1. 串行总线与并行总线各有什么优缺点？试简述造成它们接口速率不同的原因。

串行总线的优点包括:
1.较高的可靠性：串行传输只需使用一条线路传输数据，减少了连接线路的数量，降低了故障和干扰的可能性。
2.较低的成本：由于使用较少的线路，串行总线的硬件成本相对较低。
3.较长的传输距离：串行传输可以在较长的距离上进行，因为串行信号的衰减和失真相对较小。
串行总线的缺点包括:
1.串行总线不能同时传输多个数据位
2.不适用于短距离高带宽需求：如果需要在短距离上传输大量数据，串行总线的效率可能不如并行总线。
并行总线的优点包括:
1.并行传输可以同时传输多个数据位。
2.适用于短距离高带宽需求: 对于需要在短距离上传输大量数据的场景, 如内部计算机总线, 使用并行总线可以更高效地完成传输任务。
并行总线的缺点包括:
1.成本较高：并行传输需要更多的线路和接口，增加了硬件成本。
2.可靠性较低：由于并行传输使用多条线路传输数据，存在更多的故障和干扰可能性。
3.传输距离限制：由于并行传输存在时序和同步问题，传输距离通常受到限制。

造成接口速率不同的原因是:
造成串行总线和并行总线接口速率不同的主要原因是信号传输的方式和传输的带宽。
对于并行总线，多个数据位同时通过多条并行线路传输，因此在同一时钟周期内可以传输多个数据位，从而实现较高的传输速率。但容易收到干扰，导致数据错误和传输失败。

而对于串行总线，数据位依次通过单条线路传输，因此每个时钟周期只能传输一个数据位。尽管串行总线的传输速率较慢，但通过增加信号频率和使用更复杂的编码方式，可以在单位时间内传输更多的数据位，从而实现较高的传输速率。

此外，串行总线在实际应用中通常能够达到更高的传输速率，这是由于以下几个原因:
信号干扰：并行总线中的多条线路之间容易发生互相干扰的问题，特别是在较长的距离和高频率下更为明显。这可能导致数据错误和传输失败。相比之下，串行总线只需要一条线路传输数据，减少了干扰的可能性。
传输距离限制：并行总线的每条线路都需要保持相同的时序，对于长距离传输来说，这可能会面临更大的挑战。串行总线则可以通过适当的调整和设计，克服长距离传输的限制。
成本和复杂性：并行总线需要更多的线路和接口，这会增加硬件的成本和复杂性。而串行总线只需要较少的线路和接口，相对更简单和经济。

2. 假设某系统使用 UART 传输数据，每个数据包拥有 1 位起始位、7 位数据位、1 位校验位和 1 位停止位，系统每秒传输 960 个数据包。回答以下问题。
   1) 系统的波特率为多少？
   2) 系统的有效数据传输速率是多少？

(1) 波特率 = 10×960 = 9600 (位/秒)
(2) 有效数据传输速率 = 960×7 = 6720 (位/秒)

3.

阅读 I2C 相关标准，回答以下问题。
1) I2C 的数据包是如何构成的？
2) 为什么 I2C 是半双工的？
3) I2C 传输的起止条件是什么？

1）.I2C (Inter-Integrated Circuit) 的数据包由多个数据帧组成。每个数据帧包含一个起始位、8 个数据位、一个可选的应答位（ACK/NACK），以及一个停止位。数据帧按照从高位到低位的顺序传输。数据包通常由多个数据帧组成，其中每个数据帧代表一个字节的数据。在传输多字节数据时，通常使用一个或多个数据帧来传输数据。

2）.I2C 是半双工的通信协议，这意味着数据传输是在同一根线上进行的，但不能同时进行数据的发送和接收。
I2C 总线上有两根信号线：SDA（Serial Data Line）和 SCL（Serial Clock Line）。SDA 线用于数据的传输，而 SCL 线用于时钟同步。在 I2C 中，主设备控制总线的时钟，发送和接收数据的动作在时钟的控制下进行。由于 I2C 使用的是开漏（Open Drain）输出，即输出信号可以被外部设备拉低，但不能主动拉高，这样就实现了总线上的多主设备协作。在 I2C 的通信过程中，主设备会发送控制信号和数据，而从设备则负责接收数据并发送应答信号。因此，I2C 是半双工的，主设备和从设备在同一时间只能进行发送或接收操作，无法同时进行发送和接收。

