

实验 5-1 报告

学号：2016K8009929011

姓名：段江飞

一、实验任务（10%）

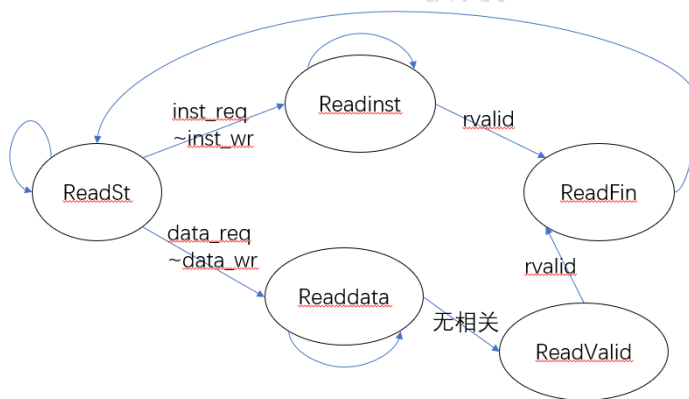
本实验要求添加类 SRAM 接口到 AXI 总线接口的转换桥支持，实现模拟 CPU 通过该桥访问 AXI 接口的 RAM，并通过用该桥的读写测试。

二、实验设计（30%）

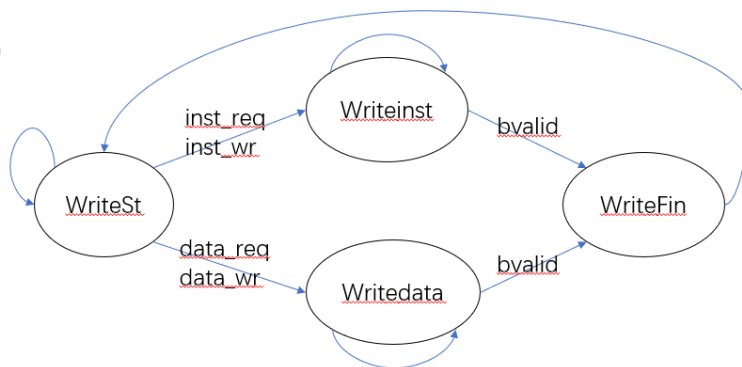
（一）硬件实现

1、状态机实现

桥的实现利用状态机，状态转移如图一、图二所示，为了实现同时读写，设立两个状态机，分别是读状态机和写状态机。



图一



图二

2、读状态机

最初在起始状态，当类 SRAM 端读请求信号发出时，进入读指令或者读数据状态，设置相应的 ar 和 r 通道接

口信号，如果是读指令，那在读指令状态发出读请求，接下来就等待 rvalid 信号，rvalid 信号有效时，表示读已经完成，进入 ReadFin 状态，将指令返回，然后回到起始状态；如果是读数据，需要判断写相关，如果当前正在进行写请求，而且写地址和读地址相同，需要在读数据状态等待，知道写完成进入读有效状态，发出读数据的请求，然后等待读数据返回。

在发生写读相关时，将数据地址存起来进行比较，同时，在写结束时，有可能出现读同时结束，这时候要向类 SRAM 接口连续发送两个数据 ok 的信号。

3、写状态机

最初在起始状态，当请求信号发出时，进入写指令或者写数据状态（虽然不会有写指令，还是写了这样一个状态），设置相应的 aw, w 和 b 通道的接口信号，然后等待 bvalid 信号，bvalid 信号有效时，表示写请求已经完成，进入 WriteFin 状态，然后回到起始状态。

