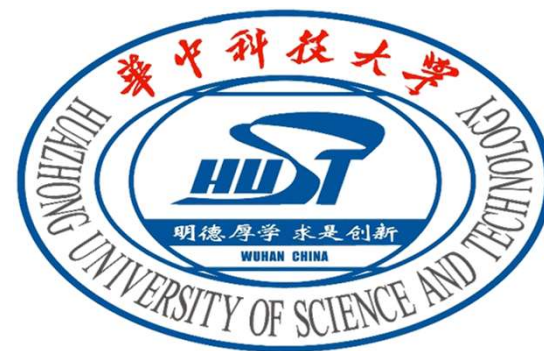


微机原理与接口技术

微处理器基本构成

华中科技大学 左冬红



回顾

微处理器指令类型

数据运算类

算术运算

逻辑运算

完成计算

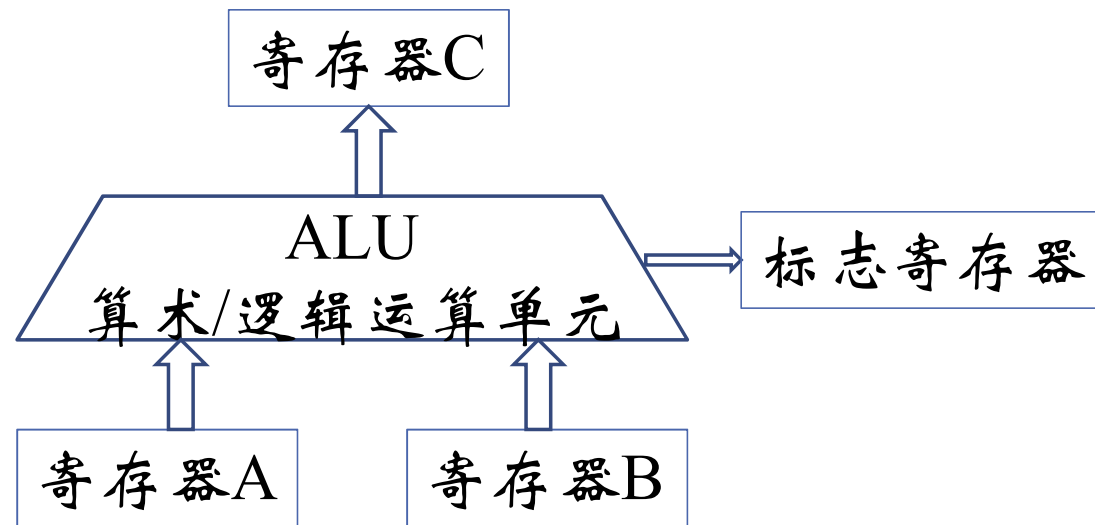
数据传输类

数据输入、输出

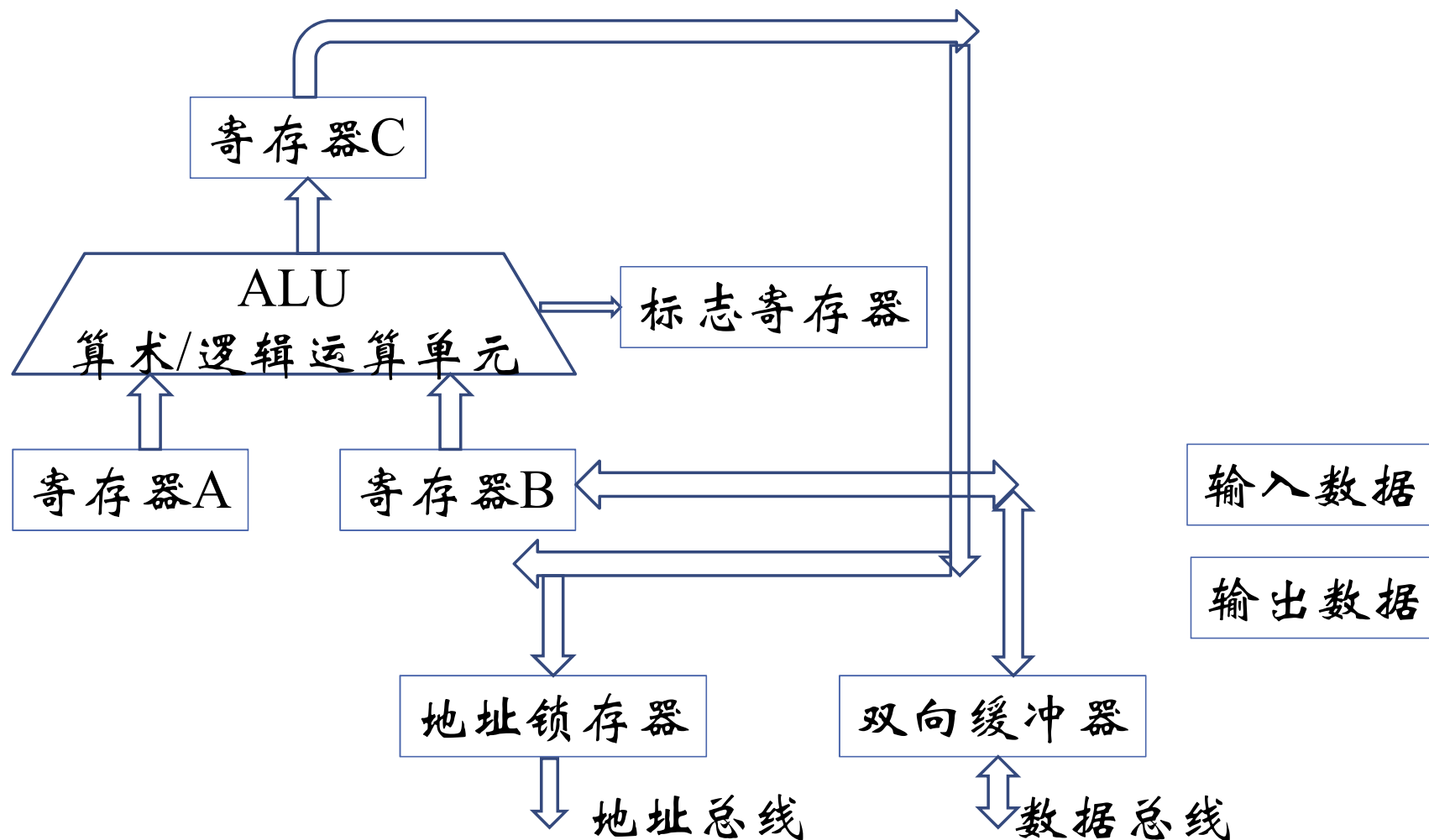
程序控制类

控制程序执行流程

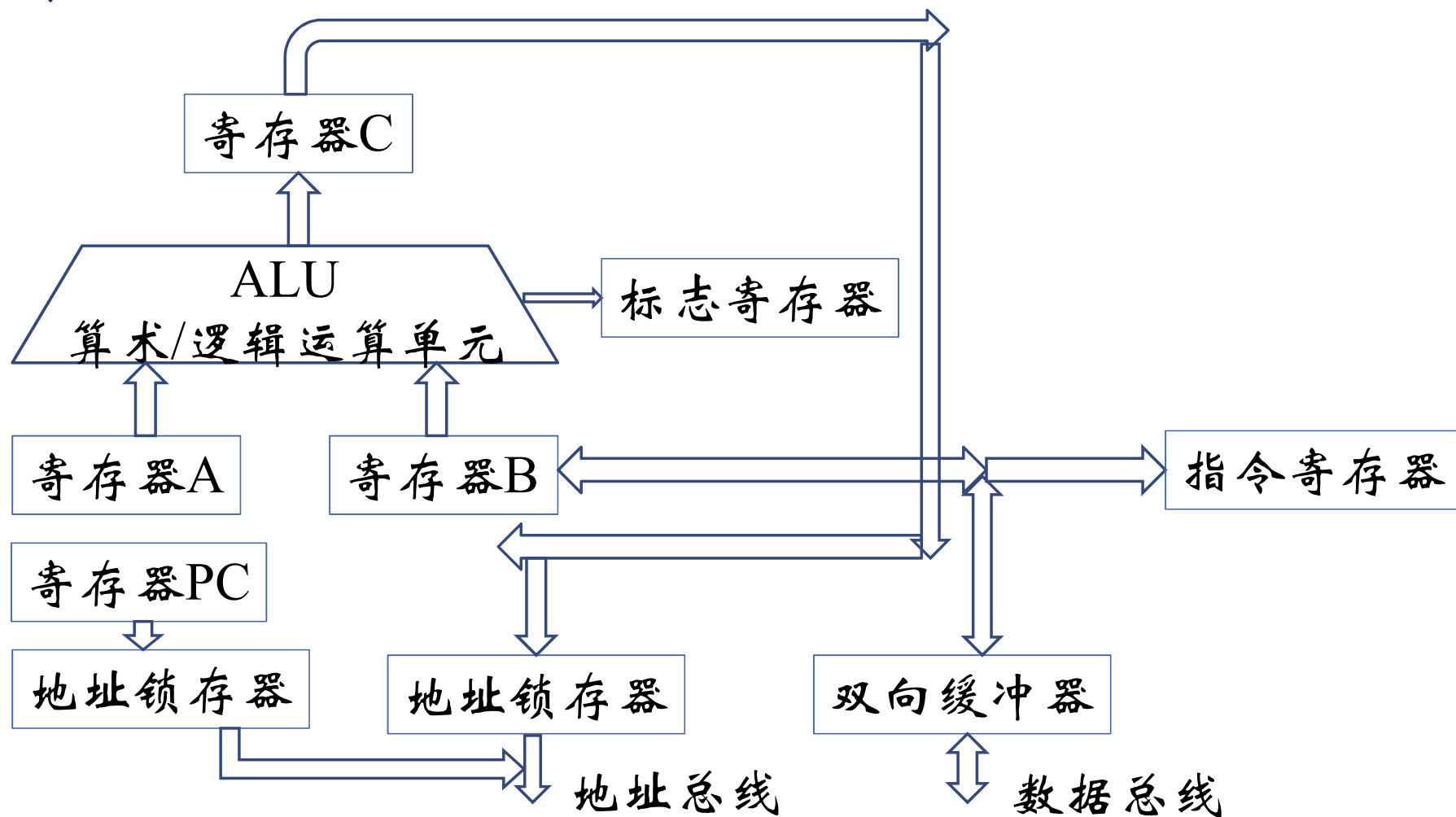
数据运算



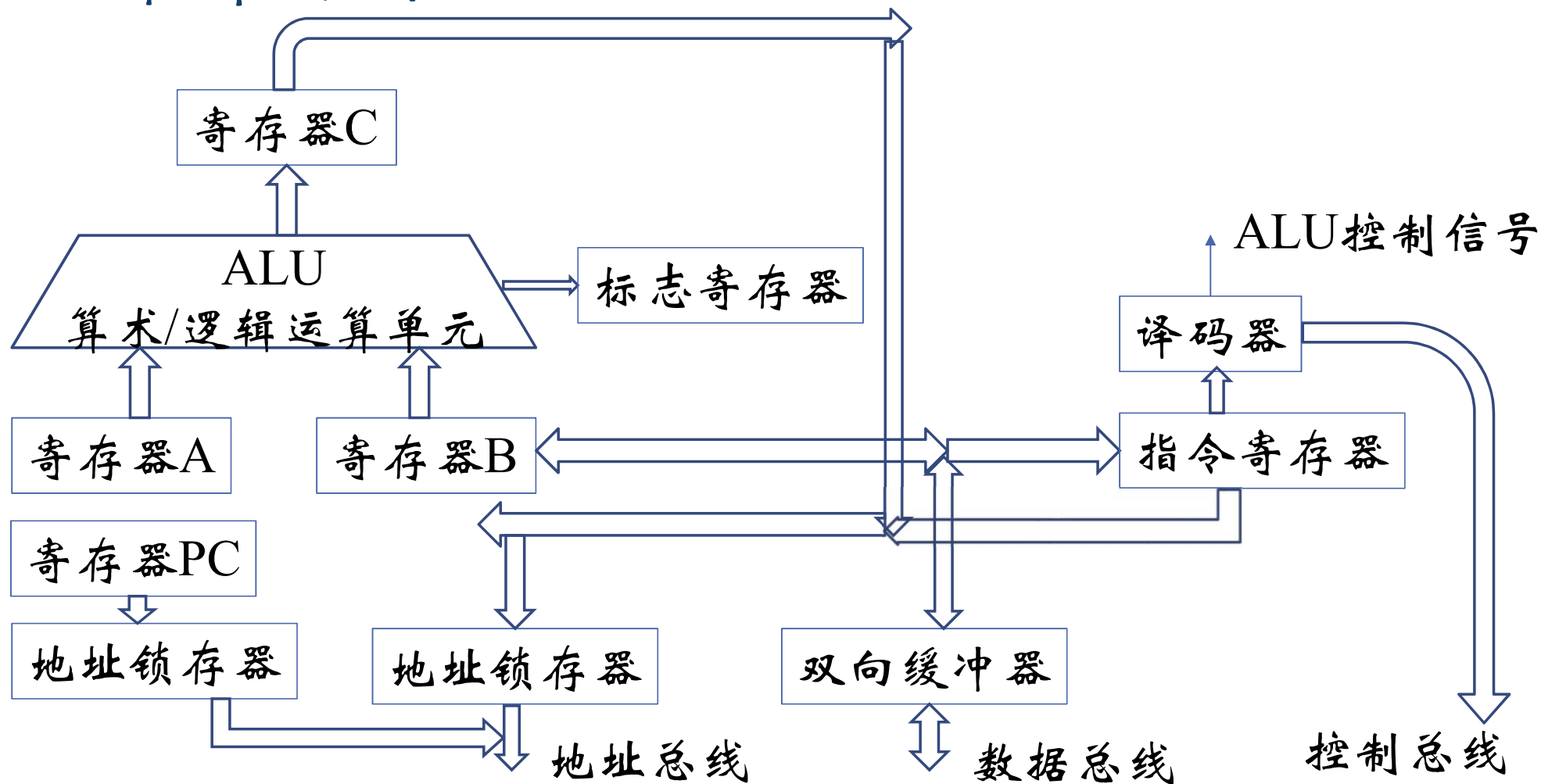
数据输入、输出



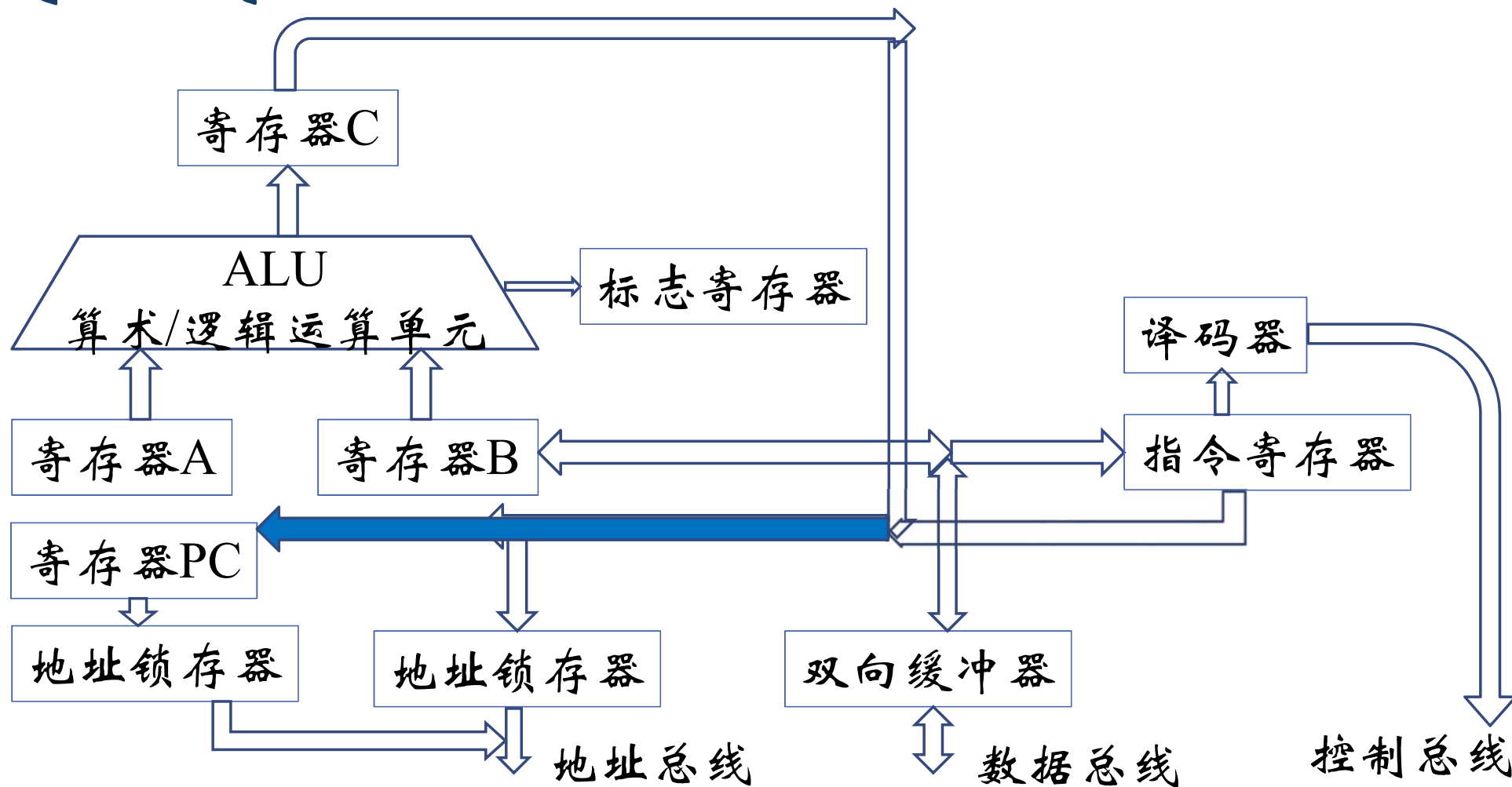
指令输入



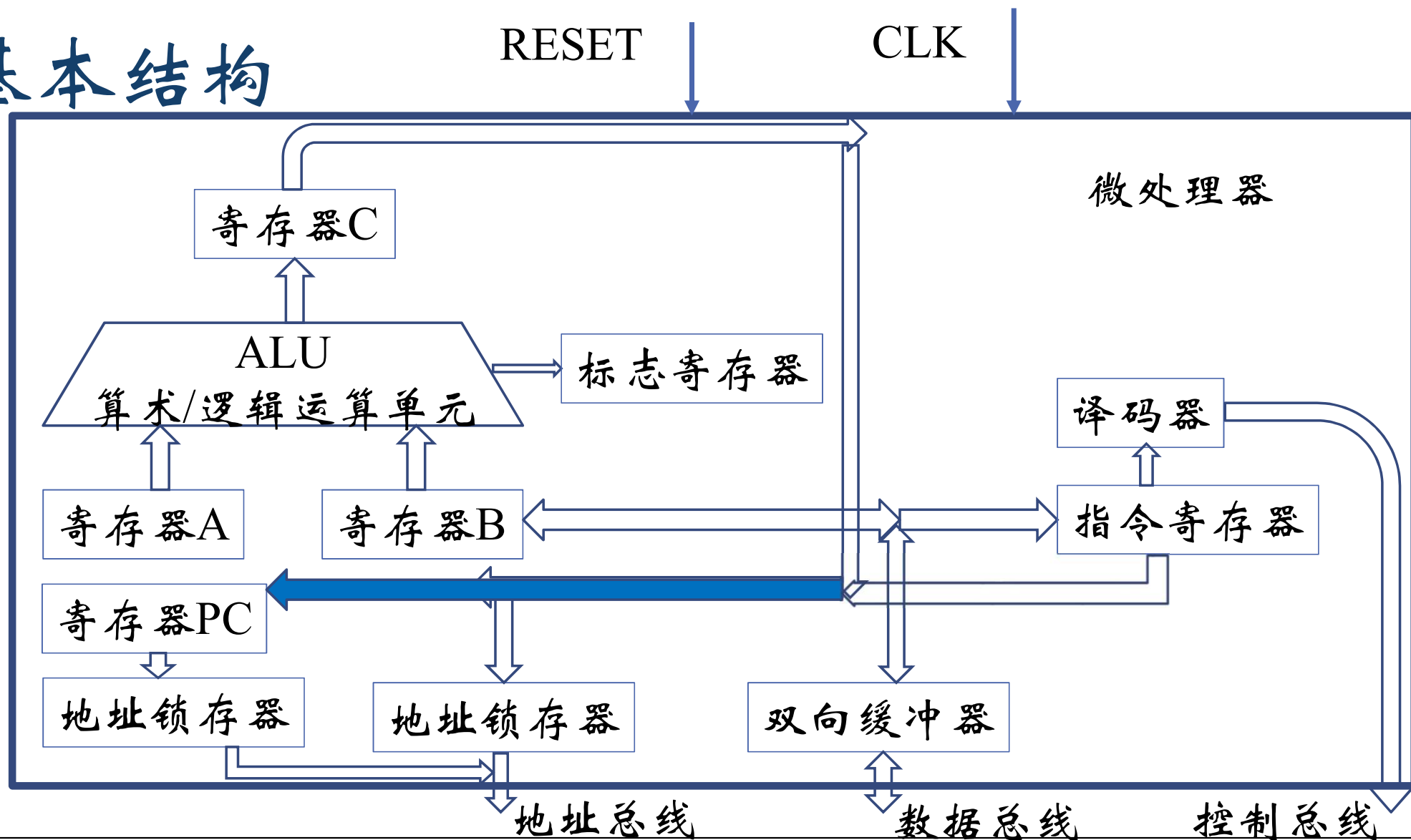
执行哪种操作



程序控制



基本结构



小结

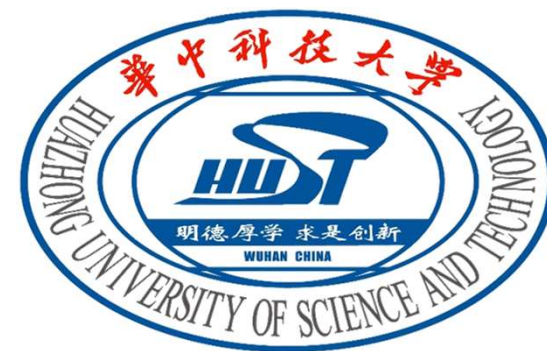
- 微处理器三类基本指令
- 微处理器各类指令执行的基本原理
- 微处理器的基本结构
 - ALU
 - 控制器
 - 寄存器
- 微处理器外部信号
 - 地址总线、数据总线、控制总线
 - 时钟、复位

下一讲：简单指令集MIPS微处理器数据通路

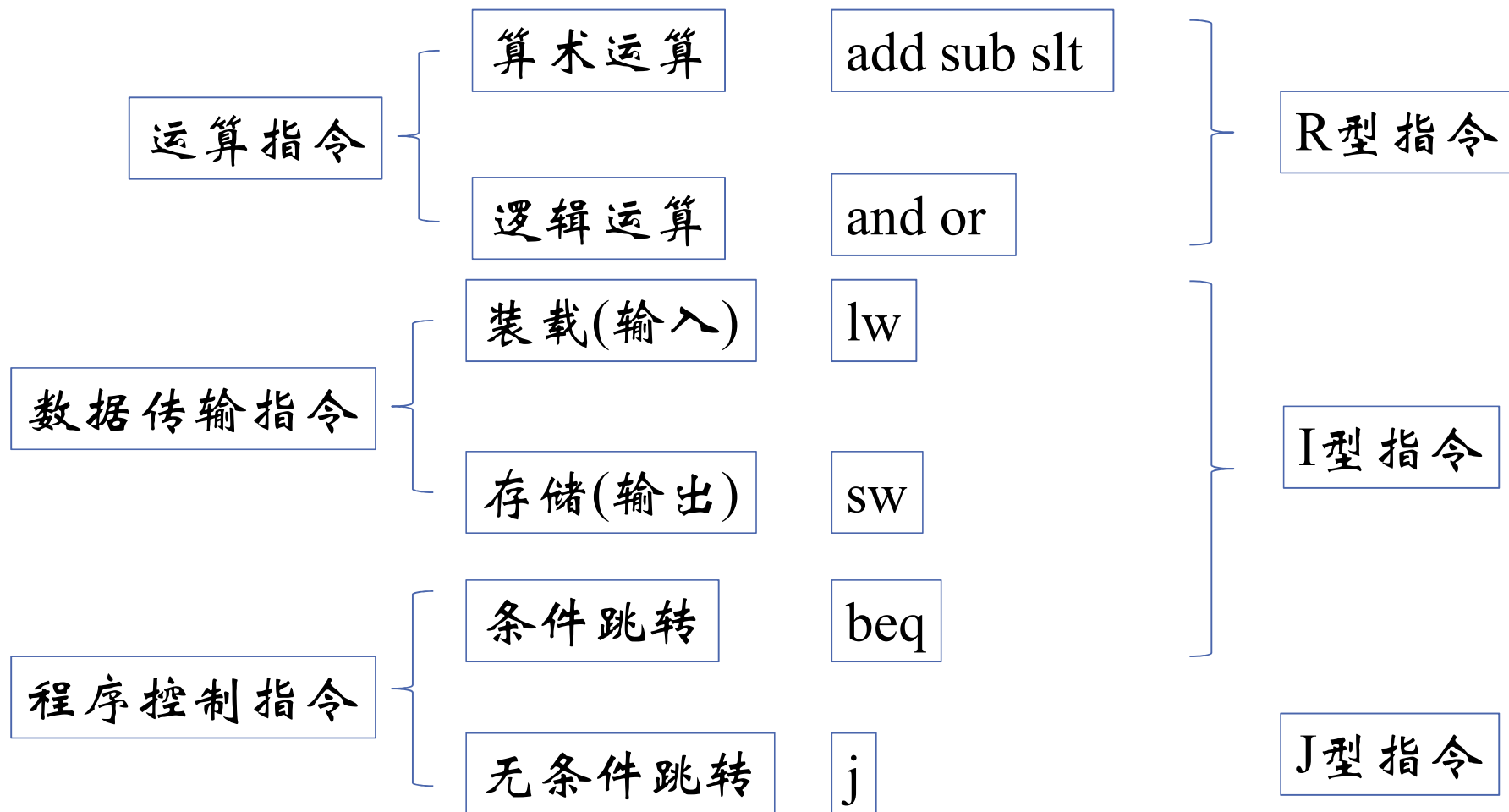
微机原理与接口技术

