

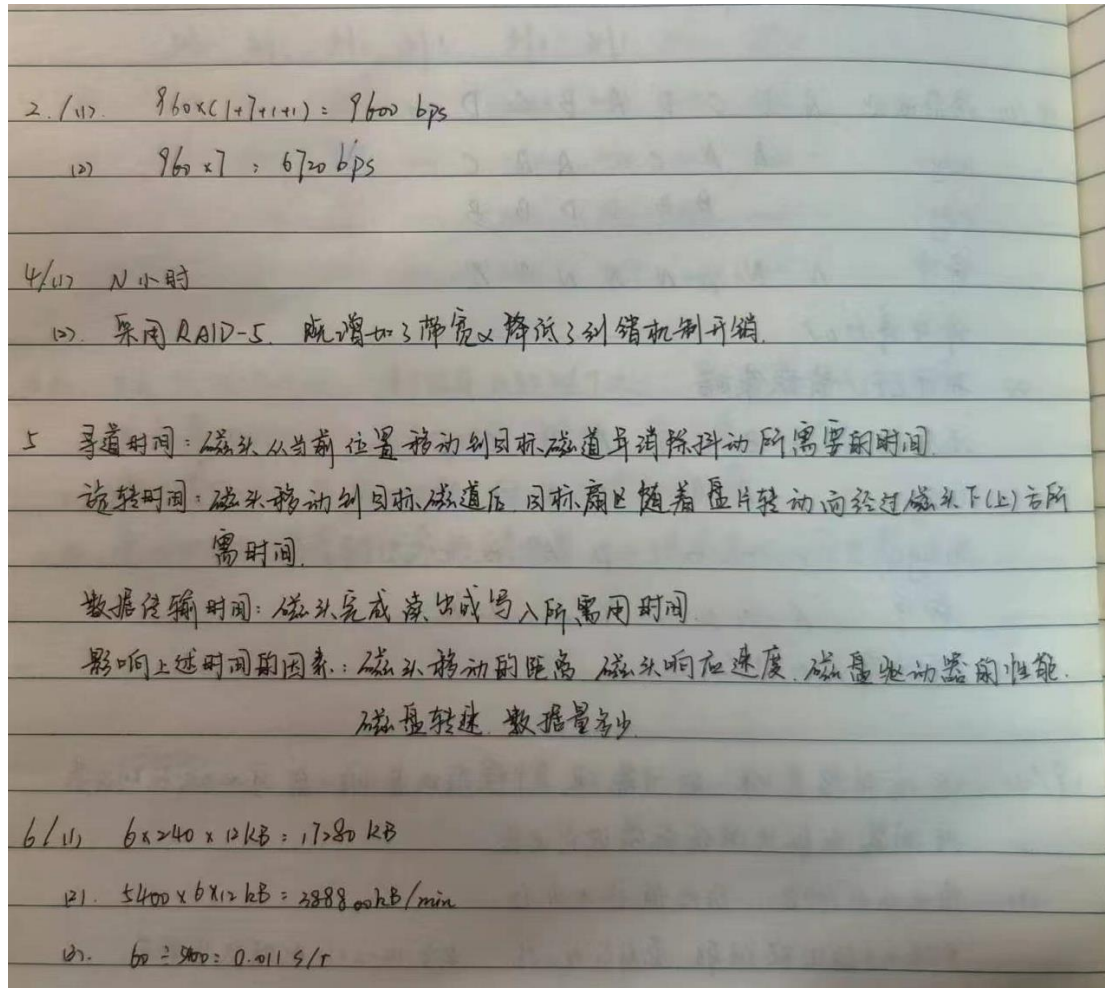
1. 串行总线的优点：较高的传输距离，简化线路布线，较高的带宽利用率

串行总线的缺点：较低的传输速率，传输延迟

并行总线的优点：高速数据传输，低传输延迟

并行总线的缺点：高复杂度和成本，时序问题

造成串行总线和并行总线接口速率不同的主要原因是：串行总线使用高速传输技术，数据位宽度不同，资源和成本考虑。



3. (1) I2C 的数据包构成如下：

起始条件：主设备发送一个起始位 (START) 来开始传输数据包。

地址字节：主设备发送一个 7 位或 10 位的地址字节，用于选择从设备。

数据字节：主设备或从设备在传输过程中通过时钟线 and 数据线进行数据交换，可以包含多个字节。

确认位：每个字节传输后，接收设备发送一个确认位 (ACK) 来确认接收到数据。

停止条件：主设备发送一个停止位 (STOP) 来结束数据传输。

(2) I2C 是半双工的，意味着数据的传输只能在一个方向上进行。这是因为 I2C 总线只有两根线，即时钟线 (SCL) 和数据线 (SDA)，它们被用于双向数据传输。在 I2C 中，总线的主设备负责发送数据，而从设备负责接收数据。在数据传输过程中，主设备控制总线的方向，可以切换为发送或接收模式，而从设备则始终处于接收模式。

