

第十四次作业

5-1

串行就是数据是一位一位的发送，并行是数据一组一组的发送的。

只要有其中一位出错，数据就不能够正确传输过去，需要重新传输。而串行数据是一位一位的传，位与位之间没有联系，不会因为这位有错误，使下一位不能传输。

并行总线由于是多个数据同时传输，需要考虑数据的协同性，这就导致了并行传输的频率不能做的很高。相对的，串行总线只有一条链路，就可以把频率做的很高，提高传输速度，速度提高了就能够弥补一次只能传输一个数据的缺陷。

此外，并行总线两根相邻的链路其数据是同时传输的，这就会导致它们彼此之间会产生严重干扰，并行的链路越多，干扰越强。因此并行总线需要加强抗干扰的能力，否则传输过程中数据就可能被损坏。如果传输过程中数据故障了，就需要重新对齐数据再传输。而串行总线如果一个数据出错了，只需要重新传输一次就好了，由于串行总线频率高，很快就可以把错误数据重新传输过去。

再次，由于并行总线是多链路一块传输数据，就需要很多线，接口需要很多针脚。

5-3

I2C 数据包由两个部分组成：地址和数据。首先是7 位或10 位设备地址，根据设备的操作模式（读/写）后面跟着一个读/ 写位。然后发送数据字节，每个字节8 位，并带有一个ACK 信号以确认接收。传输完成时，主设备可以使用STOP 条件结束传输。

I2C 是半双工的，因为在I2C 总线上，数据可以在同一时间只能被一个设备发送或接收。当一个设备发送数据时，它必须等待该数据被接收到确认后才能继续发送。如果多个设备同时尝试发送数据，则会发生冲突，导致总线出现电平错误，这就是所谓的总线抢占。

I2C 传输的起止条件如下：

起始条件：主设备将SDA 线设置为低电平，而SCL 线保持高电平。这表示一个新的传输正在开始。

停止条件：主设备将SDA 线设置为高电平，而SCL 线保持高电平。这表示传输结束。

重复启动条件：主设备在不释放总线的情况下生成一个起始条件。这用于在不断开总线的情况下连接到其他设备。

5-7

磁盘控制电路可以通过决定磁盘访问请求的最优执行次序来减少磁盘访问用时。

首先，磁盘控制电路将接收到的访问请求放在队列中，并根据请求的先后顺序，以及磁盘内部的物理结构，决定最佳的访问顺序。这个过程被称为磁盘调度算法。

常见的磁盘调度算法包括：

先来先服务（FCFS）：按照请求的先后顺序执行，不考虑访问的位置。

最短寻找时间优先（SSTF）：选择当前距离磁头最近的请求进行执行。

扫描算法（SCAN）：按照磁头移动的方向依次执行请求，当到达磁盘边界时改变方向。

循环扫描算法（C-SCAN）：类似于扫描算法，但当到达磁盘边界时直接返回另一端，不回头执行未完成的请求。

电梯算法（LOOK 或C-LOOK）：从当前位置开始向一个方向执行，直到没有更多请求为止，然后改变方向继续执行。

通过选择合适的磁盘调度算法，磁盘控制电路可以使磁盘访问的时间和距离最小化，从而提高系统效率。

5-8

RAID4 是一种磁盘阵列存储技术，它使用奇偶校验来实现数据冗余和容错，其中一个磁盘被用于存储奇偶校验信息。当写入操作发生时，RAID4 首先计算新数据块与对应奇偶校验块的异或值，并将结果写入相应磁盘上。

在RAID4 中进行写入操作时，由于需要同时更新数据块和相应的奇偶校验块，因此写入速度较慢。为了优化写入速度，RAID4 引入了写入缓存的机制。具体而言，当写入操作发生时，RAID4 会先将数据块和奇偶校验块的异或值写入写入缓存中，然后立即返回成功标志，表示写入操作已经完成。随后，RAID4 会在后台进程中，将写入缓存中的数据写入真正的磁盘中。这种机制可以提高写入速度，但也带来了一些问题。

具体而言，由于写入缓存是异步执行的，因此在某些情况下，读取操作可能会读取到尚未写入真正磁盘的数据。例如，在写入缓存中的数据还没有写入真正磁盘之前，系统崩溃或掉电了。这种情况下，读取操作就会读取到不一致的数据，从而影响读取速度。

因此，在RAID4 中进行写入优化可以提高写入速度，但也会对读取速度造成一定的影响。为了避免这种问题，RAID4 通常会使用写入缓存和校验点机制来控制异步写入操作的执行时间，并确保数据的一致性。

6-1

在多个设备共享同一个总线时，需要通过总线仲裁机制来协调各个设备的访问。常见的总线仲裁机制有以下几种：

集中式仲裁：由一个中心节点来管理总线上的访问请求，其他设备向中心节点发送请求，由中心节点确定哪个设备可以访问总线。优点是简单易实现，并且能够确保公平性；缺点是存在单点故障和串行化问题，并且不适合大型系统。

