

## 第五章

### 1、串行总线：

优点：

较少的线路数量：串行总线只需要少量的传输线路，可以减少物理连接的复杂性。

长距离传输：串行总线可以在较长的距离上进行传输，因为串行信号受干扰的可能性较小。

适用于高速传输：串行总线可以在较高的速率下进行数据传输。

缺点：

速率较低：相对于并行总线，串行总线的传输速率较低，每个传输周期只能传输一个比特位。

传输延迟：由于数据按照顺序逐位传输，传输延迟较高。

高的编解码开销：为了确保数据传输的准确性，串行总线通常需要进行编码和解码，增加了开销。

### 并行总线：

优点：

高速数据传输：并行总线可以同时传输多个比特位，因此传输速率较高。

低传输延迟：由于同时传输多个比特位，传输延迟较低。

简单的编解码：并行总线的编解码相对简单，不需要对数据进行分割和重组。

缺点：

大量的线路数量：并行总线需要大量的传输线路，特别是在数据位数较多的情况下，会增加物理连接的复杂性。

短距离传输：并行总线的传输距离相对较短，因为并行信号更容易受到干扰。

造成串行总线和并行总线接口速率不同的原因是数据位数和传输线路的限制。串行总线每个传输周期只能传输一个比特位，因此速率相对较低。而并行总线可以同时传输多个比特位，因此速率较高。但并行总线需要更多的传输线路，特别是在数据位数较多的情况下，对于物理连接和信号干扰的要求更高，因此在长距离传输和复杂系统中不太适用。

3、1) I<sup>2</sup>C (Inter-Integrated Circuit) 是一种串行通信协议，用于在集成电路之间进行短距离数据传输。I<sup>2</sup>C 数据包由以下几部分组成：

起始条件 (Start Condition)：起始条件是一个低电平到高电平的跳变，表示一个 I<sup>2</sup>C 传输的开始。在起始条件之前，时钟线 (SCL) 和数据线 (SDA) 都应处于高电平状态。

地址字节 (Address Byte)：地址字节用于指定 I<sup>2</sup>C 总线上的设备。它由 7 位地址和一个读/写位组成。地址位指示要通信的设备，读/写位指示传输是读取还是写入操作。地址字节的发送由主设备（通常是微控制器）发起。

数据字节 (Data Byte)：数据字节包含要传输的实际数据。每个数据字节有 8 位组成，它们可以是要写入设备的数据或从设备读取的数据。数据字节的传输可以由主设备或从设备发起。

确认位 (Acknowledgement Bit)：在每个数据字节的传输后，接收设备（主设备或从设备）会发送一个确认位作为响应。确认位表示数据的正确接收。如果接收设备成功接收数据，则发送一个低电平的应答 (ACK) 位；如果接收设备无法接收数据或发生错误，则发送一个高电平的非应答 (NACK) 位。

停止条件 (Stop Condition)：停止条件是一个高电平到低电平的跳变，表示一个 I<sup>2</sup>C 传输的结束。在停止条件之后，时钟线和数据线都应处于高电平状态。

2) I<sup>2</sup>C 是半双工的协议，这意味着数据在 I<sup>2</sup>C 总线上的传输方向只能是单向的，要么从主设备向从设备发送数据，要么从从设备向主设备发送数据。这与全双工的协议（如 SPI）不同，

全双工允许同时进行双向数据传输。

3) I2C 传输的起始和停止条件如下:

起始条件 (**Start Condition**) : 起始条件是在时钟线 (SCL) 为高电平时, 数据线 (SDA) 由高电平跳变到低电平。这表示一个 I2C 传输的开始。

停止条件 (**Stop Condition**) : 停止条件是在时钟线为高电平时, 数据线由低电平跳变到高电平。这表示一个 I2C 传输的结束。

7、磁盘控制电路通过决定请求的最优执行次序来减少磁盘访问用时的方法主要有以下两种:

执行调度算法: 磁盘控制电路可以使用不同的调度算法来确定磁盘访问请求的执行顺序。常见的调度算法包括先来先服务 (FCFS)、最短寻道时间优先 (SSTF)、电梯算法 (SCAN/C-SCAN)、最高响应比优先 (HRRN) 等。这些算法根据磁盘访问请求的位置和优先级等因素, 选择合适的次序进行执行, 以尽量减少寻道时间、提高磁盘访问效率。

使用预读和写缓冲: 磁盘控制电路可以利用预读和写缓冲技术来减少磁盘的读写延迟。预读是指在读取请求之前, 提前从磁盘中读取一定量的数据并存储在缓冲区中, 以便后续的读取请求可以更快地获取数据。写缓冲是指将写操作的数据暂时存储在缓冲区中, 然后由控制电路按照最优次序将数据写入磁盘, 以减少磁盘的写入延迟。

8、在 RAID 4 中, 写入优化对于读取速度的影响主要体现在以下几个方面:

