

第五章

1.

串行总线的优点有：

传输速率高，可以方便地实现长距离传输。由于数据是按位传输的，每一位之间只需要很短的时间间隔，因此可以利用高速时钟进行快速传输。

传输质量高，可以减少传输时的误码率。由于在传输过程中，不同数据位之间的干扰会相互抵消，从而提高传输质量。

硬件成本低，只需要一根或少数几根线路就可以传输数据，因此实现比较简单。

串行总线的缺点有：

传输距离受限，因为在长距离传输时，由于传输线的阻抗不均匀性、信噪比等因素的影响，会导致传输信号失真，从而影响数据传输质量。

传输开销大，因为数据是一个一个地传输的，每一个数据位都需要传输时钟、同步信号等控制信息，从而增加了传输开销。

并行总线的优点有：

传输速度快，可以同时传输多个位，因此适合于短距离高速传输。由于数据被划分成几个部分，每个部分都可以利用一条传输线进行传输，并且不同部分之间没有相互干扰，因此可以大大提高传输速度。

传输效率高，可以支持同步数据传输，在传输的过程中，不需要使用特殊的同步控制信号，从而可以减少控制开销。

并行总线的缺点有：

传输距离受限，由于在长距离传输时，信号传输线的长度和阻抗不均匀性等问题，可能会导致传输信号失真，降低数据传输质量。

硬件成本高，需要多条线路进行传输，因此设计和实现成本较高。同时还需要消耗更多的电力。

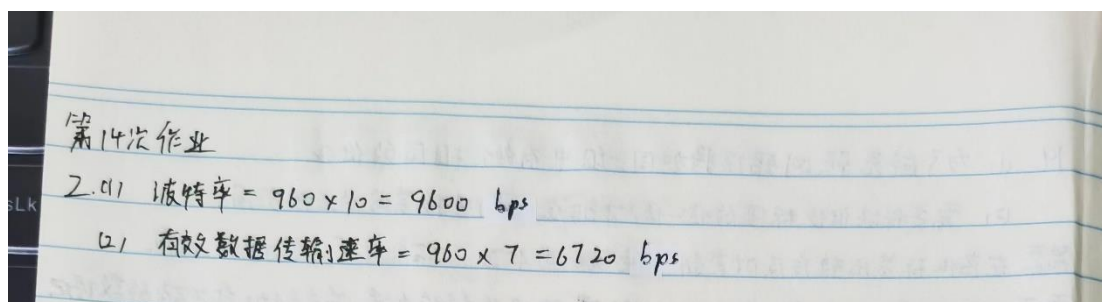
造成串行总线和并行总线接口速率不同的原因主要有以下几点：

数据位数不同。串行总线每次只能发送一个数据位，而并行总线每次可以发送多个数据位。因此，在相同的时钟频率下，并行总线的接口速率要高于串行总线。

时钟频率不同。串行总线由于只需要一根或少数几根线路进行数据传输，因此可以使用更高的时钟频率来提高接口速率。而并行总线由于需要多条线路进行数据传输，因此受到信号干扰、阻抗匹配等因素的限制，不能使用过高的时钟频率。

控制开销不同。串行总线由于每个数据位都需要时钟信号和同步信号等控制信息来保证数据正确性和同步性，因此会产生较大的控制开销。而并行总线由于只需要在帧的开始和结束部分添加控制信息来保证数据正确性和同步性，因此会减少控制开销。

2.



3.

(1) I2C 的数据包由以下几个部分构成：

起始条件：当总线空闲时，控制器将数据线 (SDA) 从高电平拉低到低电平，而时钟线 (SCL) 保持高电平，表示数据传输的开始。

设备地址：控制器发送一个 7 位或 10 位的设备地址，用来指定要通信的目标器件。

数据方向位：控制器发送一个数据方向位 (R/W)，用来表示是要从目标器件读数据 (R/W=1)，还是要向目标器件写数据 (R/W=0)。

应答位：目标器件在收到设备地址和数据方向位后，如果识别出自己是通信对象，就在下一个时钟周期内将数据线拉低，表示应答 (ACK)。

数据字节：控制器或目标器件根据数据方向位，发送或接收一个 8 位的数据字节。每发送或接收一个字节后，都要等待对方的应答位。

停止条件：当数据传输结束时，控制器将数据线从低电平拉高到高电平，而时钟线保持高电平，表示数据传输的结束。

(2) I2C 是半双工的，因为它只有一条数据线 (SDA)，而且在同一时刻只能有一个器件在发送或接收数据。如果有多个器件同时驱动数据线，就会造成总线冲突。因此，I2C 需要通过时钟同步和总线仲裁来协调不同器件之间的通信顺序和优先级。

(3) I2C 传输的起止条件如下：

起始条件：当总线空闲时，控制器将数据线 (SDA) 从高电平拉低到低电平，而时钟线 (SCL) 保持高电平，表示数据传输的开始。

停止条件：当数据传输结束时，控制器将数据线从低电平拉高到高电平，而时钟线保持高电平，表示数据传输的结束。

4.

