

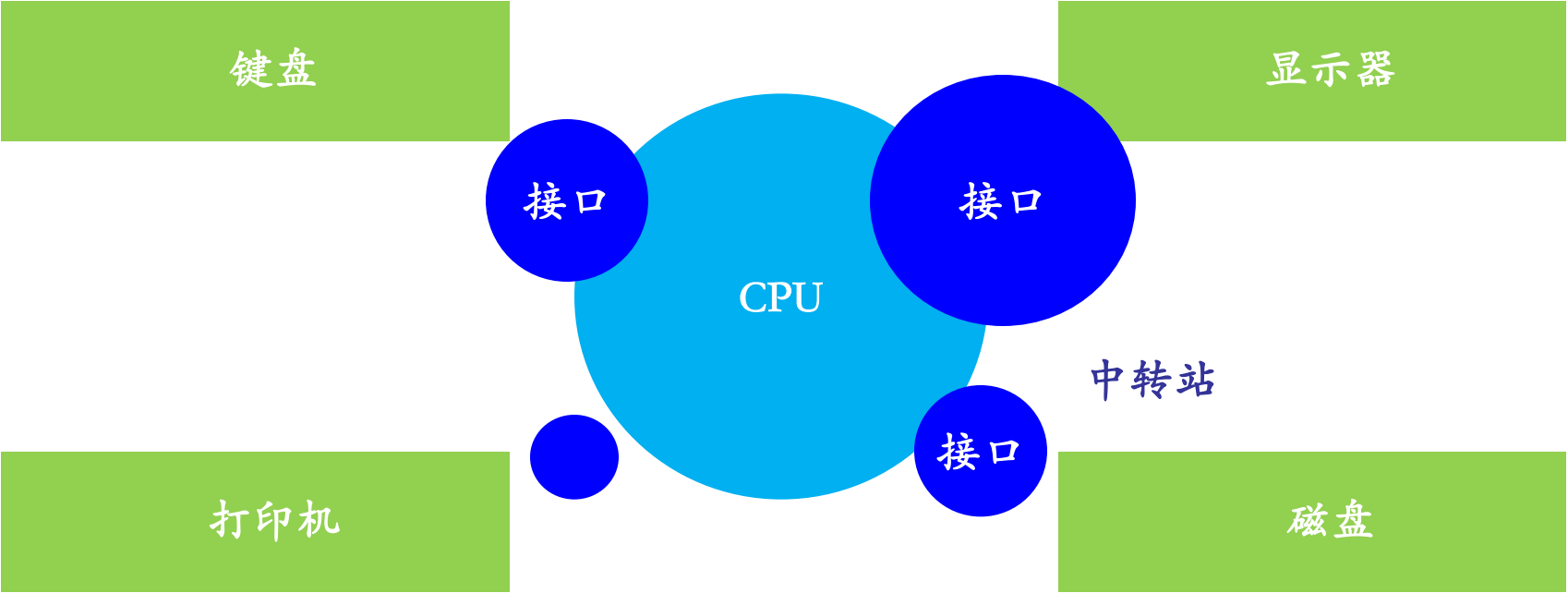
# 计算机接口技术及应用

Computer Interface Techniques & Application

# 接口概述

# 1.1 微机接口与接口技术

- **接口 (interface)**：就是微处理器与“外部世界”的连接电路，是CPU与外界进行信息交换的中转站
- **为什么要在CPU和外设之间设置接口电路？**
  - 信息类型和信号电平的匹配问题  
(信号线的功能、逻辑、时序)
  - 速度的匹配问题
  - 提高了CPU的效率
  - 使外设的发展不受限制 (硬件结构不依赖CPU)



