1 类 SRAM 接口说明

大赛要求 myCPU 封装为 AXI 接口，具体实现有以下两种方法：

(1) 自定义内部取指、访存与 AXI 接口的交互信号，然后封装为 AXI 接口。

(2) 自定义内部取指、访存与其他接口的交互辛哈，再使用转换桥转换为 AXI 接口。比如内部实现为类 SRAM 接口，使用官方提供的类 SRAM 转 AXI 的桥来转接。

我们推荐方法一，这样可以自己对接口进行设计，以提升访存性能。大赛提供的类 SRAM 转 AXI 的桥代码参考本目录下的 ***cpu_axi_interface.v***，效率偏低，且不支持 burst 传输。

在 SoC_SRAM_Lite 中，myCPU 接口是 SRAM 接口，数据访问都是单周期返回的。接口简单，却也限制了 myCPU 的频率。myCPU 频率不能超过 RAM 的读写频率。

实际 CPU 频率是普遍高与存储器读写频率的，所以很多时候访问都是需要多个 CPU 周期才能返回的。为此为 SRAM 接口增加地址传输握手信号 addr_ok 和数据传输握手信号 data_ok，这样就可以实现任意周期返回数据了，称为类 SRAM 接口。下表展示了类 SRAM 的接口信号，与 2017 年大赛提供的类 SRAM 接口定义稍有不同。

表 1-1 类 SRAM 接口信号

信号	位宽	方向	功能
clk	1	input	时钟
req	1	master—>slave	请求信号，为 1 时有读写请求，为 0 时无读写请求
wr	1	master—>slave	该次请求是写
size	[1:0]	master—>slave	该次请求传输的字节数，0: 1byte；1: 2bytes；2: 4bytes。
addr	[31:0]	master—>slave	该次请求的地址
wdata	[31:0]	master—>slave	该次请求的写数据
addr_ok	1	slave—>master	该次请求的地址传输 OK，读：地址被接收；写：地址和数据被接收
data_ok	1	slave—>master	该次请求的数据传输 OK，读：数据返回；写：数据写入完成。
rdata	[31:0]	slave—>master	该次请求返回的读数据。

需要注意，不同于 SRAM 接口，类 SRAM 的地址信号(addr)是字节寻址的，其指向读写数据的最低有效位。因而 addr 和 size 信号需配合使用，不支持 3 字节读写，有且只有以下类型组合。

1. addr[1:0]=2’b00 时，可能的组合：
   size=2’b00, size=2’b01, size=4’b10,
2. addr[1:0]=2’b01 时，可能的组合：
   size=2’b00
3. addr[1:0]=2’b10 时，可能的组合：
   size=2’b00, size=2’b01
4. addr[1:0]=2’b11 时，可能的组合：
   size=2’b00

wdata 和 rdata 有效数据字节也与 size 与 addr[1:0]信号对应，小尾端下，配合如下：

表 1-2 类 SRAM 接口数据有效情况

	data[31:24]	data[23:16]	data[15:8]	data[7:0]
size=2'b00,addr=2'b00	-	-	-	valid
size=2'b00,addr=2'b01	-	-	valid	-
size=2'b00,addr=2'b10	-	valid	-	-
size=2'b00,addr=2'b11	valid	-	-	-
size=2'b01,addr=2'b00	-	-	valid	valid
size=2'b01,addr=2'b10	valid	valid	-	-
size=2'b10,addr=2'b00	valid	valid	valid	valid