4. (1) $MTTF = \frac{N}{4}$

(2) 使用 RAID 10，它是 RAID 1 和 RAID 0 的组合，将两个 RAID 1 组合成一个 RAID 0。

50G	50G
50G	50G

共可使用 100G 的空间 (> 80G)

并且可以容忍任何一个磁盘或者任何一个 RAID 1 的故障，则

$$MTTF_{RAID10} = \frac{MTTF_{disk}}{2} \times \frac{2}{1 - \frac{MTTR_{disk}}{MTTF_{disk}}} = \frac{N}{2} \times \frac{2}{1 - \frac{MTTR_{disk}}{N}}$$

$MTTR_{disk}$ 是单个磁盘的平均修复时间，设为 10 小时

$$\text{则 } MTTF_{RAID10} = \frac{N^2}{N-20}$$

5.

5. 寻道时间: 读写磁头移动到要求的磁道位置所需要的时间。

旋转时间: 磁头到达所要求的磁道位置后, 等待所要求的扇面旋转到磁头下的平均时间。

数据传输时间: 从磁盘读出或向磁盘写入数据所经历的时间。

影响上述时间的因素主要有:

系统架构和总线设计, 不同的系统架构会影响DMA设备和处理器之间访问内存的冲突和仲裁。

程序局限性, 是指程序执行过程中对存储器访问呈现出一定程度上集中性。

磁盘调度算法, 磁盘交替编号。

6.

$$6. (1) \text{总容量} = 6 \times 240 \times 12 \times 12 \text{KB} = 20736 \text{KB}$$

$$(2) \text{传输速率} = 12 \times 12 \text{KB} \times 5400 / 60 = 1296 \text{KB/s}$$

$$(3) \text{平均旋转时间} = (60 / 5400) / 2 = 0.00556 \text{s}$$

7.

磁盘控制电路是负责控制磁盘驱动器的读写操作的硬件部件, 它可以通过采用不同的磁盘调度算法来决定请求的最优执行次序, 从而减少磁盘访问用时。常见的磁盘调度算法有以下几种:

先来先服务算法 (FCFS): 按照请求到达的先后顺序进行服务, 不做优化。这种算法简单易实现, 但是效率较低, 可能导致磁头移动距离过长。

最短寻道时间优先算法 (SSTF): 每次选择距离当前磁头位置最近的请求进行服务, 使得每次的寻道时间最短。这种算法可以减少寻道时间, 提高系统效率, 但是可能导致某些请求被无限推迟 (饥饿现象)。

电梯扫描算法 (SCAN): 磁头按照一个方向进行扫描, 依次服务沿途的请求, 当到达边界或者没有请求时, 改变方向继续扫描。这种算法消除了 SSTF 算法的不公平性, 但是仍然有利于中间位置的请求。

8.

RAID4 是一种使用奇偶校验码的磁盘阵列技术, 它将数据分成条块存储在多个磁盘上, 而将校验码存储在一个专用的磁盘上。RAID4 的写入优化是指在写入数据时, 只更新变化的数据块和校验码, 而不是整个条块和校验码, 从而减少了写入的时间和开销。

RAID4 的写入优化对于读取速度的影响有以下几点:

优点: 写入优化可以提高磁盘阵列的整体性能, 因为写入操作占用的时间和资源更少, 从而为读取操作留出更多的空间。写入优化也可以减少磁盘的磨损, 延长磁盘的寿命, 降低故障的风险。

缺点：写入优化可能会增加读取操作的复杂度，因为读取时需要判断数据块和校验码是否完整，如果不完整，则需要进行恢复或重建。写入优化也可能导致数据不一致的问题，因为在写入过程中，如果发生断电或系统崩溃等异常情况，可能会造成部分数据块和校验码没有被更新，从而影响数据的完整性和可靠性。

9.

9. 从公式可以看出，当队列的利用率接近1时，平均响应时间也会减少，当I/O请求越多时，磁盘队列系统的性能会越差。反之当I/O请求减少时，平均响应时间也会减少，但是这个减少幅度并不是线性的，而是随着利用率的降低而逐渐变小。即当I/O请求已经很少时，再减少一些请求对于平均响应时间的影响就不大了。

原因是排队论中的数学规律所决定的，

10.

10. 取决于具体的系统架构和总线设计！

DMA设备和处理器使用不同的总线或总线端口访问内存，就不会发生冲突，并行地访问内存，此时存储器层次设计优劣对此问题没有影响。

如果使用同一个总线或总线端口访问内存，就会发生冲突，需要通过某种仲裁机制来分配内存带宽资源。此时若存储器层次设计能够充分利用程序的局部性，减少对低层次存储器的访问频率和数量，那么就可以降低DMA设备和处理器之间的竞争程度，提高系统性能。

第六章

1.

