

第六章 总线和输入输出（I/O）系统

6.1 本章大纲要求与核心考点

6.1.1 大纲要求

- (一) 总线概述
 - 1. 总线的基本概念
 - 2. 总线的组成及性能指标
 - 3. 总线事务和定时
- (二) I/O 接口（I/O 控制器）
 - 1. I/O 接口的功能和基本结构
 - 2. I/O 端口及其编址
- (三) I/O 方式
 - 1. 程序查询方式
 - 2. 程序中断方式
 - 3. DMA 方式：DMA 控制器的组成，DMA 传送过程。

6.1.2 核心考点

本章内容相对比较简单，以概念和原理为主，一般以单项选择题的形式出现；综合应用题以 I/O 方式的考察为主，主要是程序中断方式和 DMA 方式，涉及与第五章内容的结合。总线的性能指标可能会涉及到一些计算（以总线带宽为主），I/O 设备和 I/O 接口的考察则可能会与第三章内容相结合。

需要重点掌握的内容包括：

- 总线的基本概念、组成、主要性能指标。
- 总线事务的定义，突发传送方式，总线定时方式，异步定时方式的 3 种类型。
- I/O 接口的功能和结构，I/O 端口两种编址方式的原理及各自的优缺点。
- 程序查询方式的概念、原理、特点。
- 程序中断方式的概念、原理、特点。
- DMA 方式的概念、原理、特点，DMA 方式的传送方式和传送过程，与中断方式的比较。

6.1.3 真题分布

考点	考查次数	
	单项选择题	综合应用题
程序中断方式	15	3
总线概念和常见总线标准	9	2
外部设备和 I/O 接口	7	1
总线的性能指标	6	1
DMA 方式	3	3
程序查询方式	1	1

6.2 总线

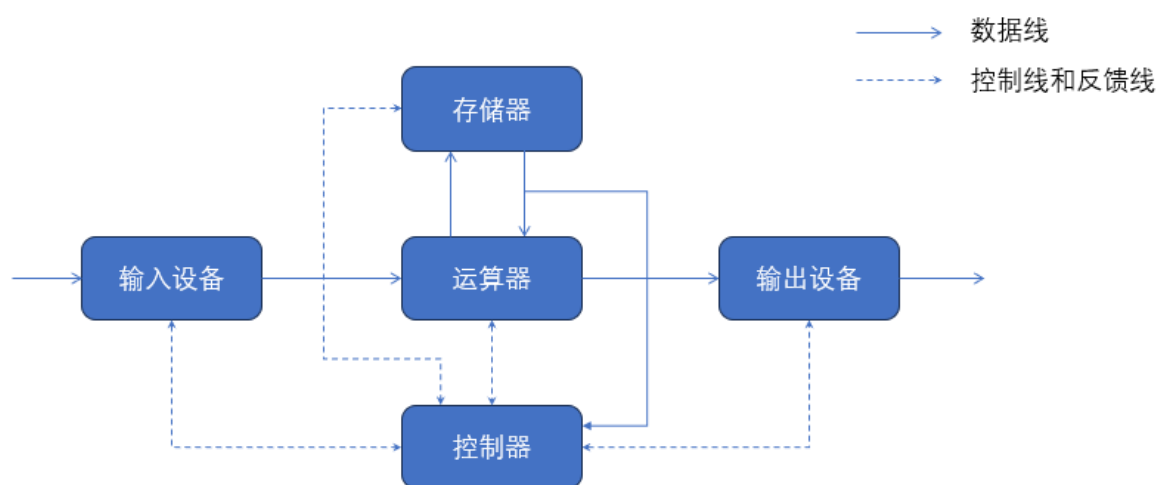
6.2.1 总线基本概念

1. 分散连接和总线连接

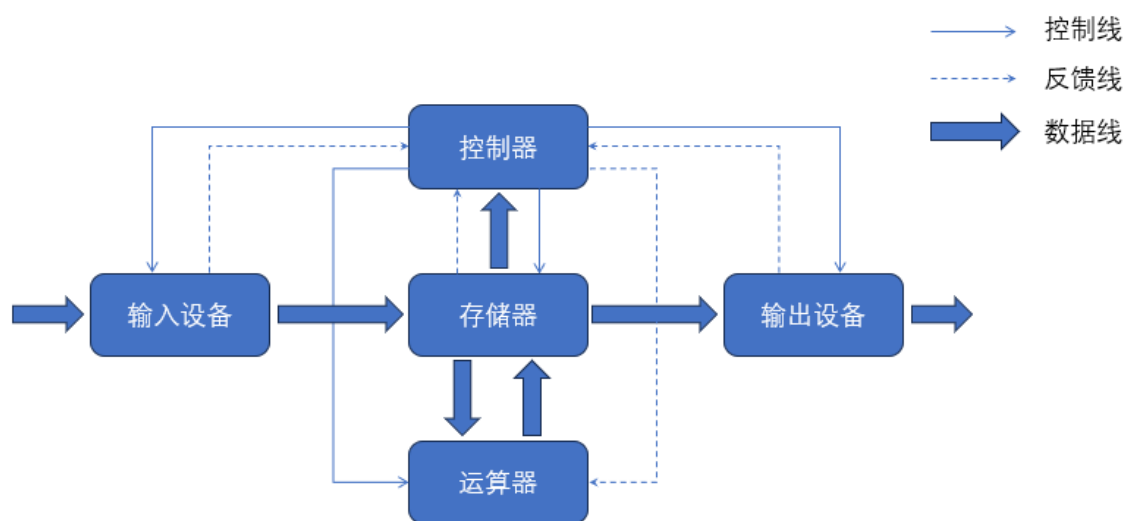
计算机系统的五大部件之间的互连方式有两种，一种是各部件之间使用单独的连线，称为 **分散连接**；

另一种则是将各部件连到一组公共信息传输线上，称为 **总线连接**。

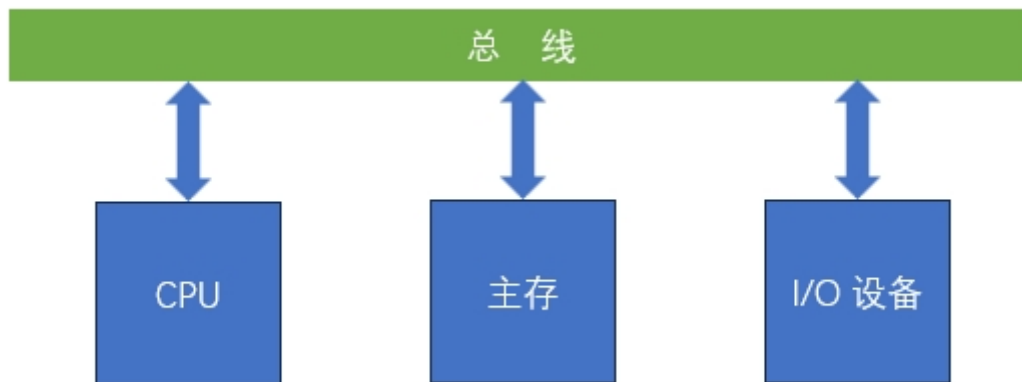
早期的计算机大多数用分散连接方式，以运算器为中心的结构，其内部连线十分复杂，尤其是当 I/O 与存储器交换信息时，都需经过运算器，致使运算器停止运算，严重影响了 CPU 的工作效率。



后来，虽然改进为以存储器为中心的分散连接结构，I/O 与主存交换信息可以不经运算器，又采用了中断、DMA 等技术，使 CPU 工作效率得到很大的提高，但是仍无法解决 I/O 设备与主机之间连接的灵活性。



随着计算机应用领域的不断扩大，I/O 设备的种类和数量也越来越多，采用分散连接方式很难实现随时增添或减撤设备，而总线连接方式则可以解决这个问题。



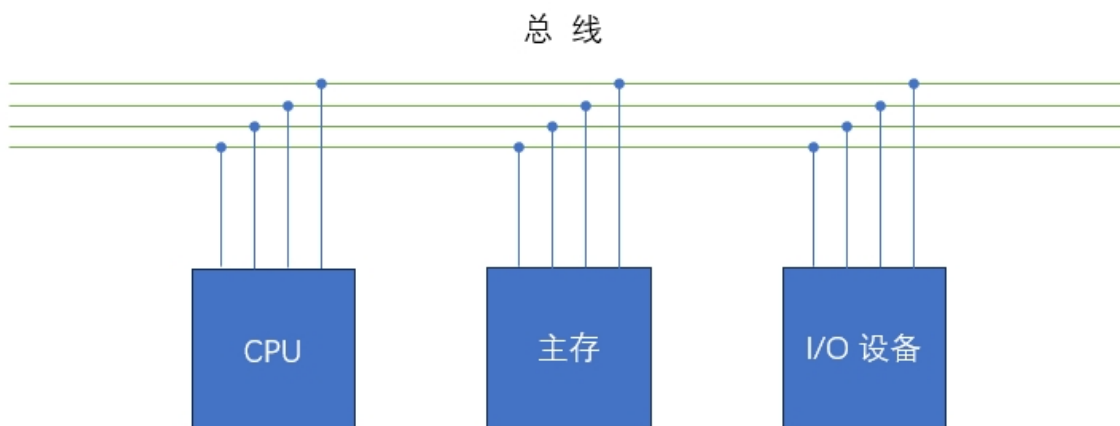
总线是连接多个部件的信息传输线，是各部件共享的传输介质。

2. 总线的特点和组成

当多个部件连接到总线上时，如果两个或两个以上部件同时向总线发送信息，就会导致信号冲突，传输无效。因此，在某一时刻，只允许有一个部件向总线发送信息，而多个部件可以同时从总线上接收相同的信息。

所以，**分时** 和 **共享** 是总线的两个特点。

总线实际上是由许多传输线或通路组成，每条线都可以通过电信号传递一位二进制代码；对于一串二进制代码，可以在一段时间内一位一位传输完成。多条传输线，则可以同时传输多位二进制代码。



6.2.2 总线的分类

总线可以应用在各种场景，从不同角度可以有不同的分类方法。

- 按数据传送方式：**并行传输总线** 和 **串行传输总线**。
- 按传输数据宽度：在并行传输总线中，又可按宽度分为 8 位、16 位、32 位、64 位等传输总线。
- 按使用范围：**计算机总线**、**测控总线**、**网络通信总线** 等。

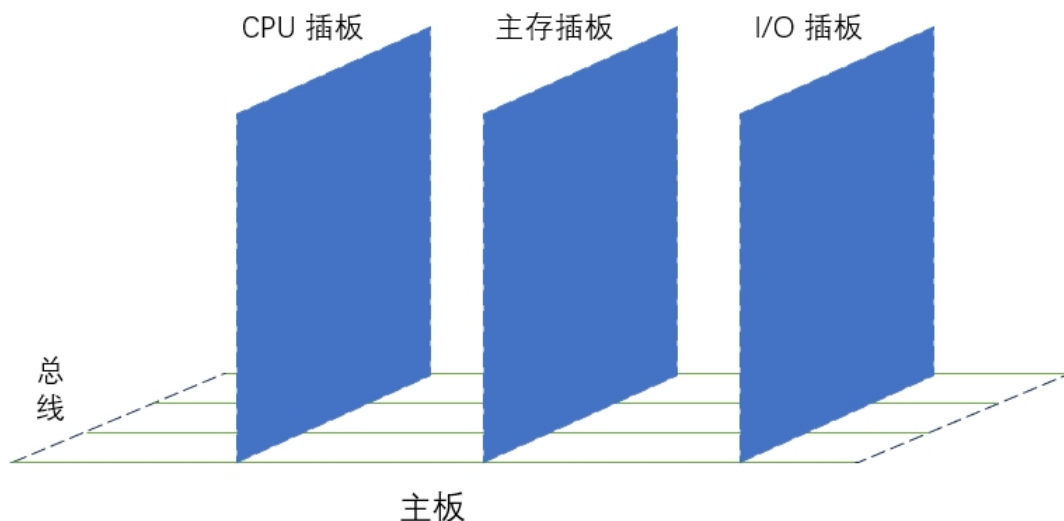
而更加常用的一种分类方式，则是按连接部件的不同，可以分为三类：**片内总线**、**系统总线** 和 **通信总线**。

1. 片内总线

片内总线是指芯片内部的总线，如在 CPU 芯片内部，寄存器与寄存器之间、寄存器与运算逻辑单元 ALU 之间，都是由片内总线连接的。

2. 系统总线

系统总线是指 CPU、主存、I/O 设备各大部件之间的 **信息传输线**。由于这些部件通常都安放在主板或各个插件板（插卡）上，故又称 **板级总线** 或 **板间总线**。从物理上看，就是由许多导线直接印制在电路板上，延伸到各个部件。



在系统总线中，按系统总线传输信息的不同，又可分为三类：**数据总线**、**地址总线** 和 **控制总线**。

(1) 数据总线

数据总线用来传输各功能部件之间的 **数据信息**，它是双向传输总线，其位数与机器字长、存储字长有关，一般为 8 位、16 位或 32 位。

数据总线的位数称为 **数据总线宽度**，它是衡量系统性能的一个重要参数。如果数据总线的宽度为 8 位，指令字长为 16 位，那么，CPU 在取指阶段必须两次访问主存。

(2) 地址总线

地址总线主要用来指出数据总线上的源数据或目的数据在 **主存存储单元的地址** 或 **I/O 设备的地址**。

地址总线上的二进制码就是一个 **地址**，由 CPU 输出，单向传输。地址线的位数与存储单元的个数有关。

(3) 控制总线

控制总线是用来发出各种 **控制信号** 的传输线。

由于数据总线、地址总线都是被挂在总线上的所有部件共享的，要使各部件能在不同时刻占有总线使用权，就需要依靠控制总线来完成分配和调度。

通常对任一控制线而言，它的传输是单向的。但对于控制总线总体来说，又可认为是双向的。例如，CPU 可以向主存、I/O 设备发出读/写控制信号，也可以接收外部设备发来的中断请求。

3. 通信总线

这类总线用于计算机系统之间，或计算机系统与其他系统（如控制仪表、移动通信等）之间的通信。

通信总线通常按传输方式分为两种：**串行通信** 和 **并行通信**。

- 串行通信：是指数据在单条 1 位宽的传输线上，一位一位地按顺序分时传送。
- 并行通信：是指数据在多条并行的 1 位宽传输线上，同时由源传送到目的地。

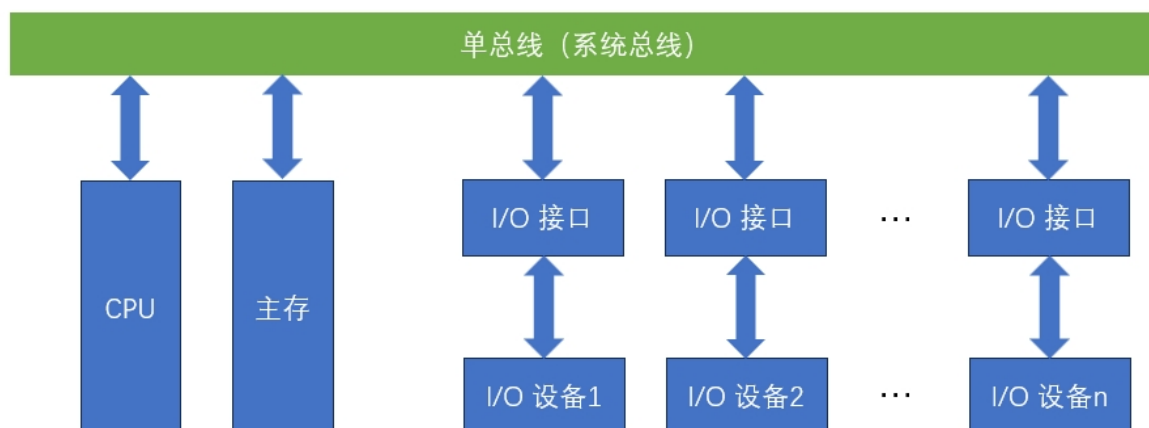
并行通信适宜于近距离的数据传输，通常小于30m；串行通信适宜于远距离传送，可以从几米达数千公里。通信总线的数据传输速率一般都与距离成反比。

6.2.3 总线结构

总线结构通常可以分为 **单总线结构** 和 **多总线结构**。

1. 单总线结构

单总线结构将 CPU、主存、I/O 设备（通过I/O接口）都挂在一组总线上，允许 I/O 设备之间、I/O 设备与 CPU之间或 I/O 设备与主存之间直接交换信息。

