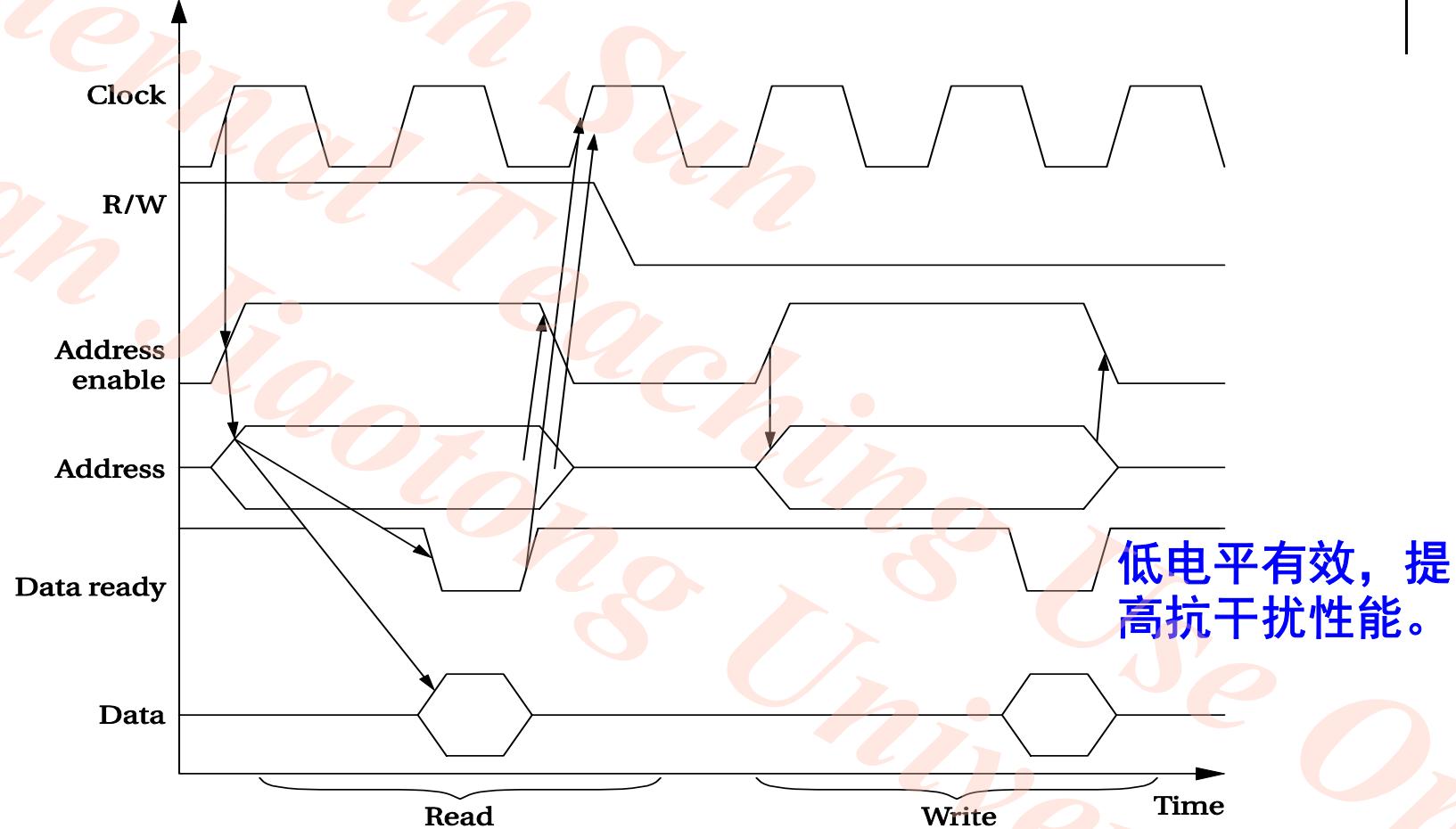


0和1之间变化的标准波形

“稳定状态”
和“变化状态”
之间交替变化。



示例总线时序图



当要传输数据时，通知CPU和设备的握手由应答方的数据ready形成，这对查询方式是隐式的。因为总线通常处于读方式，因此查询无须指定，应答必须提供。

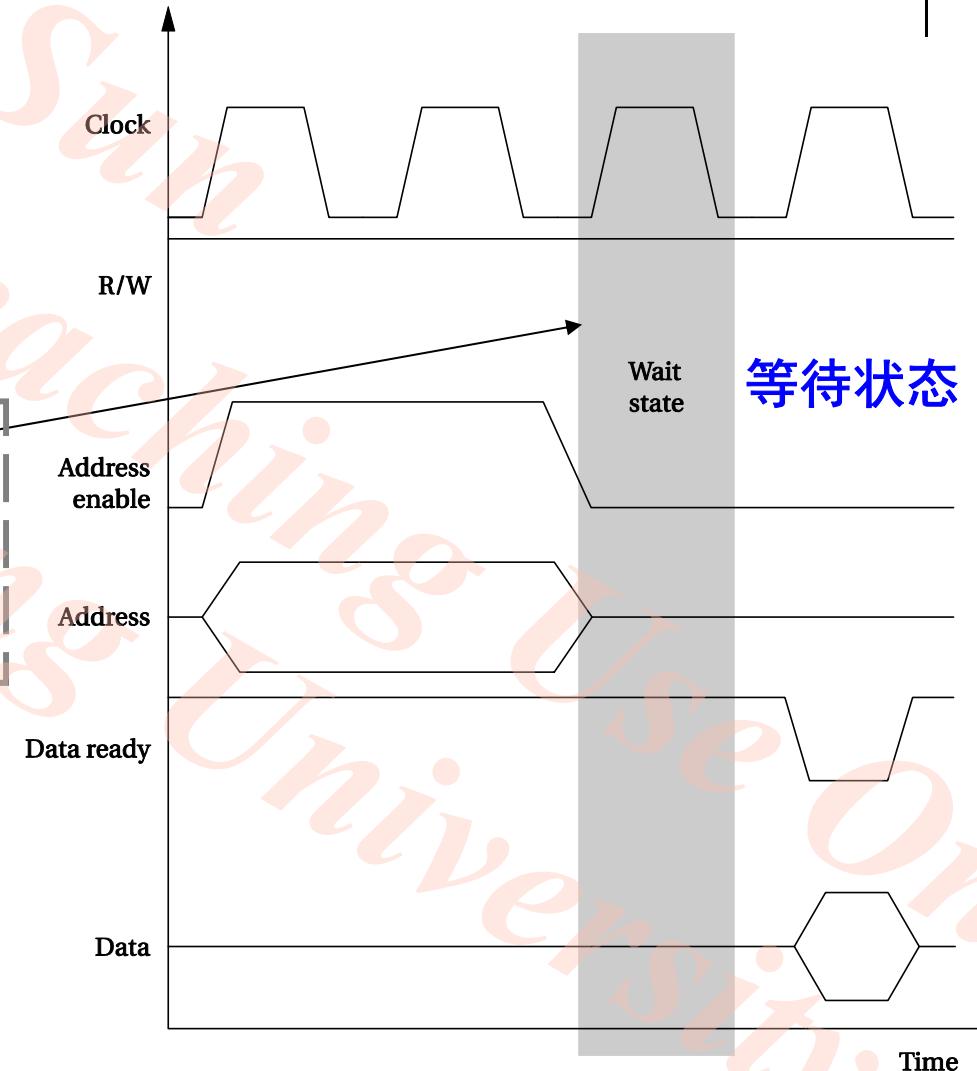


等待状态

用data ready信号可将总线连到速度比它慢的外设上，外设无须立即确认data ready信号。

在数据可被确定的最短时间与它真正被确定的时间之间的周期被称为**等待时间**。

等待状态通常用来将慢速、不贵的存储器连到总线上。



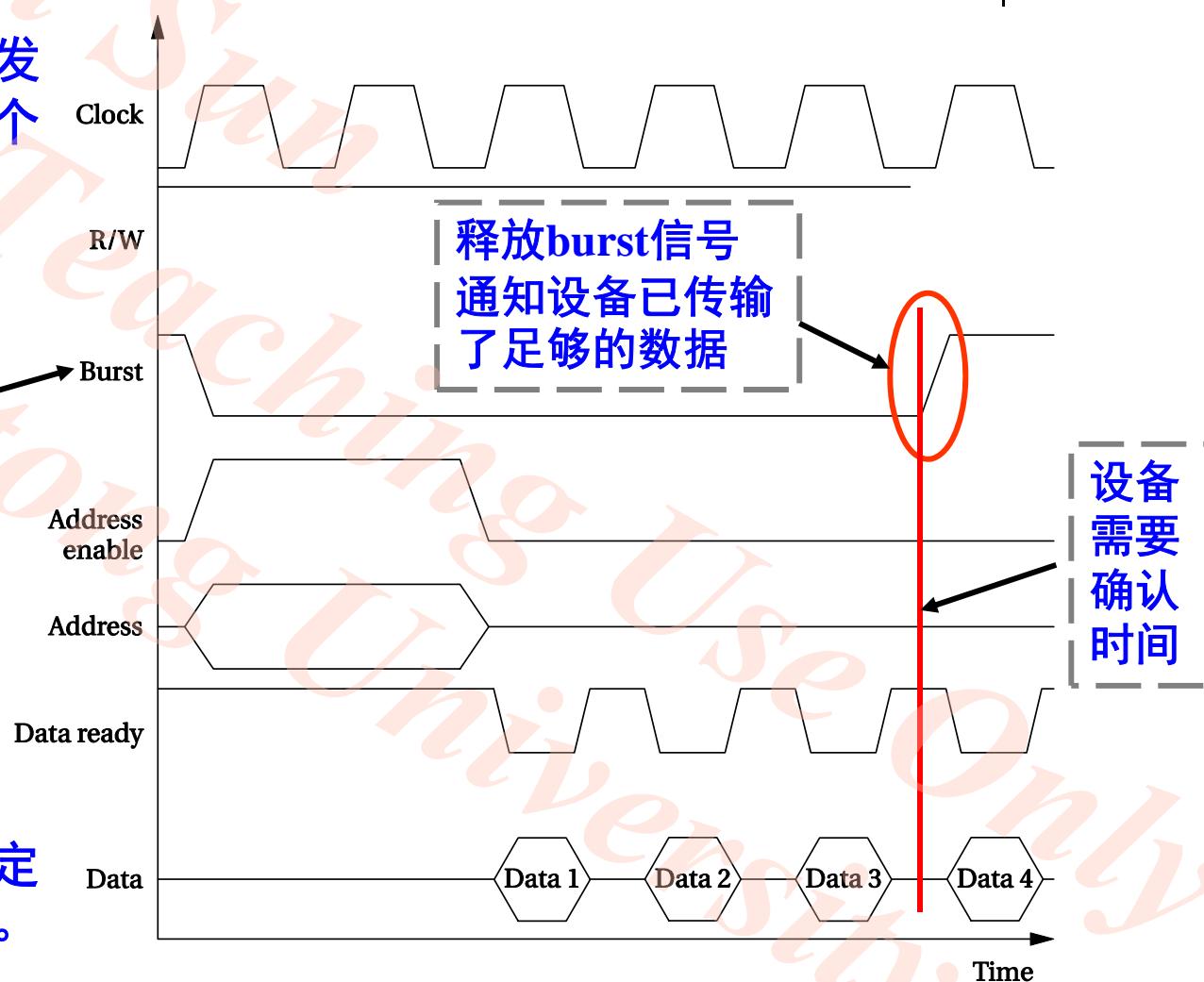


突发传输 (burst transfer)

