

第5章 习题

1. 串行总线优点: ① 简化布线: 只需要一个传输线路, 减少了线路的复杂性

② 所需线路和接口较少, 因此成本上比较经济

并行总线优点: ① 高带宽: 可以同时传输多个比特位, 适用于需要高速传输的情形

② 低延迟: 并行总线同时传输多个比特位, 可以降低传输延迟

接口速率不同: 因为数据传输方式上的差异: 串行总线按顺序传输比特位, 而并行总线同时传输多个比特位

2. (1) 1个数据包有10位, 波特率 = $10 \times 960 = 9600 \text{ bit/s}$

(2) 7位有效数据, 速率 = $7 \times 960 = 6720 \text{ bit/s}$

3. (1) start condition; address byte; acknowledgement bit; data byte; stop condition

(2) ① 线路和硬件简化: 半双工只需要两根线, 一根SDA, 一根SCL

② 节省引脚和资源

③ 适用大多数场景: I²C的典型是主从设备间的短距离通信, 半双工满足了这种需求

(3) 通过电平变化: SCL高电平时, SDA高 \rightarrow 低: 起始条件;

SCL高电平时, SDA低 \rightarrow 高: 停止条件.

4. (1) 每小时每块磁盘故障次数为 λ , 4块为 λ , $MTTF = \frac{N}{\lambda}$

(2) 可以考虑让两个磁盘存放同样的数据, 4个磁盘, 相当于100G容量, 100G备份

存储相同数据的同时故障才会导致错误: 概率为 $\lambda \cdot \lambda = \lambda^2$, 2组为 $\frac{2}{N}$, $MTTF = \frac{1}{2}N^2$

5. 寻道时间: 磁头从当前位置移动到目标磁道并消抖所需的时间

旋转时间: 磁头移动到目标磁道后, 目标扇区随盘片转动而经过磁头的时间

数据传输时间: 磁头完成读出或写入所需时间。

可以通过更精密的控制电路来减少寻道时间、旋转时间。

6. (1) $6 \times 240 \times 12 \text{KB} \approx 16.9 \text{MB}$

(2) 1r 对应 12KB, 速率为 $12 \text{KB/r} \cdot 5400 \text{r/min} = 64800 \text{KB/min} \approx 63 \text{MB/min}$

(3) 假设旋转时间内读完所有数据:

$$t = \frac{6 \times 240 \text{r}}{5400 \text{r/min}} = 10 \text{s}$$

7. ① SSTF: 最短寻道时间优先, 选择距离当前磁头位置最近的请求进行执行

② Scan Algorithm: 磁头按磁道移动, 连续执行请求, 到达边缘后改变移动方向

③ Circular Scan Algorithm: 到达边缘后, 直接返回起始位置再执行请求.

8. 这种无需读取磁盘上的数据来重新计算奇偶校验位的做法, 优化了对磁盘的^{读取操作}写入速度, 因此可以提高写入速度。然而对读取速度影响较小, 读取时仍需要取数据块和奇偶校验块来还原原始数据。

9. I/O 请求减少, 即 λ (每秒发送的 I/O 请求数) 下降, $W = \frac{1}{\mu - \lambda}$ 也会下降, 即队列等待时间下降, 即磁盘队列系统的性能将会提升

I/O 请求减少前: $W_{11} = \frac{1}{\mu_1 - \lambda_0}$, $W_{12} = \frac{1}{\mu_2 - \lambda_0}$, 提升幅度为 $\frac{W_{11} - W_{12}}{W_{11}} = \frac{\mu_2 - \mu_1}{\mu_2 - \lambda_0}$

减少后: $W_{21} = \frac{1}{\mu_1 - \lambda}$, $W_{22} = \frac{1}{\mu_2 - \lambda}$, 提升幅度为 $\frac{W_{21} - W_{22}}{W_{21}} = \frac{\mu_2 - \mu_1}{\mu_2 - \lambda}$

λ 减少, 则提升幅度也会减少

(增大 μ 即增大处理速度)

