

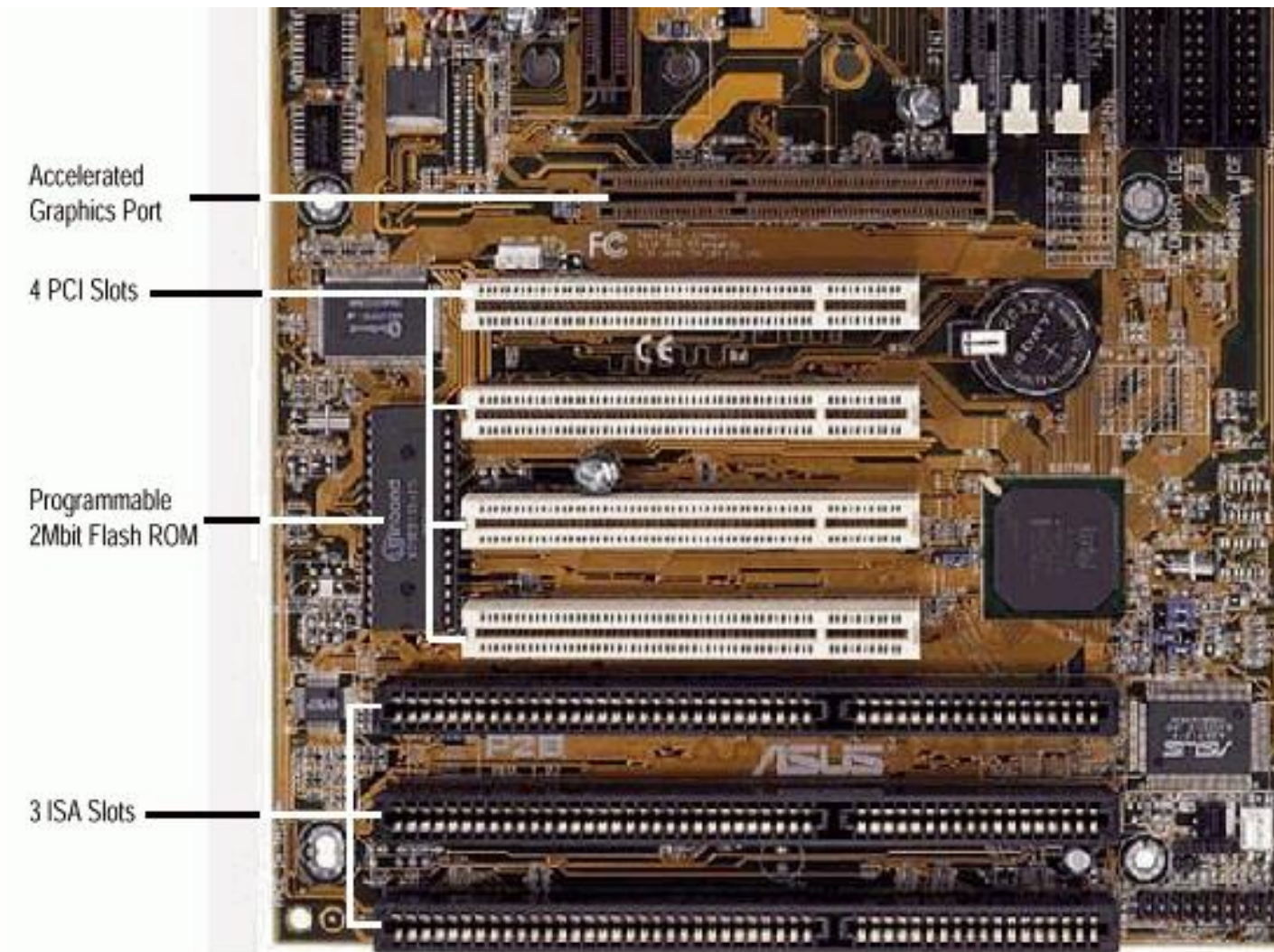


# 计算机组成原理

系统互连与通信——总线  
(RV \$6.9, \$6.10, 唐第3章)

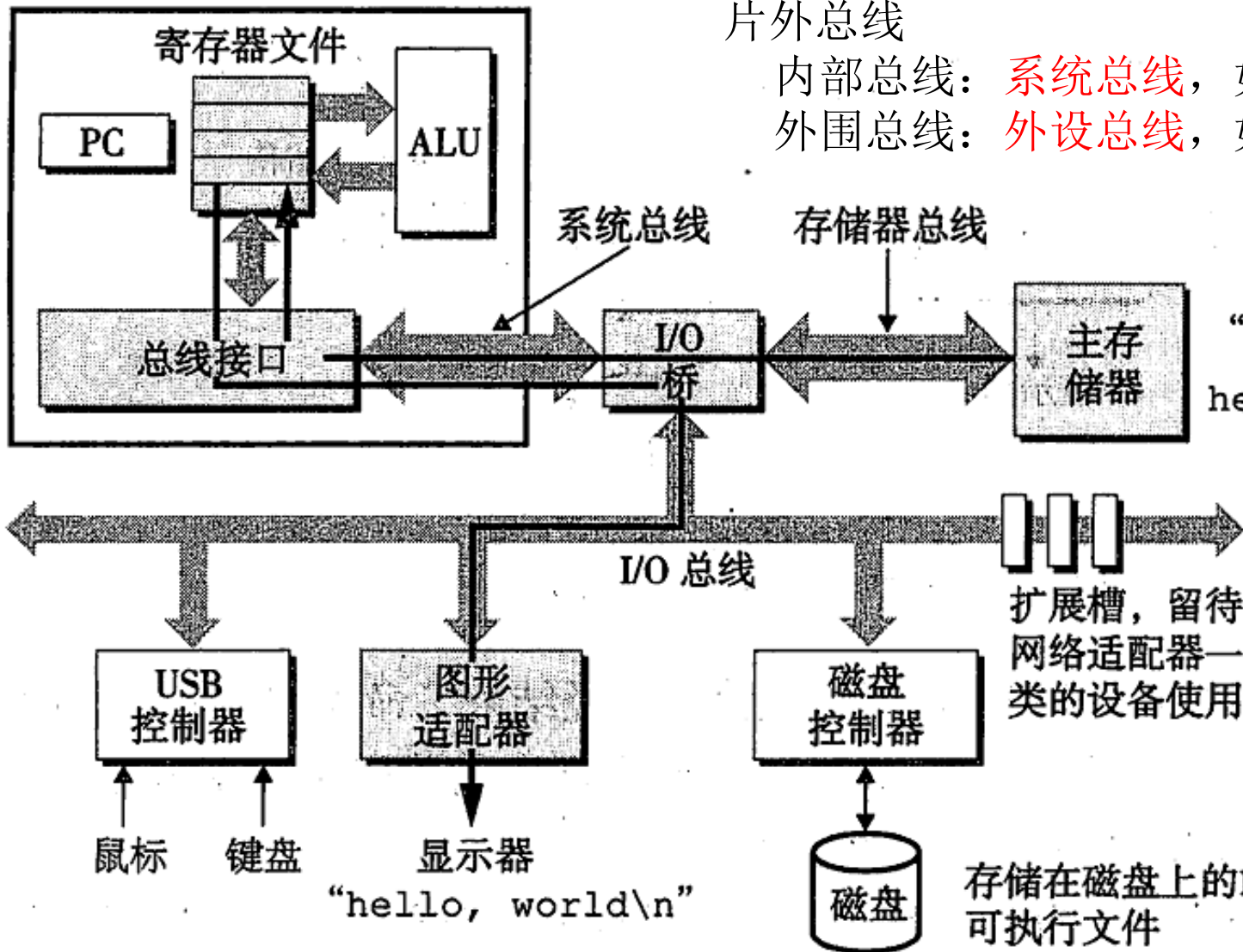
李曦 ([llxx@ustc.edu.cn](mailto:llxx@ustc.edu.cn))

# 系统中的互连总线：片上，板级，I/O



# "hello, world\n"

CPU



片上总线: SOC片内, AXI, CHI, TileLink  
片外总线

内部总线: **系统总线**, 如PCIe  
外围总线: **外设总线**, 如USB

"hello, world\n"  
hello 代码



存储在磁盘上的hello可执行文件



# 内容提要

- 互连结构：系统总线（板级互连），片上互连（CMP NoC）
- 唐第3章：系统总线
- RV \$6.9：多处理器网络（CMP）
- RV \$6.10：系统总线与I/O

## – 示例：NetFPGA 10G Ethernet NIC

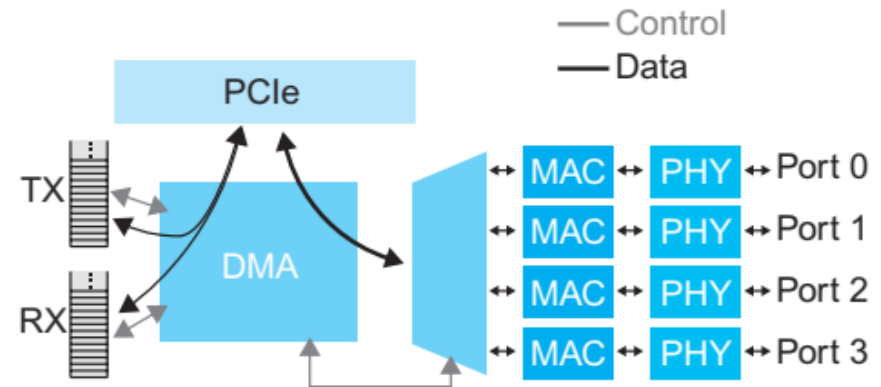
- an FPGA-based open platform for network research and classroom experimentation