(3) I2C 传输的起止条件如下：

起始条件 (START)：主设备发送一个低电平到高电平的跳变 (SCL 为高电平时，SDA 发生跳变)，表示数据传输的开始。

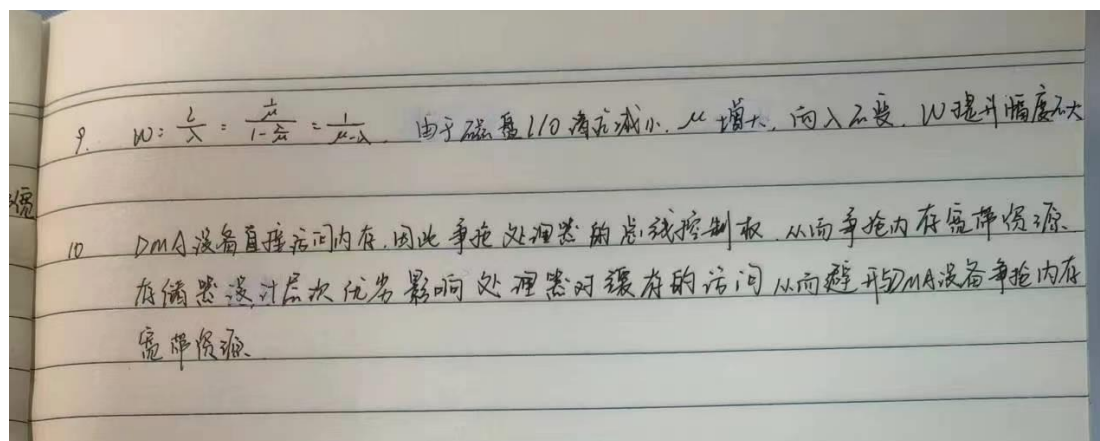
停止条件 (STOP): 主设备发送一个高电平到低电平的跳变 (SCL 为高电平时, SDA 发生跳变), 表示数据传输的结束。

起始条件和停止条件用于确定每个数据包的开始和结束, 主设备控制总线的状态变化, 从而实现数据的传输和通信。

7. 磁盘控制电路可以通过决定请求的最优执行次序来减少磁盘访问用时, 这种技术通常称为磁盘调度算法。磁盘调度算法的目标是通过优化磁头的移动路径, 减少寻道时间和旋转延迟, 从而提高磁盘的访问效率。以下是一些常见的磁盘调度算法: 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) 最短期望服务时间 (SST)

8. 在 RAID4 中, 写入优化可以对读取速度产生一定的影响, 主要体现在将写入操作转换为读取-修改-写入操作。当进行写入操作时, RAID4 需要读取原始数据块、校验块以及要写入的数据块, 然后对它们进行修改, 最后再写入到对应的磁盘上。这样的写入优化能够减少磁盘的读取次数, 从而提高写入的效率。

然而, 也会导致写入开销增加, 影响读取并发性写入操作的延迟增加, 当同时有大量的写入操作进行时, 读取操作可能需要等待写入操作完成后才能进行, 从而降低读取的响应速度。



1. 常见的总线仲裁机制包括集中式仲裁、分布式仲裁和基于总线协议的仲裁。它们各有不同的优缺点和适用场景:

1. 集中式仲裁:

- 优点: 实现简单, 易于控制和管理。
- 缺点: 可能引起总线瓶颈, 存在单点故障的风险。
- 适用场景: 适用于较小规模的系统, 例如个人电脑或嵌入式系统。

2. 分布式仲裁:

- 竞争性仲裁:
 - 优点: 可以提供较高的总线利用率。
 - 缺点: 可能引发冲突和延迟, 需要处理优先级和公平性问题。
 - 适用场景: 适用于大型系统, 例如服务器或多处理器系统。
- 非竞争性仲裁:

- 优点：避免了冲突和延迟。
- 缺点：可能浪费总线资源，无法充分利用带宽。
- 适用场景：适用于实时系统和小型系统，例如嵌入式系统或控制系统。

3. 基于总线协议的仲裁：

- 优点：具有较高的灵活性和可扩展性，适用于各种类型的系统。
- 缺点：需要实现和遵循相应的协议，增加系统复杂性。
- 适用场景：适用于具有特定总线协议的系统，例如 PCI、USB、Ethernet 等。

