实验三 单周期CPU

zjx@ustc.edu.cn

2020.5.6

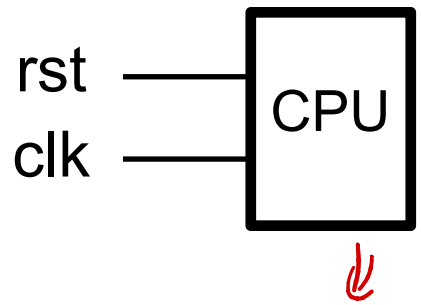
实验目标

- 理解计算机硬件的基本组成、结构和工作原理
- 掌握数字系统的设计和调试方法
- 熟练掌握数据通路和控制器的设计和描述方法

实验内容

1. **设计实现单周期CPU，可执行如下6条指令：**

- **add, addi**
- **lw, sw**
- **beq, j**



– **分析指令功能，设计数据通路和控制单元**

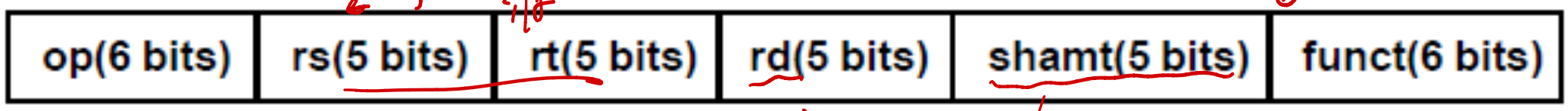
– **ALU和寄存器堆：实验1和实验2设计的模块**

– **指令存储器：256 x 32位ROM，IP例化，分布式存储器**

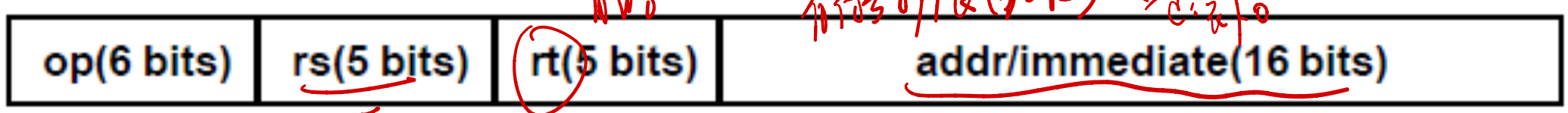
– **数据存储器：256 x 32位RAM，IP例化，分布式存储器**

指令功能与格式

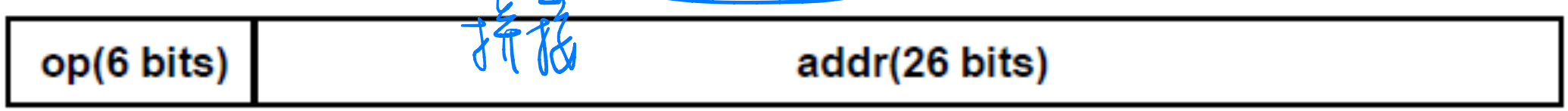
- **add**: rd <- rs + rt; op = 000000, funct = 100000