## – 系统总线

- PCI：并行总线
- PCIe总线：串行总线

## – I/O 机制

- MMIO
- Polling, Interrupt-driven I/O, DMA
- 设备驱动程序

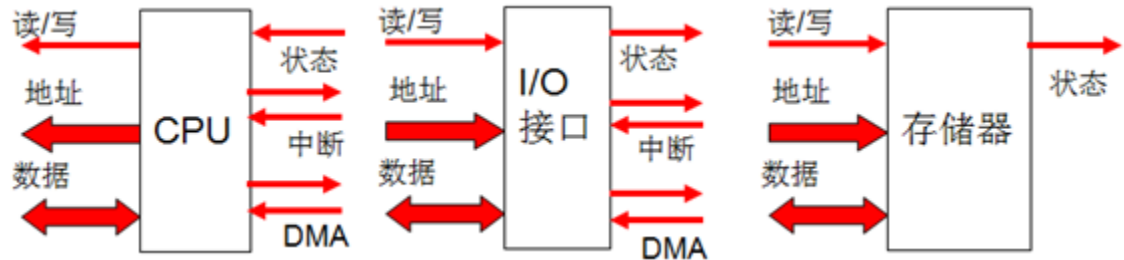


# 系统部件互连 (interconnect)



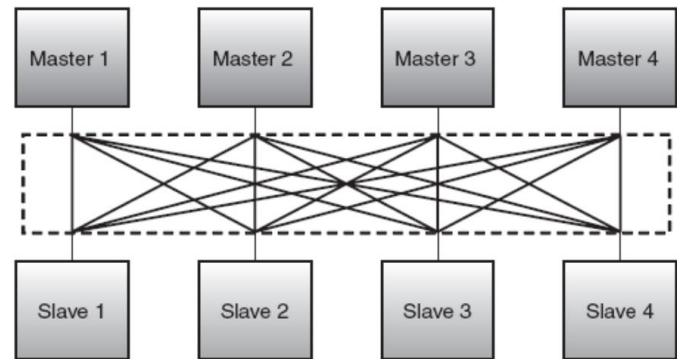
- 信号线wire

- 地址
- 数据
- 控制：状态，命令，同步



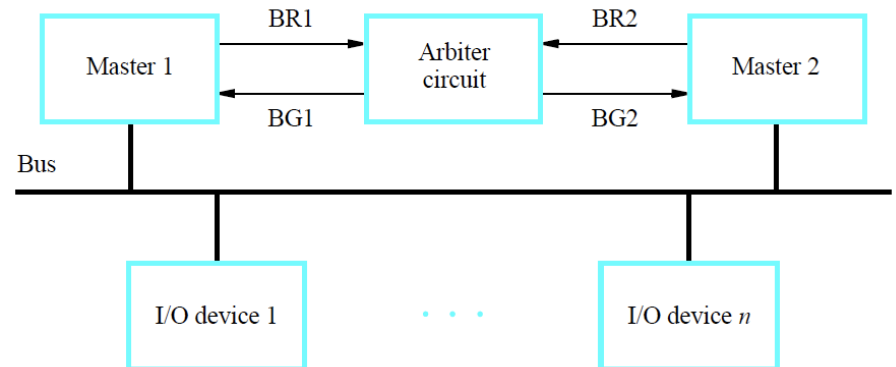
- 互连模式：物理，逻辑

- 点对点：独占式，一对一
- 总线：共享式，多对多



- 某一时刻只能有一个讲者（主设备）
- 主设备仲裁
  - 仲裁器：请求、允许

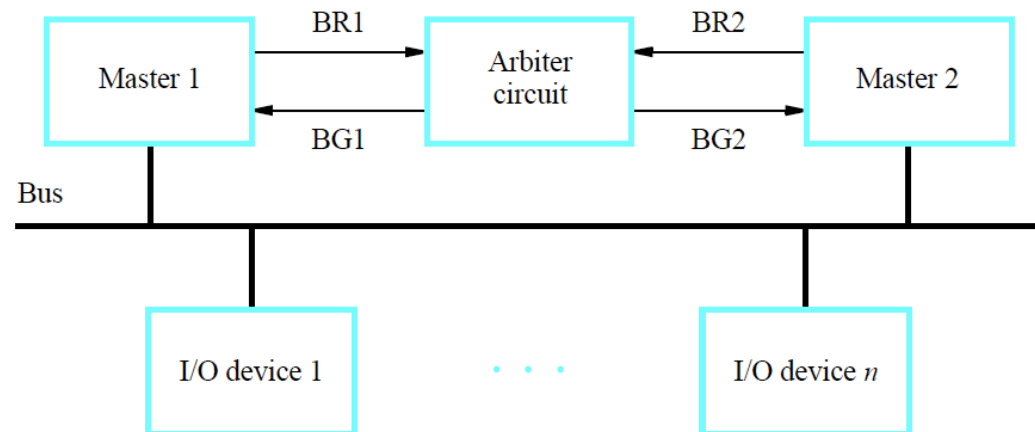
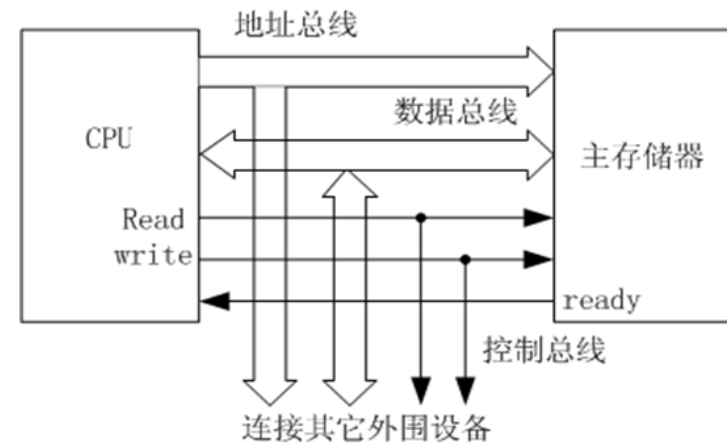
- 混合式



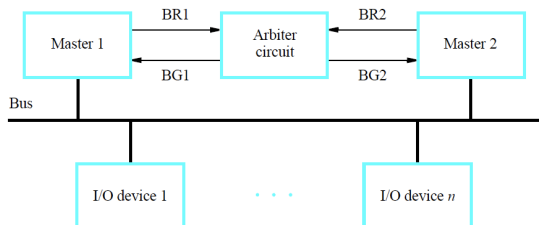
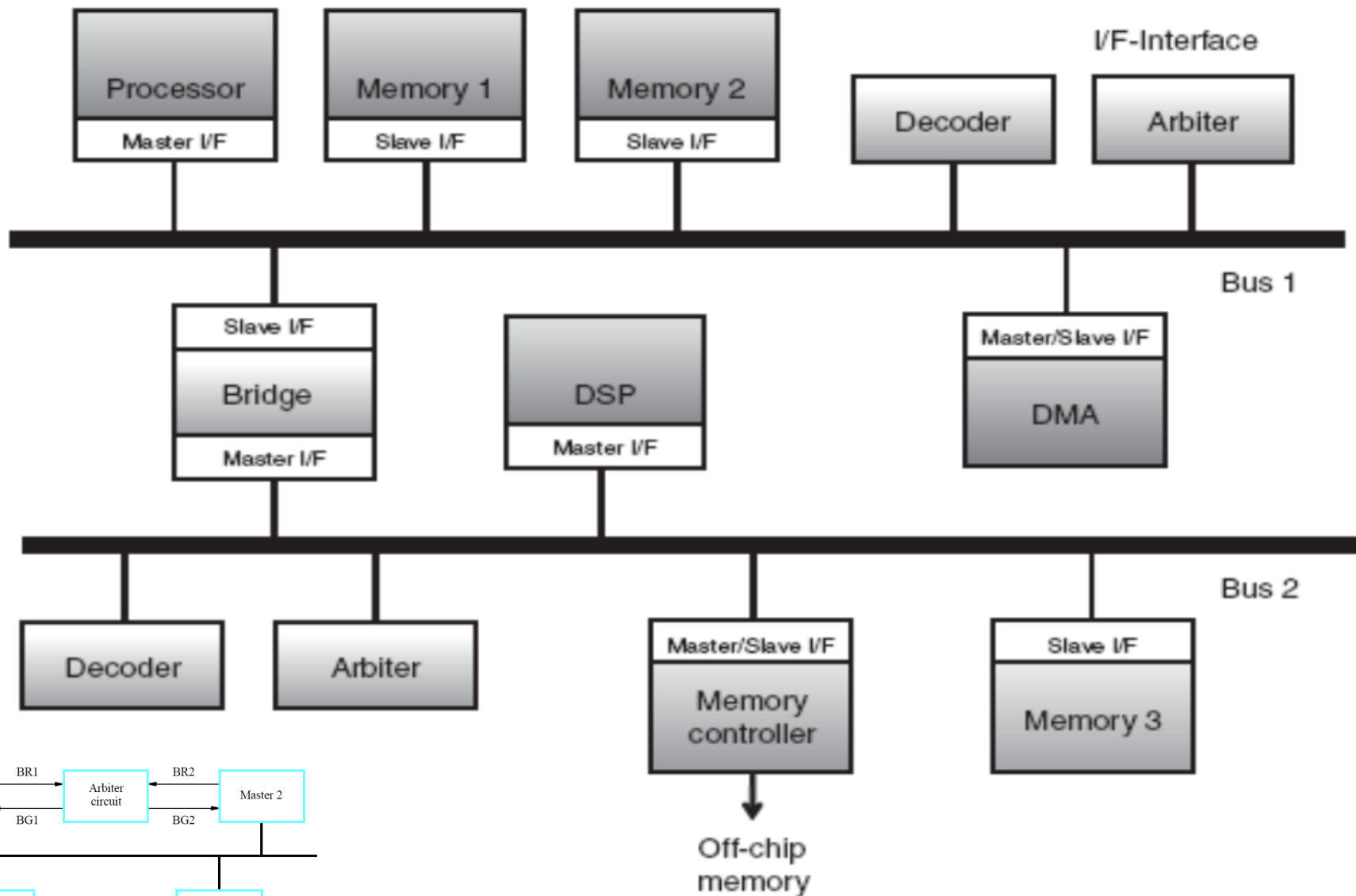
# 总线



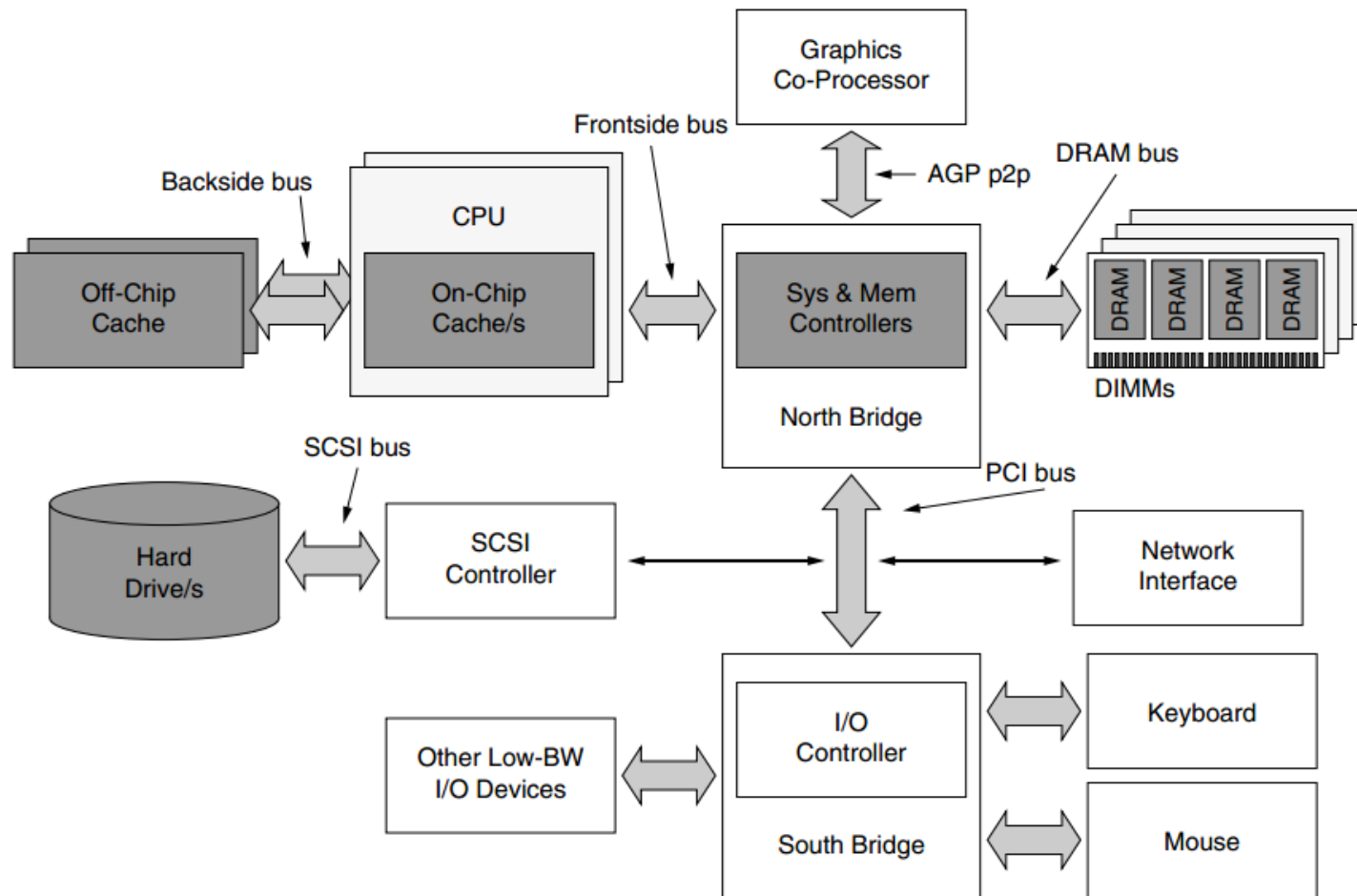
- 定义：连接多个部件的**传输线**
  - 数据、地址、控制（状态，命令，同步）
- 总线的关键特征：多设备**共享**的传输线。
  - 广播式
    - 主设备master：任意时刻**只能有一个**设备向总线**发送**信息
    - 从设备slave：**多个**部件可以同时从总线**接受**相同的信息
  - 系统瓶颈
- 总线拓扑：单总线，多总线（层次化）
- 总线周期：
  - 数据传输过程
  - **同步**方式
- 总线**仲裁**
  - 单/多master系统
  - 集中式，分布式（CAN）
- 总线标准



# 总线拓扑结构：单总线，多总线



# PC机中的总线：层次化拓扑结构

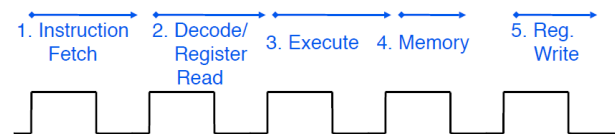


**FIGURE 0v.3:** Typical PC organization. The memory subsystem is one part of a relatively complex whole. This figure illustrates a two-way multiprocessor, with each processor having its own dedicated off-chip cache. The parts most relevant to this text are shaded in grey: the CPU and its cache system, the system and memory controllers, the DIMMs and their component DRAMs, and the hard drive/s.

