



中国科学技术大学
University of Science and Technology of China

计算机组成原理

Lab4 单周期CPU设计

计算机实验教学中心

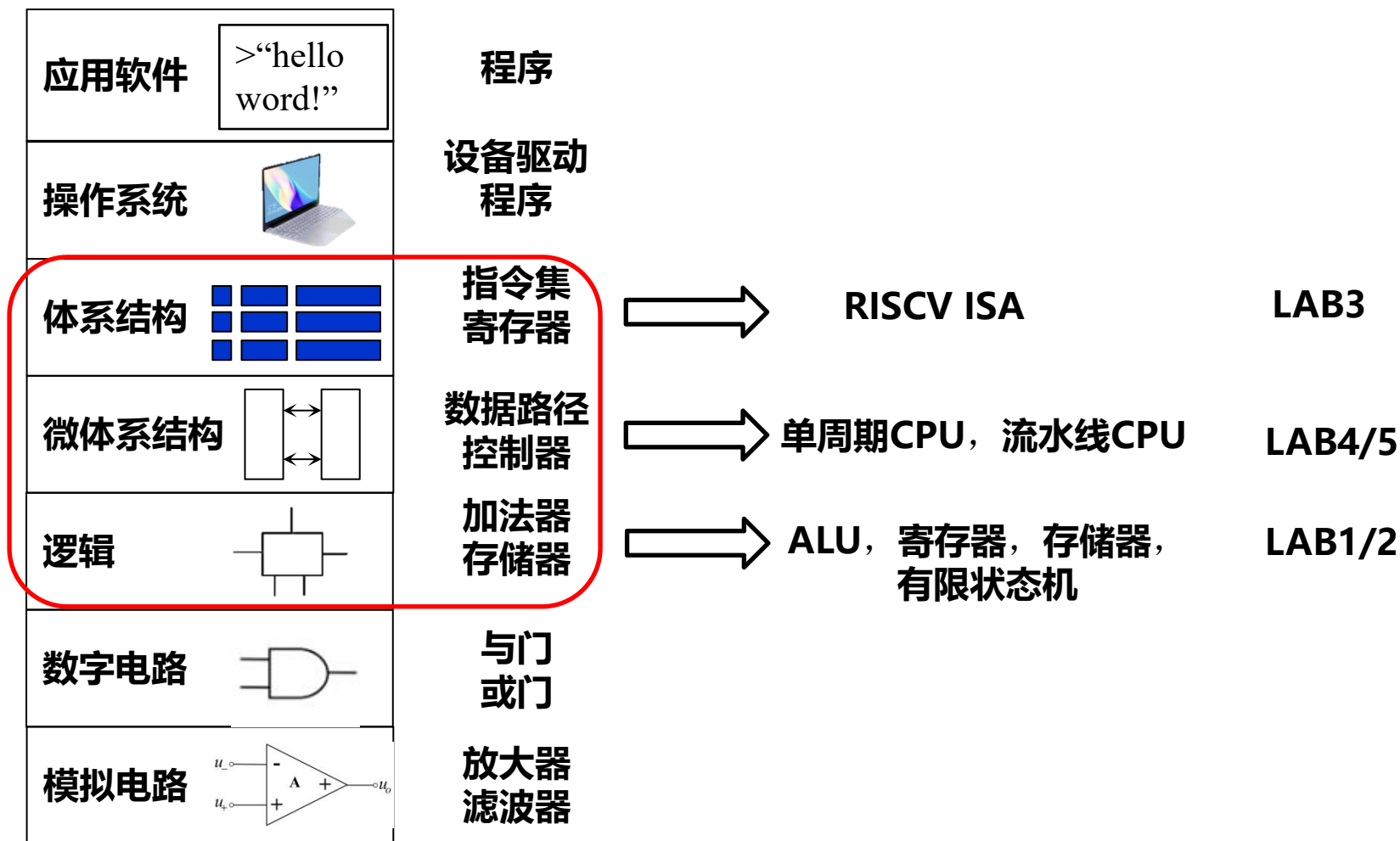
2023-4-17

实验目标

- 理解单周期CPU的结构和工作原理
- 熟练掌握单周期CPU数据通路和控制器的设计和描述方法
- 理解单周期CPU的调试方法

实验原理

1.计算机系统的抽象层次

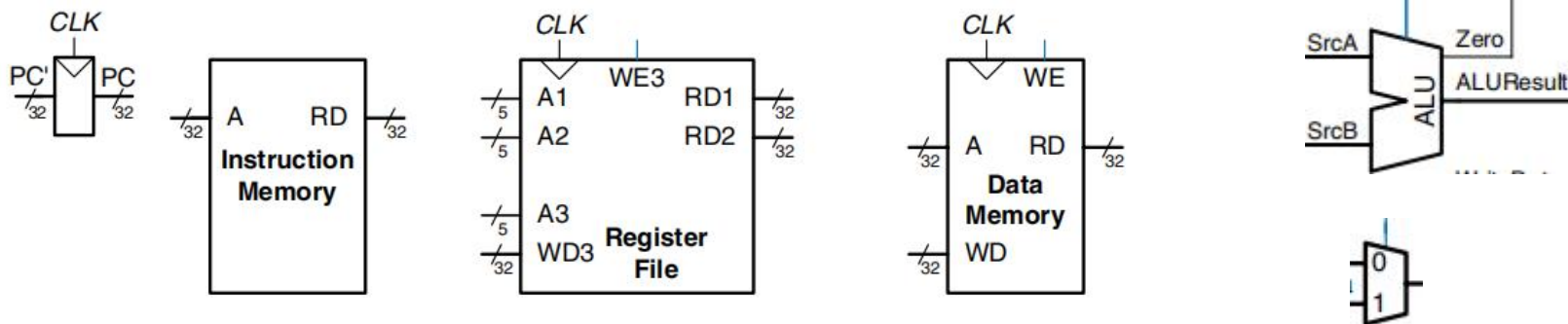


实验原理

2.微体系结构

□ 数据通路

- **存储状态的单元**：程序计数器、寄存器文件、指令存储器和数据存储器；
- **处理数据的组合单元**：与或门，ALU，复用器



□ 控制器

- 将指令译码产生ALU功能选择，复用器选择，寄存器使能，存储器写信号来**控制数据通路的操作**。

实验原理

3.RV32I指令格式及功能

□ RV32I指令格式

31	27	26	25	24	20	19	15	14	12	11	7	6	0	
funct7				rs2		rs1		funct3		rd		opcode		R-type
imm[11:0]						rs1		funct3		rd		opcode		I-type
imm[11:5]				rs2		rs1		funct3		imm[4:0]		opcode		S-type
imm[12 10:5]				rs2		rs1		funct3		imm[4:1 11]		opcode		B-type
imm[31:12]										rd		opcode		U-type
imm[20 10:1 11 19:12]										rd		opcode		J-type

□ RV32I指令功能

- ✓ add rd, rs1, rs2 # $x[rd] = x[rs1] + x[rs2]$
- ✓ addi rd, rs1, imm # $x[rd] = x[rs1] + sext(imm)$
- ✓ auipc rd, imm # $x[rd] = pc + imm[31:12] \ll 12$
- ✓ lui rd, imm # $x[rd] = imm[31:12] \ll 12$
- ✓ lw rd, offset(rs1) # $x[rd] = M[x[rs1] + sext(offset)]$
- ✓ sw rs2, offset(rs1) # $M[x[rs1] + sext(offset)] = x[rs2]$

实验原理

3.RV32I指令格式及功能

□ RV32I指令功能

- ✓ beq rs1, rs2, offset # if (rs1 == rs2) pc += sext(offset)
- ✓ blt rs1, rs2, offset # if (rs1 <_s rs2) pc += sext(offset)
- ✓ jal rd, offset # x[rd] = pc+4, pc += sext(offset)
- ✓ jalr rd, offset(rs1) # t = pc + 4; pc = (x[rs1] + sext(offset)) & ~1; x[rd]=t