外部设备

接口概念

# 1.1 微机接口与接口技术

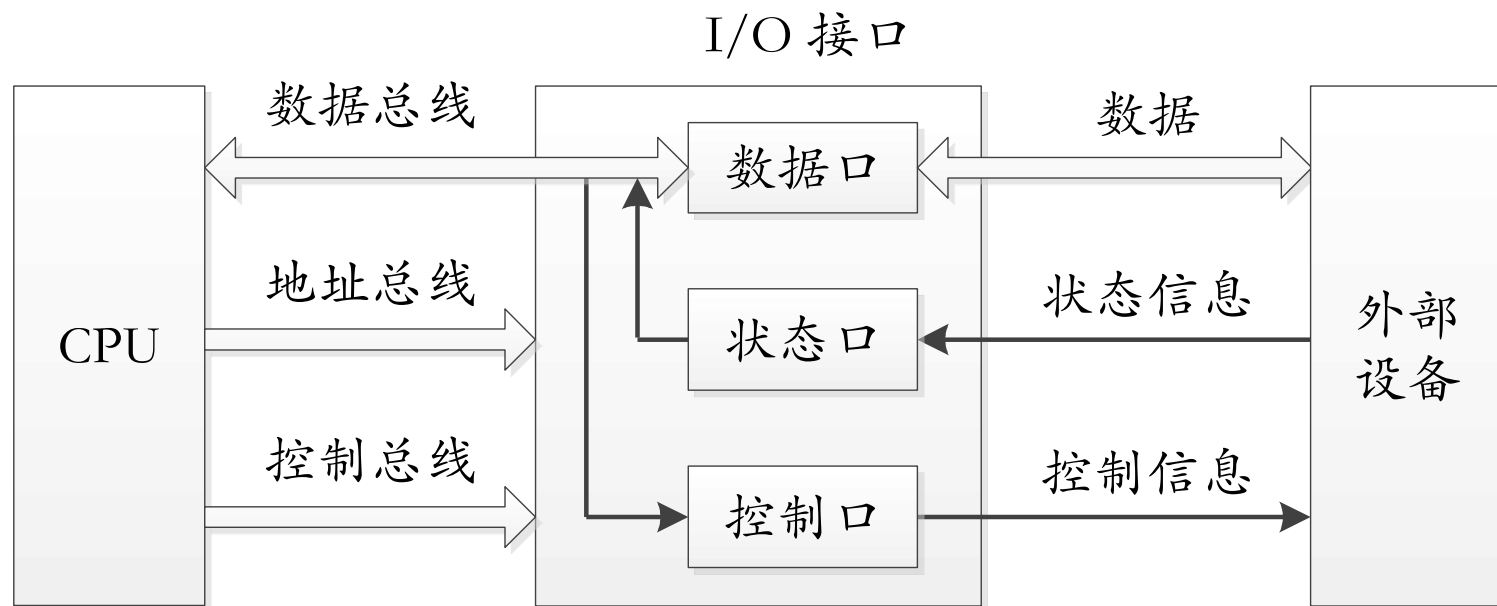
CPU和I/O设备之间通过接口交换信息

- 数据信息
  - CPU与外部设备交换的基本信息是数据（8位、16位、32位、64位 …）
  - 数据信息分三种类型：数字量、模拟量、开关量
- 状态信息
  - 反映外设当前的工作状态（外部设备通过接口传送给CPU）
- 控制信息

