

第五章

1.

串行总线:

- 优点: 可以使用较少的线路传输大量数据, 因此可以减少硬件成本, 同时功耗也更低。
- 缺点: 相同频率下, 传输速度比较慢, 传输需要较长的时间。

并行总线:

- 优点: 传输速度很快。
- 缺点: 需要使用更多的线路传输数据, 因此需要更多的物理连线 and 硬件成本, 功耗比较高; 同时不同信号线之间会有干扰, 不利于长距离传输。

2

1) 每个数据包的大小为 10 Bit.

因此波特率为: $10 \times 960 = 9600$

2) 有效数据传输速率: $7 \times 960 = 6720 \text{ Bit/s}$

3

1) I2C的数据包由两部分组成: 地址和数据。地址部分指定了要发送或接收数据的设备, 数据部分包含要发送或接收的实际数据。

2) 因为I2C只使用了两根线(时钟线SCL和数据线SDA)进行通信, 这两根线都是双向的。由于只有一根数据线, 所以在同一时间只能进行读取或者写入操作, 而不能同时进行读取和写入。因此I2C是半双工的, 在同一时间只能进行单向数据传输。

3) 起始条件 (Start Condition): 在I2C通信开始时, 主设备向从设备发送一个起始条件, 即在SCL为高电平时, SDA从高电平转换为低电平

停止条件 (Stop Condition): 在I2C通信结束时, 主设备向从设备发送一个停止条件, 即在SCL为高电平时, SDA从低电平转换为高电平。

4

1) $MTTF: \frac{N}{4}$

2) 磁盘空间足够大, 因此采用 RAID 1 的模式, 用 2 块磁盘存储系统需要的数据, 另外两块存储冗余数据作为备份。因此 $MTTF = 2 \times N / 2 = N$ 小时。

5

寻道时间：磁头臂移动到正确位置并消除抖动所需要的时间。

旋转时间：定位到正确磁道后，盘片旋转使得正确扇区被旋转至磁头的正下方所消耗的时间。

数据传输时间：目标扇区成功定位后，传送一个扇区的数据所消耗的时间。

影响因素：

①磁头臂的移动速度与移动距离，影响寻道时间。

②盘片的旋转角度与旋转速度，影响旋转时间与数据传输时间。

6

1) 总容量： $6 \times 240 \times 12 \text{KB} = 16.875 \text{MB}$

2) 数据传输速率： $5400 \div 60 \times 6 \times 12 \text{KB} = 6.328 \text{MB/s}$

3) 所需数据距磁头的平均角度为 180° ，因此平均旋转时间： $\frac{0.5 \text{r}}{5400 \text{r/min}} \times 60 = 5.56 \text{ms}$ 。

7

第一，磁盘控制电路可以想换存一段时间内磁盘I/O请求并决定请求的最优执行次序，以达到最短的磁头移动距离从而减少寻道时间；第二，控制器还可以优化执行次序，以达到最小的盘片转动次数，从而优化磁盘旋转时间。

8

写入优化对读取速度的影响不大。因为当主机需要读取数据时，RAID4仍然需要从所有的数据磁盘和奇偶校验磁盘中读取数据块和相应的奇偶校验码，并使用奇偶校验码来检查数据的正确性和恢复数据。因此，RAID4的读取速度不会受到针对写入的优化的影响。

9

$W = \frac{1}{\mu - \lambda}$ ，其中 W 为平均响应时间， μ 为服务率， λ 为平均到达率。

当I/O请求减少时， λ 下降，因此 W 会上升，即平均响应时间变大，所以队列性能提升幅度下降。

10

可能会，因为DMA设备与处理器共享总线，在进行数据冲突时都需要占用内存带宽资源，因此有可能导致冲突。较好的存储层次设计可以通过优化内存带宽资源的分配与管理，减少DMA设备与处理器之间的竞争。另外，良好的存储层次设计能够避免引入DMA导致的缓存一致性问题。

第六章

1

集中式仲裁：

- 由一个中央控制器负责协调总线上的所有设备的访问。
- 优点：实现简单。
- 缺点：当总线上的设备数量增加时，中央控制器的负担也会增加。
- 适用场景：适用于总线上设备数量较少的场景

