

1. 串行总线与并行总线各有什么优缺点?试简述造成它们接口速率不同的原因。

答：串行总线是一种逐位传输数据的总线，它使用单个传输线路将数据一位一位地传输。优点包括：

- 1、更高的带宽利用率：串行总线可以使用较少的传输线路传输数据，因此可以更好地利用可用的带宽。
- 2、较长的传输距离：由于使用单个传输线路，串行总线在较长的距离上传输数据时受到的干扰较少，可以实现较长的传输距离。
- 3、灵活性：串行总线可以支持多种设备连接和通信，因为每个设备只需要使用一个传输线路。

但串行总线也存在一些缺点：

- 1、较高的延迟：由于数据逐位传输，串行总线的传输速率相对较低，这导致较高的传输延迟。
- 2、对时序要求较高：由于数据逐位传输，串行总线对时序要求较高，传输过程中的时钟同步需要精确控制。

并行总线是一种同时传输多个位的总线，它使用多个传输线路将数据同时传输。优点包括：

- 1、高传输速率：并行总线可以同时传输多个位，因此可以提供较高的传输速率。
- 2、低延迟：由于可以同时传输多个位，相比串行总线，并行总线的传输延迟较低。
- 3、对时序要求较宽松：并行总线对时序要求较宽松，因为数据同时传输，时钟同步要求不像串行总线那样严格。

然而，并行总线也存在一些缺点：

- 1、较低的带宽利用率：并行总线需要较多的传输线路，这限制了其在相同物理接口下可用的带宽，带宽利用率相对较低。
- 2、较短的传输距离：由于使用多个传输线路，数据在较长距离上传输时容易受到干扰，因此并行总线的传输距离相对较短。

造成串行总线和并行总线接口速率不同的原因主要有两个方面：

- 1、带宽限制：串行总线使用较少的传输线路，每个传输线路能够传输的位数有限，从而限制了总线的带宽。而并行总线使用多个传输线路，可以同时传输多个位，因此具有更高的总线带宽。
- 2、传输距离和干扰：串行总线由于使用单个传输线路传输数据，对干扰的抵抗能力较强，可以实现较长的传输距离。而并行总线使用多个传输线路，相互之间的干扰较大，因此在较长距离上传输数据时受到限制，传输距离较短。

7、简述磁盘控制电路如何通过决定请求的最优执行次序来减少磁盘访问用时?

答：1、请求调度算法：

磁盘控制电路使用请求调度算法来确定磁盘上请求的执行顺序。常见的请求调度算法包括先来先服务 (FCFS)、最短寻道时间优先 (SSTF)、扫描算法 (SCAN)、循环扫描算法 (C-SCAN) 等。这些算法根据磁盘头的位置和请求队列中的请求信息来决定下一个执行的请求，以尽量减少磁盘头的移动距离和寻道时间。

2、提前读取 (Read-Ahead)：

磁盘控制电路可以采用提前读取技术，在读取请求之前预先读取相邻的数据块到缓存中。这样，当后续请求需要访问相邻数据时，可以直接从缓存中获取，而不需要进行额外的磁盘访问。提前读取可以减少延迟和提高磁盘的吞吐量。

3、延迟写入 (Delayed Write)：

磁盘控制电路可以将写入请求暂时缓存起来，而不是立即执行写入操作。通过延迟写入，可以将多个写入请求合并成一个较大的写入操作，减少磁盘写入的次数，提高效率。

4、磁盘缓存 (Disk Cache)：

磁盘控制电路通常包含一个磁盘缓存，用于临时存储磁盘上的数据。磁盘缓存可以加速读取操作，当请求的数据在缓存中存在时，可以直接从缓存中获取，避免了磁盘访问的延迟。

8、试分析 RAID4 中的写入优化对于读取速度的影响。

答：写入优化在 RAID 4 中主要涉及到以下两个方面，对读取速度有不同的影响：

1、奇偶校验计算：

在 RAID 4 中，每个数据块写入时都需要计算相应的奇偶校验信息，并将其写入奇偶校验盘。这个计算过程涉及对多个数据块进行异或运算，然后写入奇偶校验盘。由于奇偶校验盘是单独的一个磁盘，写入操作会导致奇偶校验盘的更新。这意味着写入操作会增加额外的磁盘访问和计算开销。对于写入优化来说，它可以减少奇偶校验计算的次数，从而提高写入性能。然而，这个优化可能会增加读取操作的开销，因为在读取数据时需要重新计算奇偶校验信息来验证数据的完整性。

