

1.

长距离传输

串行总线：优点：效率高，数据相互不干扰，传输质量高，使用数据线少，成本低

缺点：每次只能传输一位，速率低

并行总线：优点：多位数据同时传输，速度快，不需要单独设计同步控制，开销低

缺点：数据线多，成本高，难以长距离传输；为防止冲突需要 DMA 等额外硬件开销

由于时钟频率较高，并行总线易受到干扰，且噪声较多，所以速度会慢于串行总线

2.

$$(1) \text{波特率} = 960 \times 10 = 9600$$

$$(2) \text{速率} = 9600 \times 7 \text{ bit/s}$$

$$= 6.72 \text{ KB/s}$$

3.

(1) 1位启动位，7位数据位，1位应答位，1位停止位构成。

(2) 因为 I2C 输入输出共用一根线，数据分时进行输入输出传输。

(3) 起始条件：SDA 线由高电平变为低电平，SCL 线为高电平

终止条件：SDA 线由低电平变为高电平，SCL 线为高电平

4.

$$(1) \text{MTTF}_{\text{RAID0}} = \frac{N}{4}$$

(2) 使用 RAID 10，即 RAID 1 + RAID 0，总容量为 $100 \text{ G} > 80 \text{ G}$ ，存储 2 对镜像。

$$\therefore \text{此时 } \text{MTTF}_{\text{RAID10}} = \frac{N^2}{2}$$



5.

寻道时间: 磁头从当前磁道移动到目标磁道的时间

影响因素: 磁头移动距离、加速度和定位精度

旋转时间: 等待数据从磁盘旋转到磁头的时间

影响因素: 磁盘转速, 数据所在扇区与磁头的距离

数据传输时间: 从磁盘读取或写入数据的时间

影响因素: 数据传输速度与数据块大小

6.

$$(1) 6 \times 240 \times 12 = 17280 \text{ KB} = 16.875 \text{ MB}$$

$$(2) 5400 \times 16.875 \text{ MB/min}$$

$$= 91.125 \text{ MB/min}$$

$$= 1518.75 \text{ MB/s}$$

$$(3) \frac{60}{5400} \approx 0.011 \text{ s}$$

8.

RAID 4 的写入优先级在于写入数据时只更新奇偶校验盘上的相关部分, 不用更新整个奇偶校验块。在顺序读取时不需要频繁访问奇偶校验盘, 而在随机的读取数据时, 就可以减少奇偶校验盘上的数据写入量, 可以缩短寻道时间和数据传输时间。



9.

当磁盘 I/O 请求减少时, λ 减小。当 μ 以相同幅度提升时, λ 减小时, μ 提升前后的 W 都不高, 所以 W 减小的幅度就较小, 性能提升的幅度也变小。

$$\begin{aligned}\text{幅度 } \Delta W &= \frac{1}{\mu_2 - \lambda} - \frac{1}{\mu_1 - \lambda} \\ &= \frac{\mu_1 - \mu_2}{(\mu_2 - \lambda)(\mu_1 - \lambda)} \\ &= \frac{\mu_1 - \mu_2}{\mu_2 \mu_1 - (\mu_1 + \mu_2)\lambda + \lambda^2}\end{aligned}$$

\therefore 当 λ 减小时, ΔW 变小。

10.

DMA 会与处理器争抢内存带宽资源。因为 DMA 通过直接内存访问传输数据, 会占用总线、I/O 设备, 而处理器与 DMA 会占用相同的总线资源。

存储层级设计可以有效改善这种竞争。例如使用多级缓存, 减少 CPU 和 DMA 对主存的访问, 减少竞争; 也可以使用缓存一致性协议、内存带宽管理等方法。



5-7 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？

当多个访盘请求在等待时，系统采用一定的策略，对这些请求的服务顺序进行调整安排，使寻道时间和延迟时间都尽可能小的那个访问请求可以优先得到服务，并降低若干个访问者的总访问时间，增加磁盘单位时间内的操作次数。达到降低平均磁盘服务时间的目的，从而实现公平、高效的访盘请求。