实验原理

4.RV32I指令编码

31	25 24	20 19	15 14	12 11	7 6	0	
imm[11:0]		rs1	111	rd	0010011	I andi	
0000000	shamt	rs1	001	rd	0010011	I slli	
0000000	shamt	rs1	101	rd	0010011	I srli	
0100000	shamt	rs1	101	rd	0010011	I srai	
0000000	rs2	rs1	000	rd	0110011	R add	
0100000	rs2	rs1	000	rd	0110011	R sub	
0000000	rs2	rs1	001	rd	0110011	R sll	
0000000	rs2	rs1	010	rd	0110011	R slt	
0000000	rs2	rs1	011	rd	0110011	Rsltu	
0000000	rs2	rs1	100	rd	0110011	R xor	
0000000	rs2	rs1	101	rd	0110011	R srl	
0100000	rs2	rs1	101	rd	0110011	R sra	
0000000	rs2	rs1	110	rd	0110011	R or	
0000000	rs2	rs1	111	rd	0110011	R and	

实验原理

4.RV32I指令编码

31	25 24	20 19	15 14	12 11	7 6	0
imm[31:12]				rd	0110111	U lui
imm[31:12]				rd	0010111	U auipc
imm[20:10:1 11 19:12]				rd	1101111	J jal
imm[11:0]		rs1	000	rd	1100111	I jalr
imm[12 10:5]	rs2	rs1	000	imm[4:1 11]	1100011	B beq
imm[12 10:5]	rs2	rs1	001	imm[4:1 11]	1100011	B bne
imm[12 10:5]	rs2	rs1	100	imm[4:1 11]	1100011	B blt
imm[12 10:5]	rs2	rs1	101	imm[4:1 11]	1100011	B bge
imm[12 10:5]	rs2	rs1	110	imm[4:1 11]	1100011	B bltu
imm[12 10:5]	rs2	rs1	111	imm[4:1 11]	1100011	B bgeu
imm[11:0]		rs1	000	rd	0000011	I lb
imm[11:0]		rs1	001	rd	0000011	I lh
imm[11:0]		rs1	010	rd	0000011	I lw
imm[11:0]		rs1	100	rd	0000011	I lbu
imm[11:0]		rs1	101	rd	0000011	I lhu
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	S sb
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	S sh
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	S sw

