

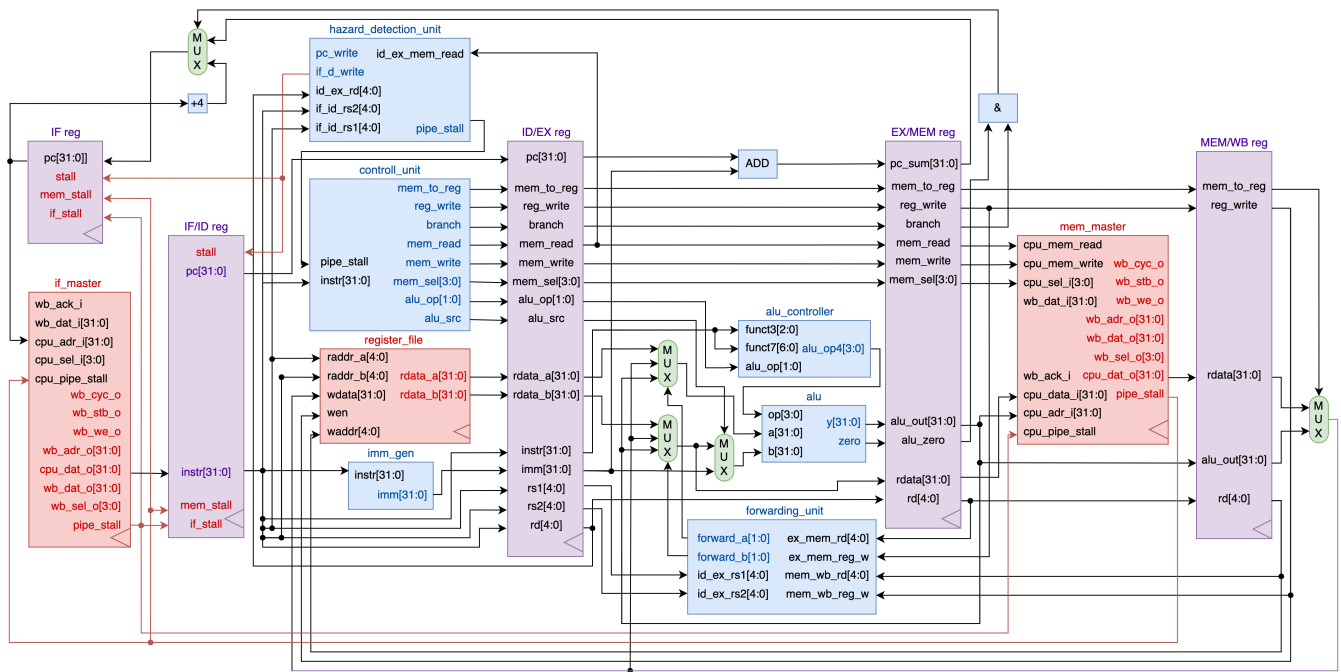
计算机组成原理 · 实验6报告

计01 容逸朗 2020010869

实验过程

CPU 设计

具体设计如下所示：



注1: 未连线的接口与总线 / 仲裁器的接口相连, 此处不再画出对应的线。

注2：同一元件内黑色字的信号为输入，彩色字的信号为输出。

信号表

具体设计如下所示：

信号名	生成阶段	最后使用阶段	含义
if_pc	IF	EX	当前流水线阶段指令地址
if_pc4	IF	IF	不发生跳转时下一条指令的地址
if_pc_next	IF	IF	下一条指令地址
if_instr	IF	EX	当前流水线阶段指令
id_imm	ID	EX	立即数
id_rs1	ID	EX	读取寄存器编号
id_rs2	ID	EX	读取寄存器编号
id_rdata_a	ID	EX	读取寄存器结果
id_rdata_b	ID	EX	读取寄存器结果
id_rd	ID	WB	写寄存器编号
id_alu_op	ID	EX	ALU 操作：加、减或按指令执行
id_alu_src	ID	EX	ALU 输入选择：寄存器或立即数
id_branch	ID	MEM	是否为跳转型指令
id_mem_read	ID	MEM	是否要读内存
id_mem_write	ID	MEM	是否要写内存
id_mem_sel	ID	MEM	读写内存的位数
id_mem_to_reg	ID	WB	寄存器写入数据选择
id_reg_write	ID	WB	是否要写寄存器
ex_forward_a	EX	EX	ALU 输入选择：寄存器或数据旁路
ex_forward_b	EX	EX	ALU 输入选择：寄存器或数据旁路
ex_alu_op4	EX	EX	ALU 操作
ex_alu_a	EX	EX	ALU 输入
ex_alu_b	EX	EX	ALU 输入
ex_sel_b	EX	EX	ALU 数据：寄存器或数据旁路
ex_alu_out	EX	WB	ALU 输出
ex_pc_sum	EX	MEM	发生跳转时下一条指令的地址
ex_alu_zero	EX	MEM	ALU 输出是否为零
mem_rdata	MEM	WB	内存读出的数据
wb_wdata	WB	WB	写入寄存器堆的数据

内存数据

工作区域

轻轻几点，就能完成原来实体硬件实验板上的复杂操作

THINPAD
教学计算机

32

24

Hex Dump

00000100: ba 13 00 00 |....|

串口输入/输出

done!

清空

发送 请输入数据

换行符 NONE 回车发送数据

接口 txd/rxd (直

波特率 115200

校验位 NONE

数据位 8

停止位 1

串口输出

工作区域

轻轻几点，就能完成原来实体硬件实验板上的复杂操作

THINPAD
教学计算机

32

24

16

8

ARTIX-7

NOR Flash

SRAM x 4

RST CLK

BTN5 BTN4

BTN3 BTN2

BTN1

D16 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1

串口输入/输出

done!

清空

发送 请输入数据，回车发送

换行符 NONE 回车发送数据时添加的换行符

接口 txd/rxd (直连串口信号

波特率 115200

校验位 NONE

数据位 8

停止位 1

思考题

1. 流水线 CPU 中，用于 branch 指令的比较器既可以放在 ID 阶段，也可以放在 EXE 阶段。放在这两个阶段分别有什么优缺点？

- ID 阶段
 - 优点：提早知道分支跳转结果，在分支预测失败时要丢弃的指令数较少；
 - 缺点：单一周期要做的事变多，降低了 CPU 的主频。
- EXE 阶段
 - 优点：可以利用 ALU 的减法作比较，避免因增加比较器所导致的额外计算时间，这种方法不会影响主频；
 - 缺点：分支预测失败时要丢弃的指令数较多。

实验总结

本次实验中，我共提交了两次评测才通过本次实验：)

- 第一次提交时由于忽略了串口部分的输出，只看到指令成功到达 `0x8000008c` 就以为完成了，故出错；
- 吸取了以往实验的教训，本次调试时仔细检查每一时刻的波形，若有误则更改之。
不断重复上述操作直至仿真通过，再上板测试，无误后方提交，最终通过。