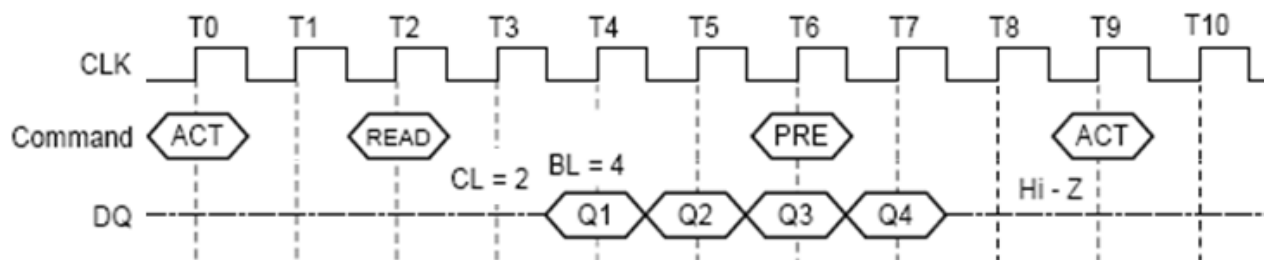
# 总线传输过程：总线周期



## 完成一次总线传输可分为3个阶段



- 申请阶段：仲裁算法（集中式、分布式）
    - 主设备提出申请，经总线仲裁机构决定将下一传输周期的总线使用权授予某一申请者。
    - 链式查询、计数器定时查询、独立请求
  - 数据传输：同步控制，传输模式（串/并、单字/突发）
    - 寻址阶段：取得了使用权的主设备，通过总线发出本次打算访问的从设备的**存储地址**或**设备地址**及有关**命令**，启动从设备。
    - 传数阶段：主设备和从设备进行数据交换。
  - 结束阶段：从总线上撤除有关信息，释放总线，让出使用权。
- ## 对于**独占系统**，只有数据传输阶段
- 独占系统：只有一个主设备的简单系统，如8086



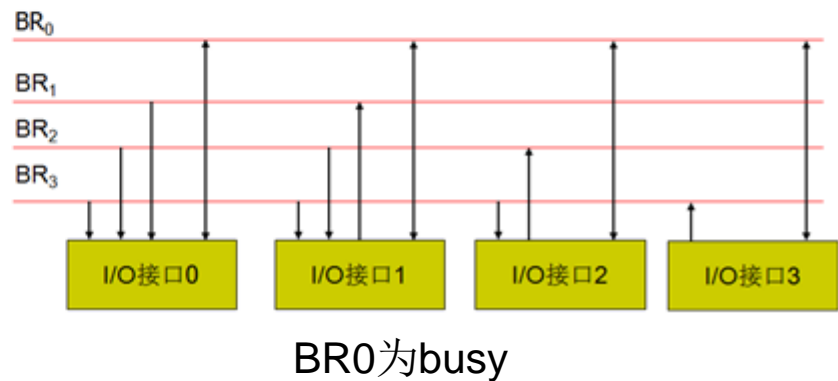
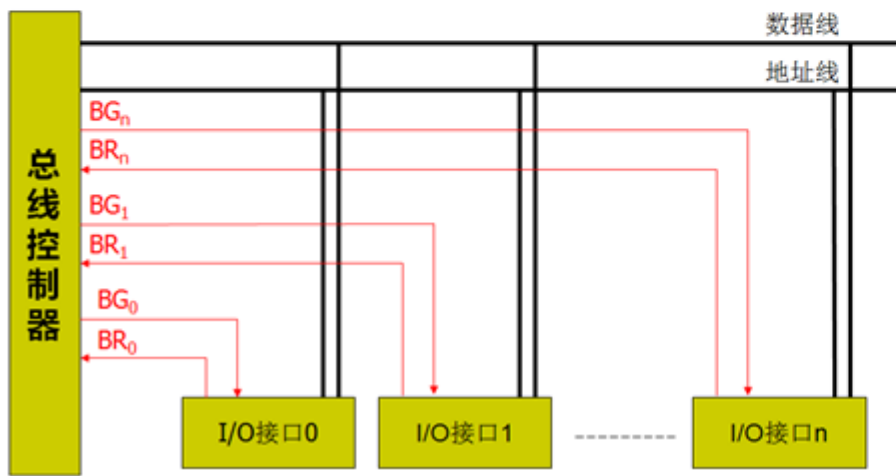
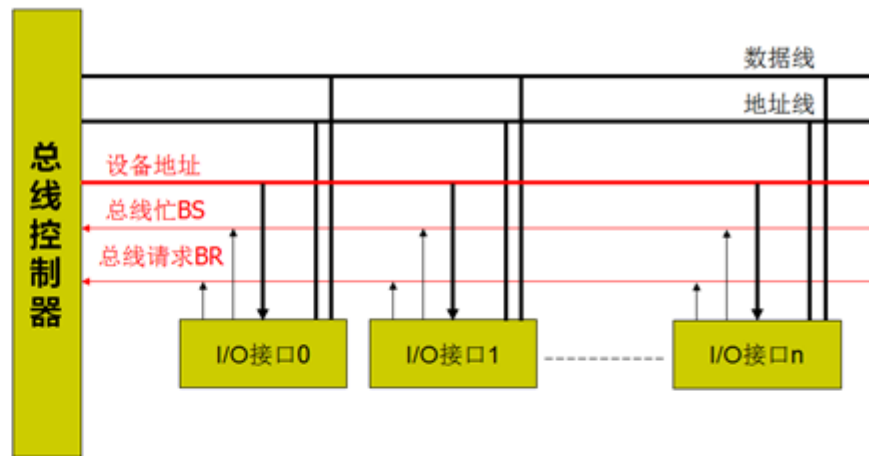
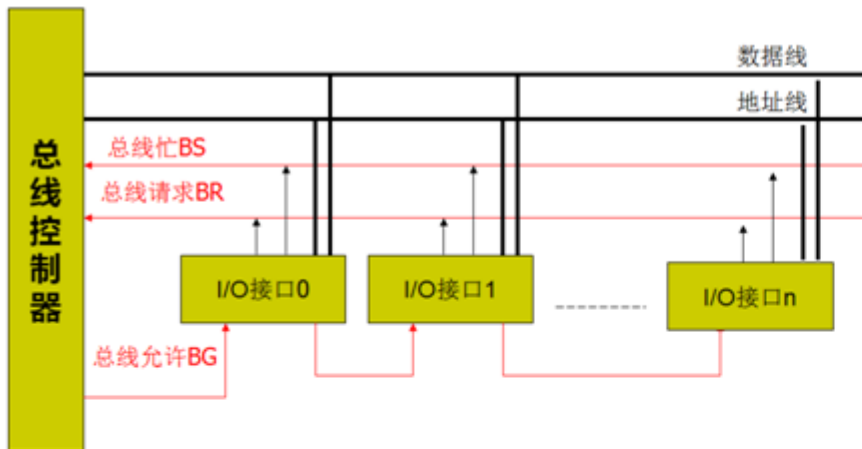


# 总线判优：复杂性、公平性、可靠性、可扩展性

集中式系统：链式查询、计数器定时查询、独立请求 【算法：RR，优先级，TDM】

分布式系统：自举式

唐2：图3.15



BR<sub>0</sub>为busy

# 总线通信同步控制：同步通信方式

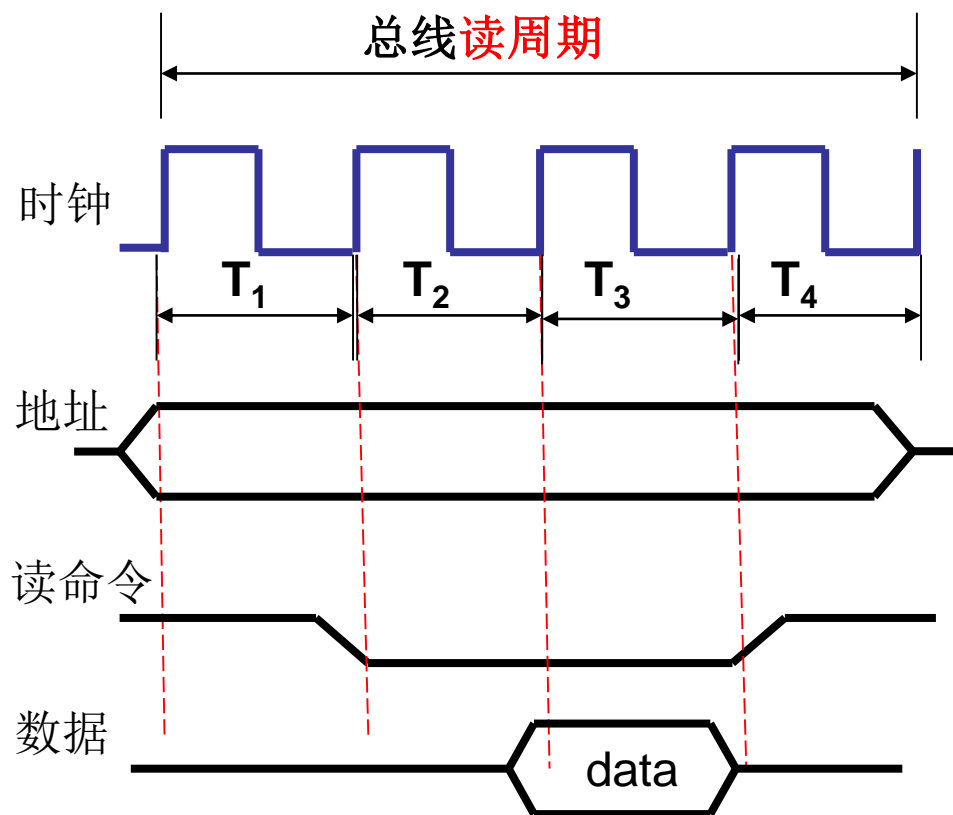


- 通信双方由**统一时标**控制数据传送称为**同步通信**。
  - 集中式：时标通常由**CPU**的总线控制部件发出，送到总线上的所有部件；
  - 分布式：由每个**部件**各自的时序发生器发出，但是必须有总线控制部件发出的时钟信号对它们进行**同步**。