突发读事务中，CPU发送一个地址但接收一个数据值序列。

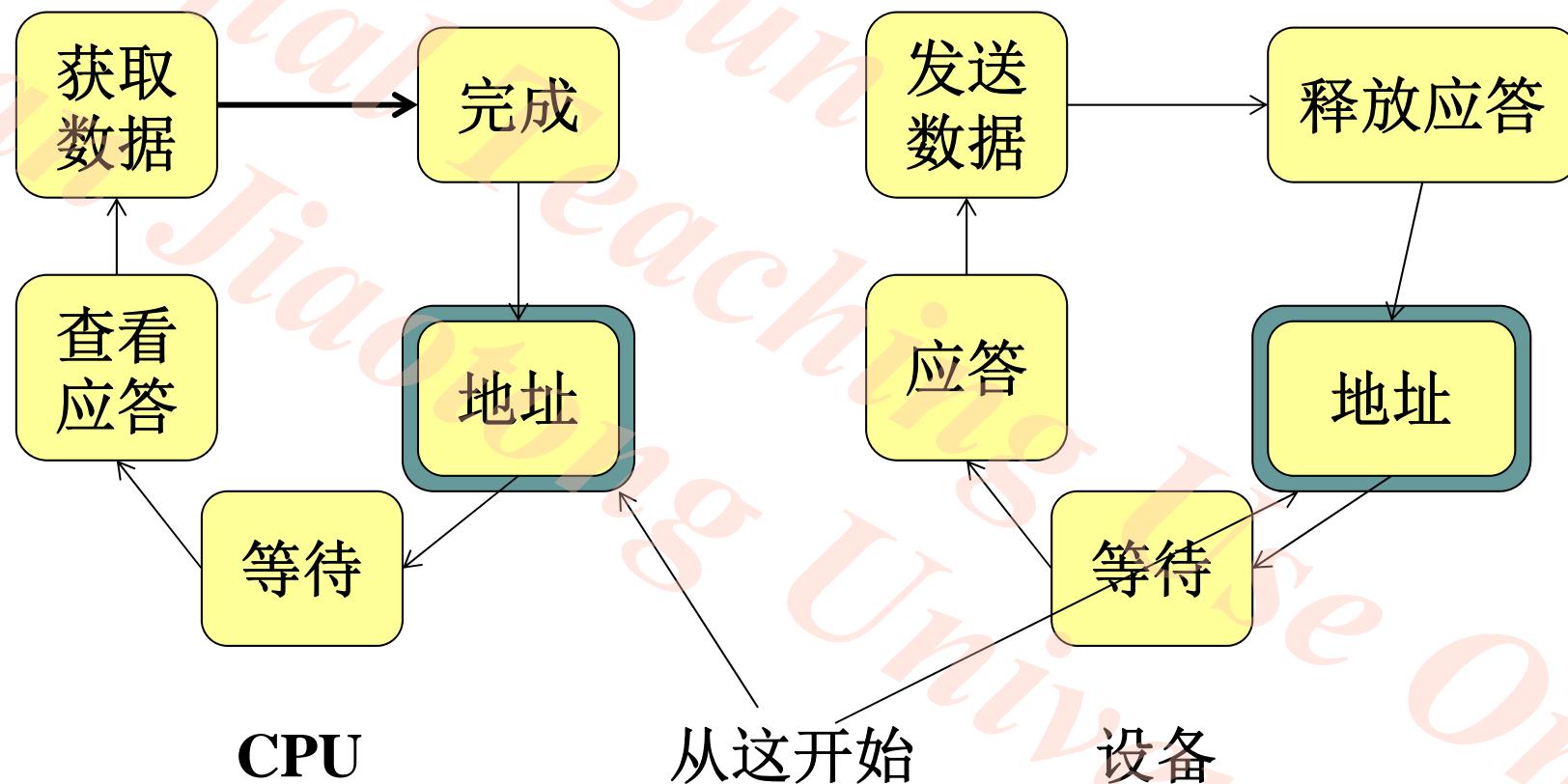
Burst 线路，在当前事务为突发事务时发信号。

这些数据值来自从给定地址开始的存储单元。





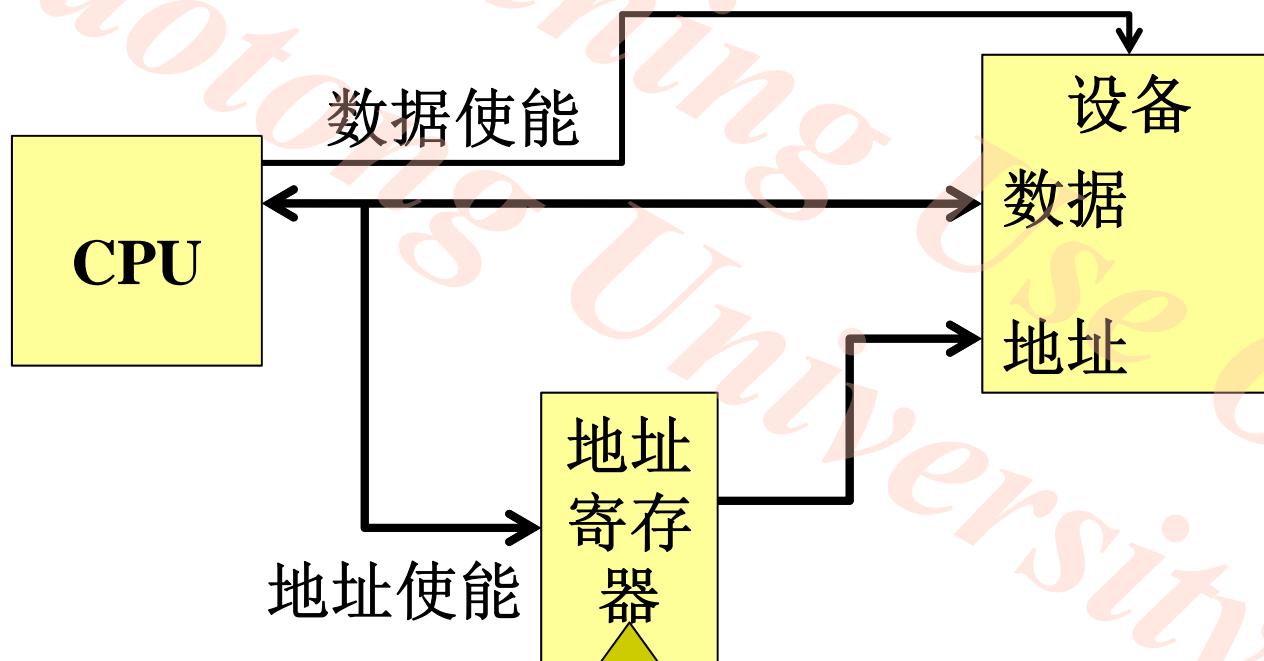
总线的状态机视图





总线数据与地址复用

- 总线多路复用地址和数据，用额外的控制线来分辨地址/数据线上的值是地址还是数据。
- 在组合地址/数据上地址先到，数据随后，地址可以存于寄存器中直到数据到来，以便二者同时传送。



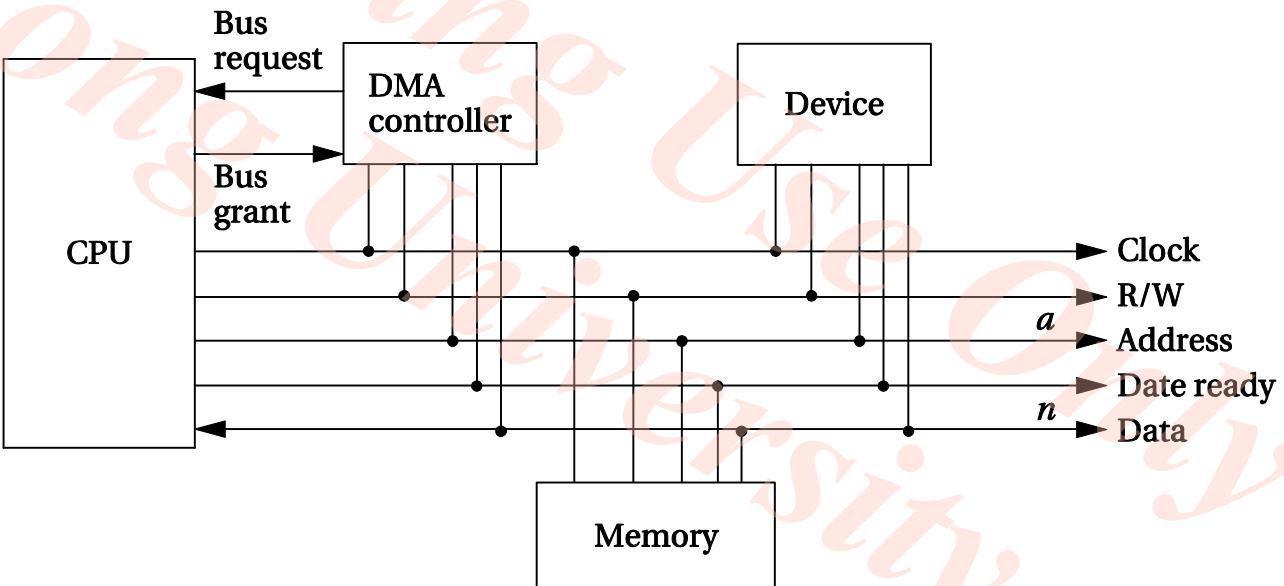


直接存储器访问DMA

- 直接存储器访问（**direct memory access, DMA**）是不由**CPU**控制器读写的总线操作。
- DMA传输由**DMA**控制器控制，它从**CPU**请求总线控制，得到控制权后，**DMA**控制器直接在设备和存储器之间执行读写操作。

总线请求 (bus request)
是CPU的输入信号，通过它DMA控制器请求
总线所有权。

总线授权 (bus grant)
信号表示总线已经授权
给DMA控制器。



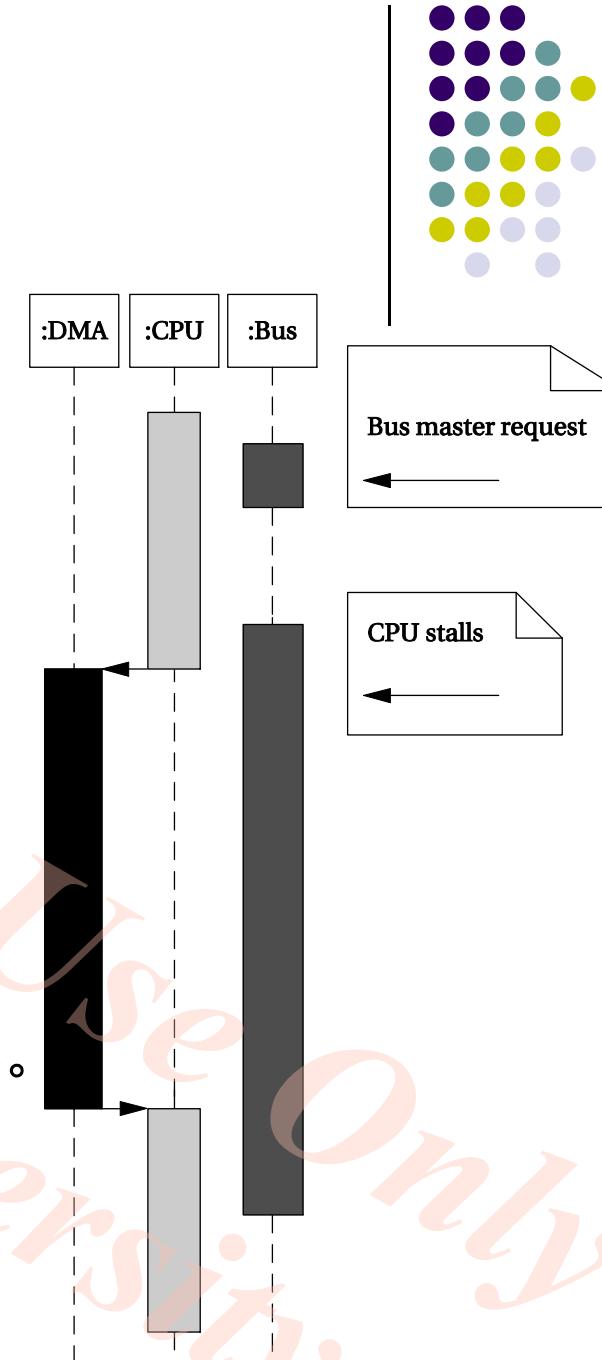


总线主控器

- 能够启动自己总线传输的设备叫**总线主控器**（**bus master**）。
- 默认情况下，**CPU**是总线控制器，由**CPU**来启动**DMA**传输。**DMA**工作期间，**CPU**不使用总线。
- **DMA**控制器也可以申请控制总线，使用总线请求和总线授权两个信号来获得使用标准四周期握手协议的总线控制权。
- **CPU**在将总线控制权授予**DMA**控制器之前，将完成所有未完成的总线事务。一旦授予控制权，它就停止驱动别的总线信号。只要**DMA**控制器成为总线控制器，它就拥有对所有总线信号的控制权。

DMA控制器设置

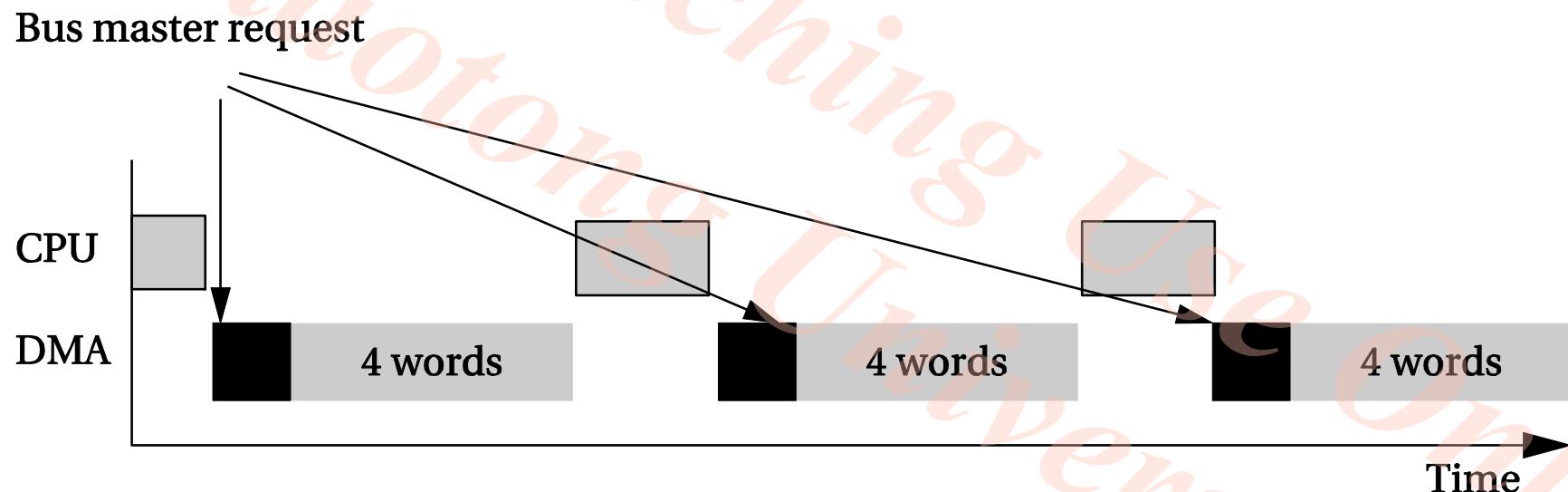
- 一旦**DMA**控制器成为总线控制器，可以使用总线进行读写操作。
- 事务完结后，**DMA**控制器通过释放总线请求信号把总线还给**CPU**，使**CPU**释放总线授权信号。
- DMA**控制器包括以下三个寄存器：
 - 起始地址寄存器：要从何处开始传输；
 - 长度寄存器：要传输的字的个数；
 - 状态寄存器：允许**CPU**操作**DMA**控制器。
- DMA**操作完成后，**DMA**控制器产生中断来告诉**CPU**传输结束。