磁盘驱动调度由“移臂调度”和“旋转调度”两部分组成：移臂调度目的是尽可能减少寻找磁道时间；旋转调度目的是尽可能减少寻找扇区时间。

移臂调度：

先来先服务调度算法（FCFS）：按照访问请求的次序为各个进程服务。

优点：简单，公平；

缺点：效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间，对机械也不利。

最短寻道时间优先调度算法（SSTF）：优先选择距离当前磁头最近的访问请求进行服务，主要考虑寻道优先。

特点：改善了平均服务时间；

缺点：经常改变磁臂的移动方向，花费时间多又影响机械部件；还会导致“饥饿”现象，即较远距离的孤立访问可能很长时间不能获得访问磁盘的机会。

扫描算法（SCAN 电梯算法）：磁头按照一个方向移动，直到达到最边缘，然后折返，继续移动。它克服了 SSTF 算法的缺点，考虑了距离和方向。

特点：较好地解决了寻道性能，又防止了“饥饿”现象。

缺点：会出现刚访问过的柱面再次提出请求时，会等待较长的时间。

循环扫描算法（C-SCAN）：在 SCAN 的基础上改进的。磁臂改为单向移动，

由外向里。

特点：较好地解决了寻道性能，又防止了“饥饿”现象。不会让刚访问过的磁道再次提出访问请求时等待较长的时间。

缺点：会出现磁臂的“黏着”现象。

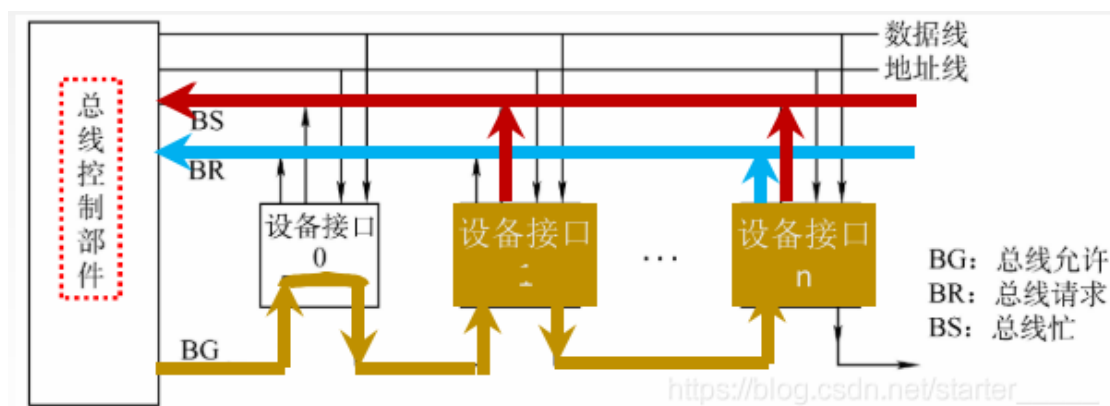
6-1 简述常见的总线仲裁机制，它们各有什么优缺点、分别适用于什么场景？

总线仲裁方式按其仲裁控制机构的设置可分为集中仲裁方式和分布仲裁方式。

集中仲裁方式：总线控制逻辑基本上集中于一个设备（如 CPU）中。将所有的总线请求集中起来，利用一个特定的裁决算法进行裁决，称为集中仲裁方式。集中仲裁方式有链式查询方式、计数器定时查询方式和独立请求方式三种。

(1) 链式查询方式

共用一根总线请求线 BR，通过总线允许线 BG 串行地传送响应信号。当总线上的部件需要使用总线时，经过总线请求线 BR 发送总线请求信号，若总线空闲时，则立即发送总线响应信号。当响应信号到达的部件有总线请求，则信号被截住，不再传下去，该设备获得总线控制权。主设备发送总线忙信号 BS，防止其它设备占用总线。



