浙江大学
ZHEJIANG UNIVERSITY

Computer Architecture Experiment

Topic 1. Multiple-cycle CPU Design

浙江大学计算机学院
陈文智、王总辉
{chenwz, zhwang}@zju.edu.cn

Outline

- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures
- Precaution
- Checkpoints

Experiment Purpose

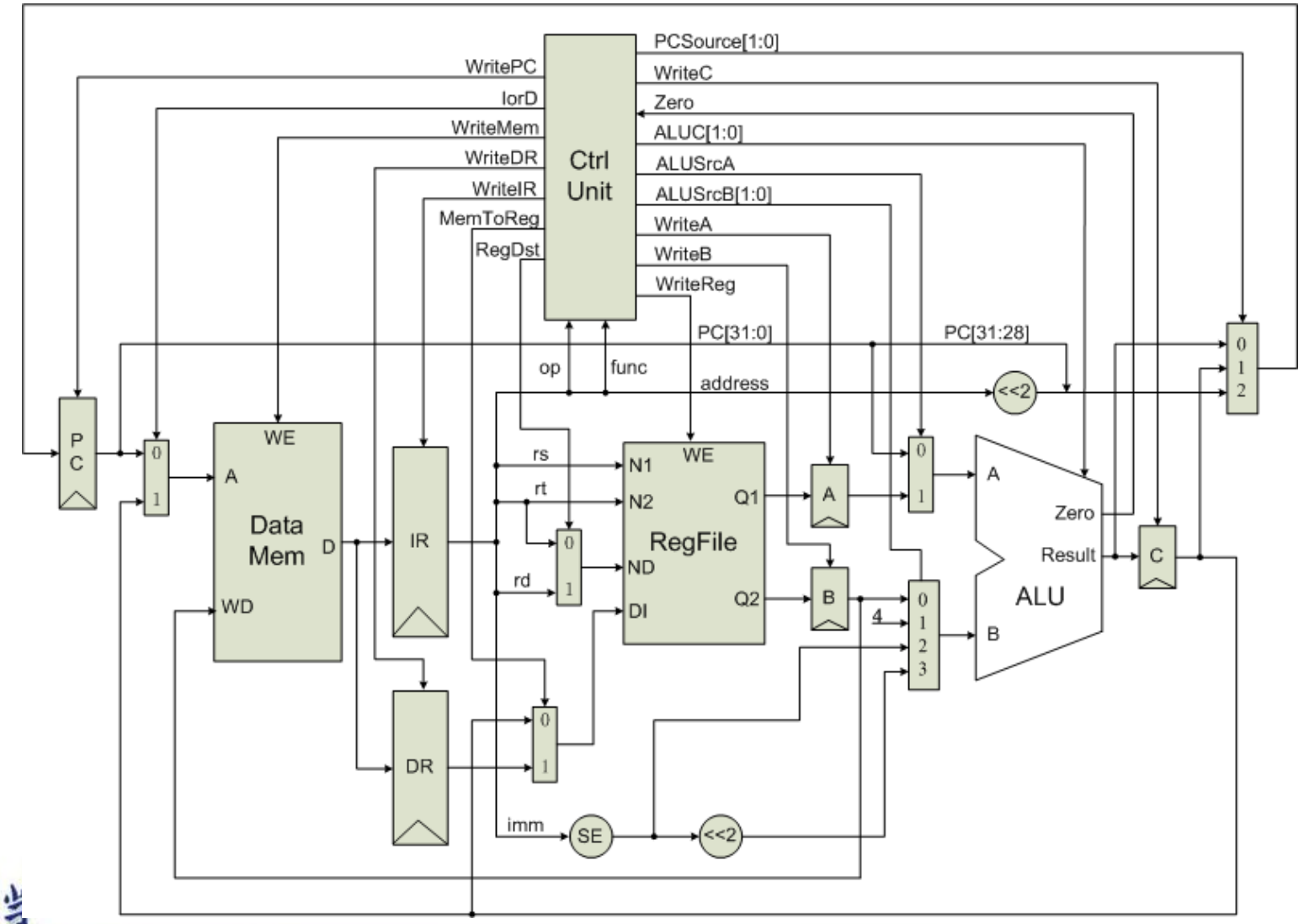
- **Understand the principles of MC CPU Controller and master methods of MC CPU Controller design**
- **Understand the principles of Datapath and master methods of Datapath design**
- **Understand the principles of MC CPU and master methods of MC CPU design**
- **master methods of program verification of CPU**