选择适当的总线仲裁机制应考虑以下因素：

- 系统规模：集中式仲裁适用于较小规模的系统，而分布式仲裁适用于大型系统。
- 实时性要求：基于令牌的仲裁和请求-应答仲裁提供可预测性和公平性，适用于对实时性有较高要求的系统。
- 设备关系：设备之间的优先级关系和协作方式影响仲裁机制选择，竞争性仲裁适用于没有明确优先级关系的设备。

总线仲裁机制的选择应综合考虑性能、实时性、可扩展性和系统复杂性等因素，以满足具体系统需求。

2. APB (Advanced Peripheral Bus) :

特点：APB 是 AMBA 中的低速外设总线，具有简单的数据传输机制和较低的带宽要求。它采用简化的控制信号和时序，适用于低速外设和低功耗的应用场景。

使用场景：APB 常用于连接较简单的外设，如 GPIO（通用输入/输出）、UART（通用异步收发器）等。

AHB (Advanced High-performance Bus) :

特点：AHB 是 AMBA 中的高性能总线，具有高带宽和低延迟的特点。它支持多主设备和多从设备的并行操作，具有灵活的仲裁机制和支持分级的存储器映射。AHB 支持高性能的数据传输和多周期的访问模式。

使用场景：AHB 常用于连接性能要求较高的外设和存储器，如 DMA 控制器、嵌入式处理器等。

AXI (Advanced eXtensible Interface) :

特点：AXI 是 AMBA 中的高性能、高带宽的总线协议。它采用了多级流水线、乱序传输和缓存一致性机制，支持并行操作和高效的数据交换。AXI 还具有扩展性和可配置性，可以满足不同应用场景的需求。

使用场景：AXI 常用于连接性能要求较高的外设和存储器，如图像处理器、高速 ADC/DAC、视频处理器等。

ACE (AXI Coherency Extension) :

特点：ACE 是在 AXI 基础上添加了一致性扩展的协议，用于处理多核处理器系统中的高速缓存一致性。它提供了高效的数据共享和一致性保证，支持多个处理器核心之间的数据共享和同步。

使用场景：ACE 常用于多核处理器系统，用于实现高速缓存一致性和多核间的数据共享。

CHI (Coherent Hub Interface) :

特点: CHI 是 AMBA 的最新一代总线协议, 提供了高性能和高度可扩展性的架构, 适用于大规模多核处理器系统。它支持高速缓存一致性、QoS (Quality of Service) 和功耗管理等高级特性。

使用场景: CHI 主要应用于大型多核处理器系统和高性能计算领域, 用于实现高性能的数据共享和处理。

3./(1)AXI 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) 向从设备 (如存储器) 发起读取数据的请求。
- 写通道 (Write Channel): 用于从主设备向从设备发起写入数据的请求。
- 写地址通道 (Write Address Channel): 用于从主设备向从设备发送写入地址的请求。

协议没有设置独立的读响应通道是因为在 AXI 协议中, 读取操作的响应通过写通道返回。读响应通过与读通道关联的写通道传输, 从而减少了总线上的通道数量, 简化了总线结构。

(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:

- 读通道的地址传输应在读通道的有效传输之前进行。
- 读通道的有效传输应在写通道的响应传输之前进行。

这样的约束是为了确保正确的数据传输和一致性。读通道的地址传输必须在读通道的有效传输之前完成, 以便从设备准备好数据并提供响应。而读通道的有效传输应在写通道的响应传输之前完成, 以确保数据和响应按正确的顺序传回主设备。

(3)AXI 的突发传输 (Burst Transfer) 是一种连续传输数据的方式, 用于提高总线传输效率。在突发传输中, 主设备通过单个地址传输指定起始地址, 并在后续连续地址上进行读取或写入操作, 而无需为每个地址单独发起传输。

AXI 总线支持以下突发传输类型:

- 固定突发 (Fixed Burst): 连续数据传输, 每个传输都在连续地址上进行。
- 增量突发 (Incremental Burst): 连续数据传输, 每个传输的地址根据一定的增量递增。
- 突发交错 (Wrap Burst): 连续数据传输, 每个传输的地址在一定范围内循环交错。

突发传输可以减少传输的开销, 并提高总线带宽利用率, 适用于需要连续读写大块数据的场景, 如存储器访问。