写入带宽利用率提升: RAID 4 采用条带化的数据布局方式, 将数据分散存储在多个磁盘上, 并使用奇偶校验盘来存储校验信息。写入优化可以提高写入性能, 减少写入延迟, 从而提高整个 RAID 4 系统的写入带宽利用率。当写入操作在多个磁盘上并行进行时, 可以同时写入多个条带, 有效利用磁盘的带宽, 提升写入速度。

读取并行性增强: RAID 4 中的数据条带化布局使得读取操作可以在多个磁盘上并行进行。写入优化可以减少写入操作对磁盘的占用时间, 释放更多的时间窗口给读取操作, 从而增强了读取操作的并行性。当读取请求到达时, 可以同时从多个磁盘中读取数据, 提高读取速度。需要注意的是, 虽然写入优化对读取速度有一定的积极影响, 但它并不是直接针对读取操作进行优化的。RAID 4 中的奇偶校验盘的特性导致在读取操作时需要额外的计算开销, 因为要使用校验信息进行数据恢复。因此, 在 RAID 4 中, 写入优化的主要目标是提高写入性能和带宽利用率, 而对读取操作的影响是间接的, 并与具体的系统和工作负载相关。

## 第六章

1、常见的总线仲裁机制有以下几种:

集中式仲裁: 由一个中央仲裁器 (Arbitrator) 负责对总线上的各个设备进行仲裁。设备在需要访问总线时向中央仲裁器发送请求, 仲裁器根据预定义的优先级或其他规则, 决定哪个设备获得总线的访问权。优点是简单易实现, 适用于设备数量较少的系统。缺点是中央仲裁器可能成为性能瓶颈, 并且在中央仲裁器发生故障时整个系统可能会发生故障。

分布式仲裁: 每个设备都具有仲裁逻辑, 根据事先约定的仲裁算法决定是否能够获得总线访问权。常见的分布式仲裁算法有时间分割多路复用 (Time Division Multiplexing, TDM)、循环仲裁 (Round Robin Arbitration) 等。优点是分布式的性质使得系统更加灵活和可扩展, 能够处理大量设备的并发访问请求。缺点是实现相对复杂, 需要保证仲裁算法的公平性和可靠性。

基于仲裁线的仲裁: 设备之间通过专门的仲裁线进行仲裁。每个设备都能监听仲裁线上的信号, 根据信号的状态来判断是否能够获得总线访问权。优点是仲裁过程简单, 响应时间快,

适用于实时性要求较高的系统。缺点是对于大规模系统，仲裁线的数量和复杂度可能会增加。

不同的仲裁机制适用于不同的场景，具体选择取决于系统的需求和约束：

如果系统设备数量较少且不需要高度并发的访问，集中式仲裁是一种简单而有效的选择。

如果系统需要处理大量设备的并发请求，并且要求灵活性和可扩展性，分布式仲裁是更好的选择。

基于仲裁线的仲裁适用于实时性要求较高的系统，因为它可以提供较快的响应时间。

2、AMBA（Advanced Microcontroller Bus Architecture）是一种用于嵌入式系统中的开放式总线标准，由 ARM 公司提出和维护。AMBA 总线协议家族包括 APB（Advanced Peripheral Bus）、AHB（Advanced High-performance Bus）、AXI（Advanced eXtensible Interface）、ACE（AXI Coherency Extensions）和 CHI（Coherent Hub Interface）等。

**APB（Advanced Peripheral Bus）：**

特点：APB 是 AMBA 总线中的低功耗外设总线，主要用于连接低速外设。它采用简单的点对点连接方式，具有低功耗和低面积的特点。

使用场景：适用于连接低带宽、低复杂度的外设，如 GPIO 控制器、定时器等。

**AHB（Advanced High-performance Bus）：**

特点：AHB 是 AMBA 总线中的高性能总线，主要用于连接高速外设和存储器。它支持多主机和多从机的并行传输，具有更高的带宽和灵活性。

使用场景：适用于连接中等复杂度和中等带宽需求的外设和存储器，如 DMA 控制器、存储器控制器等。

**AXI（Advanced eXtensible Interface）：**

特点：AXI 是 AMBA 总线中的高性能、高带宽总线，用于连接复杂的多处理器系统和大容量存储器。它支持高并发性、乱序传输和优先级处理，具有更高的吞吐量和可扩展性。

使用场景：适用于连接高性能处理器、高带宽存储器和复杂的外设，如处理器核、高速存储器控制器等。

**ACE（AXI Coherency Extensions）：**

特点：ACE 是在 AXI 基础上添加的一组协议扩展，用于提供一致性缓存一致性的支持。它支持多处理器系统中的缓存一致性、数据一致性和内存一致性。

使用场景：适用于具有多个缓存的多处理器系统，用于实现缓存一致性和数据一致性。

**CHI（Coherent Hub Interface）：**

特点：CHI 是 AMBA 总线中的高性能、高一致性总线，用于连接大规模、高性能的处理器和存储器子系统。它支持高度可扩展性、高并发性和缓存一致性。

使用场景：适用于大规模、高性能的处理器集群和分布式存储器系统，用于实现高性能计算和数据处理。

这些 AMBA 总线协议根据系统的需求和性能要求提供了不同层次的灵活性和功能。通过选择合适的总线协议，设计人员可以满足不同类型的外设和存储器的连接需求，并提供高性能、低功耗和可扩展性的系统架构。

