

第五章

1. 串行总线与并行总线:

串行总线优点: 物理连线数目少, 消耗硬件资源少, 功耗低, 可以以较高的频率传输

串行总线缺点: 在相同频率下传输速度更慢

并行总线优点: 可以同时传输更多的数据, 相同频率下传输速度更快

并行总线缺点: 物理连线更多, 硬件资源消耗大, 功耗高, 容易出现串扰, 接口同步电路难以实现, 有效传输距离和传输频率难以进一步提高

接口速率不同的原因:

1. 带宽不同: 串行总线一次只能传输单个位, 而并行总线一次能同时传输多个位
2. 传输频率: 并行总线需要克服接口同步已经串扰等问题, 传输频率不能进一步提高, 而串行总线可以做到很高的频率
3. 传输距离: 串行总线可以实现较长的传输距离, 而并行总线传输距离有限, 选择不同的传输距离会影响总线的接口速率

3. (1) I2C 的数据包由一个起始条件和终止条件, 中间先有一个 7 到 10 位的地址帧, 然后有规定任务的读/写位, 之后每过一个 ACK/NACK 位后跟一个 8 位的数据位

(2) I2C 是非常简化的, 只使用一根数据线 SDA, 所以数据一次只能从一个设备发给另一个设备, 所以是半双工的

(3) 起始条件: 当 SCL 为高电平时, SDA 由高向低跳变; 终止条件: 当 SCL 为高电平时, SDA 由低向高跳变。其中 SCL 为串行时钟线, SDA 为串行数据线

7. 磁盘控制电路需要先缓存一段时间内磁盘的 I/O 请求, 然后在这部分请求里去交由控制电路经过优化算法获得最佳的执行次序, 以达到最短的磁头移动距离和最小的盘片转动圈数。

8. RAID4 的写入优化:

因为有一个专门的奇偶校验磁盘来存对应块数据整体的异或值, 所以为了保证奇偶校验位的有效性, 每当磁盘发生数据写入导致改变时, 都需要重新读取整个条带的磁盘数据, 做异或后存给奇偶校验磁盘, 导致速度大大变慢。不过实际上整个条带中只有那个被改变的磁盘的数据发生变化并影响奇偶校验磁盘, 所以实际上可以只用先读取即将被写入的磁盘的数据, 与新写入的数据作比较, 计算需要翻转的位, 然后即可判断新的奇偶校验位要被写入的数据, 这样一来, 只需要对被写入的磁盘以及奇偶校验磁盘进行操作, 优化了速度

第六章

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?

常见的总线仲裁机制包括集中式仲裁、分布式仲裁和旁路仲裁。

1. 集中式仲裁:

优点: 在集中式仲裁中, 存在一个仲裁主模块 (Arbitration Master), 它负责协调所有设备对总线的访问请求。这种机制简单且易于实现, 适用于较小规模的系统。

缺点: 集中式仲裁可能引起单点故障, 如果仲裁主模块发生故障, 整个系统的总线访问可能会中断。此外, 当系统规模扩大时, 集中式仲裁的效率会下降。

2. 分布式仲裁:

优点：在分布式仲裁中，每个设备都具有一定的决策能力，可以根据预定的优先级规则来决定是否获得总线访问权限。这种机制可以提高系统的并行性和吞吐量，并减少对中央仲裁模块的依赖。

缺点：分布式仲裁增加了系统的复杂性，需要设备之间进行协作和通信。此外，如果没有明确的冲突解决策略或优先级规则，可能导致死锁或优先级反转等问题。

3. 旁路仲裁：

优点：旁路仲裁是一种快速响应的仲裁机制，可以在总线上的数据传输过程中进行仲裁，从而减少延迟。旁路仲裁通常使用专用的仲裁线（Arbitration Line）或信号线，设备可以通过该线发送请求和获得总线控制权。

缺点：旁路仲裁需要额外的仲裁线或信号线，并且可能增加总线的复杂性和成本。此外，如果仲裁线路出现故障，可能会导致仲裁机制失效。

总线仲裁机制的选择取决于系统的规模、性能需求和可靠性要求。集中式仲裁适用于较小规模的系统，并且易于实现。分布式仲裁适用于大规模系统，可以提高并行性和吞吐量，但需要设备之间的协作。旁路仲裁适用于对响应时间要求较高的系统，可以减少延迟，但需要额外的仲裁线路。

2. 简要描述 AMBA 总线中 APB、AHB、AXI、ACE、CHI 等总线协议的特点和使用场景

1. APB (Advanced Peripheral Bus) :

- 特点：APB 是 AMBA 总线中最简单、最低性能的总线协议。它使用同步传输模式，具有单一主设备和多个从设备的结构。传输速率较低，适用于低带宽、低功耗、较简单的外设控制器。
- 使用场景：APB 常用于连接低速外设，如 GPIO（通用输入/输出）、定时器、UART（通用异步收发器）等。