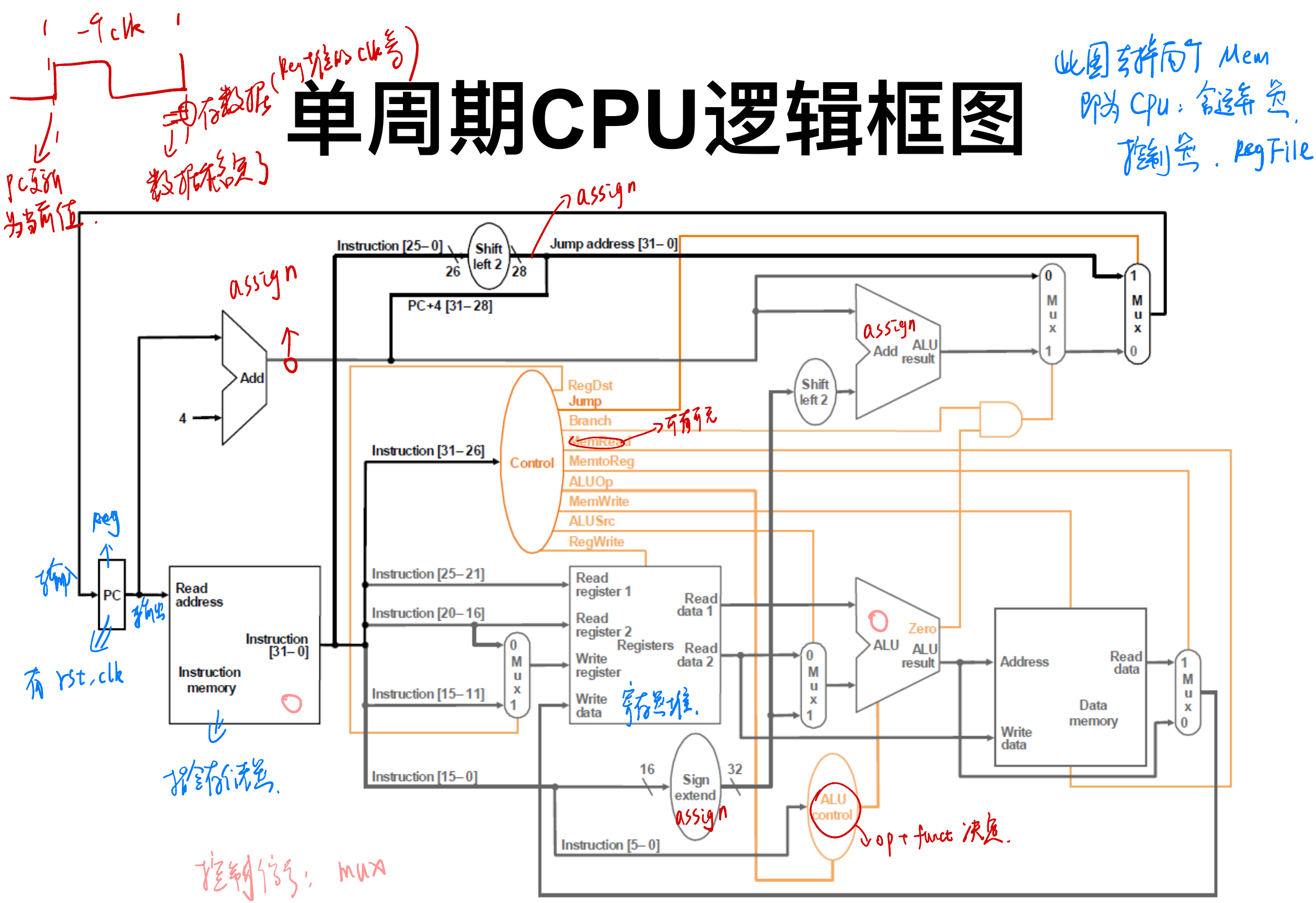
- **addi**: rt <- rs + imm; op = 001000
  lw: rt <- M(rs + addr); op = 100011
  sw: M(rs + addr) <- rt; op = 101011
  beq: if (rs = rt) then pc <- pc + 4 + addr << 2
  else pc <- pc + 4; op = 000100



- **j**: pc <- (pc+4)[31:28] | (add<<2)[27:0]; op = 000010



单周期CPU逻辑框图


Instruction [25–0]
Shift left 2
26
28
Jump address [31–0]
PC+4 [31–28]
Add
4
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
Shift left 2
Add
ALU result
0 M u x 1
1 M u x 0
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
ALU
Zero
ALU result
16
Sign extend
32
ALU control
Address
Read data
Data memory
Write data