实验原理

5.单周期CPU数据通路

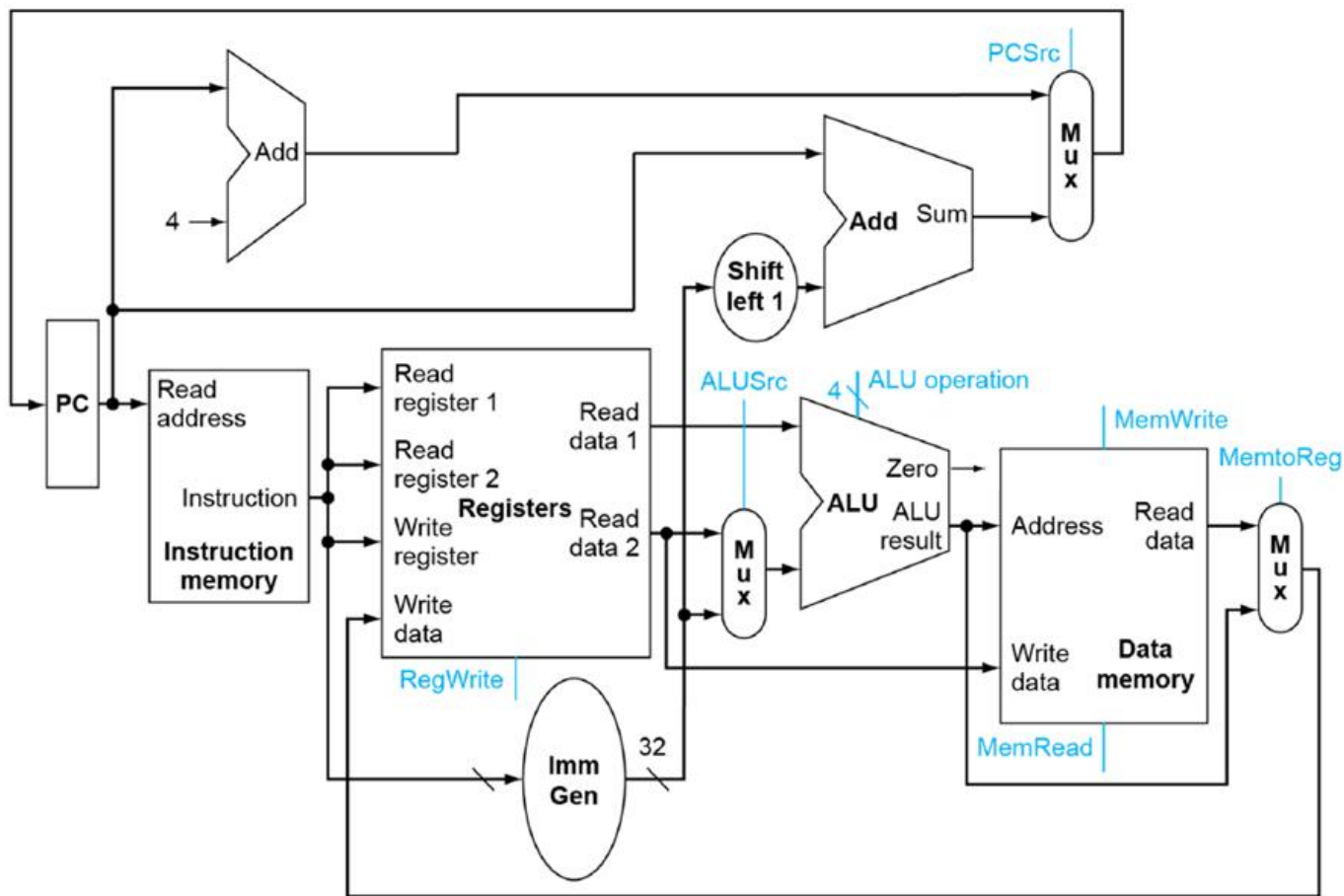


图1 单周期数据通路（无控制器）

实验原理

5.单周期CPU数据通路

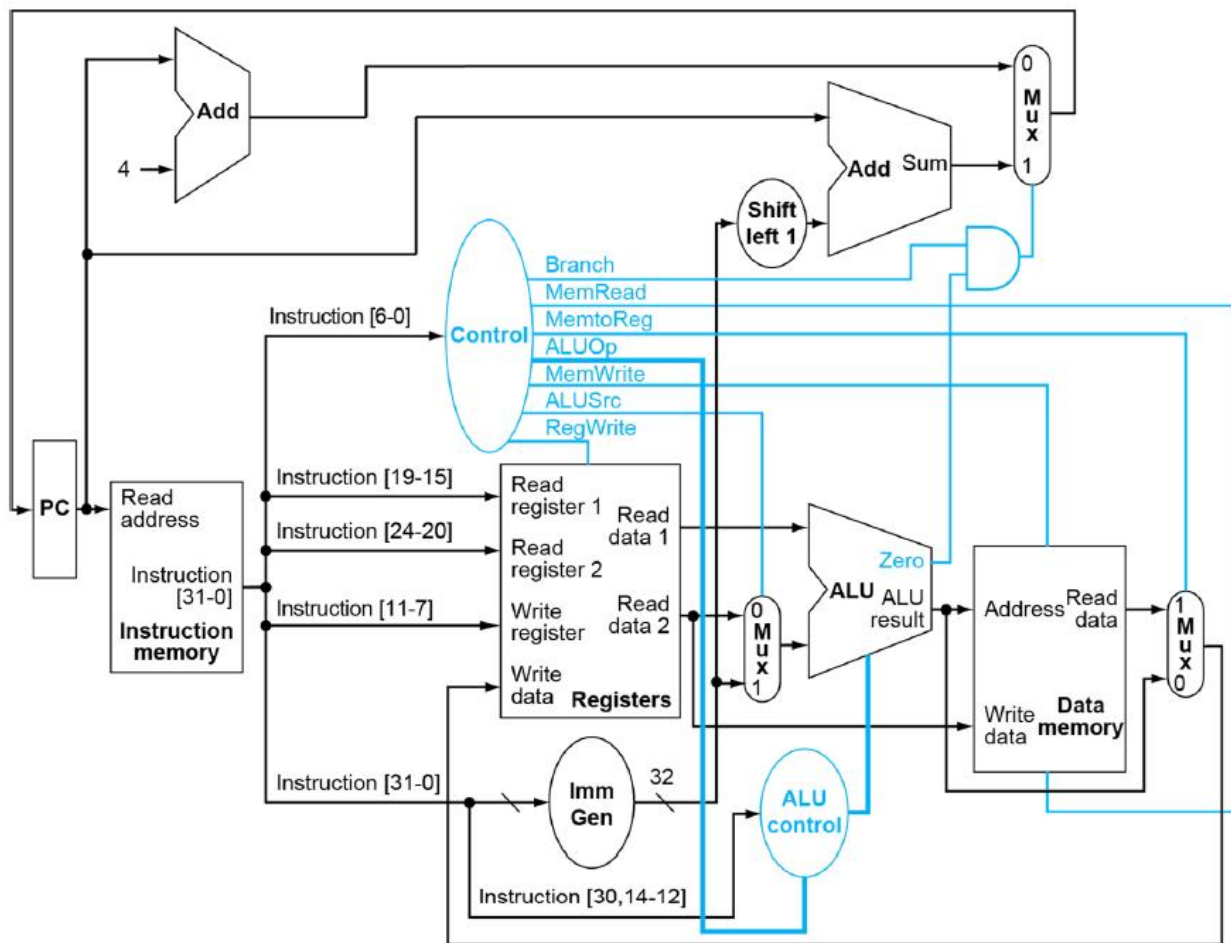


图2 单周期数据通路

实验原理

6.PDU:外设管理与调试单元-----提供PDU代码及说明手册

