|  |
| --- |
| 基于AHB协议的单口SRAM写后读分析 |
|  |
|  |
| 作者：HateHanzo |
| 联系方式：HateHanzo@163.com |

版权所有 侵权必究

**修改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改文件 | 描述 | 作者 | 时间 |
| v1.0 |  | 初稿 | HateHanzo | 20210903 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**缩略语与约定符号**

**缩略语**

|  |  |  |
| --- | --- | --- |
| **英文简写** | **英文全称** | **中文描述** |
| AHB | Advanced High-performance Bus | 先进高性能总线 |
| SRAM | Static Random-Access Memory | 静态随机存取存储器 |
| FSM | Finite State Machine | 有限状态机 |
| SR-Latch | Set-Reset-Latch | SR锁存器 |
| ICG | Integrated Clock Gating Cell | 门控时钟 |
| 2FSK | Two-Frequency-shift keying | 二进制频移键控 |

**约定符号**

1、对于某信号名A，若无特殊声明，A\_d1表示该信号延时一拍，A\_d2表示延时两拍，以此类推。如ahb\_addr\_d1表示信号ahb\_addr延时一拍。

**目 录**

[**修改记录** I](#_Toc82007183)

[**缩略语与约定符号** II](#_Toc82007184)

[**目 录** 0](#_Toc82007185)

[**1 前言** 1](#_Toc82007186)

[**2 AHB总线概述** 1](#_Toc82007187)

[**3 单Master(CPU为例)AHB总线架构** 1](#_Toc82007188)

[**3.1 AHB接口信号** 1](#_Toc82007189)

[**3.2 AHB基本传输协议** 2](#_Toc82007190)

[**3.3 AHB顶层设计** 3](#_Toc82007191)

[**4 AHB协议细节知识点补充** 3](#_Toc82007192)

[**4.1 回环传输(WRAP)有什么用？如何计算？** 3](#_Toc82007193)

[**4.2 为什么要设置burst传输，跟single有什么区别？** 4](#_Toc82007194)

[**4.3 HRESP[1:0]的知识点总结** 4](#_Toc82007195)

[**参考文献** 6](#_Toc82007196)

[**附 录** 7](#_Toc82007197)

**1 前言**

本文主要对采用AHB协议对单口SRAM进行读写操作时会碰到的典型问题—写后读进行一些分析和探讨。

**2 SRAM时序与AHB时序**

本文进行分析的单口SRAM时序可以抽象为图 1所示，可以得知SRAM写操作需要将控制信号(CEN/WEN/A)和写数据(D0)对齐，而读操作在给出控制信号后，下一个时钟周期才会吐出数据(D1)，CEN是片选信号(“0”有效)，WEN为单bit读写控制信号(“0”表示写)。设SRAM的大小为512X32。



图 1 SRAM Timing

AHB协议规定，第一个时钟周期给出控制信号，第二时钟周期给出数据，所以控制信号是需要锁存的，此处用伪代码抽象出AHB的有效操作行为。

一次有效的AHB操作信号ahb\_accs = ahb\_readyi & ahb\_sel & ahb\_trans[1]

一次有效的AHB写操作信号 write\_vld = ahb\_accs & ahb\_write

一次有效的AHB读操作信号 read\_vld = ahb\_accs & ~ahb\_write

锁存控制信号：

else if(ahb\_accs)

ahb\_addr\_f1 <= ahb\_addr[10:0]

ahb\_size\_f1 <= ahb\_size[1:0]

**3 单Master(CPU为例)AHB总线架构**

**3.1 AHB接口信号**

所示的

**3.2 AHB基本传输协议**

组计数

**3.3 AHB顶层设计**

阈值的理论计算，其中指fsk模块的工作时钟

例如，，，，N=32

取整数，

**4 AHB协议细节知识点补充**

**4.1 回环传输(WRAP)有什么用？如何计算？**

用于CPU填充cacheline的时候。第一个访问的地址是0x90，加到9c之后，会从cacheline的起始地址继续传输，对cache的访问不应该跨过cacheline的边界[3]。如何计算回环到了边界呢？以图 5为例，HSIZE是Word，也就是4个Byte，HBURST是WRAP8，也就是说回环边界为4x8=32。从0x90开始传输到0x9C，再继续累加一个word的话就变成0xA0，0xA0是32的整数倍，也就是到了回环边界了，0xA0减去32等于0x80，所以0x9C的下一个地址变为0x80。

