C5

1.

串行总线的优点：

简单易用：由于只需要一条信号线来传输数据，因此设计实现较为简单且易于控制。

高噪音容限：由于在传输过程中只有一条信号线，因此可以采用差分信号等方式提高信号传输质量。

更长距离的传输：串行总线可以通过差分信号等技术，使得信号传输的距离更远。

串行总线的缺点：

速度相对较慢：由于在同样的时间内只能传输一条数据，因此速度相对较慢。

可扩展性差：由于只有一条信号线，因此扩展性不如并行总线。

并行总线的优点：

传输速度快：由于同时传输多条数据，因此传输速度更快。

可扩展性强：由于有多条信号线，因此可扩展性更好。

并行总线的缺点：

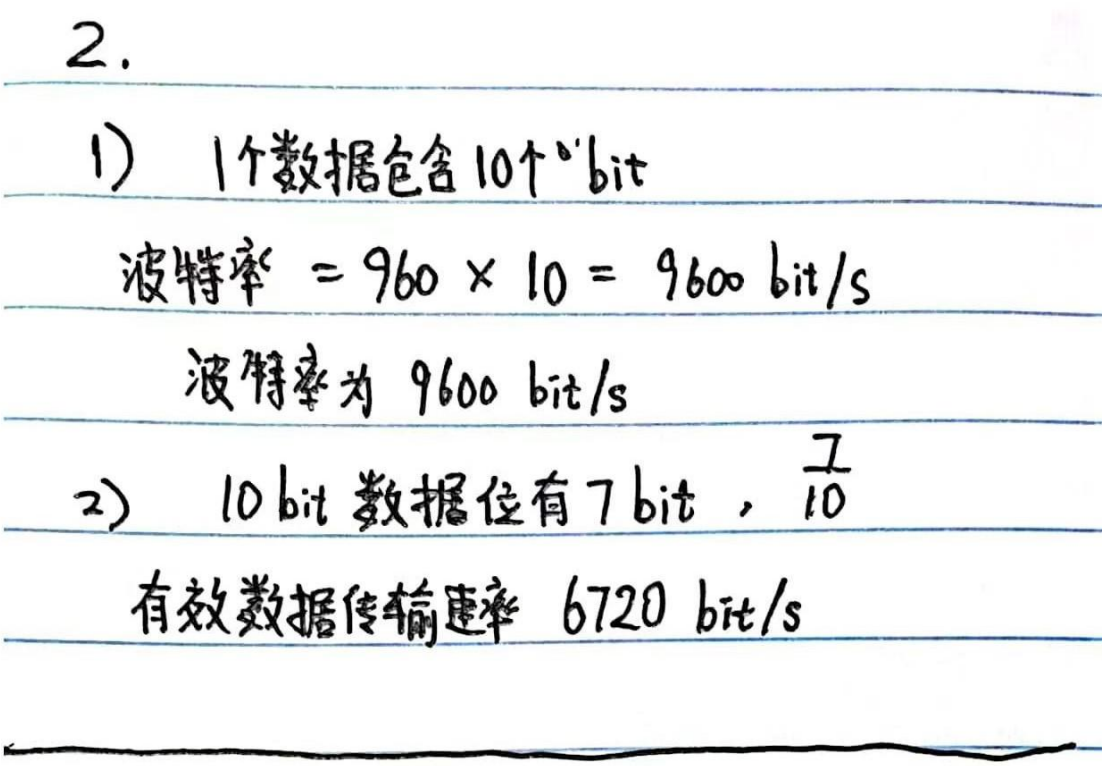
易产生干扰：由于存在多条信号线，因此很容易发生干扰，造成信号传输错误。

信号品质较差：由于存在多条信号线，容易出现互相干扰的现象，从而影响信号的传输质量。

相对短的传输距离：由于存在多条信号线，需要保证每一条线的长度要相同，因此相对较难实现更长距离的传输。

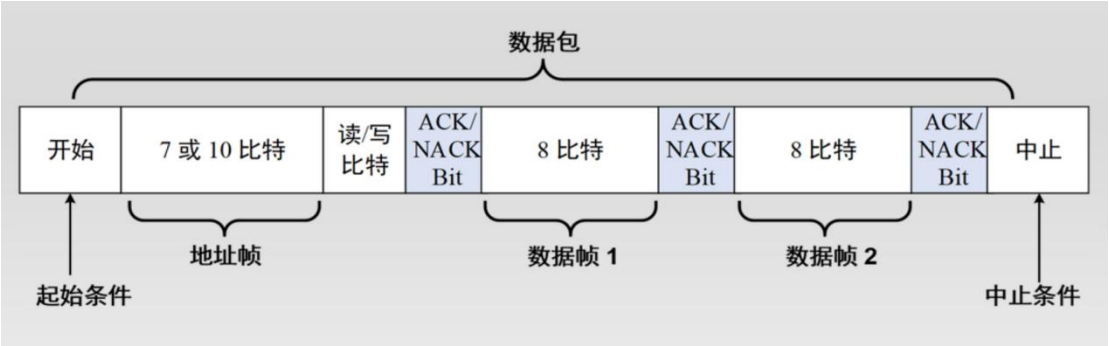
造成串行总线和并行总线接口速率不同的原因：