□ PDU 功能

- ✓ 通过串口控制CPU运行方式，及查看CPU状态寄存器数据和数据路径上的数据，详见《PDU指令手册》

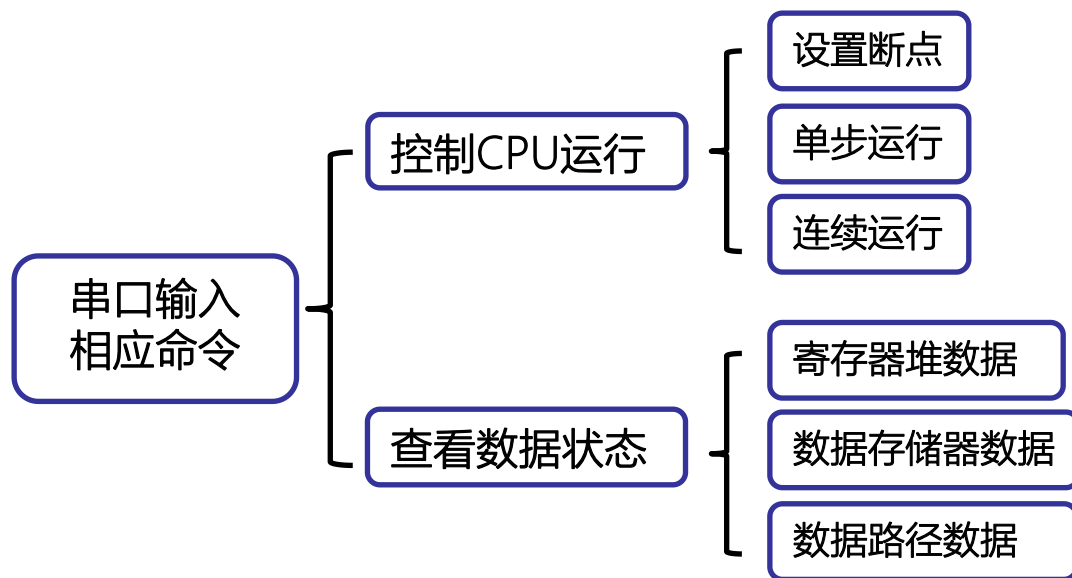


图3 串口控制功能结构简图

实验原理

6.PDU:外设管理与调试单元----提供PDU代码及说明手册

□ PDU 功能

✓ 选择和分配输入输出设备，以便数据传输；控制输入输出设备和CPU（或内存）之间交换数据，FPGAOL 平台外设，主要包括：

- 开关：共计 8 个：其中sw[6:0]作为数据输入（本实验不用）； sw[7]作为系统复位信号；
- 按钮：共计 1 个，本实验不用；
- 七段数码管：共计 8 个，采用分时复用的方式进行扫描显示，显示输入输出数据；
- led: 系统工作状态指示；
- UART 串口：支持 FPGA 与用户之间的指令传输。