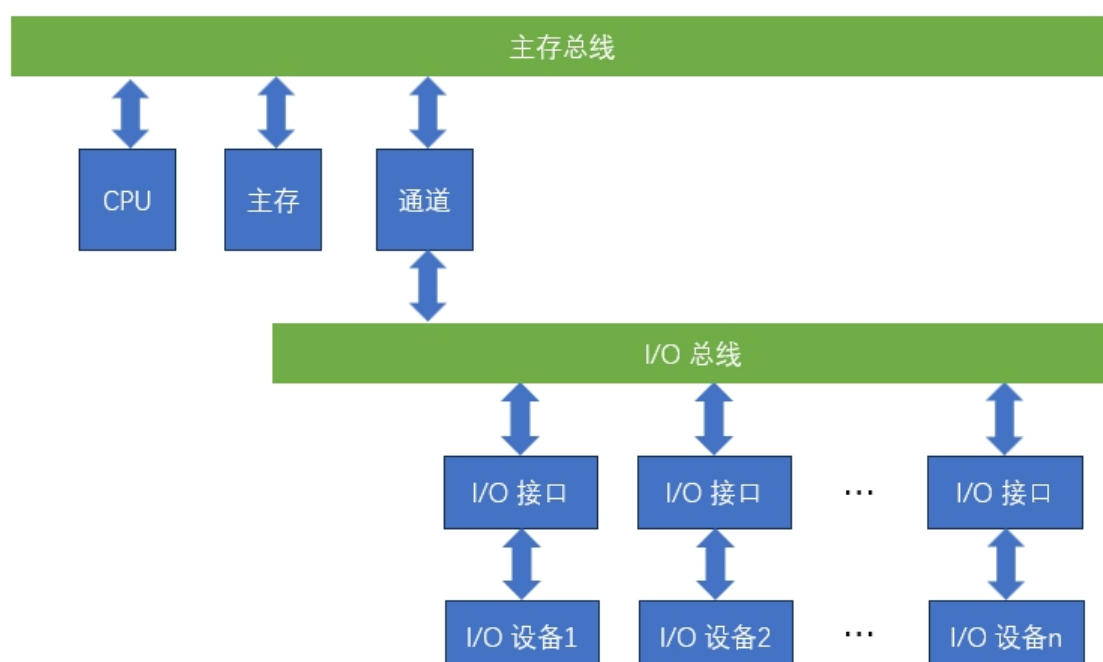


这种结构简单，也便于扩充，但所有的传送都通过这组共享总线，因此极易形成计算机系统的瓶颈。

它也不允许两个以上的部件在同一时刻向总线传输信息，这就必然会影响系统工作效率的提高。这类总线多数被小型计算机或微型计算机所采用。

2. 双总线结构

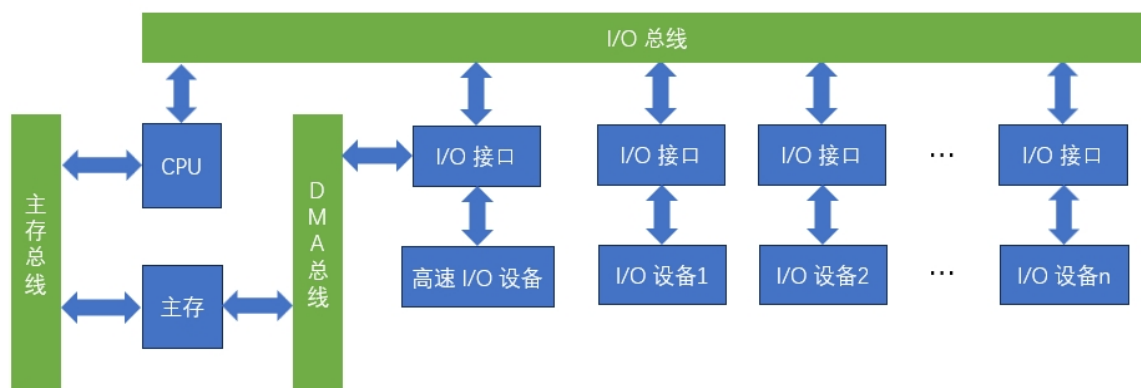
双总线结构的特点是，将速度较低的 I/O 设备从单总线上分离出来，形成 **主存总线** 与 **I/O总线** 分开的结构。



图中通道是一个具有特殊功能的处理器，CPU 将一部分功能下放给通道，让通道来统一管理 I/O 设备；这样就可以不经过 CPU，直接完成外部设备与主存储器之间的数据传送，其系统的吞吐能力可以相当大。这种结构大多用于大、中型计算机系统。

3. 三总线结构

如果继续将速率不同的 I/O 设备进行分类，然后将它们连接到不同的通道上，就可以进一步提升计算机的效率。这样就发展出了三总线结构。



主存总线用于 CPU 与主存之间的传输；I/O 总线供 CPU 与各类 I/O 设备之间传递信息；DMA 总线用于高速 I/O 设备（磁盘、磁带等）与主存之间直接交换信息。在三总线结构中，任一时刻只能使用一种总线。主存总线与 DMA 总线不能同时对主存进行存取，I/O 总线只有在 CPU 执行 I/O 指令时才能用到。

如果将高速缓存 Cache 的数据交换也考虑进来，又可以构建出不同的三总线结构，甚至可以进一步发展出四总线结构。

6.2.4 总线的性能指标

1. 总线宽度

通常是指数据总线的根数，用 bit（位）表示，如 8 位、16 位、32 位、64 位（即 8 根、16 根、32 根、64 根数据总线）。

2. 总线带宽

总线带宽代表了总线的 **数据传输速率**，即单位时间内总线上传输数据的位数，通常用每秒传输信息的字节数来衡量，单位用 MBps（兆字节每秒）表示。

例如，总线工作频率为 33 MHz，总线宽度为 32 位，则总线带宽为

$$33 \times (32 \div 8) = 132 \text{ MBps}$$

3. 信号线数

地址总线、数据总线和控制总线三种总线数的总和。

4. 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**。

5. 总线复用

一条信号线上分时传送两种信号。

例如，通常地址总线与数据总线在物理上是分开的两种总线，地址总线传输地址码，数据总线传输数据信息。为了提高总线的利用率，可以让地址总线和数据总线共用一组物理线路，在这组物理线路上分时传输地址信号和数据信号，即为总线的多路复用。

6. 总线控制方式

包括突发工作、自动配置、仲裁方式、逻辑方式、计数方式等。

其中，突发传输方式也称为 **猝发传输**，指对于在主存中连续存放的数据，访问时只需要给出一个首地址，耗费一个时钟周期；然后每个数据传送各占一个时钟周期，不必再给出地址就可以传输多个连续的数据。这样就可以大大提升数据传输速率。

6.2.5 总线标准

总线标准，就是系统与各模块、以及各模块之间进行连接的一个标准界面。

这个界面对两端的模块来说，都是透明的，只需要按照总线标准完成自己的接口功能即可。这就使得计算机软硬件的接口设计更加通用和方便。

常见的总线标准有：

(1) ISA

ISA 总线也称 AT 总线，由 IBM 公司推出，用于早期的 8 位 / 16 位计算机。它使用独立的总线时钟，因此 CPU 的时钟频率可以更高，有利于 CPU 性能的提高。不过 ISA 总线没有总线仲裁的硬件逻辑，因此不支持多台主设备系统；而且数据传送必须通过 CPU 管理，传输效率不够高。

ISA 总线时钟频率为 8 MHz，最大传输率为 16 MBps，数据线 16 位，地址线 24 位。

(2) EISA

EISA 总线是一种在 ISA 基础上扩展的总线标准，与 ISA 完全兼容。EISA 总线从 CPU 中分离出了总线控制权，有效提升了传输效率；并且支持多个总线控制器和突发传输方式。

EISA 总线时钟频率为 8 MHz，最大传输率为 33 MBps，数据线 32 位，地址线 32 位。

(3) VESA (VL-BUS)

VESA 总线是由视频电子标准协会 (Video Electronic Standard Association, VESA) 提出的局部总线标准，由 CPU 总线演化而来。所谓 **局部总线**，是指不连接 CPU、而是为其它重要模块（比如显卡）提供直接相连的高速数据传输通道的总线。

VESA 总线时钟频率达 33 MHz，最大传输率达 133 MBps，数据线 32 位，并且可通过扩展槽扩展至 64 位。

(4) PCI

PCI (Peripheral Component Interconnect, 外围部件互连) 是 Intel 公司 90 年代提出的高性能局部总线标准，支持突发传输方式，兼容性好、支持即插即用。

PCI 总线提供了一个高速数据传输通道，自身采用 33 MHz 和 66 MHz 总线时钟，与 CPU 时钟频率无关，数据线为 32 位，且可扩展至 64 位。PCI 总线的数据传输速率为 132 MBps (33 MHz 时钟，32 位数据通路)，可以升级至 528 MBps (66 MHz 时钟，64 位数据通路)。

之后 Intel 又推出了 PCI - Express (PCIe) 总线，采用串行传输方式进一步提升了传输速率，最新版本的 PCIe 总线数据传输速率可达 8 GBps，已经成为如今应用最为广泛的总线标准。

(5) AGP

AGP (Accelerated Graphics Port, 加速图形端口) 是 Intel 公司推出的显卡专用局部总线标准, 基于 PCI 2.1 版规范扩充修改而成。

AGP 总线采用点对点通道方式, 以 66.7 MHz 的频率直接与主存相连, 以主存作为帧缓冲器实现高速存取。AGP 的数据线为 32 位, 最大数据传输速率为 266 MBps。此外 AGP 还采用了一种新技术, 能在一个时钟信号的上下沿双向传输数据, 这称为 “双激励” 技术, 这样 AGP 实际的传输频率就达到了 133 MHz, 最大传输速率增至 533 MBps。后来推出的 AGP 新版本, 数据传输速率又有进一步的提升, 可达 2.1 GBps。

不过随着 PCIe 的出现, AGP 总线的应用已经越来越少了。

(6) RS-232C

RS-232C 是由美国电子工业协会 EIA 推荐的一种串行通信总线标准, 它是应用于串行二进制数据交换的数据终端设备 (DTE) 和数据通信设备 (DCE) 之间的标准接口。

RS-232C 命名中, “RS” 指 “Recommended Standard” (推荐标准), 232 为标识号, C 表示修改次数。RS-232-C 总线标准设有 25 条信号线, 包括一个主通道和一个辅助通道, 一般用于 20m 以内的通信。

(7) USB

USB (Universal Serial Bus, 通用串行总线) 是 Compaq、Intel、IBM、Microsoft、DEC、NEC 和 Northern Telecom 七大公司联合推出的计算机串行接口总线标准, 1996 年发布了 USB 1.0 版本。



USB 基于通用连接技术, 做到了真正的即插即用, 具有很强的连接和扩展能力, 标准统一而且可以为外设提供 +5V 电源, 给用户使用提供了极大方便。标准 USB 低速传输距离为 5m, 通过 HUB 或中继器可达 30m。USB 1.0 的数据传输速率, 采用普通无屏蔽双绞线时为 1.5 Mbps, 采用带屏蔽双绞线可达 12 Mbps; USB 2.0 数据传输率可达 480 Mbps (60 MBps), USB 3.0 更是可以高达 5.0 Gbps (500 MBps)。

6.2.5 总线判优 (总线仲裁)

总线上连接的各种设备, 一些是对总线有控制权的, 称为 **主设备 (主模块)**; 另一类则对总线没有控制权, 只能响应从主设备发来的命令, 称为 **从设备 (从模块)**。

总线上信息的传递都是由主设备发起的；当某个主设备需要占用总线向另一个设备发送信息时，首先需要发出 **总线请求信号**。如果多个主设备同时发出请求、希望占用总线，就需要一个专门的 **总线控制器** 来做出判断，按照一定的优先级确定哪个主设备可以使用总线。这个过程就称为 **总线判优**，或者 **总线仲裁**。

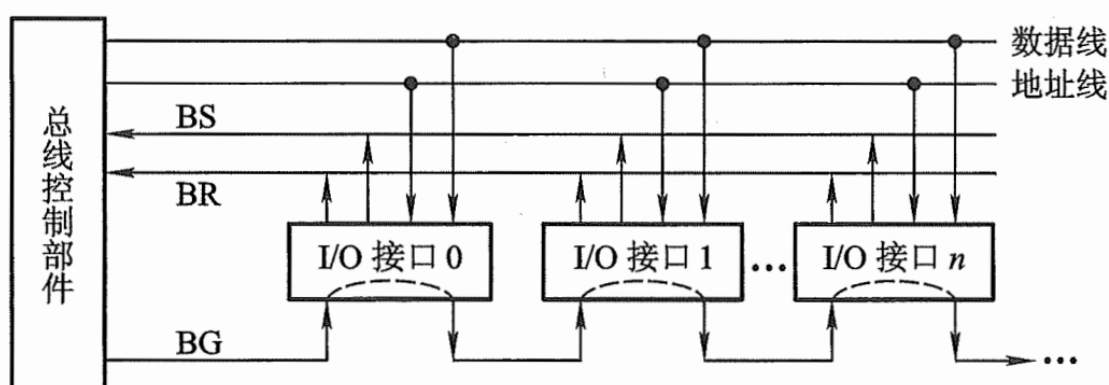
总线判优的实现可以分为两种形式：

- 集中式：将控制逻辑集中在一处（比如 CPU 中）；
- 分布式：将控制逻辑分散在与总线连接的各个部件或者设备上。

以集中式为例，常见的总线判优方式有以下三种。

1. 链式查询

顾名思义，链式查询的硬件连接方式就是将设备接口电路排成一条链，依次进行查询。原理跟中断判优的“链式排队器”类似。



上图中，控制总线中有 3 根线专门用于总线的控制：

- BS (Bus State, 总线状态)：为 1 时表示“总线忙”，即总线已被占用；为 0 时表示“总线空闲”；
- BR (Bus Request, 总线请求)：传递各设备接口电路发来的总线请求信号；
- BG (Bus Grant, 总线同意)：由总线控制器经过总线判优之后、发出的允许某个设备使用总线的信号。

BG 信号是按照链式排列顺序，串行地依次向下传递的。

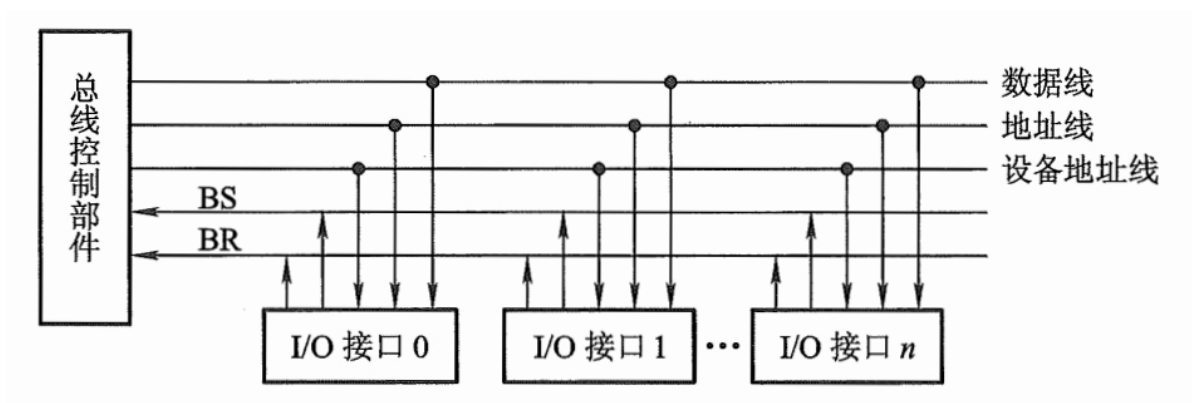
I/O 设备 0 的优先级最高，因此 BG 先到达 I/O 接口 0。如果这时接口 0 有总线请求，那么 BG 就不再向下传递，并且发出“总线忙”BS 信号，表示设备 0 获得了总线使用权。如果接口 0 没有总线请求，则 BG 信号继续传递至 I/O 接口 1，依次查询判断。

可见，链式查询中，设备距离总线控制器 **越近**，优先级就 **越高**。

这种方式实现简单，很容易实现扩展；但对电路故障很敏感，而且优先级低的设备可能很难获取总线使用权。

2. 计数器定时查询

计数器定时查询的基本思路是，为每个设备定一个 **设备地址**，总线控制器中设置一个 **计数器**，按照设备地址号依次计数；当计数器的值是某个设备的地址时，该设备的总线请求就可以被响应、从而获取总线的使用权。



跟链式查询相比，计数器定时查询增加了一组 **设备地址线**，省去了总线同意信号线 BG。

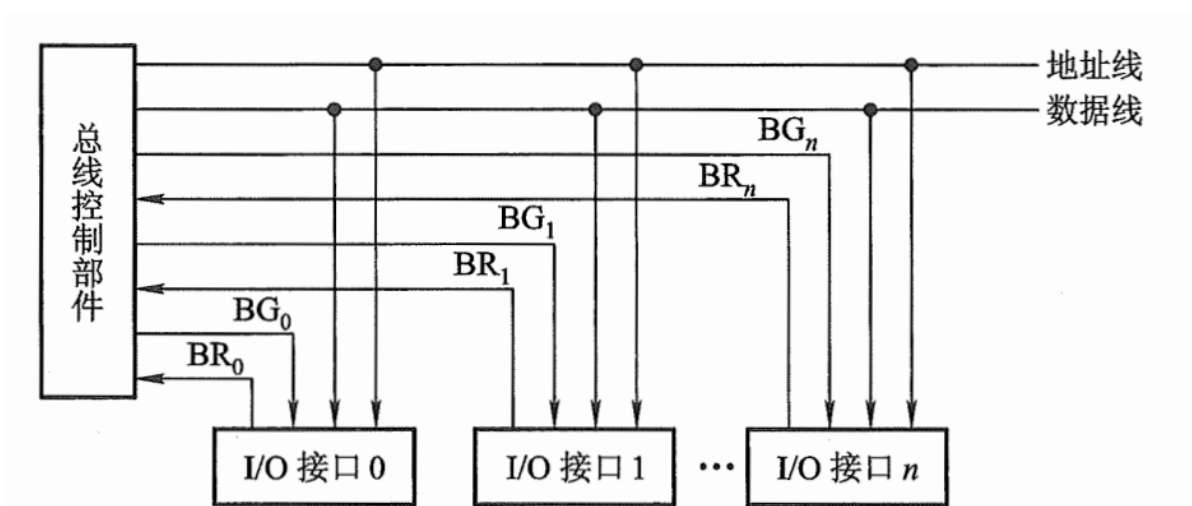
当总线控制部件接收到 BR 上的总线请求时，如果当前总线没有被占用（BS = 0），那么就让计数器开始计数；并通过设备地址线，发出一组地址信号。如果地址线选中的设备，发出了总线请求信号，那么就可以获得总线使用权，此时计数器停止计数。