3).I2C 通信中，起止条件用于指示传输的开始和结束。起止条件是通过在 SDA 线上的电平变化来表示的。
起始条件（Start Condition）在总线上的信号变化为：当 SCL 为高电平时，SDA 从高电平切换到低电平。
停止条件（Stop Condition）在总线上的信号变化为：当 SCL 为高电平时，SDA 从低电平切换到高电平。
起始条件和停止条件用于确定一个传输周期的开始和结束，它们的出现指示了总线上的通信动作。在起始条件出现后，总线上的从设备可以被选中，而在停止条件出现后，从设备将被释放。

4. 若某块磁盘的MTTF为N小时，回答以下问题。 MTTF为失效率λ的倒数

1) 计算由4块这种磁盘组成的RAID0的MTTF。

2) 若每一块磁盘的可用容量为50G，而系统只需要80G的存储空间，试设计一种方案，使得4块磁盘组成的RAID达到尽可能大的MTTF。

解：(1) RAID0把数据分散在多个物理磁盘上，一块失效后全失效，则失效率为 4λ

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

(2) 使用RAID 1方案

将4块磁盘分为两组，每组使用一块磁盘存储数据，另一块磁盘用于镜像数据存储，则能提供100G的存储空间，即使有一块磁盘故障，系统仍可恢复数据

每一组的失效率为 2λ，$MTTF = \frac{1}{\frac{1}{2\lambda}+\frac{1}{2\lambda}} = N$

5. 磁盘完成某个扇区上数据读写需要的时间可以概括为：T=寻道时间+旋转时间+数据传输时间。试说明上述各量的含义，并简要分析影响上述时间的因素。

解：1. 寻道时间：石磁头从当前位置移动到目标磁道并调整对齐所需要的时间。

影响寻道时间的影响包括磁头移动的距离、磁头的速度和磁盘的物理结构。

2. 旋转时间：磁头移动到目标磁道后，目标扇区随着盘片转动而经过磁头下让为所需的时间。

旋转时间受到磁盘的旋转速度影响。较高的旋转速度意味着数据可以更快地旋转到磁头下方，从而减少读写操作的等待时间。

3. 数据传输时间：磁头完成读出或写入所需用时间。

数据传输时间受到数据量的影响，较大的数据量需要更多的时间进行传输。此外，磁盘的传输速率也会影响数据传输时间

6. 若某块磁盘的转速为5400r/min，共有6个记录数据的盘面，每个盘面包含240个磁道，每个磁道的信息量为12KB。回答以下问题。

1) 该磁盘的总容量为多少？

2) 该磁盘的数据传输速率为多少？

3) 估算磁盘的平均旋转时间。

解 (1) 总容量 $6\times240\times12KB = 17280KB = 16.875MB$

(2) 数据传输速率为：$12KB \times 5400r/min = 64800\ KB/min$

(3) 由于转速为90r/s 则 $\frac{1}{90}s/r$

则平均旋转时间：$t = \frac{1}{180}s$

7. 简述磁盘控制电路如何通过决定请求的最优执行次序来减少磁盘访问用时？

磁盘控制电路通过确定请求的最优执行次序来减小磁盘访问用时，主要有以下几个方面的考虑：
请求调度算法：磁盘控制电路会使用请求调度算法来确定请求的最优执行次序。常见的调度算法包括先来先服务（FCFS）、最短寻道时间优先（SSTF）、电梯算法等。这些算法根据请求的位置和状态进行评估，选择最优的请求执行次序，以最大程度地减小磁盘访问用时。
寻道优化：磁盘控制电路会根据请求的磁道位置进行排序，将相邻磁道的请求进行组合，以减少磁头的寻道移动距离。通过优化磁头的移动路径，可以减小寻道时间，提高访问效率。
旋转延迟优化：磁盘控制电路会根据请求的扇区位置进行排序，尽量使请求的扇区在同一个磁道上连续排列，从而减少磁盘旋转的等待时间。通过优化数据的排列顺序，可以减小旋转延迟，提高访问速度。
缓存策略：磁盘控制电路通常会采用缓存来存储频繁访问的数据块，减少对磁盘的实际读写次数。通过合理的缓存策略，可以提高数据的命中率，减少对磁盘的访问时间。

