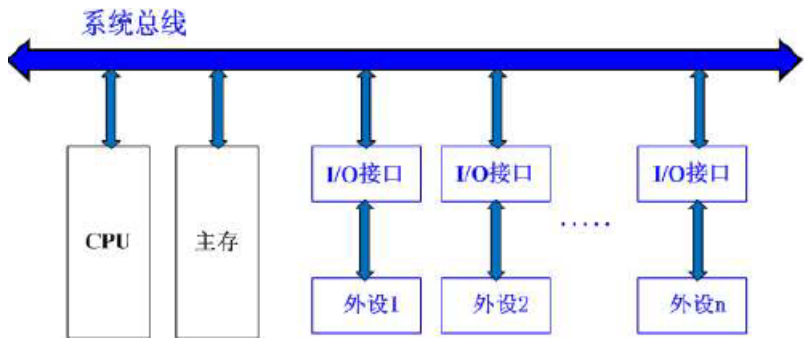


总线系统

笔记源文件: [Markdown](#), [长图](#), [PDF](#), [HTML](#)

1. 总线概述

1.1. 总线概念



- 1 总线含义：连接各部件的**信息通路&共享传输介质**
- 2 总线组成：连接计算机各部件的逻辑电路，不止是单纯的连线
- 3 总线性质：同一时刻，只允许一个部件发信，但允许多个部件收信

1.2. 总线特性

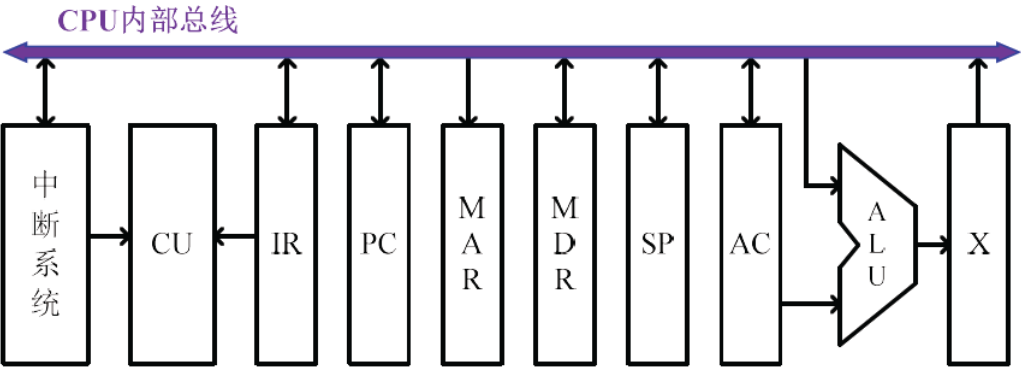
类别	特性
机械特性	尺寸、形状
电气特性	传输方向和有效的电平范围
功能特性	总线上所传信号的功能，按照功能总线可分为地址/数据/控制总线
时间特性	任一根线上的信号在什么时间有效

1.3. 总线性能指标

指标	含义	单位
总线宽度	总线的根数(一次并行传输几位)	bit
总线带宽	单位时间总线传输的数据量，等于 工作频率×总线宽度	MB/s, MBps
工作频率	总线时钟信号频率	MHz
控制方式	集中控制/分散控制	#
通信方式	同步通信/异步通信	#
信号线数	各类信号线的的总根数，可通过复用减少线数	#
负载能力	可扩增电路板的数量	#
扩展能力	总线宽度是否可扩展	#

1.4. 三种类型：片内/系统/通信

1 片内总线：连接芯片内各部件，如CPU内部总线



2 系统总线：单机系统内各大部件间，信息传输的公共通路

总线类型	传输内容	传输方向	总线宽度
数据总线	数据	双向	等于机器字长/CPU字长
地址总线	地址	单向	$2^{\text{宽度}} = \text{最大存储单元大小}$
控制总线	CPU发出控制/反馈给CPU状态	总体双向/每根单向	#