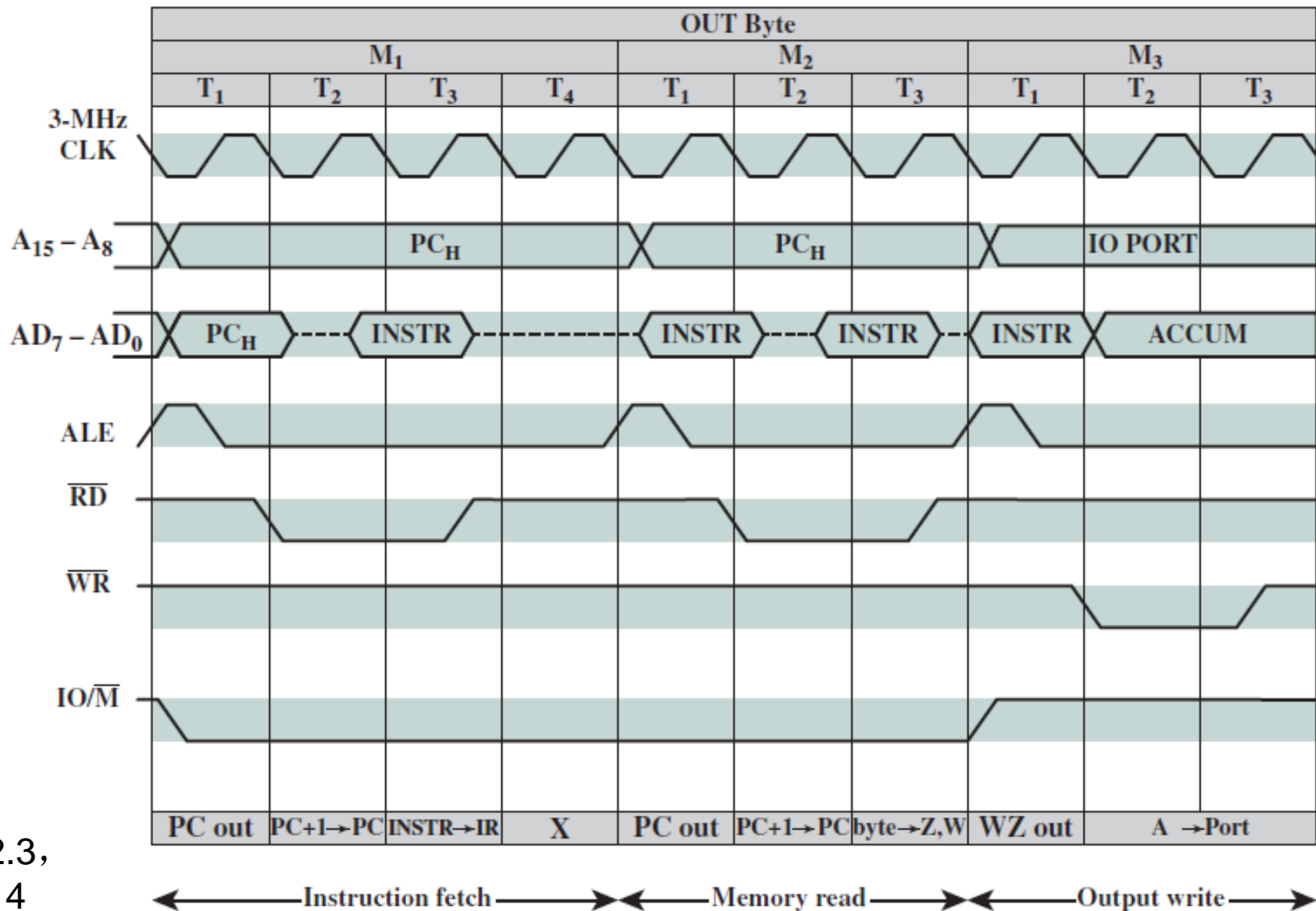
- 对于**读命令**，传输周期为：

- $T_1$ ：主模块发地址
- $T_2$ ：主模块发读命令
- $T_3$ ：从模块提供数据
- $T_4$ ：主模块撤销读命令

- 仲裁？
- 写操作时序图？



# OUT指令执行：指令周期、机器周期(总线/访存)、



唐9.2.3,  
图9.14

# 总线通信同步控制：异步通信方式



• 对总线操作的开始和结束：确认**握手**

– **没有公共的时钟**，采用应答方式

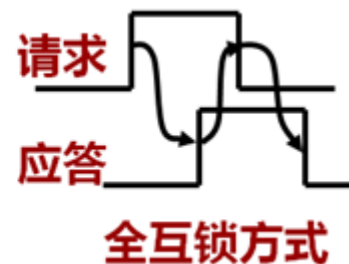
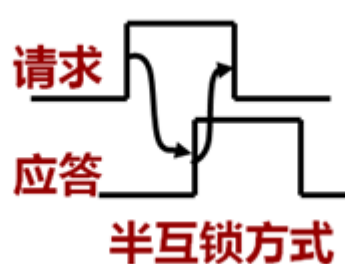
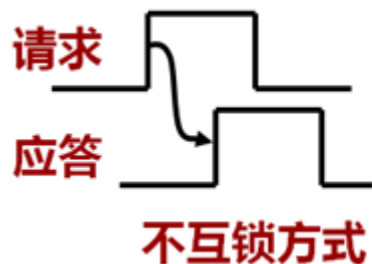
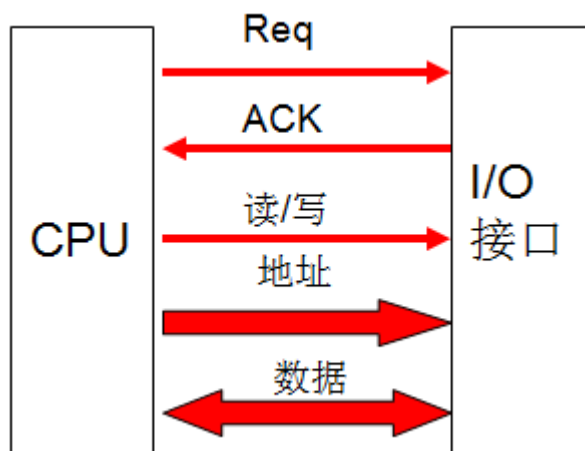
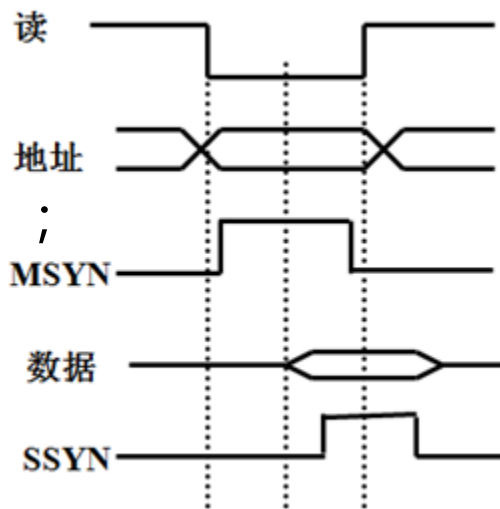
• 主设备Req (MSYN)，从设备Ack (SSYN)；

– 三种应答：不互锁，半互锁，全互锁

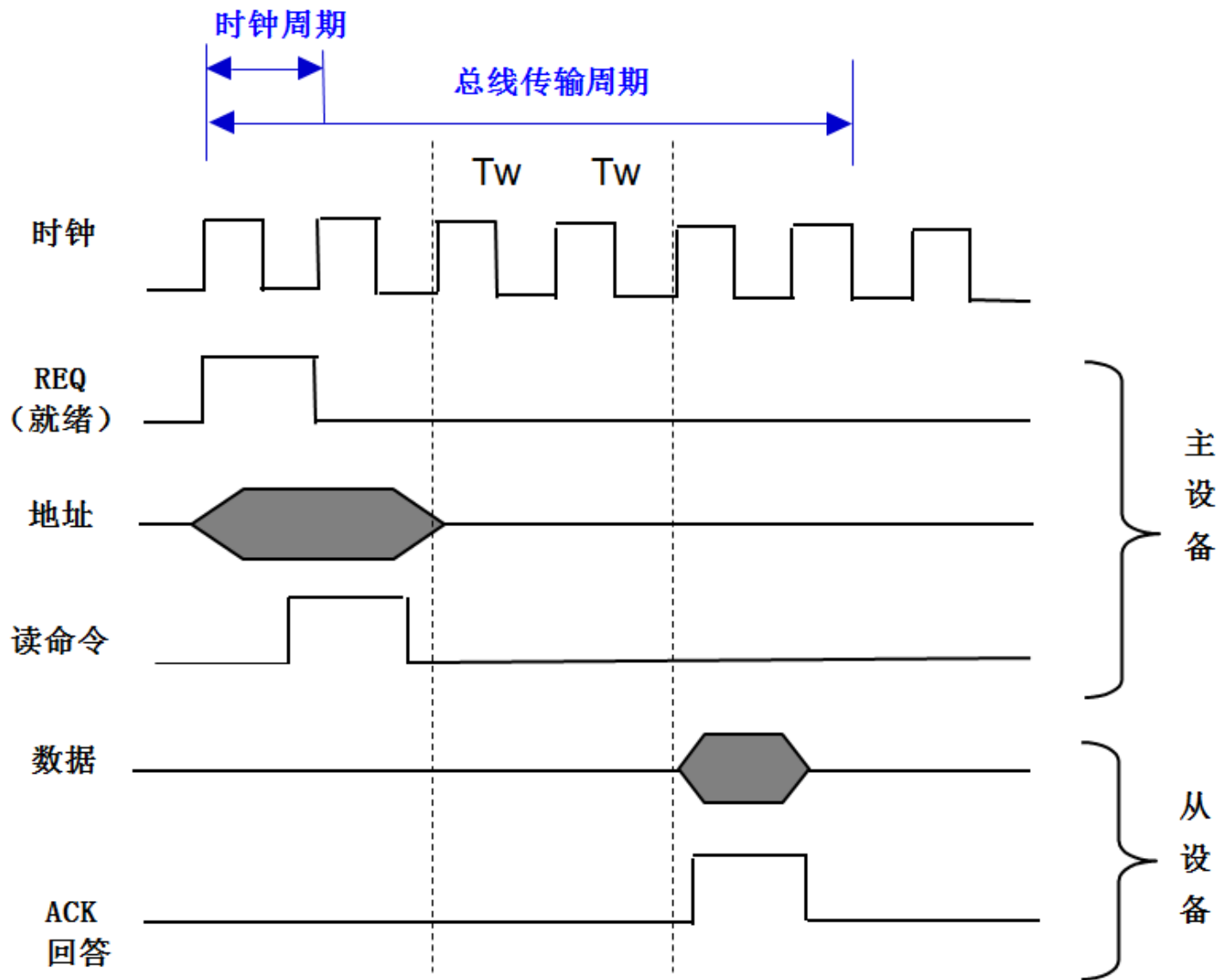
• 不互锁：请求->自动撤销，应答->自动撤销

• 半互锁：请求->等待应答，应答->自动撤销

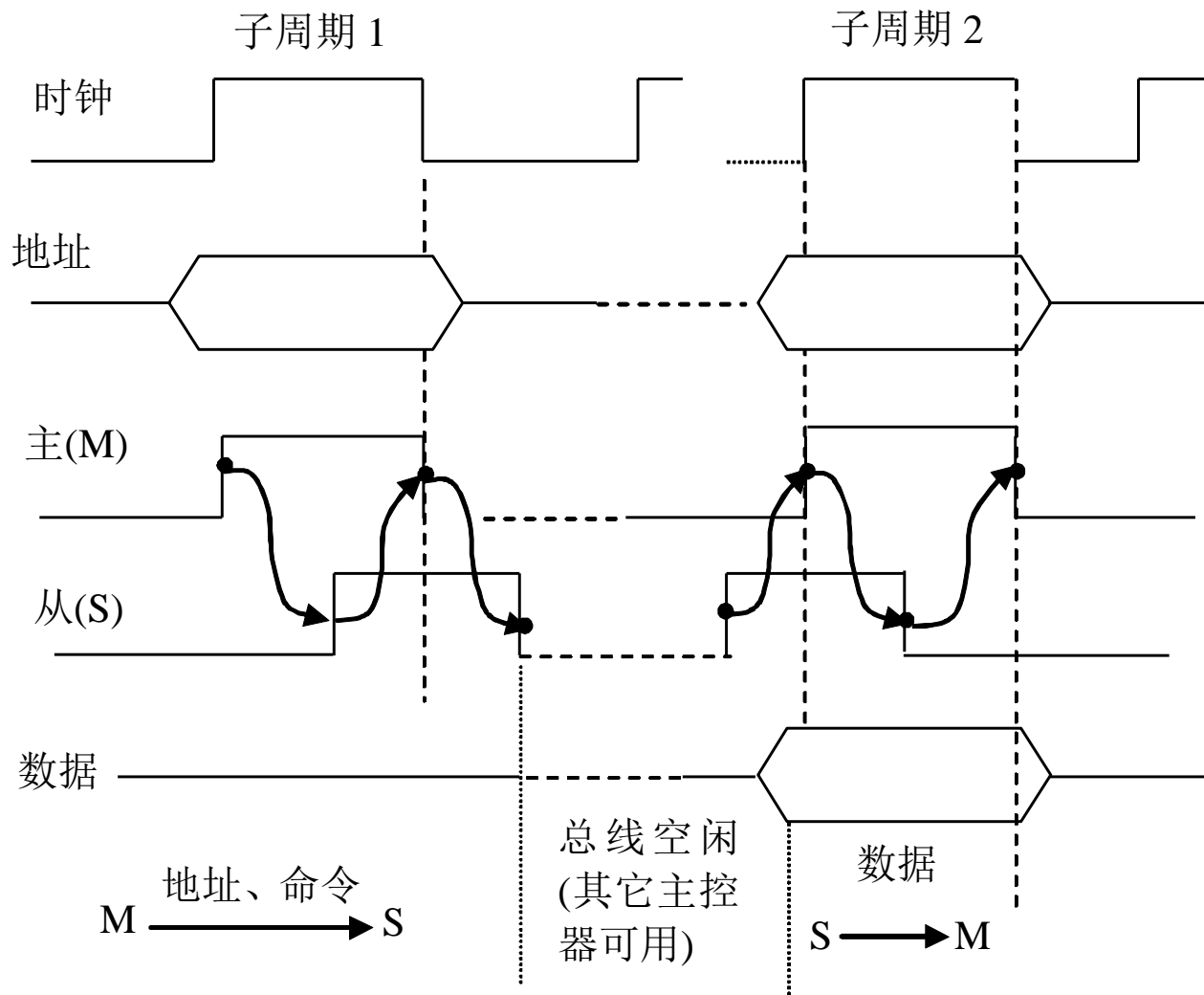
– 例：PCI半互锁



# 半同步通信协议（不互锁？）



# 分离事务通信协议：总线利用率

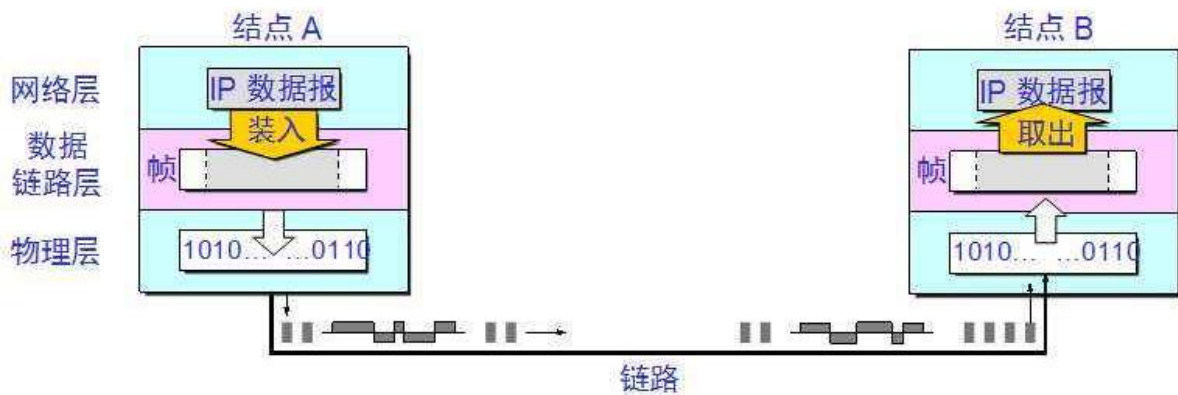
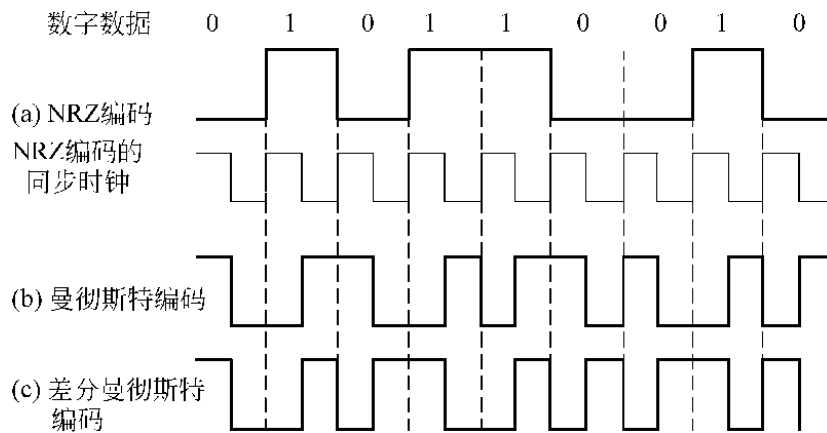