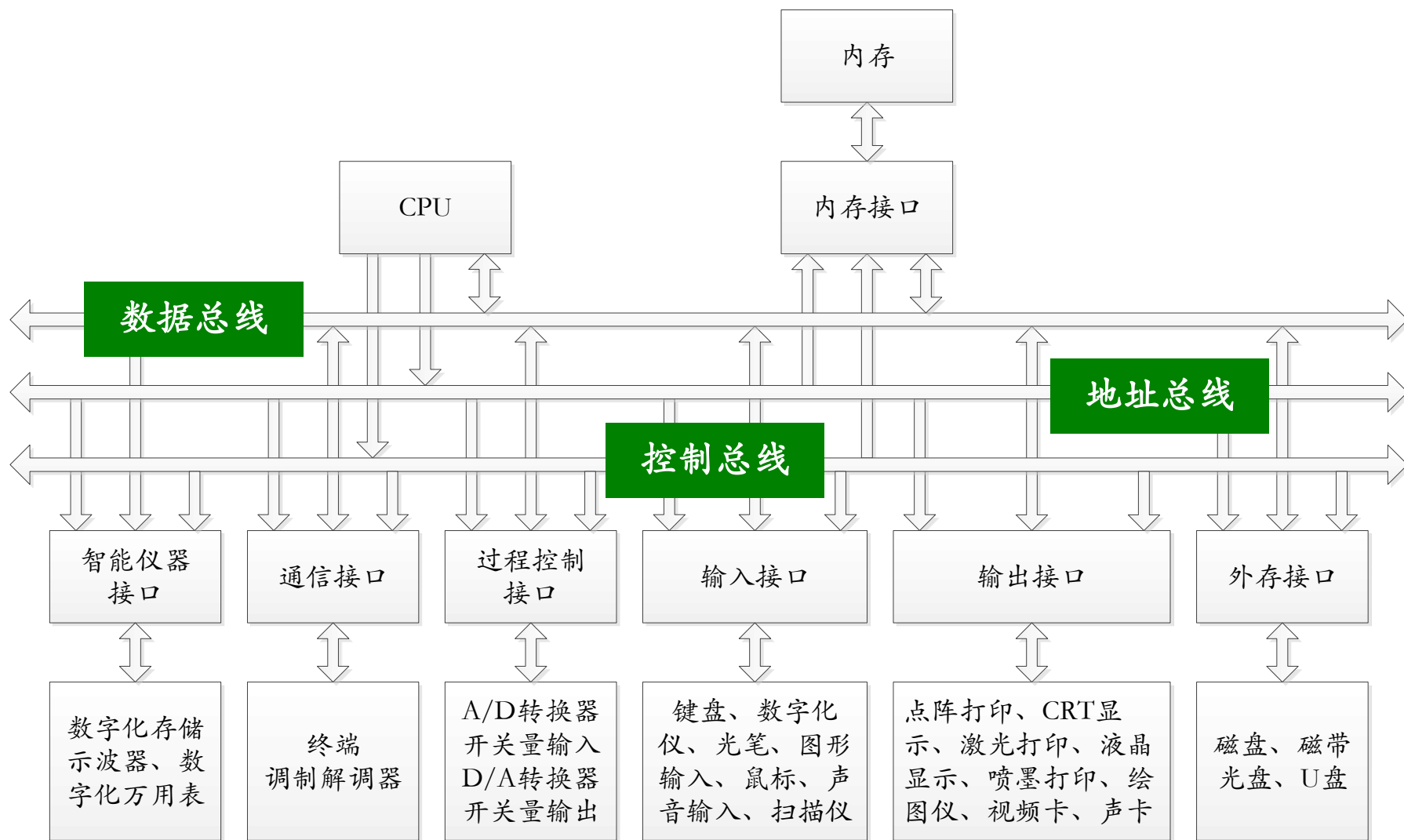
# 1.1 微机接口与接口技术

CPU和I/O设备之间通过接口 **交换信息**

- 控制信息
  - CPU发送控制信息实现对外部设备的控制  
(通过接口传送给外部设备)



外设接口框图



微机系统接口框图



# 1.1 微机接口与接口技术

- 接口技术在微机应用中的作用
  - 导弹、卫星、机器人、无人机……
  - 手机、平板电脑、条形码机……
- 计算机的应用是随着外部设备的不断更新和接口技术的发展而深入到各个领域的

# 1.2 接口的功能

## 1. 接受、解释和执行CPU命令的功能

- 接受命令

CPU对被控对象（外部设备）的控制命令以命令代码的形式传送给接口中的**命令寄存器（命令口）**

- 解释命令

由接口电路对命令代码进行识别和分析，分解成若干个控制信号

- 执行命令

传送到外部设备，从而使其产生相应操作

# 1.2 接口的功能

## 2. 返回外设状态的功能

- 为了能让CPU及时正确了解与其通信的外设的工作状态，接口电路中设置一些**状态寄存器（状态口）**，以状态代码的形式存放外部设备的状态信息。CPU通过读状态寄存器，作出判断和处理
- 执行CPU命令之**前**/执行命令过程**中**/执行命令之**后**
- 忙/闲、准备就绪/未准备就绪、满/空、溢出错、格式错、校验错

# 1.2 接口的功能

## 3. 数据的缓冲功能

- 主要解决主机高速与外设低速的矛盾，避免因速度不一致而丢失数据。一般接口电路都设置**数据缓冲寄存器（数据缓存器）或锁存器（register /latch）**：数据口（输入/输出）
- 这些锁存器常常有**驱动作用（负载能力）**
- 数据缓冲器直接连在计算机系统数据总线上，因此具有**三态特性**

使高速工作的CPU和慢速工作的外设协调工作

# 1.2 接口的功能

## 4. 信号的转换功能

- 由于外设所需的**控制信号**以及它所能提供的**状态信号**往往同微机的**总线信号**不兼容（功能定义、逻辑关系、高低电平、工作时序），信号的转换功能在接口电路设计中十分重要
- 通常遇到的信号转换有：电平转换，A/D，D/A转换，串/并和并/串的转换，时序转换，逻辑转换，数据宽度的转换