两者在传输时所采用的通信方式不同。串行总线仅使用一条信号线来进行数据传输，而并行总线则同时使用多条信号线来传输数据。由于并行总线能够同时传输多条数据，因此它的传输速度相对较快，而串行总线只能同时传输一条数据，因此速度相对较慢。



3.

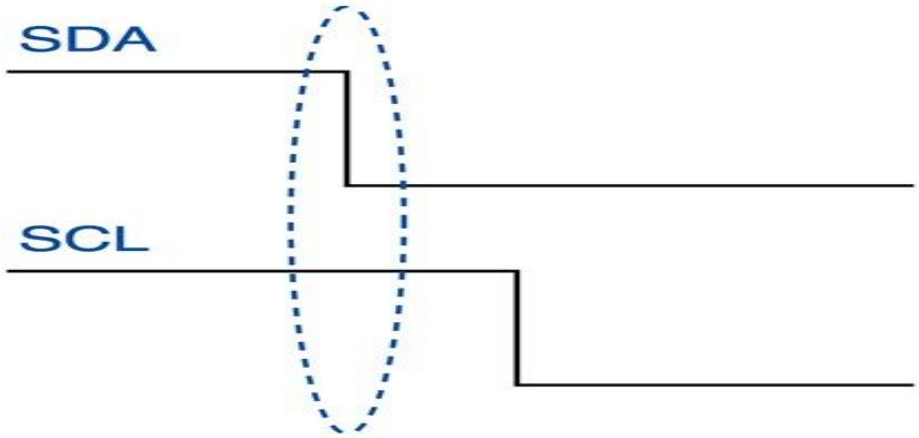
1）I2C 的数据包由一个开始位（开始条件）、七位设备地址、读写位、应答位、八位数据、应答位以及一个中止位（停止条件）组成。数据位的前后必须有应答位，同一个数据包可以包含多个数据。其中，起始位和停止位用于标识一次传输的起始和结束，从设备地址用于指定通信的目标从设备，读写位用于区分读写操作，数据位是要传输的数据内容，应答位是用于确认数据是否正确接收的信号。



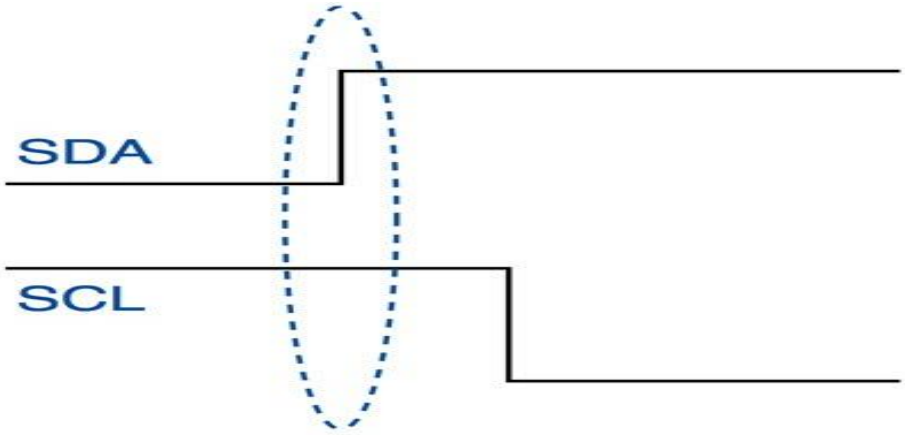
2）I2C 是半双工的通信方式，这意味着在一个时刻只有主设备或从设备能够发送数据或接收数据。这是因为 I2C 总线上只有两条线：串行数据线（SDA）和串行时钟线（SCL）。主设备通过控制时钟线来发送数据，从设备则通过控制数据线来回复数据。由于只有两条线，因此在同一时刻不能同时进行发送和接收，这就限制了 I2C 的通信方式只能是半双工的。

