处理器设计

4. 确定控制信号

主讲人: 邓倩妮
上海交通大学

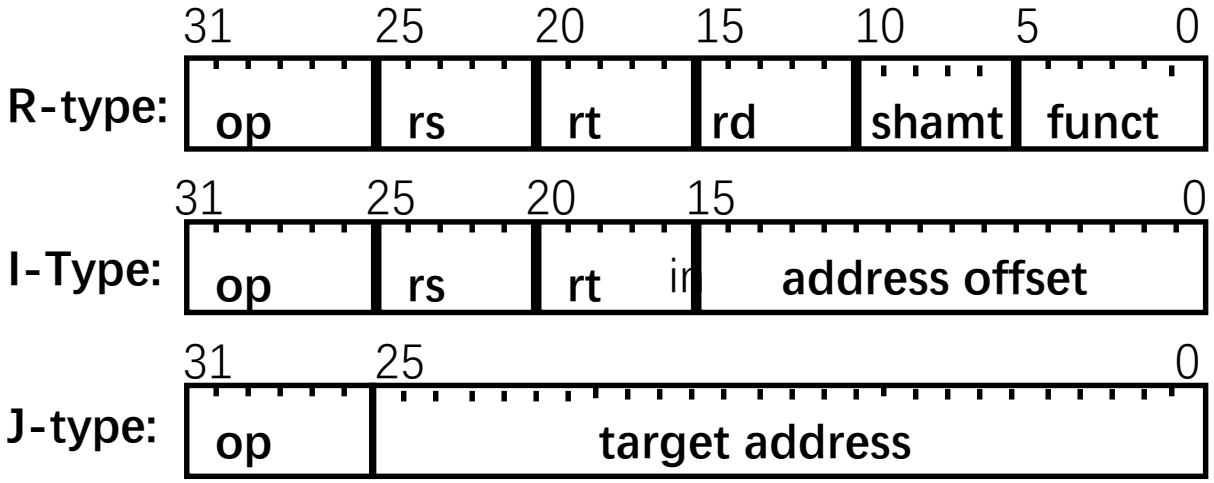
大部分内容来自于：
Computer Organization and Design, 4th Edition, Patterson & Hennessy,

设计处理器的五个步骤

1. 分析指令系统，得出对数据通路的需求
   - The meaning of each instruction is given by the register transfers
   - 例如：ADDU指令的数据通路需求：R[rd] <– R[rs] + R[rt];
2. 选择数据通路上合适的组件
   - 例如：加法器？ 算术逻辑运算单元？寄存器堆？
3. 连接组件构成数据通路
4. 分析每一条指令的实现，以确定控制信号，
   - 不同的控制信号影响寄存器之间的数据传送
5. 集成控制信号，完成控制逻辑

Adding the Control

- Selecting the operations to perform (ALU, Register File and Memory read/write)
- Controlling the flow of data (multiplexor inputs)