2. AHB (Advanced High-performance Bus) :

- 特点：AHB 是 AMBA 总线中的中等性能总线协议。它使用同步传输模式，支持多个主设备和多个从设备的结构。具有较高的传输速率和较低的延迟，支持高带宽数据传输和多设备并发访问。
- 使用场景：AHB 常用于连接具有较高性能需求的外设，如 DMA 控制器、存储器控制器、高速接口控制器等。

3. AXI (Advanced eXtensible Interface) :

- 特点：AXI 是 AMBA 总线中高性能、高度可扩展的总线协议。它使用同步传输模式，支持多个主设备和多个从设备的结构。具有高带宽、低延迟和高并发性能，支持流水线和乱序传输。
- 使用场景：AXI 广泛应用于高性能和复杂的系统-on-chip (SoC) 设计中，包括高性能处理器、图形处理器、网络处理器、高速存储器等。

4. ACE (AXI Coherency Extensions) :

- 特点：ACE 是在 AXI 基础上增加了一致性扩展的 AMBA 总线协议。它提供了缓存一致性和高效的数据共享机制，支持多个处理器和加速器之间的一致性管理。

- 使用场景：ACE 常用于高性能多核处理器系统，其中需要实现多个处理器和加速器之间的高效数据共享和一致性维护。
5. CHI (Coherent Hub Interface) :
- 特点：CHI 是 AMBA 总线中的最高性能、最复杂的总线协议。它是一种高度可扩展和灵活的协议，支持高度一致性、低延迟的数据交换。具有乱序传输、错误检测和纠正、多级缓存一致性等高级特性。
 - 使用场景：CHI 适用于高端服务器、网络交换机、高性能计算等要求极高性能和一致性的系统。

3. (1) AXI 总线包含以下独立的事务通道：

1. Read Address Channel (读地址通道)：负责发起读取数据的事务，包括指定读取的地址和读取长度等信息。
2. Read Data Channel (读数据通道)：用于传输从存储器或外设读取的数据。
3. Write Address Channel (写地址通道)：负责发起写入数据的事务，包括指定写入的地址和写入长度等信息。
4. Write Data Channel (写数据通道)：用于传输要写入到存储器或外设的数据。
5. Write Response Channel (写响应通道)：用于传输写事务的响应信息，包括写操作是否成功的确认。

为什么协议不设置独立的读响应通道？出于以下几个考虑：

1. 同步性和一致性：AXI 总线协议采用同步传输模式，其中读数据通道和读响应是通过同一个时钟进行传输。这样可以保持读数据和读响应之间的同步性，确保读响应正确地对应于读数据的返回。
2. 数据关联性：读数据和读响应之间是具有关联性的，读响应需要与对应的读请求数据一一对应。将读数据和读响应合并在同一个通道中，可以确保它们之间的关联关系正确地保持。
3. 带宽效率：合并读数据和读响应在同一个通道中可以提高带宽的利用效率。由于读操作通常需要大量的数据传输，通过共享一个通道，可以避免资源的浪费，并提高总线的吞吐量。
4. 简化设计和控制：将读数据和读响应合并在同一个通道中可以简化总线的设计和控制逻辑。减少通道数量和复杂性，有助于简化总线控制器的设计，并降低系统的成本和复杂性。

总的来说，将读数据和读响应合并在同一个通道中有助于保持同步性、关联性和提高带宽效率，同时简化设计和控制。这样的设计选择在实践中已经被广泛采用，并在高性能系统中证明了其有效性。

(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：

1. 读传输事务：

- 读地址通道（Read Address Channel）和读响应通道（Read Data Channel）之间存在依赖关系。读地址通道发送读取请求，而读响应通道返回读取的数据。读响应通道的响应必须与相应的读地址通道的请求对应，保证数据的正确性和一致性。
2. 写传输事务：
- 写地址通道（Write Address Channel）和写数据通道（Write Data Channel）之间存在依赖关系。写地址通道发送写入请求和地址，而写数据通道发送要写入的数据。这两个通道之间的时序必须匹配，确保写入的数据与相应的地址对应，以避免写入错误的位置。

这样的依赖关系和握手信号的时序约束是为了确保数据的正确传输和操作的准确性。通过符合这些约束，可以保持读/写操作的一致性，避免数据错位、冲突或丢失。同时，这些约束也有助于系统的稳定性和可靠性，保证总线协议的正确执行。

总的来说，通道的握手信号时序约束是为了保证数据的正确传输和操作的准确性。它们确保了读/写操作之间的依赖关系和一致性，以提供可靠的总线通信。

（3）什么是 AXI 的突发传输？有哪些突发传输类型？

AXI 的突发传输是一种高效的数据传输方式，它允许在单个地址请求下进行连续的数据传输，减少了传输延迟和总线开销。突发传输允许一次地址传输生成多个数据传输，提高了总线的吞吐量和效率。