3 通信总线

- 1. 串行：数据从低位→高位，逐位传送
- 2. 并行：一字节(字)一字节(字)地传送

1.5. 总线协议

1 系统总线

名称	描述
ISA	最早的微型计算机系统总线，应用在IBM的AT机上
EISA	扩展的ISA总线，适用于32位CPU，与ISA完全兼容
VESA(VL-BUS)	32位计算机局部总线，针对多媒体PC的高速传送需求设计
PCI	高性能的32位或64位总线，适用于外围部件和处理器/存储器系统的互连
PCI-Express	最新的总线接口标准，将全面取代PCI和AGP

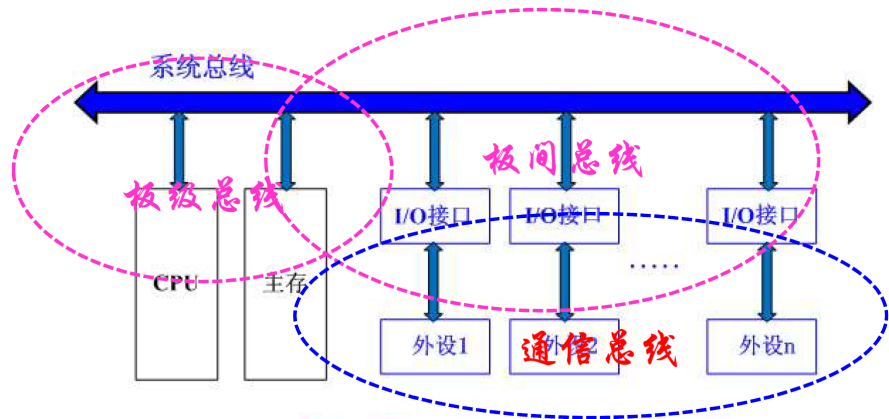
2 设备总线

名称	描述
IDE	集成设备电路接口，用于处理器和磁盘驱动器间的连接

名称	描述
图形接口	图形接口，专用于连接主存和图形存储器
USB	用于外部设备的快速连接
SATA	串行高级技术附件，一种行业标准的串行硬件驱动器接口

2. 总线结构：单/多总线

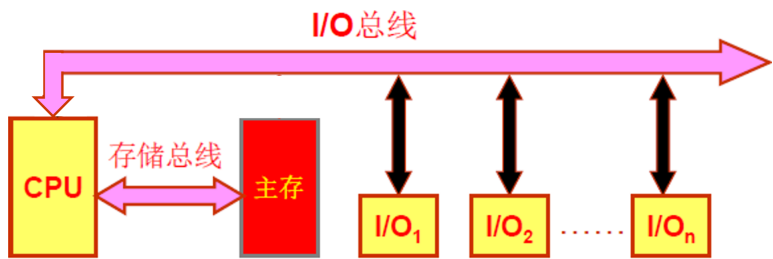
2.1. 单总线：只有一根系统总线



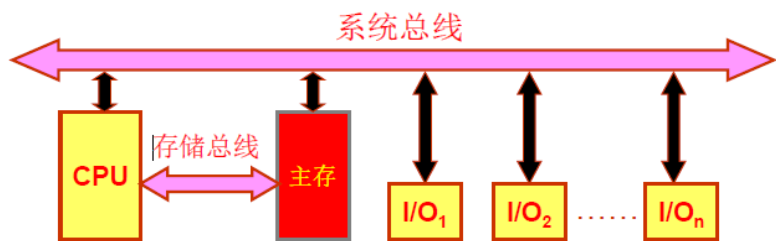
- 1 单总线：包含了数据/地址/控制总线，连接所有的部件
- 2 特点：结构简单，各部分分时使用(效率低)

