

第五章、

1,

串行:

优点: 需要物理连线数目少, 消耗硬件资源少, 功耗低。

缺点: 相同频率下, 传输速率慢

并行:

优点: 可以同时传输更多数据, 相同频率下传输速率快

缺点: 需要物理连线数目和消耗硬件资源多, 功耗高。

并行总线由于是多个数据同时传输, 需要考虑数据的协同性, 并行总线两根相邻的链路其数据是同时传输的, 这就会导致它们彼此之间会产生严重干扰, 并行的链路越多, 干扰越强, 这就导致了并行传输的频率不能做的很高。相对的, 串行总线只有一条链路, 就可以把频率做的很高, 提高传输速度

3,

I2C 通信的数据包大小为 8bit, 主要有三类——指令、字节地址、数据。数据传输时, 按照高位在前, 低位在后的顺序 (即 MSB First, LSB Last)

I2C 通信中, 数据在一根数据线 SDA 上传输, 同一时刻数据传输的方向只能是单向的, 从 A 到 B 或者从 B 到 A; 通过切换传输方向从而实现双向通信, 因此 I2C 属于半双工通信。

启动信号

在时钟线 SCL 保持高电平期间, 数据线 SDA 上的下降沿, 定义为 I2C 总线的启动信号, 它标志着一次数据传输的开始。启动信号是由主机建立的, 在建立该信号之前, I2C 总线必须处于空闲状态。

停止信号

在时钟线 SCL 保持高电平期间, 数据线 SDA 上的上升沿, 定义为 I2C 总线的停止信号, 它标志着一次数据传输的终止。停止信号是由主机建立的, 建立该信号之后, I2C 总线将返回空闲状态。

7,

磁盘带宽是传输字节的总数除以从服务请求开始到最后传递结束时的总时间。通过管理磁盘 I/O 请求的处理次序, 可以改善访问时间和带宽。

以 SSTF 为例, 在移动磁头到别处以便处理其他请求之前, 处理靠近当前磁头位置的所有请求可能较为合理。这个假设是最短寻道时间优先(SSTF)算法的基础。

8,

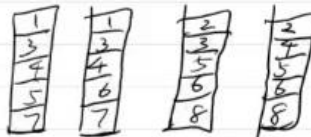
RAID4 模式同 RAID3 几乎是一样的, 数据都是分成小的数据块依次存储在多个硬盘之上, 奇偶校验码存放在独立的奇偶校验盘上。唯一不同的是, 在数据分割上 RAID3 是以 bit 为单位而 RAID4 是以 Byte 为单位。这样可以使得 RAID4 同 RAID3 具有一样的读取速度, 当然写入性能因为需要在写入过程中产生校验码并且存储到校验盘而受到了影响。

这种模式的最大好处就是不需要各个硬盘之间在转速上保持同步, 这就使得控制器不需要那么复杂。它的写入性能是所有 RAID 模式中最差的。

2. (1) $1+7+1+1=10$
 $10 \times 960 = 9600$
 波特率为 9600 (Baud) $S = 9600 \text{ Baud}$
 (2) $I = S \cdot \log_2 N = 9600 \log_2 2 = 9600 \text{ bps}$

4. (1) 无冗余备份数据
 仅数据分散
 $MTTF = \frac{N}{4}$

(2) $MTTF = \frac{N}{4} \left(\frac{2^4}{8} + \frac{3^4}{8} \right)$
 $= \frac{N}{4} \left(1 + \frac{3}{2} \right) = \frac{5}{8} N$



5. 寻道时间: 磁头臂移动到正确位置并消除抖动所需时间

旋转时间: 寻到正确磁道后, 盘片需要转过, 旋转来使得正确的扇区
 旋转到磁头的正下方

数据传输时间: 在扇区内开始按顺序读写, 因盘片需要旋转因此
 有时间

寻道时间: 与磁头臂需要移动距离和移动速度有关

旋转时间: 盘片旋转速度

数据传输时间: 盘片旋转速度

6. 5400 r/min 6面 240道 12KB
 1) $12 \text{ KB} \times 240 \times 6 = 16.875 \text{ MB}$

2) 传输速率: 1280 KB/s

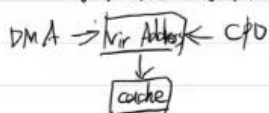
3) 平均旋转时间: $\frac{60}{5400} \times \frac{1}{2} = \frac{1}{1800} \text{ s} = 0.00055 \text{ s}$

9. 请求减少时, 到达率 λ 减少
 没有队列时 为 μ

有队列时 为 μ/λ $\lambda \downarrow$ $\mu/\lambda \uparrow$ $w \uparrow$
 μ 不变 w 个, 性能提升幅度下降