分布式仲裁：

- 每个设备都有权利请求总线访问，总线会根据一定的规则选择一个设备进行访问。
- 优点：分布式，不会出现中央控制器性能瓶颈。
- 缺点：需要设备之间协调，实现比较复杂。
- 场景：适用于总线上设备数量较多的场景。

令牌传递：

- 在总线上传递一个令牌，只有持有令牌的设备才能访问总线。
- 优点：实现简单，同时能够保证总线访问的顺序。
- 缺点：当持有令牌的设备出现故障时，总线会变得不可用。
- 场景：适用于需要保证总线访问顺序的场景。

CSMA/CD：

- 设备在访问总线之前会先监听总线，如果总线上没有其他设备正在访问，则可以进行访问。如果同时有多个设备访问总线，则会发生冲突，需要等待一段时间后重新尝试。
- 优点：实现简单，适用性广，扩展性强。
- 缺点：当总线上的设备数量增加时，冲突的概率也会增加，可能会影响总线的性能。
- 场景：适用于总线上设备数量较少且访问频率较低的场景。

2

- APB：APB的特点是简单、低功耗、低成本，但是带宽较低，不适用于高速数据传输。适用于连接低带宽外设。
- AHB：特点是高性能、高带宽、支持多主设备，但是复杂度较高，成本较高。适用于连接中等带宽的外设。
- AXI：特点是高性能、高带宽、低延迟、支持多主设备，但是复杂度较高，成本较高。适用于连接高带宽的外设。
- ACE：特点是支持高级缓存一致性，可以提高系统性能，但是复杂度较高，成本较高。
- CHI：高性能、高带宽、低延迟、支持高级缓存一致性、支持多处理器系统，但是复杂度和成本都较高。适用于用于连接高性能处理器和高速存储器。

(1) 独立的事务通道：

1. 读通道 (Read Channel)：用于从存储器或外设中读取数据。
2. 写通道 (Write Channel)：用于向存储器或外设中写入数据。
3. 写唯一通道 (Write Unique Channel)：用于向存储器或外设中写入唯一的数据，例如写入控制寄存器。
4. 读写通道 (Read-Write Channel)：用于同时进行读和写操作，例如从存储器中读取数据并进行修改后写回存储器。
5. 保留通道 (Reserved Channel)：保留用于未来的扩展。

AXI总线不设置独立的读响应通道是因为在AXI总线中，读响应的数据和读请求的地址和控制信号是在同一个通道中传输的。当AXI总线上的主设备发出读请求时，从设备会在同一个通道中返回读响应，包括读取的数据和响应状态。这种设计可以减少总线的复杂度和成本，同时也可以提高总线的带宽和性能，因为读响应的数据可以立即返回，而不需要等待独立的响应通道。此外，AXI总线还支持乱序访问，可以在等待读响应时同时进行其他的读写操作，从而进一步提高总线的性能。

(2) 依赖关系：

1. 在读事务中，读请求通道 (AR) 和读响应通道 (R) 的时序是依赖的，即读响应通道必须在读请求通道之后才能发送。
2. 在写事务中，写请求通道 (AW) 和写数据通道 (W) 的时序是依赖的，即写数据通道必须在写请求通道之后才能发送。
3. 在写事务中，写响应通道 (B) 和写请求通道 (AW) 的时序是独立的，即写响应通道可以在写请求通道之前或之后发送。

原因：这些依赖关系的设置是为了保证AXI总线的正确性和可靠性。在读事务中，读响应通道必须在读请求通道之后发送，否则主设备可能会读取到错误的数据。在写事务中，写数据通道必须在写请求通道之后发送，否则从设备可能会写入错误的数据。因此，这些依赖关系的设置可以保证AXI总线的正确性和可靠性，避免数据传输错误和系统故障。

(3) AXI的突发传输是指在一次总线事务中，主设备可以连续地读取或写入多个数据，从而提高总线的带宽和性能。类型：

1. 固定突发传输 (Fixed Burst)：在一次总线事务中，主设备可以连续地读取或写入固定数量的数据，例如4个、8个或16个数据。
2. 不规则突发传输 (Unaligned Burst)：在一次总线事务中，主设备可以连续地读取或写入不规则的数据，例如读取或写入不连续的字节或半字。
3. 增量突发传输 (Incremental Burst)：在一次总线事务中，主设备可以连续地读取或写入递增的地址，例如读取或写入连续的字或半字。