实验原理

7.最小片上系统-----提供PDU代码及说明手册

□ 系统框图

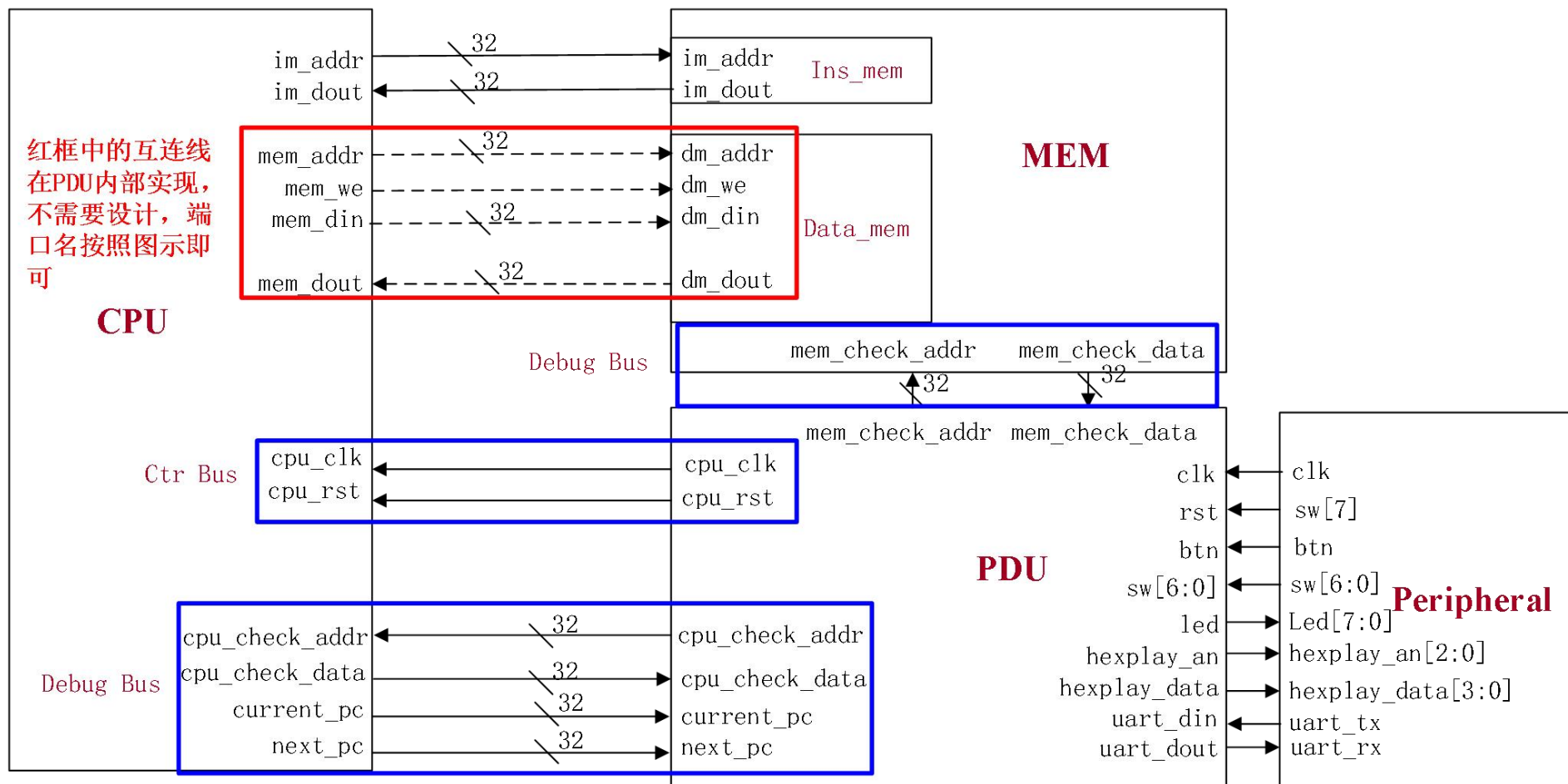


图4 最小片上系统系统框图

实验原理

7.最小片上系统----提供PDU代码及说明手册

□ 系统工作状态指示

表1 系统工作状态指示

数值 led	1	0
led[7]	CPU 正在运行	CPU暂停
led[6]	数码管显示开关输入数据	其他
led[5]	数码管显示 CPU 输出数据	其他
led[4]	数码管显示 DEBUG 数据	其他
led[1]	指令检测失败	其他
led[0]	指令检测成功	其他