集中仲裁：将所有的总线请求集中到一个中央控制器，由它按照一定的算法决定哪个设备优先获得总线控制权。集中仲裁又可以分为链式查询、计数器定时查询和独立请求三种方。

优点：优先级固定，结构简单，扩充容易。

缺点：对硬件故障敏感，优先级不能改变，控制线数量多，控制逻辑复杂。

适用场景：适用于总线设备数量较少，通信速度不高，优先级要求不严格的情况。

分布仲裁：不需要中央控制器，每个设备都有自己的仲裁号和仲裁器。当它们有总线请求时，就会把它们各自的仲裁号发送到共享的仲裁总线上，通过比较仲裁号的优先级来决定哪个设备获胜。

优点：响应速度快，对优先次序的控制灵活，对电路故障不敏感。

缺点：需要额外的仲裁总线，仲裁号占用带宽，可能出现死锁现象。

适用场景：适用于总线设备数量较多，通信速度较高，优先级要求较高的情况。

2.

AMBA 包含了多种不同的总线标准，分别适用于不同的性能和复杂度的场景。以下是一些主要的总线协议及其特点和使用场景：

APB (Advanced Peripheral Bus) 是一种简单的低功耗和低带宽的总线，通常用于连接低速的外设，如 UART、I2C、SPI 等。APB 不具备流水线功能，只支持单一传输和非连续突发传输。

AHB (Advanced High-performance Bus) 是一种高性能的总线，通常用于连接 CPU 和高速设备，如内存、DMA、加速器等。AHB 支持流水线传输，可以实现无等待、突发和分裂传输。AHB 还支持多主机和多从机的互联架构，通过仲裁器和解码器来实现总线控制和地址分配。

AXI (Advanced eXtensible Interface) 是一种更高性能和更灵活的总线，通常用于连接高带宽和低延迟的设备，如多核处理器、FPGA、DSP 等。AXI 支持多层互联结构，可以实现并行传输、突发传输、乱序传输和保序传输。AXI 还支持 QoS、保护和错误检测等功能。

ACE (AXI Coherency Extensions) 是一种在 AXI 基础上增加了缓存一致性功能的总线，通常用于连接多核处理器和共享存储器等。ACE 支持缓存维护操作、缓存共享操作、缓存独占操作和缓存清除操作等。ACE 还支持动态 QoS、虚拟化和错误检测等功能。

CHI (Coherent Hub Interface) 是一种在 ACE 基础上进一步优化了性能和扩展性的总线，通常用于连接大规模的多核处理器系统。CHI 支持分层互联结构，可以实现高效的缓存一致性协议、数据压缩、数据重排和数据重试等。CHI 还支持低功耗模式、DFX 方法、QoS 机制和 Hazard 竞争处理等功能。

3.

(1) AXI 总线包含以下五个独立的事务通道：

读地址通道：用于传输读操作的地址和控制信息。

读数据通道：用于传输从设备到主机的读数据和读响应信息。

写地址通道：用于传输写操作的地址和控制信息。

写数据通道：用于传输从主机到设备的写数据。

写响应通道：用于传输从设备到主机的写响应信息。

为什么协议不设置独立的读响应通道？一种可能的解释是，读响应信息与读数据信息具有相同的数据流向和使用频率，因此将它们放在同一个通道中可以节省资源和带宽。而写响应信息与写数据信息具有不同的数据流向和使用频率，因此将它们分开为两个通道可以提高效率和灵活性。

(2) 在读 / 写传输事务中，每个通道都有一个 VALID 信号和一个 READY 信号，用于实现握手机制。VALID 信号表示地址、数据或控制信息有效，READY 信号表示目标处于可以接收信息的状态。只有当 VALID 和 READY 同时为高时，握手才成功，信息才被传输。

通道的握手信号时序需要满足以下依赖关系：

对于读地址通道和写地址通道，VALID 和 READY 信号没有指定的有效顺序，即它们可以同时变高，也可以先后变高。

对于读数据通道，RVALID 信号需要在 ARVALID 和 ARREADY 均有效后才能变高，即设备需要在接收到有效的读地址后才能发送有效的读数据。RVALID 和 RREADY 信号没有指定的有效顺序。

对于写数据通道，WVALID 信号需要在 AWVALID 和 AWREADY 均有效后才能变高，即主机需要在发送有效的写地址后才能发送有效的写数据。WVALID 和 WREADY 信号没有指定的

有效顺序。

对于写响应通道，BVALID 信号需要在 WVALID、WLAST 和 WREADY 均有效后才能变高，即设备需要在接收到最后一个写数据后才能发送有效的写响应。BVALID 和 BREADY 信号没有指定的有效顺序。

设置这样的约束是为了保证传输事务的正确性和一致性，避免出现无效或错误的信息传输。

(3) AXI 的突发传输是一种一次性传输多个数据项的方式，它只需要提供首地址和突发长度等控制信息，而不需要为每个数据项提供单独的地址。突发传输可以提高总线利用率和传输效率。

AXI 有以下三种突发传输类型：

固定突发：每个数据项都使用相同的地址，适用于多个设备共享一个数据端口的情况。

递增突发：每个数据项的地址按照指定的步长递增，适用于连续的内存访问的情况。

包装突发：每个数据项的地址按照指定的步长递增，但当达到指定的边界时，会回到起始地址并继续递增，适用于缓存行或矩阵块的访问的情况。