

第十五周作业

姓名：王鑫尧 学号：21307140029

5.1 答：串行总线和并行总线是两种常见的数据传输方式。

对于串行总线，优点为：

（1）有较高的数据传输距离：串行总线能够在较长距离上传输数据，因为串行传输可以更好地抵抗信号失真和干扰；（2）简化布线和连接：串行总线只需要少量的传输线，布线和连接相对简单；（3）可扩展性：串行总线通常具有较好的可扩展性，可以支持更高的带宽需求。缺点为：（1）速度较慢：相比于并行总线，串行总线的传输速度较慢，因为它每次只能传输一个比特的数据（2）较高的传输延迟：由于串行总线需要一个比特一个比特地传输数据，它的传输延迟相对较高。

对于并行总线，优点为：（1）较快的传输速度：并行总线可以同时传输多个比特的数据，因此它的传输速度相对较快（2）较低的传输延迟：由于并行总线同时传输多个比特的数据，它的传输延迟相对较低。

缺点为：（1）布线和连接复杂：并行总线需要多个传输线，布线和连接相对复杂，特别是在高位数的情况下。（2）信号失真和干扰：并行总线在较长距离上的传输容易受到信号失真和干扰的影响。

造成串行总线和并行总线接口速率不同的原因主要有两点：

（1）传输线数量：并行总线需要多个传输线同时传输数据，因此它的接口速率受到布线和连接的限制。在实践中，增加并行总线的位数会导致传输线的数量成倍增加，限制了接口速率的提高。

（2）信号干扰和失真：随着并行总线的位数增加，信号干扰和失真的问题也会增加。并行总线上的信号可能会相互干扰，造成传输的误差，从而限制了接口速率的提高。为了解决这个问题，需要采取一些技术手段，如使用屏蔽和终止技术来减少干扰和失真。而串行总线由于只有一个传输线，可以更好地控制信号质量，提高接口速率。

5.3 答：

（1）I2C 的数据包由以下几个部分构成：起始条件（Start Condition）：由主设备发送的一个低电平信号，表示开始传输数据包。地址字节（Address Byte）：包含目标设备的地址信息，由主设备发送。数据字节（Data Bytes）：包含要传输的数据，由主设备发送或接收。读/写位，确认是在读还是写。停止条件（Stop Condition）：由主设备发送的一个高电平信号，表示结束传输数据包。

（2）I2C 是半双工的，意味着在同一条总线上数据的传输是单向的，只能在发送和接收之间切换。这是因为 I2C 总线只有两根线：时钟线（SCL）和数据线（SDA）。数据的传输是通过这两根线上的电平变化来完成的。在 I2C 中，主设备负责控制总线并发送数据，而从设备则负责接收数据或发送应答信号。由于总线上的信号是共享的，所以只能通过在发送和接收之间切换来实现半双工通信。

（3）I2C 传输的起止条件如下：

起始条件：起始条件由主设备发送，它是将数据线（SDA）从高电平切换到低电平时的信号。在起始条件之后，主设备发送从设备的地址和读/写位来选择要进行通信的从设备。

停止条件：停止条件由主设备发送，它是将数据线（SDA）从低电平切换到高电平时的信号。在停止条件之后，数据线上的电平变化不再有意义，通信结束。

起始条件和停止条件都由主设备产生，并且在数据传输之前和之后使用。这些条件用于标识数据包的开始和结束，同时通知其他设备在总线上释放控制权或准备接收数据。

第十五周作业

5.2 解:

(1) \because 系统每秒 960 个数据包

每个数据包有 $7+1+1+1=10$ 位

\therefore 系统每秒传输位数为 9600 位, 波特率为 9600 Baud

(2) \because 每个数据包只有 7 位有效数据位

\therefore 系统的有效数据传输速率为 $960 \times 7 = 6720$ 位/每秒

5.4 解:

(1) 对于 RAID-0, 数据均匀地分散, 存储在多个磁盘上, 磁盘故障概率相互独立。

$$\therefore \text{MTTF(RAID-0)} = \frac{1}{4 \times \frac{1}{\text{MTTF(Disk)}}} = \frac{N}{4}$$

(2) 可以采用 RAID-1 方案, 将 4 块磁盘分为两组, 每组 2 个, 其中一个为主磁盘而另一个作为镜像磁盘。数据同时写入主磁盘和镜像磁盘, 使数据可靠性更高。

当一个磁盘发生故障时, 另一个磁盘上的数据仍然可用, 提高系统的可靠性, 达到更大的 MTTF, 此种方案下 $\text{MTTF(RAID-1)} = 2N$

5.5 答:

寻道时间: 磁头从当前磁道移动到目标磁道所需的时间; 旋转时间: 磁头移动到目标磁道后, 目标扇区随盘片转动而经过磁头下/上方所需时间; 磁头完成读出或写入所需用的时间。

影响因素: 寻道时间受盘片的物理结构和磁头的定位能力影响, 旋转时间受磁盘的转速影响, 数据传输时间受磁盘的传输速率和数据大小影响。

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

(2) \because 每个盘面含 240 个磁道, $5400 \text{ r/min} = 90 \text{ r/s}$

\therefore 数据传输速率 $= 90 \times 240 \times 12 \text{ KB/s} = 259200 \text{ KB/s} = 259.2 \text{ MB/s}$

(3) 磁盘的平均旋转时间为一个完整的磁盘旋转时间的一半,

$$\therefore T = \frac{60 \text{ s}}{5400} \times \frac{1}{2} = 0.00556 \text{ s} \approx 5.56 \text{ ms}$$

5.7 答:

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主要方式是通过磁盘调度算法。磁盘调度算法负责管理和调度磁盘访问请求的执行顺序, 以最小化磁盘的寻道时间和旋转时间, 从而提高磁盘的访问效率。

几种常见的磁盘调度算法有: 先来先服务 (FCFS): 按照请求的到达顺序进行处理, 不考虑寻道时间和旋转延迟。这种算法简单直观, 但可能导致磁盘头在磁道之间频繁移动, 造成较长的平均访问时间; 最短寻道时间优先 (SSTF): 选择与当前磁头位置最接近的请求进行处理。这样可以最小化寻道时间, 但可能会导致一些请求长时间等待, 产生饥饿问题; 扫描算

法 (SCAN): 磁头按一个方向移动, 处理该方向上的所有请求, 直到到达磁盘的边缘, 然后改变方向继续处理。这种算法可以避免饥饿问题, 但可能导致某些请求等待时间较长; 循环扫描算法 (C-SCAN): 类似于 SCAN 算法, 但在到达磁盘边缘后直接返回到起始位置, 而不是改变方向。这样可以确保每个请求都有机会被服务, 减少等待时间。

通过选择合适的磁盘调度算法, 并根据当前的磁盘状态和请求队列, 磁盘控制电路可以动态地决定请求的最优执行次序, 以最小化磁盘访问时间。这样可以提高磁盘的响应速度和整体性能, 减少等待时间和延迟, 从而提升系统的整体效率。

5.8 答:

在 RAID 4 中, 写入优化是通过将数据块划分为多个数据块, 并分散存储在不同的磁盘上, 同时使用一个专用的奇偶校验盘来存储奇偶校验信息。这种方式可以提高写入操作的性能和并发性, 所以说在 RAID 4 中, 写入优化对于读取速度的影响可以是正面的, 但也可能带来一些限制。

RAID 4 采用块级别的条带化, 其中数据被分散存储在不同的磁盘上, 而校验信息则存储在单独的校验盘上。在写入数据时, RAID 4 需要执行以下操作: 读取原始数据块和相应的校验数据块, 计算新数据块和校验数据块的校验信息, 写入新数据块和更新的校验数据块。这意味着在写入操作中, RAID 4 需要访问多个磁盘来完成数据和校验信息的读取和写入。因此, 写入操作的性能会受到磁盘访问的延迟和带宽限制。所以, 在 RAID 4 中, 写入优化可以提高写入性能, 但仍需要注意对缓存的管理以及对异常情况的处理, 以避免对读取速度造成不利影响。