仿真时：例化CPU，把要看的信号引出来、⇒直接仿真时object里边拖出来

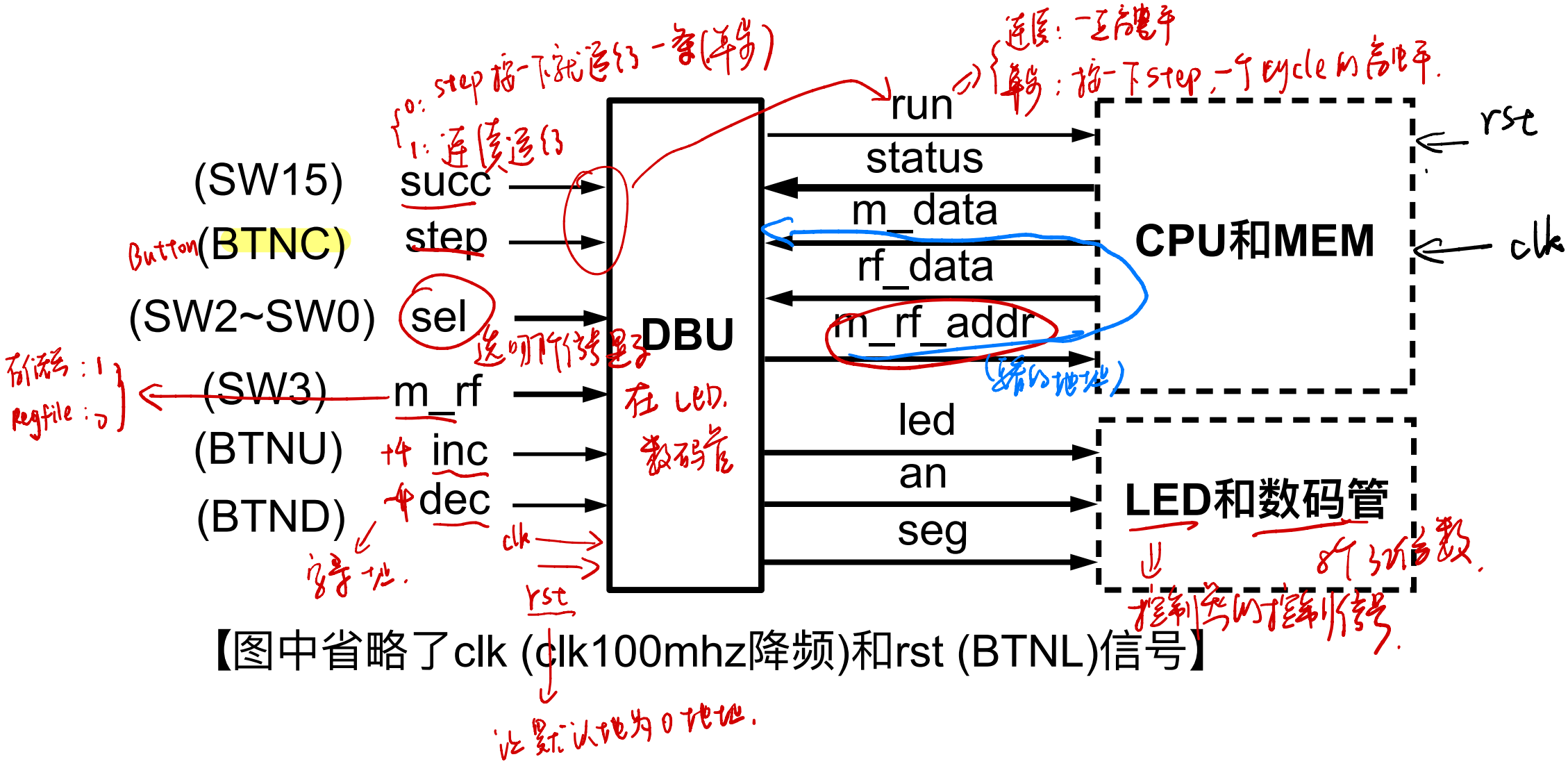
⇓

不方便.

实验内容（续1）

2. DBU：Debug Unit，调试单元

- 下载测试时，用于控制运行方式，显示运行状态和运行结果



【图中省略了clk (clk100mhz降频)和rst (BTNL)信号】

实验内容（续2）

- **控制CPU运行方式**
  - succ = 1：控制CPU连续执行指令，run = 1（一直维持）
  - succ = 0：控制CPU执行一条指令，每按动step一次，run输出维持一个时钟周期的脉冲
- **sel = 0：查看CPU运行结果 (存储器或者寄存器堆内容)**
  - m_rf： 1，查看存储器(MEM)；0，查看寄存器堆(RF)
  - m_rf_addr： MEM/RF的调试读口地址(字地址)，复位时为零
  - inc/dec：m_rf_addr加1或减1
  - rf_data/m_data：从RF/MEM读取的数据字
  - 16个LED指示灯显示m_rf_addr
  - 8个数码管显示rf_data/m_data

实验内容（续3）

- **sel = 1 ~ 7：查看CPU运行状态（status）**
  - 12个LED指示灯(SW11~SW0)依次显示控制器的控制信号(Jump, Branch, Reg_Dst, RegWrite, MemRead, MemtoReg, MemWrite, ALUOp, ALUSrc)和ALUZero，其中ALUOp为3位
  - 8个数码管显示由sel选择的一个32位数据

sel = 1： pc_in, PC的输入数据

sel = 2： pc_out, PC的输出数据

sel = 3： instr, 指令存储器的输出数据

sel = 4： rf_rd1, 寄存器堆读口1的输出数据

sel = 5： rf_rd2, 寄存器堆读口2的输出数据

sel = 6： alu_y, ALU的运算结果

sel = 7： m_rd, 数据存储器的输出数据

实验步骤

1. 结构化描述单周期CPU的数据通路和控制器，并进行功能仿真
2. 设计实现调试单元（DBU），并进行功能仿真
3. 将CPU和DBU下载至FPGA中测试

实验检查

- 检查单周期CPU的功能仿真
- 检查调试单元DBU的功能仿真
- 检查CPU和DBU下载至FPGA后的运行功能

思考题

- **修改数据通路和控制器，增加支持如下指令：**

累加

accm: rd <- M(rs) + rt;　　op = 000000, funct = 101000

(lw+add的功能实现在一条指令

op(6 bits)	rs(5 bits)	rt(5 bits)	rd(5 bits)	shamt(5 bits)	funct(6 bits)

The End