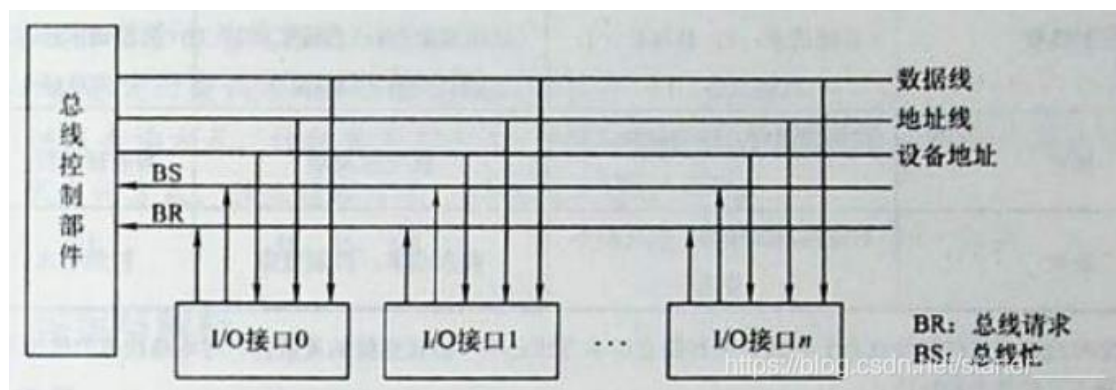
优点：链式查询方式优先级固定。只需很少几根控制线就能按一定优先次序实现总线控制，结构简单，扩充容易。

缺点：对硬件电路的故障敏感，并且优先级不能改变。当优先级高的部件频繁请求使用总线时，会使优先级较低的部件长期不能使用总线。

适用场景：系统设备较少，资源有限的情况。

(2) 计数器定时查询方式

共用一根总线请求线 BR，通过一个计数器和一组设备地址线产生并传送计数值（响应信号）。当总线上的部件需要使用总线时，经过总线请求线 BR 发送总线请求信号，若总线空闲时，则计数器开始计数，计数值通过设备地址线发向各个部件。当设备地址线上的计数值与请求使用总线设备的地址一致时，该设备获得总线控制权，同时中止计数器的计数及查询。主设备发送总线忙信号 BS，防止其它设备占用总线



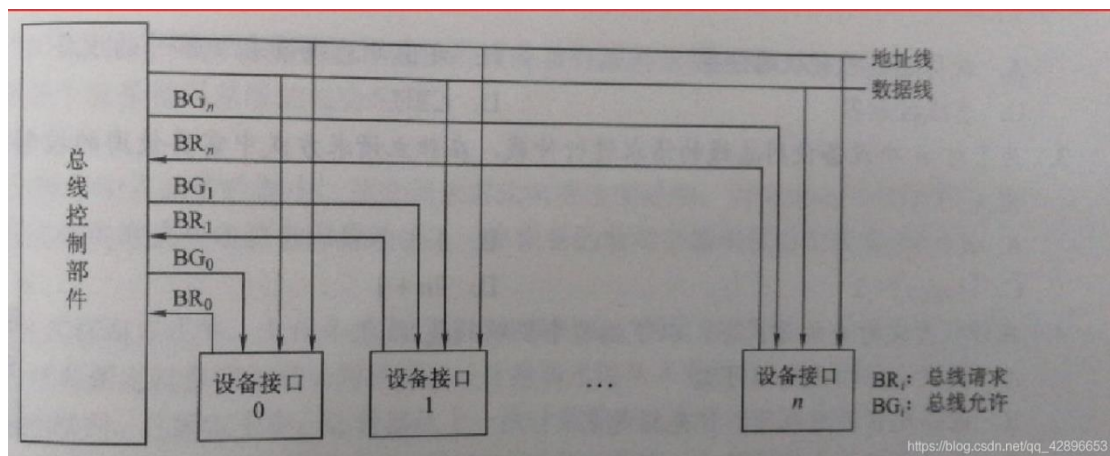
优点：计数初始值可以改变优先次序：计数每次从“0”开始，设备的优先级就按顺序排列，固定不变；计数从上一次的终点开始，此时设备使用总线的优先级相等；计数器的初值还可以由程序设置。对电路的故障没有链式敏感。

缺点：增加了控制线数，若设备有 n 个，则需 $\log_2 n + 2$ 条控制线。控制相对比链式查询相对复杂

适用场景：适用于设备数较多、比较要求公平性的情况

(3) 独立请求方式

每个设备均有一对总线请求线 BR_i 和总线允许线 BG_i ，通过各自的总线允许线 BG_i 传送响应信号。当总线上的部件需要使用总线时，经过各自的总线请求线 BR_i 发送总线请求信号，在总线控制器中排队，当总线控制器按一定的优先次序决定批准某个部件的请求时，给该部件发送总线响应信号。主设备发送总线忙信号 BS ，防止其它设备占用总线