- 如果我们设定计数器每次都从 " 0 " 开始计数，那么设备的优先级就是按照地址 0, 1, 2, ... 依次排列；
- 如果我们每次停止计数后不清零、而是继续循环计数，那就相当于所有设备可以轮流占用总线、优先级相同；
- 如果我们通过程序来设置计数器的初始值，还可以灵活改变优先级；

这种方式对电路故障没有那么敏感，不过增加了控制线的数量（设备地址线一般有多条），控制也比较复杂。

3. 独立请求方式

这种方式更加直接，每个设备独立发出总线请求；总线控制器内设置排队电路，根据优先级单独向每个设备发送总线同意信号。



上图中，每个设备都有一对总线请求线 $BR_{\sim i}$ 和总线同意线 $BG_{\sim i}$ 。当某个设备要使用总线时，就通过自己的请求线发出请求信号；总线控制器通过内部的排队电路进行判优，决定响应哪一个设备的请求，发出对应的同意信号。

这种方式响应速度快，优先次序也可以通过程序设置灵活改变，但控制线数量更多、总线控制更加复杂。

对于 n 个设备的系统，要确定总线使用权属于哪个设备，链式查询只用了 2 条线（BR、BG）；计数器定时查询则需要设备进行编址，大致需要 $\log_2 n$ 条设备地址线和 1 条 BR 线；而独立请求方式则需要 $2n$ 条线。

6.2.6 总线事务和总线周期

从一个设备发出总线请求、获取到总线使用权，到使用完成、将总线释放，这个完整的过程就是一次 **总线操作**，一般称为 **总线事务**。完成一次总线事务的时间，就称为 **总线周期**。

总线事务中包括一些具体操作，对应着总线周期中的不同阶段：

- **申请分配阶段**：主设备发出 **总线请求**，经过总线控制器中总线仲裁机构的 **判优**，决定将总线使用权交给某个设备（申请者）；
- **寻址阶段**：取得使用权的主设备，通过总线发出要访问的从设备地址和有关命令，启动从设备；
- **传输阶段**：主设备和从设备进行数据交换，数据从源模块发出，经数据总线流入目的模块；
- **结束阶段**：系统总线上撤除主设备的有关信息，让出总线的使用权。

6.2.7 总线通信控制（总线定时）

在总线周期中，核心阶段是数据的传输。由于涉及到两个不同模块，因此双方在交换数据时什么时候传输开始、什么时候传输结束，以及如何协调配合，应该有一个统一的规则。这被称为 **总线通信控制**；因为主要涉及的是时间上的控制协调，所以也叫 **总线定时**。

总线通信控制通常有四种方式：**同步通信**、**异步通信**、**半同步通信** 和 **分离式通信**。

1. 同步通信

通信双方由统一时钟信号来控制数据传送，这种方式称为 **同步通信**。

时钟信号通常由 CPU 的总线控制部件发出，送到总线上的所有部件；也可以由每个部件各自的时序发生器发出，但必须由总线控制部件发出的时钟信号对它们进行同步。

- 优点：规定明确、统一，模块间的配合简单一致；
- 缺点：主、从模块时间配合属于强制性“同步”，必须在限定时间内完成规定的要求；并且对所有从模块都用同一限时，这就势必造成，对各不相同速度的部件而言，必须按最慢速度的部件来设计公共时钟，严重影响总线的工作效率，也给设计带来了局限性，缺乏灵活性。

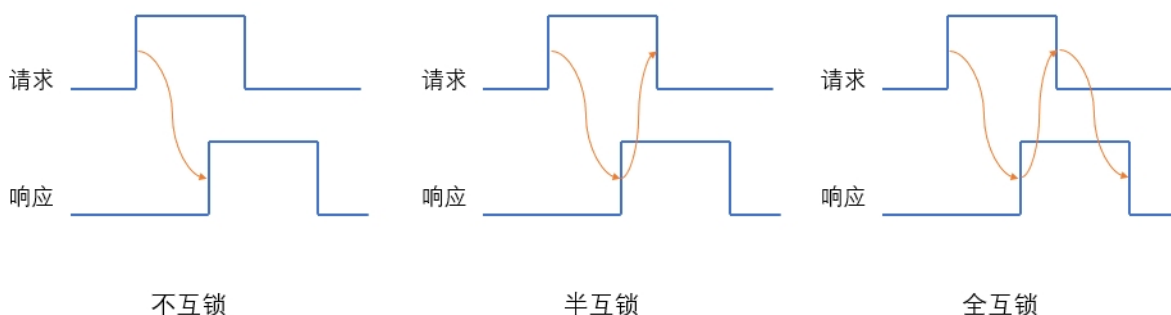
同步通信一般用于总线长度较短、各部件存取时间比较一致的场合。

2. 异步通信

异步通信则没有公共的时钟标准，不要求所有部件严格的统一操作时间，而是采用 **应答方式**（又称 **握手方式**），当主模块发出 **请求**（Request）信号时，一直等待从模块反馈回来“**响应**”（Acknowledge）信号后，才开始通信。

异步通信克服了同步通信的缺点，允许各模块速度的不一致性，给设计者充分的灵活性和选择余地。但要求主、从模块之间增加两条应答线（握手交互信号线），而且控制逻辑更加复杂，数据传输率也不及同步传输。

异步通信的应答方式，又可分为 **不互锁**、**半互锁** 和 **全互锁** 三种类型。



(1) 不互锁方式

主设备发出请求信号后，不必等待接到从设备的响应信号，而是经过一段时间便撤销其请求信号；从设备接到请求信号后，发出响应信号，并且经过一段时间自动撤销响应信号。

可见通信双方完全独立，没有互锁关系。

(2) 半互锁方式

主设备发出请求信号，必须等接到从设备的响应信号后，再撤销其请求信号，因此有互锁关系；

而从设备在接到请求信号后，发出响应信号，但不必等待主设备撤销请求信号，而是隔一段时间后自动撤销其响应信号：这两者之间没有互锁关系。

因此这种方式称为 **半互锁方式**。

(3) 全互锁方式

主设备发出请求信号，必须等从设备回答后，再撤销其请求信号；

同样，从设备发出回答信号，必须等主设备撤销请求信号后，再撤销其回答信号。

双方存在互锁关系，所以称为 **全互锁方式**。

3. 半同步通信

半同步通信是同步通信和异步通信的结合。

半同步通信既保留了同步通信的基本特点，如所有的地址、命令、数据信号的发出时间，都严格参照系统时钟的某个前沿开始，而接收方都采用系统时钟后沿时刻来进行判断识别；同时又像异步通信那样，允许不同速度的模块和谐地工作，为此需要增设一条“等待”（WAIT）响应信号线。

4. 分离式通信

分离式通信的基本思想是，将一个传输周期（或者总线周期）分解为两个子周期。

在第一个子周期，主设备 A 获取总线使用权后，将命令、地址及其它有关信息都发到系统总线上，经总线传输，由从设备 B 接收下来。这些信息的传输只占很短的时间，发送完后 A 立即放弃总线使用权，交给其它模块使用。

在第二个子周期，设备 B 接收到了 A 发来的所有信息，经过一系列内部操作，将 A 需要的数据准备好；此时便由 B 重新申请总线使用权，获准之后再进行数据传输。

这样，节省了等待 B 准备数据的时间，两个子周期中总线上都只有单向的信息流，两个设备都成为了主设备。

6.3 I/O 系统概述

在计算机中，除 CPU 和主存两大模块之外，第三个重要部分就是 **输入输出模块**，也叫做 **输入输出系统**，简称为 **I/O 系统**。

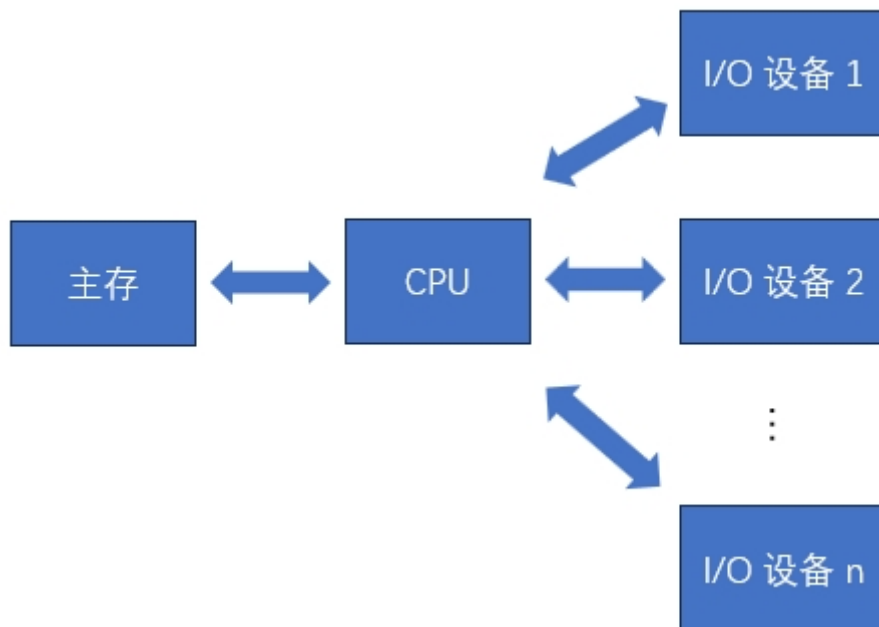
I/O 系统的主要功能，就是对不同类型数据信息的输入输出，进行高效的控制管理。

6.3.1 I/O 系统的发展

I/O 系统的结构，从简单到复杂，发展主要经历了四个阶段。

1. 早期阶段

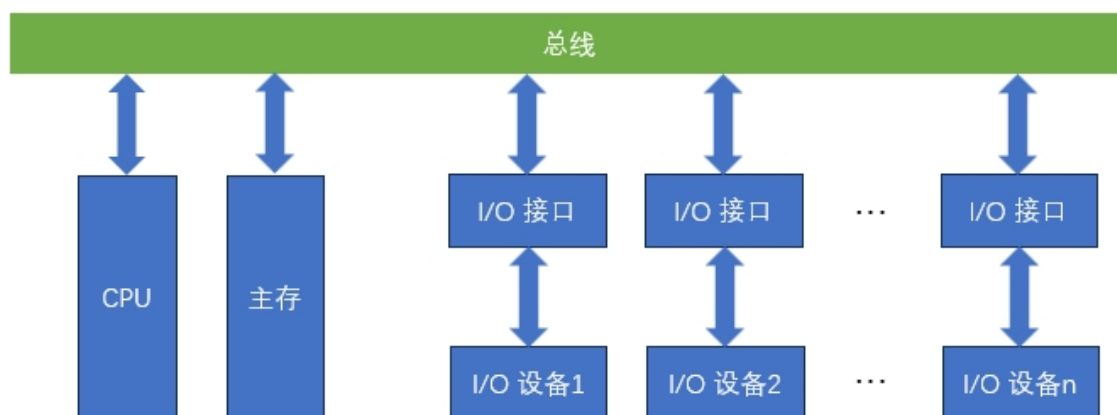
早期的输入输出设备比较少，计算机的结构是以 CPU 为中心的分散连接方式；所有 I/O 设备都直接连接到 CPU，与主存交换数据必须通过 CPU。



这样，每个 I/O 设备都必须配备一套独立的逻辑电路与 CPU 连接，线路复杂、难以维护和扩展；而且输入输出数据会打断 CPU 运行，效率比较低。

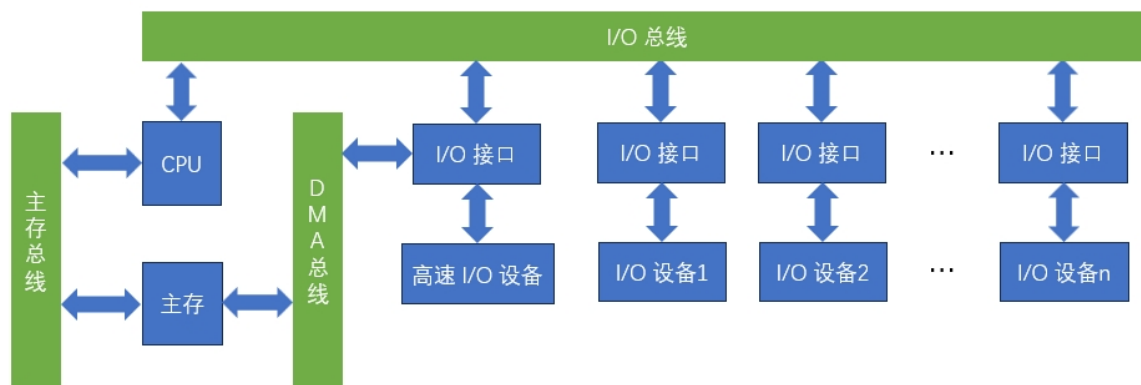
2. 接口模块和 DMA 阶段

之后计算机的连接方式发展出了总线结构，I/O 设备通过接口模块与总线相连，通过总线进行数据传输。



通常，在接口模块中都会设有 **数据通路** 和 **控制通路**。数据可以通过接口起到缓冲作用，也可以完成串并转换；控制通路则可以传送 CPU 发来的控制命令，或者向 CPU 发送来自 I/O 设备的反馈信号。许多接口还可以满足中断请求处理的要求，使 I/O 设备和 CPU 可以并行工作。

为了进一步提高 CPU 的效率，又出现了 **直接存储器存取** (Direct Memory Access, **DMA**) 技术。在 I/O 设备和主存之间增加一条专门的数据通路，让它们可以直接交换信息，而无需经过 CPU。

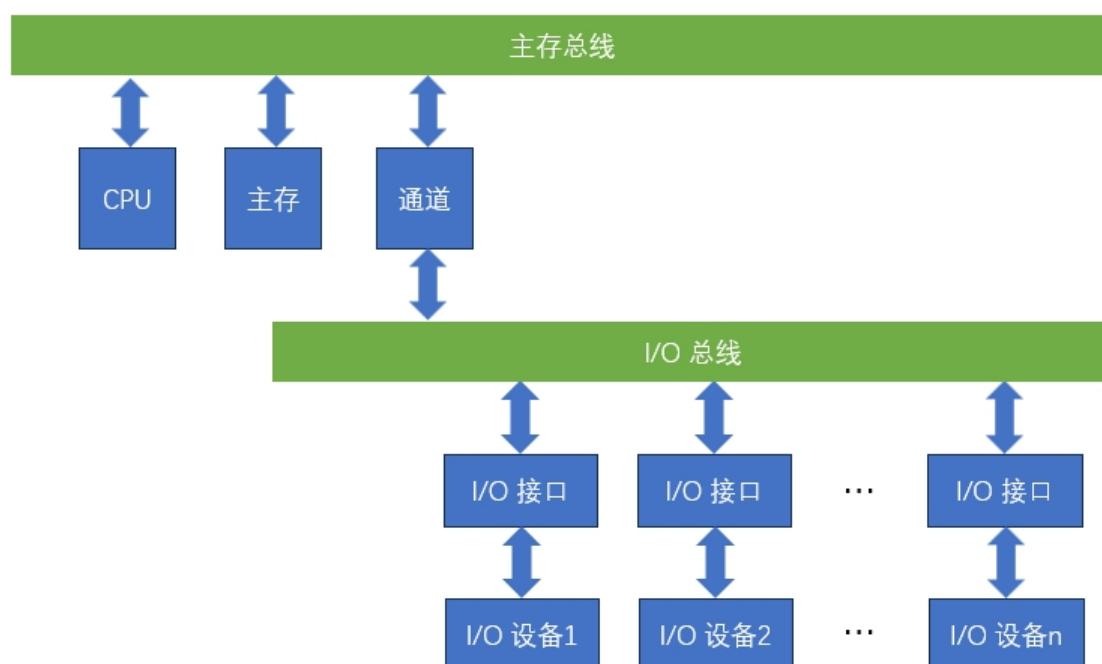


在现代的小型或微型计算机中，一般都会采用接口模块和 DMA 的接口来实现 I/O 设备和主存的数据交换。

3. 通道结构阶段

对于大中型计算机，I/O 设备数量庞大、数据传输非常频繁，采用 DMA 方式需要为每个设备都配置专用的 DMA 接口，CPU 也需要对众多的 DMA 接口进行管理；控制非常复杂、工作效率也会受到很大影响。