总线事务 (**Bus transaction**)：在一个总线周期中发生的一系列活动。  
典型：请求操作、裁决操作、地址传输、数据传输、总线释放



# 总线数据传输方式(物理层)

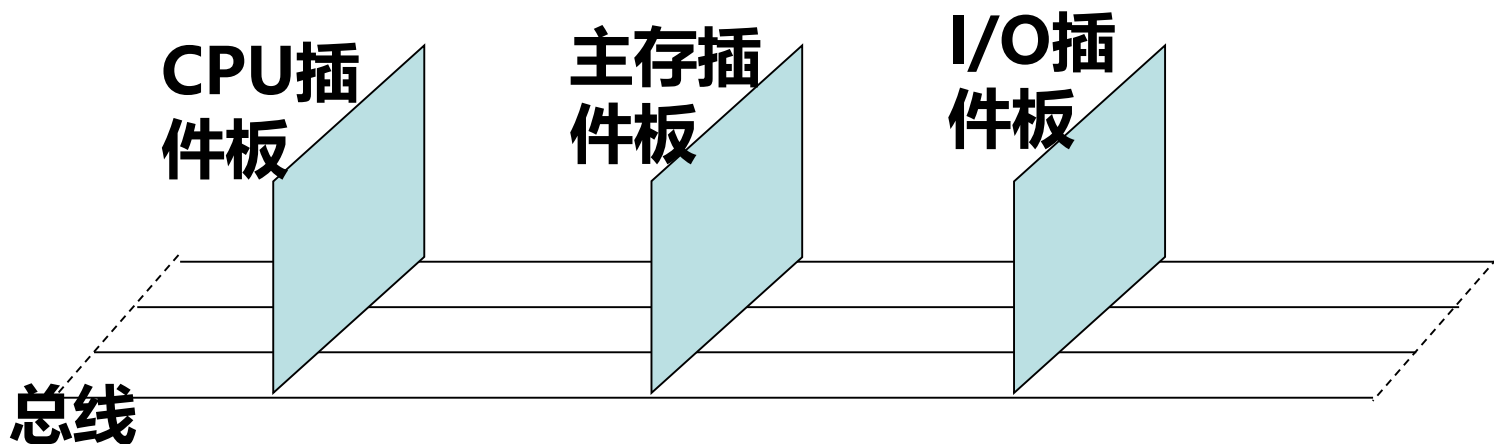
- 总线传输方式
  - 串行总线、并行总线
    - 传输一字节使用的**信道数**
  - 串行、并行传输
    - 串并转换问题
- 数据编码方式
- 数据传输模式
  - 单字节传输
  - 突发传输





# 总线特性：标准

- 从物理角度来看，总线就是一组电导线。



- 为了保证机械上的可靠连接，必须规定其**机械特性**；
- 为了保证电气上正确连接，必须规定其**电气特性**；
- 为了保证正确地连接不同部件，还需规定其**功能特性和时间特性**。
  - 数据、地址、控制



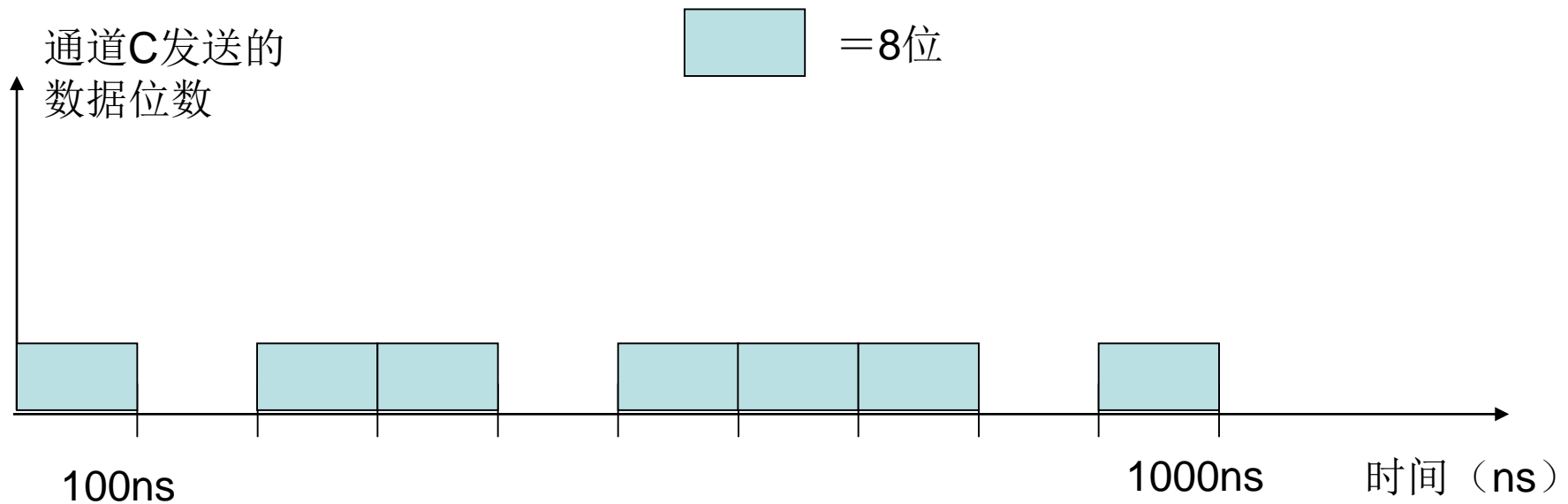
# 总线性能指标

- **信号线数**：即地址总线、数据总线和控制总线三种总线的根数总和。
- **总线宽度**：是指数据总线的根数，用bit（位）表示，如8位、16位、32位、64位（也即8根、16根、32根、64根数据线）。
- **最大传输率（总线带宽）**：总线本身所能达到的最高传输速率，用MB/s（每秒多少兆字节）表示。
  - 例：总线工作频率33.3MHz，总线宽度32位，则最大传输率 =  $33.3 \times 32 / 8 = 132 \text{ MB/s}$ 。
- **时钟同步/异步**：总线上的数据与时钟同步工作的总线称为同步总线，与时钟不同步工作的总线称为异步总线。
- **负载能力**：通常用可连接扩增电路板数来反映总线的负载能力。



# 通信速度

- 平均速率 =  $56\text{bits}/1000\text{ns} = 56\text{Mb/s}$ 
  - avgrate
- 最大速率 =  $8\text{bits}/100\text{ns} = 80\text{Mb/s}$ 
  - peakrate: 包长/发送时间





# 例1

- **题目：**某总线在一个总线周期中并行传送4个字节的数据，假设一个**总线周期**等于一个**总线时钟周期**，总线时钟频率为33MHz，则**总线带宽**是多少？  
~~如果一个总线周期中并行传送64位数据，总线时~~

**提示：**此题主要是考查对总线带宽的理解。

$$\begin{aligned}\text{总线带宽} &= \text{一次传输的字节数} / \text{总线周期} \\ &= (\text{总线宽度}/8) * \text{总线时钟频率}\end{aligned}$$

- **解：**设总线带宽用 $D_r$ 表示，总线时钟周期用 $T=1/f$ 表示，一个总线周期传送的数据量用 $D$ 表示，

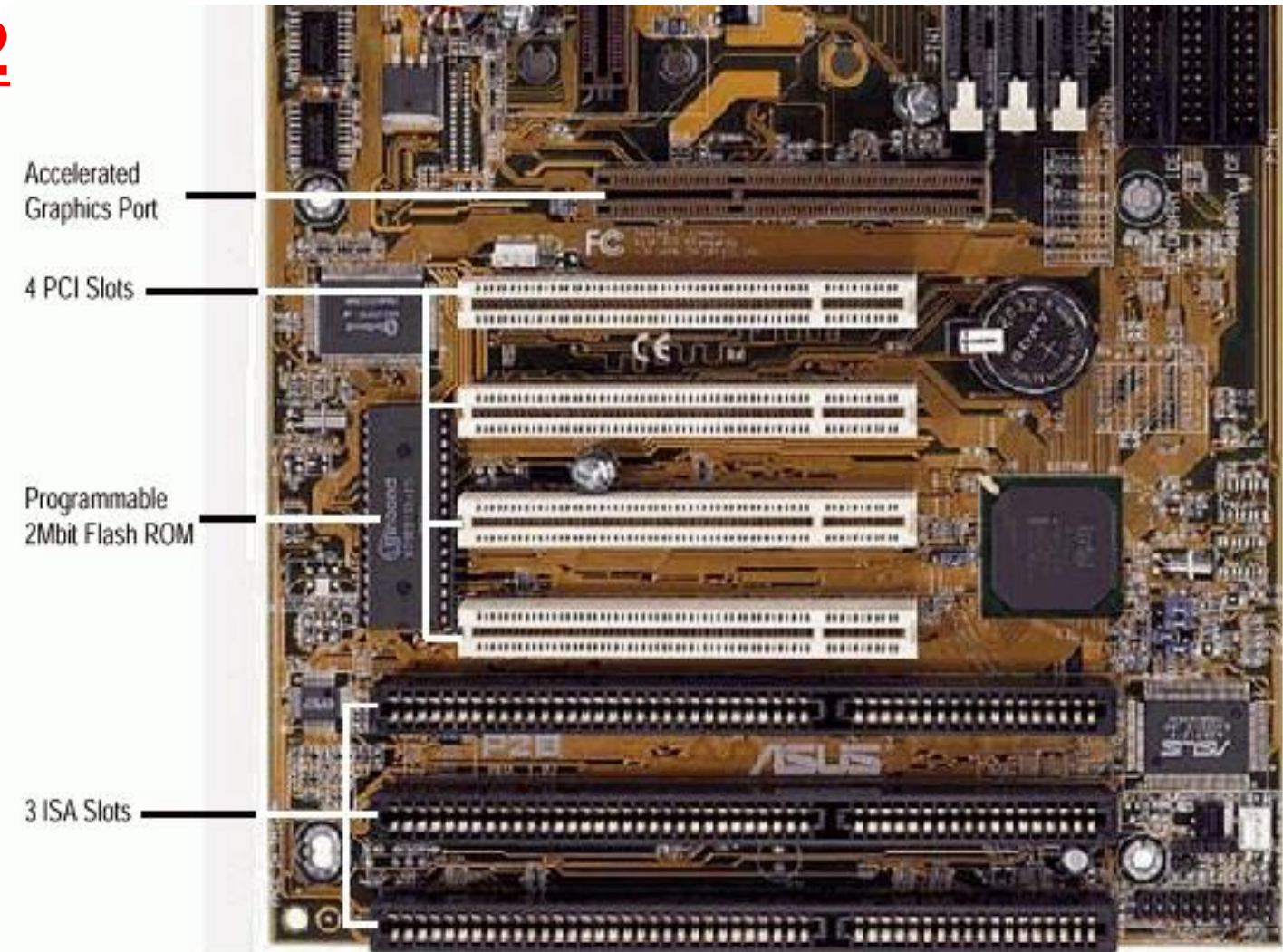
(1) 根据定义可得  $D_r = D/T = D \times 1/T = D \times f$   
 $= 4B \times 33 \times 1000000/s = 132MB/s$

(2)  $64\text{位} = 8B$ ,  $D_r = D \times f$   
 $= 8B \times 66 \times 1000000/s = 528MB/s$

# 目前流行的总线标准：板级， I/O



- RS232
- ISA
- EISA
- STD
- PCI
- AGP
- SCSI
- USB
- .....