2、数据条带化 (Data Striping)：

RAID 4 使用条带化技术将数据分布在多个磁盘上。这样，读取操作可以同时从多个磁盘中读取数据块，从而提高读取速度。写入操作通常会涉及对多个磁盘进行并行写入，因为需要同时更新数据块和奇偶校验信息。因此，写入操作可以在一定程度上受益于条带化技术，从而提高写入速度。然而，由于需要保证数据和奇偶校验信息的一致性，写入操作可能需要进行更多的磁盘访问和计算，这可能会对读取速度产生一定的负面影响。

2解: (1) 波特率 $960 \times (1+7+1+1) = 9600$
(2) 系统 有效传输速率为 $960 \times 7 = 6720$

3解: (1) 数据包包括起始条件 + 地址字节 (7位地址 + 最低位控制(读写)) + 应答位 + 数据字节 + 应答位 + 停止条件

(2) I2C 只有 2 根线来进行通信: 时钟线 (SCL) 数据线 (SDA)。在半双工模式, 这两根线可以用于发送和接收数据, 但由于只有 2 根, 发送和接收不能同时进行。

(3) 起始条件: SCL 为高电平时, SDA 从高电平拉低。

停止条件: SCL 为低电平时, SDA 从低电平拉高。

4解: (1) $MTTF = \frac{N}{4}$ 小时 (RAID-0) (RAID-1)

(2) 将数据分散存在 2 个磁盘上, 再用 2 个磁盘存放相同的数据, 容量 $60G + 60G$

$$MTTF = \frac{1}{2} (N + N) = N \text{ 小时}$$

5解: 寻道时间是磁头从当前位置移动到目标磁道并消除抖动所需要的时间。

旋转时间是磁头移动到目标磁道后, 目标扇区随着盘片转动而经过磁头下方所需时间。

数据传输时间: 磁头完成读出或写入所需时间。

影响上述时间因素:

磁盘驱动器性能: 不同型号和品牌的驱动器具有不同的寻道时间、转速和数据传输速率。

磁盘物理结构: 磁盘的密度、磁道数量和扇区大小等会影响寻道时间和数据传输时间。

数据的位置和分布: 连续存储的数据所需时间小于随机分布的数据。

6解: (1) 总容量为 $6 \times 240 \times 12 \text{ KB} = 17280 \text{ KB}$

(2) 数据传输速率为 $\frac{5400}{60} \times 12 \text{ KB} = 1080 \text{ KB}$

13) 平均旋转时间为: $\frac{60}{5400} \div 2 = 0.0056s$

9) 解: 主要由寻磁时间、磁盘旋转时间、缓存有限性造成。这些因素在 n 较大时可以通过排序、调度和缓存等技术进行优化, 但当 n 较小时, 可以优化的幅度有限, 优化效果相对较小, 导致性能提升幅度下降。

10) 解: 会, 因为它们都通过总线来访问主内存, 因此可能会争抢内存带宽资源, 存储器层次设计的影响。

1. 高速缓存的存在: 当处理器需要访问内存时, 若数据恰好在缓存中, 就可以避免与 DMA 设备争抢内存带宽资源。

2. 存储器的带宽和速度: 若存储器带宽较窄速度较慢, 那么当处理器和 DMA 同时访问内存时, 争抢带宽资源的可能性较高; 而若带宽和速度都较大, 则不容易争抢带宽。

1、简述常见的总线仲裁机制, 它们各有什么优缺点、分别适用于什么场景?