实验原理

7.最小片上系统-----提供PDU代码及说明手册

□ Debug接口

✓ 在CPU暂停时，可以依次查看CPU数据路径上的数据和寄存器堆数据以及MEM中数据存储器数据；

✓ 跟CPU运行程序无关

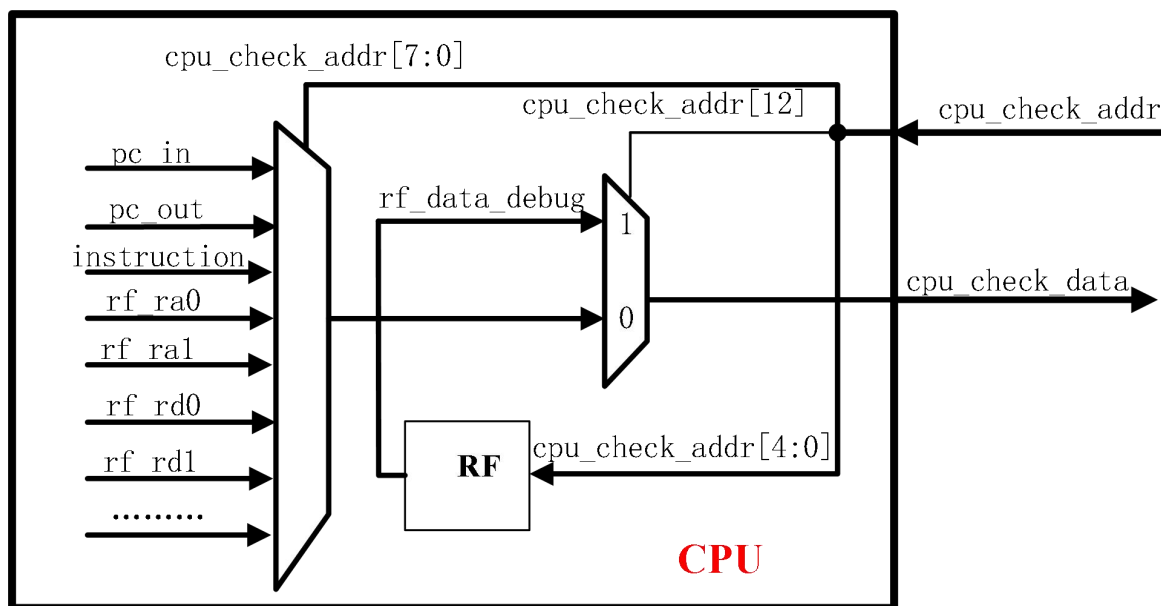


图5 CPU内部debug接口信号处理

实验原理

7.最小片上系统-----提供PDU代码及说明手册

□ 控制接口

- ✓ PDU产生cpu_clk, cpu_rst, 控制CPU运行方式;
- ✓ CPU内部不需要处理, 接线即可使用。

□ CPU模块接口

```
module CPU(  
    input clk,  
    input rst,  
    // ---MEM Data BUS with CPU---//  
    //指令存储器端口  
    output [31:0] im_addr,  
    input [31:0] im_dout,  
    //数据存储器端口  
    output [31:0] mem_addr,  
    output mem_we,  
    output [31:0] mem_din,  
    input [31:0] mem_dout,  
    // Debug BUS with PDU//  
    output [31:0] current_pc,  
    output [31:0] next_pc,  
    input [31:0] cpu_check_addr,  
    output reg [31:0] cpu_check_data  
);
```


实验内容和要求[必做]

1. 设计单周期CPU数据通路并进行功能仿真

- ✓ CPU数据通路需要支持十条指令: add addi lui auipc lw sw beq blt jal jalr;
- ✓ ALU采用Lab1设计的模块;
- ✓ 修改Lab2寄存器堆模块, 增加1个用于调试的读端口;
- ✓ 指令存储器采用分布式ROM(256x32位), 地址范围: 0x0000_3000 ~ 0x0000_33ff;
- ✓ 数据存储器采用分布式Dual Port Ram(256x32位), 地址范围: 0x0000_0000 ~ 0x0000_03ff, 其中一个读端口用于调试;
- ✓ 结构化描述单周期CPU, vivado工程结构:
 - |--+top.v
 - | |--MEM.v:内部例化数据存储器 and 指令存储器IP (需要例化IP)
 - | |--CPU.v:单周期数据通路 (需要设计)
 - | |--PDU.v:外设管理及调试单元 (提供所有代码)
- ✓ 对单周期CPU进行功能仿真。

实验内容和要求[必做]

2. 测试Lab3生成的FLS.coe---提供演示视频

- ✓ 使用Lab3必做实验要求1生成的COE文件作为指令存储器初始化文件;
- ✓ 将硬件电路下载至FPGAOL中测试;
- ✓ 采用PDU串口指令查看FLS存放在数据存储器中的数据。

3. 测试test.coe---提供演示视频

- ✓ 将给出的test.asm文件生成coe文件作为指令存储器初始化文件:
- ✓ 将硬件电路下载至FPGAOL中测试。

实验内容和要求[选做]

1. 扩展单周期 CPU 指令集

□ 在原有 10 条指令的基础上，增加对于 RV32I 中如下指令的支持

- ✓ 移位指令 sll、slli、srl、srli、sra、srai
- ✓ 算数与逻辑指令 sub、xor、xori、or、ori、and、andi
- ✓ 条件置数指令 slt、slti、slti、sltiu
- ✓ 分支指令 bne、bge、bltu、bgeu
- ✓ 访存指令 lb、lh、lbu、lhu、sb、sh

任意实现上述指令中的三类，每类任选三条及以上即可得到本项满分。注意：你需要自己设计测试程序以验证指令的正确性。



中国科学技术大学
University of Science and Technology of China

The End