# PCI (外围部件互连)总线 @intel 1990?



- 特点: **不依附于**特定处理器, 总线事务
- 独立请求仲裁: **REQ#**, **GNT#**, 仲裁算法自定义
- **FRAME#**: 一次访问 (总线周期) 开始, 长度不限, 由主设备控制
  - **AD**: 对AD线的有效信息进行**奇偶**校验
  - **C/BE#**: 命令信号, 说明接下来的传输类型
    - 地址期指示总线周期类型, 如内存读写、I/O读写、中断确认
    - 数据期指示各字节有效与否
    - **Burst** (cacheline为单位), 单字为Burst的特例
  - **IRDY#**、**TRDY#**: 主从握手, **半同步**, 半互锁
- **中断共享**: 支持大量中断设备

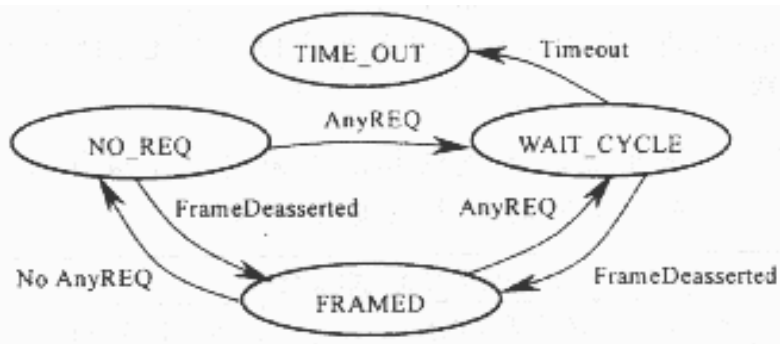
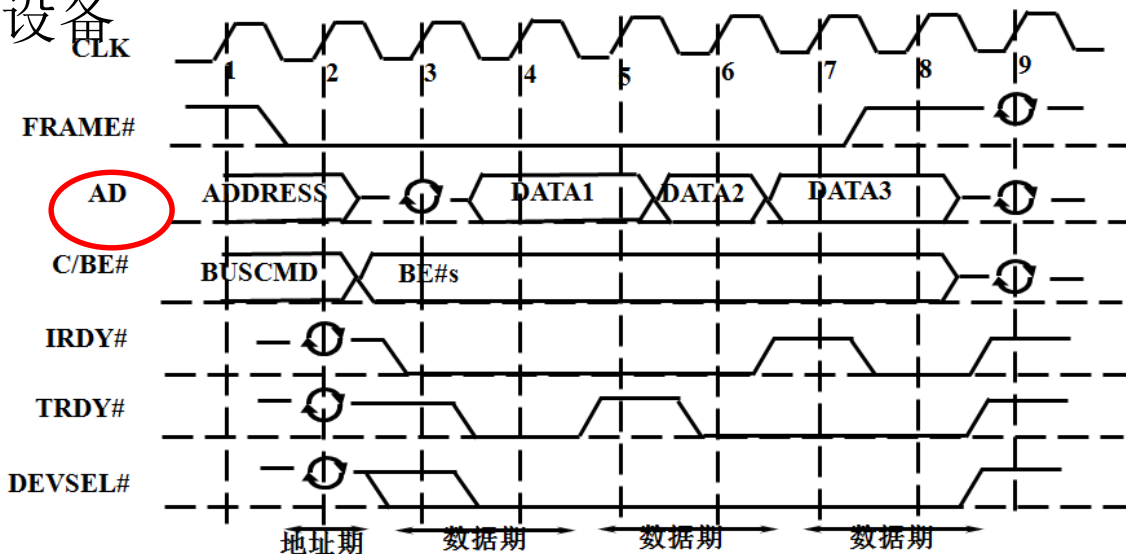
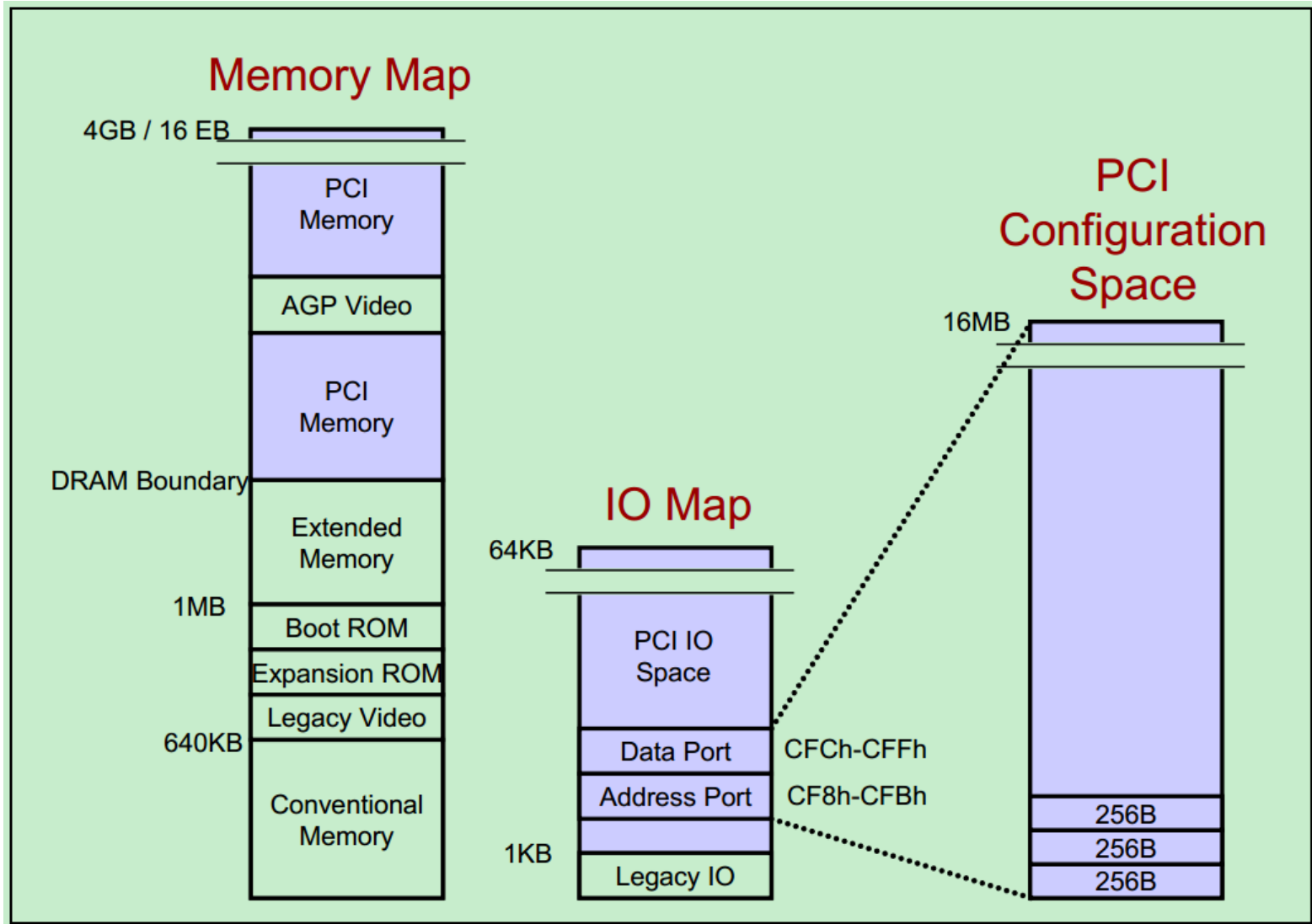


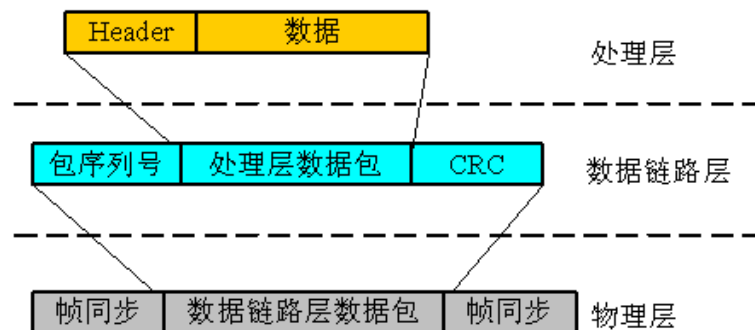
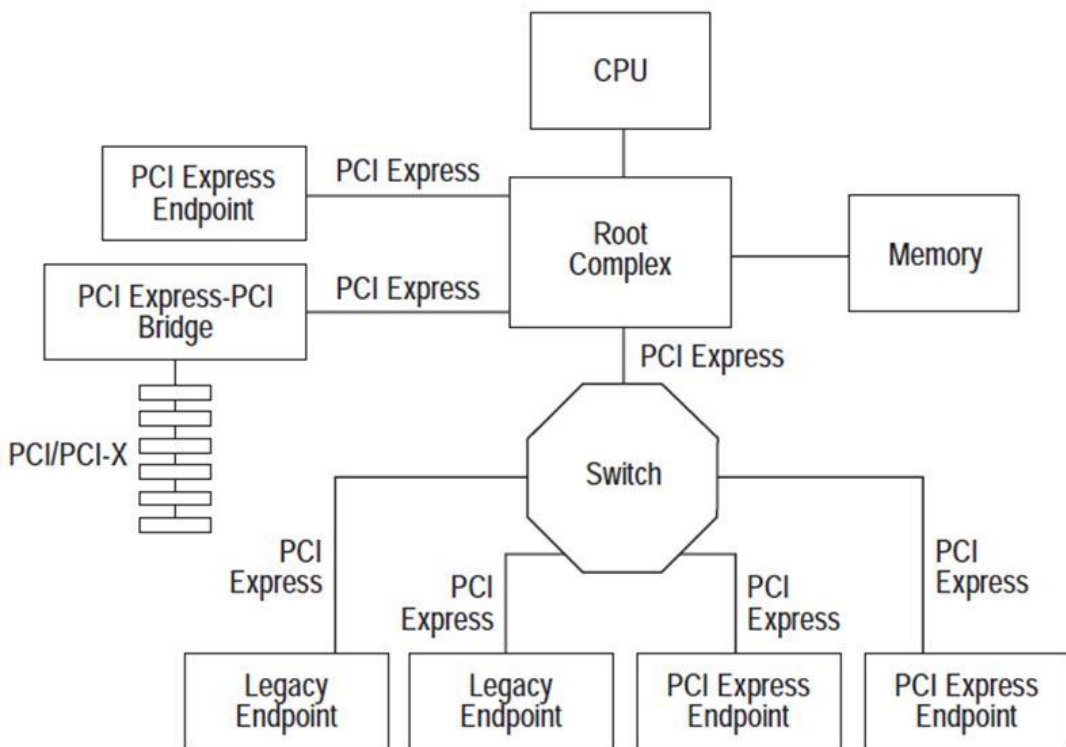
图 2 PCI 总线的状态机及状态转换机制



# PCI总线地址空间: Plug and Play

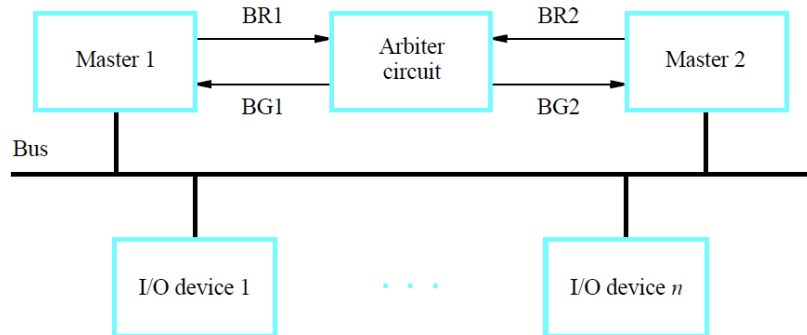


# PCIE (PCI Express, intel, 2004)



- 采用星型拓扑 (点对点)
  - 根组件Root Complex, RC
  - 交换器Switch, SW
  - 端设备, EP, <总线号, 设备号>
    - 可为多功能设备: 功能号
- 与PCI总线的寻址方式兼容
  - RC所在层级为第一级, 定义为Bus0
- 事务层、数据链路层和物理层
  - 数据报
- 串行总线: 4线 (两收两发)

# 总线驱动程序



- 设备抽象
  - 代表总线上的设备执行特定的操作
    - 是**总线控制器**、适配器等**的设备驱动程序**
    - 注意：各类**外设**的读写请求由各自的**设备驱动程序**处理！
- 主要任务（例：**PCI总线**）
  - 枚举其总线上的设备
  - 响应 **PnP** 和电源管理**IRP**（I/O请求包）
  - 总线的多路访问（对某些总线）
  - 管理其总线上的设备
- 典型
  - **PCI, PnP ISA, SCSI, USB**

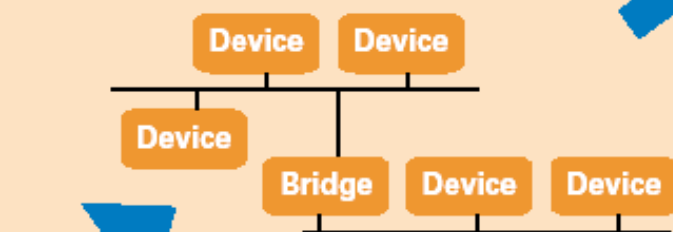
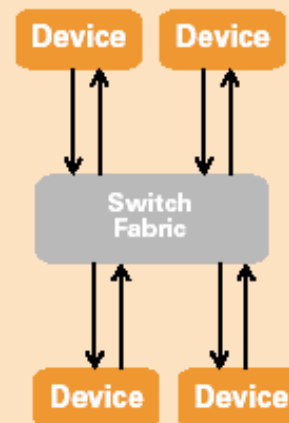
# 板级总线系统及其发展趋势：交换式