2.2. 双总线

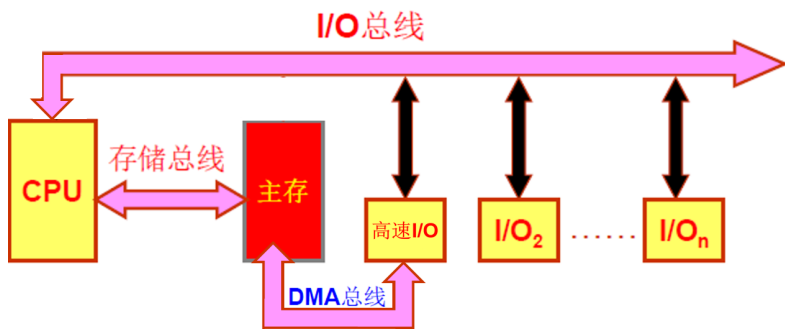
- 1 CPU中心结构：CPU $\xleftrightarrow{\text{高速通道}}$ 主存，IO $\xleftrightarrow{\text{必须经过CPU}}$ 主存



- 2 内存中心结构：CPU $\xleftrightarrow{\text{高速通道}}$ 主存，IO $\xleftrightarrow{\text{直接访问}}$ 主存，但CPU/IO不可同时访问主存



2.3. 三总线结构



- 以CPU为中心: CPU $\xleftrightarrow{\text{高速通道}}$ 主存, IO $\xleftrightarrow{\text{高速访问}}$ 主存

3. (集中式)总线仲裁

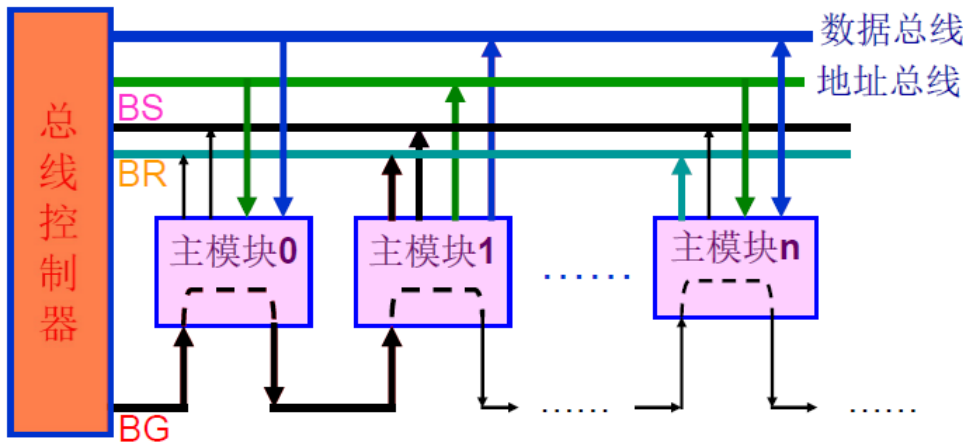
3.0. 前置概念

- 1 什么是总线仲裁: 同一时刻只有一个部件能往总线发数据, 所以, 谁来
- 2 两类连接总线的设备

类型	特征	举例
主设备	能发出总线请求, 能启动总线操作	CPU, DMA控制器, IO设备
从设备	被动响应主模块发来的命令	主存, IO设备

- 3 仲裁思路
 - 1. 软件思路: 为每个主设备设置优先级, 多个主设备请求总线是, 总线优先给优先级高的
 - 2. 硬件思路: 总线控制器中, 设置仲裁机构

3.1. 链式查询方式: 所有部件共用一根总线请求线



- 1 控制信号线

类型	功能	用途
总线请求 BR	主模块 $\xrightarrow{\text{发送请求信号}} \xrightarrow{\text{BR}}$ 总线控制器	公用, 发送请求

类型	功能	用途
总线同意 BG	主模块 $\xleftarrow{\text{回送同意信号}} \xrightarrow{\text{BG}}$ 总线控制器	是对于BR的响应，意味着请求得到响应
总线忙BS	主模块 $\xrightarrow{\text{发送忙信号}} \xrightarrow{\text{BS}}$ 总线控制器	相当于占用了总线的主模块，宣称其占有