# 1.2 接口的功能

## 5. 设备选择（端口寻址）功能

- 微机系统一般有多种外设，同一种外设也可能有多台，而一个CPU在同一时刻只能与一台外设进行信息的交换，所以要在接口电路中设置I/O端口地址译码电路对外设进行寻址
- 只有被选中的设备才能和CPU建立真正的连接，进行信息交换，反之则不可
- CPU将I/O设备的端口地址送到接口的地址译码电路，译码电路将地址代码翻译成I/O设备的选择信号（一般高位地址用于接口芯片的选择、低位地址用于芯片内部寄存器的选择）

# 1.2 接口的功能

## 6. 数据宽度与数据格式转换功能

- CPU处理并行数据（8位/16位/32位），一些外设只能处理串行数据，接口应该具有“并-串”和“串-并”转换功能（移位寄存器）
- CPU与有些外设交换数据时，要求按一定的数据格式传送

例如，串行通讯中的“起止式异步通信数据格式”和“面向字符的同步通信数据格式”

# 1.2 接口的功能

## 7. 中断管理功能

- 当外设和CPU之间用中断方式进行数据传输时，如果有多个外设同时向CPU提出中断请求，应该有一个能够进行中断优先级判别、提供中断向量等工作的功能单元（8259）
- 增加微机系统对外界的响应速度，使CPU和外设并行工作，提高CPU的效率



# 1.2 接口的功能

## 8. 可编程功能

- 为了增加接口的灵活性、可扩充性、通用性，接口电路中一般提倡使用一些具有可编程功能的芯片
- 在不改动硬件的情况下，只修改驱动程序就可改变接口的工作方式

# 1.3 接口的组成

硬件和软件两部分组成

- **硬件部分**

- 基本的逻辑电路，包括命令寄存器、状态寄存器和数据寄存器，作用：接收执行命令、返回状态、传送数据（可编程大规模接口芯片）
- 地址译码电路，作用：设备/寄存器选择（可编程逻辑芯片、译码器芯片，不包含在集成接口芯片中，由用户自行设计，必不可少）
- 供选电路

# 1.3 接口的组成

硬件和软件两部分组成

- **硬件部分**

- 供选电路

例如，数据传输方式（中断控制器或DMA控制器）；速度控制和发声（定时/计数器）；数据宽度转换（移位寄存器）

# 1.3 接口的组成

- 软件部分

- 初始化程序段

对可编程接口芯片（或控制芯片）都需要通过相应的方式命令或初始化命令设置**工作方式**及**初始条件**

- 传送方式处理程序段

查询方式：检测外设或接口状态的程序段

中断方式：中断向量修改、中断源的屏蔽/开放、中断结束等处理的程序段；主程序和中断服务程序

DMA方式：通道的开放屏蔽等处理的程序段

# 1.3 接口的组成

- 软件部分

- 主控程序段

完成接口任务的程序段

例如，数据采集程序段（发转换启动信号、查转换结束信号、读数据、存储数据）；步进电机控制程序段（运行方式、方向、速度、启/停控制）

- 程序终止与退出程序段

程序结束退出前对接口电路中硬件的**保护程序段**（引脚设置为高/低电平，输入/输出状态）

- 辅助程序段（人机界面）

# 1.4 接口电路的结构形式

- 固定式结构
  - SSI或MSI
- 半固定式结构
  - GAL/PAL, FPGA/CPLD
- 可编程结构
  - 大规模接口芯片 (本课程的内容)
- 智能型结构
  - 专门的I/O处理器 (例如I8089)
  - 通用单片微机

# 1.5 CPU与接口数据交换方式

微机与外部设备之间的数据传输实际上是  
CPU与接口之间的数据传输

1. 查询方式
2. 无条件传送方式
3. 中断方式
4. DMA方式

# 1.5 CPU与接口数据交换方式

## 1. 查询方式

- 在程序中安排一段由**输入/输出指令**、**测试指令**、**状态转移指令**组成的程序段
- CPU使用**测试指令**和**条件转移指令**循环检测设备完成准备工作的状态