MIPS微处理器数据通路设计

华中科技大学 左冬红



简单指令集



术语

数据通路

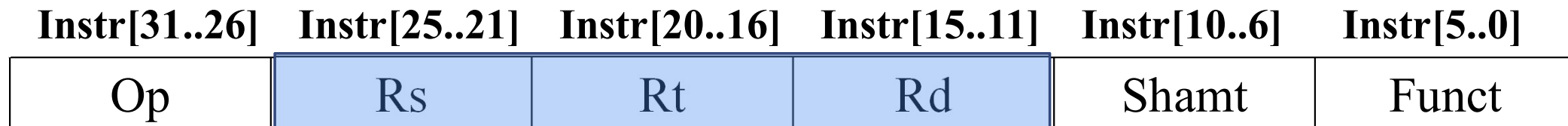
指令执行过程中实现指令获取以及数据处理的电路模块和传输路径。

约定

$\text{Instr}[31:0]$ 表示32位机器指令的二进制位

R型运算指令执行部件构成

add \$Rd,\$Rs,\$Rt



R型 Op: 000000

寄存器编号

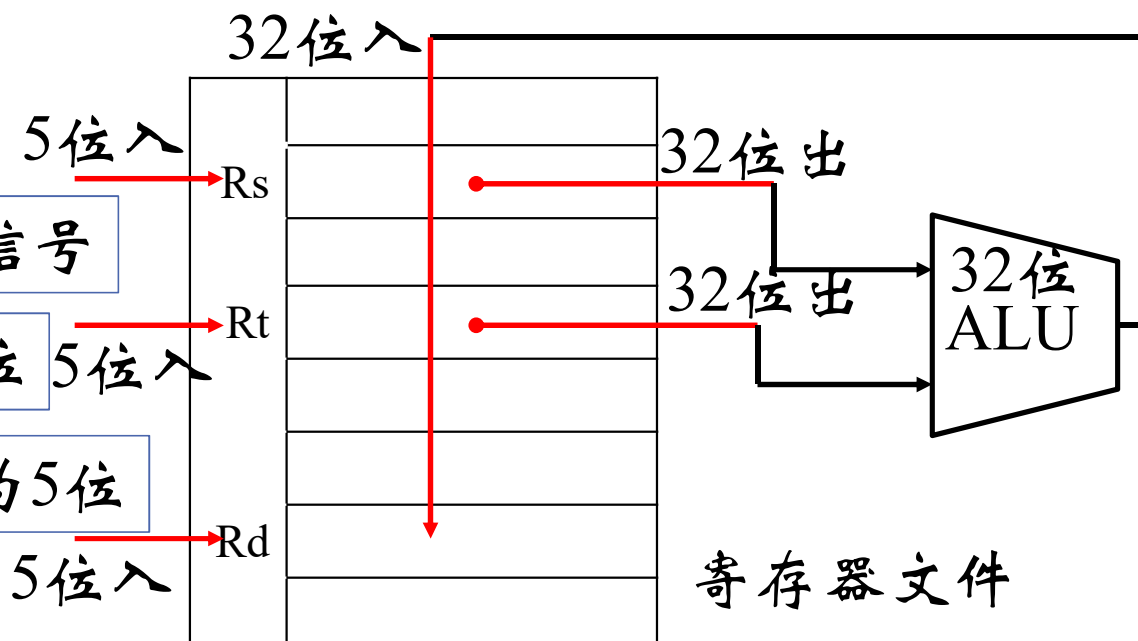
指令功能

操作执行流程

模块输入输出信号

寄存器都为32位

寄存器编号都为5位

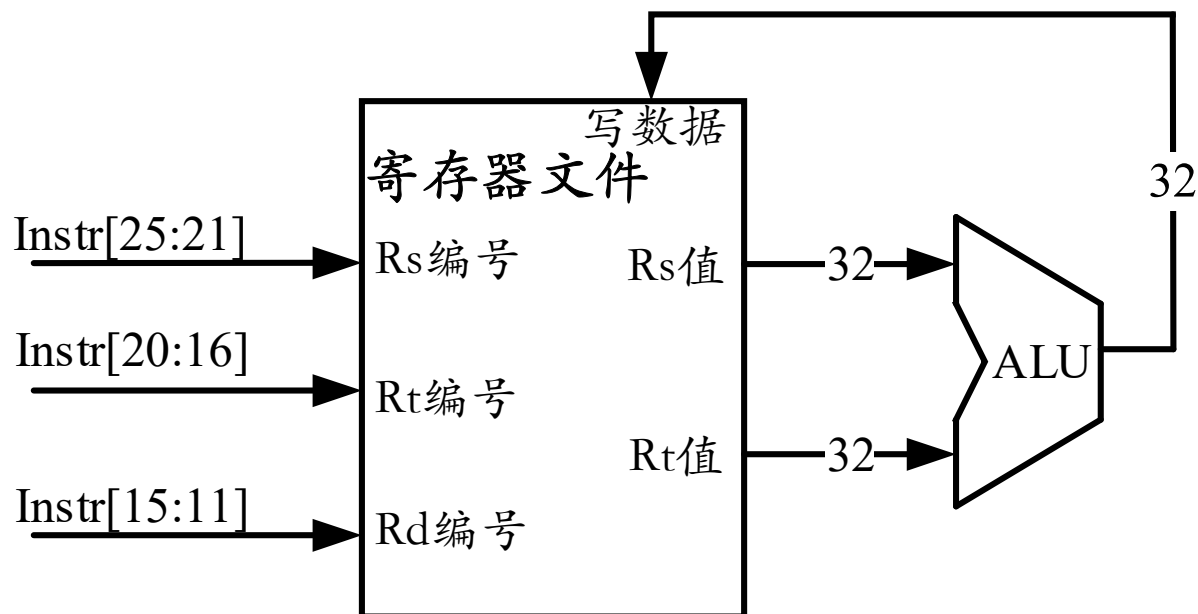


简化设计输入、
输出信号独立
并行工作

R型运算指令执行部件构成

Instr[31..26] Instr[25..21] Instr[20..16] Instr[15..11] Instr[10..6] Instr[5..0]

Op	Rs	Rt	Rd	Shamt	Funct
----	----	----	----	-------	-------



数据传输指令执行部件

lw(sw) \$Rt,Imm(\$Rs)

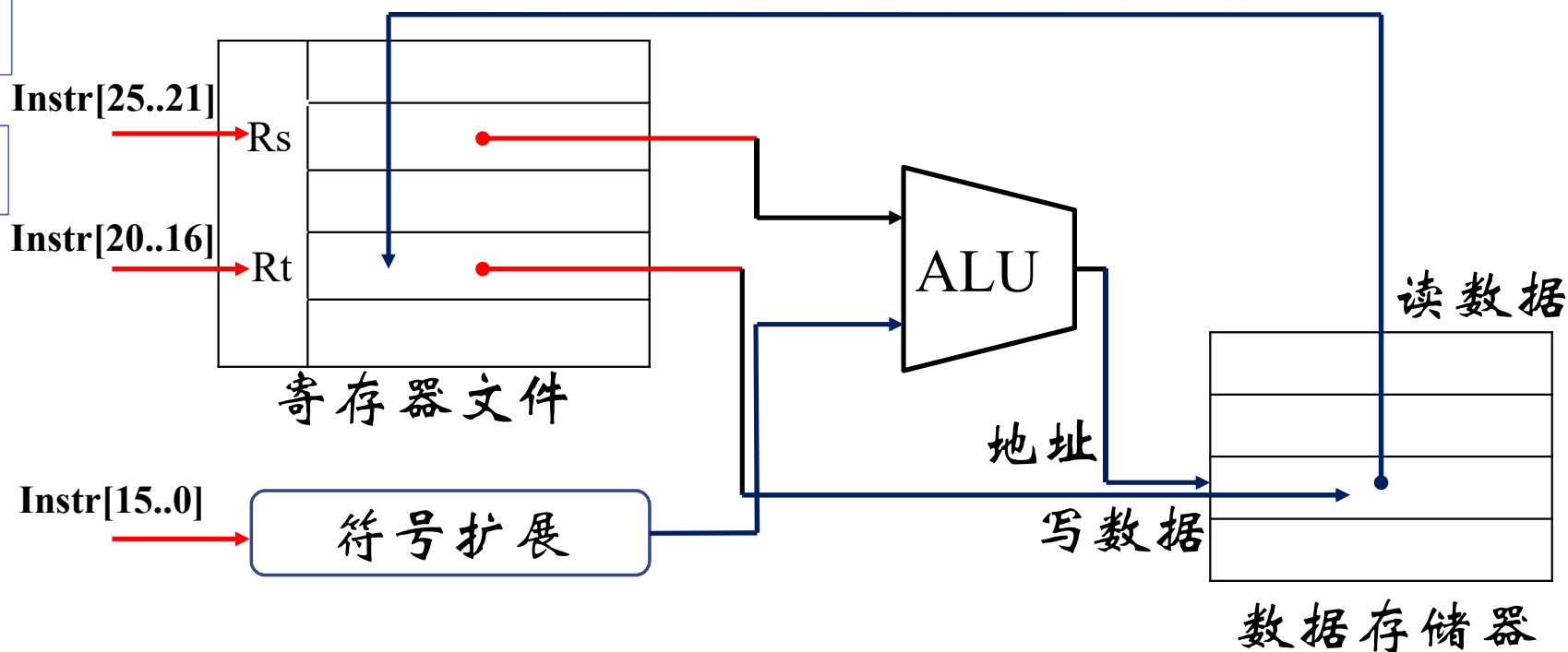
Instr[31..26]	Instr[25..21]	Instr[20..16]	Instr[15..0]
Op	Rs	Rt	Imm