2 设计要点

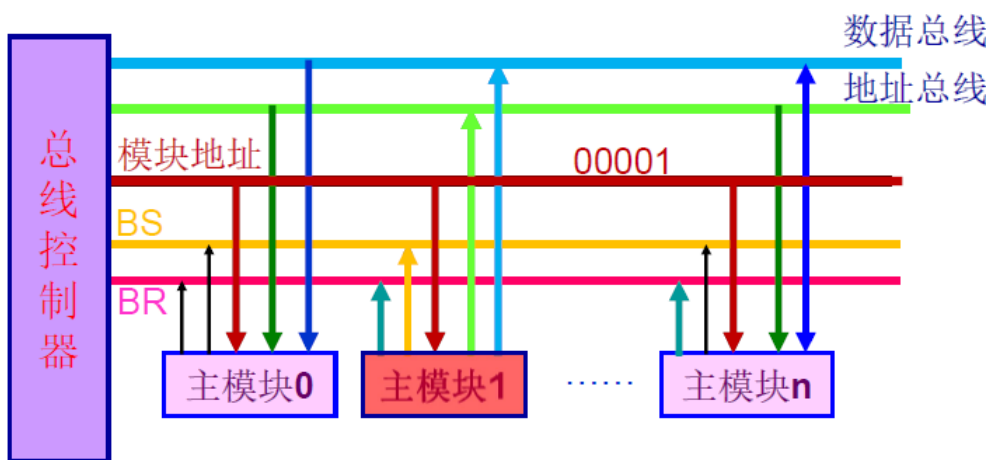
1. 结构：通过优先链，将所有主模块串起来
2. 优先级：再上述串中，离总线控制器越近的优先级越高
3. 查询方法：
 - 从最靠近控制器的主模块开始查：你要不要用总线？
 - 跳过不要的，一旦查到要的(优先级必定最高)，便让该模块使用总线
 - 总线用完后释放，然后继续往后边查：你要不要用总线？

3 查询过程

1. 请求：假设1,n模块同时请求(BR有效)
2. 查询：BG信号开始沿优先链查询，一直查询到模块1时BG截断
3. 占用：模块1回送BS信号，占用总线，撤销BG/BR信号
4. 释放：模块1使用完总线后，也撤销BS信号以此释放总线

4 缺点：查询速度慢，故障敏感(一个模块坏了后面都白搭)，优先级固定不灵活

3.2. 计数器定时查询



1 结构：总线控制器里植入一个计数器(查询用)

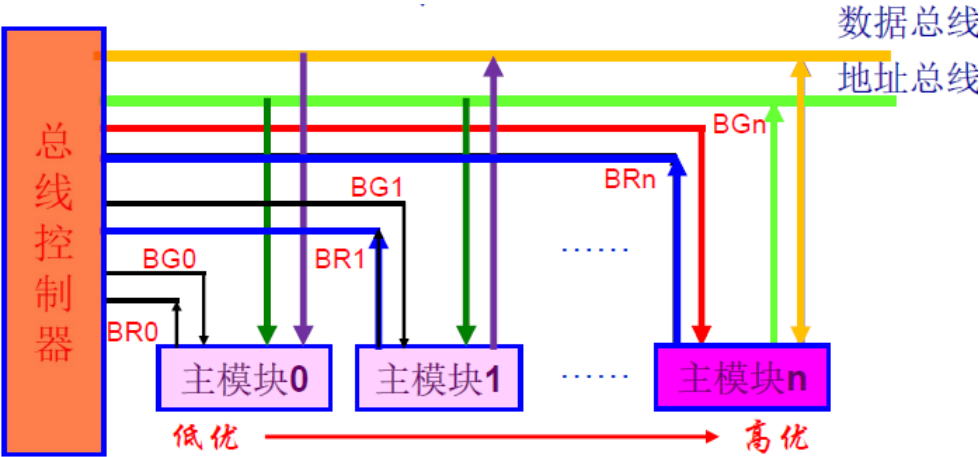
2 查询过程：

1. 初值设定：查询开始，计数器启动，设定一初值(不一定为0)
2. 初值广播：计数初值发往每个模块
3. 请求响应：如果某个模块请求了总线，且模块地址=计数值，则获得总线控制权
4. 计数更新：计数值++，每次更新后，都要将计数值广播