例如，CPU以查询方式给打印机传送数据，先查打印机忙否，不忙则传送数据，忙则继续查询，即等待

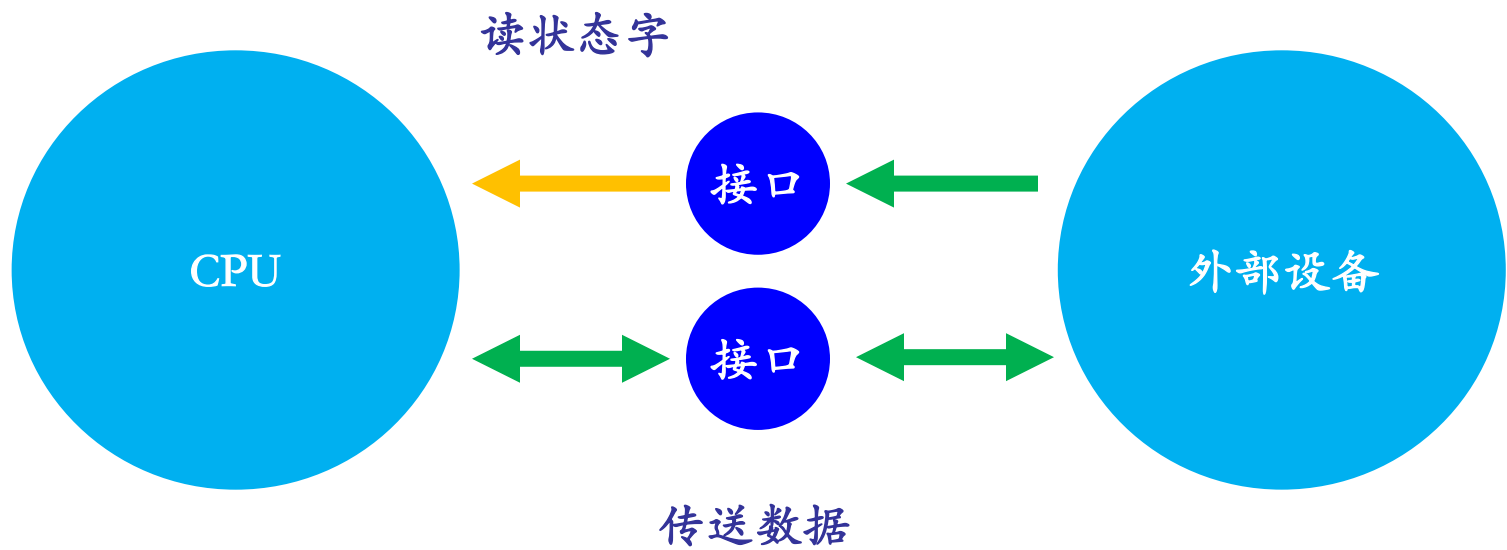
### 特点

- 接口电路简单，硬件开销小，效率低

### 适用场合

- 在CPU不太忙且传送速度不高的情形





条件传送方式

# 1.5 CPU与接口数据交换方式

## 2. 无条件传送方式

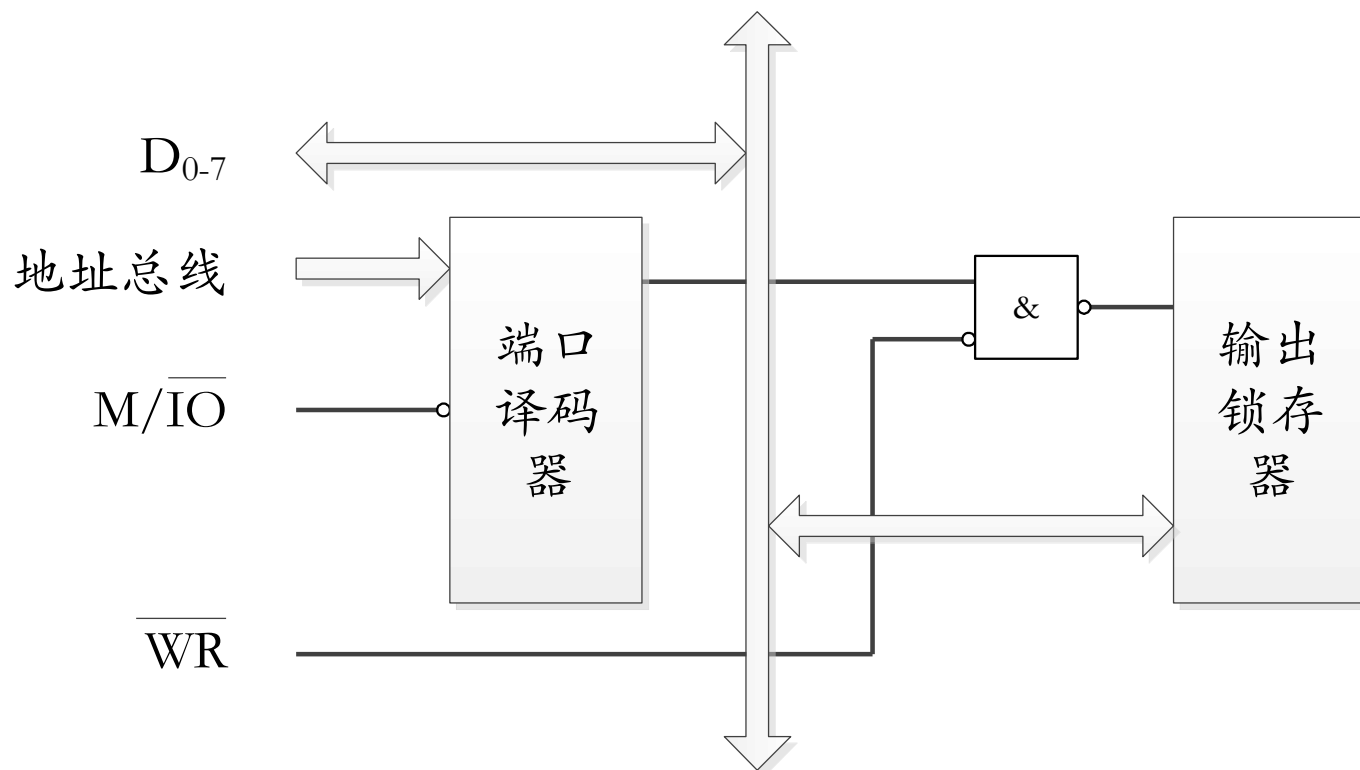