3）

开始条件：先将 SDA 线从高电平切换到低电平，然后将 SCL 线从高电平切换到低电平；



停止条件：先将 SDA 线从低电平切换到高电平，然后将 SCL 线从高电平切换到低电平；



C5

2.

1) 1个数据包含10个bit

波特率 $= 960 \times 10 = 9600$ bit/s

波特率为 9600 bit/s

2) 10 bit 数据位有 7 bit，$\frac{7}{10}$

有效数据传输速率 6720 bit/s

寻道时间由机械物理结构决定，包括磁头的移动时间，加(减)速度等参数；

旋转时间取决于磁盘转速和扇区所在的角位置，通常是固定值，与硬盘转速成反比；

数据传输时间由磁盘接口类型、控制器芯片、CPU性能、内存速度等因素共同影响，以上因素影响数据传输速度，数据密度(数据量)也影响传输时间。

4.

1) 采用RAID-0，用4块盘

系统的 MTTF 为 $\frac{N}{4}$ 小时

2) 若是RAID 1+0

50 GB < 80 GB < 50 GB×2

4块磁盘的RAID 10，数据被分成两份，每份数据被复制到两个磁盘中，每两块磁盘构成一个RAID1，将两个RAID1组合成RAID0

MTTF 为 $\frac{N^2}{2}$ 小时

6.

1) $6 \times 240 \times 12\text{KB} = 16.875\text{MB}$

总容量 16.875MB

2) 5400 r/min = 90 r/s

12KB × 90 = 1080 KB

数据传输速率 1.0546875 MB (1080 KB)/s

3) 5400 r/min = 90 r/s

$\frac{1}{2} \times \frac{1}{90} = \frac{1}{180} = 5.56 \times 10^{-3}$

平均旋转时间 5.56 ms ($\frac{50}{9}$ ms)

5.

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

旋转时间：磁头移动到目标磁道后，目标扇区随着扇片转动而经过磁头下(上)方所需的时间

数据传输时间：磁头完成读出或写入的时间

9.

提升率 $\frac{W_0}{W_1} - 1 = \frac{\mu_1 - \lambda}{\mu_0 - \lambda} - 1 = \frac{\mu_1 - \mu_0}{\mu_0 - \lambda}$

当 $\lambda \downarrow$ 时，$(\mu_0 - \lambda) \uparrow$，$\frac{\mu_1 - \mu_0}{\mu_0 - \lambda} \downarrow$

即 $\lambda \downarrow$，性能提升率 $\downarrow$

λ —— 单位时间I/O请求到达率

7.

磁盘控制电路的具体处理步骤如下：

1.读取所有挂起的磁盘访问请求，包括磁道号和读写指令等信息。

2.将读写请求按照某种优化策略进行排序，比如电梯算法、C-LOOK 算法等。

3.按照排序后的顺序依次处理每个请求，从而尽可能地减少寻道时间和旋转时间。

4.当一个请求已经被处理完成后，如果后续还有与之相邻的请求，就可以直接处理这些请求，从而进一步缩短访问时间。比如，在电梯算法中，当磁头移动到最边缘位置后，就可以立即反向移动，处理反方向的请求。

通过以上步骤，磁盘控制电路可以实现优化磁盘访问请求的执行顺序，从而减少磁盘访问用时，提高磁盘读写效率。需要注意的是，不同的优化算法适用于不同的场景，实际应用时需要根据具体情况选择合适的算法。

8.