因此在大中型计算机中，采用 **I/O 通道** 的方式来进行数据交换。



通道 可以看作一种从属于 CPU 的专用处理器，专门负责管理 I/O 设备，实现主存和 I/O 设备之间的数据交换。采用通道结构的计算机，I/O 设备的数据传输就全部交给通道来处理，提高了 CPU 的资源利用率。

4. 处理机阶段

I/O 系统进一步发展，出现了 **I/O 处理机**。I/O 处理机基本独立于主机工作，又称为 **外围处理机**。具有 I/O 处理机的 I/O 系统与 CPU 的并行性更高，进一步解放了 CPU，提升了 CPU 的工作效率。

6.3.2 I/O 系统的组成

I/O 系统主要由两部分组成：**I/O 软件** 和 **I/O 硬件**。

1. I/O 软件

I/O 系统软件的主要功能包括：

- 将用户编写的程序输入主机；
- 将运算结果传送给用户；
- 协调 I/O 系统和主机的工作。

不同结构的 I/O 系统采用的软件技术不同。

生活中常见的计算机一般采用接口模块方式，指令系统中需要设置专门的 **I/O 指令**；调用 I/O 指令并配合系统软件中的管理程序，就可以实现 I/O 设备与主机的协调工作。

如果采用通道管理方式，除 I/O 指令外，还需要有 **通道指令** 和相应的操作系统。

(1) I/O 指令

I/O 指令也是一种机器指令，它的指令格式与普通的机器指令类似。



上面的 I/O 指令格式中，分为三个字段：

- **操作码**：指明当前是一条 I/O 指令；
- **命令码**：指明 I/O 设备的具体操作；
- **设备码**：用来对多台 I/O 设备进行区分和选择，相当于设备地址。

其中，命令码一般会包括以下几种情况：

- 将数据从 I/O 设备输入主机；
- 将数据从主机输出至 I/O 设备；
- 状态测试。检测 I/O 设备的状态（“忙”还是“准备就绪”），来确定接下来是否可以进行数据交换；
- 形成具体的操作命令。例如磁盘驱动器的磁头读扇区、写扇区、寻找磁道等等。

(2) 通道指令

通道指令是具有通道的 I/O 系统专门设置的指令，又称为 **通道控制字**（Channel Control Word，**CCW**），是通道用来执行 I/O 操作的指令。

这类指令位数一般会比较多，主要功能有：

- 指明要交换的一组数据在主存中的首地址；
- 指明要传送数据的字节数，或者在主存中的末地址；
- 指明所选设备的设备码，以及要完成的具体操作的命令码。

对于采用了通道结构的计算机，CPU 执行的 I/O 指令就只负责启、停 I/O 设备，查询通道和设备状态以及对通道进行操作控制；而具体的数据传输则交给通道来完成。

2. I/O 硬件

I/O 系统的硬件组成是多种多样的，不同结构的 I/O 系统硬件组成也会有所不同。

- 采用接口模块结构的 I/O 系统，硬件主要包括 **接口模块** 和 **I/O 设备** 两大部分。I/O 设备通过接口连接到主机；

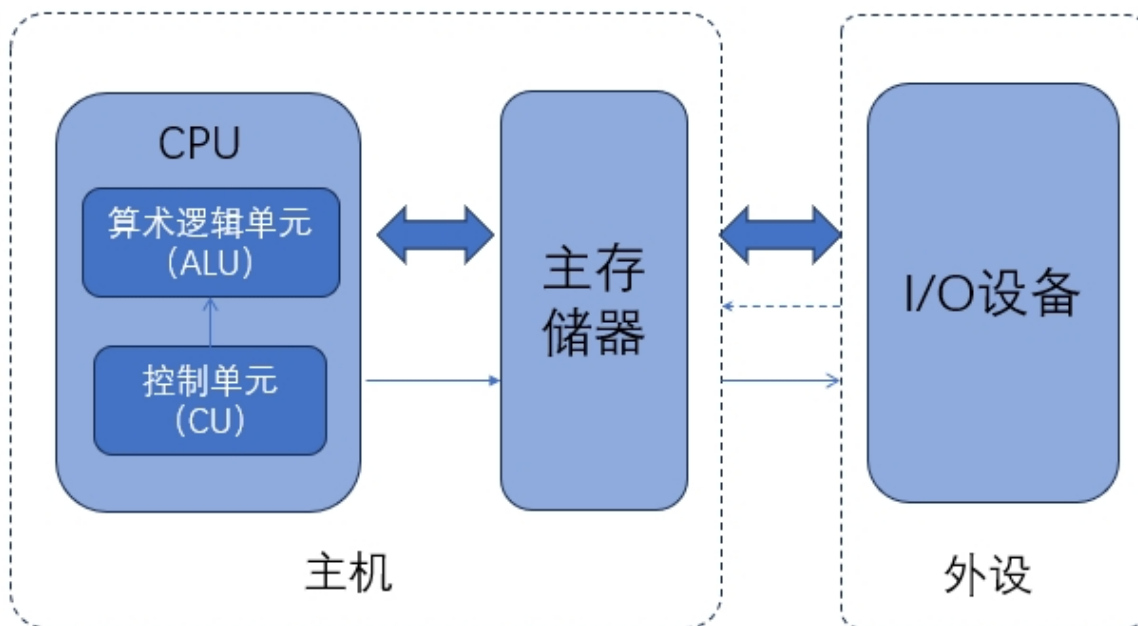
- 采用通道或处理机结构的 I/O 系统，硬件主要包括 **通道/处理机**、**设备控制器** 和 **I/O 设备**；一个通道/处理机可以和多个设备控制器相连，一个设备控制器又可以控制多台同类型的设备。

此外，连接各模块的数据通路（I/O 总线）也是 I/O 硬件的一部分。

6.3.3 I/O 设备

计算机硬件系统中，主要由三部分组成：CPU、主存储器和输入输出设备。输入输出设备和它们对应的接口模块，就构成了输入输出系统。

CPU 与主存合称 **主机**，主机之外的大部分硬件设备都可以称为 **I/O 设备**，统称为 **外部设备**，简称 **外设**。



1. I/O 设备分类

从应用场景和功能上看，I/O 设备可以分为三大类：

(1) 人机交互设备

实现用户（操作者）和计算机之间信息交流的设备，又可以分为两类：

- 输入设备：将人们熟悉的信息形式转换为机器能识别的信息形式，如键盘、鼠标、摄像头等。
- 输出设备：将机器运算结果转换为人们熟悉的信息形式，如打印机、显示器等。

(2) 信息存储设备

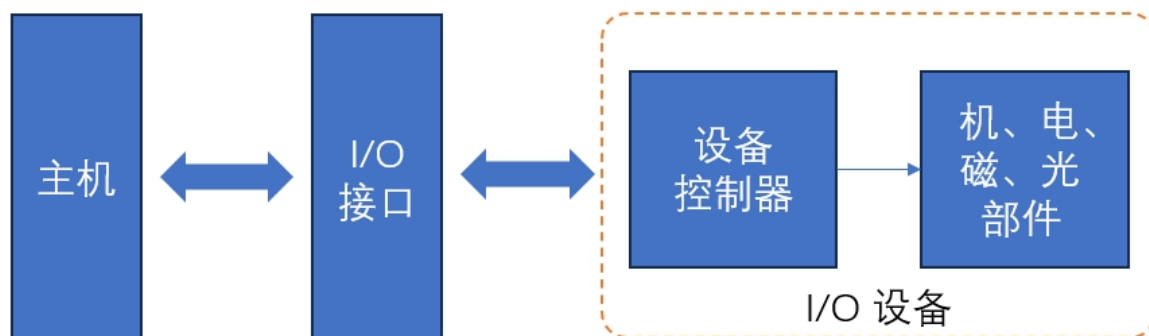
系统软件和各种有用信息，都需要进行存储保留，这就需要能够持久化保存的外部存储设备。这类存储设备一般可以作为计算机的辅助存储器，比如磁盘、光盘、固态硬盘等。

(3) 机-机通信设备

多台计算机之间、或者是计算机和其它系统之间如果需要进行通信，必须借助专门的通信设备。比如调制解调器（Modem），A/D、D/A 转换设备等。

2. I/O 设备的组成

I/O 设备的基本组成如下：



- 设备控制器用来控制 I/O 设备的具体动作，不同的 I/O 设备需要完成的控制功能也不同；
- 机、电、磁、光部件是 I/O 设备的工作部件，与具体功能有关；内部结构涉及到机、电、磁、光工作原理；
- 现代 I/O 设备一般还会通过 I/O 接口与主机相连。

6.4 I/O 接口 (I/O 控制器)

两个系统或两个部件之间的交接部分，一般就称为 **接口**。接口可以是硬件上两种设备间的连接电路；也可以是两个软件之间交互的逻辑边界。

主机与 I/O 设备之间，专门设置一套硬件电路、配合相应的软件控制，实现两者间的信息交互，这就是 **I/O 接口**。不同的 I/O 设备都有其相应的设备控制器，而它们往往都是通过 I/O 接口与主机取得联系的。

6.4.1 I/O 接口的功能

I/O 接口主要的功能如下：

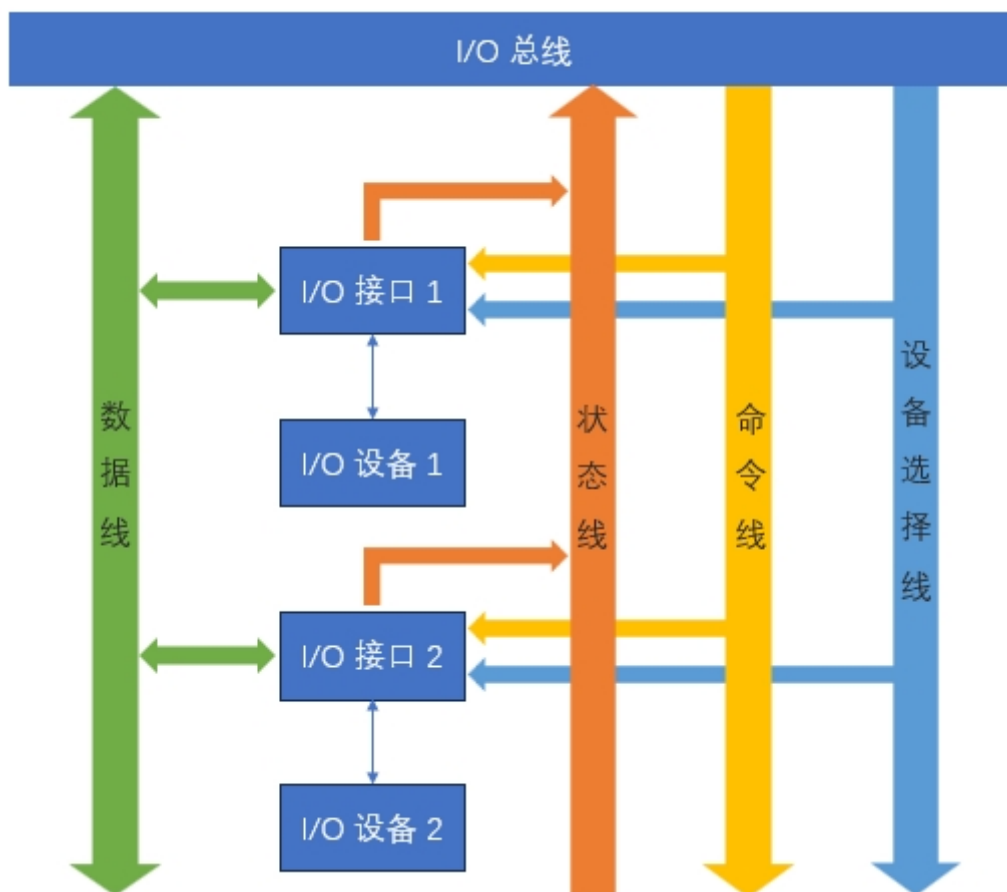
- 选址功能：根据 I/O 指令中的设备码进行设备选择；
- 数据缓冲：I/O 设备速度不一，与 CPU 相差比较大，需要进行数据的缓冲达到速度匹配、防止丢失数据；
- 串并转换：有些 I/O 设备采用串行传输方式，而 CPU 一般为并行传输，需要进行数据格式转换；
- 电平转换：I/O 设备的输入输出电平可能与 CPU 不同，需要进行电平转换；
- 传送控制命令：CPU 会向 I/O 设备发出各种控制信号，需要相应的传输通路；
- 反馈状态信息：I/O 设备需要将工作状态（比如“忙”“准备就绪”“错误”“中断请求”等）报告给 CPU，并且可能需要将状态信息进行保存，供 CPU 查询。

6.4.2 I/O 接口的基本结构

总线结构的计算机中，每一台 I/O 设备都是通过 I/O 接口挂到 I/O 总线上的。

1. 总线连接的数据通路

由于需要实现设备选择、数据缓冲、传送命令和状态等功能，总线中必须有相应的数据通路：



上图中的 I/O 总线，就包含了 **数据线**、**设备选择线**、**命令线** 和 **状态线**。

- **数据线**

数据线是 I/O 设备与主机之间传送数据的线路。

数据线的根数一般等于存储字长的位数，通常是 **双向的**。若采用单向数据总线，则必须用两组才能实现数据的输入和输出功能，而双向数据总线只需一组即可。

- **设备选择线**

设备选择线用来传送设备码，如果把设备码看做是地址号，那么设备选择线又可称为 **地址线**。

设备选择线的根数取决于 I/O 指令中设备码的位数，决定了能够连接设备的数量。设备选择线一般是一组，也可以有两组，其中一组用于主机向 I/O 设备发送设备码，另一组用于 I/O 设备向主机回送设备码。

- **命令线**

命令线主要用来传输 CPU 向设备发出的各种命令信号，比如启动、清除、屏蔽、读、写等。

命令线是一组单向总线，根数与命令信号的数量有关。

- **状态线**

状态线主要用来向主机报告 I/O 设备的状态信号，比如设备是否准备就绪、是否向 CPU 发出中断请求等。

状态线也是一组单向总线。

2. I/O 接口的基本组成

根据 I/O 接口的功能，以及总线结构中整体的数据通路，就可以推出接口应该具有的硬件配置。

(1) 选址功能

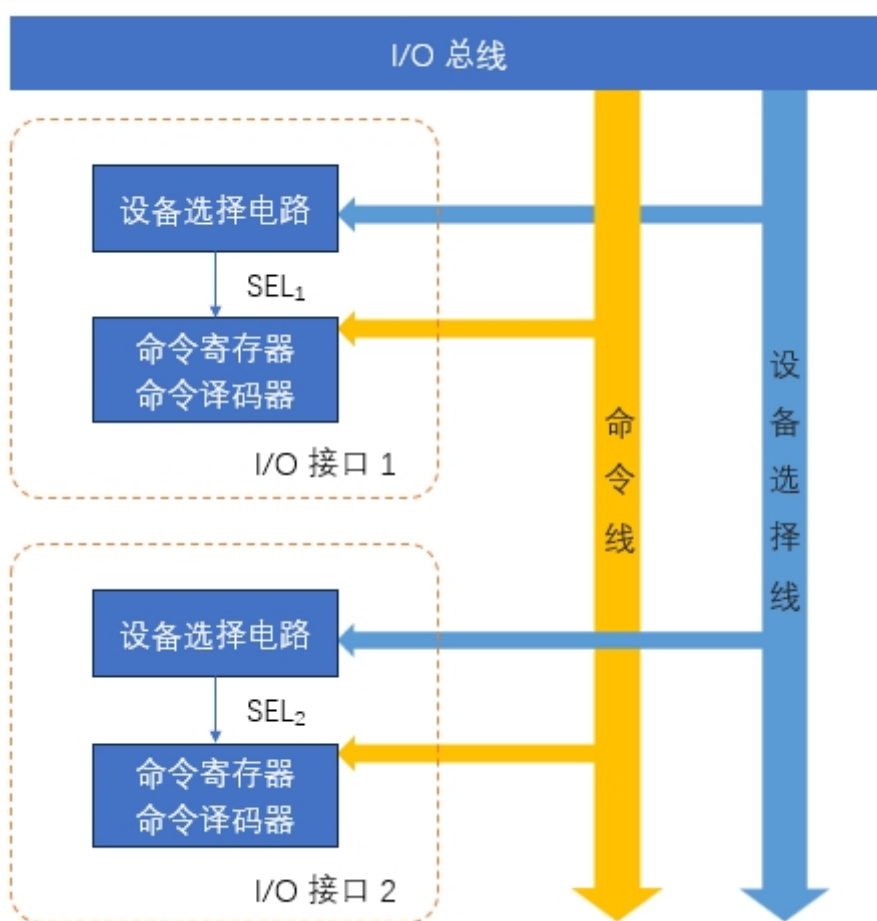
设备码通过设备选择线（地址线）送至所有设备的接口，因此需要每个接口都必须具有选址功能。