3 优先级确定：关键在于**计数器初值设置**

- 1. 初值=0，则0模块最优先
- 2. 初值=上次查询终止时计数器的值，则上次终止点最优先
- 3. 初始=程序设定

3.3. 独立请求



1 结构:

- 1. 每个模块有一个专门BR, BG各自独立请求
- 2. 总线控制器设置并行排队线路，将发来请求的模块全部进行排队

2 优先级: 队列里排前面的优先级高

4. 总线通信: 部件1 $\xleftrightarrow[\text{总线传输}]{\text{时间协调}}$ 部件n

4.1. 总线通信概念概述

1 一次总线操作的过程

操作	描述
申请分配	主模块申请总线
寻址	主模块获得总线控制权，向从模块发地址&读写命令
传送	主从模块数据交互
结束	主模块让出总线

2 总线周期: 一次完整的总线操作耗时，有以下几种

总线周期类型	地址传送时间	数据传送时间
正常总线周期	一次	一次
BURST总线周期	一次	多次*

*如果读写部件太慢了，则需要增加传送时间完成读写，增加的传送时间就是**总线等待状态**

3 总线操作类型

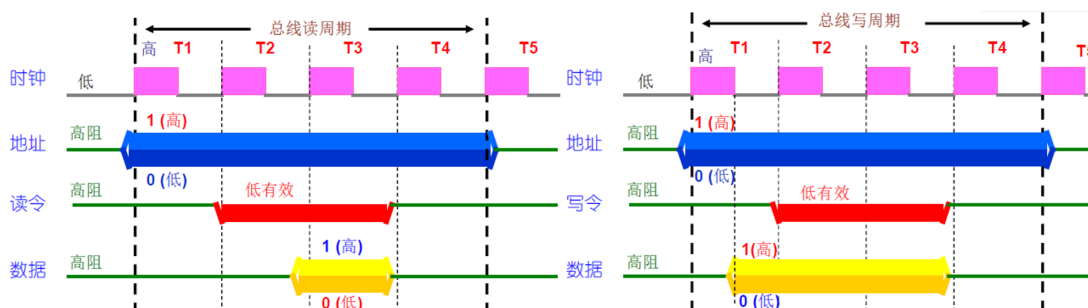
类型	方向	示例
内存读	从模块发送，主模块接收	CPU或IO接口，从主存中读
内存写	主模块发送，从模块接收	CPU或IO接口，从主存中写
外设读	从模块发送，主模块接收	CPU，从IO设备中读
外设写	主模块发送，从模块接收	CPU，从IO设备中写