5.9 解:

根据给定的公式 $W = \frac{1}{\mu - \lambda}$, 可知当磁盘 I/O 请求减少时, λ 值减小, 公式中分母 $\mu - \lambda$ 变大, 平均响应时间 W 减小, 表示磁盘队列系统的性能会相应地提升, 响应时间变短。

但随着磁盘 I/O 请求减少, 提升幅度也随之减小是因为磁盘服务率 μ 的限制。当磁盘 I/O 请求减少到接近磁盘的服务率时, 磁盘的处理能力会限制系统, 无法进一步提升性能。所以在优化磁盘队列系统的性能时, 需要考虑多方面因素。

5.10 答:

DMA 设备和处理器都需要使用内存带宽资源进行数据传输和访问, 可能同时请求访问内存就会争夺内存带宽资源。DMA 可直接访问系统内存, 不需要处理器介入, 而处理器也通过内存总线来访问内存, 所以可能会竞争内存带宽资源。

存储器层次设计的优劣对此问题的影响在于, 存储器层次设计良好则可减轻竞争, DMA 设备和处理器可以从各自的缓存中获取所需的数据, 减少对内存的访问次数, 从而减少内存带宽竞争。若存储器层次设计不合理, 则 DMA 和处理器对内存访问更频繁, 会导致更多的内存竞争。

6.1 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，适用于不同的场景。

集中式仲裁:

优点: 集中式仲裁由中央仲裁器负责决定哪个设备能够获得总线的控制权, 具有简单、可控性好的特点。

缺点: 需要一个中央仲裁器, 会增加系统的复杂度和成本。同时, 所有设备的仲裁请求都要通过中央仲裁器, 可能导致性能瓶颈。

分布式仲裁:

优点: 分布式仲裁允许每个设备独立地判断总线是否可用, 并发送请求来争夺总线的控制权, 从而减轻了中央仲裁器的负担, 提高了系统的并行性和性能。

缺点: 分布式仲裁需要设备之间相互通信来判断总线的可用性, 增加了通信开销和延迟。此外, 分布式仲裁的实现较为复杂, 需要解决设备之间的冲突问题。

链式仲裁:

优点: 链式仲裁使用一个优先级链来确定设备之间的仲裁关系, 具有简单、可扩展性好的特点。每个设备只需要知道其前一个设备的状态, 无需知道其他设备的情况。

缺点: 链式仲裁的性能受限于链中设备的数量和总线上设备的响应时间。长链可能会导致较长的延迟, 而短链可能会导致设备之间的冲突增多。

基于优先级的仲裁:

基于优先级的仲裁机制根据设备的优先级确定设备的访问顺序。该机制的优点是简单且易于实现, 适用于具有明确优先级关系的设备。然而, 基于优先级的仲裁可能导致低优先级设备长时间等待, 造成资源浪费。

时间分割仲裁:

时间分割仲裁将总线时间分割成固定的时间片段, 每个设备在自己的时间片段内独占总线访问权。该机制的优点是公平性和预测性好, 每个设备都能在自己的时间片段内获得公平的访问机会。然而, 时间分割仲裁可能导致总线带宽的浪费, 特别是当设备数量较少或设备的访问模式不规律时。

选择适当的仲裁机制取决于系统的要求和约束条件:

集中式仲裁适用于规模较小的系统, 且总线访问冲突较少的场景。

分布式仲裁适用于需要提高系统并行性和性能的大规模系统, 但需要考虑通信开销和延迟。

链式仲裁适用于中等规模的系统, 对实时性要求不高, 且可以通过优先级链简化仲裁逻辑的场景。

需要根据系统的具体需求和设计目标, 综合考虑系统的规模、性能要求、复杂度以及成本等因素, 选择合适的仲裁机制。

6.2 答:

AMBA (Advanced Microcontroller Bus Architecture) 是由 ARM (Advanced RISC Machines) 公司提出的一种总线架构, 用于构建嵌入式系统中的互联结构。AMBA 总线协议包括了多个版本, 其中比较常见的有 APB、AHB、AXI、ACE 和 CHI。

APB (Advanced Peripheral Bus):

特点: APB 是 AMBA 中的低功耗、低复杂性外设总线。它采用简单的请求-应答传输模型, 并使用时钟同步。数据传输宽度可配置, 一般为 8 位或 16 位。

使用场景: 适用于低速外设, 如 UART (串口通信) 等。

AHB (Advanced High-performance Bus):

特点: AHB 是 AMBA 中的高性能总线, 支持多主设备和多从设备, 并采用分片传输和突发传

输机制。它支持优先级和分片访问，具有较高的吞吐量和低延迟。

使用场景：适用于中等复杂性的外设和嵌入式处理器之间的连接，如存储器控制器、DMA 控制器等。

AXI (Advanced eXtensible Interface) :

特点：AXI 是 AMBA 中的高性能、高带宽总线协议。它支持多通道、乱序传输和突发传输，具有高度的可扩展性和灵活性。AXI 有多个版本，包括 AXI4、AXI4-Lite、AXI4-Stream 等。

使用场景：适用于高性能、高带宽的系统内部连接，如高速存储器、处理器互联、高性能外设等。

ACE (AXI Coherency Extensions) :

特点：ACE 是在 AXI 基础上增加了一致性协议的扩展，用于处理多核处理器和共享缓存系统中的一致性问题。它支持缓存一致性和事务一致性，并提供了高度的可扩展性和灵活性。

使用场景：适用于多核处理器系统和具有高度一致性要求的共享缓存系统。

CHI (Coherent Hub Interface) :

特点：CHI 是 AMBA 的最新一代总线协议，为大规模多核系统提供了高性能、高一致性和可扩展性。它支持多级缓存一致性和高度的灵活性。

使用场景：适用于大规模多核处理器系统和具有复杂一致性要求的高性能系统。

6.3 答：

(1) AXI 总线包含以下独立的事务通道：

读地址通道：用于发起读取事务的地址传输。

写地址通道：用于发起写入事务的地址传输。

读数据通道：用于传输读取事务的数据。

写数据通道：用于传输写入事务的数据。

写响应通道：用于传输写入事务的响应信号。

协议没有设置独立的读响应通道的原因是，读取事务的响应可以通过读数据通道返回。这样设计可以减少总线上的信号线数量和复杂性，提高总线的效率和可扩展性。

(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：

在读传输事务中，主设备发出读请求后，从设备在时钟周期内进行响应，提供所请求的数据。

在写传输事务中，主设备发出写请求后，从设备在后续的时钟周期内进行写入操作，并在时钟周期结束时发送写响应。

这样的时序依赖关系是为了确保主设备和从设备之间的正确数据传输和协调。握手信号的时序约束确保了读/写操作的一致性和正确性。确保读请求和读响应与写请求和写响应之间的顺序正确，并防止数据的混乱和错误。

(3) AXI 的突发传输是指在一次地址传输中连续传输多个数据项的方式。突发传输可以提高数据传输的效率和带宽利用率。

AXI 有以下突发传输类型：

固定突发传输：在一次地址传输中连续传输固定数量的数据，例如连续读取或写入连续的内存地址。

递增突发传输：在一次地址传输中连续传输递增的数据，例如递增读取或写入连续的内存地址。

跳跃突发传输：在一次地址传输中连续传输数据，当达到设定的最大数量后，地址回绕到起始地址继续传输，例如循环读取或写入连续的内存地址。

突发传输可以提高数据传输效率，减少总线开销，并适应高带宽数据传输的需求。不同的突发传输类型可以根据具体应用的需求选择。