3、1) AXI（Advanced eXtensible Interface）总线包含以下独立的事务通道：

**读通道（Read Channel）：**用于从主设备（如处理器）读取数据的事务。包括读地址传输和读数据传输两个阶段。

**写通道（Write Channel）**：用于向主设备写入数据的事务。包括写地址传输和写数据传输两个阶段。

**写响应通道（Write Response Channel）**：用于从从设备（如存储器控制器）返回写操作的响应。响应包括完成状态和错误状态。

**保留通道（Reserved Channel）**：用于扩展协议或特殊需求而保留的通道。

关于为什么 AXI 协议不设置独立的读响应通道，原因主要有以下几点：

**简化设计**：将读响应通道与读通道合并可以简化总线设计和实现，减少通道数目和引脚数量，从而简化硬件电路的复杂度。

**提高效率**：读响应通常与读数据相关联，将读响应与读通道合并可以减少传输延迟，提高数据的读取效率。

**降低资源消耗**：独立的读响应通道会占用额外的物理资源，如引脚、线路和硬件逻辑等。通过合并读响应通道，可以降低资源消耗，使得总线设计更加紧凑和高效。

2) 在读/写传输事务中，通道的握手信号时序需要满足以下依赖关系：

**写地址传输与写数据传输之间的依赖关系**：写地址传输必须在写数据传输之前完成。这是因为写地址传输用于指示写入的目标地址，而写数据传输是实际写入数据的操作。如果写地址传输没有完成，写数据传输就无法确定写入的目标地址。

**读地址传输与读数据传输之间的依赖关系**：读地址传输必须在读数据传输之前完成。读地址传输用于指示读取的目标地址，而读数据传输是实际读取数据的操作。如果读地址传输没有完成，读数据传输就无法确定读取的目标地址。

**读数据传输与读响应传输之间的依赖关系**：读数据传输必须在读响应传输之前完成。读数据传输用于获取读取的数据，而读响应传输用于返回读操作的状态和结果。如果读数据传输没有完成，读响应传输无法包含正确的读取数据。

3) 在 AXI 总线协议中，突发传输（burst transfer）是指一系列连续的数据传输操作，以提高数据传输的效率和带宽利用率。突发传输允许在单个传输事务中传输多个数据或地址，减少了传输过程中的开销和延迟。

AXI 总线协议定义了几种突发传输类型，包括：

**固定突发传输（Fixed Burst）**：连续的数据传输操作，每个传输操作传输一个数据。在固定突发传输中，地址逐个增加，数据按顺序传输。

**递增突发传输（Incrementing Burst）**：连续的数据传输操作，每个传输操作传输一个数据。在递增突发传输中，地址递增，但是数据可以不按顺序传输。

**突变突发传输（Wrapping Burst）**：连续的数据传输操作，每个传输操作传输一个数据。在突变突发传输中，地址递增，但是当达到指定的边界时，地址会回卷到初始地址，形成循环传输。

**突发长度优先传输（Burst Length-Priority Burst）**：具有更高优先级的短突发传输。当具有更高优先级的传输到来时，可以中断正在进行的长突发传输，从而提高对高优先级传输的响应性。

2. ①直传摘 960 个数据包 每个数据包 10 个比特

系统波特率为 9600 bps.

② 系统有效数据传输速率  $760 \times 7 = 5320 \text{ bps}$ .

+ 1) RAID0 的 MTTF 为 4 块磁盘中最小的 MTTF, 即 N 小时

2) 让 2 块磁盘轮流工作.

5. 寻道时间: 磁头移动到目标磁道并消除转动时间. 旋转时间: 目标扇区随着盘片转动而经过磁头下(上)方所需的时间. 数据读取时间: 磁头完成读出或写入所需时间.

影响因素: 可用更精密的控制电路来减少

6. (1). 总容量 = 盘面  $\times$  磁道  $\times$  磁道信息量 =  $6 \times 240 \times 12 \text{ KB} = 17280 \text{ KB}$

(2).  $1400 \text{ r/min} = 70 \text{ r/s}$  数据传输速率  $70 \times 240 \times 12 \text{ KB/s} = 259200 \text{ KB/s}$

(3).  $\frac{90}{\frac{1}{70}} \approx \frac{1}{70} = \frac{1}{15} \text{ s}$ .

7. 磁盘 I/O 请求减少, 回平均到达率减小, 此时  $W = \frac{L}{X}$  W升高

如提升响应降低.

10. 可能会有带宽资源争抢情况发生. 存储器层次设计会减少处理器对内存的访问次

数, 及时获得最新的数据. 一些高级存储器或内存调度算法可以动态分配内存带宽

这样可以降低 CPU 和 DMA 之间的内存带宽竞争

资源, 进行优先级调度, 减少资源冲突.

次