答: 常见的总线仲裁机制包括集中式仲裁、分布式仲裁和链式仲裁。它们各有不同的优缺点, 并适用于不同的场景。

1、集中式仲裁:

集中式仲裁机制由一个中央仲裁器 (Arbiter) 负责协调总线访问权的分配。当多个设备请求总线访问时, 中央仲裁器根据预定义的优先级或算法来选择一个设备获得总线访问权。优点包括实现简单、仲裁过程可控, 适用于设备数量较少且对总线访问的时序要求较高的场景。

缺点是中央仲裁器成为系统的瓶颈，限制了总线的并发性能。

2、分布式仲裁：

分布式仲裁机制不依赖于中央仲裁器，而是由每个设备在总线上进行仲裁。设备之间使用冲突检测和协议来协调总线访问权。常见的分布式仲裁协议有 CSMA/CD（载波侦听多路访问/碰撞检测）和 CSMA/CA（载波侦听多路访问/碰撞避免）。优点包括可扩展性强、减少了中央仲裁器的瓶颈，并支持高并发性能。缺点是需要较复杂的协议和冲突检测机制，容易产生冲突和延迟，适用于设备数量较多但对总线访问的时序要求较低的场景。

3、链式仲裁：

链式仲裁机制将总线连接的设备排成链状，设备按顺序依次请求总线访问。每个设备在完成自己的操作后，将总线访问权传递给下一个设备。优点包括实现简单、无需中央仲裁器，适用于设备数量较少且对总线访问的时序要求较低的场景。缺点是总线利用率较低，因为每个设备只能顺序访问总线，无法并发执行操作。

综上所述，集中式仲裁适用于设备数量较少且对总线访问的时序要求较高的场景；分布式仲裁适用于设备数量较多且对总线访问的时序要求较低的场景；链式仲裁适用于设备数量较少且对总线访问的时序要求较低的场景。

2、简要描述 AMBA 总线中 APB、AHB、AXI、ACE 及 CHI 等总线协议的特点和使用场景。

答：AMBA (Advanced Microcontroller Bus Architecture) 是一种广泛应用于片上系统 (SoC) 设计的总线架构，其中包括多个总线协议，如 APB (Advanced Peripheral Bus)、AHB (Advanced High-performance Bus)、AXI (Advanced eXtensible Interface)、ACE (AXI Coherency Extension) 和 CHI (Coherent Hub Interface)。下面是它们的特点和使用场景的简要描述：

1、APB (Advanced Peripheral Bus)：

APB 是一种简单的低功耗、低带宽总线协议，主要用于连接较简单的外设，如配置寄存器、控制器等。APB 适用于对带宽要求不高的外设，如 GPIO（通用输入/输出）、UART（通用异步收发传输器）等。

2、AHB (Advanced High-performance Bus)：

AHB 是一种高性能的总线协议，支持高带宽和多主设备的连接。它具有分时复用、分级优先级和流水线传输等特性，支持多个主设备并发地访问总线。AHB 适用于连接性能较高的外设，如存储控制器、DMA 控制器等。

3、AXI (Advanced eXtensible Interface)：

AXI 是一种高性能、高带宽的总线协议，具有低延迟、高吞吐量和可扩展性等特点。AXI 支持多主设备、多个读写通道和乱序传输，适用于连接复杂的外设和高性能处理器核心，如图形处理器、网络接口、多核处理器等。

4、ACE (AXI Coherency Extension)：

ACE 是在 AXI 基础上扩展的一种协议，用于支持高级一致性 (cache coherence) 的处理器系统。ACE 协议提供了处理器之间的缓存一致性和事务原子性的支持，适用于多处理器系统，如多核处理器和对称多处理器 (SMP) 系统。

5、CHI (Coherent Hub Interface)：

CHI 是一种面向高性能的多处理器系统设计的新一代高速片上总线协议。它提供了高度可扩展性、高带宽和一致性支持。CHI 支持大规模多处理器系统和高性能计算，适用于数据中心、高性能计算领域以及需要高性能片上通信的复杂系统。

总的来说，APB 适用于低带宽的外设，AHB 适用于连接性能较高的外设，AXI 适用于连接复杂的外设和高性能处理器核心，ACE 适用于支持高级一致性的多处理器系统，CHI 适用于高性能的多处理器系统和高性能计算场景。根据系统的性能要求和设计需求，选择适合的

AMBA 总线协议来实现片上系统的连接与通信。

3、调研 AXI 总线标准并回答以下问题。

1) AXI 总线包含哪些独立的事务通道?为什么协议不设置独立的读响应通道?