其读一个半字的时序逻辑如下：

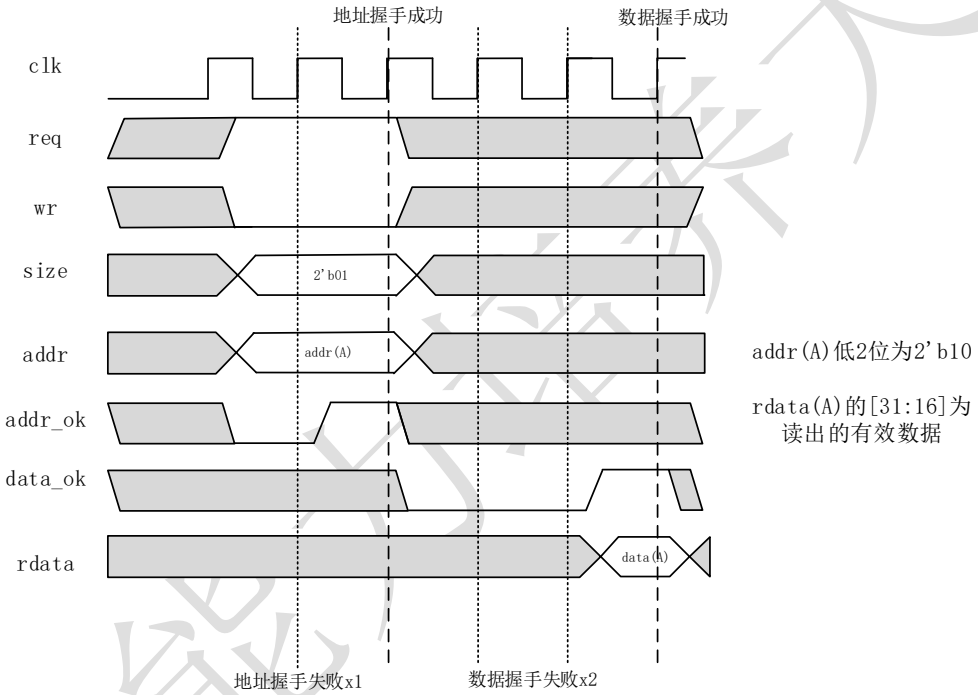

图 1-1 类 SRAM 接口一次读时序

其写一个字节的时序逻辑如下：

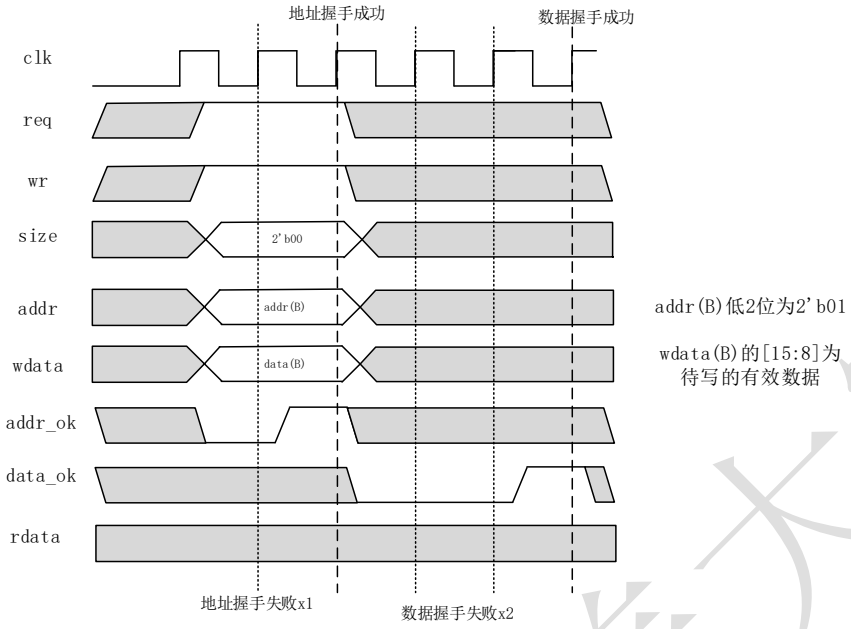

图 1-2 类 SRAM 接口一次写时序

其连续写读的的时序逻辑如下。可以看到连续写读时，slave 返回的 data_ok 应该顺序返回的，就是先写后读，必须先返回写的 data_ok，再返回读的 data_ok。另外，在一次传输 addr_ok 握手后 data_ok 握手前，是有可能出现好几次其他请求的 addr_ok 握手的，也就是可能出现握手序列 addr_ok->addr_ok->addr_ok->addr_ok->…->data_ok，master 端避免这一情况的出现可以通过拉低 req 信号，slave 避免这一情况的出现可以通过拉低 addr_ok 来避免。