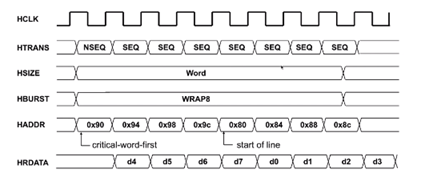


图 5 CPU填充cacheline时的回环传输

**4.2 为什么要设置burst传输，跟single有什么区别？**

为了提高传输的效率。比如CPU要拿DRAM里面的数据，DRAM可以一次读4个地址放到buffer里面，然后再一个一个发到总线上，这样就不会占用DRAM[3]。

**4.3 HRESP[1:0]的知识点总结**

HRESP是由slave产生的输出信号，提供传输状态的额外信息。HRESP共四种输出情况，okay、error、retry、split。除了okay可以持续单周期外，error、retry、split至少要持续两个周期（两个周期的响应时间使得master能够在下一个传输前将HTRANS修改为IDLE）。为了完成error、retry或split这三类responses的传输，其倒数第二个周期输出error、retry或split，同时拉低HREADY，最后一个周期仍旧输出error、retry或split，同时拉高HREADY。如果slave插入的等待周期大于等于3个，那么倒数第二个周期前HRESP均输出okay。对于split和retry response，下一个传输必须被取消，而对于error，下一个传输是否取消是可选的。（AMBA2.0\_SPEC 3.9.3）

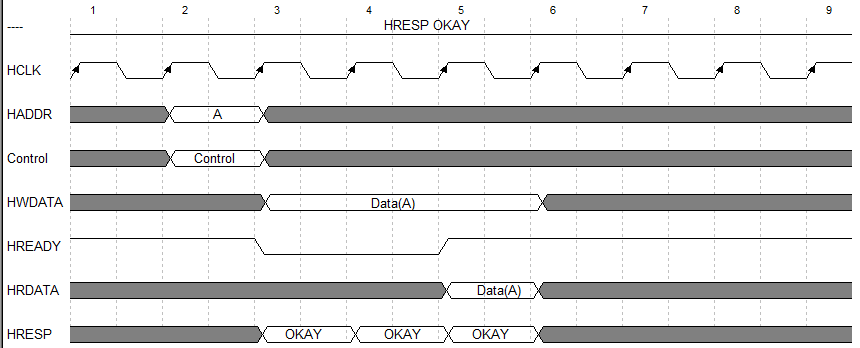


图 6 带等待周期的传输HRESP回复OKAY

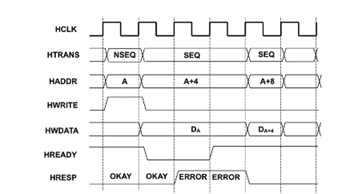
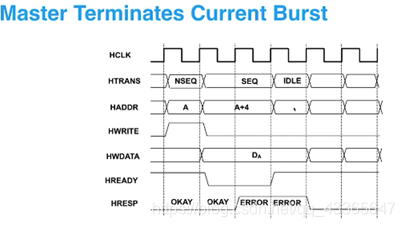
 

图 7 HRESP为ERROR时继续或终止传输[3]

图 7左图继续了A+4地址的传输，HTRANS为SEQ，而右图则舍弃了A+4地址的传输，HTRANS为IDLE，具体情况取决于master的设计。图 8中第二个RETRY周期master将HTRANS修改为IDLE。

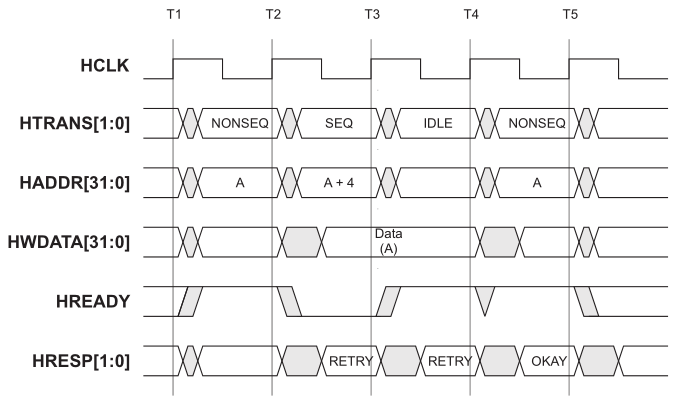


图 8 transfer with retry response

**参考文献**

1. AMBA2.0\_SPEC
2. AMBA3.0 AHB\_Lite\_SPEC

**附 录**