当设备选择线上的设备码与本设备码相符时，发出一个 **设备选中信号 SEL**，这种功能可通过接口内的设备选择电路来实现。

(2) 传送命令的功能

当 CPU 向 I/O 设备发出命令时，要求 I/O 设备能做出响应，因此通常在 I/O 接口中设有存放命令的 **命令寄存器** 以及 **命令译码器**。

命令寄存器用来存放 I/O 指令中的命令码，它受设备选中信号 SEL 控制。命令线和所有接口电路的命令寄存器相连，只有被选中设备的 SEL 信号有效，才会将命令线上的命令码存入命令寄存器。



(3) 传送数据的功能

接口处于主机与 I/O 设备之间，因此主机与 I/O 设备之间进行数据传输必须经过接口。这就要求接口中具有数据通路，完成数据传送。这种数据通路还应具有 **缓冲** 能力，也就是能将数据暂存在接口内。

接口中通常设有 **数据缓冲寄存器** (Data Buffer Register, DBR)，它用来暂存 I/O 设备与主机准备交换的信息，与 I/O 总线中的数据线是相连的。

(4) 反映 I/O 设备工作状态的功能

为了使 CPU 能及时了解 I/O 设备的工作状态，接口内必须设置一些反映设备工作状态的触发器。

比如，可以用 **完成触发器 D** 和 **工作触发器 B** 来标志设备所处的状态。

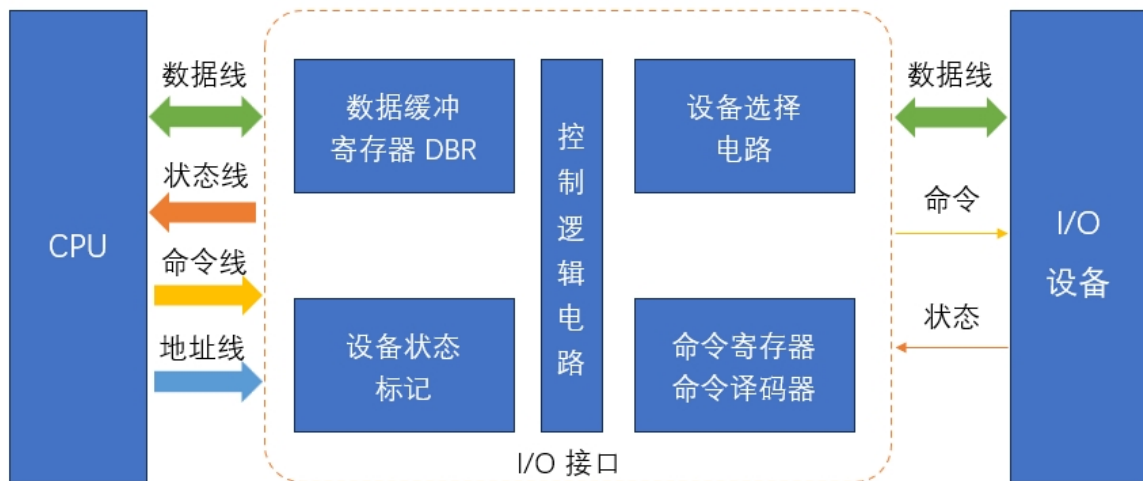
- 当 $D = 1, B = 0$ 时，表示 I/O 设备已经准备就绪；
- 当 $D = 0, B = 1$ 时，表示 I/O 设备正处于工作状态；
- 当 $D = 0, B = 0$ 时，表示 I/O 设备处于暂停状态。

由于现代计算机系统中大多采用中断技术，因此接口电路中一般还设有 **中断请求触发器 INTR**，当为“1”时，表示该设备向 CPU 发出中断请求；

接口内还有 **中断屏蔽触发器 MASK**，它与中断请求触发器配合使用，完成设备的屏蔽功能。

所有的状态标志触发器都与 I/O 总线中的状态线相连。此外，不同的 I/O 设备的接口电路中还可根据需要增设一些其他状态标志触发器。

这样就可以得到 I/O 接口的基本组成：



目前大多数 I/O 设备所共用的电路都制作在一个芯片内，作为 **通用接口芯片**。另一些 I/O 设备专用的电路，制作在 I/O 设备的 **设备控制器** 中。

6.4.3 I/O 端口及其编址

需要注意区分“**接口**”（Interface）和“**端口**”（Port）的概念：

端口 指的是接口电路中的一些寄存器，这些寄存器用来存放数据信息、控制信息和状态信息，相应的端口就称为 **数据端口**、**控制端口** 和 **状态端口**。

CPU 执行 I/O 指令中的输入操作时，从端口读入信息；执行输出操作时，将寄存器（比如 ACC）的信息写入到端口中。这样，CPU 对 I/O 设备的操作，就可以转换为对 I/O 端口的操作。一些端口，再加上对应的控制逻辑，就组成了接口。

CPU 要想访问某个 I/O 端口，就需要对 I/O 端口进行统一编号，每个端口对应一个 **端口地址**。对 I/O 端口的编址，可以选择跟内存统一，也可以完全独立。

1. 统一编址

统一编址就是把 I/O 地址看作存储器地址的一部分，也就是把 I/O 端口当做存储器单元统一进行地址分配。也称为 **存储器映射方式**。

这样，CPU 访问 I/O 端口时就可以直接用访存指令，而不需要专门的 I/O 指令，CPU 控制 I/O 设备更加方便；而缺点是端口会占用内存地址空间，减少了主存容量。

2. 不统一编址

不统一编址就是 I/O 地址和存储器地址完全分开，两者都有自己独立的地址空间，所以又叫 **独立编址**。也称为 **I/O 映射方式**。

不统一编址时，CPU 访问一个地址时，就无法从地址码上进行区分；所以必须通过设置专门的 **I/O 指令** 来访问 I/O 端口。这样就不会占用主存地址空间，用专门的 I/O 指令编写程序也会更加清晰；缺点是让指令系统更加复杂，CPU 需要提供对内存和 I/O 设备的两套读/写控制信号，硬件成本也更高。

6.4.4 I/O 接口的类型

I/O 接口按不同的方式，可以有以下几种分类。

- 按数据传送方式：分为 **并行接口** 和 **串行接口**；
- 按功能选择的灵活性：分为 **可编程接口** 和 **不可编程接口**；
- 按通用性：分为 **通用接口** 和 **专用接口**；
- 按数据传送的控制方式：分为 **程序型接口** 和 **DMA 型接口**。

程序型接口用于连接速度较慢的 I/O 设备，如显示终端、键盘、打印机等；具体的控制方式包括 **程序查询方式** 和 **程序中断方式**。DMA 型接口用于连接高速 I/O 设备，如磁盘、磁带等；控制方式采用 **DMA 方式**。

6.5 I/O 方式

I/O 设备与主机交换信息时，对信息传送的控制方式一共有五种：**程序查询方式**、**程序中断方式**、**直接存储器存取方式 (DMA)**、**I/O 通道方式**、**I/O 处理机方式**。

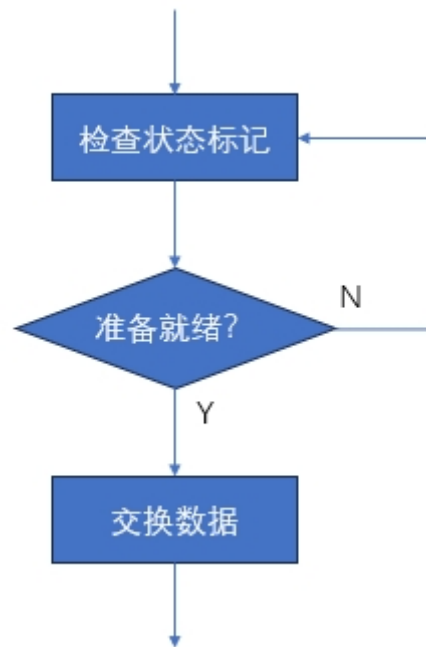
其中，前两种方式都是通过 CPU 中程序指令的执行来控制的。

6.5.1 程序查询方式

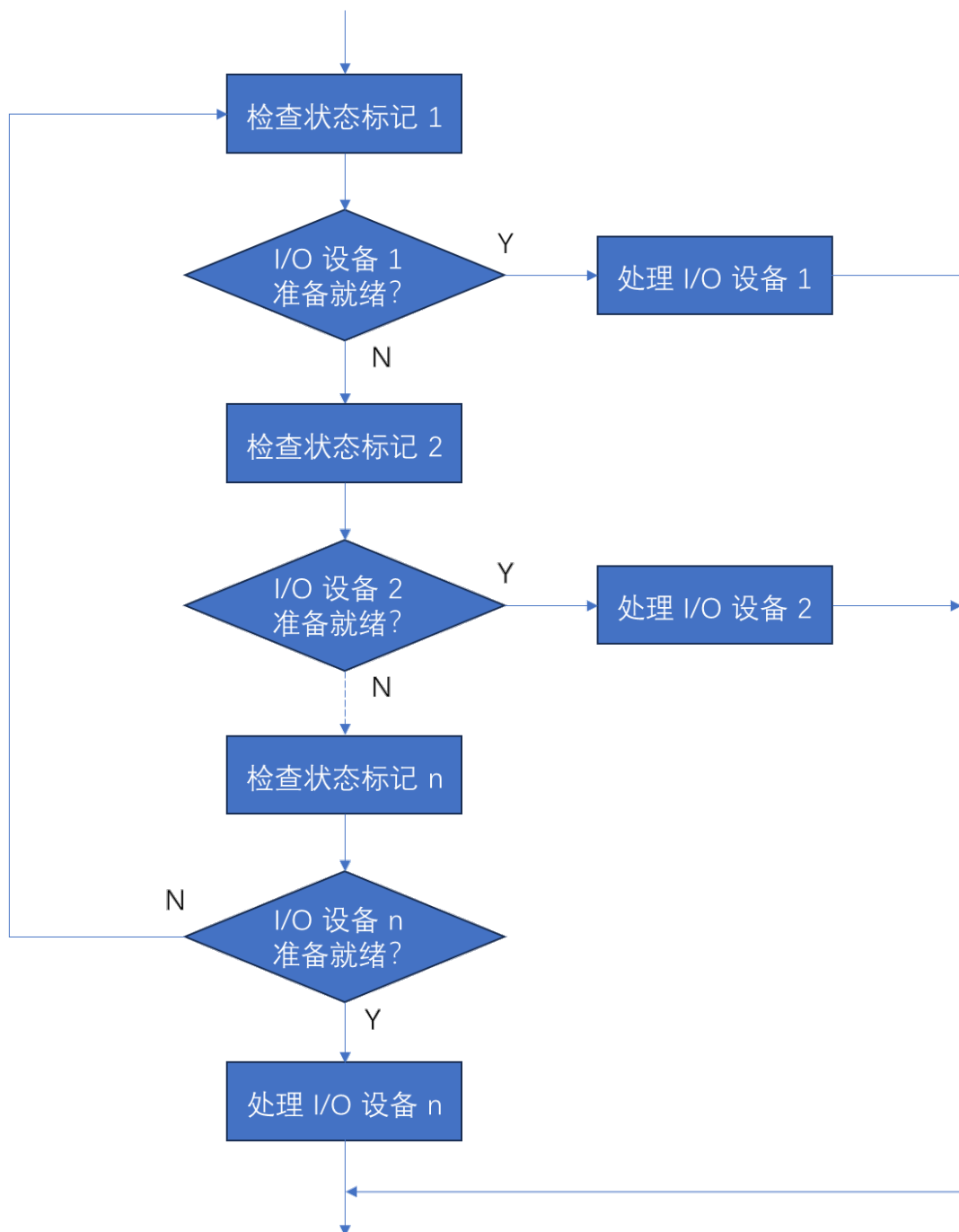
程序查询方式是由 CPU 通过程序不断查询 I/O 设备是否已做好准备，从而控制 I/O 设备与主机交换信息。

1. 程序查询基本流程

采用这种方式实现主机和 I/O 设备交换信息，要求 I/O 接口内设置一个 **状态标记**，用来反映 I/O 设备是否准备就绪。CPU 通过检测这个标记，就可以了解 I/O 设备的准备情况。



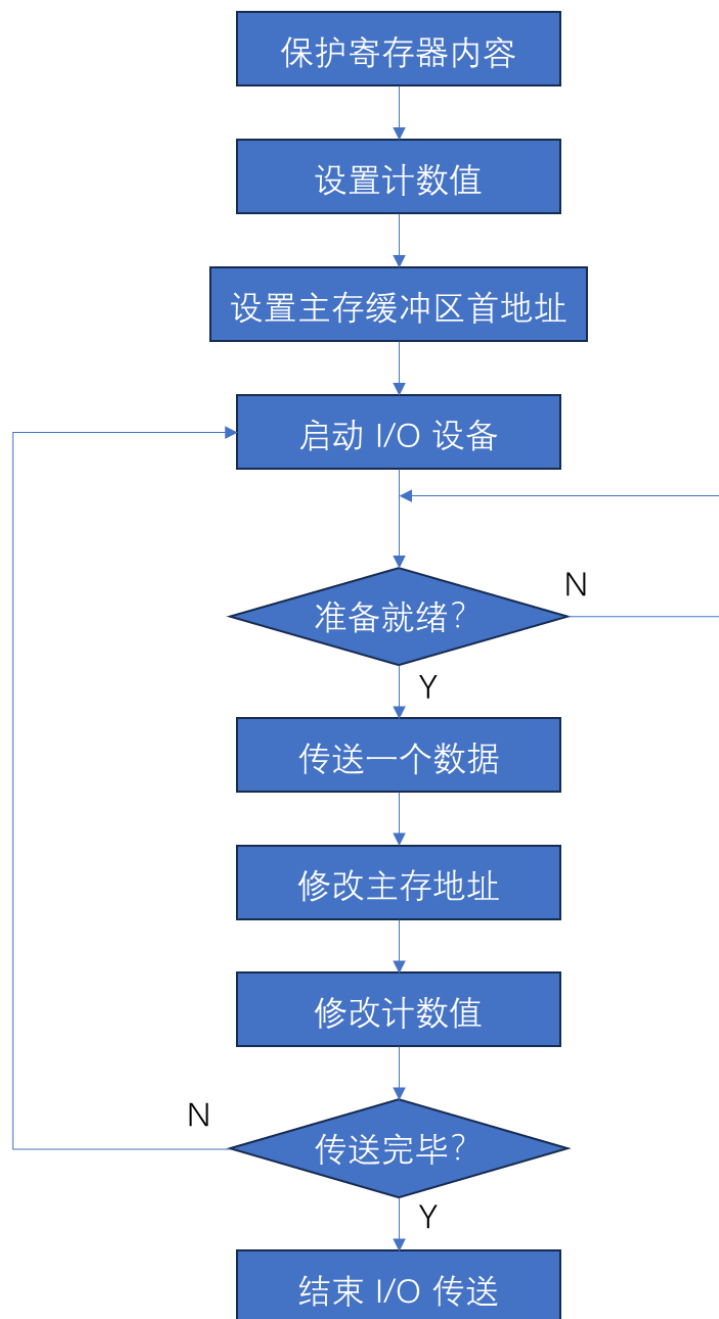
当 I/O 设备较多时，CPU 就需要按 I/O 设备在系统中的 **优先级** 进行逐级查询。



为了完成这个查询的流程，CPU 通常需要执行以下 3 条指令：

- **测试指令**：用来查询 I/O 设备是否准备就绪；
- **传送指令**：当 I/O 设备准备就绪时，执行数据的传送指令；
- **转移指令**：如果 I/O 设备未准备就绪，应执行转移指令；转至测试指令，继续测试 I/O 设备的状态。

当执行一段程序，需要启动某个 I/O 设备进行数据交互时，就把查询流程插入到运行的程序中。
具体的查询流程如下：



- ① 这种方式传送数据时要占用 CPU 中的寄存器，所以首先要将寄存器原内容保护起来；
- ② 传送的往往是一批数据，所以需要设置 I/O 设备与主机交换数据的计数值，用来控制数据量；
- ③ 设置要传送的数据在主存缓冲区的首地址；
- ④ CPU 启动 I/O 设备；

⑤ 将 I/O 接口中的设备状态标志取至 CPU 并测试 I/O 设备是否准备就绪。如果未准备就绪，则等待，直到准备就绪为止；当准备就绪时，接着可实现传送。

对输入而言，准备就绪意味着接口电路中的数据缓冲寄存器已装满欲传送的数据，称为 **输入缓冲满**，CPU 可以取走数据；对输出而言，准备就绪意味着接口电路中的数据已被设备取走，称为 **输出缓冲空**，这样 CPU 可以再次将数据送到接口，设备则可以再次从接口接收数据。

⑥ CPU 执行 I/O 指令，从 I/O 接口的数据缓冲寄存器中读出一个数据（输入），或者把一个数据写入 I/O 接口中的数据缓冲寄存器内（输出），同时将接口中的状态标志复位；