操作执行流程

计算存储地址

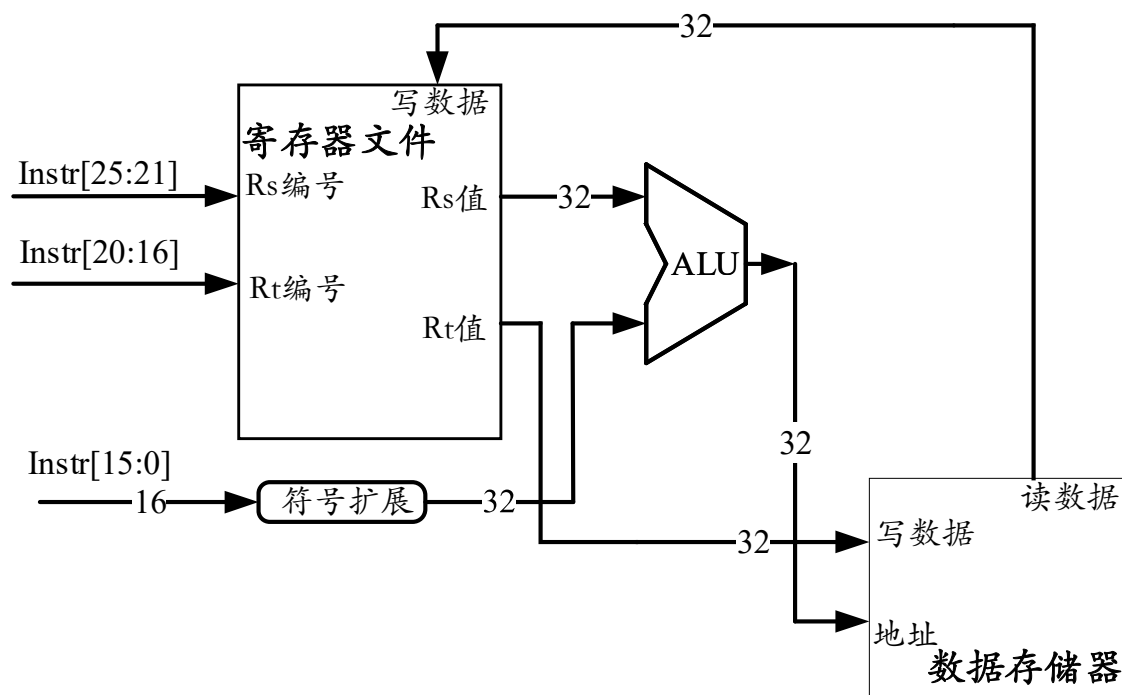
lw

sw



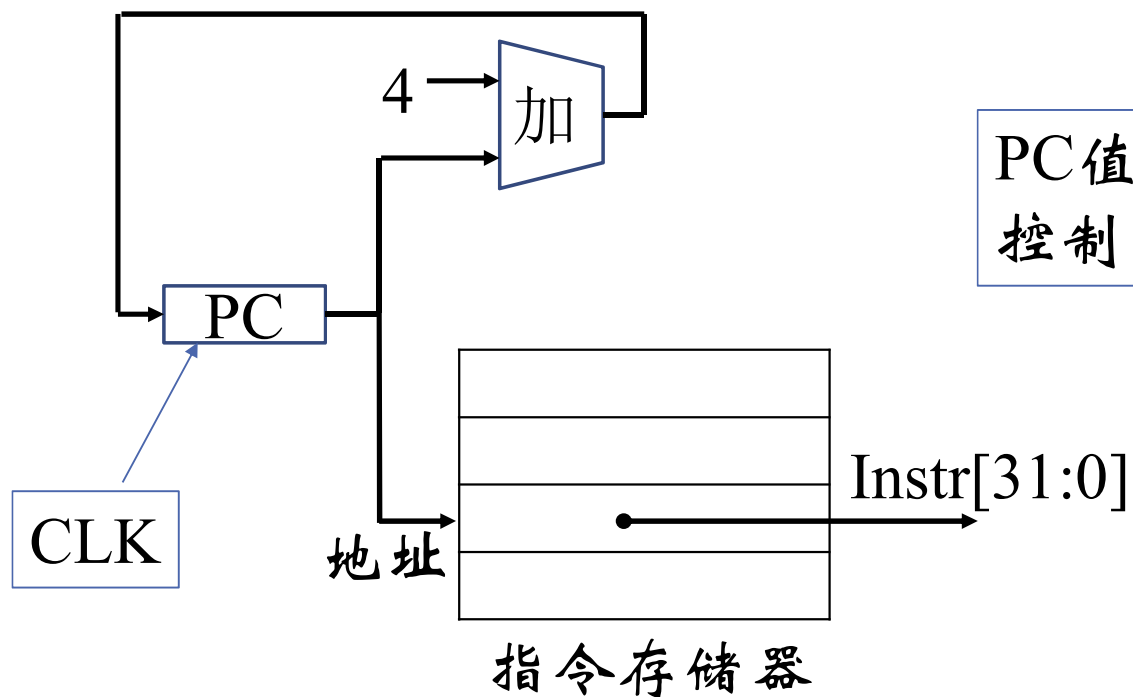
数据传输指令执行部件

Instr[31..26]	Instr[25..21]	Instr[20..16]	Instr[15..0]
Op	Rs	Rt	Imm



顺序获取指令部件

程序顺序执行时，PC自动加4指向下一条指令



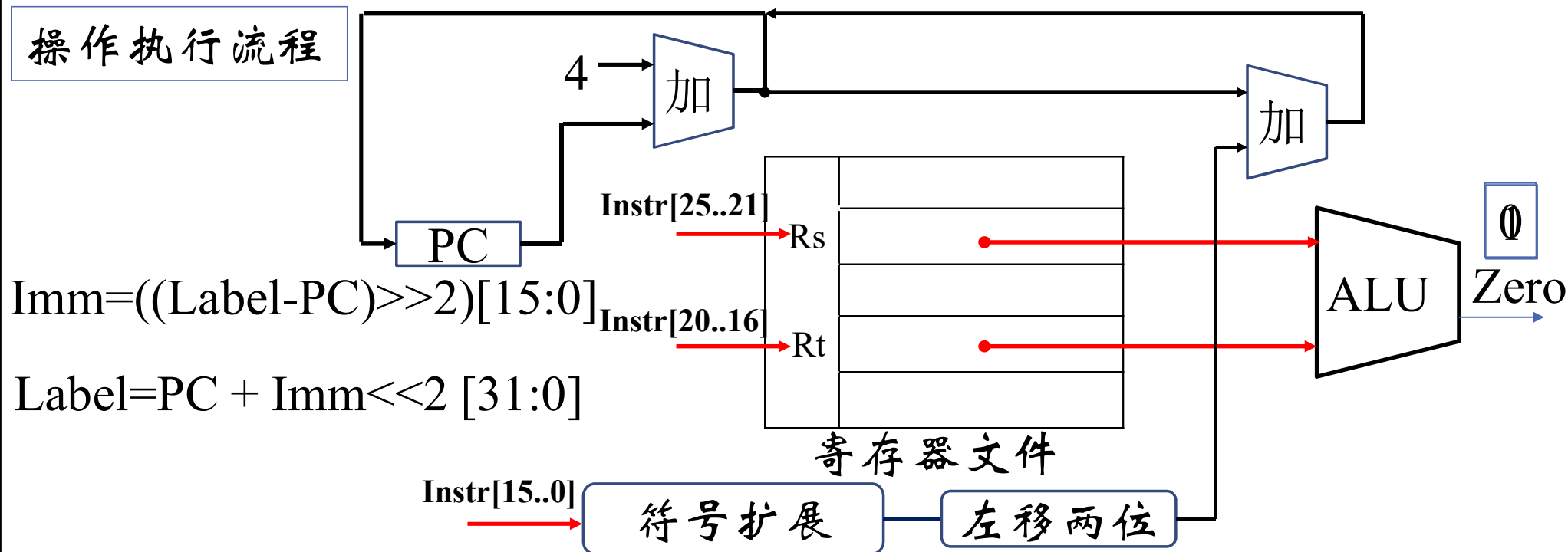
PC值的修改需由时钟信号控制节拍，为时序逻辑电路

条件跳转指令beq执行部件

beq \$Rs,\$Rt,Label

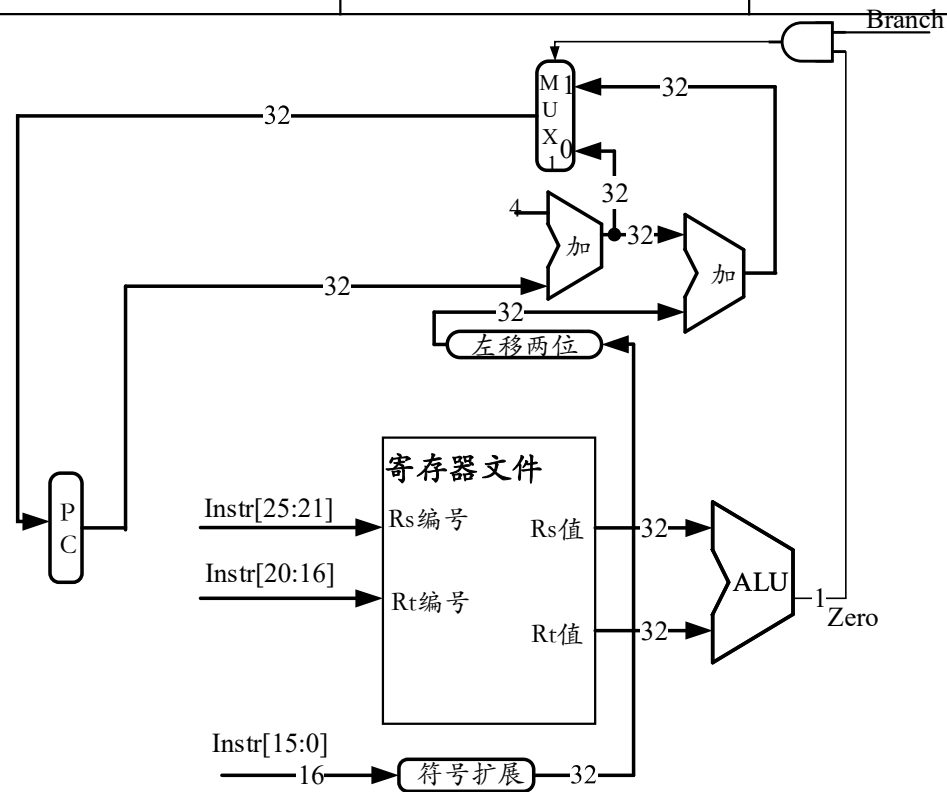
Instr[31..26]	Instr[25..21]	Instr[20..16]	Instr[15..0]
Op	Rs	Rt	Imm