DMA控制器工作模式

- DMA传输期间，CPU不能使用总线。如需要总线，则停止，等待DMA交出控制权。
- 为防止CPU空闲太长时间，大多数DMA控制器实现了一次只占用总线几个周期的工作模式（4、8、16个字）。



DMA请求的周期调度：每块数据后，DMA控制器交还控制权给CPU，进入一个预置的睡眠周期，之后再次为下一块数据传输请求总线。

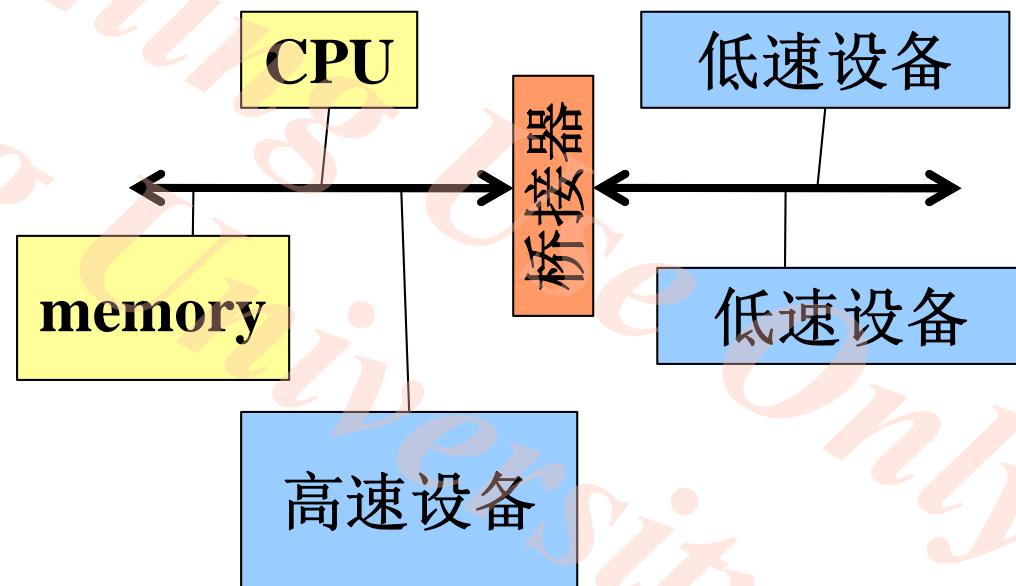


系统总线配置

- 一个微处理器系统可能含有多条总线，高速设备连到高速总线上，低速连的别的总线上。**桥接器**（bridge）使总线互连。

使用多总线与桥接器的优点：

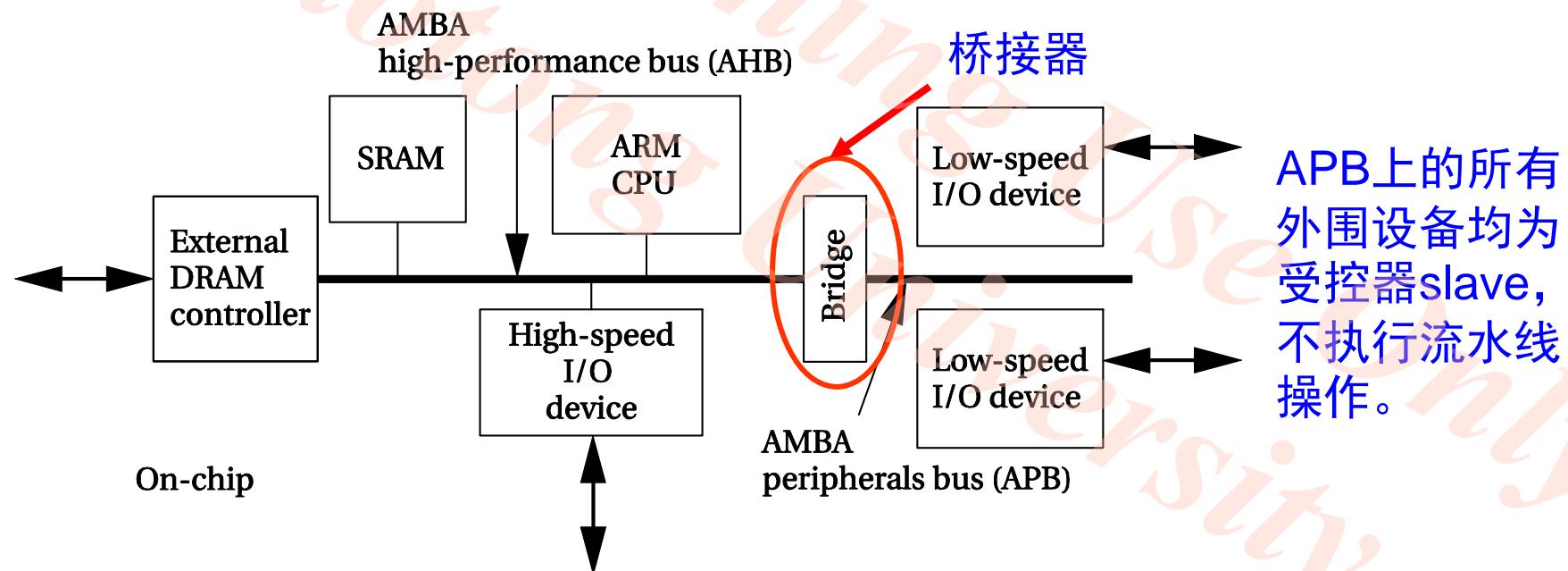
- 总线速度越高，能提供的数据连接越宽。
- 高速总线通常需要更昂贵的电路和连接器。可通过使用较慢、较便宜的总线来降低低速设备成本。
- 桥接器允许总线独立操作，这样在I/O操作中可提供某些并行性。





AMBA总线

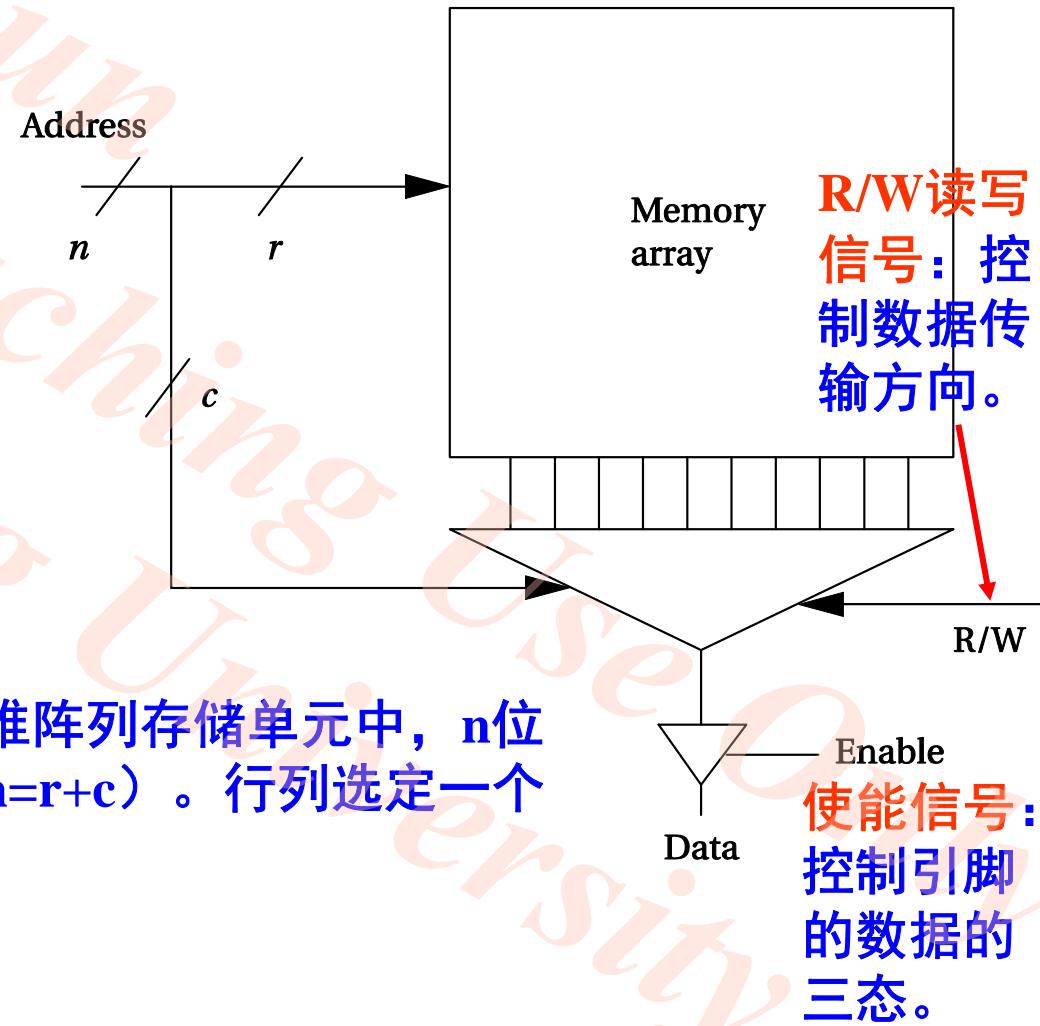
- AMBA规范包括两条总线：
 - 高性能总线**AHB**: 经过优化专为高速传输服务，直接连到**CPU**上。支持一些高性能特性：流水线技术、突发传输、分离事务和多总线控制器。
 - 外围总线**APB**: 设计简易，便于实现，功耗也较低。





2 存储设备

- 存储设备组织:
 - 容量
 - 数据宽度
- 256MB存储器
 - 64MX4位
 - 32MX8位
- 长宽比 (**aspect ratio**)



存储器内部，数据是存放在二维阵列存储单元中， n 位地址被分成行地址和列地址（ $n=r+c$ ）。行列选定一个特定存储单元（1位或子集）。



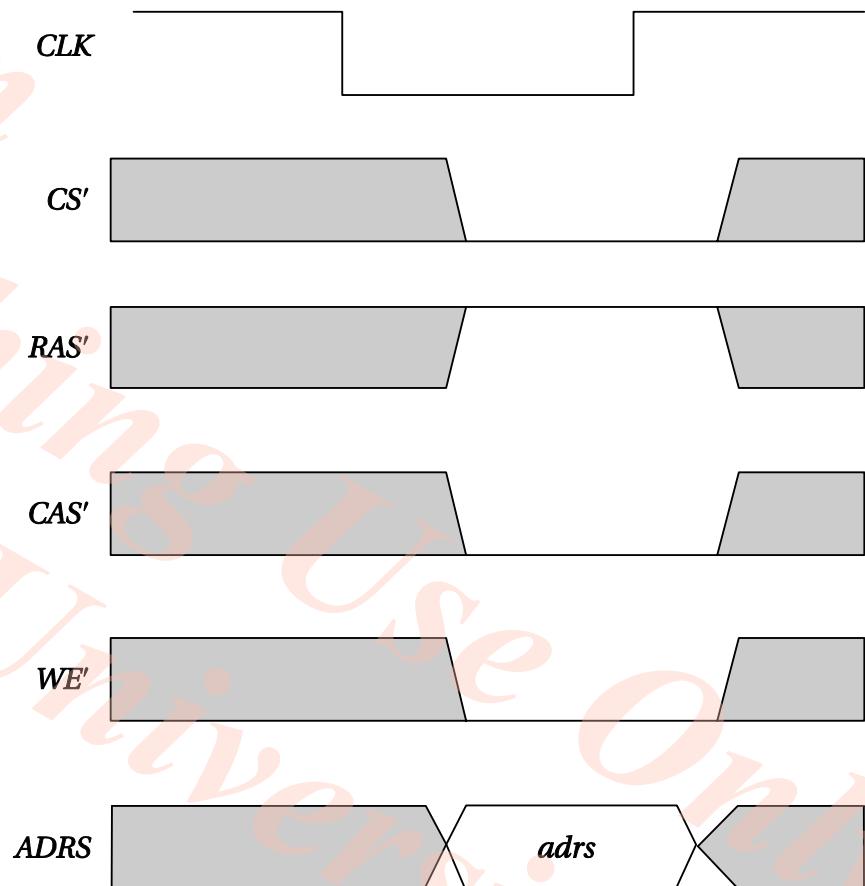
随机存储器（RAM）

- 随机存储器（RAM）可以被读和写，地址可以按任意顺序读取。
 - 静态随机存储器（SRAM）：小容量快速存储器。
 - 动态随机存储器（DRAM）：密度大、需要周期刷新。
- 同步动态随机存储器（Synchronous DRAM），使用时钟来提升DRAM的性能（最常见）。



