2. (1) 波特率 = 960 × (1+7+1+1) = 9600 Bd

(2) 有效数据传输速率 = 960 × 7 = 6720 bit/s

4. (1) ∵ RAID0 ∴ MTTF = Nh

(2) 使用 RAID-6 MTTF = 2Nh

5. 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。

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要的时间。

数据传输时间：磁头完成读出或写入所需时间

影响因素：磁盘请求的执行次序；读写数据的速率

6. (1) 总容量 = 6 × 240 × 12kB = 17280 kB

(2) 5400 r/min = 90 r/s

数据传输速率 = 90 × 12 kB/s = 1080 kB/s

(3) 平均旋转时间 = $\frac{1}{2} \times \frac{60s}{5400r}$ = 5.56 ms

9. 设 $\mu_1 > \mu_0$ 则 $w_0 - w_1 = \frac{1}{\mu_0 - \lambda} - \frac{1}{\mu_1 - \lambda} = \frac{\mu_1 - \mu_0}{(\mu_0 - \lambda)(\mu_1 - \lambda)}$

$$\therefore w_0 - w_1 = \frac{\mu_1 - \mu_0}{\mu_0\mu_1 + \lambda(\lambda - \mu_0 - \mu_1)}$$

$\because \lambda < \mu_0 < \mu_1 \quad \therefore 0 < \lambda < \frac{\mu_0 + \mu_1}{2}$

$\therefore \lambda \downarrow \quad [\lambda(\lambda - \mu_0 - \mu_1)] \uparrow \Rightarrow (w_0 - w_1) \downarrow$

∴ 性能提升幅度下降

10. 争抢内存带宽资源：优秀的存储器层次设计可以通过改进缓存、内存控制器和数据传输优化等方面减轻内存带宽的竞争，提高系统性能。

1.

串行总线和并行总线是两种不同的数据传输方式，它们各有优缺点。

串行总线的优点包括：

1. 线路复杂度低：串行总线只需要一对传输线（或单根线）来传输数据，相对较少的线路简化了物理连接的设计和布线。

2. 线长可扩展性好：串行总线的信号在单根线上传输，使得信号传输更加稳定，能够支持更长的线长，适合于大规模系统或远距离通信。

3. 成本较低：由于使用较少的线路和简化的接口电路，串行总线的成本相对较低。

并行总线的优点包括：

1. 传输速率高：并行总线在同一时刻可以传输多个数据位，因此在相同的频率下，传输速率更高，适用于需要高带宽的应用场景。

2. 数据传输稳定性好：并行总线将数据分为多个位同时传输，相对于串行总线，每个位的传输相对独立，传输稳定性较好。

3. 实时性较强：由于并行总线可以同时传输多个数据位，适合于实时性要求高的应用，如音视频传输。

造成串行总线和并行总线接口速率不同的原因有以下几点：

1. 信号干扰：并行总线中多个数据线在紧密排列下会导致相邻线之间的干扰，限制了并行总线的频率，而串行总线只有一对传输线，减少了信号干扰的可能性，可以实现更高的频率。

2. 线路长度匹配：在并行总线中，多个数据线的长度需要非常精确地匹配，以保持信号的同步性，而在串行总线中，只需要对一对传输线进行匹配，使得线路设计更加简单。

3. 接口复杂度：并行总线的接口电路相对复杂，需要更多的引脚和线路，增加了设计难度和成本。而串行总线的接口电路相对简单，降低了设计复杂度。

综上所述，串行总线和并行总线各有自己的优缺点，选择合适的总线取决于具体应用需求和系统设计的约束条件。

3.

1) I2C 的数据包由两部分组成：起始条件和数据传输。

- 起始条件（Start Condition）：在每个数据包的开始时，主设备发送一个低电平的起始条件信号（S）。起始条件告诉所有设备总线上将开始进行数据传输。

- 数据传输：在起始条件之后，主设备发送一个 7 位的设备地址（Address），用于选择要与之通信的从设备。之后是一个读/写位（R/W）来指示是读取数据还是写入数据。接下来是数据字节（Data Byte），可以是 8 位的数据位。数据字节的数量可以是 1 个或多个，具体取决于通信协议和数据传输需求。最后是一个确认位（Acknowledge）由接收方发送，用于确认数据是否成功接收。

2) I2C 是半双工的，意味着数据传输在同一条总线上只能在一个方向上进行。这是由于 I2C 总线的设计原理决定的。I2C 总线上的通信由主设备控制，并且主设备决定了数据传输的方向。在 I2C 总线上，主设备会发出读或写命令，从设备在接收到命令后进行相应的数据传输。因为只有一个总线线路用于数据传输，所以在同一时间内只能有一个方向的数据传输，所以 I2C 是半双工的。