操作执行流程



条件跳转指令beq执行部件

Instr[31..26]	Instr[25..21]	Instr[20..16]	Instr[15..0]
Op	Rs	Rt	Imm

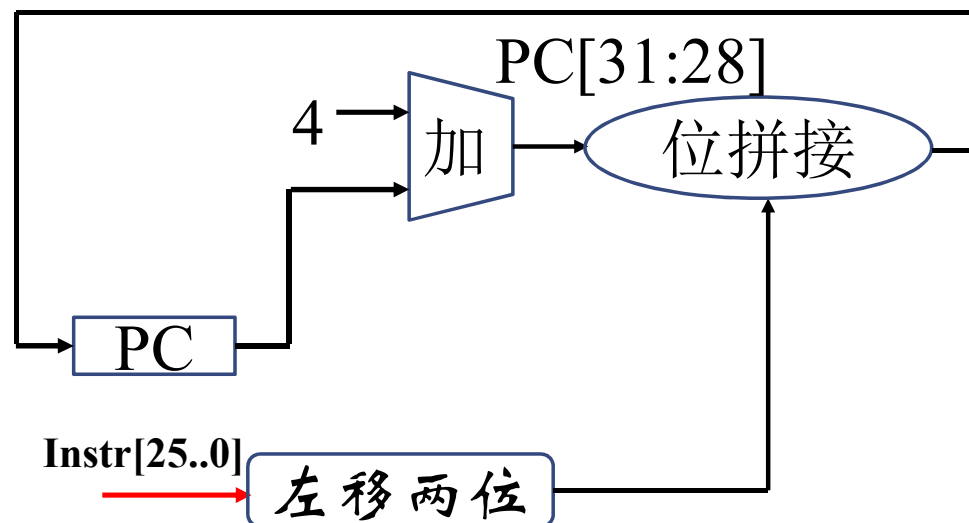


j指令执行部件

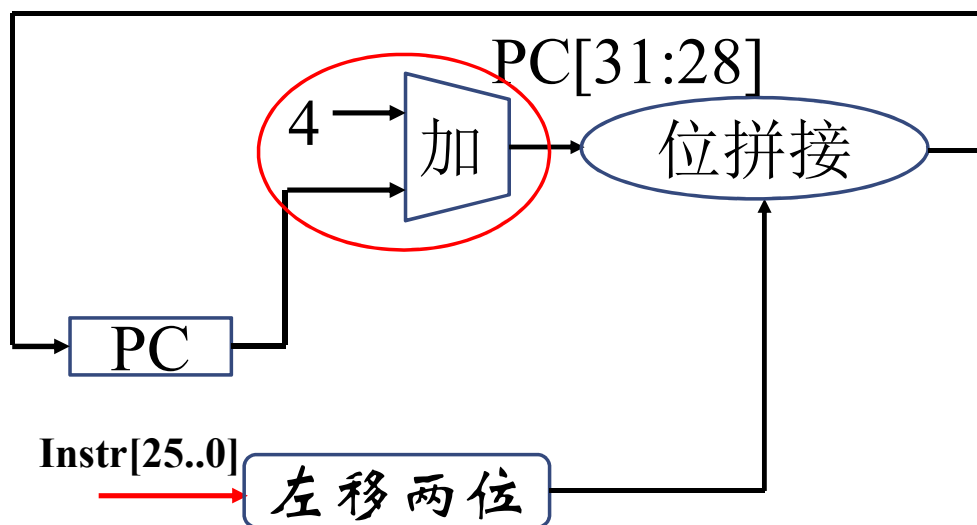
j Label

Instr[31:26]	Instr[25:0]
Op	Imm

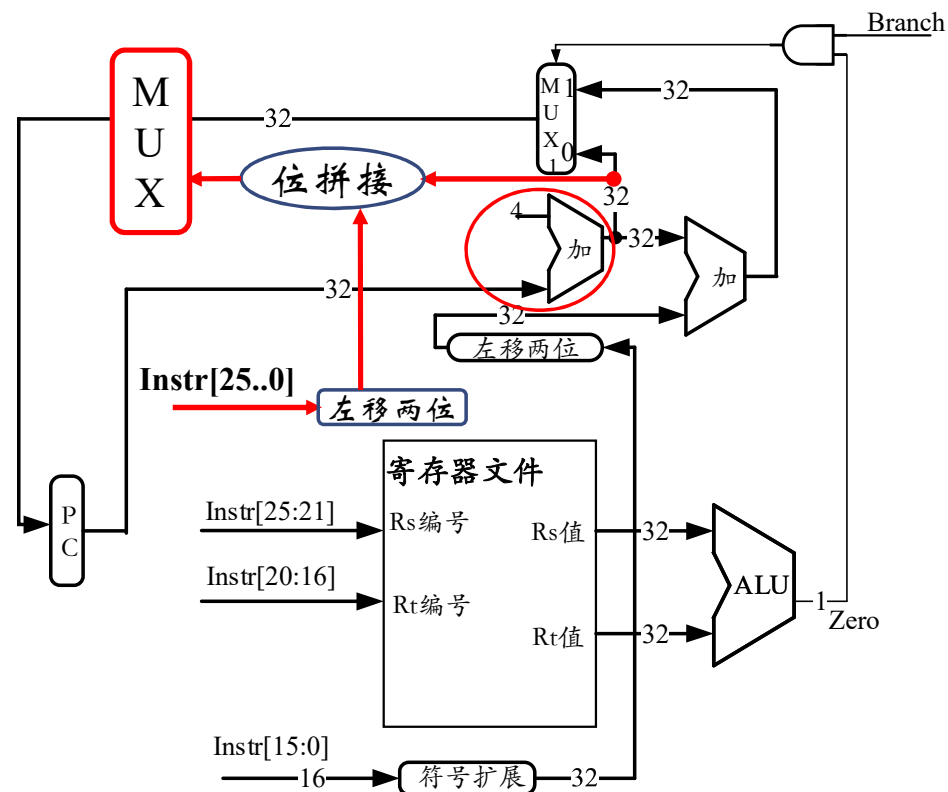
Label={PC[31:28],Imm,2'b00}



PC部件合并



无条件跳转

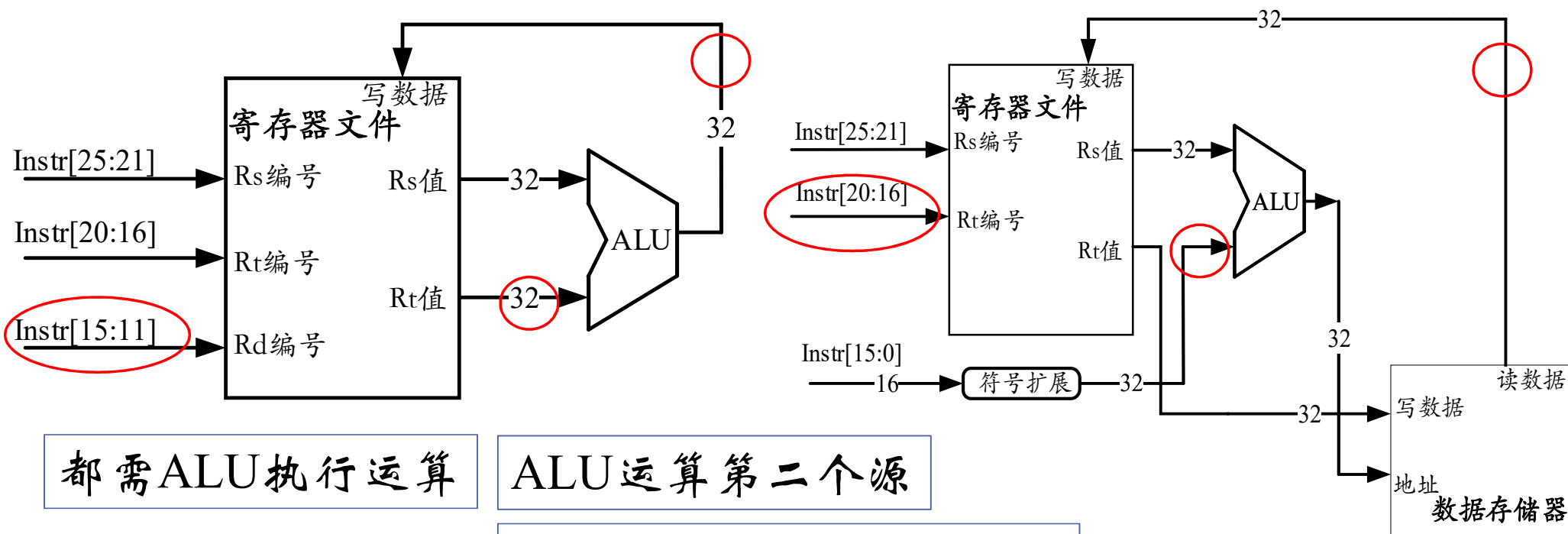


条件跳转

含顺序执行

数据传输与运算合并

多源输入电路实现采用复用器



都需ALU执行运算

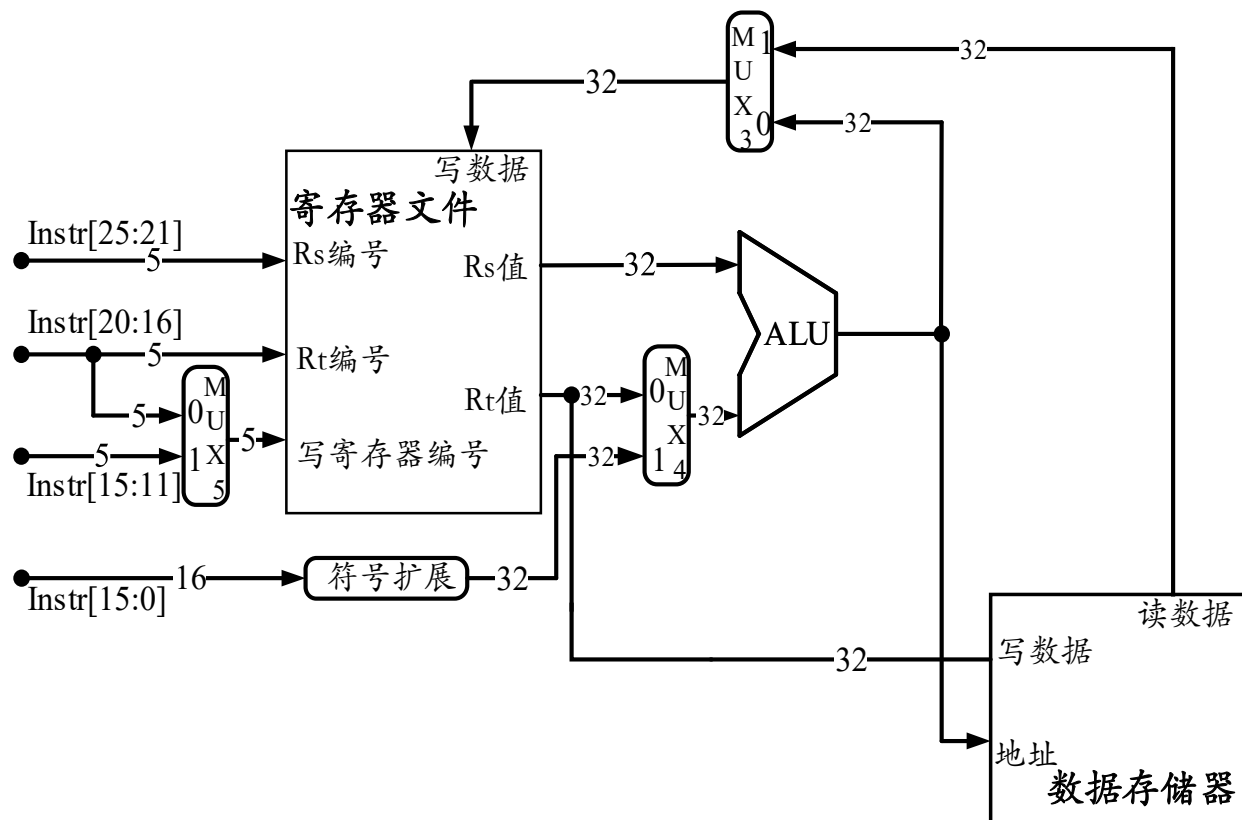
ALU运算第二个源

写数据到寄存器

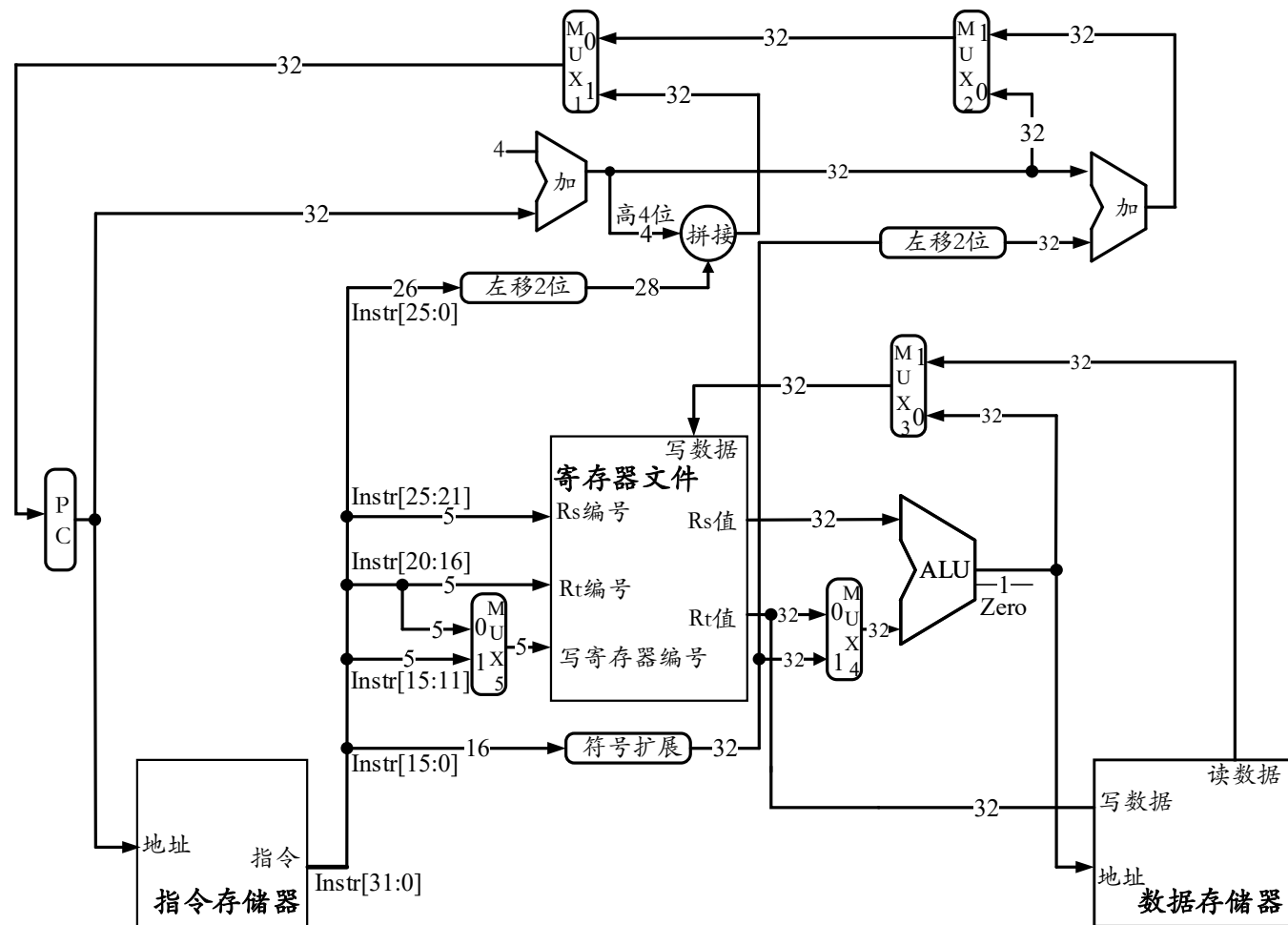
写入寄存器文件的数据源

寄存器文件的写寄存器编号

数据传输与运算合并



完整数据通路



小结

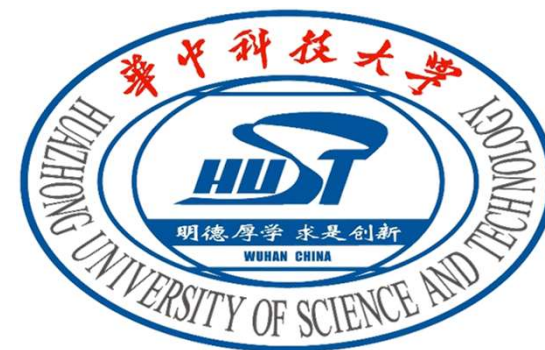
- 简单指令集构成
- 各类型指令执行部件构成
- 部件合并
- 完整数据通路

下一讲：简单指令集MIPS微处理器控制器

微机原理与接口技术

MIPS微处理器控制器设计

华中科技大学 左冬红



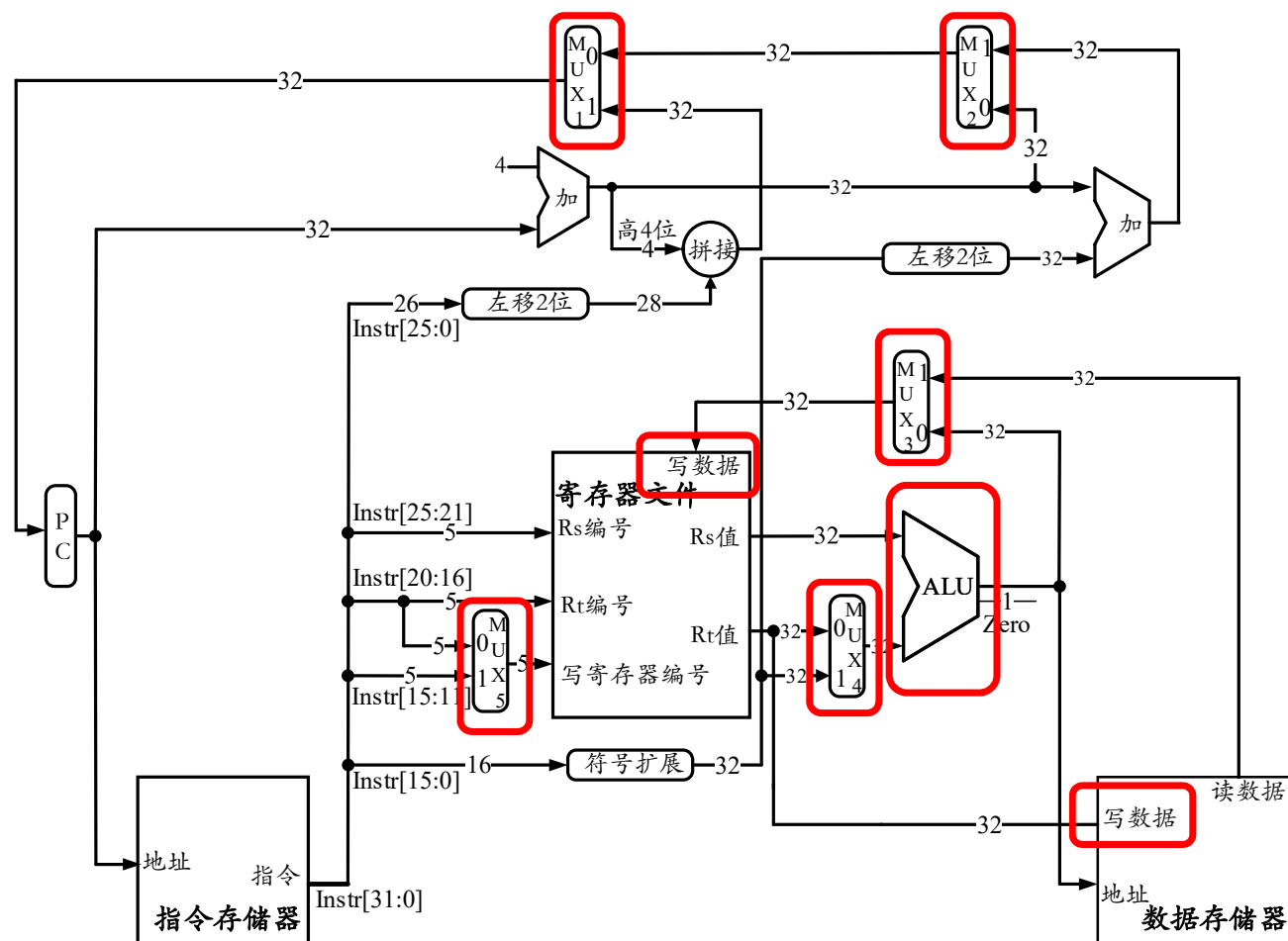
回顾——数据通路

需要哪些控制信号？

1. ALU执行哪些运算

2. 复用器通道选择信号

3. 写入使能信号



ALU控制信号

1.ALU执行哪些运算

指令	add	sub	and	or	slt	lw	sw	beq
运算	加	减	与	或	小于设置	加	加	减

5种运算需要多少位编码表示?

3位二进制码

为便于MIPS指令集扩展, 直接采用统一编码

产生依据?

ALUCtr[3:0]	0000	0001	0010	0110	0111
操作类型	与	或	加	减	小于设置

ALU控制信号

1.ALU执行哪些运算

指令	add	sub	and	or	slt	lw	sw	beq
运算	加	减	与	或	小于设置	加	加	减

Instr[31..26]	Instr[25..21]	Instr[20..16]	Instr[15..11]	Instr[10..6]	Instr[5..0]
Op	Rs	Rt	Rd	Shamt	Funct
Op	Rs	Rt	Imm		