Experiment Task

- **Design the CPU Controller, Datapath, bring together the basic units into Multiple-cycle CPU**

- **Verify the MC CPU with program and observe the execution of program**

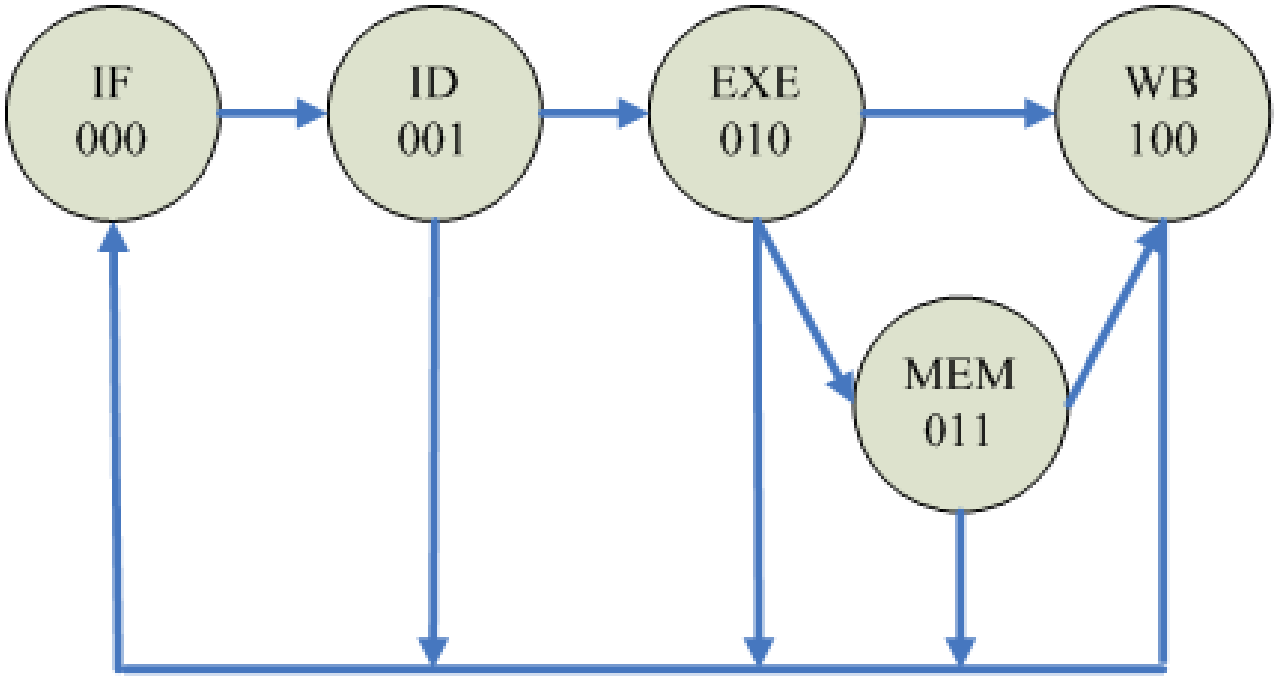
CPU Controller



Output of CPU Controller

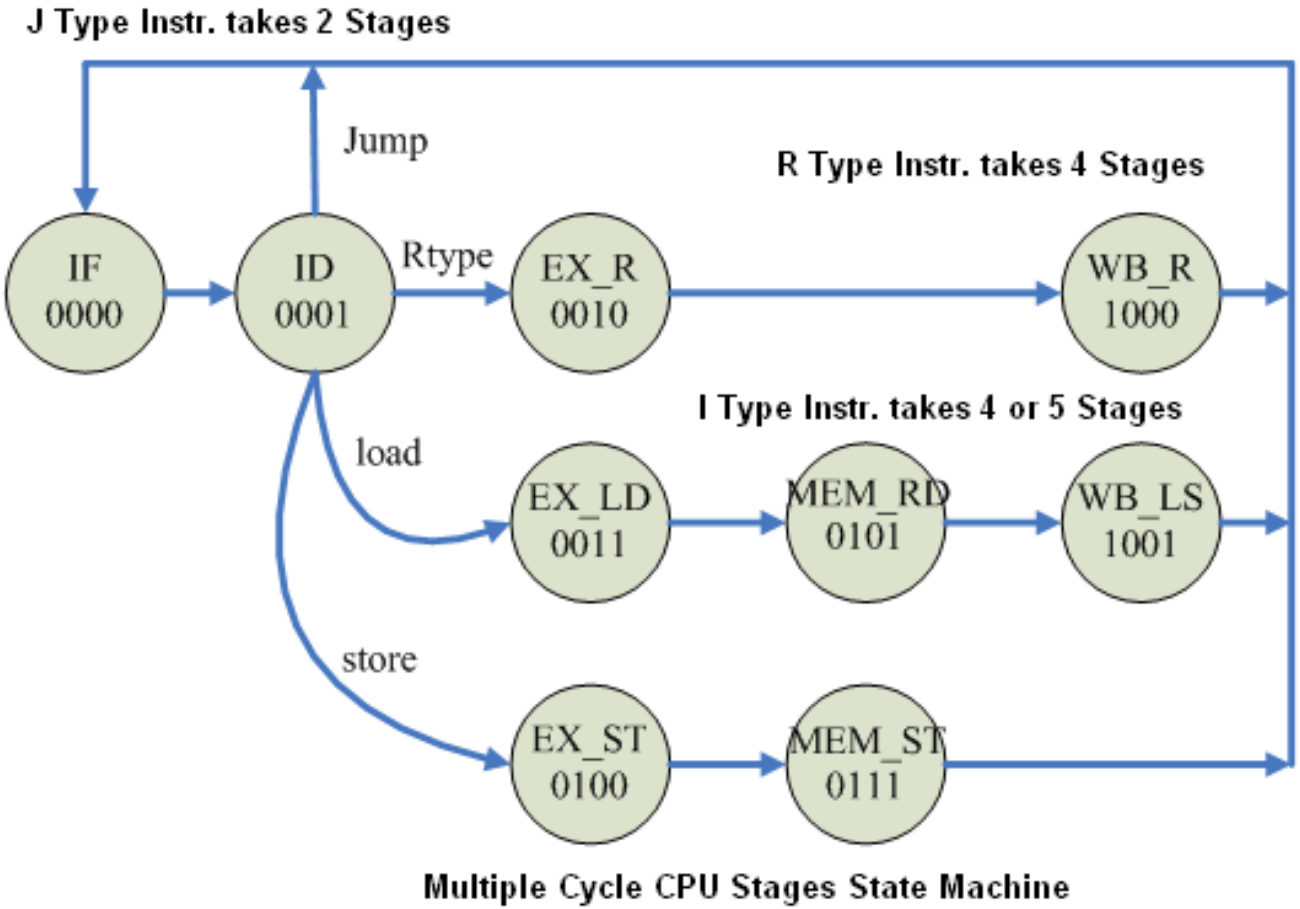
	Output Signal	Meaning When 1	Meaning When 0
1	PCSrc[1:0]	00: PC + 4;01: Branch Instr.;10: jump Instr	
2	WritePC	Write PC	Not Write PC
3	IorD	Instruction Addr	Data Addr.
4	WriteMem	Write Mem.	Not Write Mem.
5	Write DR	Write Data. Reg	Not Write Data. Reg
6	Write IR	Write Instr. Reg	Not Write Instr. Reg
7	MemToReg	From Mem. To Reg	From ALUOut To Reg
8	RegDest	rd	rt
9	ALUC	ALU Controller Op	
10	ALUSrcA	Register rs	PC
11	ALUSrcB	Selection:00:Reg rt; 01:4; 10:Imm.; 11: branch Address	
12	WriteA	Write A Reg.	Not Write A Reg.
13	WriteB	Write B Reg.	Not Write B Reg.
14	WriteC	Write C Reg.	Not Write C Reg.
15	WriteReg	Write Reg.	Not Write Reg.

