

第五章:

1. 串行总线将组成字符的各位串行地发往线路。

缺点: 传输速度较低, 一次一位地送入送出

优点: 通信成本低, 只需一个信道。支持长距离传输, 针脚少

并行总线的字符编码的各位同时传输

优点: 传输速度快: 一位的时间内可传输多个二进制位的数据

缺点: 通信成本高; 不支持长距离 (由于多信道之间的电容感应); 针脚多

串行接口的速率会比并行快, 主要有四个方面原因:

不需要时钟信号, 就没有时钟周期性的边沿, 频谱不集中, 噪声干扰少;

采用差分总线传输, 外界噪声加载到两条差分线上, 相减后可以抵消;

差分信号没有同步时钟, 不存在时钟和数据流的对齐问题。

线少, 干扰少。并行传输一般 32 根或者 64 根, 一根线跳变, 会给旁边的线带来噪声, 频率越高, 这种噪声越大, 很容易导致别的线值被篡改。

2. 1) $960 \times (1+7+1) = 9600 \text{ bps}$ 2) $960 \times 7 = 6720 \text{ bps}$

3.

1) 由启动信号; 数据字节; 应答位; 停止信号组成的, 数据字节常为 7 位, 其余常为 1 位

2) I2C 输入输出数据共用一根线, 数据可以沿着两个方向传输, 但要分时进行 (即半双工)

3) 起始: SDA 线由高电平切换成低电平, SCL 线由高电平切换成低电平

终止: SDA 线由低电平切换成高电平, SCL 线由高电平切换成低电平

4. 1) - 4 分钟, 后改为 $\frac{N}{4}$ 小时 2) 只用三个磁盘亦可满足 50 G 的需求, MTF 为 $\frac{N}{5}$ 小时.

5. 寻道时间: 磁头从当前位置移动到目标磁道的时间; 旋转时间: 移动到目标磁道后, 目标扇区转过磁头的过程并消除抖动. 数据传输时间: 磁头完成读出/写入的用时。前两个时间可让控制电路改变其次序而达到尽可能小, 后者主要取决于磁盘的材料特性和数据读写量.

6. 1) $6 \times 240 \times 12 \text{ KB} = 16.875 \text{ M}$. 2) 转一圈 $\frac{60}{5400} = \frac{1}{90} \text{ s}$, 故传输速率为 $\approx 1.48 \text{ G/s}$.

3) $60 \times 1000 \div 5400 \div 2 \approx 5.56 \text{ ms}$.

7. 磁盘控制电路常用的优化算法有以下几种:

1) 先来先服务算法: 根据请求到达的先后顺序进行调度。这种算法简单但效率低, 因为磁头可能来回地移动, 寻道时间长。

2) 最短寻道时间优先算法: 每次选择距离当前磁头位置最近的请求进行调度。这种算法可以提高效率, 减少寻道时间, 但是可能导致某些距离远的请求一直得不到服务。

3) 扫描算法: 磁头按照一个方向扫描整个磁盘, 依次服务沿途遇到的请求, 当到达边界时改变方向, 继续扫描并服务。可以避免 “饥饿” 现象, 但两端的请求等待时间较长。

8. RAID4 的写入优化是指在写入数据时，只更新奇偶校验磁盘上的相关部分，而不是整个奇偶校验块。这样可以减少写入操作对奇偶校验磁盘的负载，提高写入性能。

它对于读取速度的影响取决于读取操作的类型和频率。如果是顺序读取，那么写入优化对读取速度没有太大影响，因为顺序读取不需要频繁访问奇偶校验磁盘。如果是随机读取，那么写入优化会对读取速度有一定的提升，因为随机读取可能需要访问奇偶校验磁盘来恢复数据，而写入优化可以减少奇偶校验磁盘上的数据量，从而缩短寻道时间和传输时间。

9. 请求减少，平均响应时间增大，^{而磁盘的}平均响应时间不变，性能自然变低
从数学上说就是 $(\frac{1}{x-a})' = -\frac{1}{(x-a)^2}$ ， $x-a$ 为正时 a 越小负得绝对值越大，而 a 即是 I/O 请求数。

10. 在交错模式和突发模式下，DMA 均会占用访存带宽。
但如果层次设计较好的话，可以提高内存带宽资源的利用配置，减少同处理器的竞争。
存储器

第六章

1. 总线仲裁机制是用于解决多个主设备同时竞争总线控制权的问题，根据仲裁器的位置，可以分为集中仲裁和分布仲裁两种。

集中仲裁是指将所有的总线请求集中到一个中央仲裁器（通常是 CPU）进行处理，根据一定的算法决定哪个主设备优先获得总线控制权。集中仲裁又可以分为以下几种方式：

1) 链式查询：每个主设备共用一根总线请求线和一根总线响应线，总线控制器按照固定的顺序依次查询每个主设备是否有请求，直到找到第一个有请求的主设备，然后给它发送响应信号。这种方式优点是简单公平，缺点是效率低，对硬件故障敏感，优先级不能改变。