- 认为被控对象外部设备总是处于准备好的状态，CPU不用查状态就可以直接读/写数据，作为一种特殊的查询方式。

例如，

电机的启/停

继电器的吸合/释放控制

定时启动的数据采集（CPU启动数据采集的时间间隔不小于A/D转换器的转换时间）



无条件传送方式

# 1.5 CPU与接口数据交换方式

## 3. 中断方式

- 在CPU与外设之间设置中断控制器

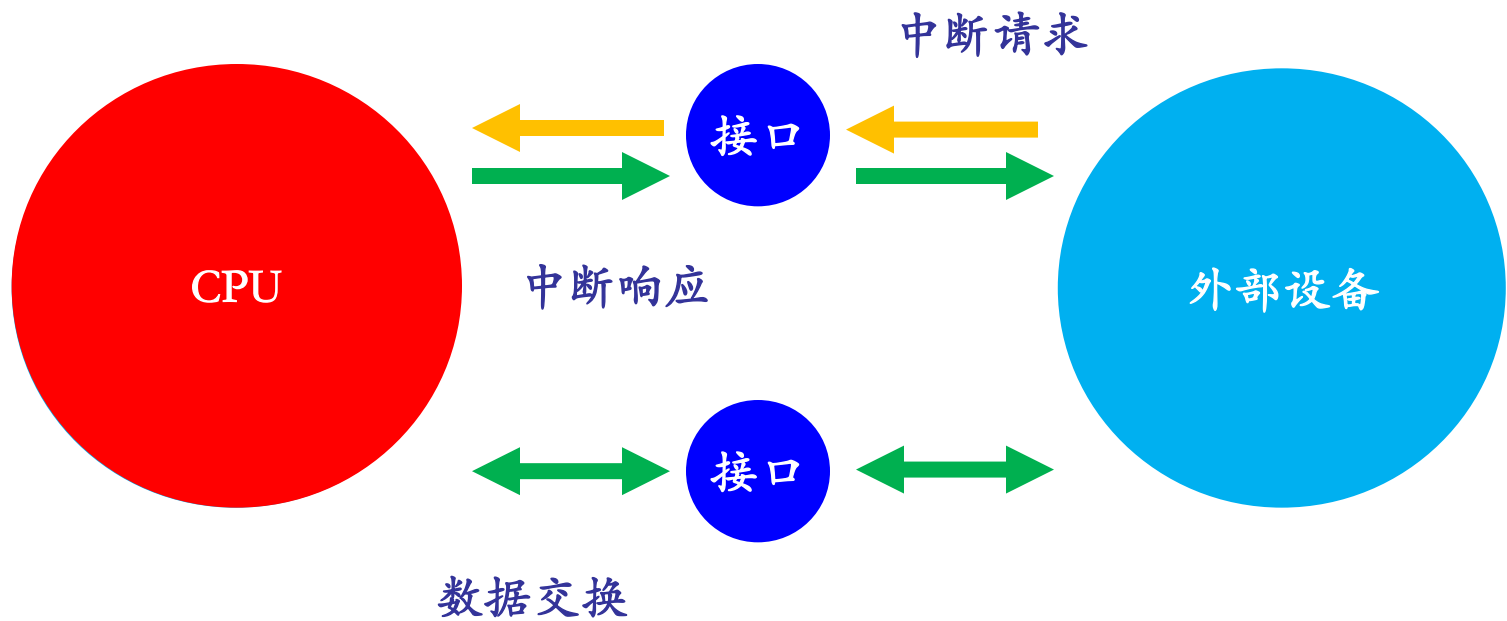
CPU响应中断申请，停止当前程序，保留当前程序指令地址和当前标志，转向执行中断子程序