SDRAM

- 行地址选择（**RAS**）和列地址选择（**CAS**）将地址分为两个部分。
- 双倍速率（**DDR**）**SDRAM**: 速度更快的同步动态随机存储器。
- **SIMM**: 单列直插内存模块
- **DIMM**: 双列直插内存模块





只读存储器

- 只读存储器 (**read-only memory ROM**) 用于固定数据预编程：嵌入式中很多代码或数据不随时间改变，对辐射感应错误相对不敏感。
- 工厂编程只读存储器 (**factory-programmed ROM**)
- 掩膜编程只读存储器 (**mask-programmed ROM**)
- 现场可编程只读存储器 (**field-programmable ROM**)



闪存存储器

- **闪存**: 非易失性存储器，可以电擦除。闪存使用标准系统电压擦写和编程，允许芯片在标准系统内部再编程。
- 随机读访问
- 写: 将块擦成1，然后写0。
- 写操作比读操作慢的多。**1.6 μs write, 70 ns read.**
- 擦写次数有限: 10~4到10~5次
- **NOR Flash**: 按字访问，整块擦除。
- **NAND Flash**: 读一页，整块擦除。更便宜，擦写速度块。



3 I/O设备 – 定时器和计数器

- 定时器和计数器很相似，区别主要在用途，而不是其逻辑电路。
- 它们都是由带有保存当前值的寄存器和向当前寄存器值加1的一个增量输入的加法器逻辑电路组成。
- 定时器让它的计数连到周期性时钟信号上以测量时间间隔，而计数器让它的计数连到非周期性信号上以计算外部事件的发生次数。
- 完成规定的定时或计数次数时通常会产生**中断**。



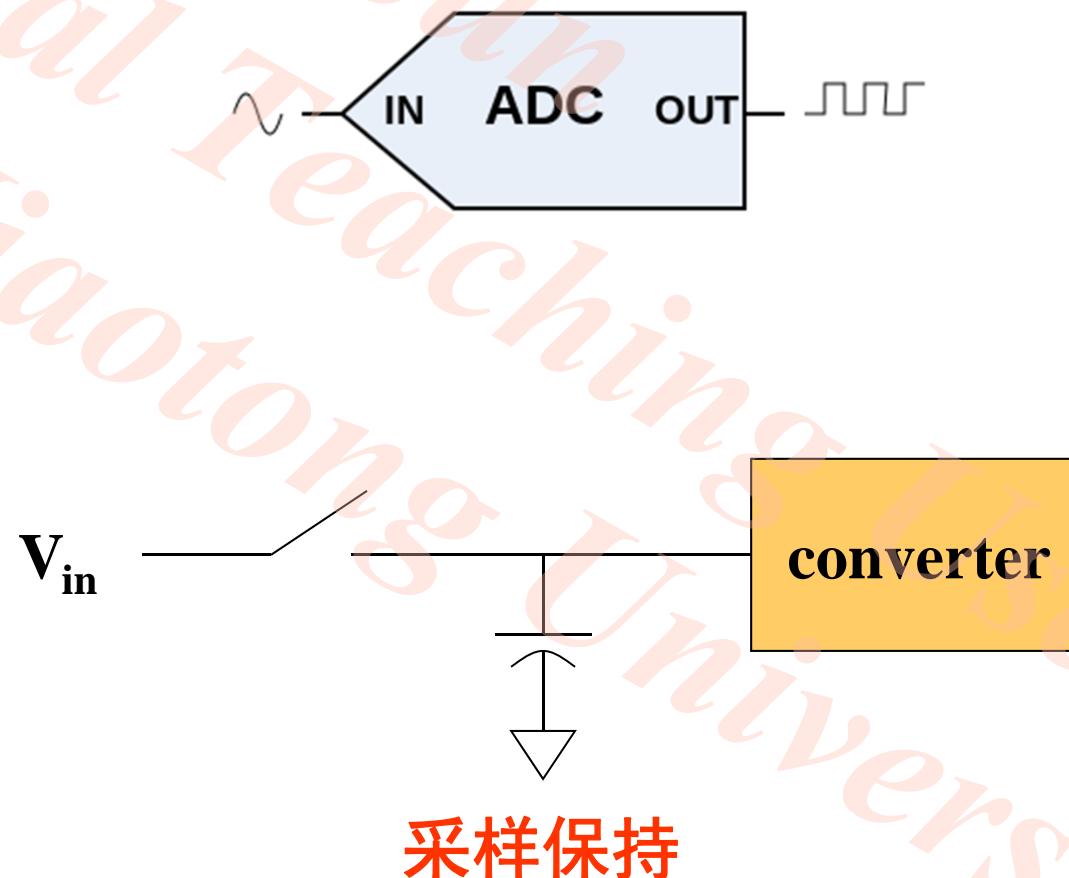
监视定时器

- 看门狗定时器（watchdog timer）是一个用于系统内部操作的I/O设备。
- 看门狗定时器连接到CPU总线和CPU复位线上。
- CPU的软件在定时器未达到超时限制前，周期性的重置监视定时器。
- 一旦定时器达到该限制，它的超时动作就复位处理器。





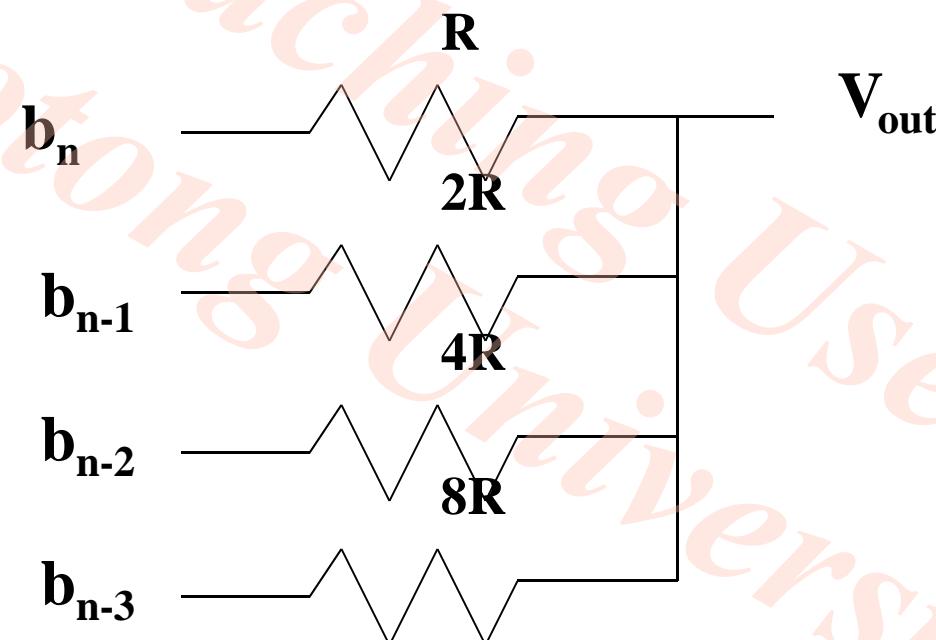
A/D转换器





D/A转换器

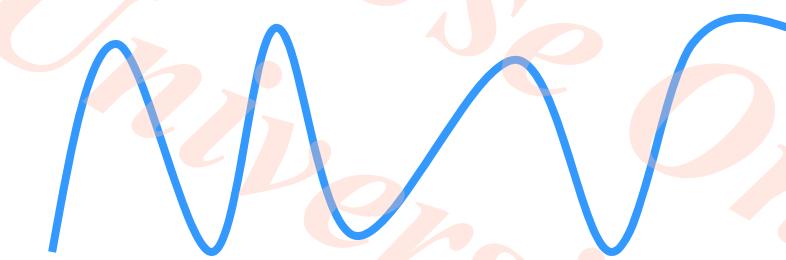
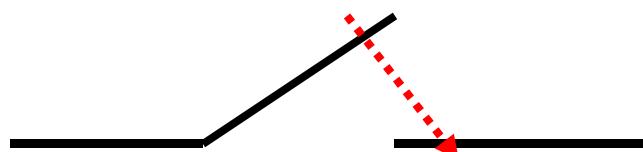
- D/A转换器接口仅包括数据值，输入值被连续转换成模拟信号。
- 使用电阻树（网络）。





开关 (switch)

- 开关使用机械接触来断开或接通电路。
- 机械开关的主要问题是颤动 (**bounce**)。当通过按下连接到开关臂上的按钮将开关压上时，压力导致接触颤动几次才停下来。显现为按了几次开关，给出错误的输出。
- 硬件**消颤** (**debouncing**) 使用一个单步定时器。
- 软件也可以用来消除开关输入颤动。

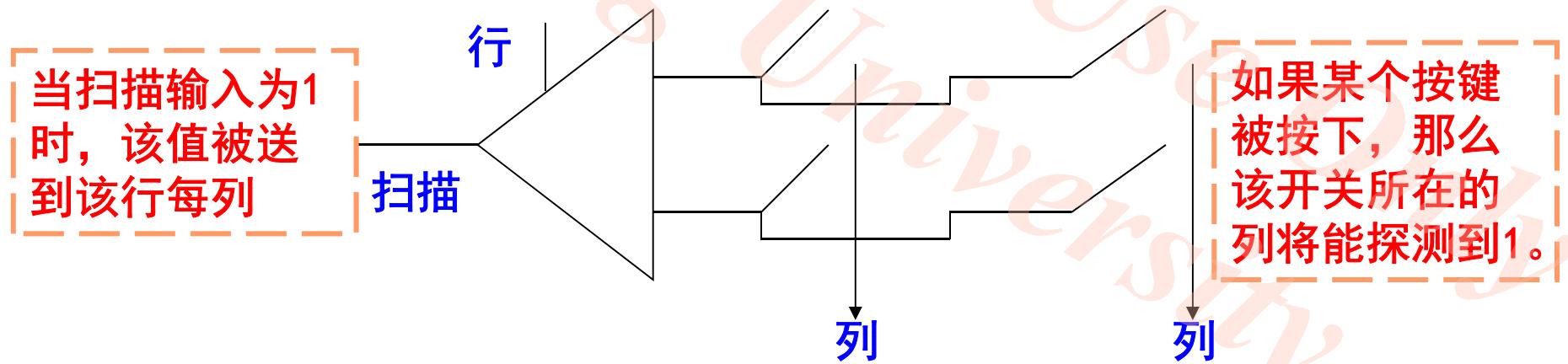


开关颤动



键盘 (keyboard)

- 键盘基本上由一个开关阵列组成，但它还包括一些逻辑电路来简化它到微处理器的接口。
- 编码键盘 (**encoded keyboard**) 使用编码来表示按下的开关，编码键盘的核心是开关扫描阵列。
- 扫描键盘阵列每次只读开关的一行。阵列左边的多路分配器选择要读的行。





PC键盘

- PC键盘，通常包括一个4位微处理器来预处理键盘输入。微处理器在计算机和键盘之间提供接口，除了可提供消颤之外，还有其他功能。
- Ctrl+Q类组合键
- N个键被同时按下，使得同时按键被识别、入栈，而在键被释放时，再依次传输。

Mac keyboard



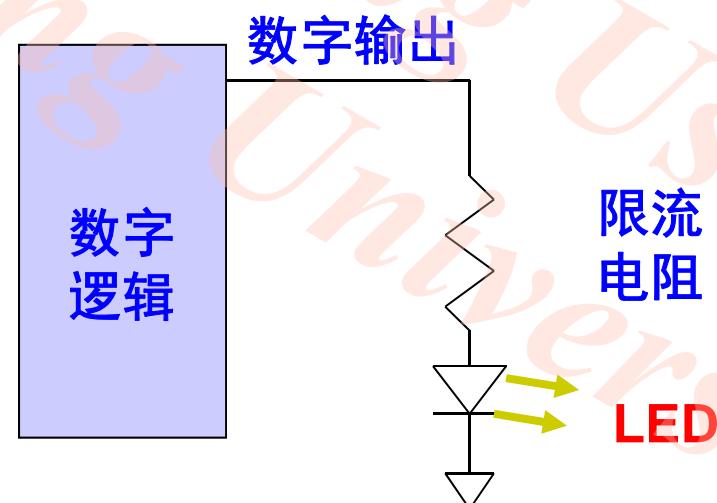
PC keyboard





发光二极管 (LED)