在 AXI 总线中，有以下几种突发传输类型：

1. INCR (Incremental)：在突发传输中，每个地址请求的地址会按照递增的方式进行，每次传输的地址是前一次传输地址加上数据宽度。例如，如果前一个传输的地址是 0x1000，数据宽度为 4 字节，则下一个传输的地址将是 0x1004，再下一个传输的地址是 0x1008，以此类推。
2. WRAP (Wrap)：在突发传输中，每个地址请求的地址会在事先指定的范围内循环。当到达指定范围的最大地址时，下一个传输的地址将回到范围的起始地址。这种方式可以在有限的地址范围内循环传输数据。
3. FIXED (Fixed)：在突发传输中，每个地址请求的地址保持不变，即每次传输的地址都是固定的，不会发生变化。这种方式适用于需要重复传输相同数据的情况。

通过使用这些突发传输类型，AXI 总线可以实现高效的数据传输，减少总线的负载和开销。具体使用哪种突发传输类型取决于应用的需求和数据传输的模式。

第5章

2. (1) 读写速率: $(1+7+1+1) \times 960 = 9600 \text{ Bit/s}$

$$(2) 7 \times 960 = 6720 \text{ bit/s}$$

4. (1) RAID 0 只是单纯组合 $MTTF = \frac{N}{4}$

(2) 可以采用 RAID 6, 此时实际上只用 2 块磁盘作数据存储即可满足系统有冗余空间的需求, 于是采用 RAID 6, 另外两块磁盘用作热备份, 此时即使两块用作数据存储的磁盘同时发生故障, 数据仍可恢复, 以达到最大的 MTTF

5. 寻道时间: 磁头臂移动到正确位置并消除抖动所需时间 影响因素: 磁头臂需要移动的距离以及移动的速度 旋转时间: 在定位到正确的磁道后, 盘片通过旋转使正确的扇区被转到磁头下方的时间 影响因素: 旋转速度; 目标扇区与当前扇区的相对位置

数据传输时间: 扇区内容被读取所需要的时间 影响: 盘片旋转速度; 数据块大小

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

(2) $5400 \text{ r/min} \rightarrow 90 \text{ r/s} \rightarrow 1 \text{ r} \frac{1}{90} \text{ s} \rightarrow 6 \text{ 盘面 每次读取 } \frac{1}{6} \text{ s} = \frac{1}{540} \text{ s 读取 } 12 \text{ KB}$

RJ/读取速率: $12 \text{ KB} \div \frac{1}{540} \text{ s} = 6.33 \text{ MB/s}$

(3) 平均旋转时间: $1 \text{ r} \cdot \frac{1}{90} \text{ s} \times 5 \text{ 扇区 } \frac{1}{90} \text{ s} \times \frac{5}{6} \quad \frac{1}{90} \text{ s} \times \frac{4}{6} \quad \frac{1}{90} \text{ s} \times \frac{3}{6} \quad \frac{1}{90} \text{ s} \times \frac{2}{6} \quad \frac{1}{90} \text{ s} \times 0$



$$\text{RJ} E(T_{\text{旋转}}) = \frac{1}{6} \times \frac{1}{90} \text{ s} \times \frac{1}{6} \times (1+2+3+4+5)$$
$$= \frac{1}{90} \text{ s} \times \frac{15}{36} = \frac{1}{216} \text{ s}$$

9. I/O 请求减少 $\rightarrow \lambda$ 变小, M 不变 假设原本性能下 $M=M_1$, 现在性能提升后 $M=M_2$,
 $M_2 > M_1$, 则 $W_1 = \frac{1}{M_1 - \lambda}$ $W_2 = \frac{1}{M_2 - \lambda}$ W_2 自然比 W_1 小 可以用平均响应时间比例系数来衡量性能, 则性能提升: $\frac{W_2 - W_1}{W_1} = \frac{M_2 - \lambda - M_1 + \lambda}{M_1 - \lambda} = \frac{M_2 - M_1}{M_2 - \lambda} = A$

M_1, M_2 均为定值, 当 λ 变小, $M_2 - \lambda$ 变大, 分母变大, 提升的比例 A 会变小, 性能提升后反而降低

10. 会争抢内存带宽资源。① 高速缓存层可以减少对主内存的访问需求, 从而减少对内存带宽资源的争夺 ② 主存-总线缓冲器确保 DMA 设备与处理器之间对共享数据的一致性, 当 DMA 设备更新内存中数据时, 主存一致性机制可使处理器缓存中相应数据失效, 减少数据错误和冲突。