12. 有可能

缓存存储器层次设计, 可以在硬件层面上由虚拟存储系统
 对内存进行标记, 来决定 DMA 不能使用 CPU 已占用内存



第六章

1,

仲裁方式可分为集中仲裁方式和分布式仲裁方式。

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 3 种。

链式查询方式：优点：优先级固定，结构简单，扩充容易；缺点：对电路故障敏感，优先级不灵活。

计数器定时查询：优点：优先级灵活；缺点：控制线多，控制复杂

独立请求：优点：响应速度快，优先级灵活；缺点：控制线多，控制复杂

分布式仲裁方式：优点:优先权控制灵活；缺点:硬件成本增加,控制复杂度高。为基础适用

2,

APB：是低带宽总线协议，为了支持外围设备，在低功耗和低复杂性方面进行了特定的优化。读写操作共享同一组信号，不支持 burst 数据传输。它用作外围设备的低成本接口，不需要大量数据，低延时的传输，因此不需要高性能的流水线总线接口。

AHB：主要是针对高性能、高时钟频率及快速系统模块所设计的总线，它充当高性能系统主干总线，可以连接如微处理器、芯片上或芯片外的内存模块和 DMA 等高效率模块。它支持多个总线主控并支持高带宽操作。

AXI：是一种并行、高性能、高频率、同步的总线协议。适合于高带宽和低延迟互连。这是一个点对点的互连协议，支持乱序传输、读写通道分离，该协议支持多个 outstanding 的数据传输（多组未完成事务共存）、burst 数据传输、单独的读写通道和支持不同的总线宽度。它针对高性能和高频系统，具有以下主要特点：多笔未完成交易；乱序数据补全；仅发布起始地址的基于突发的事务；支持使用选通的非对齐数据传输；同时读取和写入事务；用于高速操作的流水线互连

ACE：AXI4 协议的扩展，应用于在一个芯片上集成多个 CPU 核心与一致性 cache 的场景。ACE 协议扩展了 AXI 读写数据通道，同时引入了单独的 snoop 地址、snoop 数据和 snoop 响应通道。这些额外的通道提供了实现基于 snoop 的一致性协议的机制。ACE 一致性协议确保所有 master 都能看到任何地址位置的正确数据。这避免了对缓存之间主要一致性的软件缓存维护。ACE 还提供屏障事务来保证系统内多个事务的排序，以及分布式虚拟内存 (DVM) 功能来管理虚拟内存。

CHI：定义了连接完全一致的处理器接口。它是一种基于数据包的分层通信协议，具有协议、链路和网络层。它与拓扑无关，并提供基于服务质量 (QoS) 的机制来控制系统中的资源。它支持处理器之间的高频和非阻塞相干数据传输，为数据中心等应用提供性能和规模。

3,

1) AXI4 和 AXI4-Lite 接口包含 5 个不同的通道：两个读通道和三个写通道。每一个 AXI 传输通道都是单方向的。每一个传输都有地址和控制信息在地址通道 (address channel) 中，用来描述被传输数据的性质。两个读通道：读地址通道(read address channel)、读数据通道(read data channel)。三个写通道：写地址通道(write address channel)、写数据通道(write data channel)、写响应通道(write response channel)。

读数据通道传送着从模块到主模块的读数据和读响应信息。

2) 五个独立的通道均使用 VALID/READY 进行握手，VALID 由源端产生，用于指示源端发出的地址、数据、控制信息什么时候生效，READY 则由目的端产生，用于指示源端发送信息什么时候被接收，只有当 VALID 和 READY 同时为高时，才会表示本次传输完成。

对于五个通道之间的关系，AXI 协议仅定义了下面的关系：写响应必须跟随写操作的最后一个传输。读数据必须紧跟读地址数据。

除了上述两点定义，AXI 未定义其他任何通道间的关系，那么对于通道间的握手信号先后顺序，必须按照特定的顺序操作，不然很容易引起接口上的 dead-lock。

3) AXI 协议是基于突发的，Master 通过驱动传输控制信息和传输中第一个字节的地址来开始每个突发。随着突发事务的进行，Slave 负责计算突发中后续传输的地址。突发不得跨越 4KB 边界，以防止它们跨越 Slave 之间的边界并限制 Slave 内所需的地址增量器的大小。

AXI 协议定义了三种突发类型，如下所述：固定突发（地址固定的突发）；增量突发（地址递增的突发）；回环突发。（在回环边界处回绕到较低地址的递增地址突发）