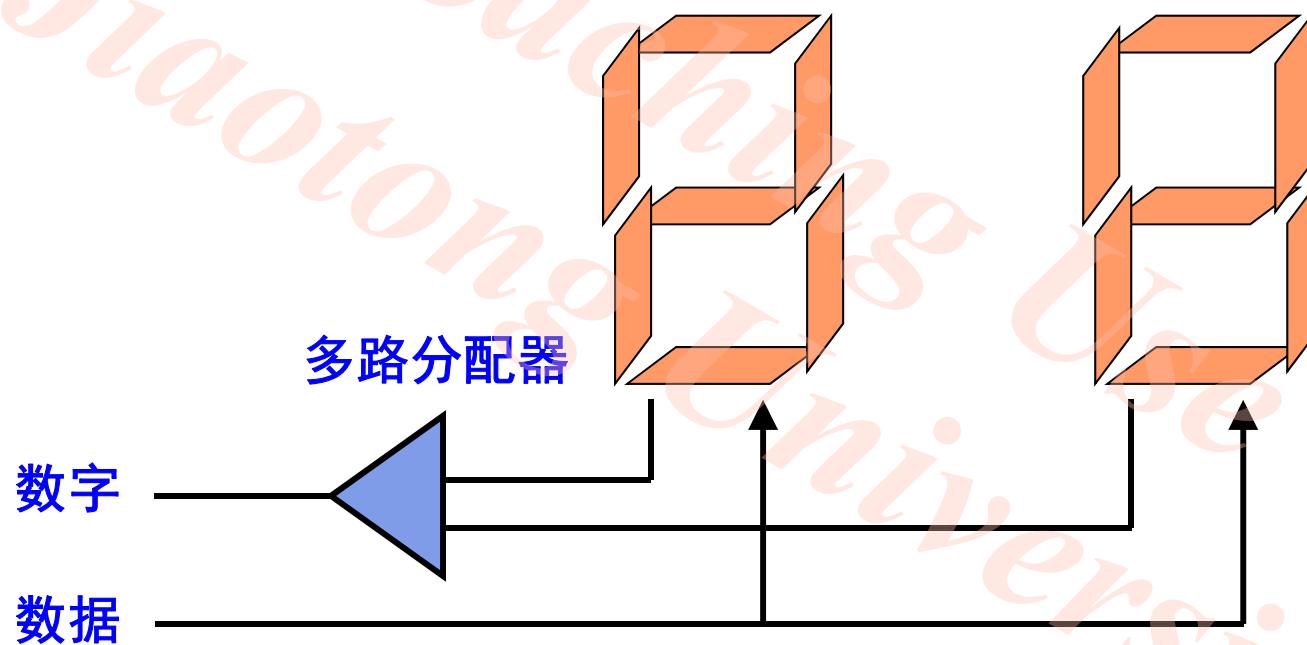
- 发光二极管本身经常被用作简单显示器，而发光二极管阵列是更复杂显示器的基础。
- 将电阻接在输出引脚和LED之间来分担数字输出电压和LED上的0.7伏导通电压之间的电压差。
- 当数字输出为0时，LED上的电压处于非工作区，二级管不发光。





7段LED数码管

- 显示元素较少的显示器直接由逻辑电路驱动。
- 利用视觉暂留效应扫描刷新。





液晶显示器 (LCD)

- LCD是当前市场占有率最高的显示器件。
- LCD是被动发光器件，当LCD中的电极产生电场时，液晶分子就会产生扭曲，从而将穿越其中的光线进行有规则的折射。
- Plasma、OLED (Organic Light-Emitting Diode)

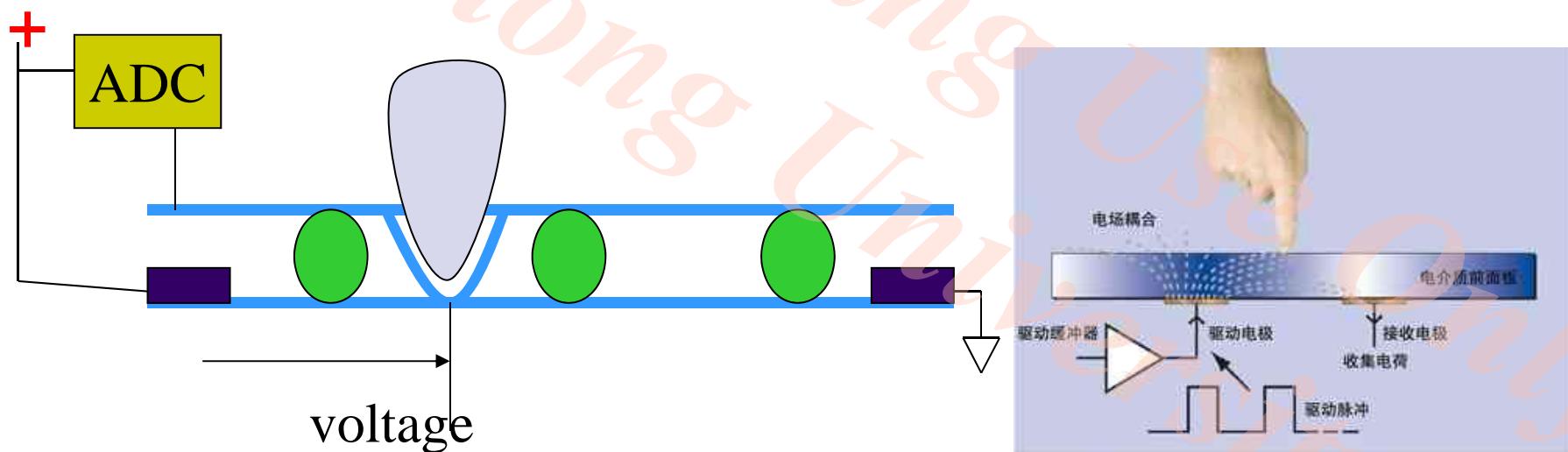
帧缓冲区是一个连到系统总线的RAM。微处理器可以以任意所需次序将值写入帧缓冲区。帧缓冲区中的像素通常以光栅的顺序写入显示器。

中文字模	位代码	字模信息
	0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0	0x08, 0x80 0x08, 0x80 0x08, 0x80 0x08, 0x80 0x11, 0xfe 0x11, 0x02 0x32, 0x04 0x54, 0x20 0x10, 0x20 0x10, 0xa8 0x10, 0xad 0x11, 0x26 0x12, 0x22 0x10, 0x20 0x10, 0x20 0x10, 0xa0 0x10, 0x40



触摸屏 (touchscreen)

- 触摸屏是覆盖在输出设备上的输入设备，可记录触摸位置。覆盖在显示器上，使用者可对显示信息作反映。
- 触摸屏分为电阻屏和电容屏两种。
- 使用电阻屏的手机（触笔）和电容屏手机（iphone）





苹果与电容触摸屏

- 电阻屏：精度高、噪声小；需要触笔、单点触控。
- 电容屏：精度低、噪声大；**用手指、多点触控。**
- 苹果之前的大部分触摸屏手机都是电阻屏。

苹果选择了电容屏

1. 通过采用电容屏，让手指取代了触笔
2. 通过改进操作系统，实现了多点触控
3. 通过制作支持多点触控的APP，实现了最佳用户体验

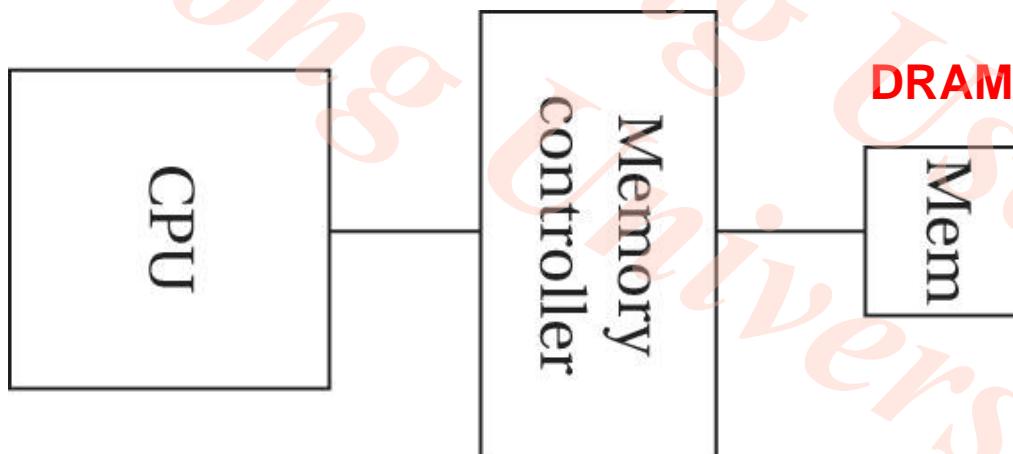
纵向系统级创新！！！





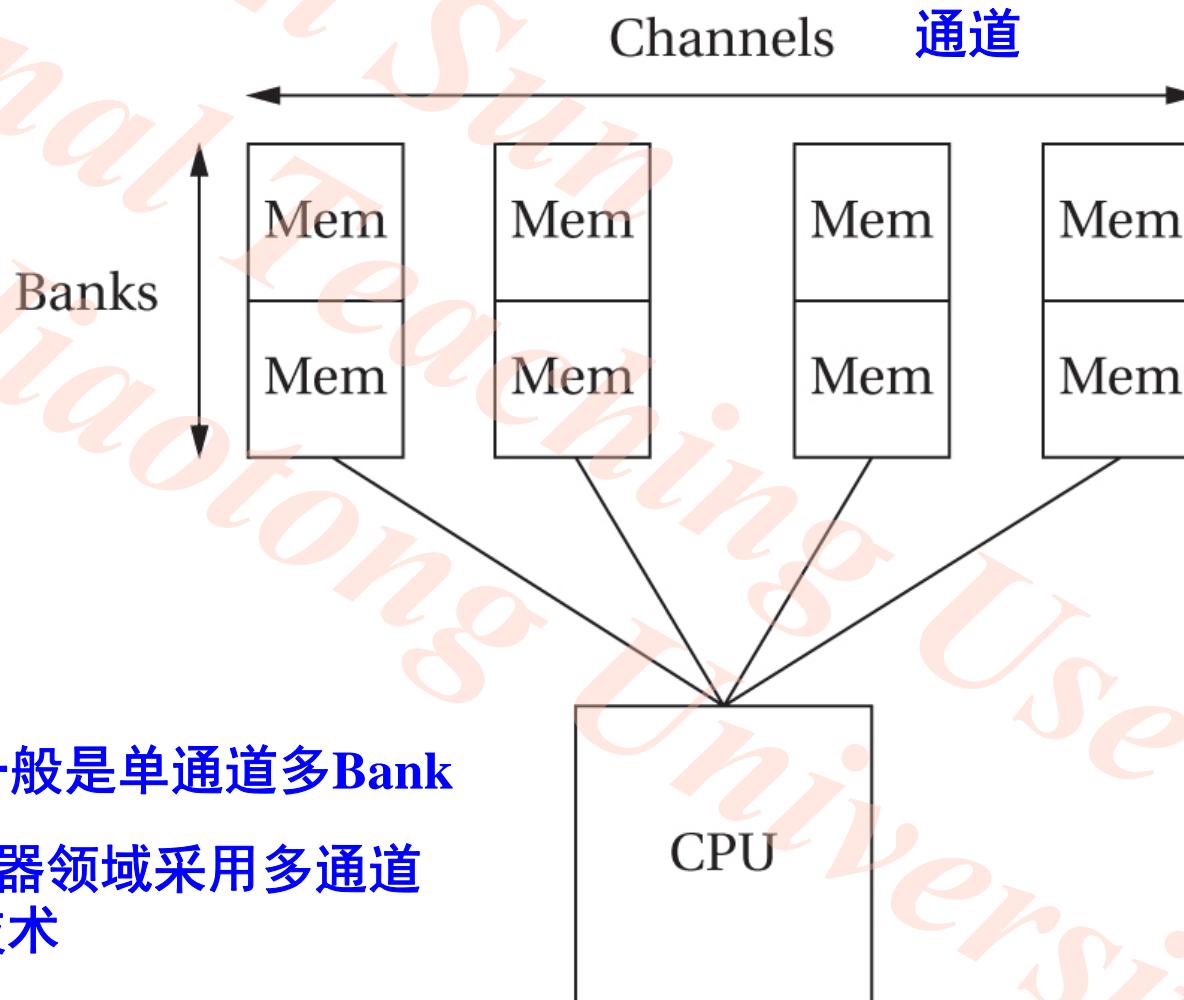
4 组件接口 - 存储器接口

- 存储器接口：
 - 多片存储器芯片串联组合成更大容量。
 - 多片存储器芯片并联组合成更大数据宽度。
 - **DRAM**芯片接口：需要逻辑电路将总线信号转变成合适的存储器信号。如行列地址和刷新。





多通道DRAM





设备接口

- 一些I/O设备接口被设计成直接连到特定的总线上，形成**无胶连接口**（**glueless interface**）。
- 当把设备连到不是专门为它设计的总线上时，**胶连逻辑**（**glue logic**）是必需的。



5 系统体系结构

- 体系结构指共同构成一个单元的一组元件及元件之间的关系。
- 嵌入式计算系统的体系结构是实现嵌入式计算系统的蓝图（**blueprint**），定义需要的元件和如何连接。
- 嵌入式计算系统的体系结构包括：
 - 硬件
 - 软件