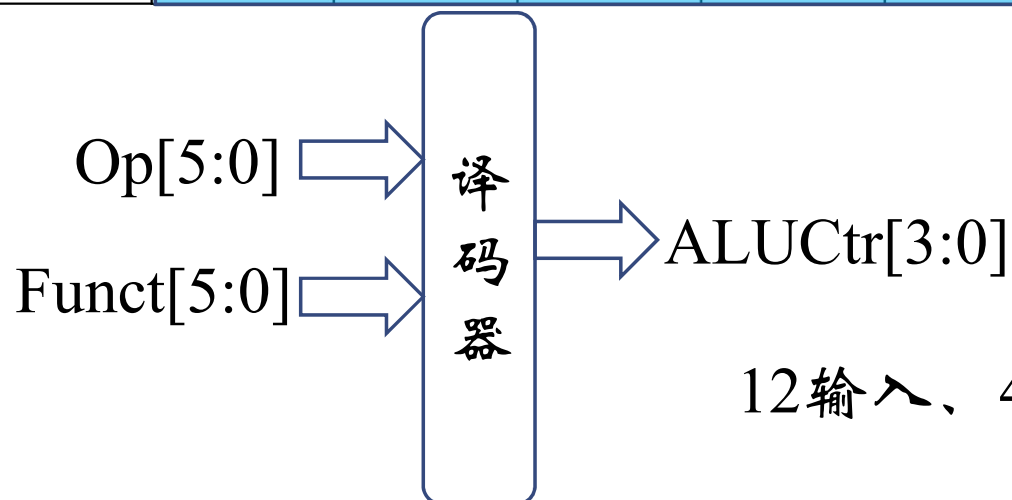
产生依据？

R型 Op: 000000

操作码和功能码共同决定ALUCtr[3:0]的取值

ALU控制信号

指令	add	sub	and	or	slt	lw	sw	beq
运算	加	减	与	或	小于设置	加	加	减



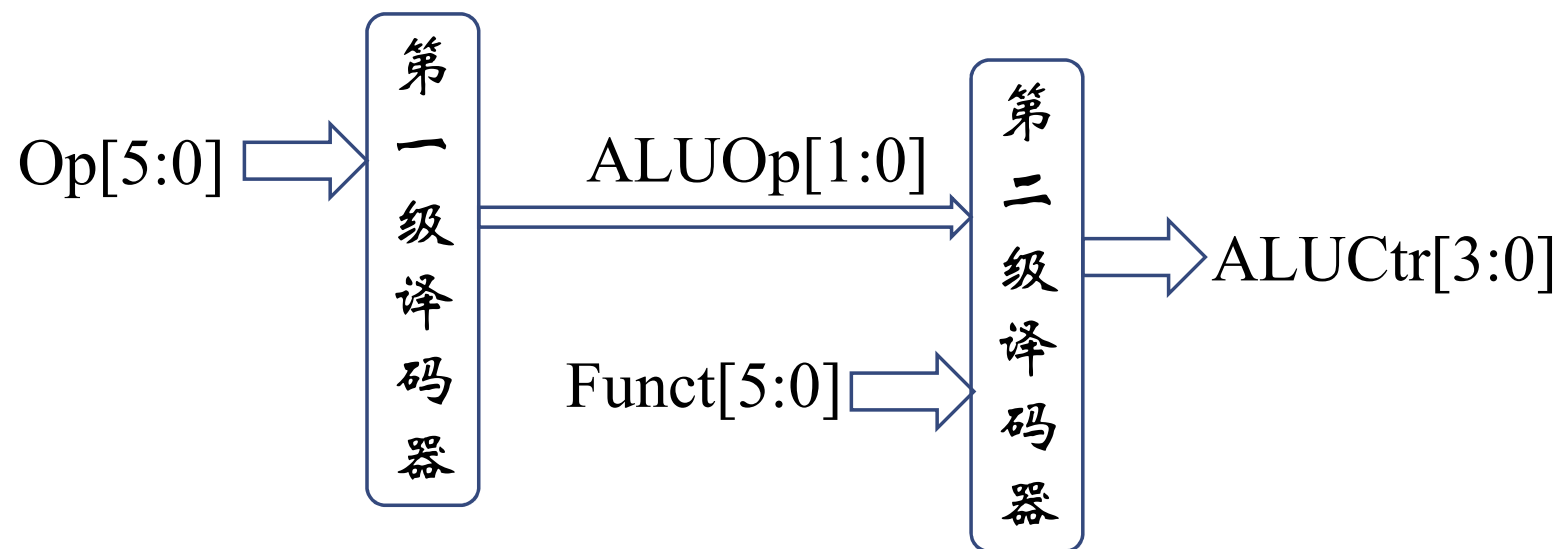
12输入、4输出译码器，能否简化？

R型 Op: 000000

I型无Funct[5:0],由Op直接决定运算类型

根据Op分为三种类型,由2位编码ALUOp[1:0]表示

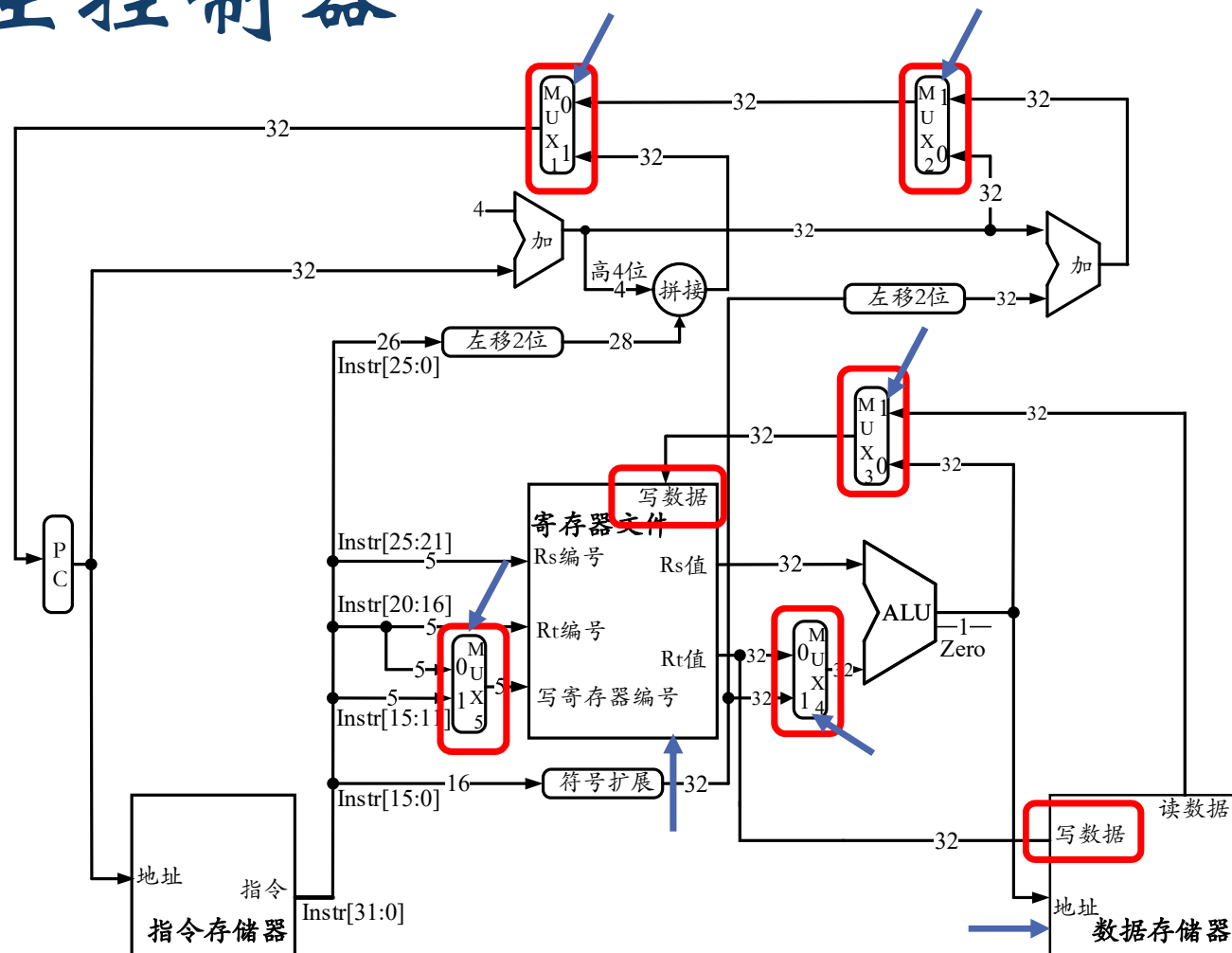
ALU控制信号两级译码



ALU控制信号译码功能表

指令类型	运算类型	第一级输入	第一级输出 (第二级输入)	第二级输入	第二级输出
		Op[5:0]	ALUOp[1:0]	Funct[5:0]	ALUCtr[3:0]
lw	加	100011	00	xxxxxx	0010
sw	加	101011	00	xxxxxx	0010
beq	减	000100	01	xxxxxx	0110
add	加	000000	10	10 00000	0010
sub	减	000000	10	10 00010	0110
and	与	000000	10	10 0110	0000
or	或	000000	10	10 0111	0001
slt	小于设置	000000	10	10 1010	0111

主控制器



2. 复用器通道选择信号

每个复用器1位通道选择信号

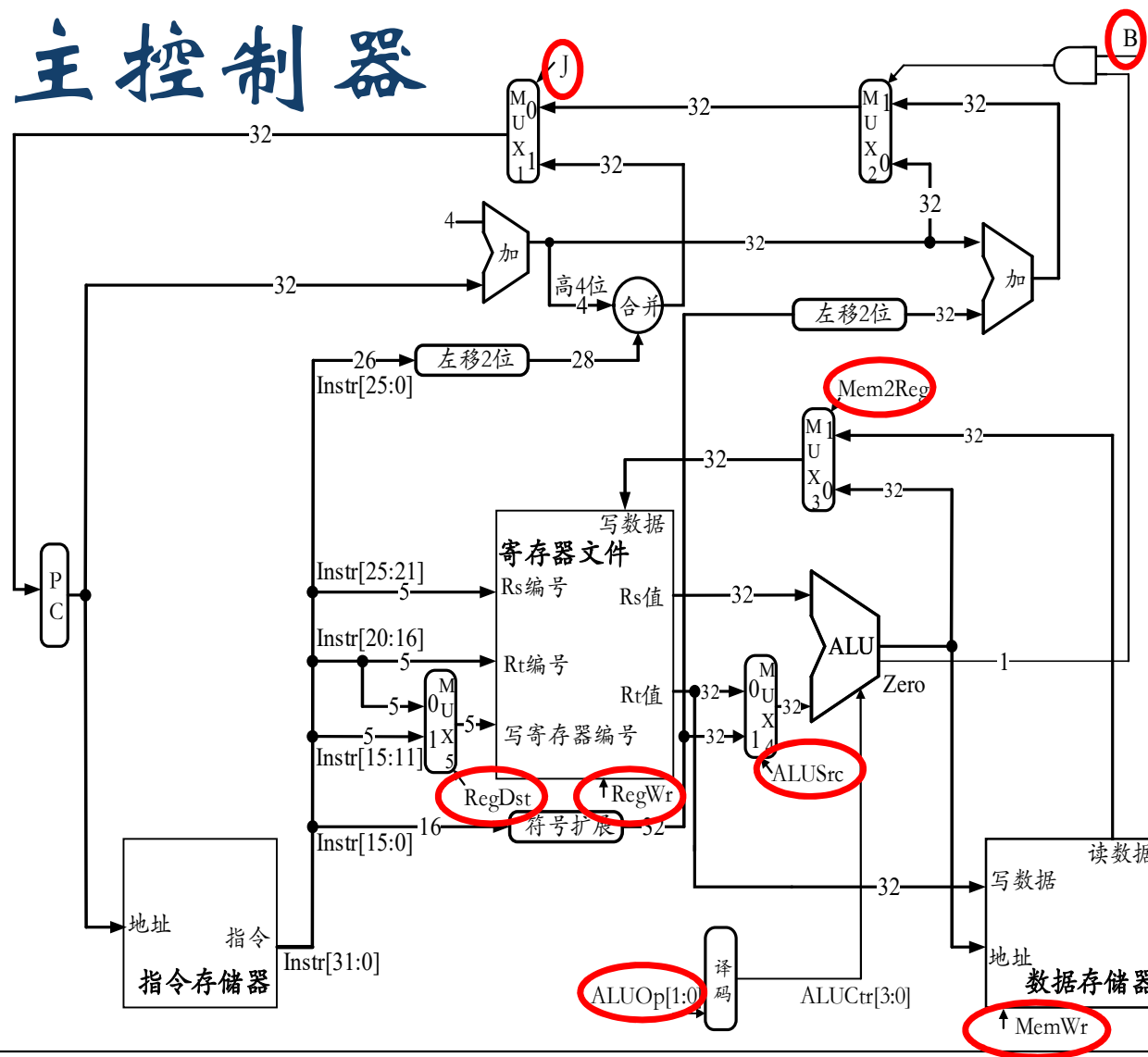
3. 写入使能信号

寄存器文件、数据存储器各1位写使能信号

产生的依据是?

Op [5:0]

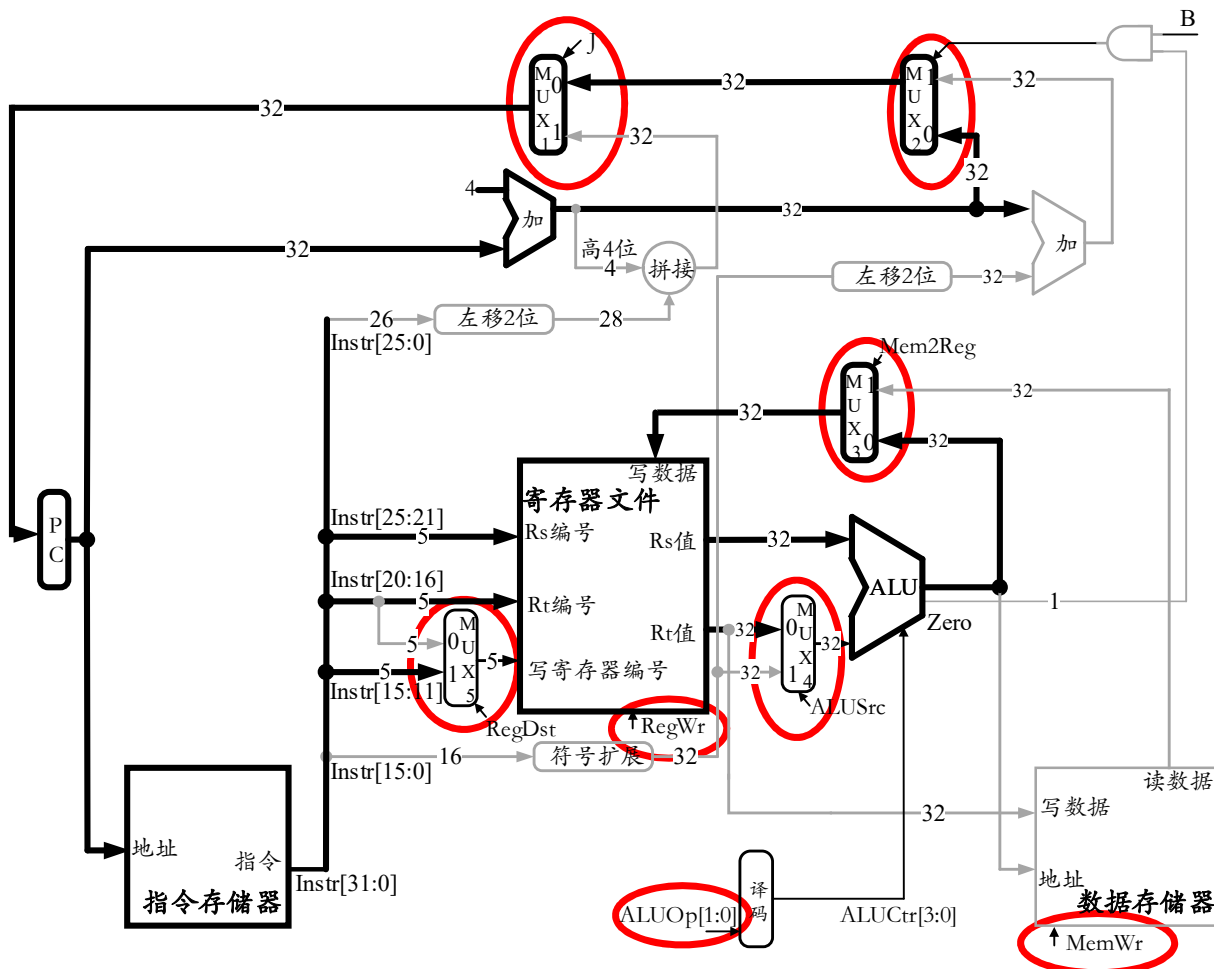
主控制器



控制信号	含义
J	1=j 指令
B	1=beq 指令
RegDst	1=写入Rd
RegWr	1=写入寄存器
ALUSrc	1=立即数参与运算
MemWr	1=写入数据存储器
Mem2Reg	1=lw 指令
ALUOp[1:0]	00=lw、sw 指令
	01=beq 指令
	10=R 型 指令

