



# 架构实验5

## 使用计分板的动态调度流水线

### 任务

- 重新设计具有IF/ID/FU/WB阶段且FU阶段支持多周期的流水线操作。
- 重新设计CPU控制器。

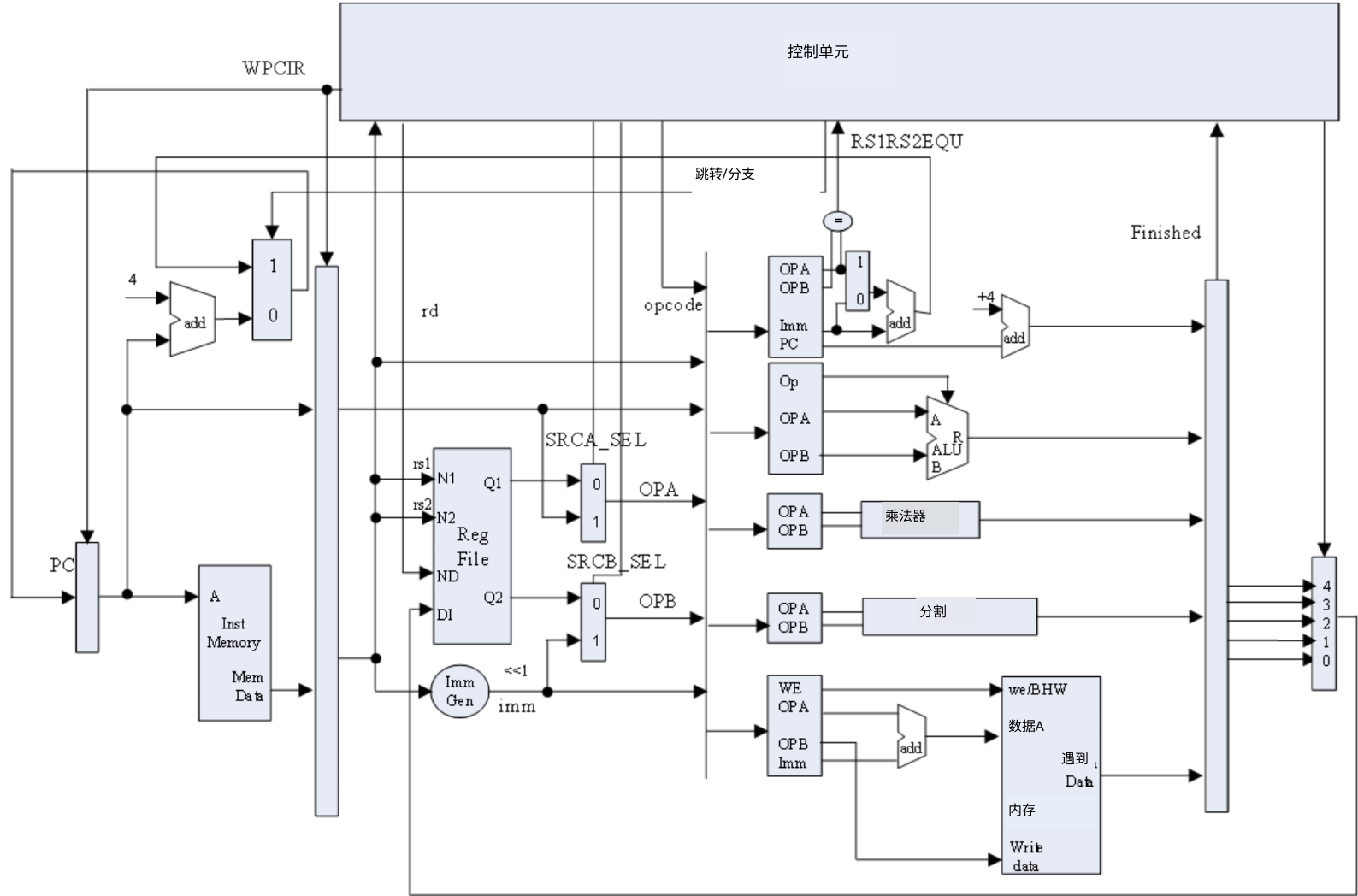
### 概述

- 架构概述
- 控制单元
- 功能单元
- 解决数据冒险的流水线
- 解决控制冒险的流水线

缪晨路 2022年11月

### 架构

### 概述



### 架构概述 - 指令获取阶段

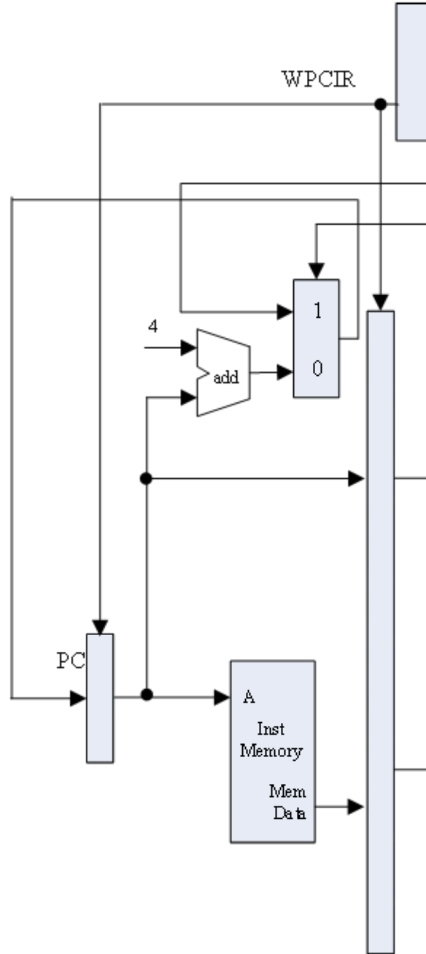
```
// IF
REG32
REG_PC(.clk(调试时钟),.rst(复位信号),.CE(寄存器指令获取阶段使能信号),.D(下一指令获取阶段程序计数器值),.Q(指令获取阶段程序计数器值));
```

32位加法器add\_IF(a(指令获取阶段程序计数器值),.b(32位十进制数4),.c(指令获取阶段程序计数器值加4));

32位二路选择器

选择器IF\_ID(指令获取阶段程序计数器值加4),.I1(功能单元跳转程序计数器值),.s(分支控制信号),.o(下一指令获取阶段程序计数器值));

指令只读存储器inst\_rom(.a(指令获取阶段程序计数器值(8:2)),.spo(指令获取阶段指令));