（三）验证

1、仿真验证

在 vivado 上仿真，测试 10 个功能点，通过测试。

2、上板验证

数码管从全 1 加到全 a，各位同步累加。调整随机种子，数码管变化均正确。

三、实验过程（60%）

（一）实验流水账

1、12 月 2 日下午 3 点到 17 日凌晨 3 点

理解实验，写代码，找 bug，写实验报告。

2、12 月 3 日上午 11 点到下午 5 点

找环的 bug，上板验证，写实验报告。

3、12 月 3 日晚上 8 点到 12 点

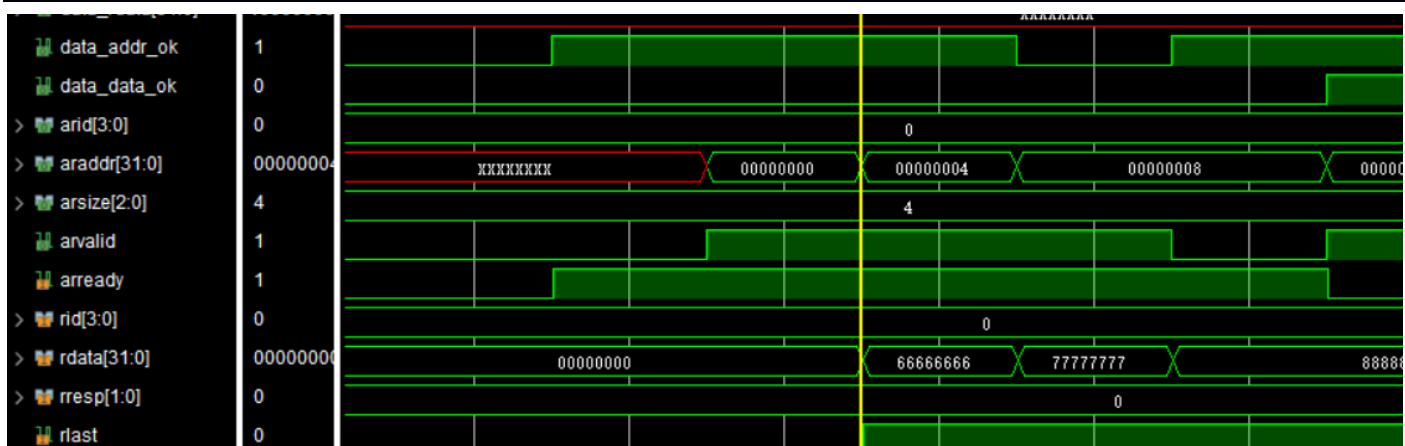
上板错误 debug，写实验报告。

（二）错误记录

1、错误 1

（1）错误现象

刚写完数据通路，进行仿真的时候，波形非常混乱，测试错了一半，还不会停止。



图三

(2) 分析定位过程

最初我是直接写的，没有用状态机，然后错误很多，仿真看波形图发现同时读写，连续读写，然后信号处理出现了问题，波形实在是太乱了，我一直看啊看不知道怎么改，询问了同学和查阅 AXI 总线的资料之后发现有用状态机写的，思考之后发现状态机很好实现，于是我改成用状态机写，写好之后测试就通过了 9 个点，还剩下一个写读相关的问题没有解决。

2、错误 2

(1) 错误现象

```
[ 2055 ns] OK!!!read inst 0
[ 2095 ns] OK!!!read inst 1
[ 2135 ns] OK!!!read inst 2
[ 2175 ns] OK!!!read inst 3
[ 2215 ns] OK!!!read inst 4
[ 2255 ns] OK!!!read data 0
[ 2305 ns] OK!!!read data 1
[ 2345 ns] OK!!!read data 2
[ 2385 ns] OK!!!read data 3
[ 2475 ns] Fail!!!read from data sram-like, ref_data[31: 0]=32'h55004444, my_data[31: 0]=00004444

Test end!
Fail!!!Total          1 errors!
```

图四

(2) 分析定位过程

根据最后一个错误，首先我去看了一下这个测试数据都是什么，在虚拟 CPU 里，我发现最后一对测试的是先写后读的情况，然后我去波形图抓取了对应的信号，发现错误。

(3) 错误原因

读请求读取的地址和之前的写请求一样，但是写请求还未完成，读请求已经发送并且完成，读取出来的是旧值。在最初我直接添加判断，根据地址和 bready 信号判断是否存在写读同一个地址的情况，但是出现了波形停止，然后我综合布局布线，也没有报出来 loop，我能确定添了一个判断条件导致的，于是仔细看这个条件，很偶然发

现了环是在哪里。

我的 data_addr_ok 是根据状态组合赋值，状态是用 data_addr 做条件来选择，在虚拟 CPU 里，data_addr 是根据 data_addr_ok 信号组合选择进行组合赋值，这里出现了 loop。

```
assign data_addr_ok = r_curstate == ReadSt && r_nxtstate == Readdata || w_curstate ==  
WriteSt && w_nxtstate == Writedata;  
  
always@(*)  
begin  
    case(r_curstate)  
        ReadSt:  
            begin  
                if(inst_req && ~inst_wr)  
                    r_nxtstate = Readinst;  
                else if(bready && awaddr_t[31:2] == data_addr[31:2]) //出现环的判断条件  
                    r_nxtstate = r_curstate;  
                else if(data_req && ~data_wr)  
                    r_nxtstate = Readdata;  
                else  
                    r_nxtstate = r_curstate;  
            end  
    end
```

后来我对原本的状态机进行修改，在读数据之后新添加一个读有效状态，用于解决相关，同时避免出现环。

(4) 修正效果

测试报了 9 个对的，但是不结束。

3、错误 3

(1) 错误现象

控制台输出 9 个读对的现象，然后不报错，一直执行下去。

(2) 分析定位过程

抓取 data_data_ok 信号，发现总共只返回了 9 次 ok，但是连读带写应该返回 10 次，然后我将最后一次读写的信号抓出来看，发现了问题。

(3) 错误原因

最后一次是读和写出现了相关，但是两个是同时结束了，而我只返回来一个 data_data_ok 的信号，少返回了一次，做一个单独的判断，如果读写同时结束，将 valid 信号存一拍，然后连续返回两个数据 ok 信号。

(4) 修正效果

通过了仿真测试。

4、错误 4

(1) 错误现象

原以为上板过了，没想到调了几组随机种子之后，数码管累加出现了错误。数字最开始是一个奇怪的值。

(2) 分析定位过程

根据讲义中调试方法，将 tb 设置为上板环境，然后修改随机种子，进行仿真，根据报错信息，读数据错误，

抓取相应的信号，发现信号错位了。

（3）错误原因

没有考虑到先后的时序的问题，当先后进行一次数据的写和读时，只考虑了地址相关的时序问题，没有考虑即使地址不相关，也要保持读写顺序，因为类 SRAM 接口不想 AXI 接口那样，有独立的读写通道，对应数据来说，必须保证返回的 data_ok 顺序也一致。

（4）修正效果

通过了测试。

四、实验总结

又写出了组合环，组合环找出来全靠运气。