3) I2C 传输的起止条件分别是起始条件（Start Condition）和停止条件（Stop Condition）。

- 起始条件（Start Condition）：起始条件是主设备发出一个低电平的信号（S）来表示将开始进行数据传输。这个信号告诉总线上的所有设备数据传输即将开始。

- 停止条件（Stop Condition）：停止条件是主设备发出一个高电平的信号（P）来表示数据传输结束。这个信号告诉总线上的所有设备数据传输已经完成。

起始条件和停止条件标志着每个数据包的开始和结束，用于同步和分隔不同的数据传输。在每个数据包之间，可以有多个数据字节的传输，但是起始条件和停止条件是必需的。

7

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，这个过程通常称为磁盘调度。

磁盘调度的目标是优化磁盘访问的顺序，以最大程度地减少寻道时间、旋转延迟和传输时间，从而提高磁盘的效率和响应速度。下面是一些常见的磁盘调度算法和它们的工作原理：

1. 先来先服务（FCFS）：按照请求的到达顺序进行处理。这种算法简单且公平，但可能会导致不连续的访问和长的平均寻道时间。

2. 最短寻道时间优先（SSTF）：选择距离当前磁道最近的请求进行处理。这样可以减少寻道时间，但可能会导致某些请求长时间等待。

3. 扫描算法（SCAN）：磁头按照一个方向移动，处理位于该方向上的所有请求，然后折返并处理另一侧的请求。这种算法可以减少长时间等待的情况，但可能导致一些请求在某些情况下等待较长时间。

4. 循环扫描算法（C-SCAN）：类似于 SCAN 算法，但在到达磁盘末端后，直接返回到磁盘起始端，形成一个循环。这样可以确保请求在等待时间上更加公平。

5. 最短服务时间优先（SRTF）：根据每个请求的执行时间进行排序，优先处理执行时间最短的请求。这种算法可用于 I/O 请求的优先级调度，以最大程度地减少整体的执行时间。

选择最适合的磁盘调度算法取决于特定的应用和工作负载。不同的算法在不同的情况下可能具有不同的效果。一些磁盘控制电路也可能结合多个调度算法，并根据实时的负载情况动态选择最佳的调度策略。通过合理的磁盘调度算法和策略，可以最大限度地减少磁盘访问的用时，提高系统的性能和响应速度。

8

在 RAID 4 中，写入优化对于读取速度的影响主要体现在以下几个方面：

1. 并行性提高：RAID 4 使用独立的奇偶校验盘（Parity Disk）来存储奇偶校验信息，而数据则分布在其他数据盘中。写入优化可以提高并行性，允许多个读取请求同时在不同的数据盘上进行操作。因此，读取速度可以得到提升，特别是在多个读取请求同时发生时。

2. 写入延迟：RAID 4 的写入操作需要计算奇偶校验信息，并将其写入奇偶校验盘。由于奇偶校验盘的特殊性，每个写入请求都需要涉及奇偶校验盘的读取和写入。因此，写入操作相对于单个数据盘的写入操作会更慢一些，因为需要额外的奇偶校验计算和盘片寻道。这种写入延迟可能会对读取速度产生一定的影响。

3. 磁盘吞吐量：RAID 4 中，写入操作需要写入数据盘和奇偶校验盘，而读取操作只需要读取数据盘。由于奇偶校验盘的写入操作可能会限制总体的磁盘吞吐量，因此写入优化可能会略微降低读取速度。

总的来说，RAID 4 的写入优化对于读取速度的影响取决于具体的工作负载和系统配置。在写入优化方面，RAID 4 在提高并行性方面有优势，可以提高多个读取请求的并发处理能力，从而提高读取速度。然而，写入操作需要额外的奇偶校验计算和盘片寻道，可能导致写入延迟，对读取速度产生一定影响。同时，奇偶校验盘的写入操作可能会限制总体的磁盘吞吐量，略微降低读取速度。因此，在设计 RAID 4 系统时需要综合考虑写入优化和读取速度之间的权衡。

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

1.