### 特点

- 可使CPU和外设并行工作，提高了CPU的效率

### 适用场合

- 适用于CPU的任务比较忙（多个外设与CPU交换数据），传送速度不太高的系统，尤其适合实时控制及紧急事件的处理



中断传送方式

# 1.5 CPU与接口数据交换方式

## 4. DMA方式

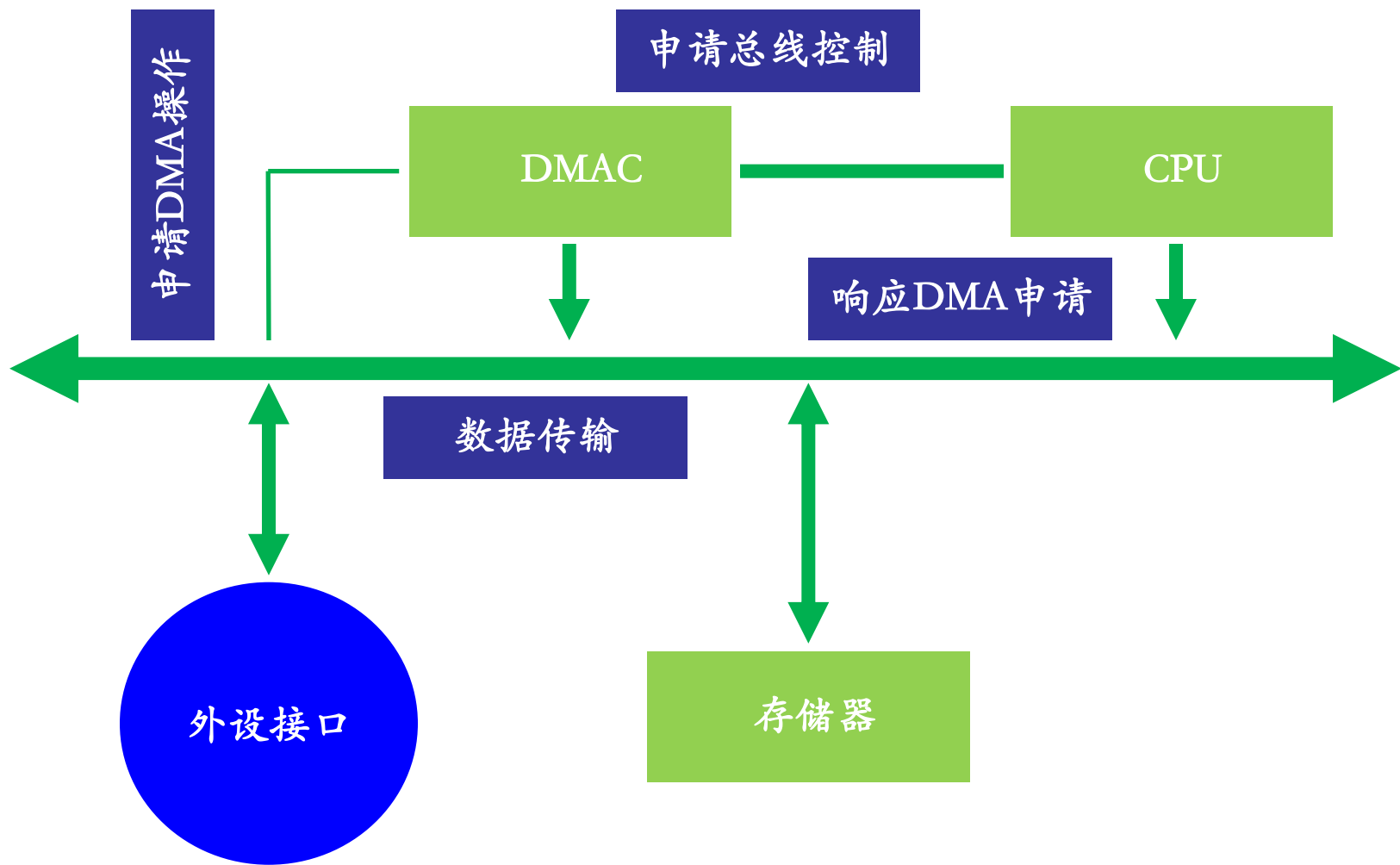
- 条件传送，无条件传送，中断传送都是CPU执行IN指令，OUT指令来实现的，不适用于大批量的数据高速传送
- DMA传送是在DMA控制器的控制下，**外设与内存，内存与内存，外设与外设**直接传送，这就减轻了CPU的负担，提高了传输速率
- CPU不参加数据的输入输出，DMA实现数据存取，地址刷新、计数、检测传送结束

