

实验 5-1 报告

2016K8009909006

刘杰

一、实验任务（10%）

为 CPU 添加总线支持，本次实验仅实现带握手信号的从类 SRAM 接口到 AXI 接口的转接桥，将其接上提供的模拟 CPU 和 AXI 接口的 RAM，并通过含 5 个读指令测试，5 个读数据测试和 5 个写数据测试。

二、实验设计（30%）

AXI 接口的 SRAM 接受地址后，返回数据的延迟是随机的，所以需要搭建一个状态机来描述转接桥的状态，实现转接。状态机共 5 个状态：INIT，WAIT_RADDR，WAIT_RDATA，WAIT_WADDR，WAIT_WDATA。

INIT 代表 AXI 读写通道空闲，没有读写事务正在被处理。在 INIT 态，如果 data_req 为高且 data_wr 为低，将或者 inst_req 为高且 inst_wr 为低，代表有读请求，根据请求来源，将 arid 置成相应值，arid 为高代表读数据，为低代表读指令，将下一拍转到 WAIT_RADDR 态；如果 data_req 和 data_wr 都为高，代表有数据写请求，将记录请求来源的 awid 置成 1，下一拍转到 WAIT_WADDR 态。

WAIT_RADDR 代表类 SRAM 接口等待 AXI 读通道接受发送的读地址。在 WAIT_RADDR 态，如果 arready 和 arvalid 同为高，握手成功 AXI 接口已经接受读地址，下一拍转到 WAIT_RDATA 态。

WAIT_WADDR 代表类 SRAM 接口等待 AXI 写通道接受发送的写地址。在 WAIT_WADDR 态，如果 awready 和 awvalid 同为高，握手成功 AXI 接口已经接受读地址，用寄存器暂存写数据 data_wdata 和由写地址，写字节大小产生的写字节选通 data_wstrb 信号，下一拍转到 WAIT_WDATA 态的同时将寄存器的值传给 AXI 写通道的相应端口。

WAIT_RDATA 态代表 SRAM 接口等待 AXI 读通道返回数据或者指令。在 WAIT_RDATA 态，如果 rvalid 和 rready 同为高，握手成功 AXI 接口已经返回数据或者指令，下一拍转到 INIT 态。

WAIT_WDATA 态代表 SRAM 接口等待 AXI 写通道写数据完成。在 WAIT_WDATA 态，如果 bvalid 和 bready 同为高，握手成功 AXI 接口已经完成写数据，下一拍转到 INIT 态。

根据状态机的设计，可以看出采用暴力的方法解决了读写通道分离时发生的同地址读写问题，即每次 AXI 读写通道只处理一个事务，只有在完成一个事务才会处理下一个，读写通道不能同时工作。