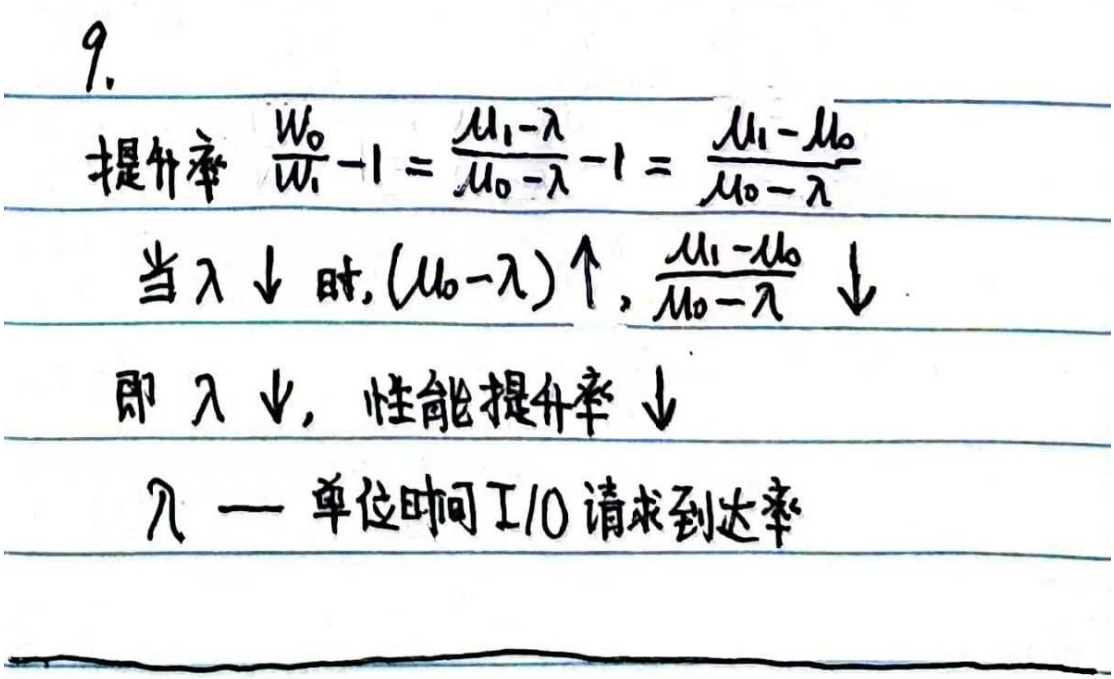
写入优化指的是将当前写入磁盘的数据与旧数据进行对比可以计算奇偶校验位的改变，避免读取所有磁盘来重新计算奇偶校验位。此技术对磁盘读取速度的影响相对较小，但是仍然存在一定的影响。具体来说，写入优化对 RAID4 的读取速度主要从以下两个方面产生影响：

1.奇偶校验计算的复杂度增加：RAID4 通过奇偶校验方式保证数据的容错性和冗余性，而在写入优化时，需要重新计算被修改的数据块和相关奇偶校验块的新值。这个过程可能会增加计算复杂度，导致读取速度变慢。

2.磁盘寻址时间增加：在写入优化时，系统需要读取被修改的数据块和相关的奇偶校验块，计算后再写入磁盘中。由于数据块可能分布在不同的磁盘上，执行读取和写入操作时需要进行多次磁盘扇区的读写操作，因此会增加磁盘寻址时间，进而降低了读取速度。

需要注意的是，影响的大小还受到访问模式的影响。对于顺序读取，影响相对较小，因为数据块分布在连续的位置，磁盘寻址时间差异不大。但是对于随机读取，写入优化可能需要多次读取和写入数据块，增加了磁盘寻址时间，导致读取速度更加明显地降低。

9.



10.

DMA与设备与处理器可以共享内存带宽，因此在高频率I/O操作时，可能会发生DMA设备与处理器争抢内存带宽资源的情况。

存储器层次设计可以通过缓存和预读技术等方式来优化内存带宽资源的利用效率。每个层次都有不同的访问速度、容量、成本以及其他特性。如果系统中的存储器层次设计得合理，DMA设备和处理器可以在不同的层次中找到自己需要的数据，减少相互干扰，提高内存带宽的利用率。但是，如果存储器层次设计得不合理则会导致内存带宽资源浪费，影响系统性能。

C6

1.