10. 如果 DMA 设备和处理器使用相同的存储器层次, 可能会导致资源争用, 即争夺高速缓存的访问权。

而良好的存储器结构可以提供足够的带宽和容量, 满足设备的需求, 而减少争用。

6-1

1.集中式仲裁：

集中式仲裁是由一个中央仲裁器负责处理总线上的仲裁请求。常见的集中式仲裁机制有轮询、固定优先级和仲裁请求线的方式。

优点：简单、易实现、低成本。

缺点：对于请求密集型系统，中央仲裁器可能成为性能瓶颈；请求的响应时间可能不公平，低优先级的请求可能长时间等待。

适用场景：对于规模较小、请求较为均匀分布的系统，集中式仲裁是一种经济高效的选择。

2.分布式仲裁：

分布式仲裁是将仲裁功能分散到总线上的各个设备上，设备之间通过协议进行仲裁。常见的分布式仲裁协议有请求-响应和令牌传递。

优点：减轻中央仲裁器的负担，提高系统吞吐量；具有一定的公平性，每个设备有机会获得总线访问权。

缺点：实现复杂，对系统的时序和同步要求较高。

适用场景：对于请求密集型系统，分布式仲裁能够提高总线的并发性能，适用于对总线访问有严格要求的场景。

3.基于优先级的仲裁：

基于优先级的仲裁是通过为设备分配优先级来进行仲裁。常见的方式有固定优先级和动态优先级。

优点：可以灵活地调整设备的优先级，满足不同设备的访问需求；能够提供优先级的公平性。

缺点：需要进行优先级的管理和配置，增加了设计和管理的复杂性。

适用场景：对于系统中具有不同访问需求和优先级的设备，基于优先级的仲裁能够灵活地满足各设备的需求。

6-2

APB (Advanced Peripheral Bus)：

特点：APB 是一种低功耗、低带宽的串行总线协议，适用于连接低速外设，如控制寄存器和配置寄存器等。

使用场景：APB 常用于与外设通信的低速接口，如 GPIO（通用输入/输出）、UART（通用异步收发传输器）等。

AHB (Advanced High-performance Bus)：

特点：AHB 是一种高性能、低延迟的并行总线协议，支持多主设备和多从设备，并具有流水线传输和分段传输特性。

使用场景：AHB 适用于连接高性能设备，如处理器、DMA 控制器和高速存储器等。在复杂的系统中，AHB 还可以用作内部总线进行设备之间的通信。

AXI (Advanced eXtensible Interface):

特点: AXI 是一种高性能、高带宽、流水线传输的总线协议, 支持多个主设备和从设备, 并提供优先级和突发传输特性。

使用场景: AXI 广泛应用于高性能系统, 如高性能处理器、高速存储器控制器、图像处理器和网络处理器等。

ACE (AXI Coherency Extension):

特点: ACE 是在 AXI 基础上添加了一致性 (coherency) 支持的扩展, 用于处理多核处理器和高速缓存一致性问题。

使用场景: ACE 主要用于多核处理器系统, 支持高速缓存的一致性管理, 确保多个核心的缓存内容一致。

CHI (Cache Coherent Interconnect):

特点: CHI 是一种高性能、高带宽的缓存一致性互联协议, 支持多个处理器核心和高速缓存的一致性管理。

使用场景: CHI 主要应用于高性能多核处理器系统和高级服务器, 具有更高的性能和可扩展性, 用于处理复杂的缓存一致性场景。

6-3

(1)

AXI 总线包含的独立的事务通道:

1. 读通道 (Read Channel): 用于从主设备读取数据的事务, 包括读地址和读数据传输。

2. 写通道 (Write Channel): 用于向主设备写入数据的事务, 包括写地址、写数据传输和写响应。

3. 写唤醒通道 (Write Widen Channel): 用于写传输的扩展通道, 支持高带宽的写操作, 可以同时传输多个数据。

4. 写回通道 (Write Response Channel): 用于返回写操作的响应, 包括成功或失败的状态。

不设置独立的读响应通道:

1. 提高效率: 将读响应与读数据合并可以在同一个通道中可以减少总线的开销和复杂性。读响应可以直接与读数据一起返回, 减少了独立传输响应的开销。

2. 降低时序延迟: 合并读响应和读数据可以减少时序延迟, 提高总线的性能。读数据和读响应可以同时在线上传输, 减少了传输延迟。

3. 节省总线带宽: 将读响应与读数据合并在一起可以节省总线带宽。由于读数据和读响应通常是一一对应的关系, 所以共享同一个通道可以节省带宽资源。

(2)

读事务:

读地址 (ARADDR) 信号应该在读通道的起始点上升沿之前稳定。

读数据 (RDATA) 信号应该在读通道的起始点上升沿之后的一段时间内有效, 保持稳定直到读通道的结束点。

写事务:

写地址 (AWADDR) 信号应该在写通道的起始点上升沿之前稳定。

写数据 (WDATA) 信号应该在写通道的起始点上升沿之前稳定。

写响应 (BRESP) 信号应该在写通道的结束点上升沿之后的一段时间内有效, 保持稳定直到写通道的结束点。

这些时序依赖关系的约束是为了确保读/写传输的正确性和一致性。具体原因如下:

1. 确保数据的准确性: 在读事务中, 读地址信号必须在读通道开始之前稳定, 以确保主设备能够正确地识别要读取的地址。类似地, 在写事务中, 写地址和写数据信号必须在写通道开始之前稳定, 以确保主设备正确地识别要写入的地址和数据。

2. 防止数据冲突: 为了避免数据冲突, 读数据信号应该在读通道开始后的一段时间内有效, 以确保从设备能够在读通道期间提供正确的数据。类似地, 在写事务中, 写响应信号应该在写通道结束后的一段时间内有效, 以确保主设备能够正确地接收到写操作的结果。

3. 时序一致性: 通过定义严格的时序约束, 可以确保各个通道的数据和信号在正确的时间点进行传输, 从而保持读/写传输的一致性和正确性。

(3)

在 AXI 总线中, 突发传输可以提高数据传输的带宽利用率和性能效率。传统的非突发传输方式需要为每个数据或地址传输发起一个独立的事务, 而突发传输可以在一个事务中传输多个数据或地址, 从而减少了总线的开销。

AXI 定义了以下几种突发传输类型:

1. INCR (Incrementing): 在这种突发传输类型中, 连续的地址按照递增的方式传输, 每个传输的地址值比前一个地址增加一个固定的偏移量。通过这种方式, 可以连续地读取或写入一系列相邻的地址。

2. WRAP (Wrapping): 在这种突发传输类型中, 地址按照循环的方式传输。当到达地址范围的末尾后, 下一个地址将回到范围的起始点。通过这种方式, 可以循环地读取或写入地址范围内的数据。

3. FIXED (Fixed): 在这种突发传输类型中, 所有的传输地址都是相同的, 传输的数据或者传输的地址为固定值。通过这种方式, 可以多次读取或写入同一个地址的数据。