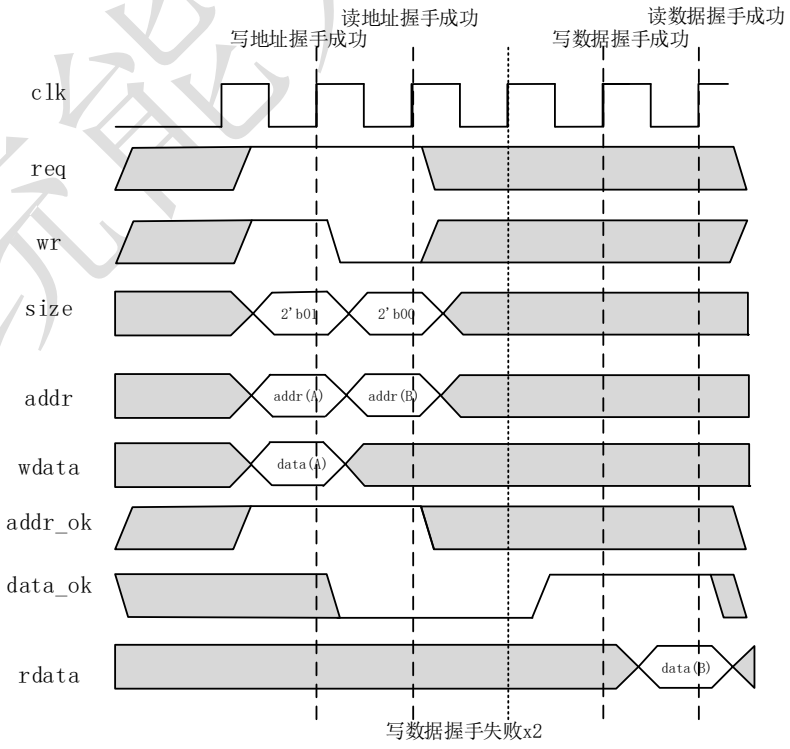

图 1-3 类 SRAM 接口连续写读时序

其连续读写的时序逻辑如下。从下图可以看到，当 addr_ok 和 data_ok 同时有效时，是针对不同请求的握手：addr_ok 是当前传输的地址握手成功，data_ok 是之前传输的数据握手成功。另外，图中读数据握手成功是有可能在写地址握手成功前完成的。

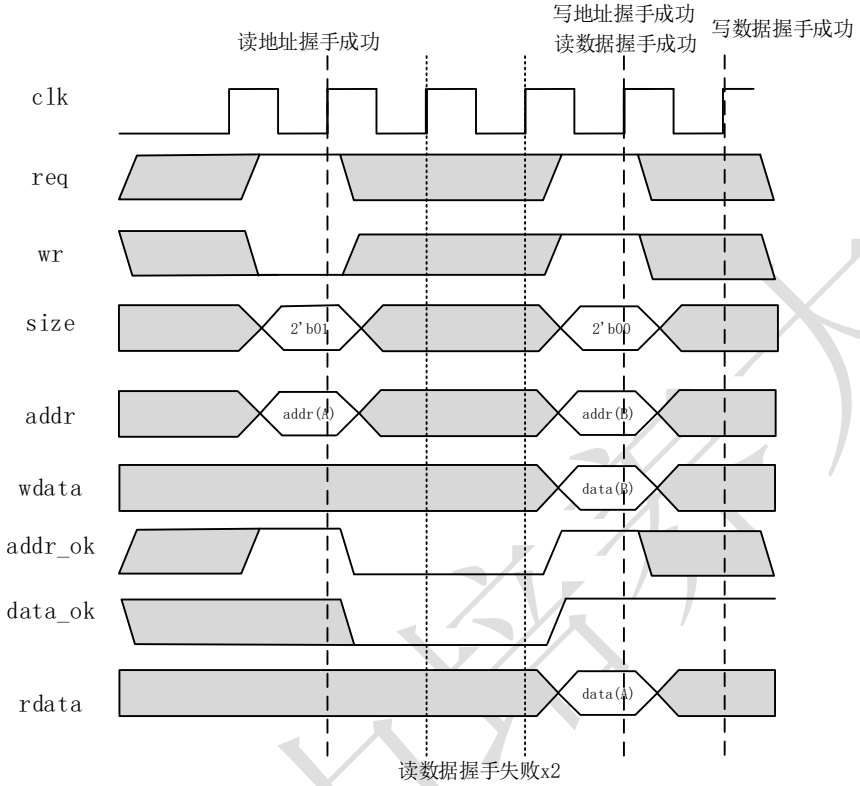

图 1-4 类 SRAM 接口连续读写时序