8. 试分析 RAID4 中的写入优化对于读取速度的影响。

写入优化：将当前写入磁盘的数据与旧数据对比，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
原本写入数据时，因受限于校验硬盘，同一时间只能做一次，启动所有硬盘读取数据形成同一校验分段的所有数据分段，与要写入的数据做好校验计算再写入。
写入优化后，避免了读取所有磁盘，则同一时间能并行写入数据，加快了读取速度。

9. 根据 I/O 请求花费在队列系统中的平均响应时间公式：

$$W = \frac{L}{\lambda} = \frac{\frac{1}{\mu}}{1 - \frac{\lambda}{\mu}} = \frac{1}{\mu - \lambda}$$

分析随着磁盘 I/O 请求减少，磁盘队列系统的性能提升幅度下降的原因。

解：当磁盘 I/O 请求减小时，即入减小，
由于 $\mu > \lambda$，当入减时 $\mu - \lambda$ 增大，则平均响应时间变长
磁盘队列系统的性能提升幅度下降。

10. DMA 设备与处理器是否会争抢内存带宽资源？存储器层次设计的优劣对此问题有何影响？

解：DMA设备与处理器在访问内存时可以争抢内存带宽资源

在传统的计算机系统中，处理器和 DMA设备共享同一条内存总线。当它们访问内存时，它们会争夺内存带宽资源。

通过存储器层次设计，处理器可以更频繁地从高速缓存中获取数据，而不必每次都去访问主内存。DMA设备也可以利用这种层次结构，将数据首先传到高速缓存中，然后再由处理器访问，从而减少对主内存的竞争。

1. 简述常见的总线仲裁机制，它们各有什么优缺点、分别适用于什么场景？

在计算机系统中，总线仲裁机制用于解决多个设备之间共享同一总线资源时的冲突问题。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。

1.集中式仲裁：集中式仲裁机制使用一个仲裁器或仲裁控制器来管理总线的访问。当多个设备请求访问总线时，仲裁器根据预先定义的优先级进行仲裁，决定哪个设备获得总线的控制权。这种机制具有简单、易于实现和低成本的优点。然而，它存在单点故障的风险，如果仲裁器出现故障，整个总线系统将无法正常工作。此外，集中式仲裁机制对于大规模系统可能面临性能瓶颈。集中式仲裁适用于小规模系统，要求简单、低成本且没有严格的性能要求的情况。

2.分布式仲裁：分布式仲裁机制将仲裁的责任分散到各个设备上，每个设备都具有一定的仲裁逻辑来决定自己是否能够访问总线。这种机制避免了单点故障的风险，并且可以实现更好的并行性和系统扩展性。然而，分布式仲裁机制的实现较为复杂，需要设备之间进行通信和协调，增加了系统的开销和复杂性。分布式仲裁适用于大规模系统，需要具备高并行性、扩展性和容错性的场景。

3.链式仲裁：链式仲裁机制将设备按照一定的顺序连接成链，每个设备都有权利在特定时刻请求总线的使用。当一个设备完成对总线的访问后，将仲裁权传递给下一个设备。链式仲裁机制具有简单、可靠和低成本的优点，适用于设备数量较少且固定的系统。然而，链式仲裁机制对于较大规模和动态变化的系统来说可能不够灵活和高效。链式仲裁适用于设备数量较少且固定的系统，对于实时性要求不高的场景，如嵌入式系统或小型局域网。