硬件体系结构

- 硬件结构包括如下单元：
 - **CPU**: 多种不同体系结构，一种体系结构内也有不同的时钟速度、存储结构、集成的外围设备。。。。
 - **总线**: 总线的选择对系统的限制也许比**CPU**更多，必须保证总线能够提供足够的数据带宽。
 - 存储器：不是有没有，而是有什么。容量？**ROM**与**RAM**的比例？**SRAM or DRAM**？速度？
 - 输入/输出设备：网络？传感器？接口？使用某一设备的难度与耗费在最终选择中起着重要作用。



软件体系结构

- 良好的程序设计需要代表体系结构的结构设计。
- 软件体系结构设计的基本任务是**划分**—以利于实现、测试和修改的方式把功能分成块。
- 面条式代码（**spaghetti code**）：
 - 嵌入式软件通常将多个功能模块混合一起
 - 代码控制流结构差，并且过多使用全局变量和通常不可靠的程序。
- **好的选择**：将系统的功能按照操作的主要方式和设备的功能分成块。程序风格与性能要求。
- 一些软件功能可以在硬件或**I/O**设备中实现：更智能的**I/O**设备或**DMA**代替软件循环。





软硬件体系结构

- 嵌入式计算系统中，软件与硬件是密切相关的。
 - 缺少了硬件平台，软件就不可能执行
 - 软件需求决定了我们需要的硬件
 - 性能
 - 存储空间
- 所以，嵌入式计算系统在设计之初，就要同时考虑软件与硬件设计。
- 软硬件协同设计（**hardware and software co-design**）。



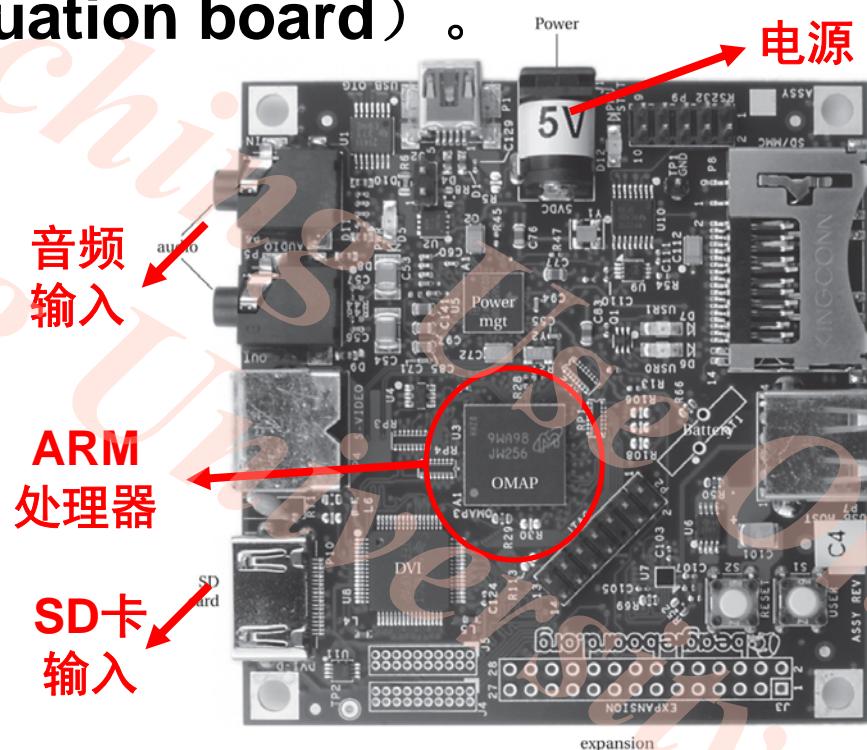
硬件设计

- 从完全现成的方案到高度定制的设计，硬件平台的设计复杂性可以有很大的不同。
- 在主板设计层次上，首先要考虑微处理器制造商和合作公司提供的评估板（**evaluation board**）。

典型的评估板包括一个CPU、一个存储器、一个下载程序的串行链接和少数的I/O设备。

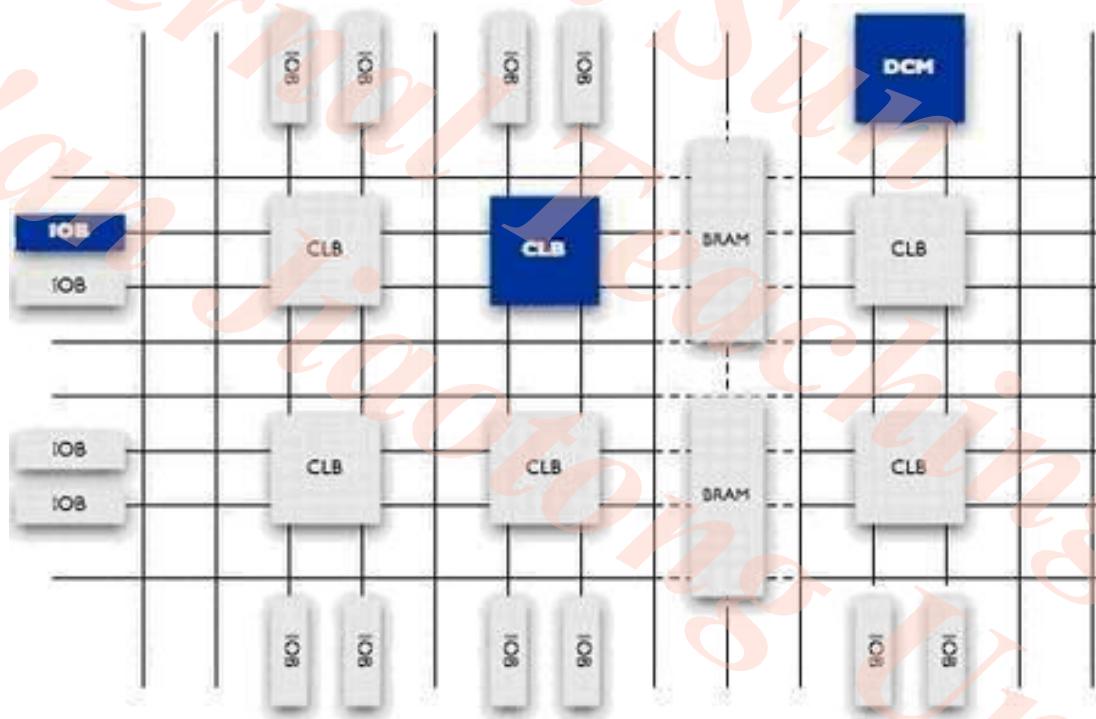
评估板可以是完整的解决方案，也可以仅通过少量的修改提供所需的解决方案。

存储器和I/O设备很重要，可以选择已有（数据手册 **datasheet**）也可以自己设计。





使用CPLD和FPGA



IOB: 可编程输入输出单元

CLB: 可配置逻辑单元

DCM: 数字时钟管理单元

BRAM: 嵌入式块RAM

丰富的布线资源

内嵌功能单元: **PLL、DSP**



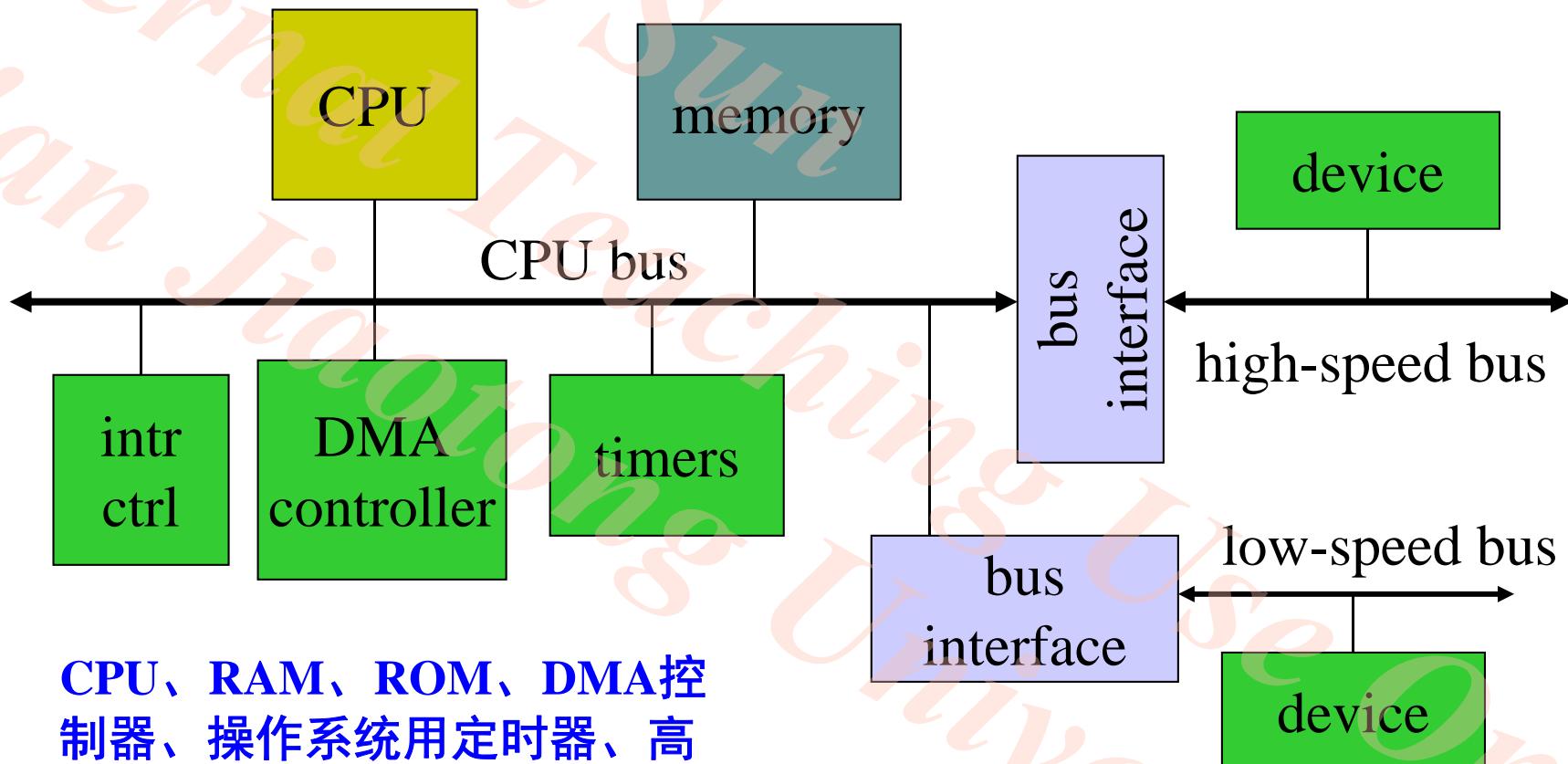


PC作为平台

- 个人计算机（PC）经常用作嵌入式计算的平台。
- 优点：
 - 预留I/O接口多，可扩展丰富的I/O设备。
 - 提供丰富的编程环境。
- 缺点：
 - 体积大
 - 耗电量大
 - 比定制硬件平台昂贵
- PC嵌入式系统通常用在对大小和耗电不敏感的场合。



典型的PC硬件系统





USB通用串行总线



版本	时间	最高传输速率	最大输出电流
USB1.0	96.1	1.5Mbps(192KB/s)	500mA
USB1.1	98.9	12Mbps(1.5MB/s)	500mA
USB2.0	00.4	480Mbps(60MB/s)	500mA
USB3.0	08.11	5Gbps(640MB/s)	900mA

- **USB** 是一种支持热插拔的高速串行传输总线，它使用差分信号来传输数据，支持“总线供电”和“自供电”两种供电模式。
- 一个采用**USB 3.0**的闪存驱动器可以在**15秒钟将1GB**的数据转移到一个主机，而**USB 2.0**则需要**43秒**。



PCI Express

- PCI Express与PCI等早期计算机总线相比最大的区别在于：它抛弃了之前的共享并行架构，采用点对点串行连接，每个设备都有自己的专用连接，不需要向整个总线请求带宽，且可以把数据传输率提高到很高的频率。

PCI-E采用多对高速串行的差分信号进行高速传输，每对差分线上的信号速率可以是1代的**2.5Gbps**、2代的**5Gbps**以及现在正逐渐开始应用的**3代8Gbps**。





PC软件平台

- PC还提供标准软件平台，该平台提供到基础硬件和更先进设备的接口。
- **BIOS**: 基本输入/输出系统，提供底层硬件驱动和引导程序（从其他设备如磁盘、网络等装载整个操作系统）。
- 操作系统：提供高层次驱动程序、执行过程控制、用户接口等等。
- 操作系统精简版本：**Windows vs. WinCE, Linux vs. Embedded Linux**。
- 即插即用（**plug and play**）：允许设备提供它们自己的配置信息。