分布式仲裁：每个设备都有权决定自己是否要访问总线，如果多个设备同时请求访问，则通过一定的协议进行竞争，最终获胜者可以获得总线访问权。优点是高效并行，并且不会出现单点故障；缺点是可能会出现饥饿问题。

基于仲裁器的仲裁：将仲裁器作为一个独立的硬件模块，它根据事先约定好的优先级来分配总线访问权。优点是具有可扩展性和灵活性，并且能够快速响应紧急请求；缺点是复杂度较高。

6-2

AMBA 是一种总线协议，主要用于连接处理器、存储器和外设等组件。常见的AMBA 总线协议包括APB、AHB、AXI、ACE 和CHI 等。

APB：针对低带宽和低功耗设备，如控制寄存器等。其特点是简单、低延迟，但带宽较小。

AHB：针对中等带宽的设备，如存储器控制器等。其特点是高效、灵活，支持多主设备和分支总线，但复杂度较高。

AXI：针对高带宽和高性能设备，如处理器和高速IO 接口等。其特点是高吞吐量、低延迟、支持多种传输类型和多主设备，并且具有良好的扩展性。

ACE：在AXI 基础上增加了缓存一致性协议，主要用于处理器互连和SoC 内部通信。

CHI：是ARM 最新推出的一种高性能互连总线协议，用于连接多核处理器和高速IO 接口等，具有更高的吞吐量和更好的可扩展性。

6-3

AXI 总线包含了5 个独立的事务通道，分别是：读数据通道、写数据通道、读地址通道、写地址通道和写返回通道。读操作和写操作都需要发送地址和数据，因此需要分别设置读地址通道、写地址通道、读数据通道和写数据通道。而读响应只需要告诉主设备是否成功读取数据即可，因此它被包含在读数据通道中。

1. 在读事务中，读通道（Read channel）的读请求信号（ARVALID）必须在读地址信号（ARADDR）稳定之后有效。这确保了在发送读请求时，地址已经准备好。
2. 读通道的读请求信号（ARVALID）有效后，读应答信号（RVALID）会在一定延迟后变为有效。这个延迟是为了给被读取的设备足够的时间来处理读请求，并返回数据。
3. 在写事务中，写通道（Write channel）的写请求信号（AWVALID）必须在写地址信号（AWADDR）稳定之后有效。这确保了在发送写请求时，地址已经准备好。
4. 写通道的写请求信号（AWVALID）有效后，写数据信号（WVALID）必须在一定延迟后变为有效。这个延迟是为了给被写入的设备足够的时间来接收写请求并准备好接收数据。
5. 写通道的写数据信号（WVALID）和写应答信号（BVALID）之间也存在一定的延迟，以确保设备有足够的时间来处理接收的数据并发送应答。

这些依赖关系的设置是为了确保在传输事务中的各个阶段，相关的信号按照正确的顺序传递和响应。通过满足这些约束，可以确保数据的正确性和一致性，并确保系统的可靠性和稳定性。此外，这些约束还有助于在高速和并发操作的系统中实现流水线操作，提高系统的性能和效率。

在AXI总线中，突发传输（burst transfer）是指在一个事务中连续传输多个数据或连续读取/写入多个地址的操作。它允许在一次事务中传输多个数据或地址，减少了总线开销和传输延迟，提高了总线带宽和效率。AXI总线支持以下几种突发传输类型：

1. 固定突发（Fixed Burst）：在一个事务中连续传输固定数量的数据或连续读取/写入固定数量的地址。这种突发传输类型使用一组固定的地址，以及一个固定的传输长度。
2. 递增突发（Incrementing Burst）：在一个事务中连续传输递增的数据或连续读取/写入递增的地址。这种突发传输类型使用一个起始地址，以及一个递增值（通常是数据宽度），用于生成连续的地址序列。
3. 周期突发（Wrap Burst）：在一个事务中连续传输一组数据或连续读取/写入一组地址，然后从事务的起始地址重新开始传输。这种突发传输类型适用于循环访问存储器或使用环形缓冲区的场景。

以上突发传输类型的选择取决于应用的需求和设计要求。它们可以根据具体的数据传输模式和存储器结构进行选择，以最大程度地提高总线的带宽和效率。

5.2 (1) 一个包有 $1+7+1+1=10 \text{ bit}$
则波特率 $960 \times 10 = 9600$

(2) $7 \times 960 = 6720$

5.4 (1) RAID 100 为4倍读写速率 则

$$MITF = \frac{N}{4}$$

(2) 组 RAID1, 有100h, 两块冗余, $MITF = \frac{(N-2)^6}{N^4}$

5.5 (1) 寻道: 磁头旋转到目标的时间

旋转: 磁盘转速

传输时间 硬盘读写时间, 与转速有关

5.6 (1) $6 \times 240 \times 12 \text{ K} = 17280 \text{ KB}$

(2) $5400 \div 60 \times 6 \times 240 \times 12 = 1555200 \text{ KB/s}$

(3) $60 \div 5400 = 11.1 \text{ ms}$

5.9 I/O 减少, 请求排队时间降低, 则提升不明显

5.10 会, DMA有几种模式来处理这种情况, 缓存的设计使得DMA修改后, 缓存也会失效。