

计算机组成原理

实验4 AXI4总线接口设计

2025 · 春

哈工大



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验目的

- 了解CPU与SoC的关系，熟悉SoC的基本架构
- 了解基于ready-valid握手协议的总线工作原理
- 熟悉AXI4总线协议，掌握其总线接口的实现方法



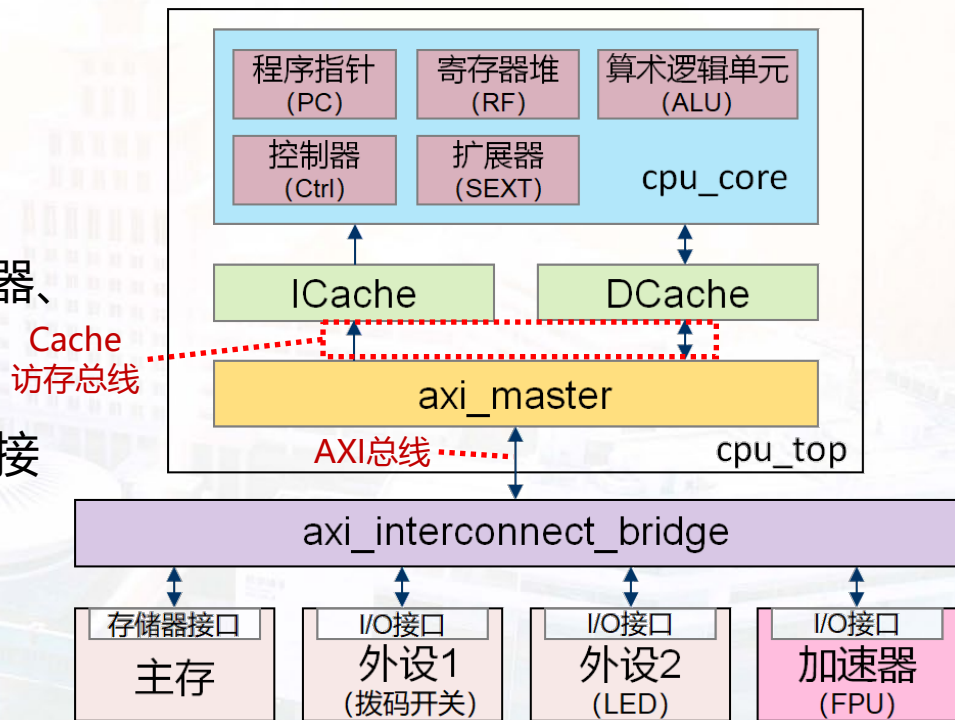
实验内容

- 在SoC模板工程上，自行实现**axi_master模块**，使得CPU能够通过AXI4总线协议访问存储器和I/O设备
 - 实现读地址（AR）、读数据（R）两个通道
 - 所实现的AR、R通道支持猝发传输
 - 为简化设计，axi_master模块总是先处理完一个请求，再处理下一个请求，即不考虑连续数据访问、读写请求重叠的情况
- 运行功能仿真，通过所有测试用例

实验原理

- 模板工程的SoC架构

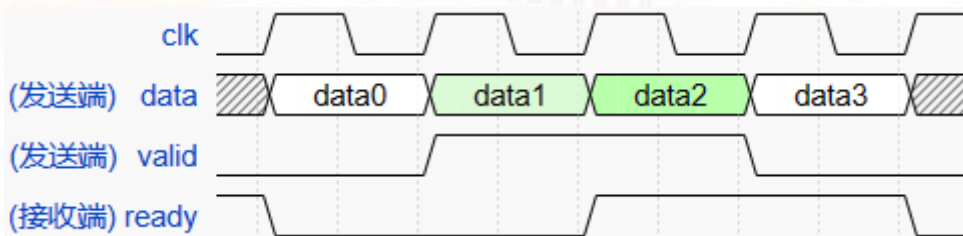
- CPU和Cache通过访存总线与axi_master互连
- axi_master通过AXI总线与存储器、外设互连
- 总线桥实现CPU与多个设备的连接



实验原理

◆ ready-valid握手协议：保证信息在发送端、接收端之间准确无误传输

- ready信号：接收端的就绪状态
- valid信号：发送端发送数据的有效状态

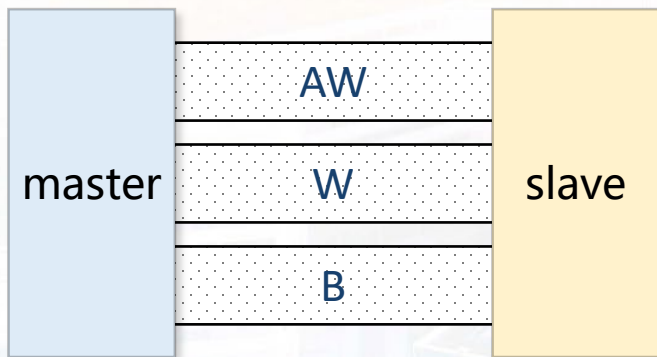


- 对发送端：当且仅当valid有效时，数据才有效 —— data1、data2是有效数据
 - 对接收端：当且仅当ready有效时，才可能接收数据 —— 只有data2被接收
- ◆ 当且仅当**valid、ready同时有效**时，数据才能在clk上升沿时成功传输

