

实验 5-2 报告

学号：2016K8009929060

姓名：王晨赳

一、实验任务（10%）

本次实验需要将 CPU 顶层的接口修改为 AXI 接口，使得 CPU 对外接口为 AXI 接口。之后集成到 SOC_AXI_Lite 系统上，并完成行为仿真功能测试，最后完成上板验证。

二、实验设计（30%）

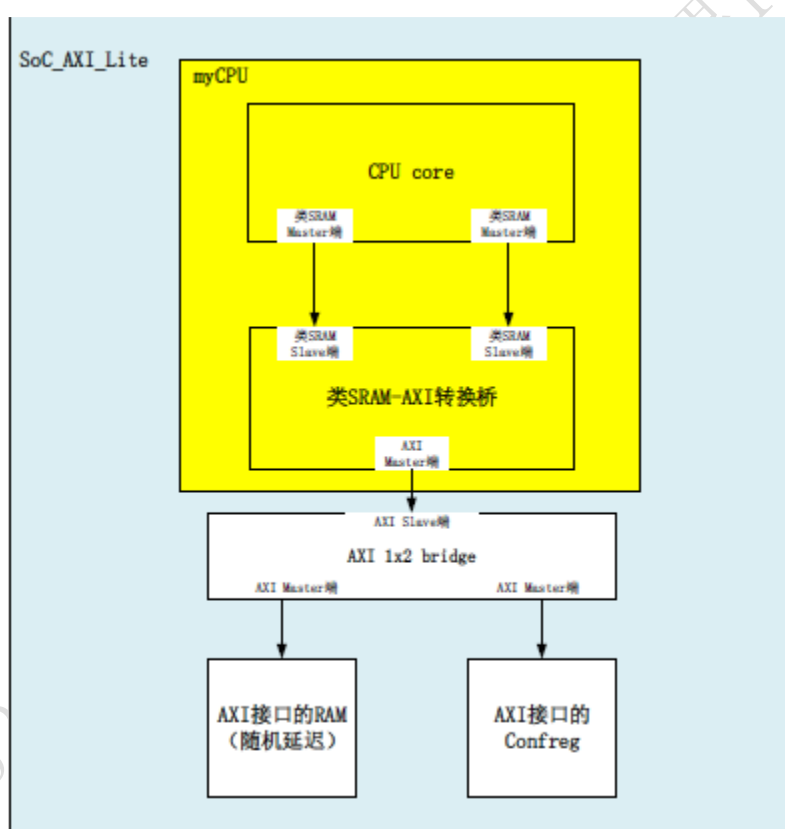


图 1: SOC_AXI_Lite 示意图

实验首先需要修改原来的 CPU 核仿存取指的接口为类 SRAM 接口，然后通过类 SRAM 接口到 AXI 接口的转换桥将 CPU 核发出的信号转换为 AXI 总线信号。于是新建了一个模块 `cpu_inner_core`，其对外的仿存取指接口为类 SRAM 接口。然后在 CPU 顶层模块 `mycpu_top` 中实例化两个模块 `cpu_inner_core`，和 `cpu_axi_interface` 模块。后者是实验 5-1 实现的转换桥模块。因为 `mycpu_top` 模块对外的接口为 AXI 接口，故在 `mycpu_top` 模块中需要将 `cpu_inner_core` 的端口信号经过 `cpu_axi_interface` 模块转换。在 `cpu_inner_core` 模块中，由于取指和仿存需要握手才能进行，所以需要修改相应的代码。对于取指，在 IF 阶段发请求，收到 `inst_addr_ok` 后进入 ID 阶段，在 ID 阶段

收到 inst_data_ok 后得到指令进行译码，然后执行。对于仿存指令，在 EX 阶段发请求，收到 data_addr_ok 后进入 MEM 阶段，在 MEM 阶段收到 data_data_ok 表示已将数据写去或取得数据，之后可进入 WB 阶段。但是根据讲义上对类 SRAM 接口的描述，类 SRAM 接口不支持 3 字节读写，故对于 swl 和 swr 指令的某些情况而言，不能直接写 3 字节的数据，需要转换为两次写操作。对于例外的处理要注意总线请求信号和寄存器写信号的控制。