- ⑦ 修改主存地址；

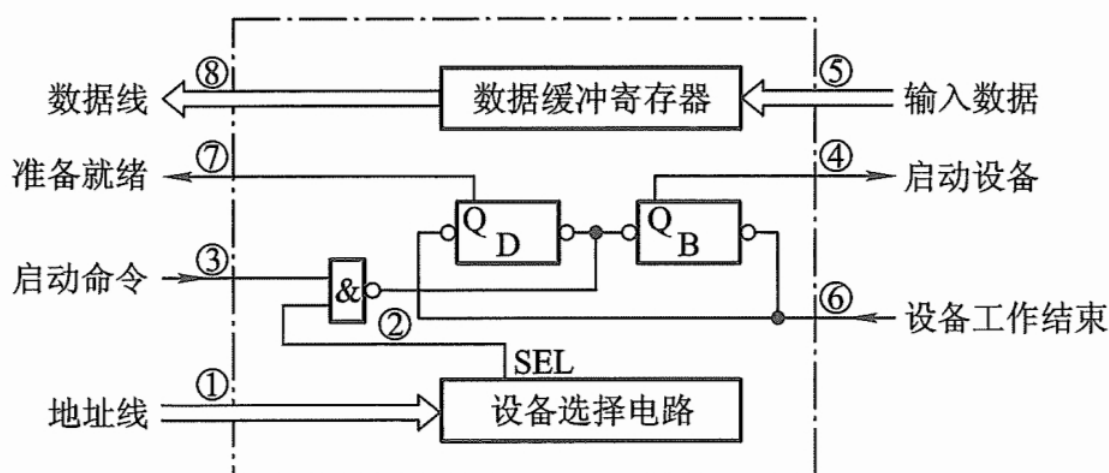
- ⑧ 修改计数值，若原设置计数值为原码，则依次减 1；若原设置计数值为负数的补码，则依次加 1；
- ⑨ 判断计数值。若计数值不为 0，表示一批数据尚未传送完，重新启动外设继续传送；若计数值为 0，则表示一批数据已传送完毕；
- ⑩ 结束 I/O 传送，继续执行原程序。

只要一启动 I/O 设备，CPU 就不断地查询 I/O 设备的准备情况，这就会暂停原程序的执行。当 I/O 设备准备就绪后，就将数据逐个传送；直到数据全部传送结束，CPU 才重新回到原程序继续执行。

所以在程序查询方式下，CPU 和 I/O 设备是串行工作的，效率不高。

2. 接口电路

程序查询方式对应的接口电路基本组成如下：



图中的数据缓冲寄存器用来存放要传送的数据；D 是完成触发器，B 是工作触发器。

3. 接口工作过程

以输入设备为例，数据应该从 I/O 设备传送至主机。I/O 接口的具体工作过程为：

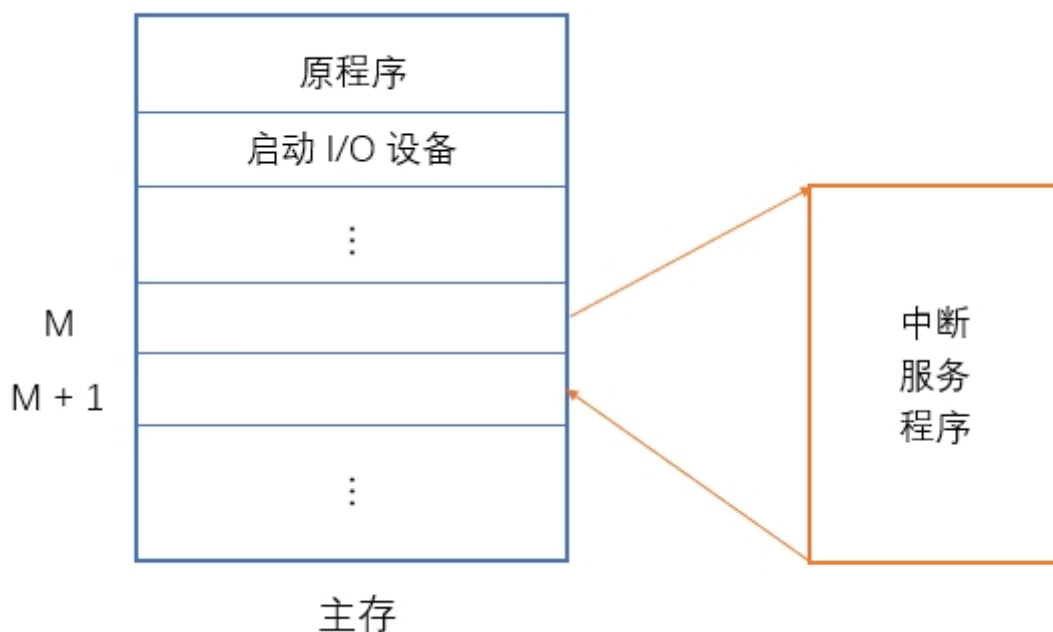
- ① CPU 执行 I/O 指令启动输入设备；指令中的设备码字段，通过地址线（设备选择线）送至设备选择电路；
- ② 如果接口的设备码与地址线上信号相同，设备被选中，输出 SEL 有效；
- ③ I/O 指令中的启动命令输入，通过与非门将完成触发器 D 置 0（复位），将工作触发器 B 置 1（置位）；
- ④ 工作触发器 B 输出启动信号，启动设备开始工作；
- ⑤ 输入设备将数据送至数据缓冲寄存器；
- ⑥ 输入缓冲满，由设备发出工作结束信号，将完成触发器 D 置 1，工作触发器 B 置 0；
- ⑦ 完成触发器 D 输出 1，通知 CPU “准备就绪”；
- ⑧ CPU 执行输入指令，将数据缓冲寄存器中的数据取出，送至通用寄存器，再存入主存。

6.5.2 程序中断方式

程序查询方式效率较低，主要原因就是 I/O 设备工作速度较慢，CPU 启动设备后，需要等待一段时间 I/O 设备才能准备就绪、开始信息交换；而 CPU 则耗费了大量时间进行状态查询。

如果 CPU 在启动 I/O 设备后，继续执行自身的原程序；等到 I/O 设备准备就绪后，主动向 CPU 发出请求再予以响应，就可以大大提升工作效率。这个请求就可以以 **I/O 中断** 的形式出现。

1. 程序中断基本流程



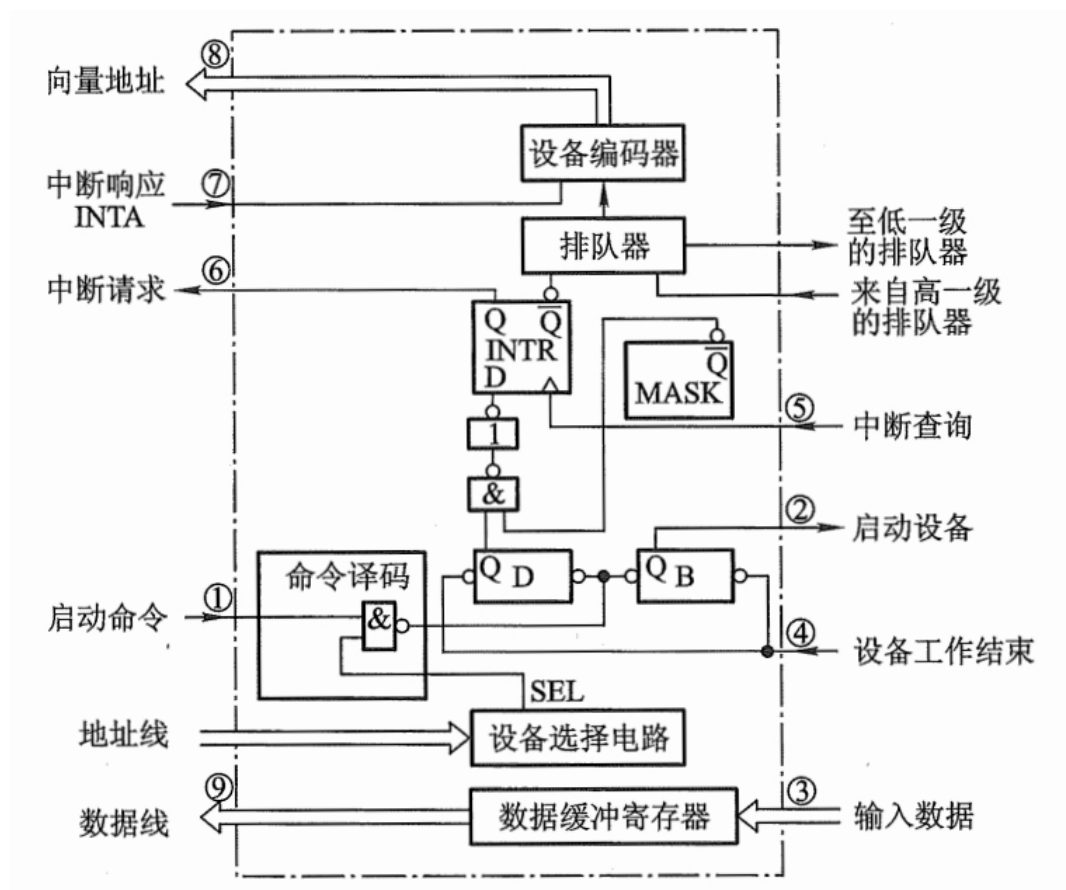
上图中，CPU 在执行 I/O 指令启动 I/O 设备之后，继续执行原程序；在执行第 M 条指令时，I/O 设备准备就绪，发来了中断请求，于是 CPU 在第 M 条指令执行结束后进入中断周期，转而执行中断服务程序；等到中断服务程序执行完毕，再返回到程序断点处，继续执行第 M + 1 条指令。

这种利用 I/O 中断，暂时中断 CPU 现行程序、转入 I/O 服务程序的方式，就称为 **程序中断方式**。

采用程序中断方式，CPU 就可以不必等待 I/O 设备的准备过程，工作效率得到了明显的提升。不过 CPU 和 I/O 接口需要增加响应的硬件电路，还要编制对应的中断服务程序。

2. 接口电路

很明显，采用程序中断方式，必须在接口电路中增加中断处理相应的硬件，比如中断请求触发器 INTR、中断屏蔽触发器 MASK、排队器、中断向量地址形成部件等。



上图中，完成触发器 D 和 屏蔽触发器 MASK 的输出通过与非门连接，产生中断请求信号；并通过排队器判优，交给设备编码器（中断向量地址形成部件）产生向量地址。

3. I/O 中断处理过程

同样，CPU 响应 I/O 设备提出中断请求的条件，是允许中断触发器 $EINT = 1$ ；CPU 响应 I/O 中断请求的时间，是在某条指令执行阶段的结束时刻。

以输入设备为例，I/O 中断处理的过程为：

- ① CPU 发出 I/O 设备启动命令，将完成触发器 D 置 0，将完成触发器 B 置 1；
- ② 工作触发器 B 启动设备，开始工作；
- ③ 输入设备将数据送至数据缓冲寄存器；
- ④ 输入缓冲满，设备发出工作结束信号，将完成触发器 D 置 1，完成触发器 B 置 0；
- ⑤ 指令执行阶段的结束时刻，CPU 发出中断查询信号；
- ⑥ 当设备准备就绪（ $D = 1$ ），且未被屏蔽（ $MASK = 0$ ）时，中断请求触发器 INTR 被置 1，向 CPU 发出 **中断请求**；同时 INTR 信号送至排队器，进行 **中断判优**。
- ⑦ 当 $EINT = 1$ （允许中断），而设备又被排队器选中时，进入 **中断响应** 阶段；CPU 发来的中断响应信号 INTA 将排队器的输出送至编码器，形成向量地址；
- ⑧ 向量地址送至 PC，作为下一条指令的地址；随后跳转至中断服务程序入口地址，进入 **中断服务** 阶段；
- ⑨ 执行中断服务程序，通过输入指令将数据缓冲寄存器的数据取出，送至通用寄存器，再存入主存。
- ⑩ 中断服务程序最后一条指令是 **中断返回** 指令，执行结束后返回原程序的断点处。

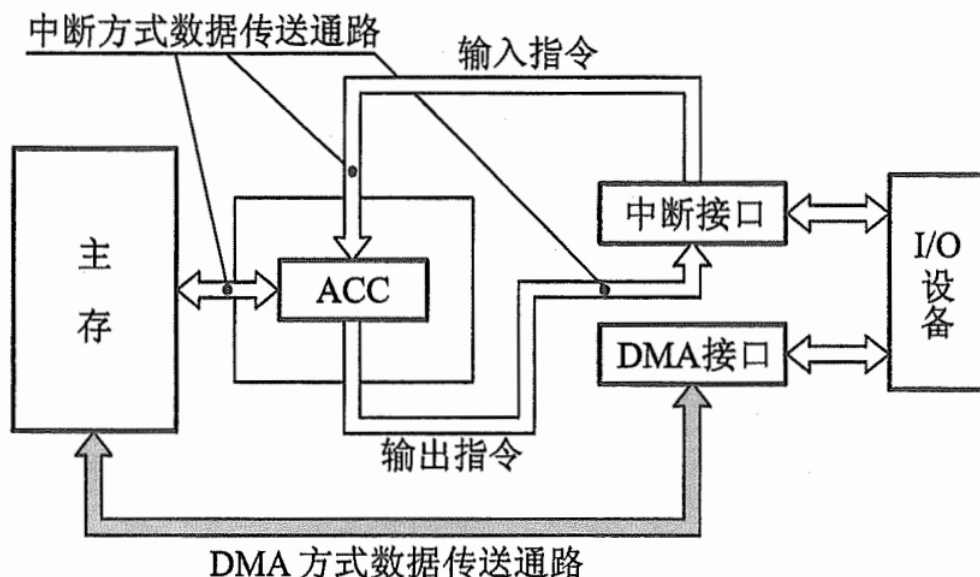
这里也可以看出，一次中断处理的过程可以简单分为 **中断请求**、**中断判优**、**中断响应**、**中断服务** 和 **中断返回** 5 个阶段。

6.5.3 DMA 方式

1. DMA 的概念和特点

如果 I/O 设备能直接与主存交换信息而不占用 CPU，那么 CPU 的资源利用率就可以进一步提高，这种方式就被称为 **直接存储器存取**（Direct Memory Access，**DMA**）。

DMA 方式的特点是，I/O 设备与主存之间有一条直接数据通路。因此，I/O 设备可以与主存直接交换信息，而不需要调用中断服务程序。这样，CPU 就不必暂停现行程序、专门去为设备服务，省去了保护现场和恢复线程的过程；所以工作效率比程序中断方式更高。



对于 **高速 I/O 设备** 或者 **辅存**，如果采用程序中断方式，那么每次与主机进行数据交互时都要等待 CPU 做出中断响应，很可能造成数据丢失；因此更适合采用 **DMA 方式**。

2. DMA 与 CPU 的访存冲突

在 DMA 方式中，由于 DMA 接口与 CPU 都可以访问主存，这就有可能出现两者争用主存的冲突。

I/O 设备请求进行 DMA 传送时，会遇到三种情况：

- CPU 此时不需要访问主存：这时 I/O 设备与 CPU 不发生冲突；
- CPU 正在访问主存：这时必须等存取周期结束，CPU 才会将总线占有权让出；
- CPU 也同时要求访问主存：这就出现了访存冲突。

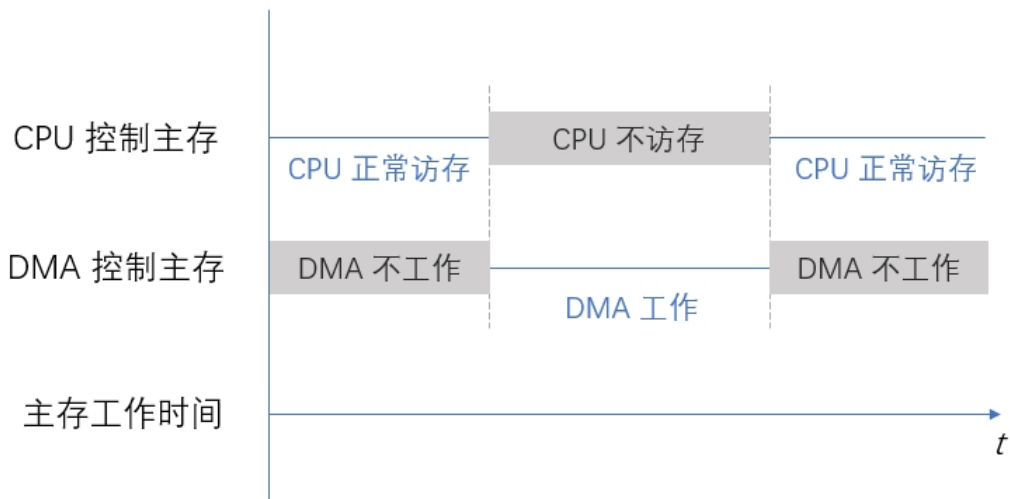
发生访存冲突时，I/O 访存要优先于 CPU 的访存，因为 I/O 设备不立即访问主存的话就可能丢失数据。所以一般这时需要 CPU 进行“让步”。

为了有效地分时使用主存，通常采用以下三种方法来解决访存冲突：

(1) 停止 CPU 访问主存

这种方法就是 CPU 彻底停止访存、等 I/O 访存结束之后再进行。

当外设要求传送一批数据时，由 DMA 接口向 CPU 发一个停止信号，要求 CPU 放弃地址线、数据线和有关控制线的使用权。DMA 接口获得总线控制权后，开始进行数据传送；数据传送结束后，DMA 接口通知 CPU 可以使用主存，并把总线控制权交回给 CPU。