### 特点

- 速度快，数据量大，电路结构复杂，硬件开销大

### 适用场合

- 适用于高速大批量的数据传送



DMA传送方式

# 1.6 分析和设计接口电路的基本方法

## 分析接口两侧的情况

- 微机系统一侧
  - 需要清楚总线的类型，CPU的类型
  - 系统总线（数据总线、地址总线）的宽度
  - 控制总线的逻辑定义、时序关系的特点及要求
- 外设一侧
  - 分析重点应放在搞清有关外设的工作原理，性能特点（编写接口程序的依据）
  - 所连接的信号线的定义和时序（外特性）



## 1.6 分析和设计接口电路的基本方法

- 软、硬件结合

以硬件为基础，软件和硬件相结合是设计接口电路的基本方法

# 1.6 分析和设计接口电路的基本方法

## 硬件设计方法

### 1. 合理选用接口芯片

- 利用现有的各种用途的通用或专用的可编程大规模集成电路接口芯片，并结合少量的中、小规模集成电路进行接口设计
- 利用FPGA/CPLD等各类可编程逻辑芯片，并借助VerilogHDL/VHDL或众多的EDA工具进行接口设计
- 采用传统的中、小规模的标准TTL、CMOS系列集成电路器件及传统的数字逻辑系统的设计方法进行接口设计

# 1.6 分析和设计接口电路的基本方法

## 硬件设计方法

### 2. 有针对性的设计附加电路

- 逻辑关系、电平转换、时序配合、驱动能力
- 反相器、三极管、与/或门、缓冲器、锁存器、驱动器芯片

# 1.6 分析和设计接口电路的基本方法

## 软件设计方法

1. 采用汇编语言（或高级语言C/C++）直接对硬件编程
  - 接口芯片和被控对象的外特性
  - 接口芯片的编程命令
2. 采用DOS系统功能调用和BIOS功能调用
  - 涉及到使用系统资源（键盘、显示器、打印机、串口等）
3. 采用Windows/Linux的提供的接口函数

# 1.7 接口技术的发展趋势

1. 早期的计算机系统，没有独立的接口电路，对外设的控制与管理由CPU直接操作
  - 主机效率低
  - 外设种类多（信息格式、电平高低、逻辑关系），主机对外设的控制电路复杂
  - 固定连接
2. 简单的接口电路，独立的接口和设备控制器
  - 简化CPU对外设的管理和控制
  - CPU和外设按各自规律发展（标准化、系列化）

## 1.7 接口技术的发展趋势

3. 微机接口向智能化、标准化、多功能化、高度集成化方向发展
  - 各种高性能接口**标准**的不断推出和使用（向下兼容）
  - 超大规模接口集成芯片
  - 接口控制软件固化技术

# 1.7 接口技术的发展趋势

- 固定式简单接口
- 可编程复杂接口
- 功能强大的智能接口

# 1.7 接口技术的发展趋势

## 关于接口技术的几点说明

- 在工作速度（数据传输速率）、数据宽度、芯片集成度等方面与CPU的发展存在巨大差距
  - 新型外设
  - 先进的总线技术
  - 新的接口标准、芯片组
- CPU和外部设备及接口在计算机系统中的作用不同，目的在于满足实际应用的需要
- 集成度的提高和物理结构的改变，必须保持逻辑兼容