The principle of CPU Controller(1)



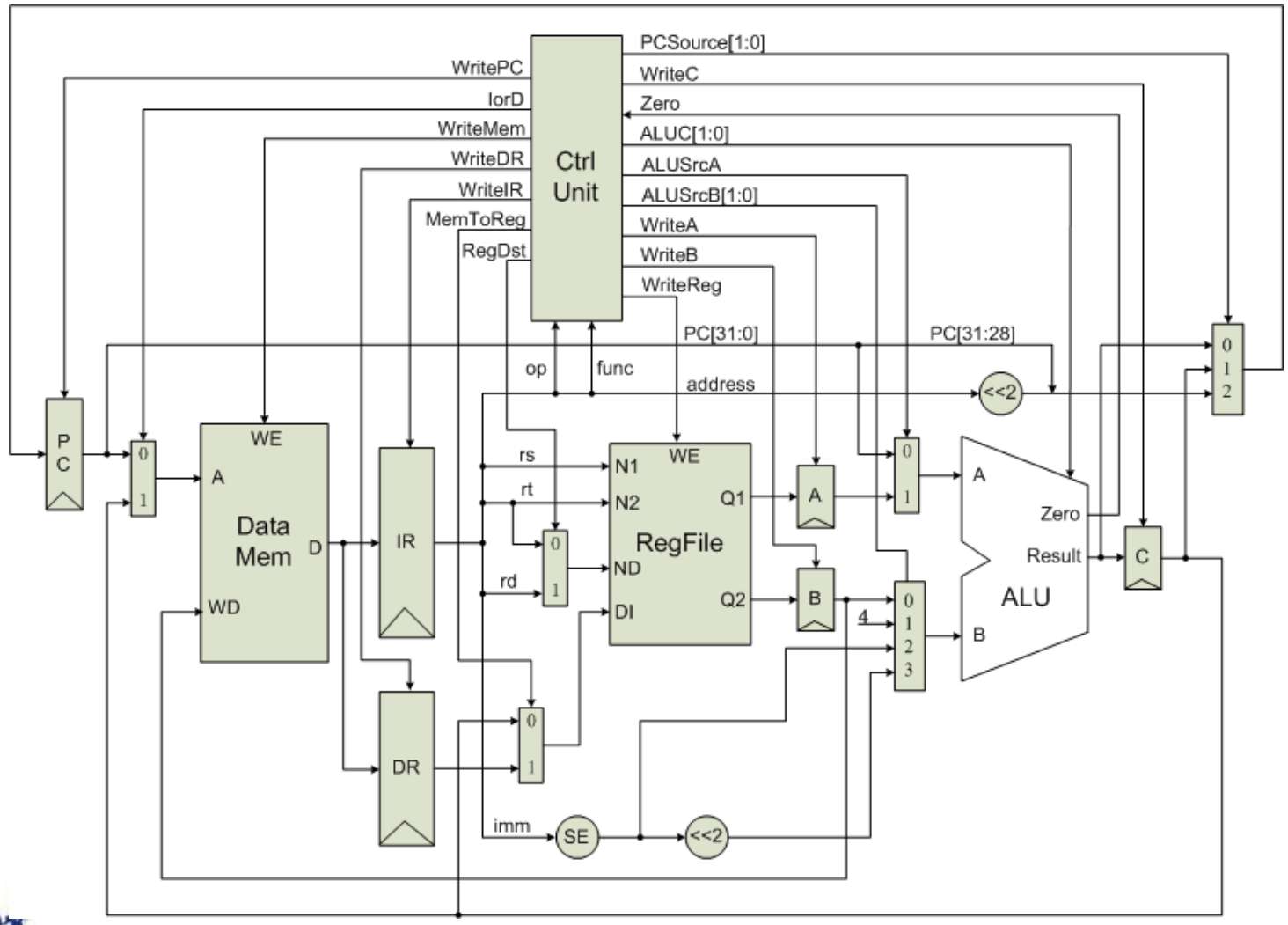
Stages of Multiple-Cycle Execution of Typical MIPS CPU