状态机示意图如下：

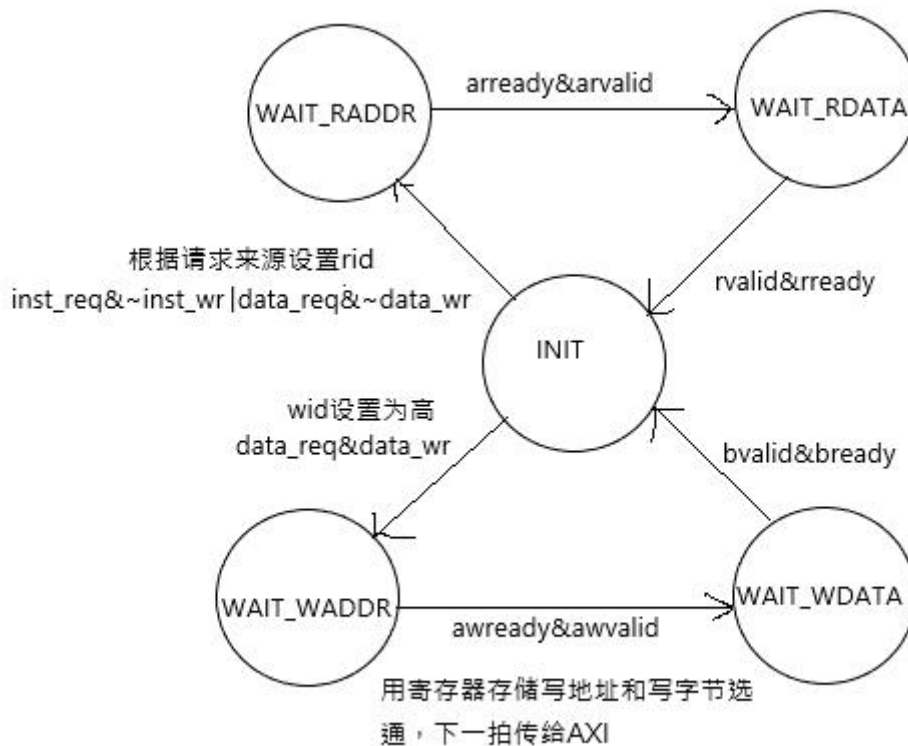


图 1：状态机示意图

三、实验过程（60%）

（一）实验流水账

2018.12.3 13:30-15:00 阅读讲义并完成转接桥 RTL 代码编写

2018.12.3 15:20-16:00 仿真测试，读指令测试通过，但读数据测试失败，debug 找出错误

2018.12.3 16:00-17:30 改完 bug 仿真测试，仿真通过，上板多次尝试发现数码管一直为 0，debug 找出错误

2018.12.3 19:00-19:30 改完 bug 仿真测试通过，上板通过

（二）错误记录

1、错误 1

（1）错误现象

读数据时 AXI 返回的数据有误，5 个测试点都为 0。

（2）分析定位过程

读地址传入正确，读数据按照正确的时序返回，但返回的值不正确，都为 0，推测写数据时没有正确写入。查看写请求的处理过程，发现写数据和写地址在同一拍传给 AXI，但是 AXI 接口接收写数据会比写地址慢一拍。

(3) 错误原因

写入数据和写入地址同时传给 AXI，但是 AXI 接受写入数据要比接受写入地址慢一拍，所以 AXI 写入的数据不符合预期值。

(4) 修正效果

在写入地址被 AXI 接受的同时，即握手成功时，用寄存器暂存写入数据，在下一拍传给 AXI 对应的端口。

(5) 归纳总结（可选）

没有认真阅读 AXI 接口的讲义。

2、错误 2

(1) 错误现象

仿真通过，但是上板测试数码管一直为 0。

(2) 分析定位过程

检查仿真测试数码管的累加情况，发现正确。逐行检查代码，发现读请求数据通道的握手信号中后缀为 valid 才是 slave 端发来的，和读请求地址通道握手信号中后缀为 ready 为 slave 端发来的，正好相反，但是写代码时没注意，以为 slave 端发送的握手信号都是 ready 后缀。

(3) 错误原因

在给 inst_data_ok 和 data_data_ok 赋值时使用如下逻辑

```
assign inst_data_ok = ((r_id==1'd0)&rready &(state==WAIT_RDATA));
```

```
assign data_data_ok=(r_id&rready&(state==WAIT_RDATA))|(w_id&bvalid&(state==WAIT_WDATA));
```

上述赋值中因为误认为 slave 端发送的握手信号是 rready 而使用 rready，但实际上应该为 rvalid。

(4) 修正效果

将上述赋值逻辑中的 rready 修改为 rvalid 即可，再次测试上板通过。

四、实验总结（可选）

AXI 转接桥能够支持同时处理读写，但是由于采用暴力方法解决了同地址读写不保证顺序的问题，所以 AXI 接口只能在上一件读写事务完成后，才能进行新的事务处理，不能同时读写，AXI 接口的效率下降，之后嵌入 CPU 后会降低主频。然而这个星期 deadline 真的很多，所以我选择偷懒，也许之后会优化。