R型指令的控制信号取值

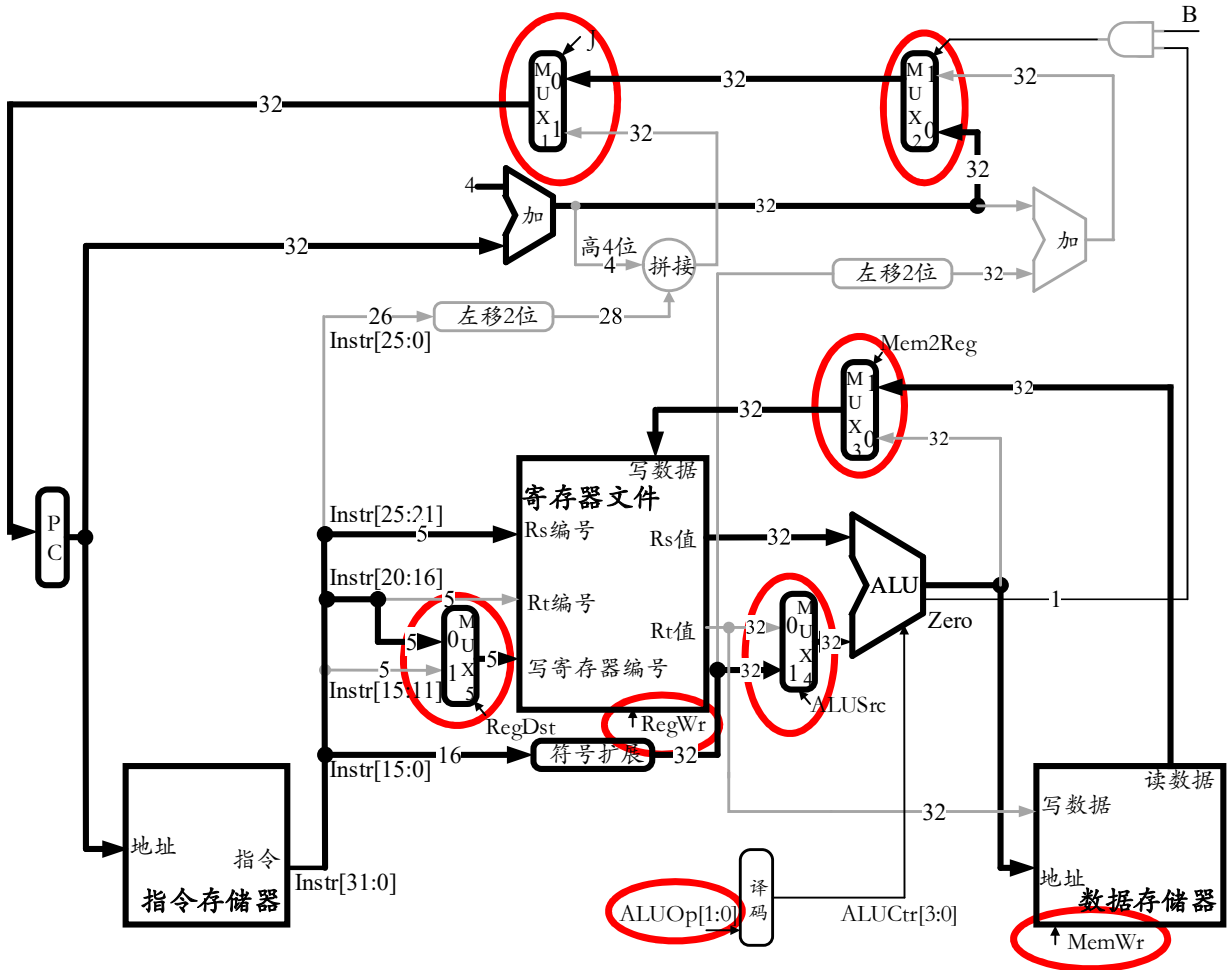
Op[5:0]=000000



控制信号	取值
J	0
B	0
RegDst	1
RegWr	1
ALUSrc	0
MemWr	0
Mem2Reg	0
ALUOp[1:0]	10

lw指令的控制信号取值

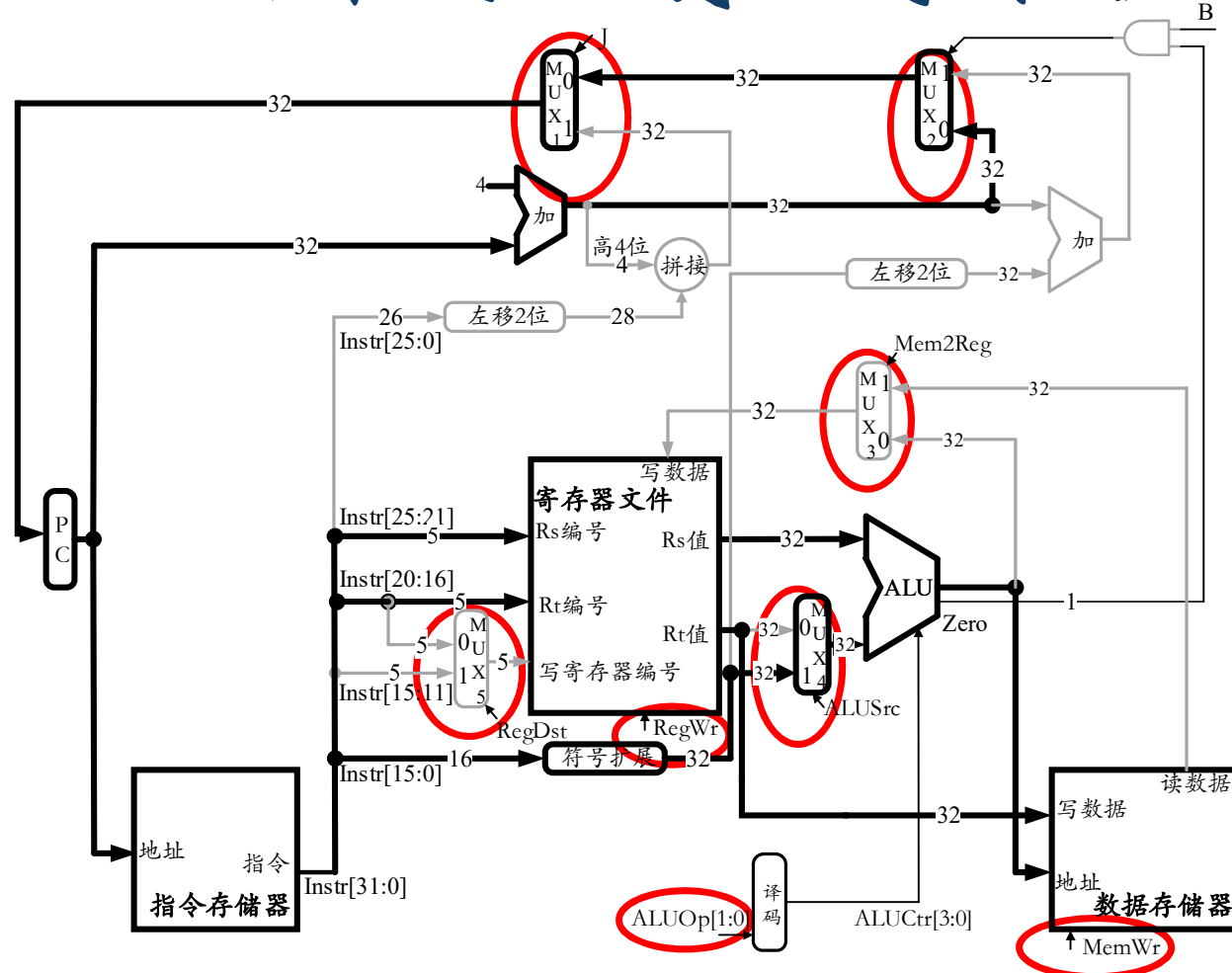
Op[5:0]=100011



控制信号	取值
J	0
B	0
RegDst	0
RegWr	1
ALUSrc	1
MemWr	0
Mem2Reg	1
ALUOp[1:0]	00

SW指令的控制信号取值

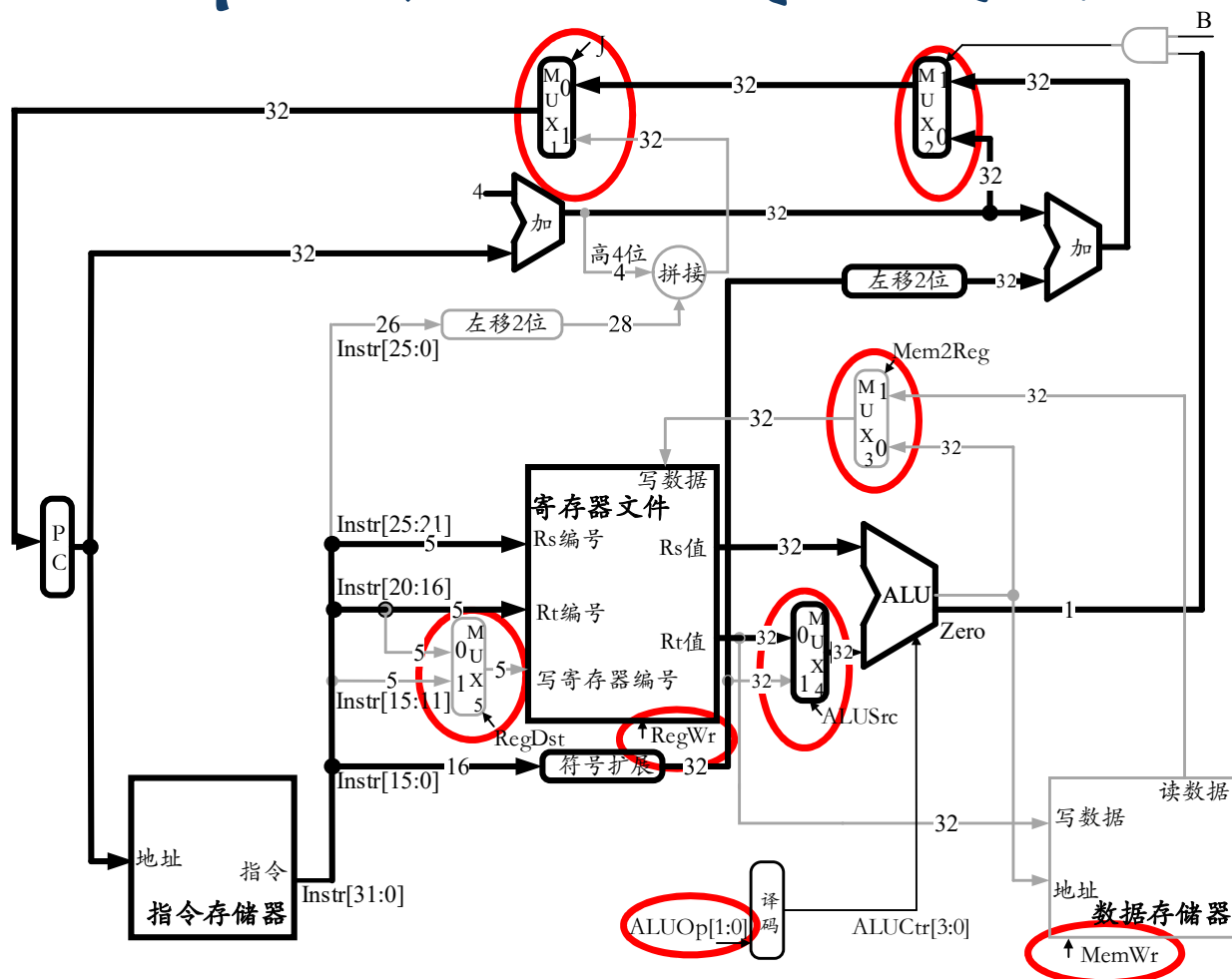
Op[5:0]=101011



控制信号	取值
J	0
B	0
RegDst	x
RegWr	0
ALUSrc	1
MemWr	1
Mem2Reg	x
ALUOp[1:0]	00

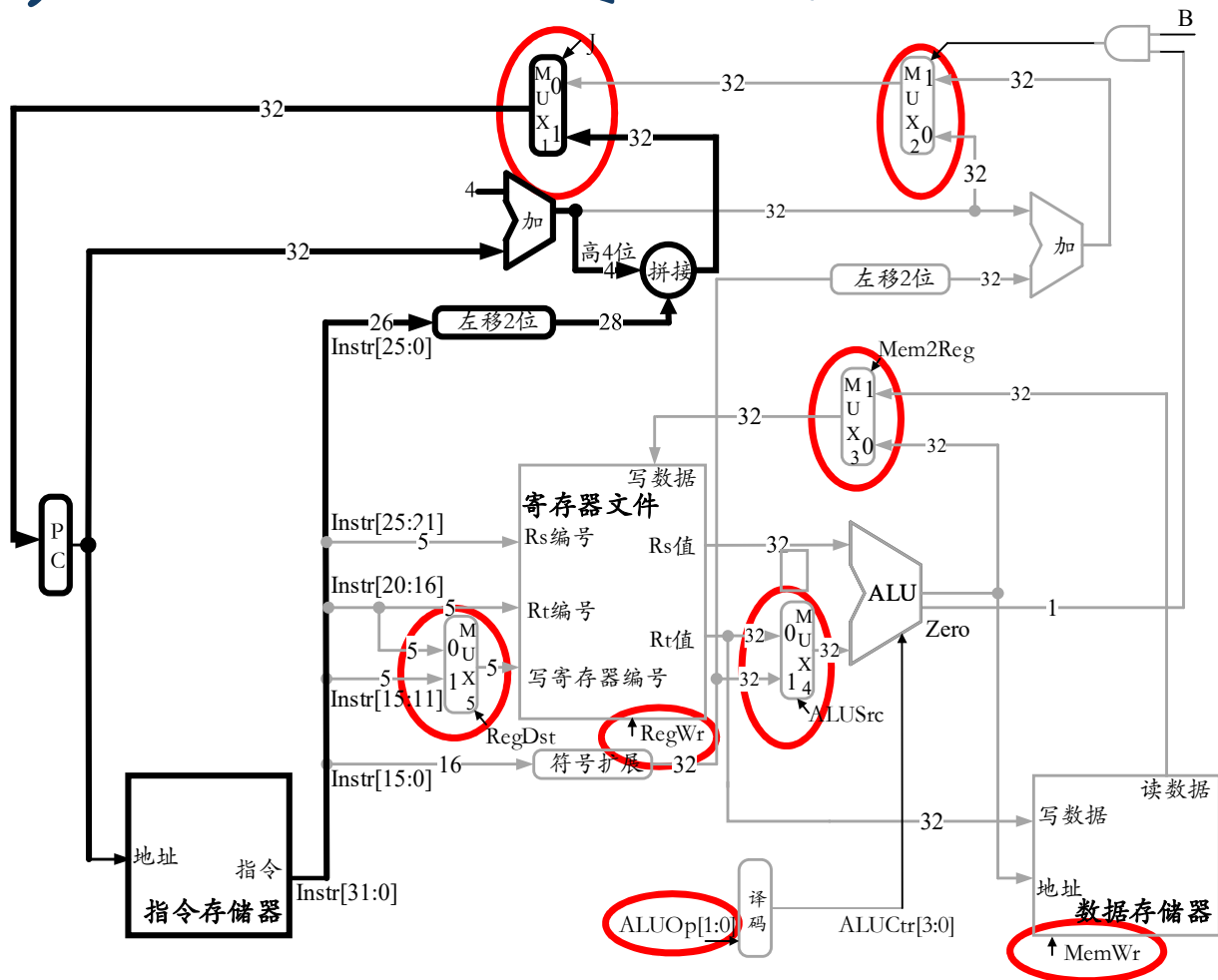
beq指令的控制信号取值

Op[5:0]=000100



控制信号	取值
J	0
B	1
RegDst	x
RegWr	0
ALUSrc	0
MemWr	0
Mem2Reg	x
ALUOp[1:0]	01

j指令的控制信号取值



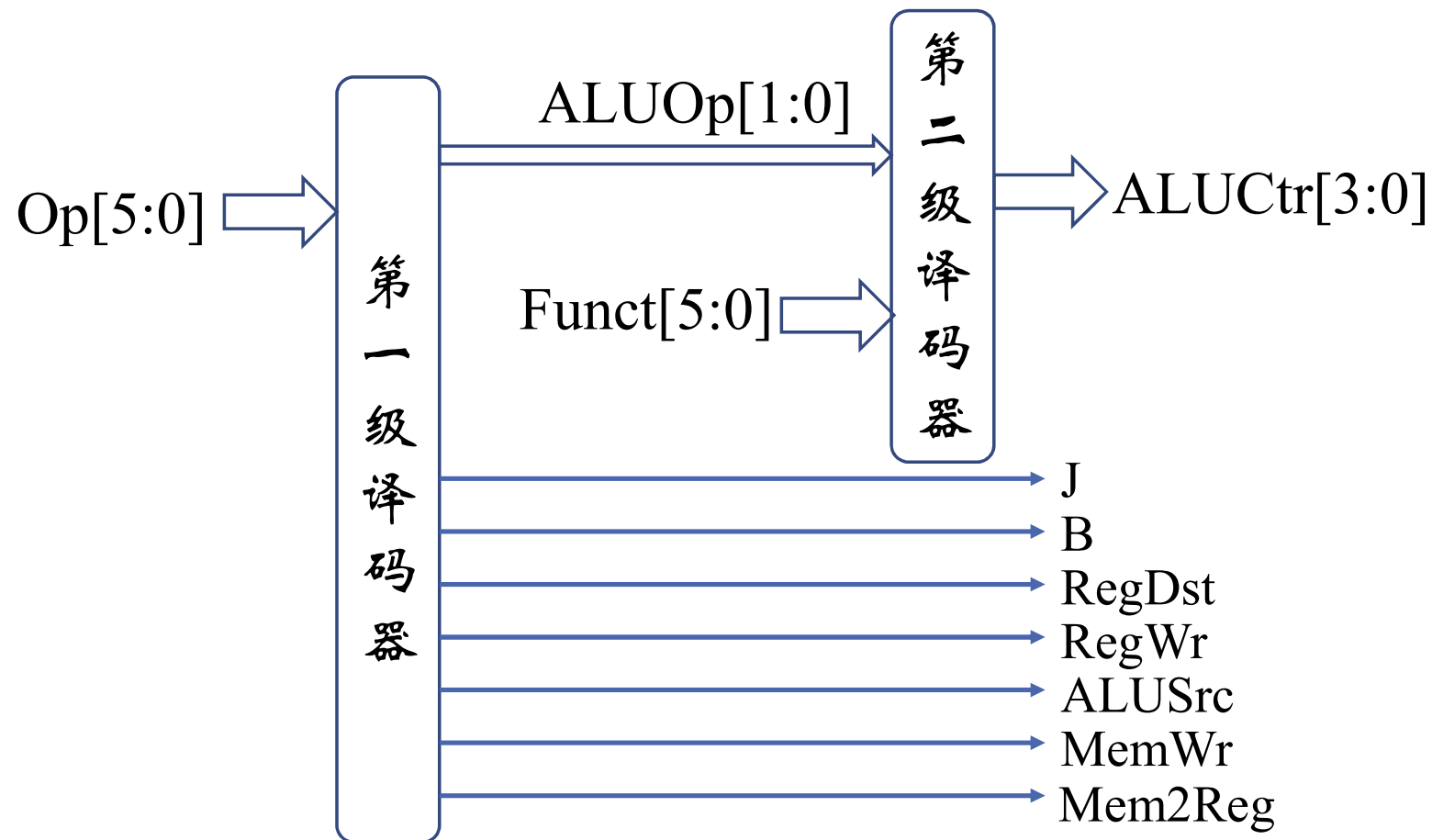
Op[5:0]=000010

控制信号	取值
J	1
B	x
RegDst	x
RegWr	0
ALUSrc	x
MemWr	0
Mem2Reg	x
ALUOp[1:0]	xx

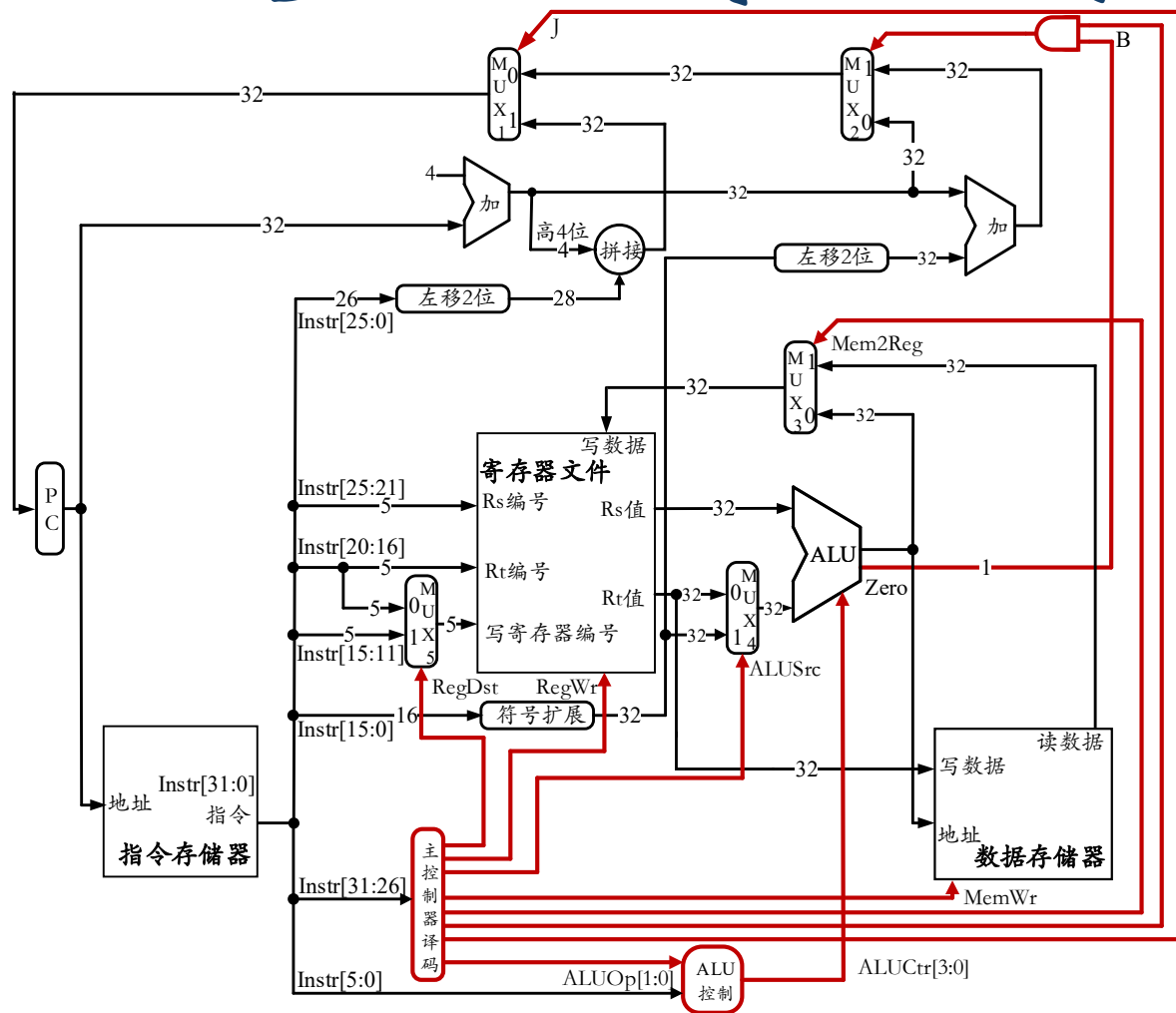
主控制器功能表

	R	lw	sw	beq	j
控制信号	000000	100011	101011	000100	000010
J	0	0	0	0	1
B	0	0	0	1	x
RegDst	1	0	x	x	x
RegWr	1	1	0	0	0
ALUSrc	0	1	1	0	x
MemWr	0	0	1	0	0
Mem2Reg	0	1	x	x	x
ALUOp[1:0]	10	00	00	01	xx