这种方式的优缺点如下：

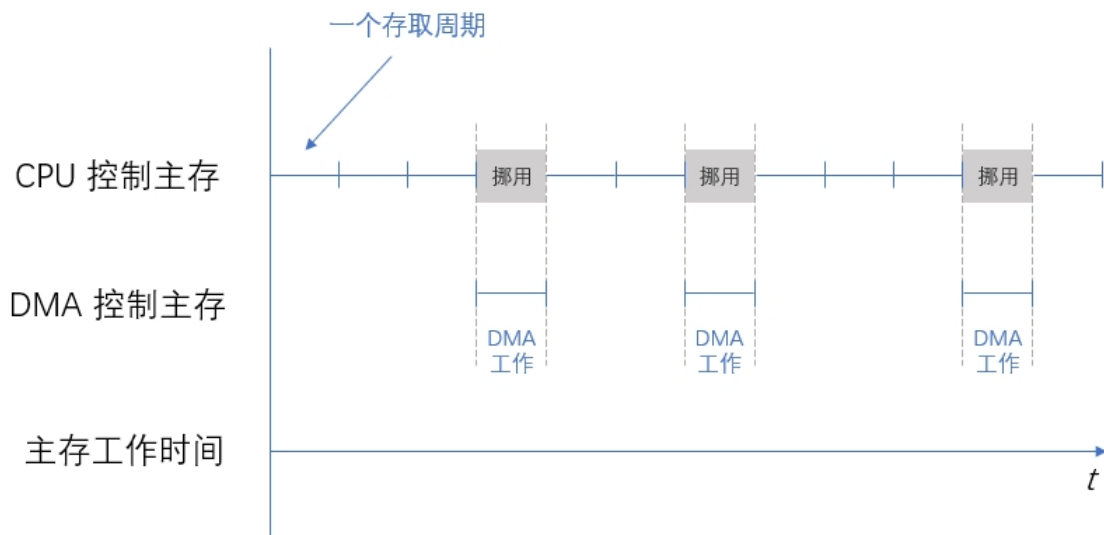
- 优点：控制简单，适用于数据传输率很高的 I/O 设备，进行成组数据的传送。
- 缺点：DMA 接口在访问主存时，CPU 基本上处于不工作状态或保持原状态。

所以这种方式下，CPU 对主存的利用率不高。

(2) 周期挪用 (或周期窃取)

如果 I/O 设备发出 DMA 请求时，CPU 并不是完全放弃总线的使用权，而是允许 I/O 设备 “挪用” 或 “窃取” 总线使用权一个或几个主存周期；这种方式就被称为 **周期挪用**，或 **周期窃取**。当 DMA 不请求时，CPU 仍可以继续访问主存。

这就意味着，CPU 在执行访问主存指令过程中，插入了 DMA 请求，那么就会被窃取了若干个存取周期，使 CPU 延缓若干存取周期后再访问主存。

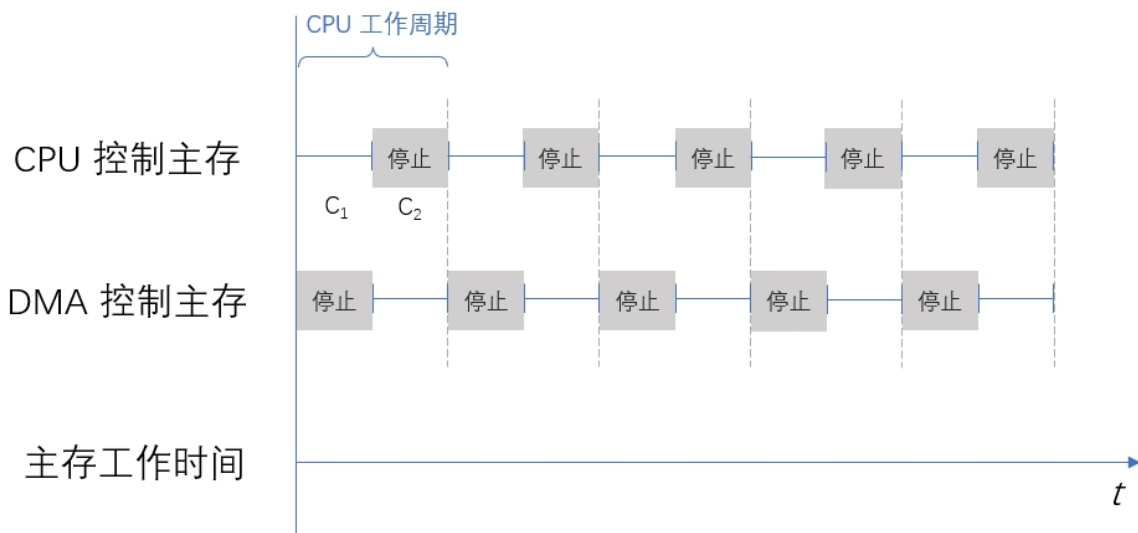


与 CPU 暂停访存的方式相比，这种方式既实现了 I/O 传送，又较好地发挥了主存与 CPU 的效率，是一种广泛采用的方法。

(3) DMA 与 CPU 交替访问

这种方法就是 DMA 与 CPU 轮流进行访存，平均分配主存的使用时间。

交替访问的方法适合于 CPU 的工作周期比主存存取周期长的情况。



例如，CPU 的工作周期是主存取周期的 2 倍以上，那么可以将一个 CPU 周期分为 C~1~ 和 C~2~ 两个子周期，

其中 C~1~ 专门进行 CPU 访存，而 C~2~ 专门进行 DMA 访存。

这种方式不需要总线使用权的申请、建立和归还过程，总线使用权是通过 C~1~ 和 C~2~ 分别控制的。CPU 与 DMA 接口各自有独立的访存地址寄存器、数据寄存器和读/写信号。

在这种工作方式下，CPU 既不停止主程序的运行，也不会进入等待状态。当然，其相应的硬件逻辑也会变得更为复杂。

3. DMA 接口的功能

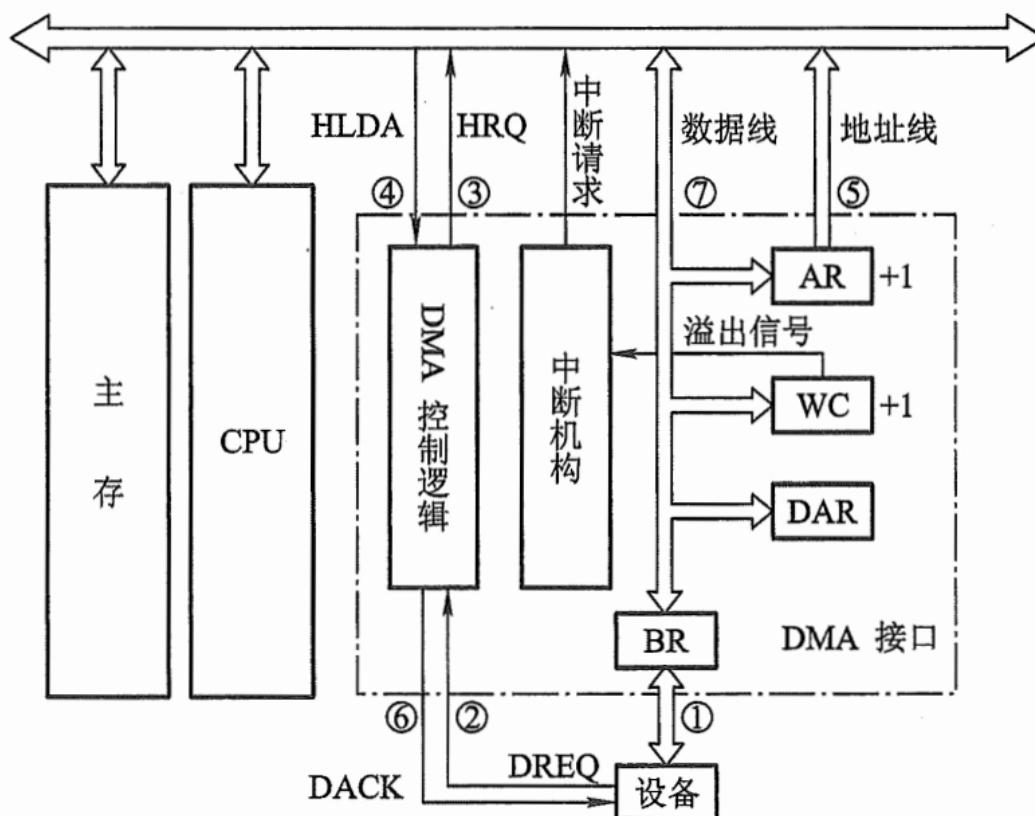
采用 DMA 方式进行数据传送时，数据的传输过程完全由 DMA 接口电路控制；因此 DMA 接口也被称为 **DMA 控制器**。

DMA 接口应该具有以下功能：

- 向 CPU 申请 DMA 传送；
- 在 CPU 允许 DMA 工作时，处理总线控制权的转交；
- 在 DMA 期间管理系统总线，控制数据传输；
- 确定数据传送的起始地址和数据长度，并在传送过程中进行修正；
- 数据块传送结束时，向 CPU 提交 DMA 操作完成的信号。

4. DMA 接口的组成

最简单的 DMA 接口组成如下图所示：



接口中的主要部件有：

- **主存地址寄存器 (AR)**：用于存放主存中需要交换的数据的地址。DMA 传送数据前，应该将数据在主存中的首地址送至 AR；DMA 传送过程中，每交换一次数据，就将 AR 的内容加 1，直到一批数据传送完毕。
- **字计数器 (WC)**：用来记录传送数据的总字数。DMA 传送过程中，每传送一个数据字，WC 就减 1 (若是补码则加 1)，直到计数器为 0 (溢出)，就表示这批数据传送完毕；DMA 接口就可以向 CPU 发出中断请求信号了。
- **设备地址寄存器 (DAR)**：存放 I/O 设备的设备码，或者辅存中的寻址信息。
- **数据缓冲寄存器 (BR)**：用来暂存每次要传送的数据。
- **DMA 控制逻辑**：由控制电路、时序电路、命令状态控制寄存器等组成，负责管理 DMA 的传送过程。每当设备准备好一个数据字，就向 DMA 接口提出请求 (DREQ)，DMA 控制逻辑就向 CPU 请求 DMA 服务，发出总线使用权的请求信号 HRQ；CPU 发出响应信号 HLDA 后，DMA 控制逻辑就开始负责管理 DMA 传送的全过程，并通知设备已经被授权了一个 DMA 周期 (DACK)。
- **中断机构**：用来提出中断请求。当字计数器 WC 溢出 (全 0) 时，一批数据传送完毕，这个“溢出信号”就通过中断机构向 CPU 提出中断请求，由 CPU 做 DMA 操作的后处理。

需要注意的是，这里和程序中断方式中的中断，都属于 I/O 中断；但中断的目的不同：程序中断方式的中断是为了传送数据，而 DMA 中的中断只是为了报告一批数据传送结束。

5. DMA 的工作过程

DMA 的数据传送过程可以分为 **预处理**、**数据传送** 和 **后处理** 三个阶段。

(1) 预处理

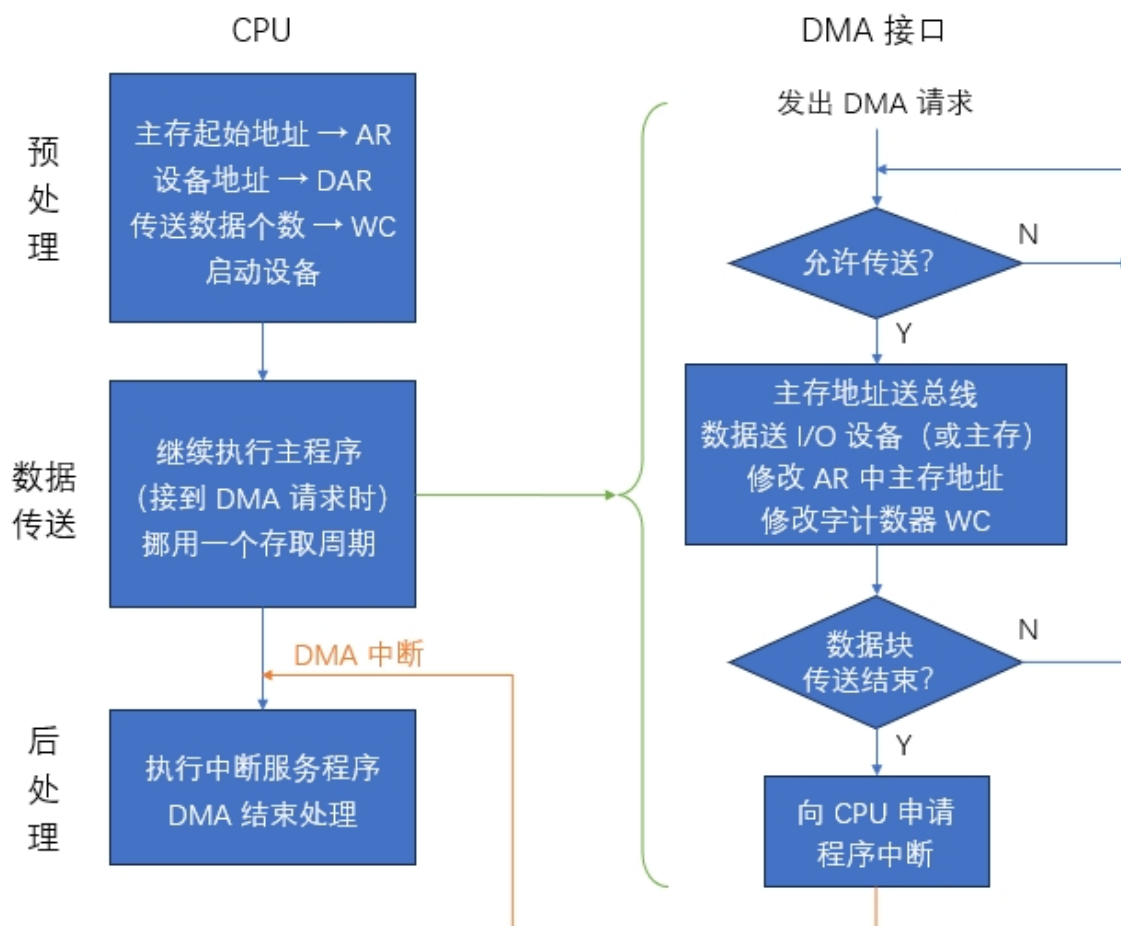
在 DMA 接口开始工作之前，需要先做一些准备工作，由 CPU 给它预置一些信息；这就是“**预处理**”阶段。

- 告诉 DMA 控制逻辑数据传送的方向，是输入 (写主存) 还是输出 (读主存)；
- 将设备码写入 DMA 设备地址寄存器 DAR，并启动设备；
- 将交换数据的主存起始地址，写入 DMA 主存地址寄存器 AR；
- 将交换数据的个数，写入字计数器 WC

这些准备工作由 CPU 执行几条 I/O 指令完成，也就是程序的初始化阶段。这些工作完成后，CPU 可以继续执行原来的程序。

当 I/O 设备准备就绪时，就通过 DMA 接口向 CPU 提出占用总线的请求；若有多个 DMA 接口同时申请，就按优先级由 **硬件排队判优逻辑** 决定响应顺序。

I/O 设备得到主存总线的控制权后，数据的传送就全部由 DMA 接口进行管理。



(2) 数据传送

DMA 方式是以 **数据块** 为单位进行数据传送的。

这里以周期挪用的 DMA 方式为例，对于数据输入的场景，数据传送的流程如下：

① 当设备准备好一个数据字时，发出选通信号，将该字读到数据缓冲寄存器（BR）中，表示数据缓冲寄存器“满”；

② 与此同时，设备向 DMA 接口发出请求（DREQ）；

③ DMA 接口向 CPU 申请总线控制权（HRQ）；

④ CPU 发回应答信号 HLDA，表示允许将总线控制权交给 DMA 接口；

⑤ 将主存地址寄存器 AR 中的主存地址，送至地址总线，并向主存发出写命令；

⑥ 通知设备已被授予一个 DMA 周期（DACK），并为交换下一个字做准备；

⑦ 将数据缓冲寄存器 DAR 的内容，送至数据总线；

⑧ 主存将数据总线上的信息，写至地址总线指定的存储单元中；

⑨ 修改 AR 中的主存地址和字计数值 WC。

⑩ 判断数据块是否传送结束，如果没有结束则继续传送；如果已结束（字计数器溢出），则向 CPU 发出程序中断请求，标志数据块传送结束。

若为输出数据，整体步骤类似，主要的不同就是：

- 第①步需要将 BR 中的数据送至 I/O 设备，表示 BR 已“空”；
- 第⑤步需要向主存发出读命令；
- 第⑦⑧步，是将主存相应存储单元的内容，通过数据总线写入 BR，并送至输出设备；