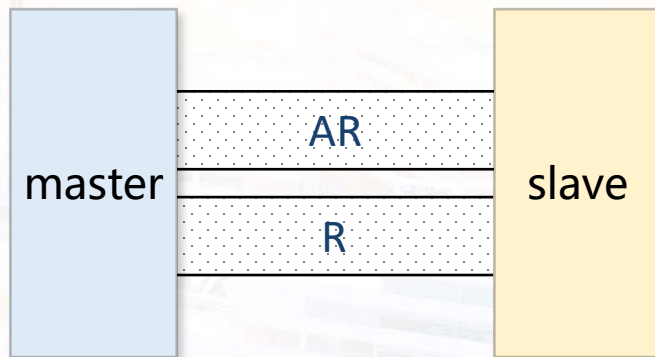
实验原理

◆ AXI4总线协议：基于ready-valid握手协议的高性能嵌入式总线

- 包含5个独立通道：写地址(**AW**)、写数据(**W**)、写响应(**B**)、读地址(**AR**)、读数据(**R**)



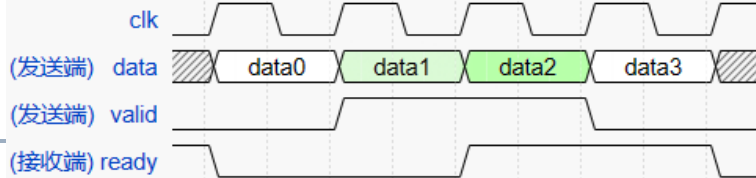
3个通道用于写请求



2个通道用于读请求

- 每个通道都基于ready-valid信号实现握手

实验原理



◆ AXI4读通道：AR通道、R通道

- 读通道信号：

		通道	信号	位宽	方向	释义
master 发送 slave 接收	读地址 (AR)		araddr	32	主→从	读地址
			arlen	8	主→从	猝发传输的数据包个数
			arsize	3	主→从	每次传输的数据包大小
			arburst	2	主→从	猝发传输的地址生成方式
			arvalid	1	主→从	读地址的有效信号
			arready	1	从→主	从设备AR通道的就绪信号
slave 发送 master 接收	读数据 (R)		rready	1	主→从	主设备R通道的就绪信号
			rdata	32	从→主	读数据
			rlast	1	从→主	猝发传输最后一个数据包的标志位
			rvalid	1	从→主	读响应的有效信号

实验原理

◆ AXI4读通道：AR通道、R通道

- 猝发传输相关信号：
 - arlen (0~255): 控制猝发传输时连续传输arlen+1个数据包
 - arsize (0~7): 控制猝发传输数据包的大小为 2^{arsize}
 - arburst: 控制猝发传输的地址生成方式

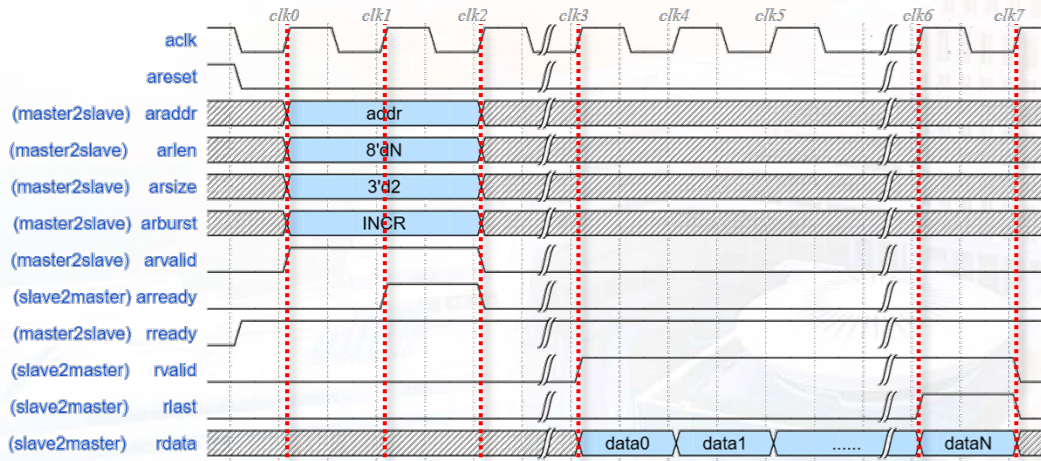
取值	地址生成方式	释义
00	FIXED	固定地址模式，每次访问同一个地址（如FIFO）
01	INCR	递增地址模式，每次访问地址+1（如存储器）
10	WRAP	回环地址模式，地址在一定范围内循环（如缓存器）