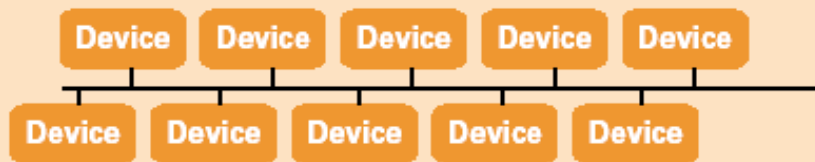
System Performance

- Packet Switched
- Point-to-Point
- Low Pin Count

Very High Frequency

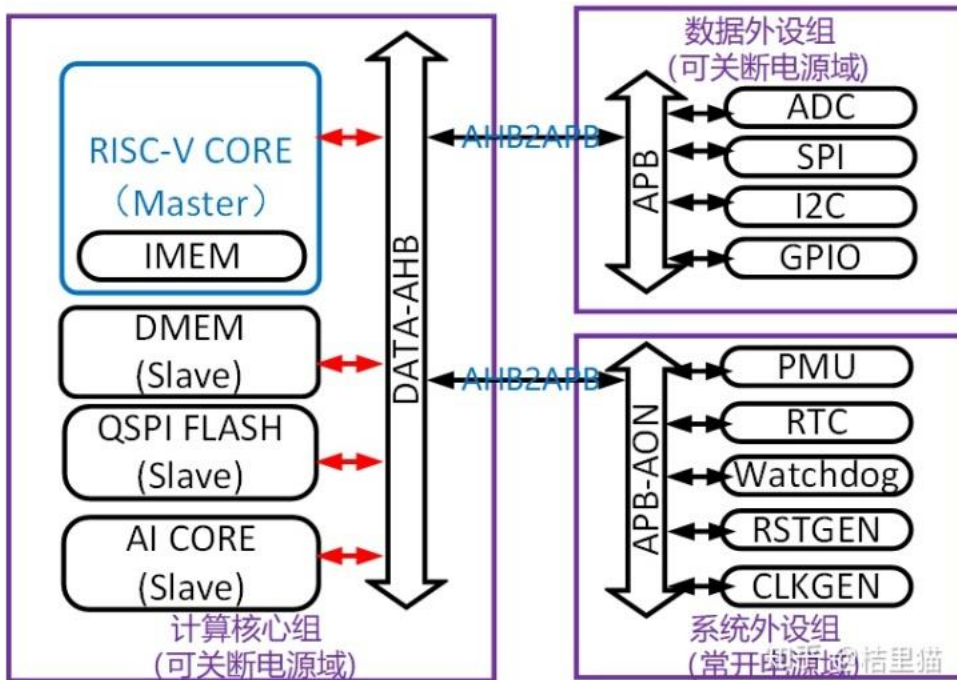


**Bridged Hierarchy**  
Broadcast  
< 133MHz  
Examples: PCI, PCI-X

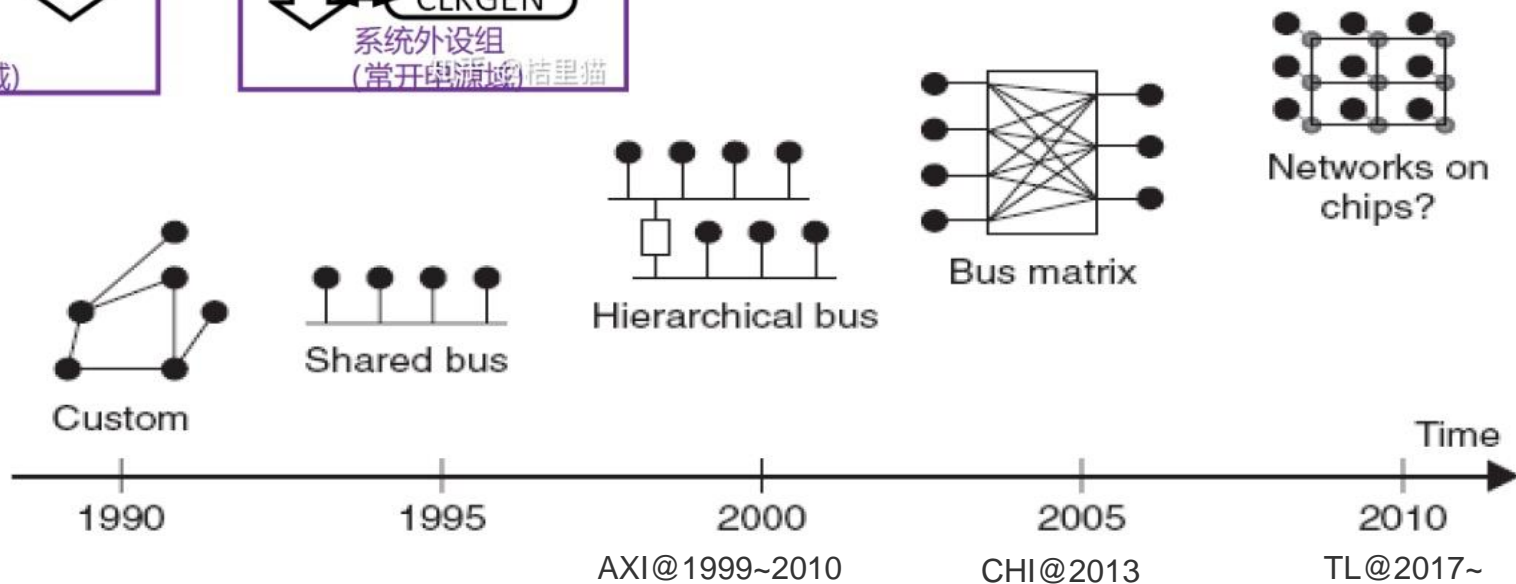


**Single Segment**  
Broadcast  
< 33MHz  
Examples: CompactPCI, VME

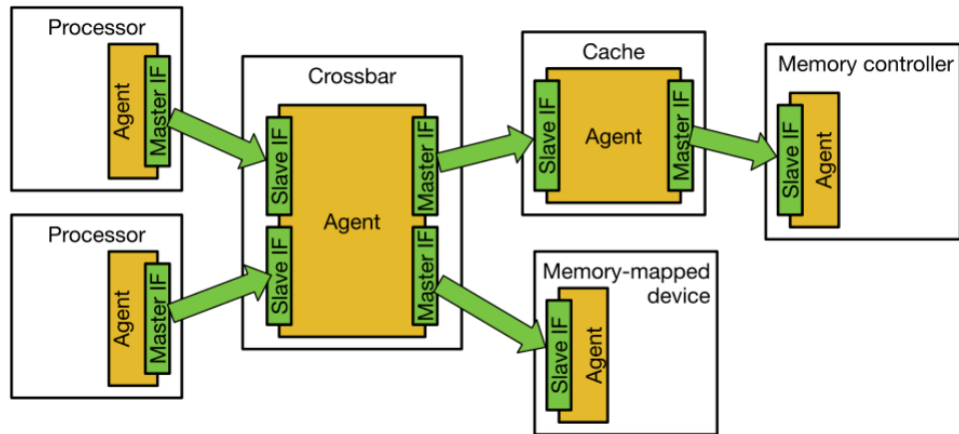
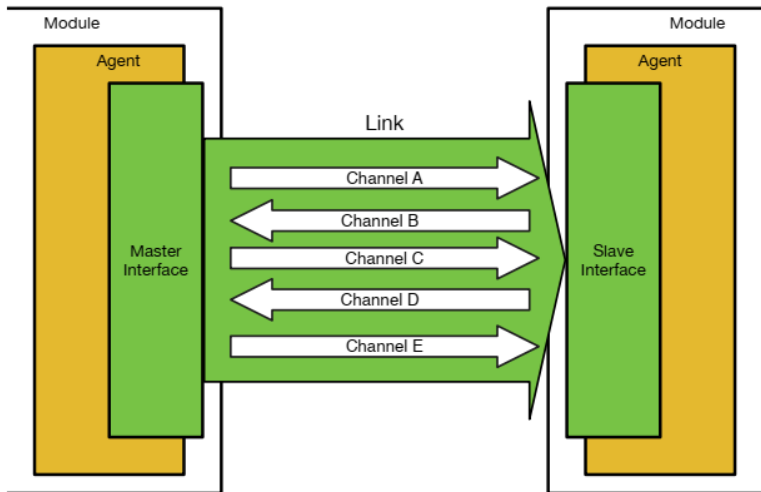
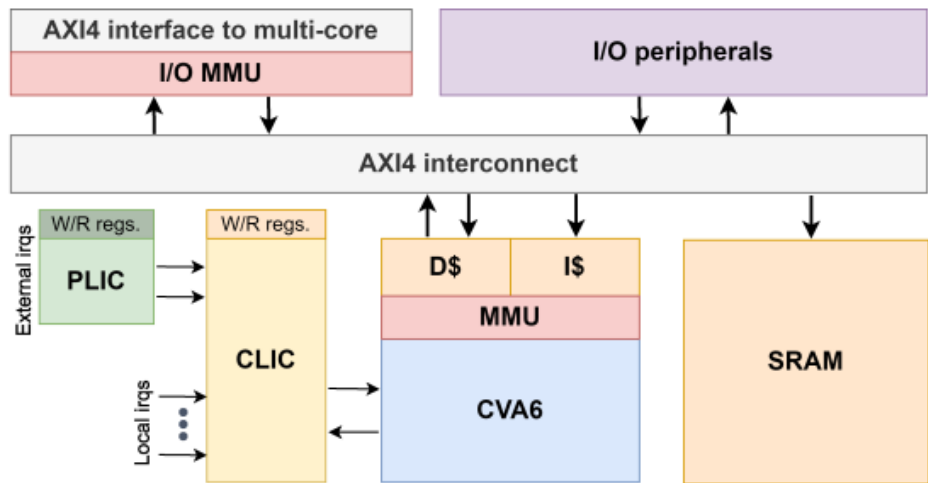
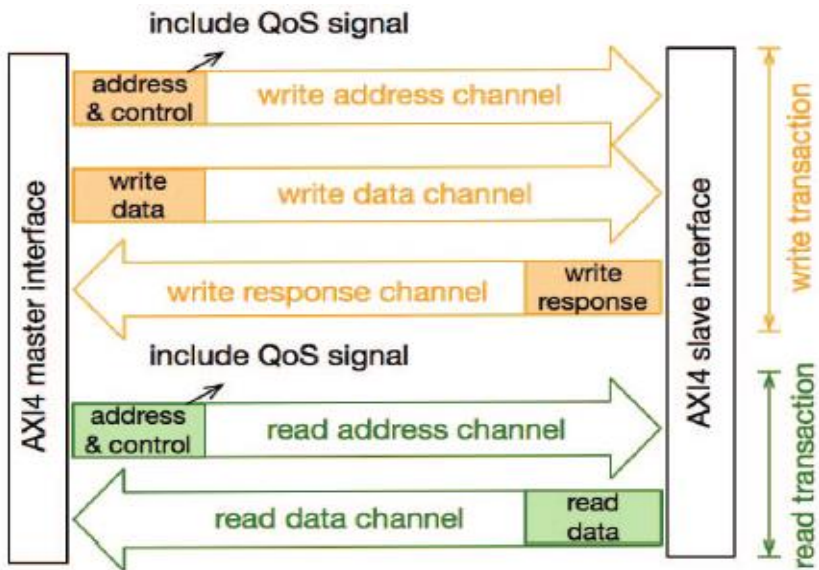
# 片上总线：SOC, NoC, RV处理器