### 架构概述 - 标识

```
//问题
REG_ID reg_ID(.clk(调试时钟),.rst(复位),.EN(reg_ID使能),
.flush(reg_ID刷新),.PCOUT(程序计数器_IF),.IR(指令_IF),
.IR_ID(指令_ID),.PCurrent_ID(当前程序计数器_ID),.valid(有效_ID));

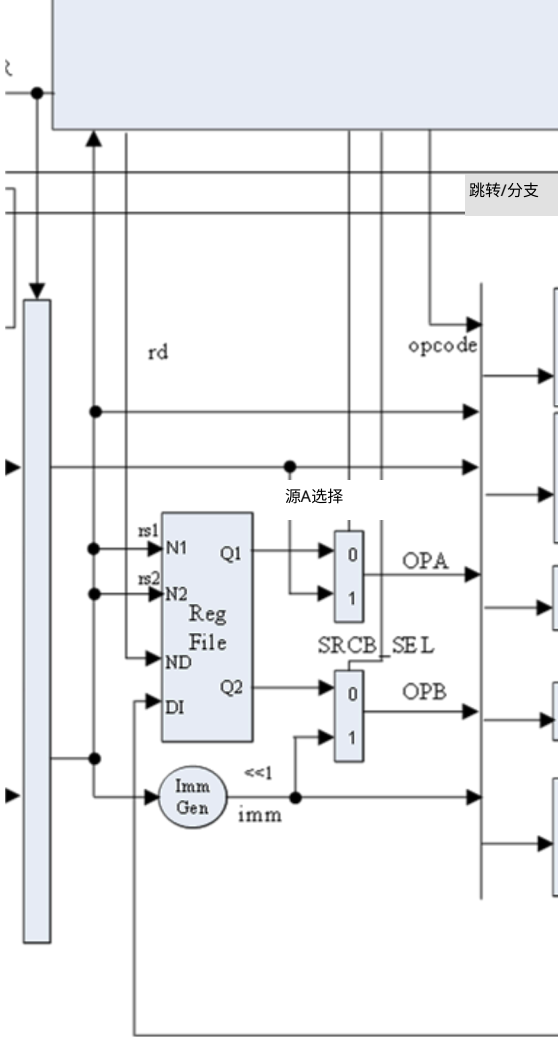
控制单元 控制...

ImmGen模块 (.ImmSel (.ImmSel_ctrl), .Inst_field (.inst_ID), .Imm_out (.Imm_out_ID)) ;

寄存器模块 (.clk (debug_clk), .rst (rst),
.R_addr_A (.inst_ID(19:15)), .rdata_A (rs1_data_ID),
.R_addr_B (.inst_ID(24:20)), .rdata_B (rs2_data_ID)
.L_S (RegWrite_ctrl), .Wt_addr (rd_ctrl), .Wt_data (wt_data_WB),
.Debug_addr (debug_addr(4:0)), .Debug_regs (debug_regs)) ;

二选一32位多路选择器mux_imm_ALU_ID_A (.I0 (rs1_data_ID), .I1 (PC_ID), .s (ALUSrcA_ctrl), .o (ALUA_ID)) ;

二选一32位多路选择器mux_imm_ALU_ID_B (.I0 (rs2_data_ID), .I1 (.Imm_out_ID), .s (ALUSrcB_ctrl), .o (ALUB_ID)) ;
```



### 架构概述 - 功能单元

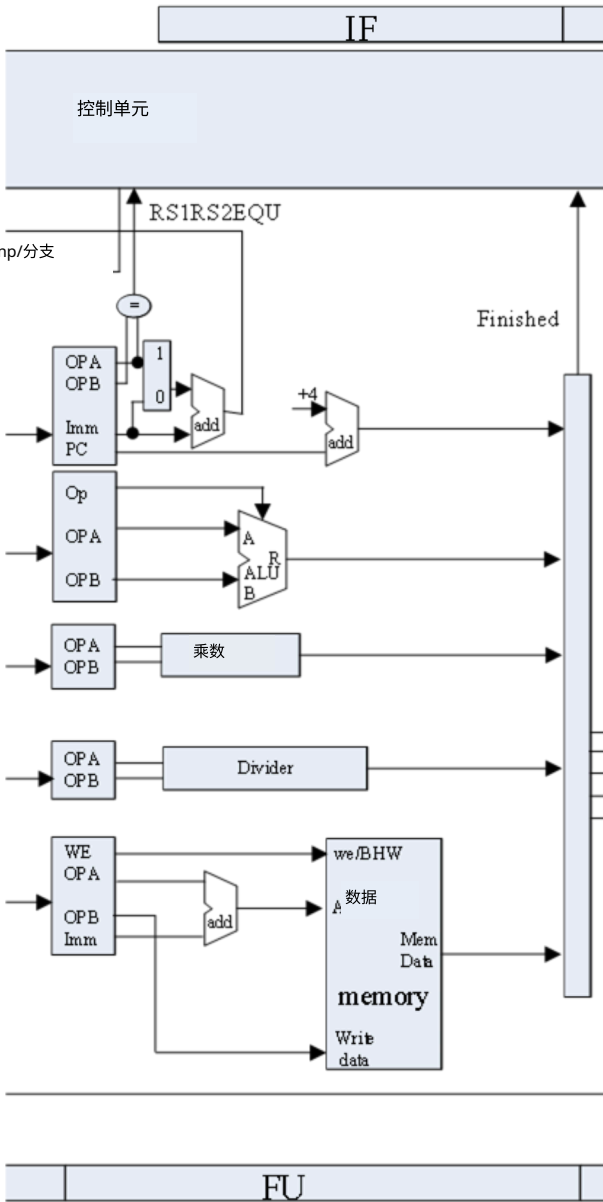
```
// FU
功能单元算术逻辑单元 alu(...);

功能单元内存单元 mem(...);

功能单元乘法器 mul(...);

除法功能单元 div(...);

跳转功能单元 ju(...);
```