单芯片系统：Intel StrongARM

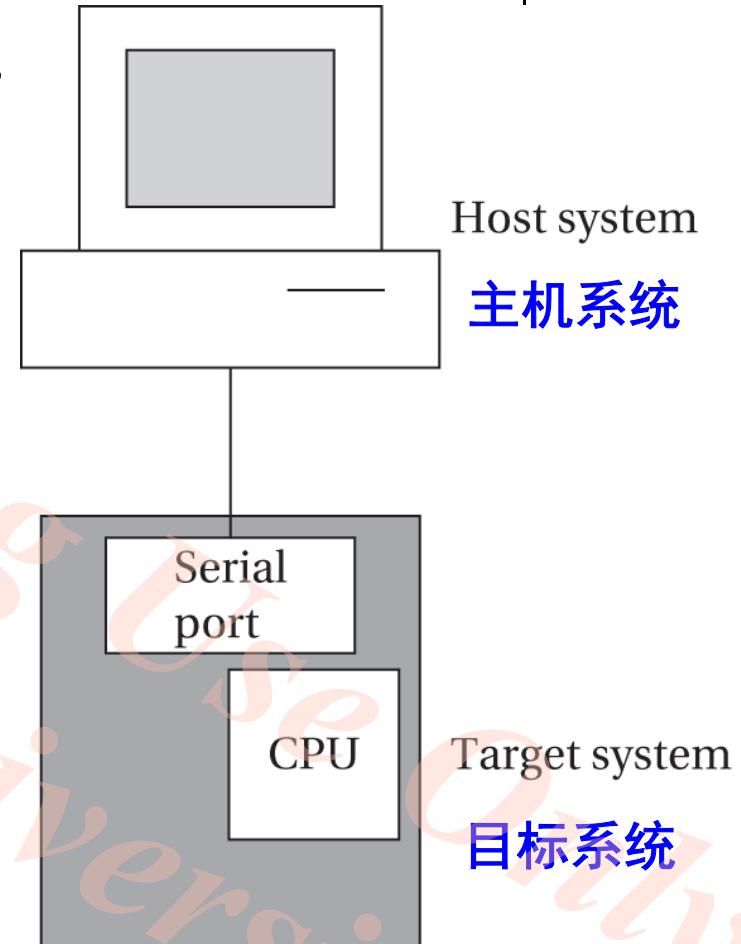
- 单芯片系统也称为片上系统（**System on Chip, SoC**）
- **Intel StrongARM SA-1100**芯片包括两个片上总线：一个高速系统总线和一个低速外围总线。芯片用两种不同的时钟：**3.686MHz**时钟来驱动**CPU**和高速外围设备，**32.768kHz**时钟是系统控制模块的一个输入。
- 系统控制模块包括：
 - 一个实时时钟**RTC**和一个操作系统定时器
 - **28个通用I/O**
 - 一个中断控制器
 - 一个电源管理控制器
 - 一个负责复位处理的重置控制器。





6 开发和调试 – 开发环境

- 典型的嵌入式计算系统各组件都很小，一般至少有一部分软件开发工作要在**主机**（PC或工作站）上进行。
- 最终运行代码的硬件称为**目标**。
- 主机与目标系统之间通常由串口、**USB**或以太网连接。
- 目标必须包括与主机系统通信的一小部分软件，尽量减少资源占用。
- 主机应能做以下工作：
 - 将程序装载到目标中
 - 在目标上启动或停止程序执行
 - 检测内存和**CPU**寄存器





交叉编译器（cross compiler）

- **交叉编译器**是可以在一种机器上运行却为另一种机器生成代码的编译器。
- 编译完后，可执行代码可以通过串行链路下载到嵌入式系统中，也可以烧写到**PROM**中。
- **主机-目标调试程序**：用于调试的基本程序分支由目标提供，更复杂的用户接口由主机产生。
- **测试台程序**（**testbench program**）：生成输入模拟输入设备，也可接收输出值并把它们与期望值比较。
- **集成开发环境**（**Integrated Development Environment, IDE**）：用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面工具。



调试技术 (debug)

- 交叉调试器 (**cross debugger**)：可以显示目标系统状态，能允许目标系统被控制。
- 目标系统中一般运行一个监控程序，提供基本的调试功能。监控程序一般占用内存很小。
- 用户程序必须编写谨慎而不破坏监控程序，但监控程序也要能具备一定的被用户程序破坏后的恢复能力。
- 其他重要的调试工具：
 - 串口：发送目标系统状态，用于开发调试和现场诊断。
 - 断点 (**breakpoint**)：用户指定地址，程序执行到该地址自动暂停，转入监控程序：可测试或修改**CPU**寄存器，之后程序继续执行。
 - **LED**灯：指示系统状态，特别是错误状态。



ARM断点

0x400 MUL r4,r6,r6

0x404 ADD r2,r2,r4

0x408 ADD r0,r0,#1

0x40c B loop

uninstrumented code

0x400 MUL r4,r6,r6

0x404 ADD r2,r2,r4

0x408 ADD r0,r0,#1

0x40c BL bkpoint

code with breakpoint



在断点位置，程序停止执行而转到调试工具或监控程序。

实现断点：用监控程序的子程序调用代替断点处的指令。

断点处理程序的行为：

- 保存寄存器内容
- 向用户显示**CPU**状态并接收命令
- 返回之前，恢复系统状态



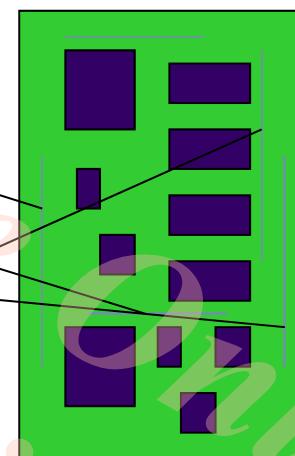
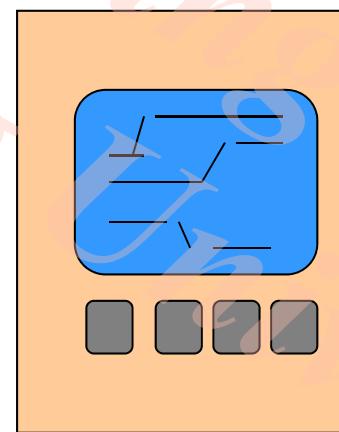
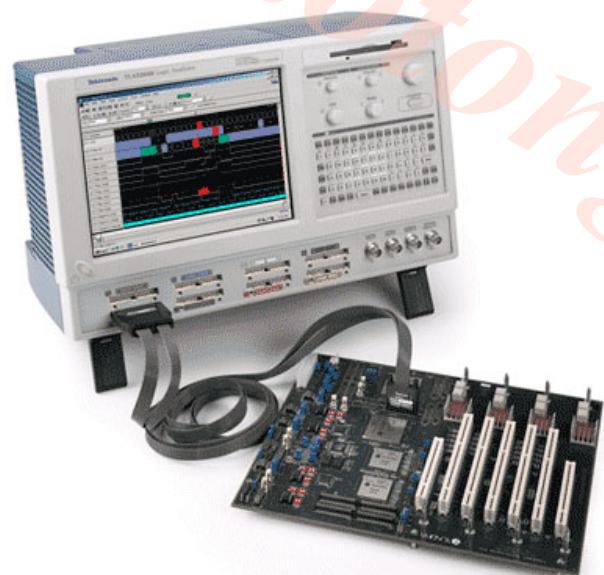
电路内部仿真器（ICE）

- 微处理器的电路内部仿真器（**in-circuit emulator**）是一种专用的硬件工具，能在嵌入式系统中帮助调试软件。
- 它是微处理器的一个特殊版本，当它停止时，其内部寄存器的内容可被读出，允许用户指定断点，检测和修改**CPU**状态。
- **ICE**提供的调试功能和监控程序中的调试器一样多，但不占用任何内存。
- **ICE**的不足：专门用于特定微处理器，当系统使用几个微处理器时，匹配成本会很大。



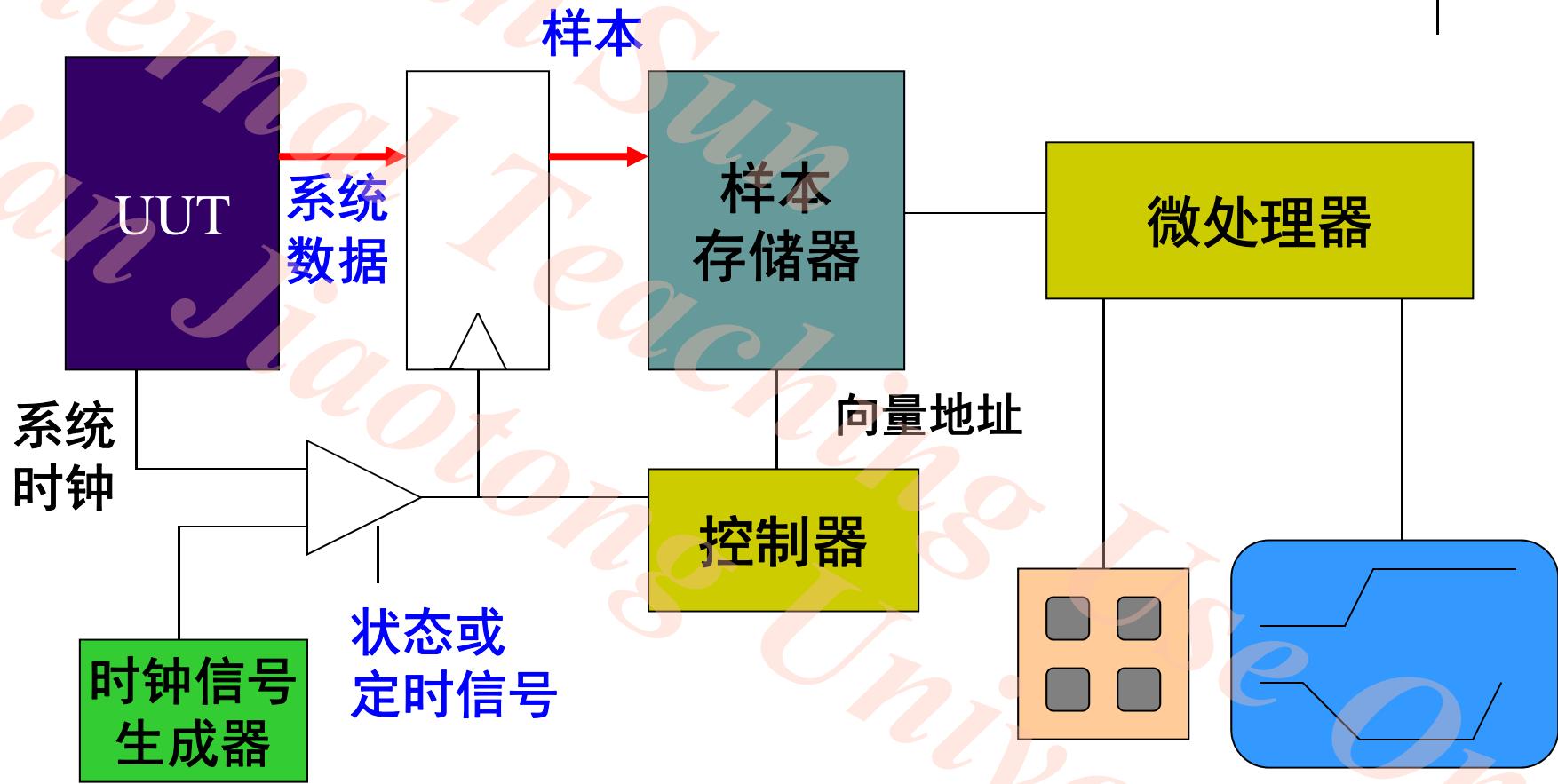
逻辑分析仪

- 逻辑分析仪（**Logic Analyzer**）是嵌入式系统设计者的主要工具之一，可以把逻辑分析仪看作是多通道简易示波器。
- 逻辑分析仪能同时采样几十到几百路信号，但只显示0或1
- 通过记录各路信道的信号活动，供显示分析，可提供比示波器更大的时间窗口。





逻辑分析仪内部结构



状态模式：使用系统自带时钟控制采样

定时模式：逻辑分析仪内部生成采样时钟

键盘

显示器

大量用于查看数据的格式

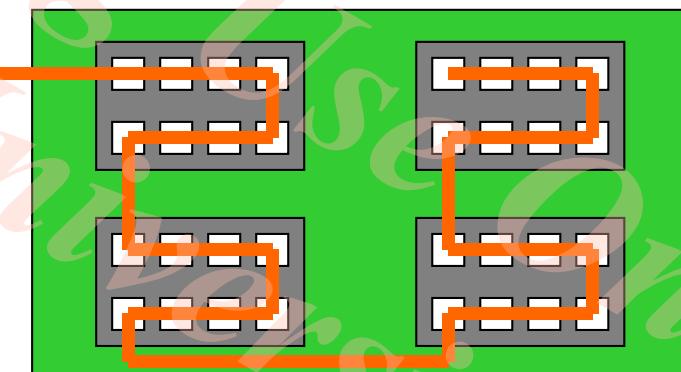


边界扫描 (boundary scan)

- 边界扫描是一种检查印刷电路板上的连线或集成电路中模组的方式。
- 联合测试行为组织 (**Joint Test Action Group**) **JTAG** 定义了这种测试方法。