2) 简要描述在读/写传输事务中,通道的握手信号时序需要满足怎样的依赖关系?为什么要设置这样的约束?

3)什么是 AXI 的突发传输?有哪些突发传输类型?

答: 1) AXI 协议没有独立的读响应通道,而是将读响应与读通道合并在一起传输,即通过读通道返回读数据和读响应。这样设计的原因主要有以下几点:

1、精简总线信号: 合并读响应到读通道可以减少总线上的信号线数量,简化物理连接和实现复杂度。

2、降低延迟: 合并读响应可以减少传输延迟。在读操作中,读数据和读响应可以在同一个通道上同时传输,减少了传输的阶段和延迟。

3、增强并发性能: 合并读响应可以提高总线的并发性能。在读操作中,主设备可以同时发起多个读事务,而不需要等待读响应返回。

综上所述,将读响应与读通道合并在一起的设计可以简化总线结构、降低延迟并提高并发性能,从而更好地满足高性能系统的需求。

2) 在 AXI 总线标准中,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:

1、读传输事务的握手信号时序依赖:

主设备发出读请求 (valid 信号置高) 后,从设备必须在一定延迟内响应 (ready 信号置高)。

主设备收到从设备的响应后,必须在一定延迟内结束读传输 (valid 信号置低)。

从设备在响应期间必须保持有效的读数据 (read data 信号有效)。

2、写传输事务的握手信号时序依赖:

主设备发出写请求 (valid 信号置高) 后,从设备必须在一定延迟内响应 (ready 信号置高)。

主设备收到从设备的响应后,必须在一定延迟内结束写传输 (valid 信号置低)。

这些依赖关系的约束是为了确保数据传输的正确性和一致性,以及保持总线的稳定和可靠性。具体原因如下:

1、数据一致性: 通过依赖关系的约束,确保在读传输中,主设备只在从设备已经准备好数据时才接收读数据,避免了不一致的数据读取。在写传输中,主设备只在从设备准备好接收数据时才发送写数据,避免了数据写入到不可用的位置。

2、时序正确性: 通过依赖关系的约束,保证了读/写事务的时序按照规定的顺序进行,避免了传输冲突和数据丢失。主设备和从设备在收到对方的握手信号后才进行下一步操作,保证了传输的顺序和正确性。

3、总线稳定性: 依赖关系的约束确保了主设备和从设备在传输过程中的同步和平衡,避免了过早或过晚的握手信号导致的总线不稳定或死锁情况。

综上所述,握手信号时序的依赖关系约束在 AXI 总线标准中起到了确保数据一致性、时序正确性和总线稳定性的重要作用,保障了系统的可靠性和正确性。

3) 在 AXI (Advanced eXtensible Interface) 总线中,突发传输 (burst transfer) 是指一次传输中连续读取或写入多个数据项的操作。通过使用突发传输,可以在减少总线开销的同时提高数据传输效率。

AXI 总线定义了几种突发传输类型,包括:

1、固定突发传输 (Fixed Burst)：在固定突发传输中，连续的数据项在传输开始时已经确定。主设备在发起读或写传输时，指定了起始地址和传输长度，从设备按照固定的地址顺序传输数据项。

2、递增突发传输 (Incrementing Burst)：在递增突发传输中，连续的数据项的地址是按照递增的顺序生成的。主设备在发起读或写传输时，指定了起始地址和传输长度，从设备根据起始地址和数据项的大小生成递增的地址。

3、轮换突发传输 (Wrap Burst)：在轮换突发传输中，连续的数据项的地址在一次传输中循环轮换。主设备在发起读或写传输时，指定了起始地址、传输长度和轮换长度，从设备按照起始地址和轮换长度循环轮换传输数据项。

4、不规则突发传输 (Unaligned Burst)：在不规则突发传输中，传输的数据项的地址不是按照固定的规则或递增方式生成的。主设备在发起读或写传输时，可以指定不连续的地址来传输数据项。

通过使用这些突发传输类型，可以根据具体应用需求选择最适合的传输模式，从而提高数据传输效率和总线利用率。不同类型的突发传输在处理不同的数据访问模式时具有不同的优势和适用性。