指令译码电路框图



数据通路与控制器合成的微处理器



小结

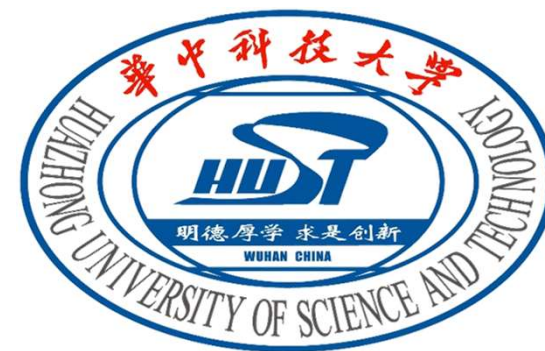
- 分析指令译码功能表
 - ALU控制信号
 - MUX通道选择信号
 - 写使能信号
- 两级译码，公用部分电路，简化电路

下一讲：指令执行过程示例

微机原理与接口技术

MIPS微处理器典型指令 电路执行过程

华中科技大学 左冬红



目标

- 结合指令功能以及微处理器电路微结构，进一步理解软件如何控制硬件实现特定功能

MIPS汇编指令序列

```
L1:  add $t1,$t2,$t3  
      sw $t1,2($t2)  
      beq $t1,$t2,L1  
      j L1
```

\$t1~\$t3:\$9~\$11

L1:0x0

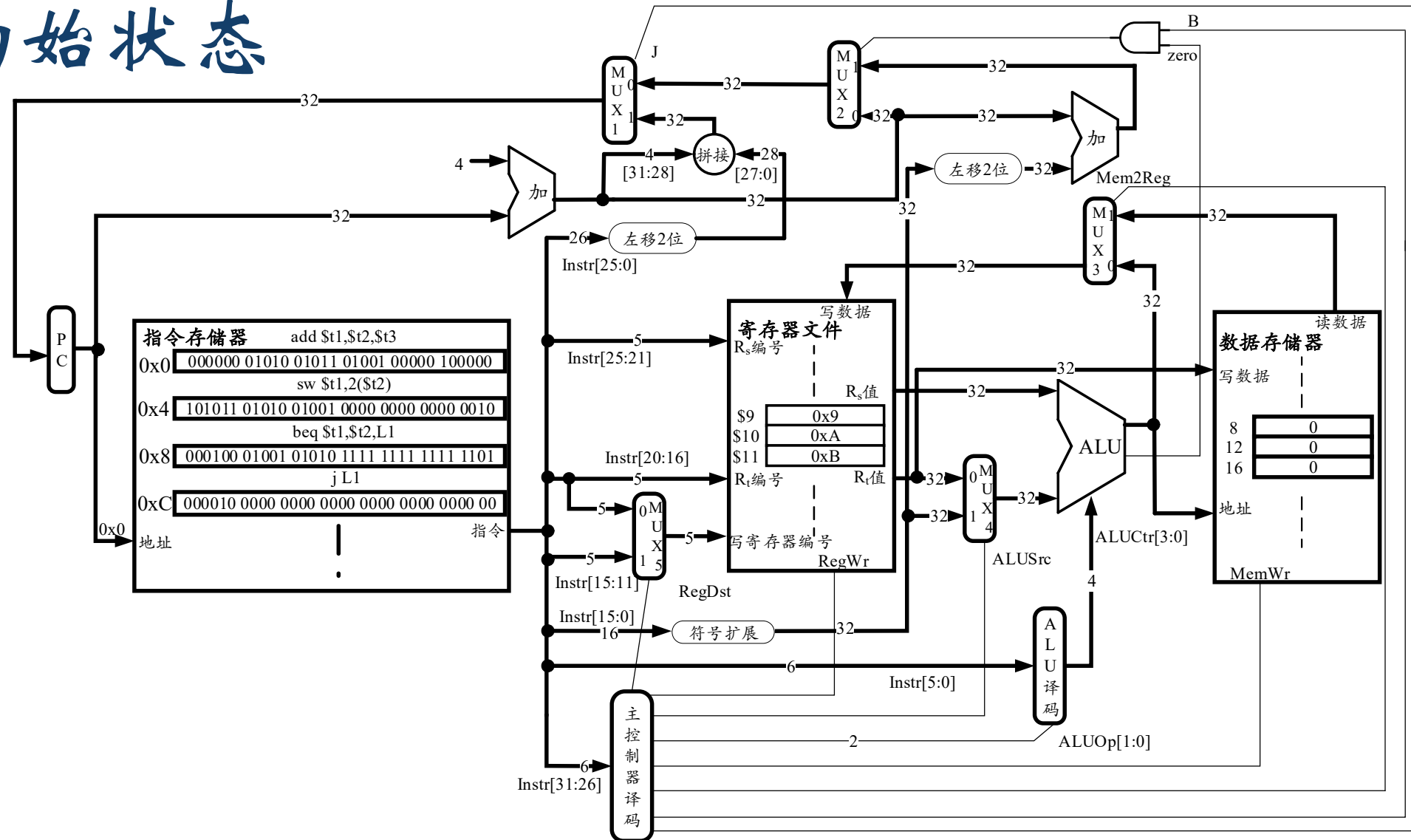
存储地址

机器码

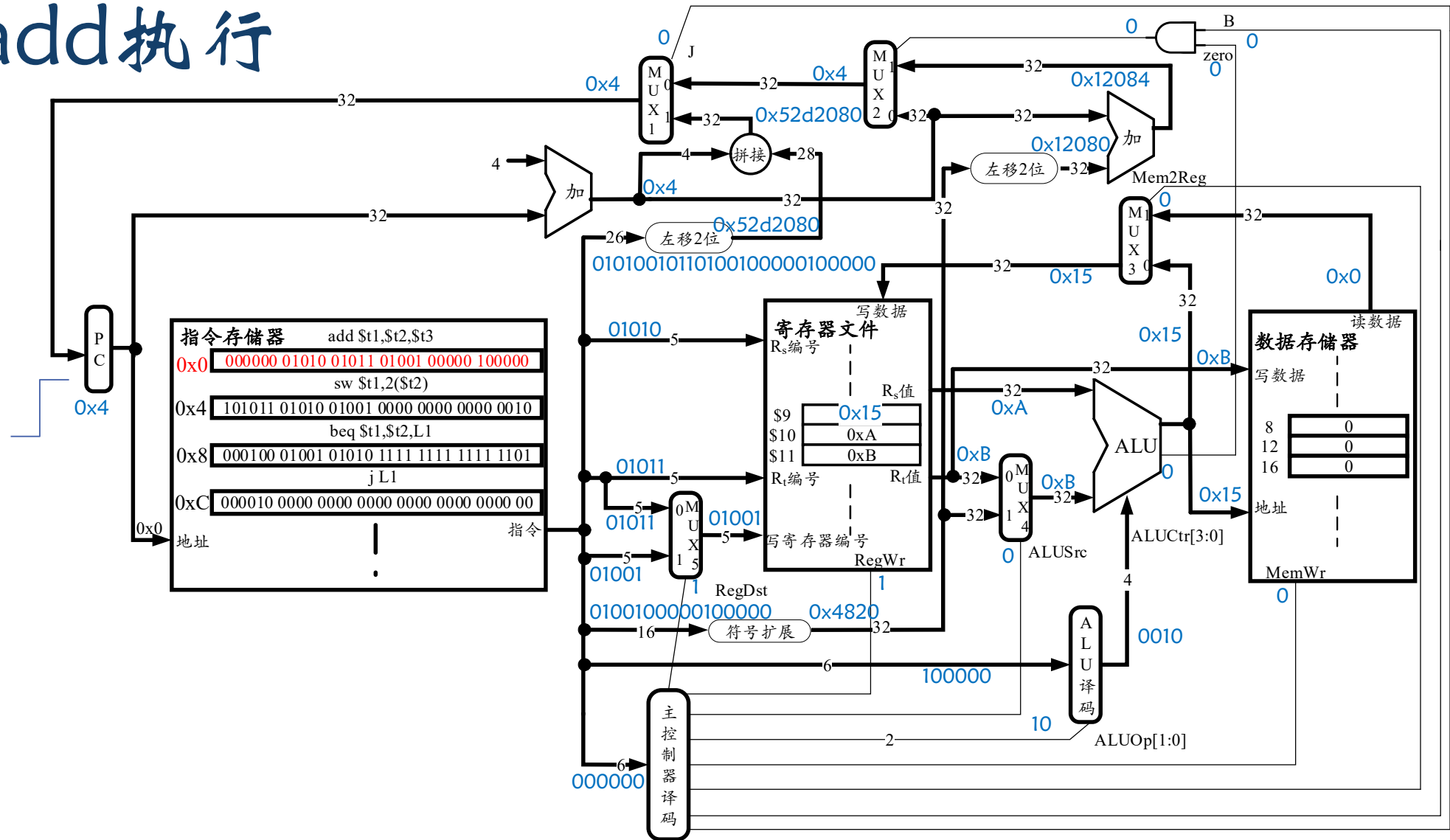
0x0	000000	01010	01011	01001	00000	100000
0x4	101011	01010	01001	0000 0000 0000 0010		
0x8	000100	01001	01010	1111 1111 1111 1101		
0xC	000010	0000 0000 0000 0000 0000 0000 00				