### 架构概述 - 写回阶段

```
// WB
32位寄存器reg_WB_ALU (.clk (debug_clk), .rst (rst), .CE (FU_ALU_finish), .D (ALUout_FU), .Q (ALUout_WB)) ;

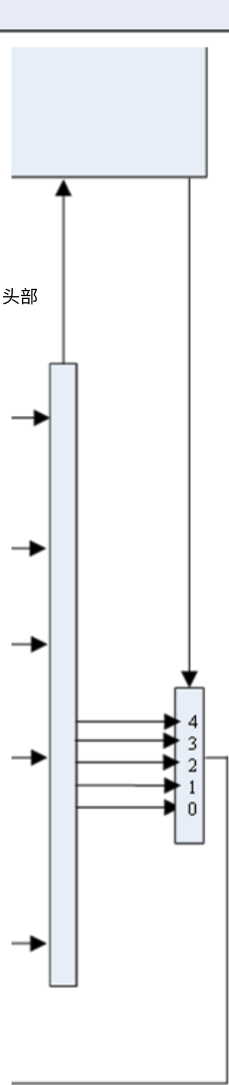
REG32
寄存器reg_WB_mem (.clk (debug_clk), .rst (rst), .CE (FU_mem_finish), .D (mem_data_FU), .Q (mem_data_WB)) ;

32位寄存器reg_WB_mul (.clk (调试时钟), .rst (复位信号), .CE (乘法功能单元完成信号), .D (乘法功能单元结果), .Q (写回阶段乘法结果)) ;

32位寄存器reg_WB_div (.clk (调试时钟), .rst (复位信号), .CE (除法功能单元完成信号), .D (除法功能单元结果), .Q (写回阶段除法结果)) ;

32位寄存器reg_WB_jump (.clk (调试时钟), .rst (复位信号), .CE (跳转功能单元完成信号), .D (功能单元写回阶段程序计数器值), .Q (写回阶段程序计数器值)) ;

8选1 32位多路选择器
数据到寄存器多路选择器mux_DtR (.s (数据到寄存器控制信号), .I0 (32位常数0), .I1 (算术逻辑单元写回阶段输出), .I2 (存储器数据写回阶段值), .I3 (写回阶段乘法结果),
.I4 (写回阶段除法结果), .I5 (写回阶段程序计数器值), .I6 (32位常数0), .I7 (32位常数0), .o (写回阶段写入数据)) ;
```