(3) 后处理

当 DMA 的中断请求得到响应后，CPU 停止原程序的执行，转而去执行中断服务程序，做一些 DMA 的结束工作，这就是“后处理”阶段。

DMA 的结束阶段工作主要包括：

- 校验送入主存的数据是否正确；
- 决定是否继续用 DMA 传送其他数据块。若继续传送，则又要对 DMA 接口进行初始化；若不需要传送，则停止外设；
- 测试在传送过程中是否发生错误；如果出错，则转错误诊断及处理程序。

6. DMA 方式和程序中断方式的对比

DMA 方式和程序中断方式是 I/O 设备最为常见的信息传送控制方式。两者都用到了中断，但中断的目的不同：程序中断方式的中断是为了传送数据，而 DMA 中的中断只是为了报告一批数据传送结束；而且中断的优先级上，DMA 中断的优先级更高。

	程序中断方式	DMA 方式
数据传送方法	程序传送	硬件传送
CPU 响应时间	一条指令执行结束时	指令周期内任一存取周期结束时
处理异常事件能力	有	无
保护现场	需要	不需要
中断目的	传送数据	报告 CPU 数据传送结束
中断优先级	相对低	相对高

6.6 章节练习

一、单项选择题

1. 【2011真题】在系统总线的数据线上，不可能传输的是（ ）
- A. 指令 B. 操作数 C. 握手（应答）信号 D. 中断类型号

答案：C

要点：握手（应答）信号应该在通信总线上传输。

2. 【2012真题】下列关于 USB 总线特性的描述中，错误的是（ ）
- A. 可实现外设的即插即用和热拔插 B. 可通过级联方式连接多台外设
- C. 是一种通信总线，连接不同外设 D. 同时可传输 2 位数据，数据传输率高

答案：D

要点：USB（通用串行总线）属于串行总线，不能同时传输 2 位数据。

3. 【2016真题】下列关于总线设计的叙述中, 错误的是 ()

- A. 并行总线传输比串行总线传输速度快 B. 采用信号线复用技术可减少信号线数量
C. 采用突发传输方式可提高总线数据传输率 D. 采用分离事务通信方式可提高总线利用率

率

答案: A

要点: 实际时钟频率比较低的情况下, 并行总线因为可以同时传输多位数据, 速率确实比串行总线快。但是, 随着技术的发展, 时钟频率越来越高, 并行导线之间的相互干扰越来越严重, 当时钟频率提高到一定程度时, 传输的数据已经无法恢复。而串行总线因为导线少, 线间干扰容易控制, 反而可以通过不断提高时钟频率来提高传输速率。

4. 【2009真题】假设某系统总线在一个总线周期中并行传输 4B 信息, 一个总线周期占用 2 个时钟周期, 总线时钟频率为 10MHz, 则总线带宽是 ()

- A. 10MB/s B. 20MB/s C. 40MB/s D. 80MB/s

答案: B

要点: 重点考虑每个时钟周期传输的数据量, 总线带宽 = 总线时钟频率 × 每个总线时钟周期传输数据量

5. 【2018真题】下列选项中, 可提高同步总线数据传输率的是 ()

I .增加总线宽度 II .提高总线工作频率 III.支持突发传输 IV.采用地址/数据线复用

- A. 仅 I 、 II B. 仅 I 、 II 、 III C. 仅 III、IV D. I 、 II 、 III和IV

答案: B

要点: 总线数据传输率 = 总线工作频率 × (总线宽度 / 8) 。

采用地址/数据线复用只是减少了线的数量, 节省了成本, 并不能提高传输率。

6. 【2020真题】QPI 总线是一种点对点全双工同步串行总线, 总线上的设备可同时接收和发送信息, 每个方向可同时传输 20 位信息 (16 位数据 + 4 位校验位), 每个 QPI 数据包有 80 位信息, 分 2 个时钟周期传送, 每个时钟周期传递 2 次。因此, QPI 总线带宽为: 每秒传送次数 × 2B × 2。若 QPI 时钟频率为 2.4GHz, 则总线带宽为 ()

- A. 4.8GB/s B. 9.6GB/s C. 19.2GB/s D. 38.4GB/s

答案: C

要点: 公式中最后的 × 2, 是指全双工模式下两个方向的数据传输; 而 QPI 采用了双倍数据率技术 (DDR), 时钟信号上升沿和下降沿都可以进行数据传输 (题目中提示 “每个时钟周期传递 2 次”), 因此带宽为:

$$2.4GHz \times 2B \times 2 \times 2 = 19.2GB/s$$

7. 【2014真题】一次总线事务中, 主设备只需给出一个首地址, 从设备就能从首地址开始的若干连续单元读出或写入多个数据。这种总线事务方式称为 ()

- A. 并行传输 B. 串行传输 C. 突发传输 D. 同步传输

答案: C

8. 【2012真题】某同步总线的时钟频率为 100MHz, 宽度为 32 位, 地址/数据线复用, 每传输一个地址或数据占用一个时钟周期。若该总线支持突发 (猝发) 传输方式, 则一次 “主存写” 总线事务传输 128 位数据所需要的时间至少是 ()

A. 20ns B. 40ns C. 50ns D. 80ns

答案: C

要点: 对于宽度为 32 位的总线, 传输 128 位数据需要 $128 / 32 = 4$ 个时钟周期; 如果数据地址连续, 在突发方式下只需要传递一次地址, 而地址/数据线复用, 所以共需 5 个时钟周期。

9. 【2015真题】下列有关总线定时的叙述中, 错误的是 ()

- A. 异步通信方式中, 全互锁协议的速度最慢
- B. 异步通信方式中, 非互锁协议的可靠性最差
- C. 同步通信方式中, 同步时钟信号可由各设备提供
- D. 半同步通信方式中, 握手信号的采样由同步时钟控制

答案: C

要点: 在同步通信方式中, 系统必须采用统一的时钟信号, 否则无法实现统一的时钟。

10. 【2021真题】下列关于总线的叙述中, 错误的是 ()

- A. 总线是在两个或多个部件之间进行数据交换的传输介质
- B. 同步总线由时钟信号定时, 时钟频率不一定等于工作频率
- C. 异步总线由握手信号定时, 一次握手过程完成一位数据交换
- D. 突发 (Burst) 传送总线事务可以在总线上连续传送多个数据

答案: C

要点: 一次握手过程可能完成不止一位的数据交换。

11. 【2014真题】下列有关 I/O 接口的叙述中, 错误的是 ()

- A. 状态端口和控制端口可以合用同一个寄存器
- B. I/O 接口中 CPU 可访问的寄存器称为 I/O 端口
- C. 采用独立编址方式时, I/O 端口地址和主存地址可能相同
- D. 采用统一编址方式时, CPU 不能用访存指令访问 I/O 端口

答案: D

要点: 采用统一编址时, CPU 访存和访问 I/O 端口用的是一样的指令, 所以访存指令可以访问 I/O 端口。

12. 【2017真题】I/O 指令实现的数据传送通常发生在 ()

- A. I/O 设备和 I/O 端口之间
- B. 通用寄存器和 I/O 设备之间
- C. I/O 端口和 I/O 端口之间
- D. 通用寄存器和 I/O 端口之间

答案: D

要点: 在执行一条指令时, CPU 使用地址总线选择所请求的 I/O 端口, 使用数据总线在 CPU 寄存器和端口之间传输数据。

13. 【2021真题】下列选项中, 不属于 I/O 接口的是 ()

- A. 磁盘驱动器
- B. 打印机适配器
- C. 网络控制器
- D. 可编程中断控制器

答案: A

要点：I/O 接口即 I/O 控制器，其功能是接收主机发送的 I/O 控制信号，并实现主机和外部设备之间的信息交换。磁盘驱动器是由磁头、磁盘和读写电路等组成的，也就是我们平常所说的磁盘本身。

14. 【2014真题】若某设备中断请求的响应和处理时间为 100ns, 每 400ns 发出一次中断请求, 中断响应所允许的最长延迟时间为 50ns, 则在该设备持续工作过程中, CPU 用于该设备的 I/O 时间占整个 CPU 时间的百分比至少是 ()

A. 12.5% B. 25% C. 37.5% D. 50%

答案：B

要点：允许的延迟为干扰信息，无论延迟多久，每 400ns 都要花费 100ns 处理中断。

15. 【2015真题】在采用中断 I/O 方式控制打印输出的情况下, CPU 和打印控制接口中的 I/O 端口之间交换的信息不可能是 ()

A. 打印字符 B. 主存地址 C. 设备状态 D. 控制命令

答案：B

要点：在程序中断 I/O 方式中, CPU 和打印机直接交换数据, 打印字符直接传输到打印机的 I/O 端口, 不会涉及到主存地址。

16. 【2018真题】下列关于外部 I/O 中断的叙述中, 正确的是 ()

A. 中断控制器按所接收中断请求的先后次序进行中断优先级排队
B. CPU 响应中断时, 通过执行中断隐指令完成通用寄存器的保护
C. CPU 只有在处于中断允许状态时, 才能响应外部设备的中断请求
D. 有中断请求时, CPU 立即暂停当前指令执行, 转去执行中断服务程序

答案：C

要点：中断优先级由屏蔽字决定，而不是根据请求的先后次序。

中断隐指令完成的工作有：1) 关中断；2) 保存断点；3) 引出中断服务程序，通用寄存器的保护由中断服务程序完成。

有中断请求时，先要由中断隐指令完成中断前程序的状态保存。

17. 【2019真题】某设备以中断方式与 CPU 进行数据交换, CPU 主频为 1GHz, 设备接口中的数据缓冲寄存器为 32 位, 设备的数据传输率为 50kB/s。若每次中断开销 (包括中断响应和中断处理) 为 1000 个时钟周期, 则 CPU 用于该设备输入/输出的时间占整个 CPU 时间的百分比最多是 ()

A. 1.25% B. 2.5% C. 5% D. 12.5%

答案：A

要点：设备接口中的数据缓冲寄存器为 32 位, 所以一次中断可以传输 4B 数据; 设备数据传输率为 50kB/s, 因此 1s 内共需要 $50\text{kB} \div 4\text{B} = 12.5\text{k}$ 次中断。而每次中断开销为 1000 个时钟周期, 所以 I/O 时间占比为:

$$(12.5k \times 1000) / 1G = 1.25\%$$

18. 【2019真题】下列关于 DMA 方式的叙述中, 正确的是 ()

I. DMA 传送前由设备驱动程序设置传送参数
II. 数据传送前由 DMA 控制器请求总线使用权
III. 数据传送由 DMA 控制器直接控制总线完成

IV. DMA 传送结束后的处理由中断服务程序完成

- A. 仅 I、II B. 仅 I、III、IV C. 仅 II、III、IV D. I、II、III、IV

答案: D

19. 【2020真题】若设备采用周期挪用 DMA 方式进行输入和输出, 每次 DMA 传送的数据块大小为 512 字节, 相应的 I/O 接口中有一个 32 位数据缓冲寄存器。对于数据输入过程, 下列叙述中, 错误的是 ()

- A. 每准备好 32 位数据, DMA 控制器就发出一次总线请求
B. 相对于 CPU, DMA 控制器的总线使用权的优先级更高
C. 在整个数据块的传送过程中, CPU 不可以访问主存储器
D. 数据块传送结束时, 会产生 “DMA 传送结束” 中断请求

答案: C

要点: 周期挪用方式下, DMA 利用 CPU 不访问存储器的那些周期来实现 DMA 操作, 此时 DMA 可以使用总线而不用通知 CPU, 也不会妨碍 CPU 的工作。

20. 【2021真题】异常事件在当前指令执行过程中进行检测, 中断请求则在当前指令执行后进行检测。下列事件中, 相应处理程序执行后, 必须回到当前指令重新执行的是 ()
- A. 系统调用 B. 页缺失 C. DMA 传送结束 D. 打印机缺纸

答案: B

21. 【2022真题】下列关于中断 I/O 方式的叙述中, 不正确的是 ()

- A. 适用于键盘、针式打印机等字符型设备
B. 外设和主机之间的数据传送通过软件完成
C. 外设准备数据的时间应小于中断处理时间
D. 外设为某进程准备数据时 CPU 可运行其他进程

答案: C

要点: 若外设准备数据的时间小于中断处理时间, 则可能导致数据丢失。

二、综合应用题

1. 【2018真题】假定计算机的主频为 500MHz, CPI 为 4。现有设备 A 和 B, 其数据传输率分别为 2MB/s 和 40MB/s, 对应 I/O 接口中各有一个 32 位数据缓冲寄存器。请回答下列问题, 要求给出计算过程。

(1) 若设备 A 采用定时查询 I/O 方式, 每次输入/输出都至少执行 10 条指令。设备 A 最多间隔多长时间查询一次才能不丢失数据? CPU 用于设备 A 输入/输出的时间占 CPU 总时间的百分比至少是多少?

(2) 在中断 I/O 方式下, 若每次中断响应和中断处理的总时钟周期数至少为 400, 则设备 B 能否采用中断 I/O 方式? 为什么?

(3) 若设备 B 采用 DMA 方式, 每次 DMA 传送的数据块大小 1000B, CPU 用于 DMA 预处理和后处理的总时钟周期数为 500, 则 CPU 用于设备 B 输入/输出的时间占 CPU 总时间的百分比最多是多少?

答案:

(1) 程序定时向缓存端口查询数据, 由于缓存端口大小有限, 必须在传输完端口大小的数据时访问端口, 以防止部分数据没有被及时读取而丢失。

设备 A 准备 32 位 (4B) 数据所用时间为

$$4B \div 2MB/s = 2\mu s$$

所以最多每隔 $2\mu s$ 必须查询一次。

这样, 每秒的查询次数至少是 $1s / 2\mu s = 5 \times 10^5$, 每秒 CPU 用于设备 A 输入/输出的时间至少为

$$5 \times 10^5 \times 10 \times 4 = 2 \times 10^7$$

个时钟周期, 占整个 CPU 时间的百分比至少是

$$2 \times 10^7 \div 500M = 4\%$$

(2) 设备 B 准备 32 位 (4B) 数据所用时间为

$$4B \div 40MB/s = 0.1\mu s$$

而中断响应和中断处理的时间为

$$400 \times (1/500M) = 0.8\mu s$$

准备数据的时间小于中断响应和中断处理的时间, 所以等待中断处理的过程中数据就会被刷新, 而造成丢失。因此, 设备 B 不适合采用中断 I/O 方式。

(3) 在 DMA 方式中, 数据的传送过程由 DMA 控制, 只有预处理和后处理需要 CPU 处理。

设备 B 每秒的 DMA 次数最多为

$$40MB \div 1000B = 40000$$

CPU 用于设备 B 输入/输出的时间占 CPU 总时间的百分比最多为

$$40000 \times 500 \div 500M = 4\%$$

2. 【2022真题】假设某磁盘驱动器中有 4 个双面盘片, 每个盘面有 20000 个磁道, 每个磁道有 500 个扇区, 每个扇区可记录 512 字节的数据, 盘片转速为 7200 RPM (转 / 分), 平均寻道时间为 5ms。请回答下列问题。

(1) 每个扇区包含数据及其地址信息, 地址信息分为 3 个字段。这 3 个字段的名称各是什么? 对于该磁盘, 各字段至少占多少位?

(2) 一个扇区的平均访问时间约为多少?

(3) 若采用周期挪用 DMA 方式进行磁盘与主机之间的数据传送, 磁盘控制器中的数据缓冲区大小为 64 位, 则在一个扇区读写过程中, DMA 控制器向 CPU 发送了多少次总线请求? 若 CPU 检测到 DMA 控制器的总线请求信号时也需要访问主存, 则 DMA 控制器是否可以获得总线使用权? 为什么?

答案:

(1) 3 个字段的名称为: 柱面号 (或磁道号)、盘面号 (或磁头号)、扇区号。

每个盘面有 20000 个磁道, 因此该磁盘共有 20000 个柱面, 柱面号字段至少占 15 位;

磁盘共有 8 个盘面, 因此盘面号字段至少占 3 位;

每个磁道有 500 个扇区, 因此扇区号字段至少占 9 位。

(2) 一个扇区的访问时间由 **寻道时间**、**等待时间**、**传输时间** 三部分组成。

平均寻道时间为 5ms;

平均等待时间, 等于磁盘转半圈所需要的时间:

$$60s \div 7200 \div 2 = \frac{1}{240}s \approx 4.167ms$$

平均传输时间, 等于一个扇区划过磁头下方所需要的时间:

$$60s \div 7200 \div 500 = \frac{1}{60000}s \approx 0.0167ms$$

因此一个扇区的平均访问时间约为:

$$5ms + 4.167ms + 0.0167ms \approx 9.18ms$$

(3) 磁盘控制器中的数据缓冲区每充满一次 (64位, 8B), DMA 控制器就需要发出一次总线请求。

因此在一个扇区读写过程中, DMA 控制器向 CPU 发送的总线请求次数为:

$$512B \div 8B = 64$$

采用周期挪用 DMA 方式, 当 CPU 和 DMA 控制器都需要访问主存时, DMA 控制器可以优先获得总线使用权。因为一旦磁盘开始读写, 就必须按时完成数据传送, 否则数据缓冲区中的数据会发生丢失。