常见的总线仲裁机制如下：

1.集中式仲裁：由一个“仲裁器”集中控制总线的分配，可以保证可靠但效率低下，适用于设备数量较小且对实时性要求不高的场景。

2.分布式仲裁：每个设备都拥有“抢占权”，设备之间通过协议（如 CSMA/CD）自行争抢总线权限，效率高但存在可能的冲突和错误，适用于设备数量较大且对实时性要求较高的场景。

3.基于令牌传递的仲裁：将一个令牌在设备之间传递来获取总线访问权，有效地避免了冲突，并能够提供良好的实时性能，但是如果一个设备失效会导致整个系统瘫痪，适用于一些高实时性环境，如工业自动化与控制、武器指挥与控制等领域。

2.

APB：面向低带宽、低功耗外设的总线，具有简单、易用、省电等特点，适合于若干主从设备需要进行高速数据传输但总线中各个从设备带宽需求较小的场景。

AHB：面向中等带宽、中等复杂度的外设，具有更高的带宽和性能，还支持 split 和 burst 等特性，适用于多种场景，包括大型 MCU、ASIC 芯片等。

AXI：面向高带宽、多处理器、内存访问的总线，具有多级流水线、out-of-order 执行、性能优异和可扩展等特点，适合于高性能需求的 SOC 和 FPGA 等复杂应用场景。

ACE：基于 AXI，增加了缓存一致性支持，使得可以在并行性和一致性之间平衡，并且可以在 cache 内部实现可靠交互，适用于需要高并发和更高处理能力的神经网络芯片组架构等场景。

CHI：基于 ACE，增加了纤细粒度的 QoS 支持、分组虚拟化和高效缓存使能等特性，从而提供更好的多级调度性能和灵活性。适用于高性能服务器、数据中心等各种高要求场景。

3.

1）AXI 有五个独立的事务通道，分别是：

读地址（Read Address，简称 AR）

读数据（Read Data，简称 R）

写地址（Write Address，简称 AW）

写数据（Write Data，简称 W）

写响应（Write response，简称 B）

读响应信号是包含在读数据通道中，和写响应通道不同。读数据通道传送着从 slave 到 master 的读数据和读响应信息。读响应信息指明读事务的完成状态。

2）协议规定发送方的 VALID 不能依赖接收方的 READY；反过来，接收方的 READY 可以等待，也可以不等待发送方的 VALID。当发送方的数据准备好时，会发出和维持高电平的 VALID 信号，表示数据有效；当接收方准备好接受数据时，会发出 READY 信号。数据只有在 VALID 与 READY 信号同时有效时才开始传输。若这两个信号继续保持，发送方将会继续传输下一个数据，直到发送方撤销 VALID 信号或接收方撤销 VALID 信号时，传输终止

这样的约束是为了让发送方（Source）和接收方（Destination）都有能力控制传输

3）突发传输，就是在一次事务中，连续地传输多个地址相邻的数据。一次突发传输中可以包含一至多次数据（Transfer）。每个 transfer 因为使用一个周期，又被称为一拍数据（Beat）。每个数据可以是多个 Byte 构成。

突发传输有以下几种类型：

1.FIXED：所有数据都使用起始地址。该模式适合对某个固定地址进行多次数据更新，比如读写一个 fifo 时，读写地址就是固定的。

2.INCR：后续数据的地址在初始地址的基础上进行递增，递增幅度与传输宽度相同。适合对于 RAM 等通过地址映射（mapped memory）的存储介质进行读写操作。

3.WRAP：首先根据起始地址得到绕回边界地址与最高地址。当前地址小于最高地址时，WRAP 与 INCR 类型完全相同，地址递增。但到递增后的地址到达最高地址后，地址直接回到绕回边界地址，再进行递增，就这样循环往复。最高地址由绕回边界地址计算得到。这种传输类型适用于对缓存行（cache line）的操作。