2. 简要描述 AMBA 总线中 APB、AHB、AXI、ACE 及 CHI 等总线协议的特点和使用场景。

1.APB（Advanced Peripheral Bus）：APB 是 AMBA 总线协议家族中的低速外设总线，适用于低带宽、低功耗和相对简单的外设。APB 通常用于连接低速外设，如 GPIO（通用输入输出）、定时器、UART（通用异步收发器）等。它具有简单的设计和低功耗特性，适合于嵌入式系统中对性能要求不高的外设。

2.AHB（Advanced High-performance Bus）：AHB 是 AMBA 总线协议家族中的中等带宽、中等性能的系统总线。AHB 具有高性能、高带宽和灵活性的特点，可连接多个高性能外设和处理器核。AHB 总线适用于中等复杂度的系统，如微控制器、数字信号处理器（DSP）等。它支持多主设备、多从设备和流水线传输，具有良好的时序控制和优秀的数据传输效率。

3.AXI（Advanced eXtensible Interface）：AXI 是 AMBA 总线协议家族中的高性能、高带宽的系统总线，用于连接复杂的处理器、高速存储器和外设。AXI 总线具有高度可扩展性和灵活性，支持多主设备、多从设备、乱序传输和突发传输等特性。AXI 总线适用于要求高性能和高并发性的系统，如高性能嵌入式处理器、数字信号处理器（DSP）、图像处理器等。

4.ACE（AXI Coherency Extension）：ACE 是基于 AXI 总线协议扩展的一种协议，用于处理多核处理器之间的一致性和高性能通信。ACE 支持高性能的缓存一致性和数据一致性，能够有效地处理多个处理器核之间的数据一致性问题。ACE 适用于多核处理器系统和具有共享缓存的系统，提供高性能的数据共享和一致性管理。

5.CHI（Coherent Hub Interface）：CHI 是 ARM 最新推出的高性能片上总线协议，专为大规模多核处理器系统设计。CHI 提供了高度的可扩展性、高带宽和高一致性，支持 CHI（Coherent Hub Interface）是 ARM 最新推出的高性能片上总线协议，专为大规模多核处理器系统设计。CHI 在 AXI 和 ACE 协议的基础上进行了扩展，提供了更高的性能、带宽和一致性。
CHI 的特点包括：
（1）高性能和高带宽：CHI 支持更高的频率和更大的数据宽度，提供了更高的总线带宽和数据传输效率。它能够满足多核处理器系统中对高性能和高并发性的要求。
（2）高度可扩展性：CHI 采用了分层结构，支持多级拓扑结构和多个集线器（hubs）之间的连接。这种设计使得系统能够灵活扩展和添加更多的处理器核，同时保持高性能和一致性。
（3）高一致性和缓存一致性：CHI 提供了强大的缓存一致性机制，确保多核处理器系统中的数据一致性。它支持高效的缓存一致性协议和高级的一致性管理功能，提供了更可靠的数据共享和协同处理能力。
CHI 适用于需要大规模多核处理器系统、高性能计算和高并发数据处理的场景。它能够满足对高带宽、高性能和高度一致性的要求，适用于服务器、数据中心、高性能计算领域等对计算能力和数据处理能力有严格要求的应用。

3. 调研 AXI 总线标准并回答以下问题。
   1) AXI 总线包含哪些独立的事务通道？为什么协议不设置独立的读响应通道？
   2) 简要描述在读/写传输事务中，通道的握手信号时序需要满足怎样的依赖关系？为什么要设置这样的约束？
   3) 什么是 AXI 的突发传输？有哪些突发传输类型？