优点：响应速度快，总线允许信号 BG 直接从控制器发送到有关设备，不必在设备间传递或者查询。对优先次序的控制相当灵活。

缺点：控制线数量多，若设备有 n 个，则需要 $2n+1$ 条控制线。其中+1 为 BS 线，其用处为，用于设备向总线控制部件反馈已经使用完毕总线。总线的控制逻辑更加复杂。

适用场景：优先级可设置，适用于对速度要求比较高的情况。

分布仲裁方式：不需要中央仲裁器，每个潜在的主模块都有自己的仲裁器和仲裁号，多个仲裁器竞争使用总线。

工作流程：当设备有总线请求时，它们就把各自唯一的仲裁号发送到共享的仲裁

总线上。每个仲裁器将从仲裁总线上得到的仲裁号与自己的仲裁号进行比较。如果仲裁总线上的号优先级高，则它的总线请求不予响应，并撤销它的仲裁号。最后，获胜者的仲裁号保留在仲裁总线上。

集中仲裁适用于设备数量较少、对于速度要求较低、可靠性要求较高的情况。

6-2 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。

为了减少功率的消耗，APB 的地址和读写控制信号在下一笔数据传递前，不会发生任何改变。适用场景为用于连接低速、低功耗、低复杂度的外设，如 UART、I2C、SPI 等。

AHB 协议需要一次突发传输的所有地址，地址与数据锁定对应关系，后一次突发传输必须在前次传输完成才能进行；

AXI 只需要一次突发的首地址，可以连续发送多个突发传输首地址而无需等待前次突发传输完成，并且多个数据可以交错传递，此特征大大提高了总线的利用率；

AHB 总线与 AXI 总线均适用于高性能、高带宽的 SoC 系统，但 AXI 具有更好的灵活性，而且能够读写通道并行发送，互不影响；更重要的是，AXI 总线支持乱序传输，能够有效地利用总线的带宽，平衡内部系统。因此 SoC 系统中，均以 AXI 总线为主总线，通过桥连接 AHB 总线与 APB 总线，这样能够增加 SoC 系统的灵活性，更加合理地把不同特征 IP 分配到总线上；

ACE 是在 AXI 协议基础上添加的一组协议扩展，用于处理多核处理器中的一

致性（coherency）和缓存一致性问题。ACE 协议支持多核处理器之间的数据一致性和缓存一致性维护，适用于连接多核处理器和高级缓存等组件。

CHI 是 AMBA 协议中最新的一种总线协议，用于支持更高性能和更复杂的 SoC 设计。CHI 协议提供高带宽、低延迟、高度可扩展的互连，支持高级一致性和缓存一致性协议，适用于连接具有复杂互连结构和高级一致性要求的处理器核心、缓存和外设。

6-3 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？

AXI 总线包含以下独立的事务通道：读数据通道、写数据通道、写响应通道。

协议没有设置独立的读响应通道，主要出于设计简化和减少总线引脚数量的考虑。由于读操作的响应数据在读数据通道上进行传输，因此可以通过读数据通道的有效数据和有效标志来传递读操作的响应信息，避免了额外的控制线和引脚。

(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？

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：对于读传输：读地址通道的有效信号必须先于读数据通道的有效信号。这确保了主设备在读取数据之前先提供有效的读地址。对于写传输：写地址通道的有效信号必须先于写数据通道的有效信号。这确保了主设备在写入数据之前先提供有效的写地址。

这样的依赖关系约束是为了确保读/写事务的正确执行。通过强制这些时序依赖，确保了在读取数据之前先提供有效的地址，并在写入数据之前先提供有效

的地址。这样可以保证总线传输的一致性和正确性。

(3) 什么是 AXI 的突发传输?有哪些突发传输类型?

AXI 的突发传输是指在单个总线事务中连续传输多个数据项。突发传输减少了总线开销, 并提高了数据传输的效率和带宽利用率。

AXI 总线支持以下突发传输类型:

固定突发: 在一个总线事务中连续传输固定数量的数据项。数据地址按照固定增量进行递增。

递增突发: 在一个总线事务中连续传输递增数量的数据项。数据地址按照递增增量进行递增。

非递增突发: 在一个总线事务中传输非递增数量的数据项。数据地址可能会在传输过程中保持不变或者以不连续的方式改变。