

5-1:

串行总线和并行总线

①串行总线

优点:

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;

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;

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:

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

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。

②并行总线

优点:

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;

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:

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;

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。

一般需要较高传输速度、较低延迟时使用并行, 长距离传输、要求稳定性强、布线简单时使用串行。

5-3:

(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, 所以只能通过发送和接收之间切换来实现半双工通信。

(3) 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。

5-7:

有特定的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。

常见的磁盘调度算法有:

先来先服务 (FCFS): 仅按照请求的到达顺序进行处理;

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优先处理;

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;

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。

5-8:

RAID 4 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。

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。

RAID 4 访问多个磁盘来完成数据和校验信息的读取和写入。并行的读写可以增加速度但是操作的性能也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。

6-1:

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

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。

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。

6-2:

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。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

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:

(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。

读取事务的响应可以通过读数据通道返回，故不设置独立的读响应通道，这样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。

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。

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。

手写题见此文件下一页

5-2. 解: (1) 波特率为 $(1+7+1+1) \times 960 = 9600$

(2) 有效数据传输速率: $7 \times 960 = 6720 \text{ bit/s}$

5-4. 解: (1) 所有数据简单分开存入4个盘, 4个盘被使用几率相同
故MTTF为4M小时。

(2) 采用RAID-2, 2块盘用于存储, 2块盘用于纠错

5-5. 解: ① 寻道时间: 磁头从当前位置移动到目标磁道并消除抖动所需时间
影响因素: 磁盘半径、磁道细密度、控制电路精度、磁头速度。

② 旋转时间: 磁头到目标磁道后, 目标扇区旋转到磁头上/下方所需时间

影响因素: 转动速度, 控制电路精度

③ 数据传输时间: 磁头完成读出或写入所需时间

影响因素: 磁头灵敏度, 磁头与盘片距离

5-6: (1) 总容量 $12\text{KB} \times 240 \times 6 = 17280\text{KB}$

(2) 传输速率 $5400\text{r}/60\text{s} \times 12\text{KB} = 1080\text{KB/s}$

③ 平均等待时间 $60 \div 2 \div 5400 = 5.55\text{ms}$ 。

5-9. 解: $W = \frac{1}{\mu - \lambda}$ $\frac{dW}{d\mu} = -\frac{1}{(\mu - \lambda)^2}$

性能提升会提高 μ , 每秒 I/O 请求为 λ

所以 λ 越大, $\frac{dW}{d\mu}$ 越负, W 下降越快,

同样的 μ 提升 W 下降更多,

所以 I/O 请求减少时 W 下降幅度更低。

5-10. 解: 看具体的设计, 如突发模式DMA可能会争抢, 透明模式DMA则不会
若存储器分层设计得好, DMA与处理器并不操作同一层次存储器, 可以有效解决带宽争抢的问题。