（1）AXI（Advanced eXtensible Interface）总线包含以下独立的事务通道：
读通道（Read Channel）：用于从主设备（如处理器或 DMA 控制器）向从设备（如存储器或外设）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。
写响应通道（Write Response Channel）：用于从从设备向主设备发送写操作完成的响应。
AXI 总线协议没有单独设置读响应通道的原因主要是出于设计的简化和效率的考虑。
为什么协议不设置独立的读响应通道?
1.AXI 协议通过在读通道中包含读数据和读响应信息，将读取操作的数据和响应信息一起传输。这样可以减少总线上的信号线数量，简化硬件设计。
2.由于读取操作涉及数据的传输，读取数据的时延可能会比写入操作更长。因此，通过将读数据和读响应信息捆绑在一起传输，可以减少由于数据延迟导致的不必要的等待时间。
3.通过将读数据和读响应信息绑定在一起，还可以提高总线的利用率和吞吐量，因为在读取数据过程中，读取响应信息可以立即返回，而不需要单独的响应通道。

（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：
1.在读传输中：
读通道的地址信号（ARADDR）和读通道的传输开始信号（ARVALID）必须同时有效。
读通道的传输结束信号（ARREADY）必须在读通道的传输开始信号（ARVALID）有效之后的某个时间点开始有效。
写响应通道的传输结束信号（RVALID）必须在读通道的传输开始信号（ARVALID）有效之后的某个时间点开始有效。
写响应通道的传输结束信号（RVALID）必须在读通道的传输结束信号（ARREADY）有效之后的某个时间点开始无效。
2.在写传输中：
写通道的地址信号（AWADDR）和写通道的传输开始信号（AWVALID）必须同时有效。
写通道的传输结束信号（AWREADY）必须在写通道的传输开始信号（AWVALID）有效之后的某个时间点开始有效。
写响应通道的传输结束信号（BVALID）必须在写通道的传输开始信号（AWVALID）有效之后的某个时间点开始有效。
写响应通道的传输结束信号（BVALID）必须在写通道的传输结束信号（AWREADY）有效之后的某个时间点开始无效。

这些时序依赖关系的设置是为了保证数据的正确传输和一致性。具体原因如下：
1.读传输时序依赖关系保证了读取操作的顺序性和一致性。读通道的地址信号和传输开始信号同时有效，确保了读取操作的目标地址已被确定。传输结束信号的依赖关系则保证了读取操作的数据和响应信息一起传输，并且在读取完成后响应被送回。

2.写传输时序依赖关系保证了写入操作的顺序性和一致性。写通道的地址信号和传输开始信号同时有效，确保了写入操作的目标地址已被确定。传输结束信号的依赖关系则保证了写入操作的响应信息能够及时返回，以指示写入操作的完成状态。
通过设置这样的约束，可以确保读/写传输事务的正确执行和数据的一致性，避免数据冲突、错误的读取或写入操作以及系统的死锁等问题。这样可以保证总线协议的正确性和可靠性，提高系统的性能和稳定性。

（3）在 AXI（Advanced eXtensible Interface）总线协议中，突发传输（Burst Transfer）是指在单个事务中连续传输多个数据项或字节的一种方式。它可以有效地利用总线带宽，提高数据传输的效率。
AXI 总线协议定义了几种突发传输类型，包括：
1.固定突发传输（Fixed Burst）：在固定突发传输中，连续的数据项或字节在事务中以固定的长度进行传输。每个突发传输事务的长度是固定的，并且在事务开始时确定。
2.递增突发传输（Incrementing Burst）：在递增突发传输中，连续的数据项或字节在事务中以递增的地址进行传输。每个数据项的地址都是前一个数据项地址加上一个增量值，可以是字节地址、半字地址或字地址。
3.轮询突发传输（Wrap Burst）：在轮询突发传输中，连续的数据项或字节在事务中以循环的方式进行传输。当传输到最高有效地址后，下一个地址会回到起始地址。这种传输类型可用于循环缓冲区等应用场景。
4.延迟突发传输（Wait Burst）：在延迟突发传输中，连续的数据项或字节在事务中以间隔的方式进行传输。每个数据项之间可以插入一个或多个等待周期，用于提供额外的处理时间或数据处理逻辑。
这些突发传输类型可以根据应用的需求进行选择，以实现最佳的数据传输效率和性能。根据读取或写入操作的特点、数据的排列方式以及外设或存储器的要求，选择适当的突发传输类型可以提高数据传输的效率，并降低总线带宽的占用。