三、实验过程（60%）

（一）实验流水账

2018.12.9 20:00~23:00 构思，写代码

2018.12.10 20:30~23:30 构思，写代码

2018.12.11 8:30~9:30 写代码 15:30~17:00 写代码，调试

2018.12.16 19:00~23:00 调试

2018.12.17 16:00~18:00 19:00~23:00 调试

2018.12.18 8:30~11:30 13:00~16:00 调试

（二）错误记录

1、错误 1

（1）错误现象

0xbfc00000 处指令出错。

（2）分析定位过程

查看波形，检查第一条指令和第二条指令的执行。

（3）错误原因

第一条指令从 IF 阶段进入到 ID 阶段的 ready_go 信号有问题。

（4）修正效果

修改 ready_go 信号的赋值，问题解决。

2、错误 2

（1）错误现象

控制台报错，显示 PC 不正确。

（2）分析定位过程

查看波形，找到相应时刻，对比汇编代码寻找原因。

（3）错误原因

一条条件跳转指令若跳转，应该用分支延迟槽的 PC 加上偏移得到目标地址。

（4）修正效果

修改分支指令跳转时的目标地址的计算，问题解决。

3、错误 3

(1) 错误现象

一条分支指令的延迟槽指令被执行了两次。

(2) 分析定位过程

查看波形，找到相应时刻，查看发出的取指请求。

(3) 错误原因

分支延迟槽的指令响应了两次取指请求，返回来两条指令，所以执行了两次。

(4) 修正效果

修改分支指令之后的取指请求，问题解决。

4、错误 4

(1) 错误现象

一条除法指令之后的指令执行不正确。

(2) 分析定位过程

查看波形，找到相应时刻，查看指令执行情况。

(3) 错误原因

乘法执行的时候要阻塞 EX 阶段，此时 EX 阶段的 allowin 应该为零。

(4) 修正效果

修改阻塞 EX 阶段的代码，问题解决。

5、错误 5

(1) 错误现象

一条 lui 指令的写寄存器堆信号未置起。

(2) 分析定位过程

查看波形，找到相应时刻，检查写寄存器信号的赋值。

(3) 错误原因

lui 指令之前 lw 指令意外产生了一个例外标记，使得写寄存器信号未被置起。

(4) 修正效果

修改 badvaddr 例外的产生，问题解决。

6、错误 6

(1) 错误现象

控制台报错，写寄存器的值不对。

(2) 分析定位过程

查看波形，找到相应时刻，对比汇编代码寻找原因。

(3) 错误原因

swl 和 swr 指令某些情况需要些 3 字节，而类 SRAM 接口不支持 3 字节读写，故需要将这些情况的写请求转换为两次写请求。

(4) 修正效果

修改 swl 和 swr 指令将数据写入 RAM 的逻辑，问题解决。

7、错误 7

(1) 错误现象

syscall 指令执行不正确。

(2) 分析定位过程

查看波形，找到相应时刻，对比汇编代码寻找原因。

(3) 错误原因

syacall 后的 sw 指令的写请求未拉低。

(4) 修正效果

修改例外发生后，其后指令的控制信号，问题解决。

8、错误 8

(1) 错误现象

波形停止，发现组合环。

(2) 分析定位过程

先进行综合，查看 warning 和时序报告。

(3) 错误原因

地址错的例外判断用到了 data_req，而 data_req 的赋值又用到了例外判断，形成了一个组合环。

(4) 修正效果

修改地址错例外判断和 data_req 的赋值，问题解决。

9、错误 9

(1) 错误现象

CPU 陷入死循环。

(2) 分析定位过程

查看波形，找到相应时刻，对比汇编代码寻找原因。

(3) 错误原因

未打开时钟中断。

(4) 修正效果

打开时钟中断，增加对中断的处理。

10、错误 10

(1) 错误现象

控制台报错，显示的是写回寄存器的值不对。

(2) 分析定位过程

查看波形，找到相应时刻，对比汇编代码寻找原因。

(3) 错误原因

发现是一条 jr 指令之后跟了 syscall 指令，jr 的目标地址会发生地址错例外，本应该响应的是 syscall 的例外，但却错误的相应了地址错的例外。

(4) 修正效果

添加对类似情况的处理，问题解决。

11、错误 11（上板未通过，数码管最终显示为 5E00005C，调试中）