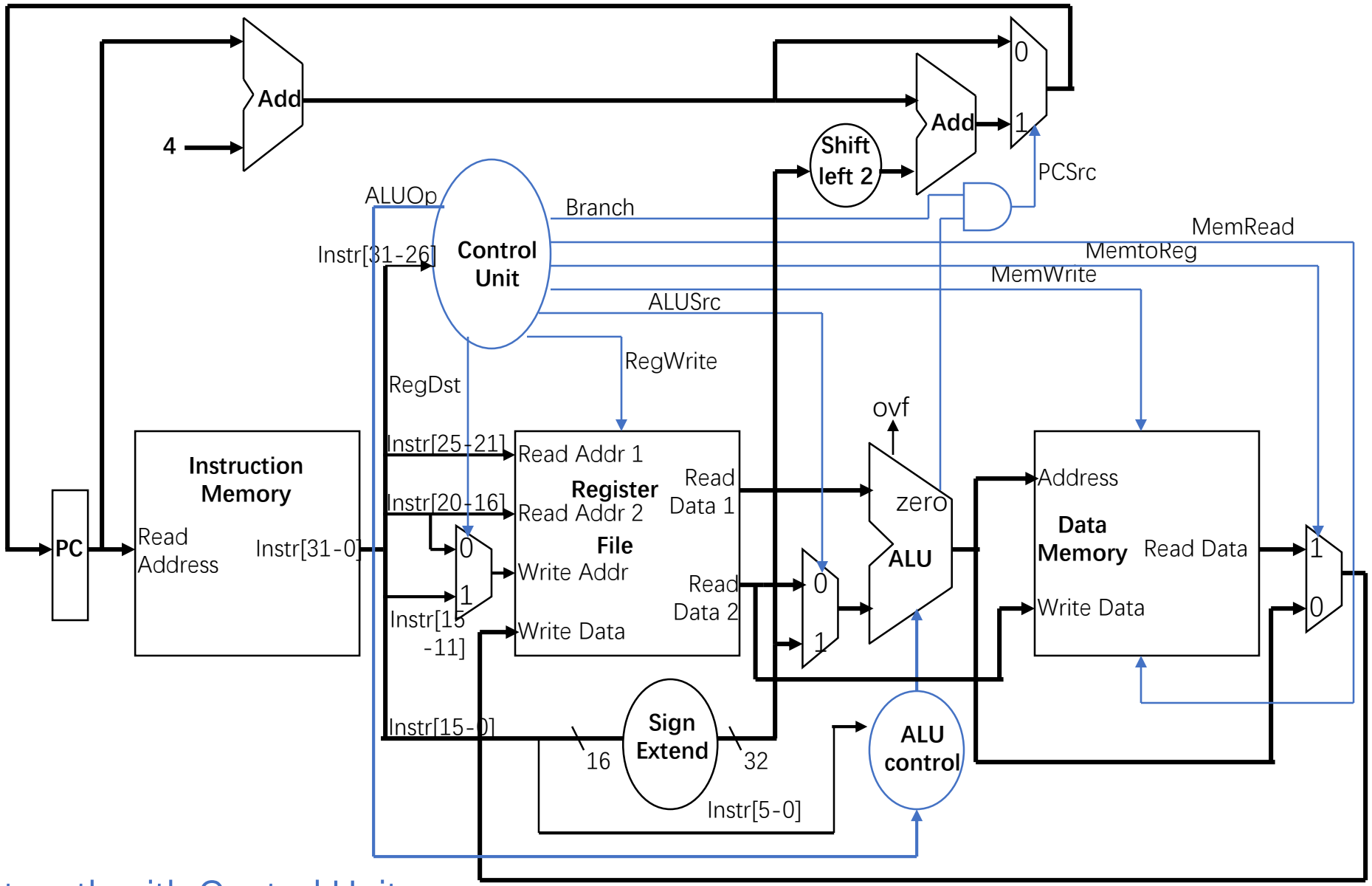
- Observations

  op field always bits 31-26

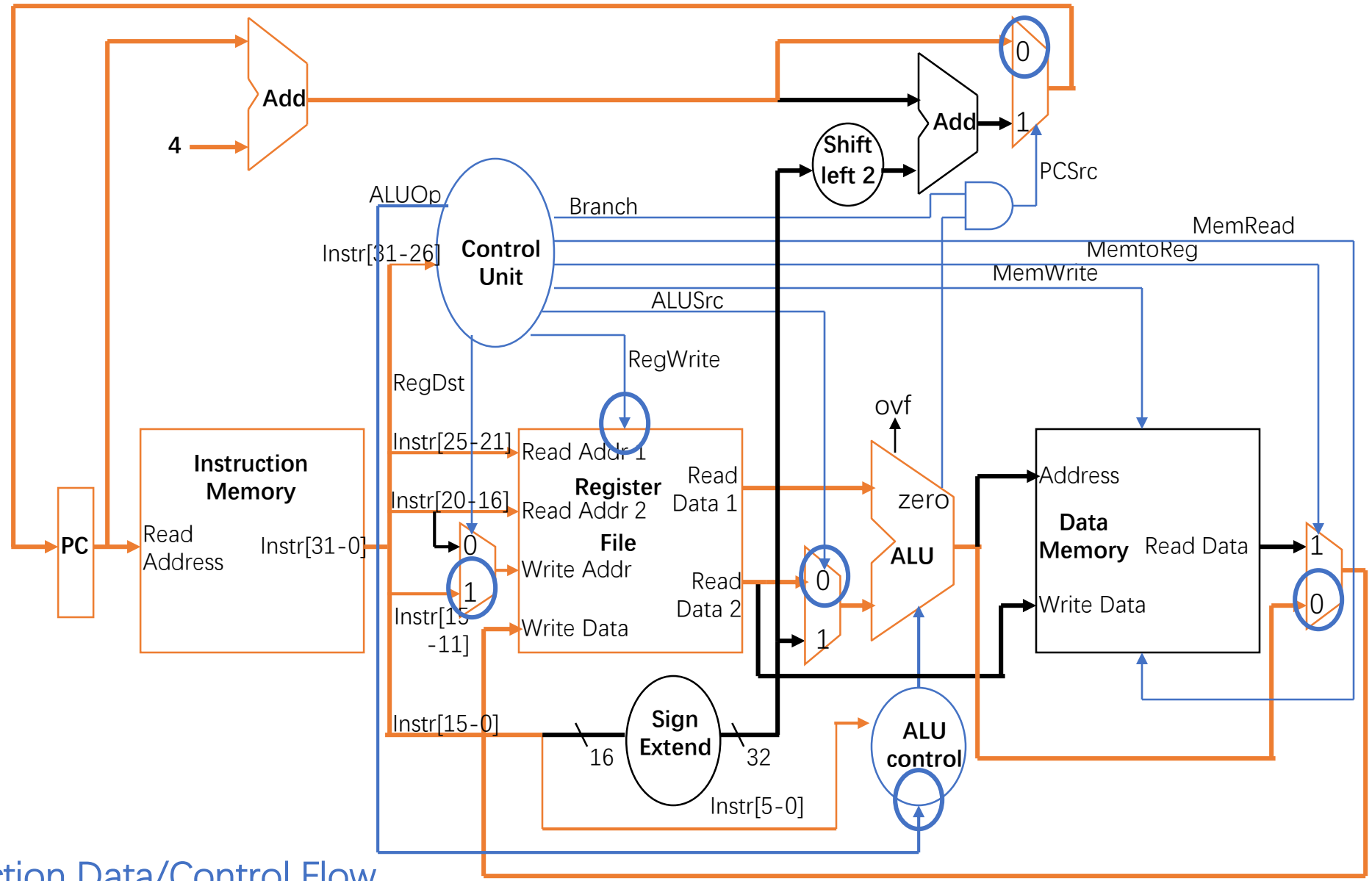
  addr of registers to be read are always specified by the rs field (bits 25-21) and rt field (bits 20-16); for lw and sw rs is the base register

  addr. of register to be written is in one of two places – in rt (bits 20-16) for lw; in rd (bits 15-11) for R-type instructions