The principle of CPU Controller(2)



Multiple Cycle CPU Stages State Machine

The Datapath of Multiple-cycle CPU



Basic Units of Multiple-cycle CPU

- **CPU Controller**
- **ALU and ALU Controller**
- **Register file**
- **Mem. (Instruction and Data together).**
- **others： Register, sign-extend Unit, shifter, multiplexor**

Memory

- **Memory**
  - Dual Port Block Memory
  - Port A: Read Only, Width: 32, Depth: 512
  - Port B: Read and Write, Read After Write
  - Rising Edge Triggered

Multiple-cycle CPU Top Module

memory x_memory(.addra(raddr),.addrb(waddr),.clka(clk), .clkb(clk),.dinb(b_data),.douta(mem_data),.web(write_mem));

ctrl x_ctrl(clk, rst, ir_data, zero,write_pc, iord, write_mem, write_dr, write_ir, memtoreg, regdst, pcsource, write_c, alu_ctrl, alu_srcA, alu_srcB, write_a, write_b, write_reg, state_out, insn_type, insn_code, insn_stage);

pcm x_pcm(clk, rst, alu_out, c_data, ir_data, pcsource, write_pc,pc);

alu_wrapper x_alu_wrapper(a_data, b_data, ir_data, pc, alu_srcA, alu_srcB, alu_ctrl, zero, alu_out);

reg_wrapper x_reg_wrapper(clk, rst, ir_data, dr_data, c_data, memtoreg, regdst, write_reg, rdata_A, rdata_B, r6out);

Observation Info

- **Input**
  - West Button: Step execute
  - South Button: Reset
  - Slide Button: Address of Register
  - East Button: Switch to High 16-bit of Register
- **Output**
  - 0-7 Character of First line: Instruction Code
  - 8 of First line : Space
  - 9-10 of First line : Read Address
  - 11 of First line : Space
  - 12-13 of First line : Write Address
  - 0/2/4/6 of Second line : state/type/code/stage
  - 8-9 of Second line : PC
  - 11-14 of Second line: Selected Register Content

Program for verification

<0> lw r1, $20(r0); 0x8c01_0014 State:0,1,3,5,9 Type:3 Code:1 (LD)

<1> lw r2, $21(r0); 0x8c02_0015 State:0,1,3,5,9 Type:3 Code:1 (LD)

<2> add r3, r1, r2; 0x0022_1820 State:0,1,2,8 Type:1 Code:3 (AD)

<3> sub r4, r1, r2; 0x0022_2022 State:0,1,2,8 Type:1 Code:4 (SU)

<4> and r5, r3, r4; 0x0064_2824 State:0,1,2,8 Type:1 Code:5 (AN)

<5> nor r6, r4, r5; 0x0085_3027 State:0,1,2,8 Type:1 Code: 6 (NO)

<6> sw r6, $22(r0); 0x ac06_0016 State:0,1,4,7 Type:3 Code: 2 (ST)

<7> J 0; 0x0800_0000 State:0,1 Type:2 Code:7 (JP)

DataMem(20) = 0xbeef_0000 ;

DataMem(21) =0x0000_beef ;

Precaution

- **1. Add Anti-Jitter**
- **2. Finish the State Machine**
- **3. Add Stage Status**
- **4. Selected Register Content**
- **5. Display “A-F” correctly**

Checkpoints

- **CP 1: Waveform Simulation of Multiple-cycle CPU ALU**
- **CP 2: Waveform Simulation of Multiple-cycle CPU Controller**
- **CP 3 (Optional): Waveform Simulation of Multiple-cycle CPU with the verification program**
- **CP 4: FPGA Implementation of Multiple-cycle CPU with the verification program**

Thanks!

浙江大学
ZheJiang University