Progress of on-chip comm arch



# AXI总线（ARM）， TileLink总线（RV）：通道级

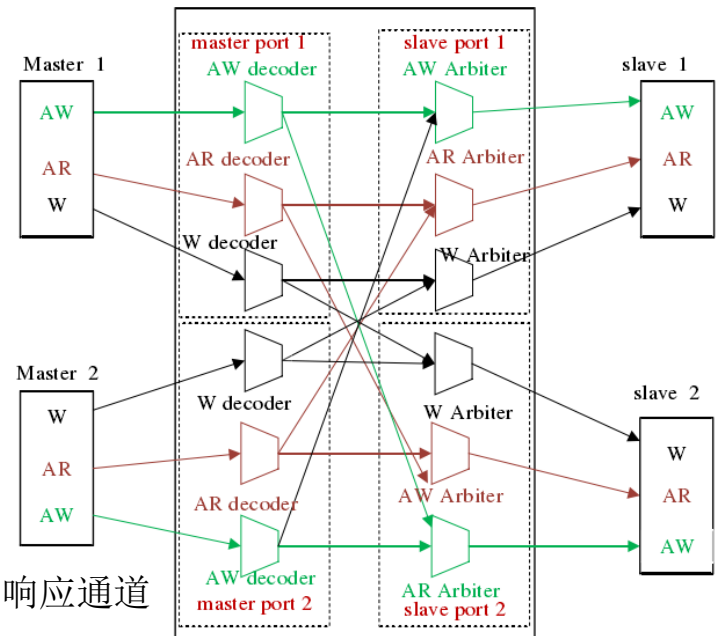
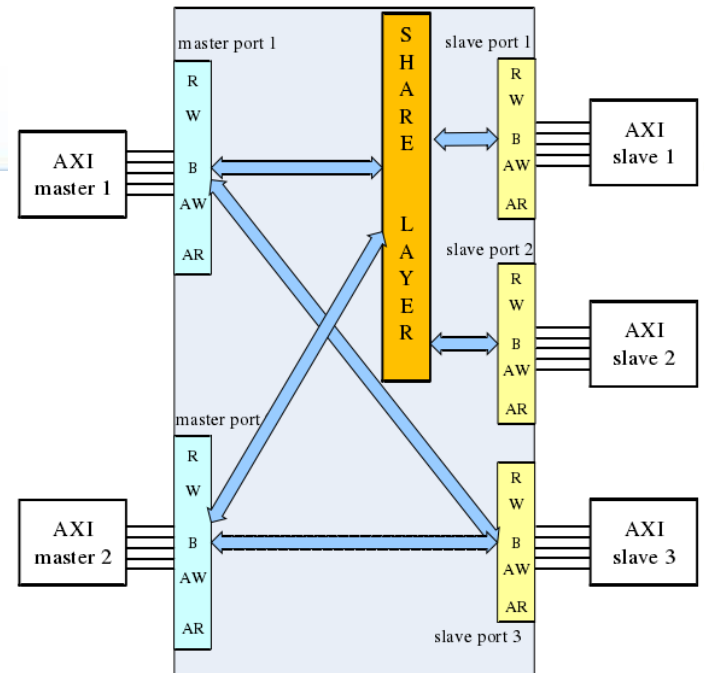


层次化：事务-通道，命令-信号

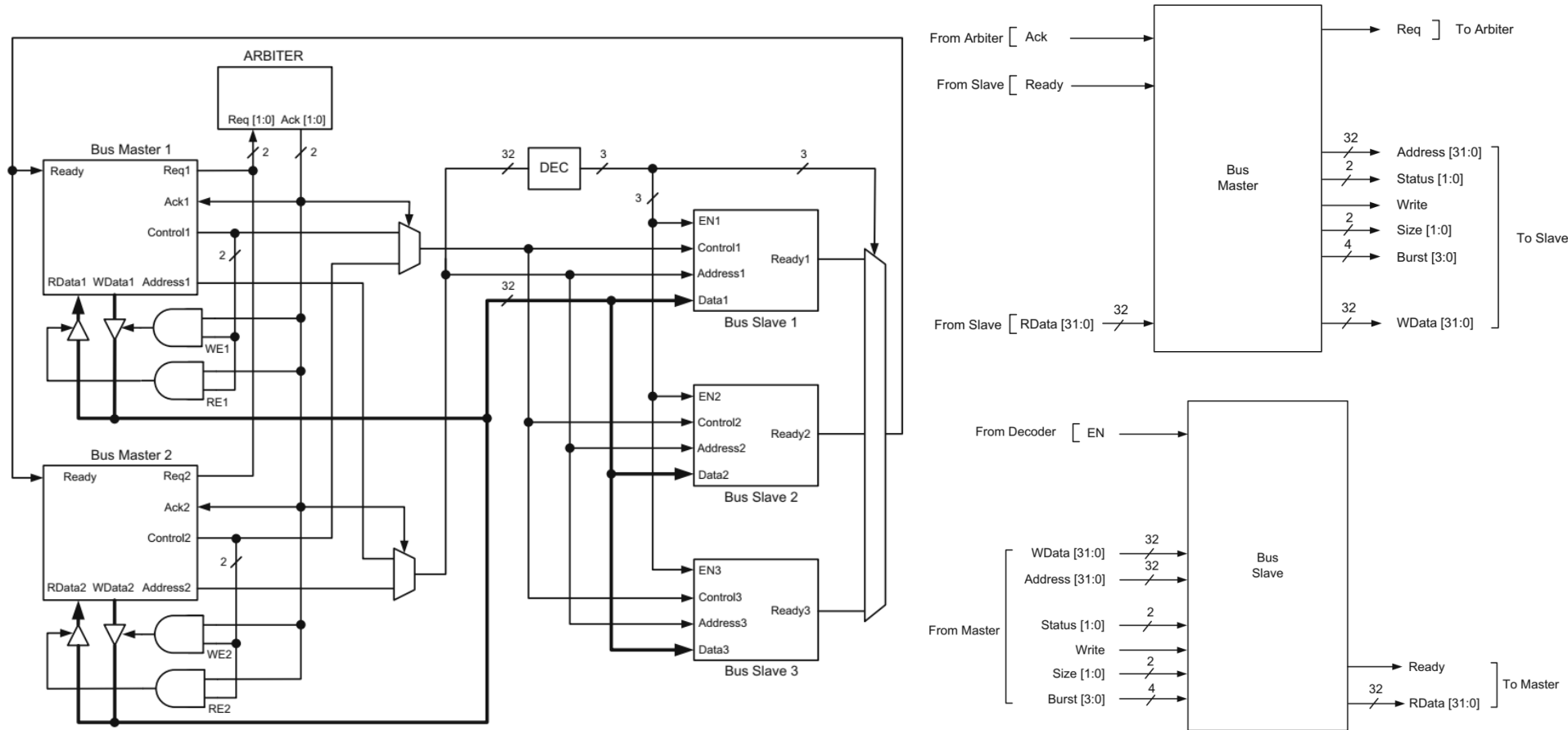
# AXI bus matrix

- 架构分类
  - 共享地址/数据通道
    - 主从间共享5个通道
  - 多地址/多数据通道：下图
    - 主从间独立的5个通道
  - 混合型：上图
    - slave1和slave2为共享型
    - slave3为多地址/多数据型
- Decoder
  - 读/写地址分配器：AW、AR
    - 所有从设备统一编址
  - 写数据分配器：W
- Arbiter：固定优先级，TDMA
- 配置模块

AW写地址通道，W写数据通道，B写响应通道  
AR 读地址通道，R读数据通道



# Bus Physical Structure 示例：信号级

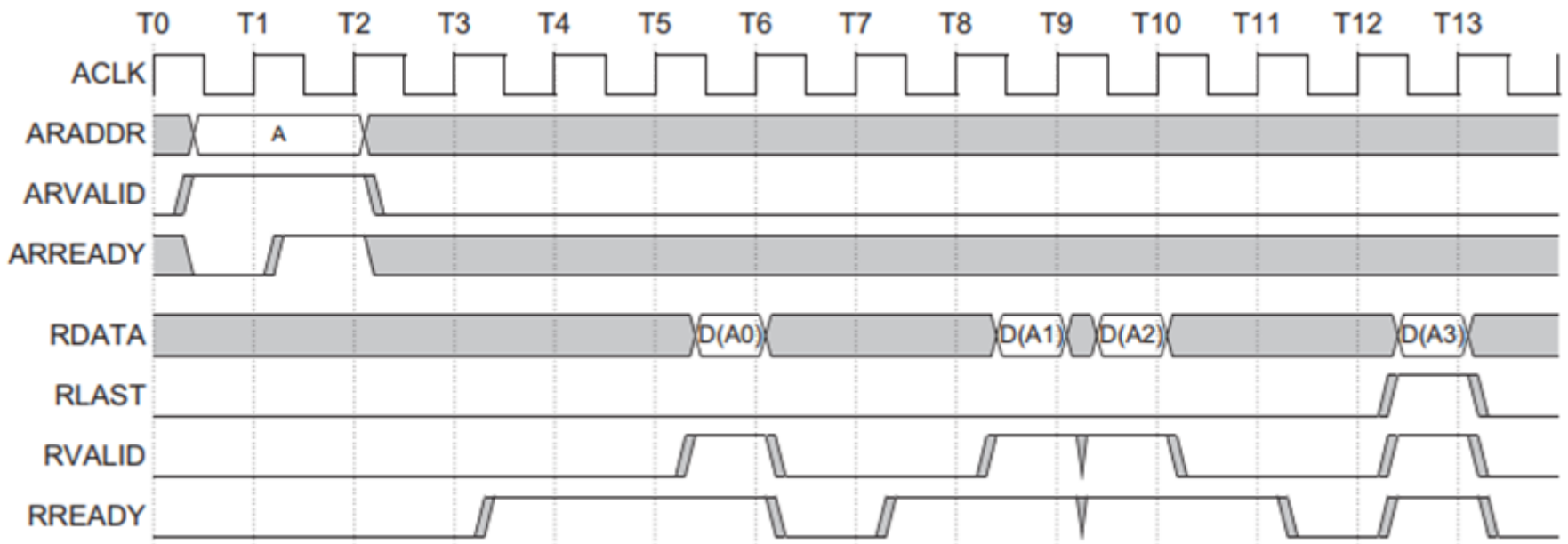


- use a shared-wire bus with an atomic bus protocol
- Tri-state buffer based bidirectional signals

# AXI时序示例：半同步，半互锁



- 长度为4的突发读时序（省略其他控制信号）
  - 当ARVALID-m和ARREADY-s都为高，地址被传递给从接口。
  - 当RVALID-s和RREADY-m都为高，数据被传递给主接口。
  - RLAST-s与第四个有效数据同步，表示最后一个数据。

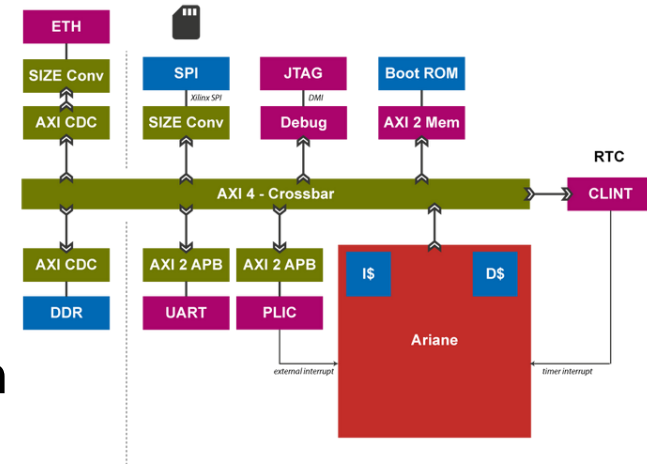


- AXI协议支持事务乱序（OoO）完成
  - 每个事务标记ID。要求相同ID按顺序，不同ID可OoO。

# 例：CVA6处理器和Ariane SoC (AXI4)



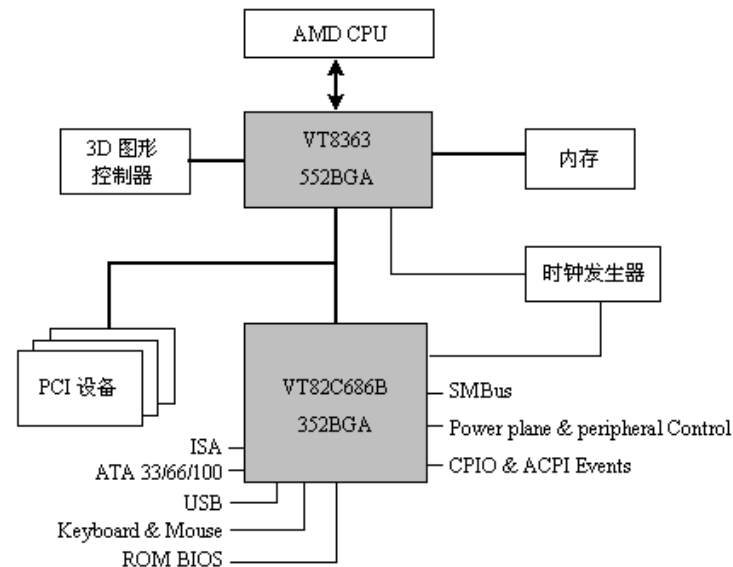
- `cva6_lib`: CVA6源码和验证工具
  - RTL源码、boot程序、FPGA原型、测试套
  - <https://github.com/openhwgroup/cva6>
- RV GNU编译链: `gcc`, `gdb`
  - <https://github.com/riscv/riscv-gnu-toolchain>
- RV工具包:
  - `spike`, 片上调试`riscv-openocd`, 虚拟内核`riscv-pk`, `rv`测试集
  - <https://github.com/riscv-tools>
- Verilator: RTL代码编译/仿真, Verilog模型转换为SC模型
  - <https://www.veripool.org>
- Ariane-sdk: 引导程序BBL, Linux内核
  - <https://github.com/pulp-platform/ariane-sdk>





# 小结

- 总线分类、特性与性能指标, 拓扑结构
- 总线通信控制
  - 传输过程
  - **同步**控制: 时序图
    - 总线时钟宽度?
- 总线数据传输
  - 串并行方式, 编码方式
  - 数据传输模式: 单字节、突发式
- 总线控制器: 功能? 如何实现?
  - 总线**仲裁**
- 软硬件接口: 总线驱动程序?
- 作业
  - 唐2: 3.4、3.8、3.12 (选)、3.14、3.16





Thank You