边界扫描测试可以简化板级多片芯片的测试。

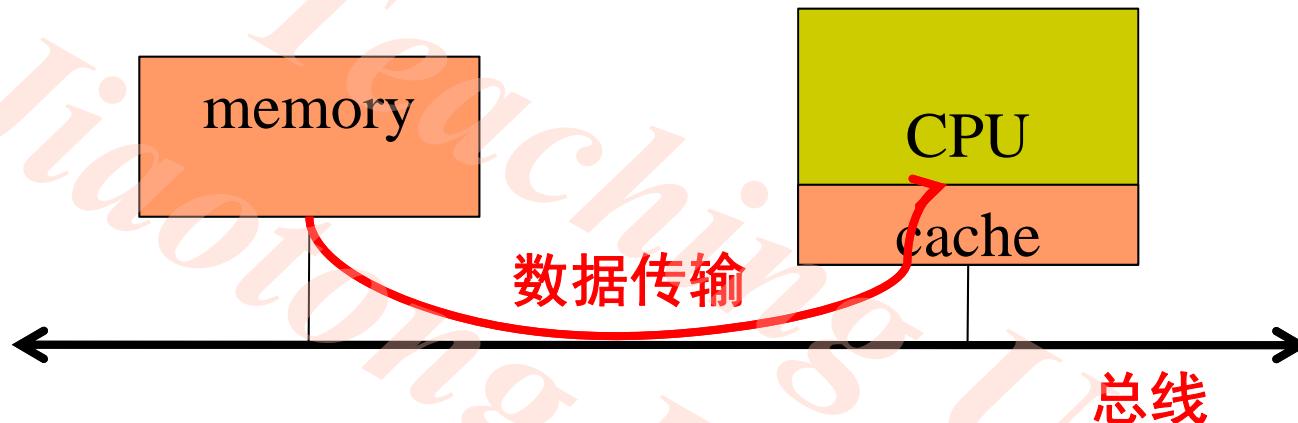
- 每个寄存器管脚可以被配置为扫描链的一个单元。
- 可在调试器或ICE中使用。





7 系统级性能分析

- 系统性能不仅包括**CPU**，系统任何一部分都将影响整体的系统性能。



- 从内存获取数据到**CPU**需要以下几步：
 - 从内存读
 - 通过总线传输到高速缓存
 - 从高速缓存传输到**CPU**

从高速缓存到**CPU**的传输时间包含在指令执行时间中，其他两段传输时间没有。



带宽 (bandwidth)

- 带宽指数据传输的速率，涉及到嵌入式计算系统中的多个组件：
 - 总线
 - 存储器
 - CPU获取
- 系统的不同部分采用不同的时钟频率
- 系统不同部分的接口数据宽度也不相同
- 必须经过时钟周期和宽度换算，为每一部分的性能评估提供正确的参考。



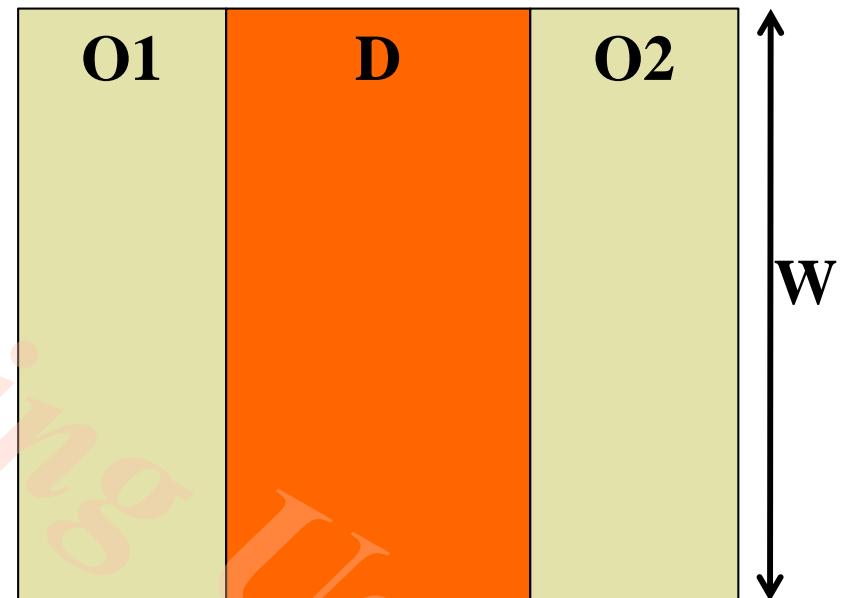
带宽与数据传输

- 以传输视频数据为例:
- 视频帧数据 Video frame:
 - $320 \times 240 \times 3 = 230,400$ bytes.
 - Transfer in $1/30$ sec == **0.033 sec per frame.**
- 假设传输速度 1 byte/usec (等价于1MHz)
 - 0.23 sec per frame.
 - Too slow.
- 两个方面可以提高带宽:
 - 提高总线的宽度: 每次传**4**字节, 传输时间降至**0.058s**。
 - 提高总线的时钟频率: **2MHz**, 传输时间降至**0.029S**。



总线带宽

- T: # bus cycles.
- P: time/bus cycle.
- Total time for transfer:
 - $t = TP$.
- D: data payload length.
- $O_1 + O_2 = \text{overhead } O$.

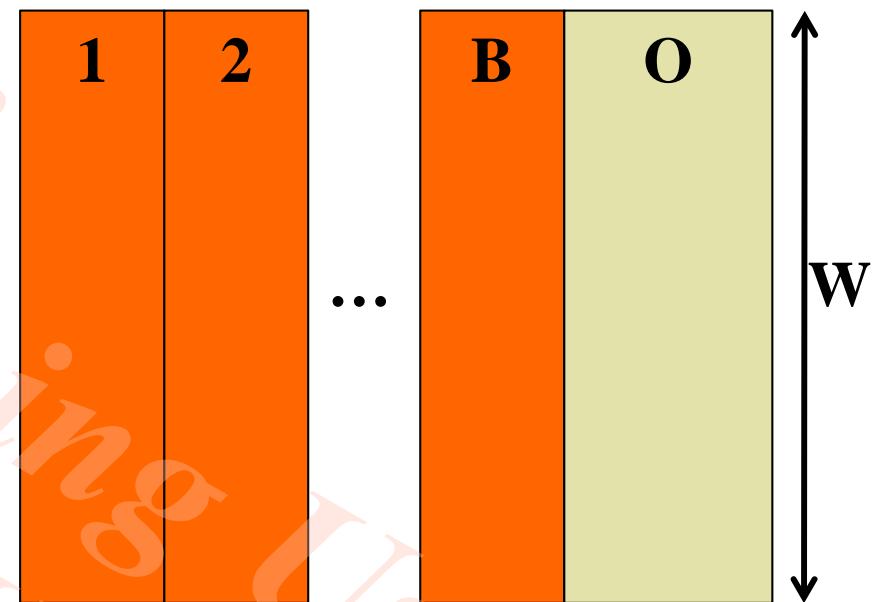


$$T_{\text{basic}}(N) = (D+O)N/W$$



突发总线带宽

- T: # bus cycles.
- P: time/bus cycle.
- Total time for transfer:
 - $t = TP$.
- D: data payload length.
- $O_1 + O_2 = \text{overhead } O$.

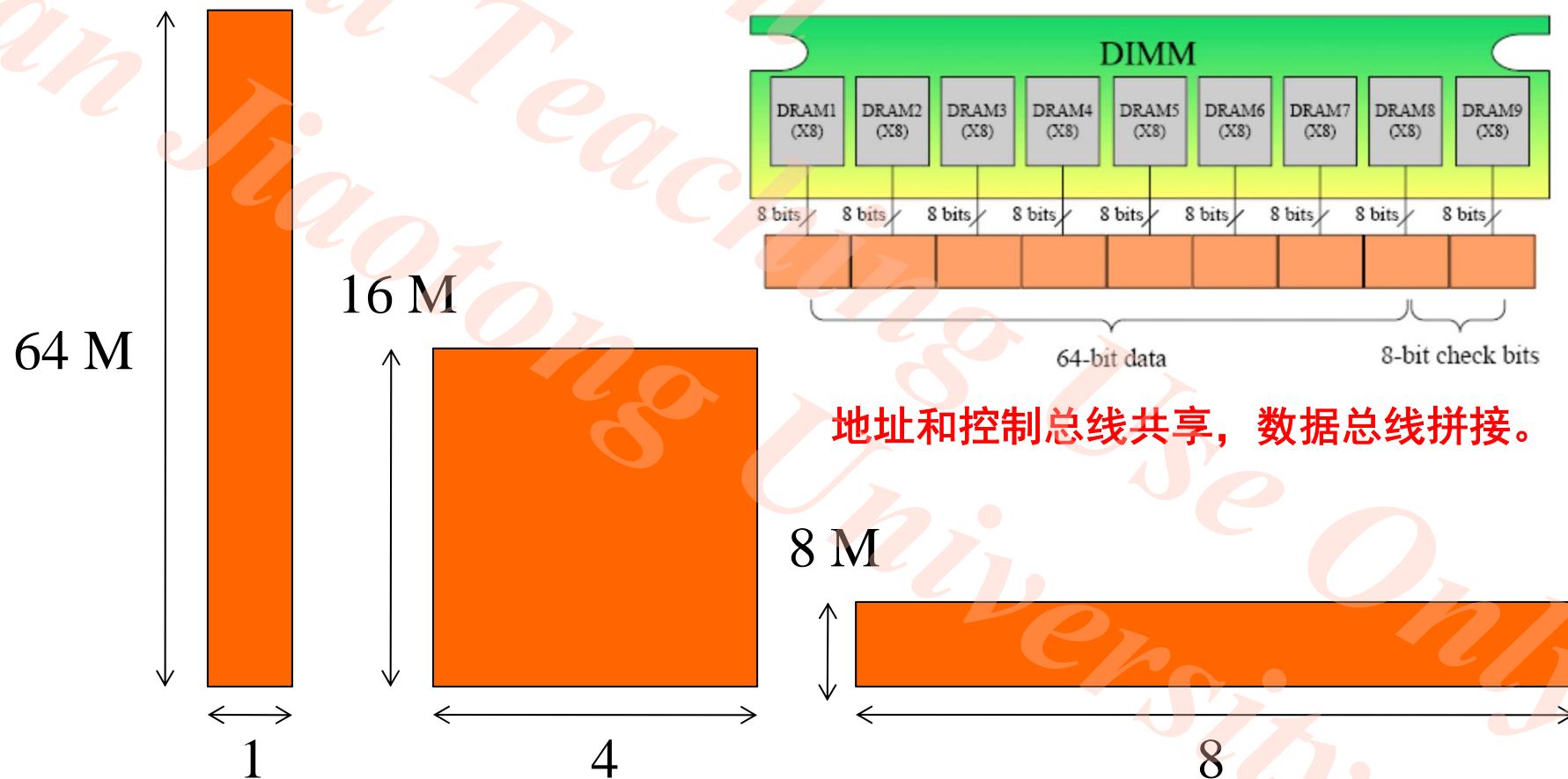


$$T_{\text{burst}}(N) = (BD + O)N / (BW)$$



存储器长宽比 (aspect ratio)

存储器宽度也是数据带宽的一种形式。存储器没有非常宽的长宽比，可以组合构建，选择**SIMM**或**DIMM**。

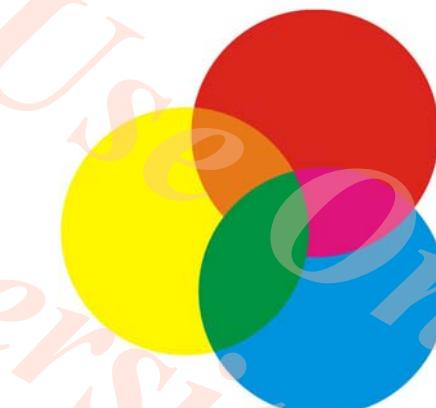




带宽分析

- 当数据类型无法自然对应到存储器宽度上时，情形会变得复杂。
- 每个标准像素是 $3 \times 8 = 24\text{bit}$ (红、绿、蓝)。
- 一个 24 位宽的存储器允许我们每次访问一个完整像素。
- 一个 32 位宽的存储器？ **Two choice**
 - 每次传输浪费 1 个字节
 - 不浪费，将像素拼接起来

$$\boxed{\Delta} A = [(E^* w) \bmod W] + 1$$





带宽瓶颈

- 假设以**30帧/秒**的速度在总线传输**320x240**的视频，带宽需求是**612000字节/秒**。
- 带宽计算：
- 总线：
 - 总线时钟频率**1MHz**（周期**10⁻⁶秒**），宽度**2字节**，D=1，O=3.
 - $T_{basic} = (1+3)612,000/2 = 1,224,000 \text{ cycles} = \textcolor{red}{1.224 \text{ sec.}}$
- 存储器：
 - 提供突发模式B=4，宽度仅为**4bit**（W=0.5），D=1，O=4，时钟频率**10MHz**。
 - $T_{mem} = (4*1+4)612,000/(4*0.5) = 2,448,000 \text{ cycles} = \textcolor{red}{0.2448 \text{ sec.}}$



带宽分析表

Bus	Memory		
Clock period	1.00E	- 06	Clock period
W	2	W	1.00E
D	1	D	- 08
O	3	O	0.5
N	612000	B	1
N	612000	N	4
T_{basic}	1224000	T_{mem}	4
t	1.22E + 00	t	2448000
			2.45E - 02



并行 (parallelism)

DMA在CPU
不使用总线
时为系统提
供并行。