- rlast: 指示猝发传输最后一个数据包的标志位信号

实验原理

◆ AXI4读时序:

- 主设备: axi_master 从设备: 存储器/外设



【clk0】 master发送读地址, 设定猝发传输参数

【clk1】 slave的AR通道就绪(arready有效), 接收读地址

【clk2】 master撤掉读地址请求; slave执行读操作

【clk3】 读操作完成, slave每个时钟返回1个读数据

【clk6】 slave返回最后1个数据时拉高rlast信号

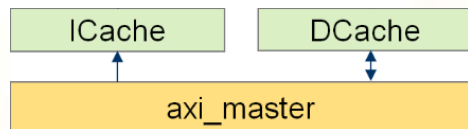
【clk7】 slave拉低rvalid和rlast, 传输结束

在rlast有效后, 主设备才能发起新请求

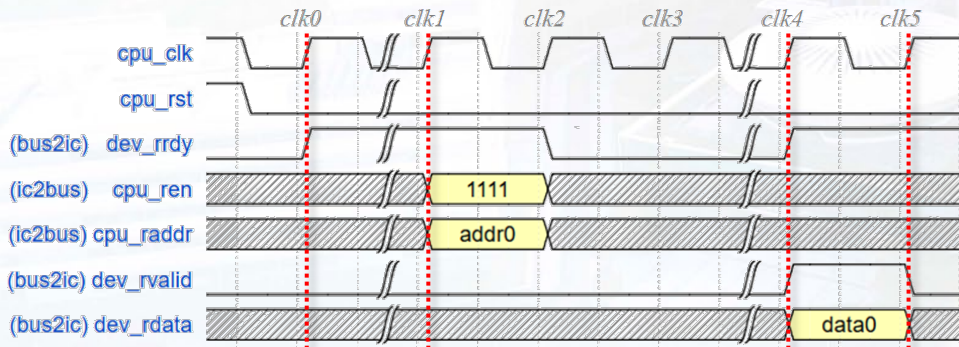
实验原理

◆ Cache访存总线

- 主设备：Cache 从设备：axi_master



信号	位宽	方向	释义
dev_rrdy	1	从→主	从设备R通道的就绪信号
cpu_ren	1	主→从	读使能，支持读字、半字和字节
cpu_raddr	32	主→从	读地址
dev_rvalid	1	从→主	读数据的有效信号
dev_rdata	256	从→主	读数据



【clk0】slave就绪，拉高rrdy以接收master访存请求

【clk1】master发出请求，slave拉低rrdy，开始处理读请求

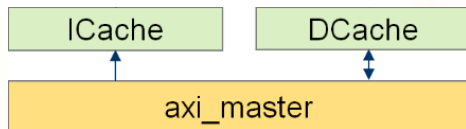
【clk4】slave拉高rvalid，把读取的数据块返回给master

【clk5】slave撤掉rvalid，传输结束

实验原理

◆ Cache访存总线

- axi_master同时与ICache、DCache互连



```
1 | // ICache Interface
2 | output reg          ic_dev_rrdy;
3 | input  wire         ic_cpu_rren;
4 | input  wire [ 31:0] ic_cpu_raddr;
5 | output reg          ic_dev_rvalid;
6 | output reg [255:0] ic_dev_rdata;
7 | // DCache Read Data Interface
8 | output reg          dc_dev_rrdy;
9 | input  wire         dc_cpu_rren;
10 | input  wire [ 31:0] dc_cpu_raddr;
11 | output reg          dc_dev_rvalid;
12 | output reg [255:0] dc_dev_rdata;
```

- 若同时检测到ICache、DCache的访存请求，优先处理DCache的访存请求
- 本实验不考虑同时出现的情形，但实现时注意**DCache的访问请求处理应优先于ICache**

实验步骤

1. 打开SoC模板工程，完成axi_master.v的AR、R通道
2. 为缩短调试时间，注释defines.vh头文件前3行

```
1|// `define RANDOM_DELAY  
2|// `define ENABLE_ICACHE  
3|// `define ENABLE_DCACHE
```

3. 运行功能仿真，根据仿真波形完成调试
4. 总结及反思，撰写实验报告

验收及提交

- **课堂验收：**

- 课上检查是否通过Testbench的所有测试用例：2.5分

- **提交内容**

- axi_master.v: 1.5分
 - 实验报告（按模板完成）：4分

- 将上述文件打包成.zip，以“学号_姓名.zip”命名提交到作业系统

- ◆ 注意：**如有雷同，双方均0分！**



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