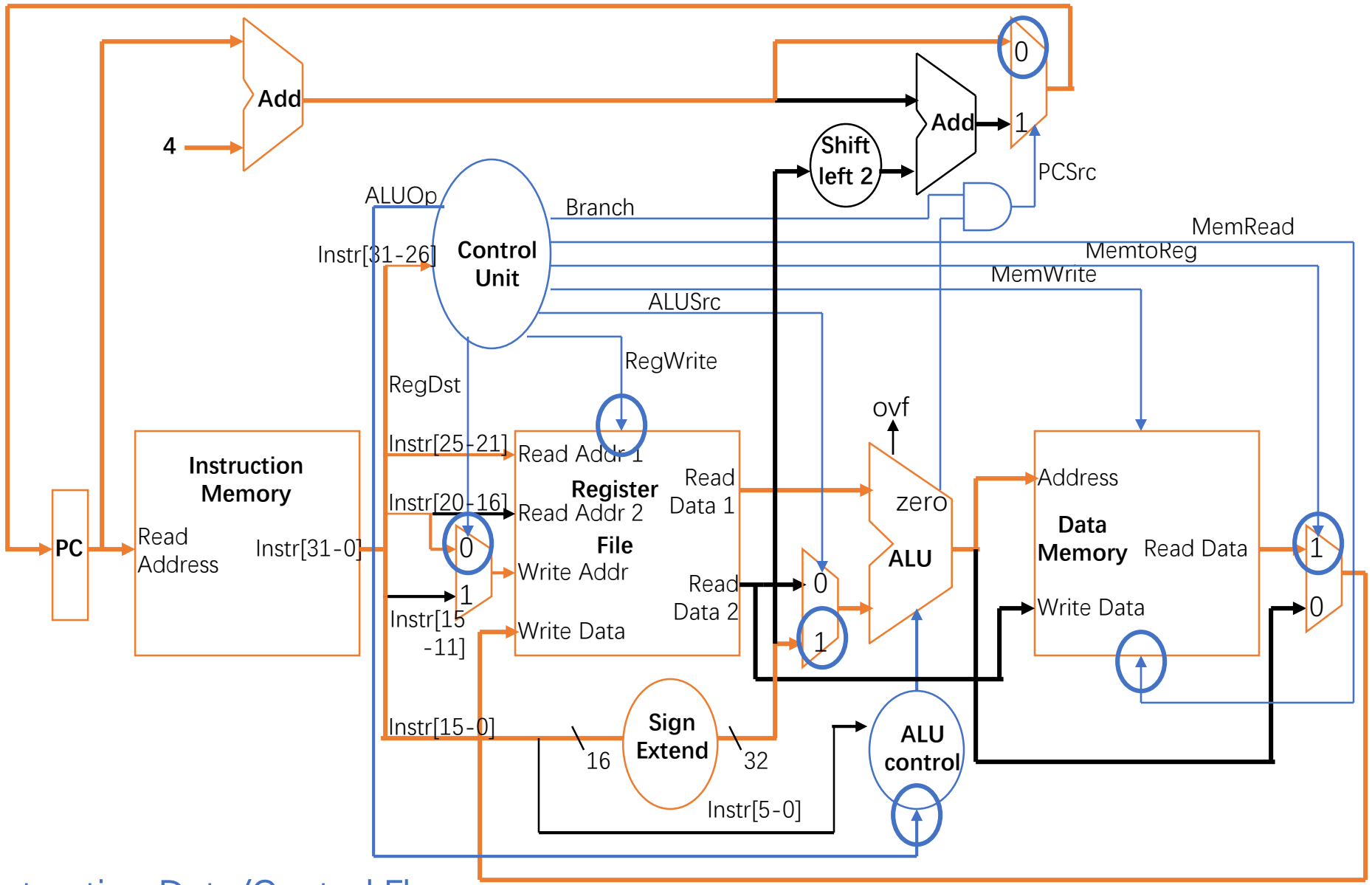
  offset for beq, lw, and sw always in bits 15-0



