

第5章

且同步的

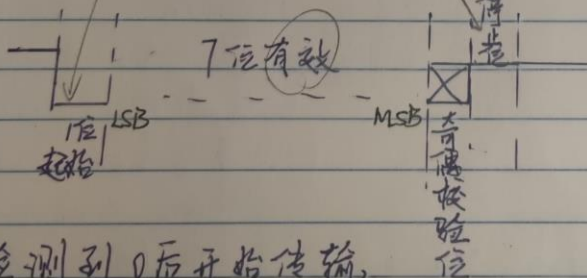
1. 并行通信带宽高延迟低，占用口线多，串扰严重
串行在远距离传输中未适用，~~但速率较慢~~
传输频率较高
接口速率差异源于串、并行接口通路数量的差异

2. UART: 通用异步收发器，串行，全双工

单工: 发送 → 接收

半双工: 发送 → 接收
接收 ← 不能同时 发送

全双工: 发送 → 接收
接收 ← 可同时 发送



1. 检测到0后开始传输，
7位 + 1位后恢复1，结束传输

空闲时为1

2. 波特率表示 - 一秒内传输的码元数
Band (Bps) 可以多bit
bps 是比特率

且是同步的

串扰严重

差异

由于 UART 是串行通信 ... 码元即 1 bit

1). 波特率 = $960 \times 10 = 9600 \text{ bps}$

2). $960 \times 7 = 6720 \text{ bps}$

3. I2C 串行数据总线 (同步)
串行时钟总线

在 CL 线低电平时, DL 线准备数据; CL 线高电平时,
DL 线为值即要传输的数据

启动: SCL 线高电平时, SDA 高 \rightarrow 低

终止: SCL 线低电平时, SDA 低 \rightarrow 高

4种

空闲: SDA, SCL 同时高, 保持 $t \geq t_{BUF}$

忙碌: 启动后, 终止前

每传 8 位有一个应答信号

数据包格式:

1位	MSB	LSB	1位	1位	1位	1位	1位	1位
启动	7位	R/W	从机	8位	从机	8位	接收者	停止
地址	从机地址	↑	应答	数据	应答	数据	应答	↑

1: 从机 \rightarrow 主机

0: 主机 \rightarrow 从机

I2C中, 所有数据在一根线SDA上传输, 同一时刻
数据的传输只能是单向的
∴ 是半双工

4/1. $MTTF = \frac{N}{P}$

2). 用80G存储, 再用80G将存储的数据复制,
剩余的40G按每位做异或的方法存储运算结果

I. 寻道时间指磁头沿盘面半径从里向外移动, 在多个
同心圆磁道中找到所需磁道的时间
(受移动距离和速度影响)
旋转时间指盘片通过旋转使正确扇区移动到
磁头下方的时间
(受旋转速度和距离影响)
数据传输时间指读写扇区内数据的时间
(受旋转速度影响)

刻

$$1) \quad 6 \times 240 \times 12 \text{KB} = 17280 \text{KB} = 16.875 \text{MB}$$

$$2) \quad 6 \times 12 \text{KB} \times 90 \text{r/s} \quad 26 \text{个盘面同时读取}$$

$$\approx 6.33 \text{MB/s}$$

结果

3) 磁盘的平~~均~~旋转距离从 0 ~ 周长不等
平均距离取 $\frac{\text{周长}}{2}$

$$\therefore \bar{T} = \frac{1}{2} \times \frac{1}{90} = \frac{1}{180} \text{s}$$

个

7. 通过决定请求最优执行次序达到最短磁头移动距离, 减小寻道时间; 并减小磁盘旋转距离以减少旋转时间

8. 原有方式需要在每次写入时读取所有的磁盘, 优化后只要读取一个磁盘, 可以显著提升速度 (?? 读取速度)

9. 从公式上看, I/O 请求 \downarrow 即 $\lambda \downarrow$

$$\frac{dW}{d\lambda} = \frac{1}{(\mu - \lambda)^2}, \quad (\mu > \lambda) \quad \therefore \text{随 } \lambda \downarrow W \text{ 的增长率下降}$$

从实际过程看: 由于请求数量减少, 等待时间减少, W 趋于 0。由于 W 不可能小于 0, 随着逼近极限, W 的变化程度变小。

10. 在突发模式下会
周期窃取模式下可能会
透明模式下不会

在合理的多级存储设计下, CPU 有大几率在 Cache 中找到目标数据, 因此可以降低 DMA 与 CPU 争抢内存带宽的概率。

第6章

1. 轮询: 每个主设备优先级相同, 依次获得使用权。在各设备对总线的访问需求相近时性能较好。

优先级: 优先级更高的设备先获得访问权。如果有设备访问频次高于其它设备, 这种方式性能更好。

2. AHB: 功耗低, 易使用, 主要应用于低速率低速外设
AHB: 高速, 可扩展, 应用于高性能系统模块连接
AXI: 更高带宽, 低延时, 接口庞大, 机制复杂, 应用于各种高性能 SoC
ACE: 最大程度利用缓存, 兼顾高性能与低功耗, 应用于 Cache 间的通信
CHI: (?) 用于连接 Cache 与 memory, 可扩展性强

3/1). 读地址通道, 写地址通道, 读数据通道, 写数据通道, 写响应通道。
没有读响应是因为读响应作为读数据的一部分从读数据通道传递

2). 握手在 Ready 和 Valid 都为高时下一个时钟上升沿完成

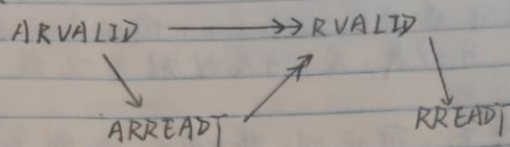
每个 AXI 组件都使用 ACLK 时钟信号

所有输入信号都在 ACLK 上升沿采样

所有输出信号的变化必须在 ACLK 上升沿之后

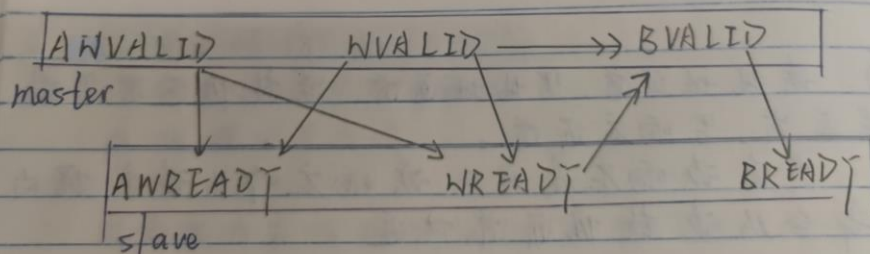
1 → 表示信号的先后关系可相反
 → 表示信号的先后关系不能相反

读:



满足上述关系的原因是在读数据时需要
 先传地址再传数据

写: (不运用上述标记)



① 左侧 4 个信号中, master 不能等 slave 发送后再发送, 否则会出现相互等待的僵局

② slave 可以 master 发送完后再发送, 也可以不等

③ 响应信号要在 AN 和 N 信号之后,
 具体则是要等 NLAST 高后 BVALID 才能拉高

③ 突发传输，就是一次事务中，连续传输多个地址相邻的数据

(由 ARBURST 和 ABURST 选择)

④ 固定式突发读写: (b00) 每次传输地址固定

⑤ 递增式: 每次传输地址是上一次增加固定值
(b01)

⑥ 包装式: 包装到包边界的一个低地址?
(b10)