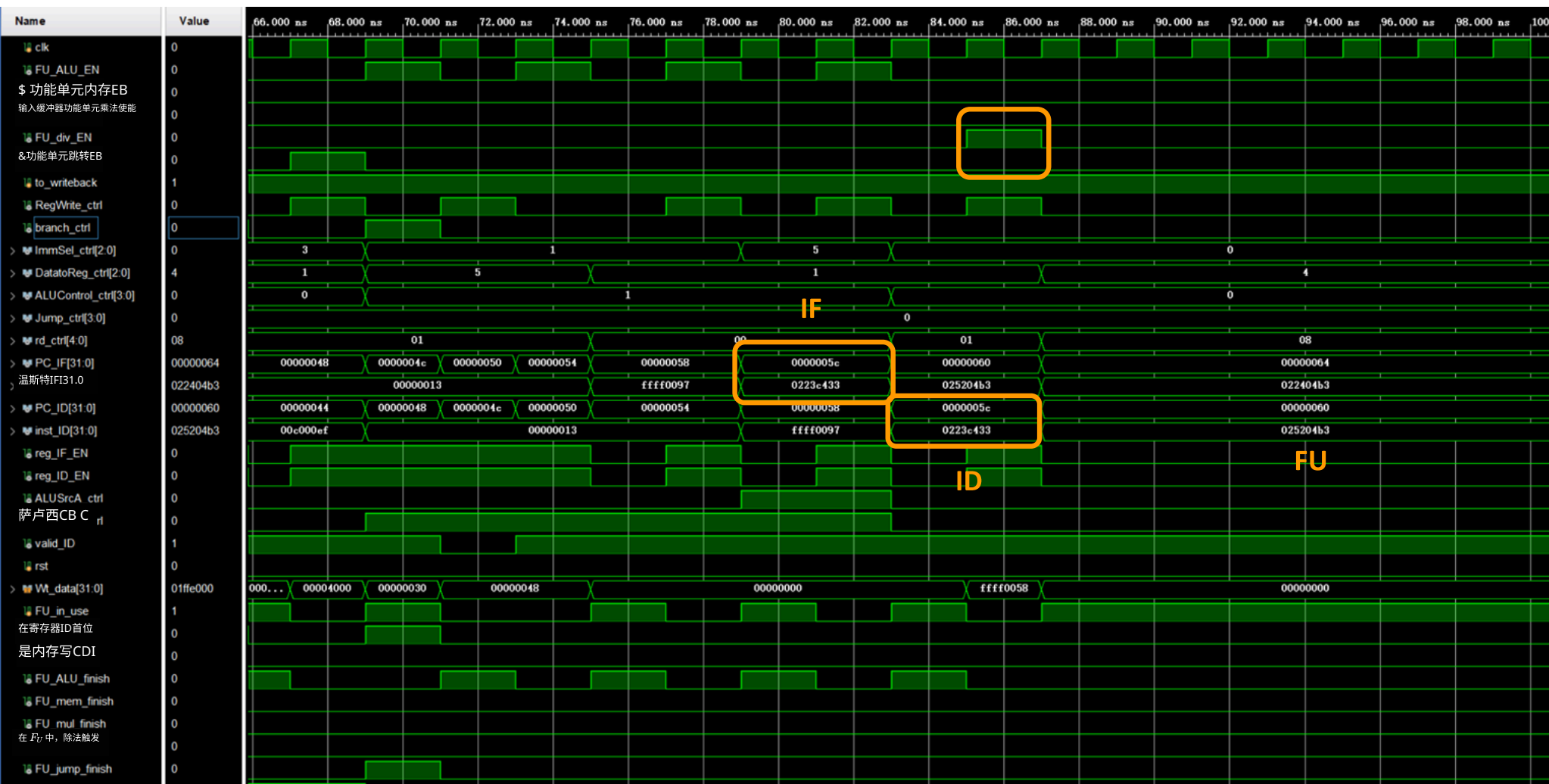


## RAM

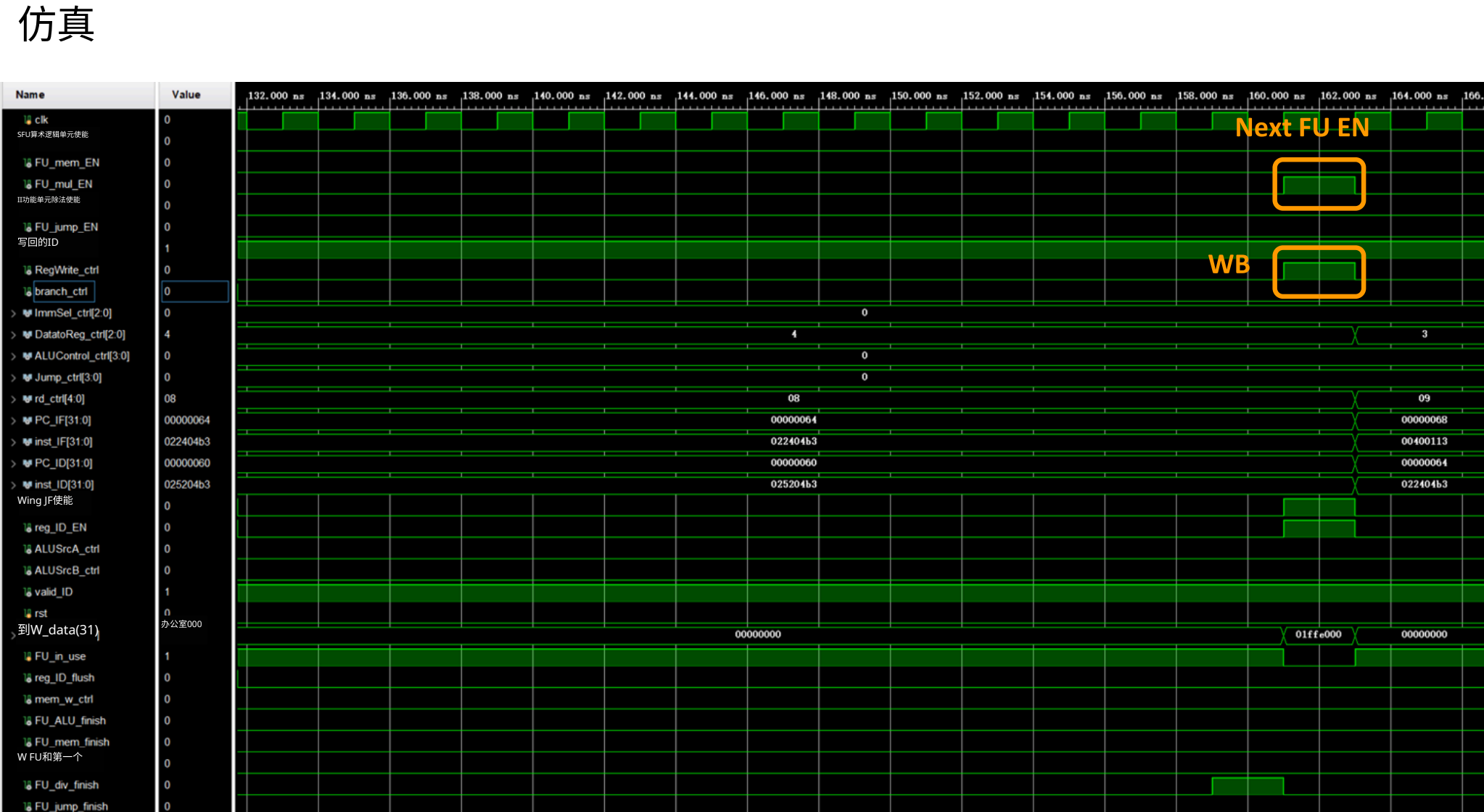
NO.	指令	地址	标签	ASM	注释
15	00000013	3C		addi x0,x0,0	
16	000040b7	40	标签0:	lui x1,4	
17	00c000ef	44		跳转到地址12并链接到x1寄存器	
18	00000013	48		将x0寄存器的值加0后存回x0寄存器	
19	00000013	4C		将x0寄存器的值加0后存回x0寄存器	
20	00000013	50		将x0寄存器的值加0后存回x0寄存器	
21	00000013	54		将x0寄存器的值加0后存回x0寄存器	
22	ffff0097	58		将0xffff0与程序计数器相加并将结果存入x1寄存器	
23	0223c433	5C		将x7寄存器的值除以x2寄存器的值，结果存入x8寄存器	
24	025204b3	60		将x4寄存器的值乘以x5寄存器的值，结果存入x9寄存器	
25	022404b3	64		将x8寄存器的值乘以x2寄存器的值，结果存入x9寄存器	
26	00400113	68		将立即数4加到x0寄存器的值上，结果存入x2寄存器	
27	000000e7	6C		跳转到x0寄存器的值加上偏移量的地址，并将返回地址存入x1寄存器	

NO.	指令	地址
16	00000000	40
17	00000000	44
18	00000000	48
19	00000000	4C
20	A3000000	50
21	27000000	54
22	79000000	58
23	15100000	5C
24	00000000	60
25	00000000	64
26	00000000	68
27	00000000	6C
28	00000000	70
29	00000000	74
30	00000000	78
31	00000000	7C

## 仿真



## 仿真





## 模拟

## 模拟