2) 计数器定时查询：每个主设备共用一根总线请求线，但有一组设备地址线，用于传输计数器的值。当有请求时，计数器开始计数，并将计数值与每个主设备的地址比较，如果相同，则给该主设备发送响应信号。这种方式优点是优先级可以改变，对故障不敏感，缺点是控制线多，控制复杂。

3) 独立请求：每个主设备都有一对总线请求线和总线允许线，当有请求时，通过自己的请求线向总线控制器发送信号，在控制器中排队，按照一定的优先级决定哪个主设备获得响应信号。这种方式优点是响应速度快，优先级灵活，缺点是控制线多，控制逻辑复杂。

分布仲裁是指每个主设备都有自己的仲裁器和仲裁号，并通过一个共享的仲裁总线进行比较。当有请求时，每个主设备都将自己的仲裁号发送到仲裁总线上，并与其他主设备的仲裁号进行比较。如果发现自己的仲裁号优先级低，则撤销自己的请求，并关闭输出级。最终只有一个最高优先级的主设备保留在仲裁总线上，并获得总线控制权。这种方式优点是不需要中央仲裁器，缺点是需要额外的仲裁总线和仲裁器。

一般来说，集中仲裁适用于主设备数量较少、通信速率较低、可靠性要求较高的场合；分布仲裁适用于主设备数量较多、通信速率较高、可靠性要求较低的场合。

2.

- 1) APB: 高级外设总线, 用于连接低速、低功耗、低复杂度的外设, 如 UART、I2C、SPI 等。APB 不具备流水线功能, 传输速率较低, 但接口简单, 占用资源少。
- 2) AHB: 高级高性能总线, 用于连接高速、高带宽、高复杂度的设备, 如 CPU、DSP、DMA、RAM 等。AHB 具备流水线功能, 传输速率较高, 支持突发传输和分裂传输, 但接口较复杂, 需要仲裁器和解码器。
- 3) AXI: 高级可扩展接口, 是 AHB 的改进版本, 用于连接更高速、更高带宽、更高复杂度的设备, 如多核 CPU、GPU、FPGA 等。AXI 具备流水线功能, 传输速率更高, 支持突发传输和分裂传输, 同时增加了独立的读写通道和响应通道, 提高了总线效率和灵活性。
- 4) ACE: AXI 一致性扩展, 是 AXI 的扩展版本, 用于支持多核处理器之间的缓存一致性协议。ACE 在 AXI 的基础上增加了一致性请求通道和一致性响应通道, 以及一些新的控制信号和状态信号, 实现了缓存共享和缓存维护等功能。
- 5) CHI: 一致性集线器接口, 是 ACE 的替代版本, 用于支持更大规模的多核处理器之间的缓存一致性协议。CHI 在 ACE 的基础上增加了源同步时钟域交叉和动态服务质量控制等特性, 提高了总线性能和扩展性。

3.

- 1) 分别是读地址通道、读数据通道、写地址通道、写数据通道和写响应通道。

读响应只需要一个 2 位的信号来表示读传输的状态, 而且读响应必须与读数据同时返回, 所以将它合并到读数据通道中, 不会影响传输效率, 但是简化了接口和节省资源。

- 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:

读地址通道: 主设备在给出有效的地址和控制信号之前, 必须等从设备准备好接收。从设备在给出准备好接收之前, 必须等待主设备给出有效的地址和控制信号。

读数据通道: 从设备在给出有效的数据和响应信号之前, 必须等待主设备准备好接收。主设备在给出准备好接收之前, 必须等从设备给出有效的数据和响应信号。

写地址通道: 主设备在给出有效的地址和控制信号之前, 必须等从设备准备好接收。从设备在给出准备好接收之前, 必须等待主设备给出有效的地址和控制信号。

写数据通道: 主设备在给出有效的数据和字节选通信号之前, 必须等从设备准备好接收。从设备在给出准备好接收之前, 必须等待主设备给出有效的数据和字节选通信号。

写响应通道: 从设备在给出有效的响应信号之前, 必须等待主设备准备好接收。主设备在给出准备好接收之前, 必须等从设备给出有效的响应信号。

设置这样的约束的目的是为了实现握手机制, 保证每次传输都有明确的开始和结束, 避免数据丢失或冲突。

- 3) AXI 的突发传输是指一次事务中连续传输多个数据拍的操作。突发传输可以提高总线利用率和性能, 同时减少地址信息的开销。AXI 有四种突发传输类型, 分别是:

单拍突发: 只传输一个数据拍, 没有突发长度和突发大小的限制。

递增突发: 传输多个数据拍, 每个数据拍的地址都是在前一个数据拍的地址基础上递增一个突发大小的值, 突发长度可以是 1 到 256 之间的任意值。

固定突发: 传输多个数据拍, 每个数据拍的地址都是相同的, 突发长度可以是 1 到 16 之间的任意值。

回环突发: 传输多个数据拍, 每个数据拍的地址都是在一个突发边界内循环递增一个突发大小的值, 突发长度可以是 1 到 16 之间的任意值。