4.2. 同步通信：由总线控制器时钟统一协调

1 时间分配

1. 一个总线周期中，所含时钟个数固定
2. 所有的模块，总线周期都一样长

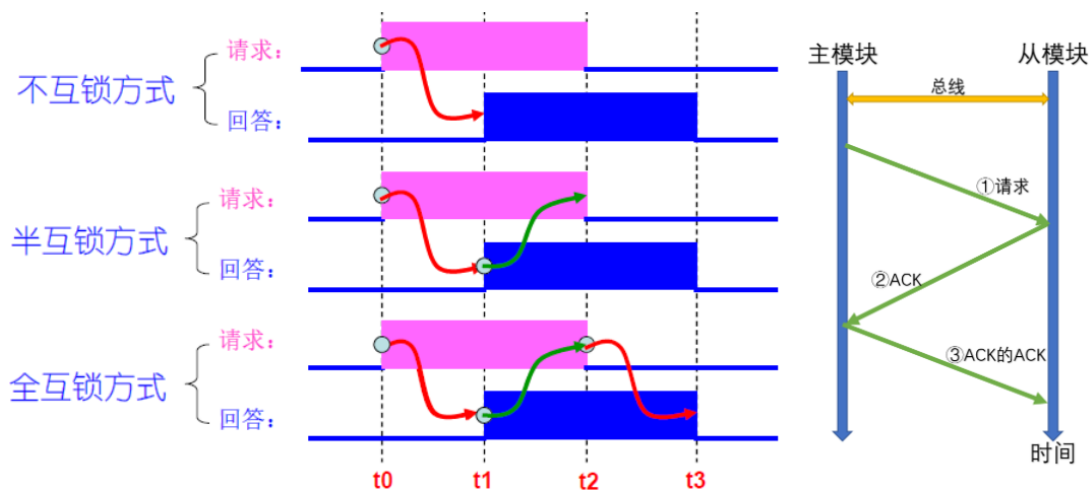
2 总线同步的数据IO：左边为读时序，右边为写时序



时间	从模块读，数据输入主模块	从模块写，数据输出主模块
T_1	主模块 $\xrightarrow[\text{总线}]{\text{地址}}$ 从模块	主模块 $\xrightarrow[\text{总线}]{\text{地址}}$ 从模块
$T_{1.5}$	#	主模块 $\xrightarrow[\text{总线}]{\text{待写数据}}$ 从模块
T_2	主模块 $\xrightarrow[\text{总线}]{\text{读命令}}$ 从模块	主模块 $\xrightarrow[\text{总线}]{\text{写命令}}$ 从模块
T_3	主模块 $\xleftarrow[\text{总线}]{\text{读出数据}}$ 从模块	等待数据写入从模块
T_4	主模块撤消读命令、地址	主模块撤消读命令、地址、数据

4.3. 异步通信：无时钟/总线周期随模块工作时间变动

1 三种异步通信



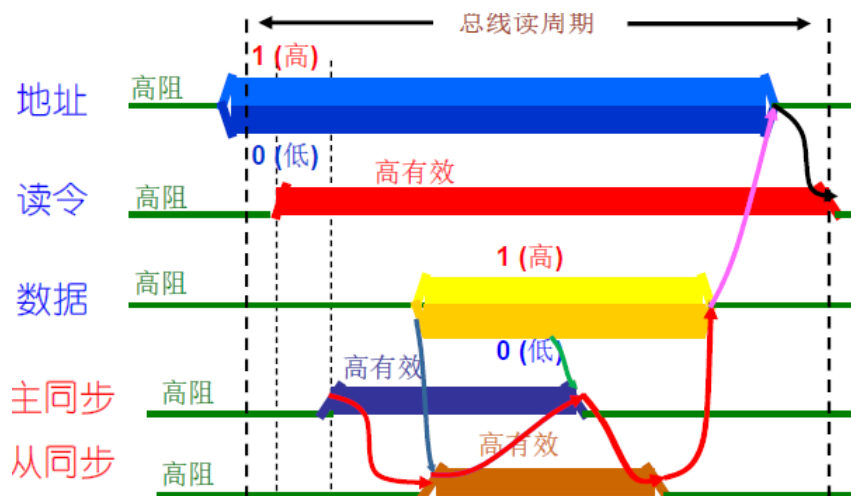
方式	主模块	从模块
不互锁	发出①后，过段时间默认对方收到了①*	发出②后，过段时间默认对方收到了②**
半互锁	发出①后，收到②后才认定对方收到①	发出②后，过段时间默认对方收到了②
全互锁	发出①后，收到②后才认定对方收到①	发出②后，收到③后才认定对方收到②***

*确认对方收到①后，主模块撤销请求信号

**确认对方收到②后，从模块撤销ACK信号

***ACK的ACK实质上是主模块撤销请求信号的确认

2 全互锁异步通信的时序



1. 发出信号

部件	发送方向	发送内容
CPU	往主存	地址/读指令/主同步
主存	#	数据/从同步

2. 收到同步信号后的响应

部件	同步信号来源	操作
CPU	从同步	读数据，撤销主同步
主存	#	撤销数据，撤销从同步

3. 最后：CPU撤销地址和读指令