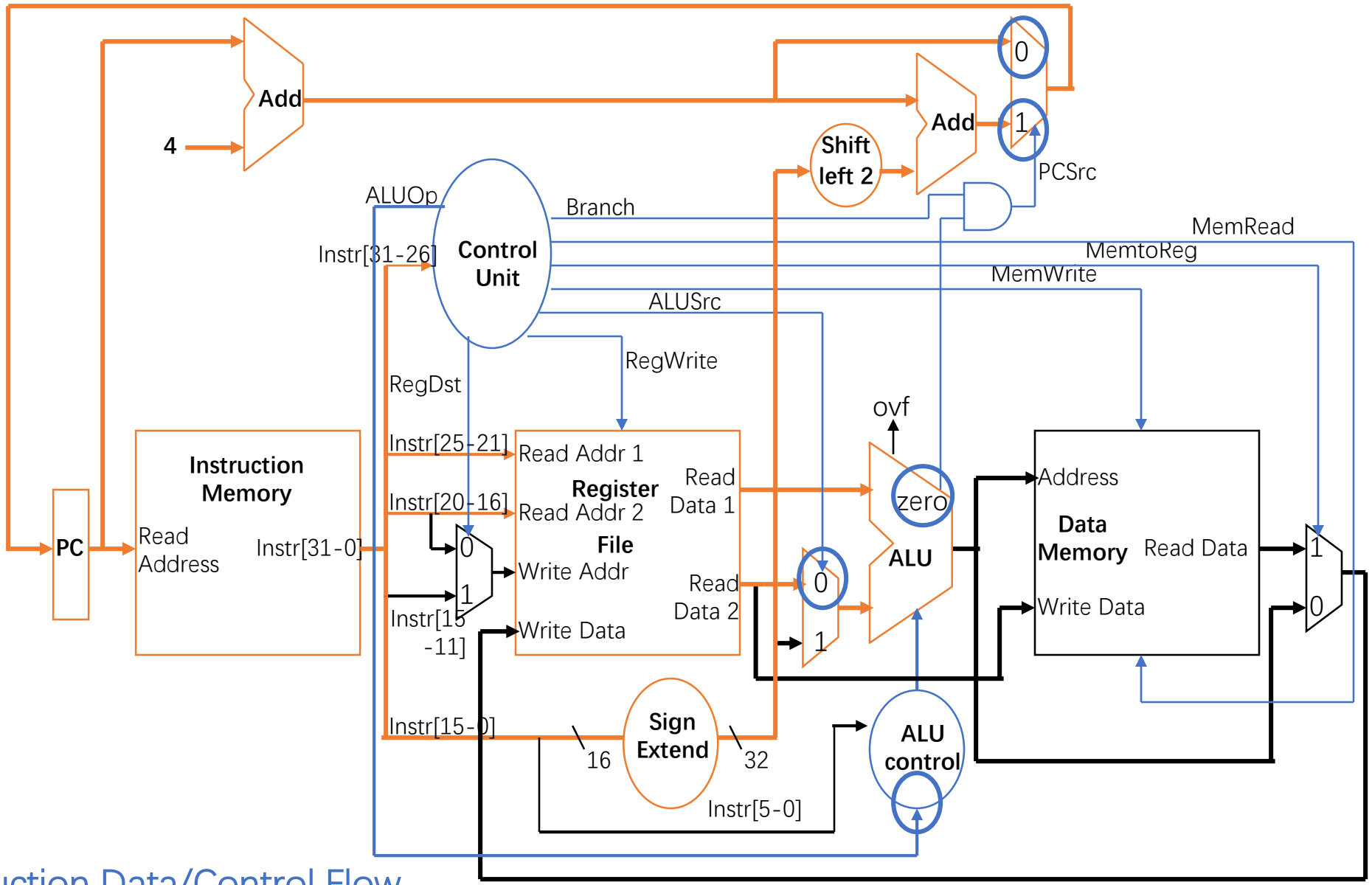
Single Cycle Datapath with Control Unit



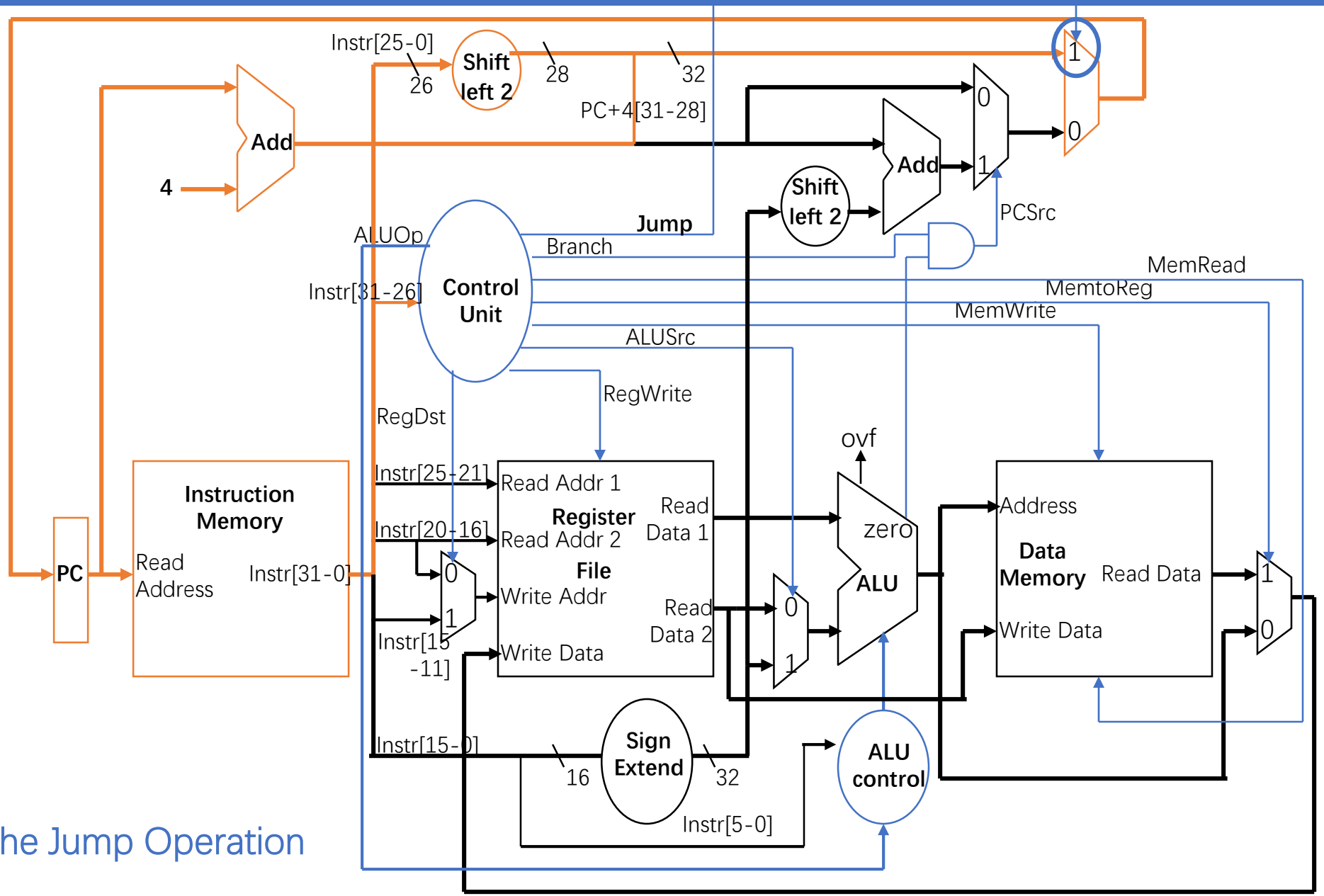
R-type Instruction Data/Control Flow



Load Word Instruction Data/Control Flow



Branch Instruction Data/Control Flow



Adding the Jump Operation

A Summary of Control Signals

inst	Register Transfer	
ADD	R[rd] <– R[rs] + R[rt];	PC <– PC + 4
	ALUsrc = RegB, ALUctr = “add”, RegDst = rd, RegWr, nPC_sel = “+4”	
SUB	R[rd] <– R[rs] – R[rt];	PC <– PC + 4
	ALUsrc = RegB, ALUctr = “sub”, RegDst = rd, RegWr, nPC_sel = “+4”	
ORi	R[rt] <– R[rs] + zero_ext(Imm16);	PC <– PC + 4
	ALUsrc = Im, Extop = “Z”, ALUctr = “or”, RegDst = rt, RegWr, nPC_sel = “+4”	
LOAD	R[rt] <– MEM[R[rs] + sign_ext(Imm16)];	PC <– PC + 4
	ALUsrc = Im, Extop = “Sn”, ALUctr = “add”, MemtoReg, RegDst = rt, RegWr, nPC_sel = “+4”	
STORE	MEM[R[rs] + sign_ext(Imm16)] <– R[rs];	PC <– PC + 4
	ALUsrc = Im, Extop = “Sn”, ALUctr = “add”, MemWr, nPC_sel = “+4”	
BEQ	if (R[rs] == R[rt]) then PC <– PC + sign_ext(Imm16)] || 00 else PC <– PC + 4	
	nPC_sel = “Br”, ALUctr = “sub”	

小结：设计处理器的五个步骤

1. 分析指令系统，得出对数据通路的需求
   - The meaning of each instruction is given by the register transfers
   - 例如：ADDU指令的数据通路需求：R[rd] <– R[rs] + R[rt];
2. 选择数据通路上合适的组件
   - 例如：加法器？ 算术逻辑运算单元？寄存器堆？
3. 连接组件构成数据通路
4. 分析每一条指令的实现，以确定控制信号，
   - 不同的控制信号影响寄存器之间的数据传送
5. 集成控制信号，完成控制逻辑