1. 集中式仲裁：

- 优点：集中式仲裁由一个中央仲裁器负责决定总线的使用权，可以确保公平性和按照优先级分配总线资源。

- 缺点：存在单点故障风险，如果中央仲裁器发生故障，整个总线系统将无法正常工作。

适用场景：适用于小规模系统，总线参与设备数量较少且预期的请求冲突较少的场景。

2. 分布式仲裁：

- 优点：分布式仲裁将仲裁功能分散到总线上的各个设备，每个设备根据一定的规则决定是否获取总线的使用权，可以避免单点故障风险。

- 缺点：可能出现冲突和竞争，需要额外的协议和算法来处理仲裁冲突。

适用场景：适用于大规模系统，需要更高的容错性和可扩展性的场景，例如多处理器系统。

3. 旁路仲裁：

- 优点：旁路仲裁使用专门的仲裁线来传递仲裁信息，不占用主总线带宽，避免了总线冲突。

- 缺点：需要额外的硬件和线路来实现旁路仲裁，增加了成本和复杂性。

适用场景：适用于对总线带宽要求较高的场景，例如高性能计算和多媒体处理系统。

选择适当的总线仲裁机制取决于系统的规模、容错需求、总线带宽要求以及对公平性和成本的考虑。在实际设计中，常常需要综合考虑各种因素，选择最适合特定应用场景的仲裁机制。

2.

AMBA 是一种由 ARM 公司定义的总线架构，包括多个总线协议，其中常见的包括 APB、AHB、ACE 和 CHI。

1. APB：

- 特点：APB 是 AMBA 总线中速度较慢、面向低带宽外设的总线协议。它采用简单的点对点连接和多主从结构，适用于低功耗、低速度和简单的外设连接。

- 使用场景：APB 常用于连接较简单的外设，如控制寄存器、配置寄存器等。

2. AHB：

- 特点：AHB 是 AMBA 总线中中等速度、高性能的总线协议。它支持多主多从结构和高带宽传输，具有较低的延迟和更强的总线控制能力。

- 使用场景：AHB 常用于连接性能要求较高的外设和存储器，如高速存储器、DMA 控制器等。

3. AXI：

- 特点：AXI 是 AMBA 总线中最广泛使用的高性能总线协议。它采用多层互联结构，支持高带宽、低延迟和高度可扩展的通信。

- 使用场景：AXI 适用于连接复杂的高性能外设和存储器，如高速 DSP、图形处理器、高带宽存储器等。

4. ACE：

- 特点：ACE 是在 AXI 基础上添加的一组协议扩展，用于处理多核系统中的缓存一致性问题。它提供了缓存一致性的支持，确保多个处理器核心之间的数据一致性。

- 使用场景：ACE 适用于多核处理器系统，用于解决缓存一致性和数据一致性的问题。

5. CHI：

- 特点：CHI 是 AMBA 总线中最新的一代协议，用于高性能和高度可扩展的系统互联。它提供了更高的带宽、更低的延迟和更强的一致性支持。

- 使用场景：CHI 适用于连接复杂的多处理器系统、高性能服务器和网络交换设备等。

这些 AMBA 总线协议根据不同的性能需求和连接外设的复杂度，提供了灵活的选择，以满足各种应用场景的要求。选择适当的总线协议取决于系统设计的性能要求、功耗限制和外设的复杂度等因素。

3.

1) AXI 总线包含以下独立的事务通道：

- 读通道：用于主设备向从设备发起读取数据的事务。
- 写通道：用于主设备向从设备发起写入数据的事务。
- 写应答通道：从设备向主设备发送写入应答的通道。

协议没有设置独立的读响应通道的主要原因是，读取数据的响应可以通过读通道进行传输。这样可以减少总线上的信号线数量和硬件复杂性。

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：

- 读通道的读取地址信号（ARADDR）和写通道的写入地址信号（AWADDR）应该在时钟上升沿之前稳定。
- 在读通道或写通道发起时钟上升沿之前，必须保证地址信号和其他相关信号的稳定。
- 读通道和写通道的响应信号（RRESP 和 BRESP）应该在时钟上升沿期间保持稳定。

这些时序依赖关系的设置是为了确保正确的事务顺序和数据的一致性。通过约束握手信号的时序，可以保证读/写传输事务的正确执行和数据的准确传输。

3) AXI 的突发传输是一种在连续地址范围内进行数据传输的机制，以提高总线的效率。突发传输允许主设备在一次地址事务中连续读取或写入多个数据。

AXI 总线定义了以下突发传输类型：

- 固定突发：在一次地址事务中传输连续的固定数量的数据。
- 增量突发：在一次地址事务中传输连续的递增地址的数据，每个数据的地址比前一个数据的地址增加一个固定的偏移量。
- 未对齐突发：在一次地址事务中传输连续的数据，但起始地址不对齐于数据宽度。

突发传输可以减少地址和控制信号的传输次数，提高总线的效率和带宽利用率。根据应用的需求和数据的连续性，选择合适的突发传输类型可以优化系统性能。