

# Lab 9

22302010022 曾华正

## ▼ Lab 9

- [上板测试通过截图](#)

## ▼ 回顾

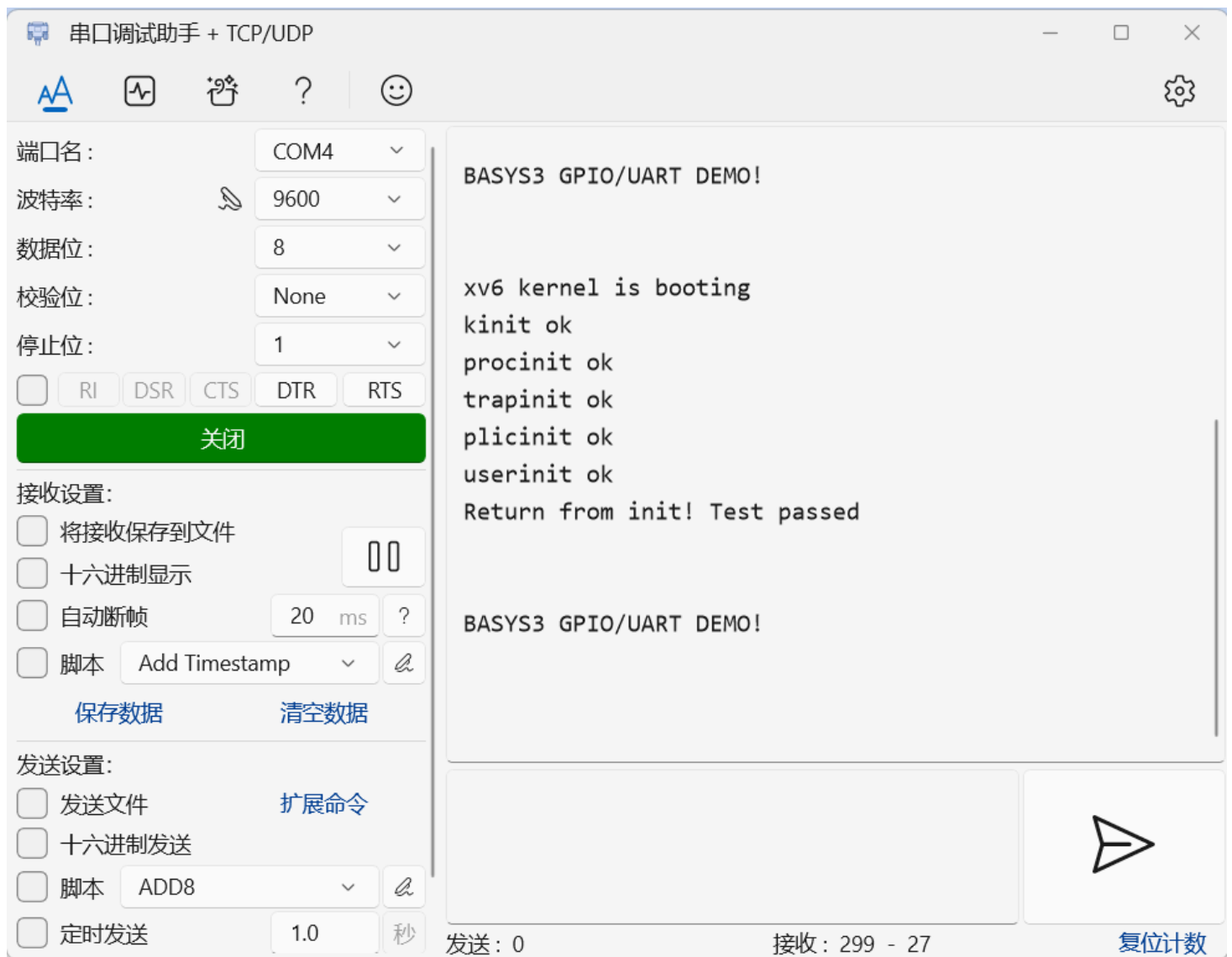
- [单周期cpu](#)
- [流水线cpu](#)
- [特权指令](#)
- [Thinking](#)

本次实验报告包括本学期的lab总结

2024Spring-Arch完结 感谢各位老师的帮助 祝一切顺利

## 上板测试通过截图

需要按reset按钮才能触发



## 回顾

### 单周期cpu

- 基本指令  
使用decoder、controlunit解析
- 解决访存指令的延迟
  - 阻塞
  - 暂存
- 乘除法
  - div rem divu remu
    - 引入新的单元除法器，在除法器中，类似ibus与dbus，引入start和ok信号，标识是否开始除法，除法是否结束
  - mul

- 引入新的单元乘法器，与除法器类似，引入start, ok, 以及阻塞计算过程为根据乘数的对应位
- L S类型指令
  - 新增 readdata 和writedata 用于对W,H,B类型的写入与读取进行对齐；并且输出相应的strobe、写入内存的数据、从内存中读取的数据

## 流水线cpu

### 详见lab6文档

- 使用中间寄存器传递各个组件之间的信息
- 采用转发以及阻塞的技术来解决数据冒险，loadanduse需要阻塞，WAR不需要阻塞
- 对于分支冒险，采用阻塞的方法来解决，未使用分支预测

## 特权指令

- CSR寄存器
    - 由于当是csr指令且fun3为0时 mret会被置高位，需要过滤encall、sfence等指令（当时设计信号时没想到会有这些指令，所以后续有点像打补丁）
- ```
else if (is_mret & wa!=12'b0 & wa != 12'h120)
```
- 在阻塞结束后 写入寄存器
    - 阻塞为了防止在执行csr指令时候的顺序 导致结果异常 例如先写入register后 写入csr的数据就会错
  - mode 在core中进行赋值
- 虚拟地址
  - 先识别模式以及stap，来判断是否为虚拟地址
    - 物理地址 直接访存
    - 虚拟地址
      - i访存：设置count=3，先进行三次d访存，将最后一次访存结果拼接后，进行i访存
      - d访存：设置count=4，进行四次访存，前三次访存结果拼接得到下一次的地址，最后一次结果存入memdata

## Thinking

- 做实验之前需要了解内容，不能像过OJ一样来编写代码，否则就会造成大量补丁，然后就会有一堆multi-driven
- 时序处理在cpu中极为重要

- 一定要初始化所有信号，vivado不会把未赋值的变量设置为0，而是x，导致错误