指令存储器

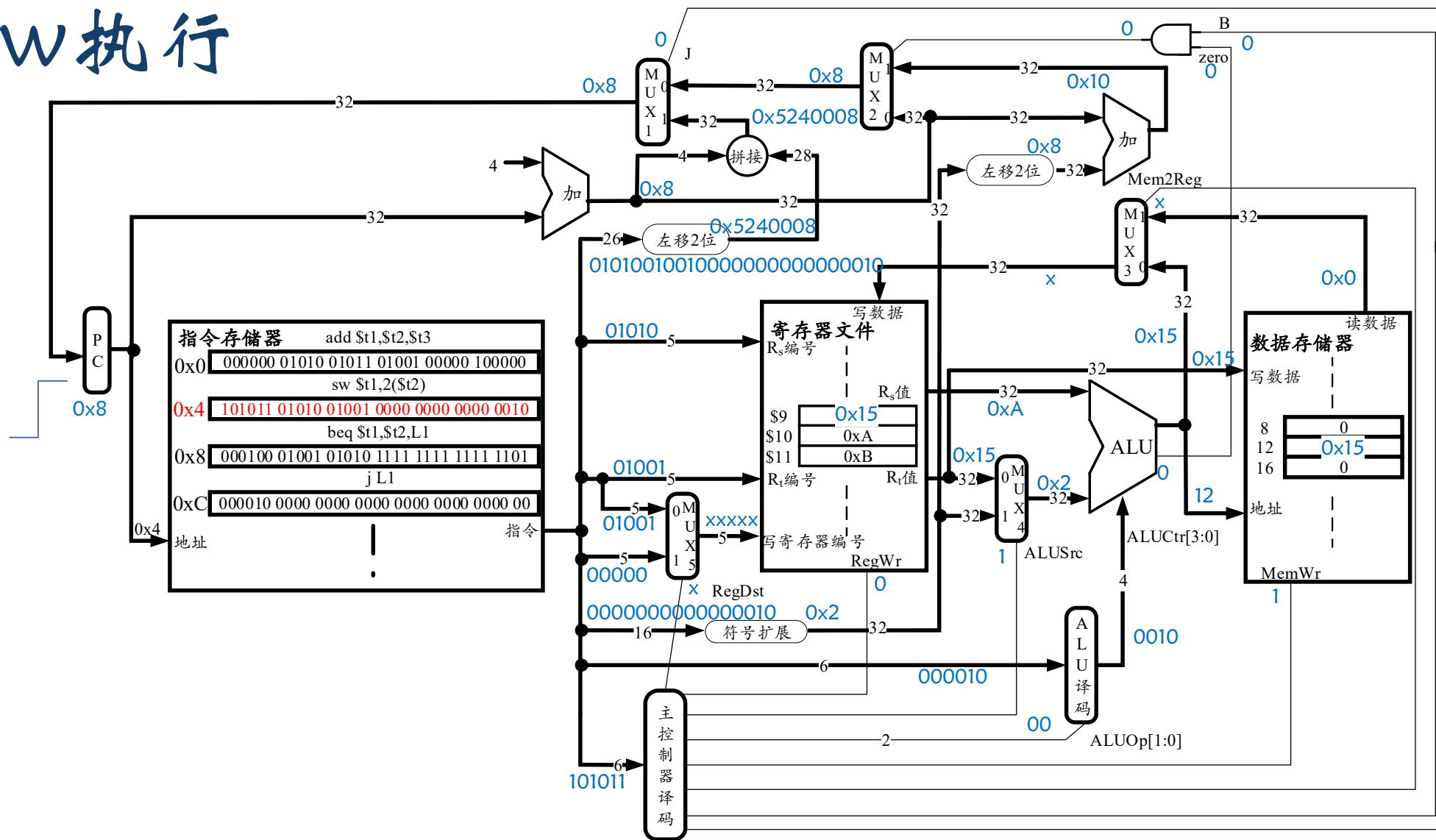
初始状态



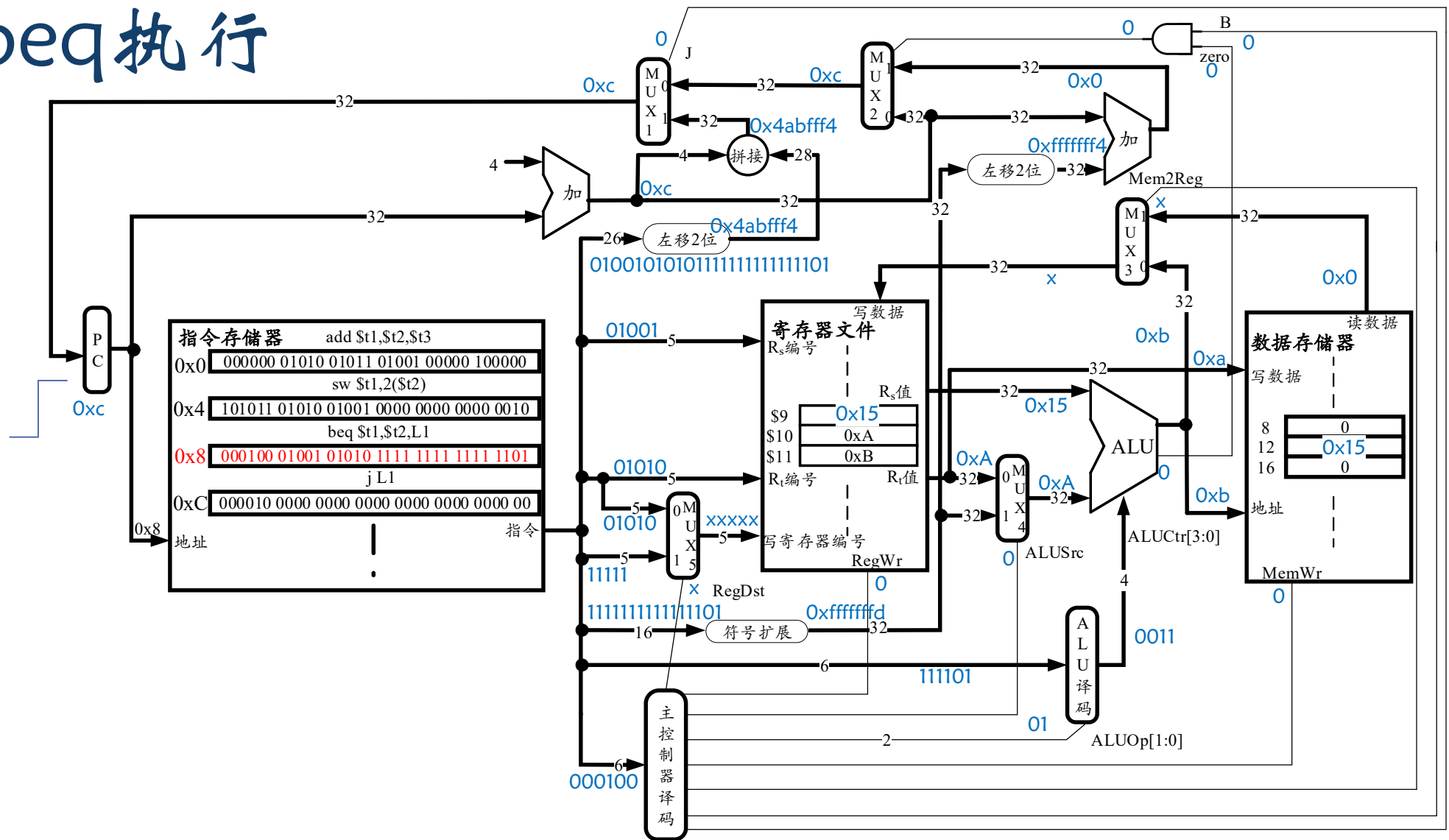
add 执行



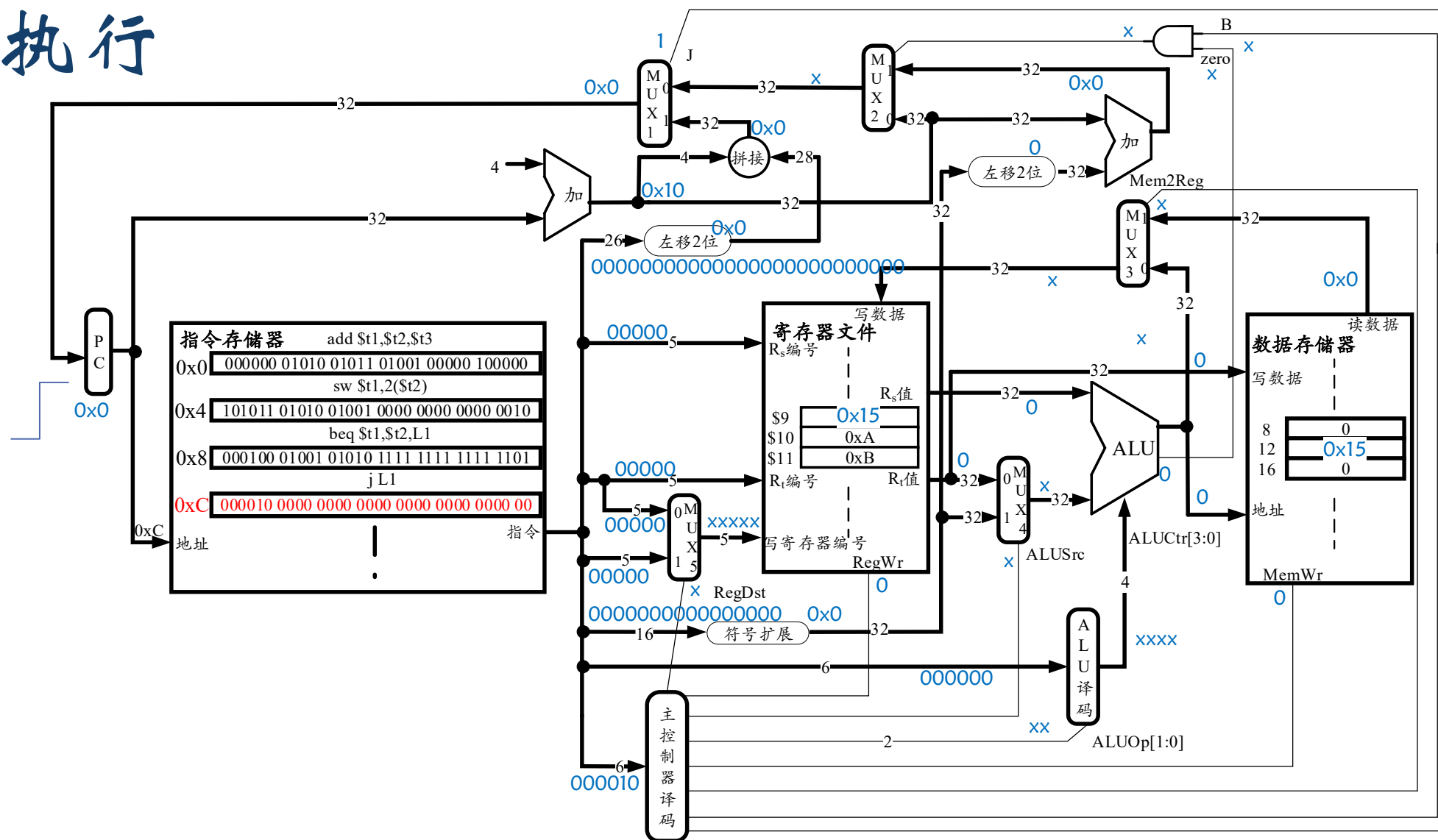
SW 执行



beq 执行



j 执行



小结

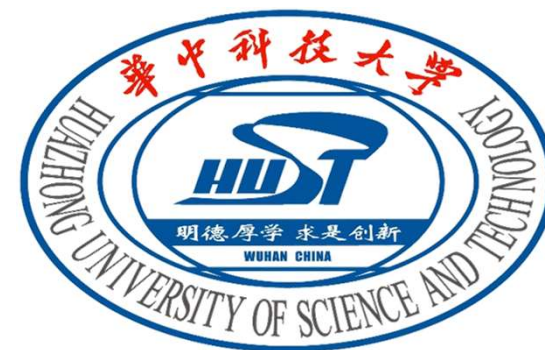
- 回顾微处理器微结构
- 不同指令执行时数据通路以及控制信号取值

下一讲：微处理器新技术

微机原理与接口技术

微处理器新技术

华中科技大学 左冬红



微处理器新技术

目标：实现并行处理

- 流水线

- 把一个重复的过程分解为若干子过程，前一子过程为下一子过程创造执行条件，每一个子过程可以与其它子过程同时进行

指令内并行

- 超标量

- 集成多个ALU、多个译码器和多条流水线，以并行处理的方式提高性能

指令级并行

- 多核

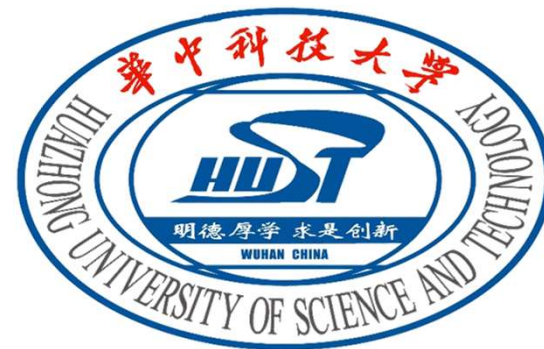
- 在一个芯片上集成两个或多个完整的计算引擎(逻辑CPU核)

程序级并行

微机原理与接口技术

微处理器异常处理机制

华中科技大学 左冬红



异常

计算机正常工作是指在预先存储的程序控制下一步步按照程序执行流程执行程序中的指令。

用户干预

设备故障

非法指令

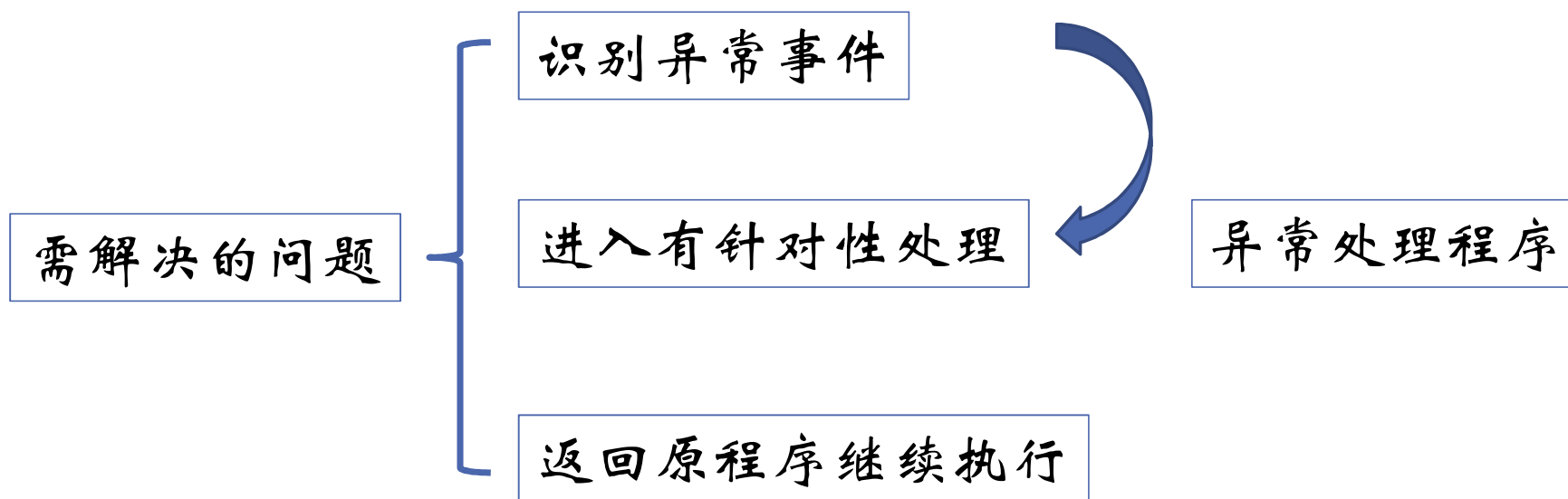
外设交互

异常事件

异常事件响应措施：异常处理机制

异常处理原则

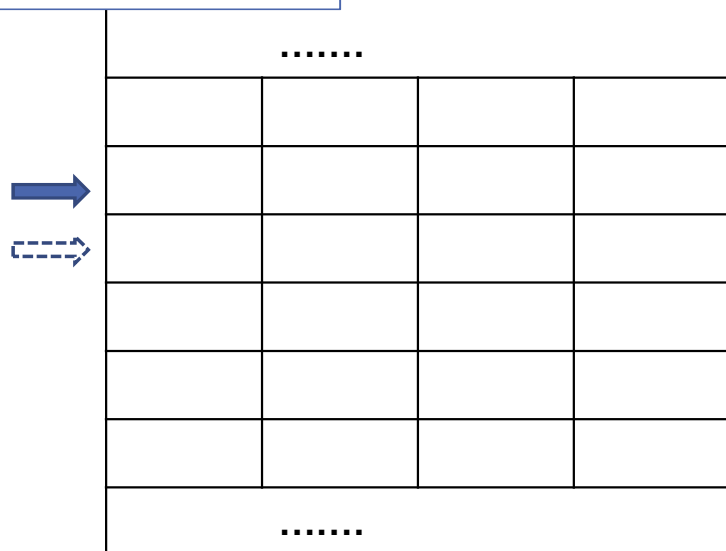
不同异常事件需有针对性的处理，处理结束返回原程序且不破坏原程序的正常执行。



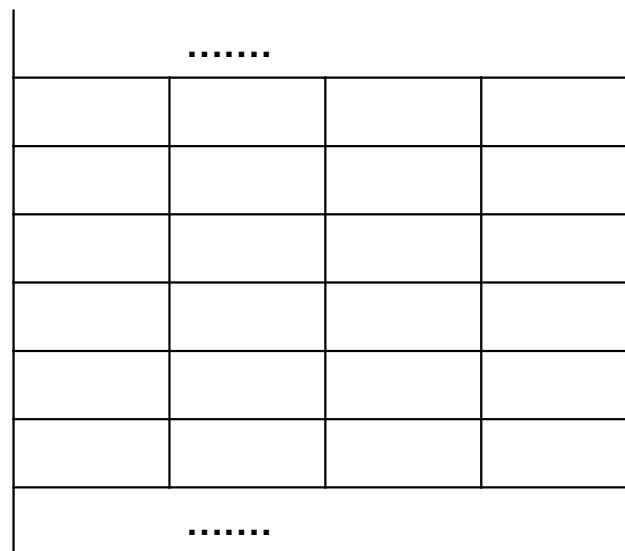
原程序无法预知异常事件，即不能由原程序调用异常处理程序，需由微处理器硬件直接调用异常处理程序

异常处理流程

正常执行流程



异常发生后



正常执行程序

异常处理程序

事先并不知道异常处理程序的存在

异常事件识别

硬件电路实现识别

每个异常事件都有状态信号，电路保持此状态

状态位法

每个异常事件都有状态信号，电路保持此状态，
且对事件的状态编码

类型码法

返回原程序

原程序被异常事件中断时，PC指向下一条指令，保存PC的值，即可返回

特殊寄存器保存

MIPS EPC

栈保存

所有程序的栈空间连续

进入异常处理程序

异常事件发生时，CPU需根据异常事件类型，将异常处理程序的入口地址赋给PC

查表

类似C语言switch语句

硬件实现或软件实现

进入异常处理程序

查表

软件实现

异常事件发生时，直接进入查表程序

异常事件发生时，硬件需将查表程序入口地址赋给PC

即计算机系统中仅一个总中断服务程序，为便于硬件实现，通常该程序入口地址为固定值

异常发生时， $PC = \text{固定值}$

进入异常处理程序

查表

硬件实现

通过跳转指令跳转到
真正异常处理程序

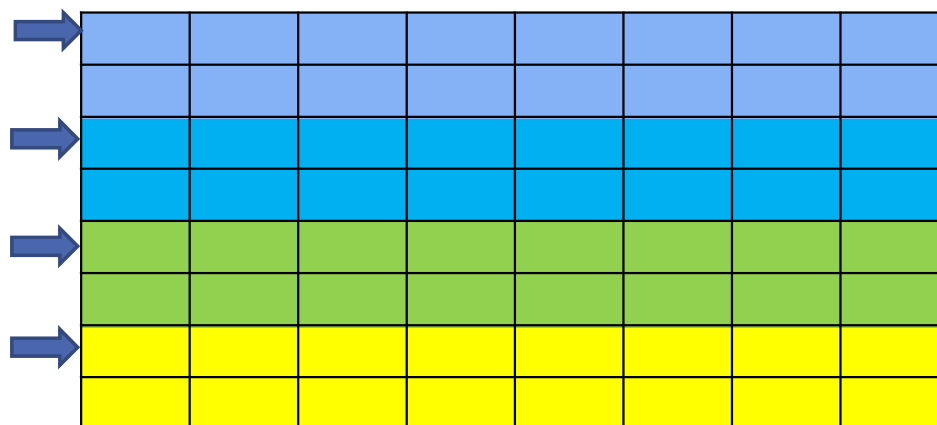
异常处理程序存放具某种规则，如在存储器中
依序存放，且大小一致，如都为16个字节

异常处理程序0

异常处理程序1

异常处理程序2

异常处理程序3



异常发生时， $PC=16 \times n$

n 为异常事件类型码

进入异常处理程序

查表

硬件实现

异常处理程序大小一般很难一致，硬件查地址表

异常处理程序0入口地址
异常处理程序1入口地址
异常处理程序2入口地址
异常处理程序3入口地址

异常发生时， $PC = \text{mem}[4 \times n]$

n 为异常事件类型码

小结

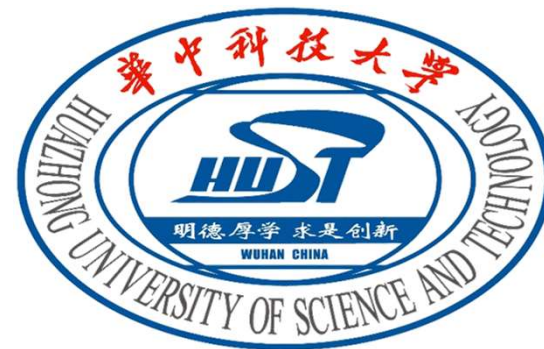
- 异常处理是微处理器必不可少的功能
- 异常事件识别（硬件）
- 断点保存（硬件）
- 进入异常处理程序机制（查地址表）
 - 软、硬件协同
 - 纯硬件

下一讲：微处理器外部接口

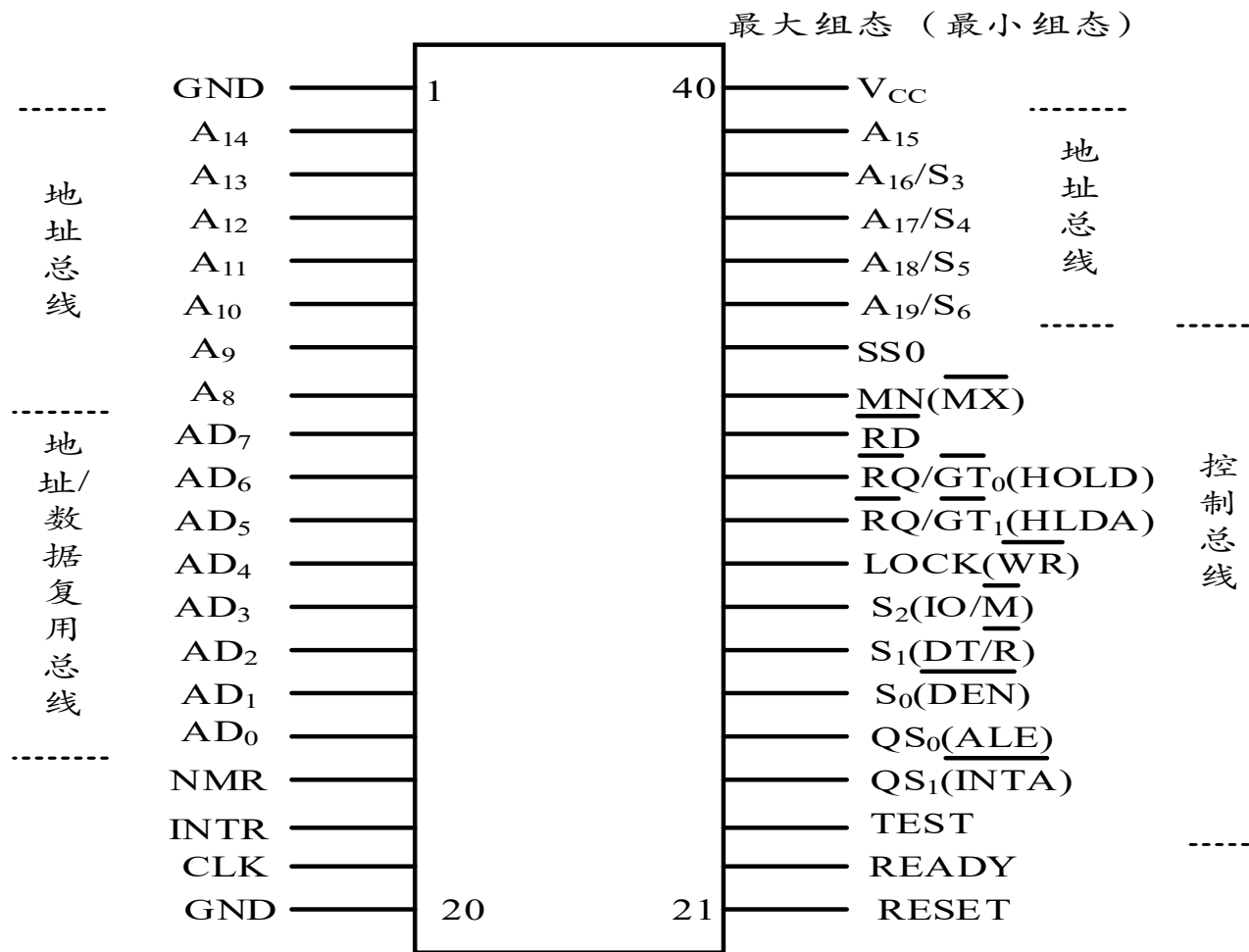
微机原理与接口技术

微处理器外部接口

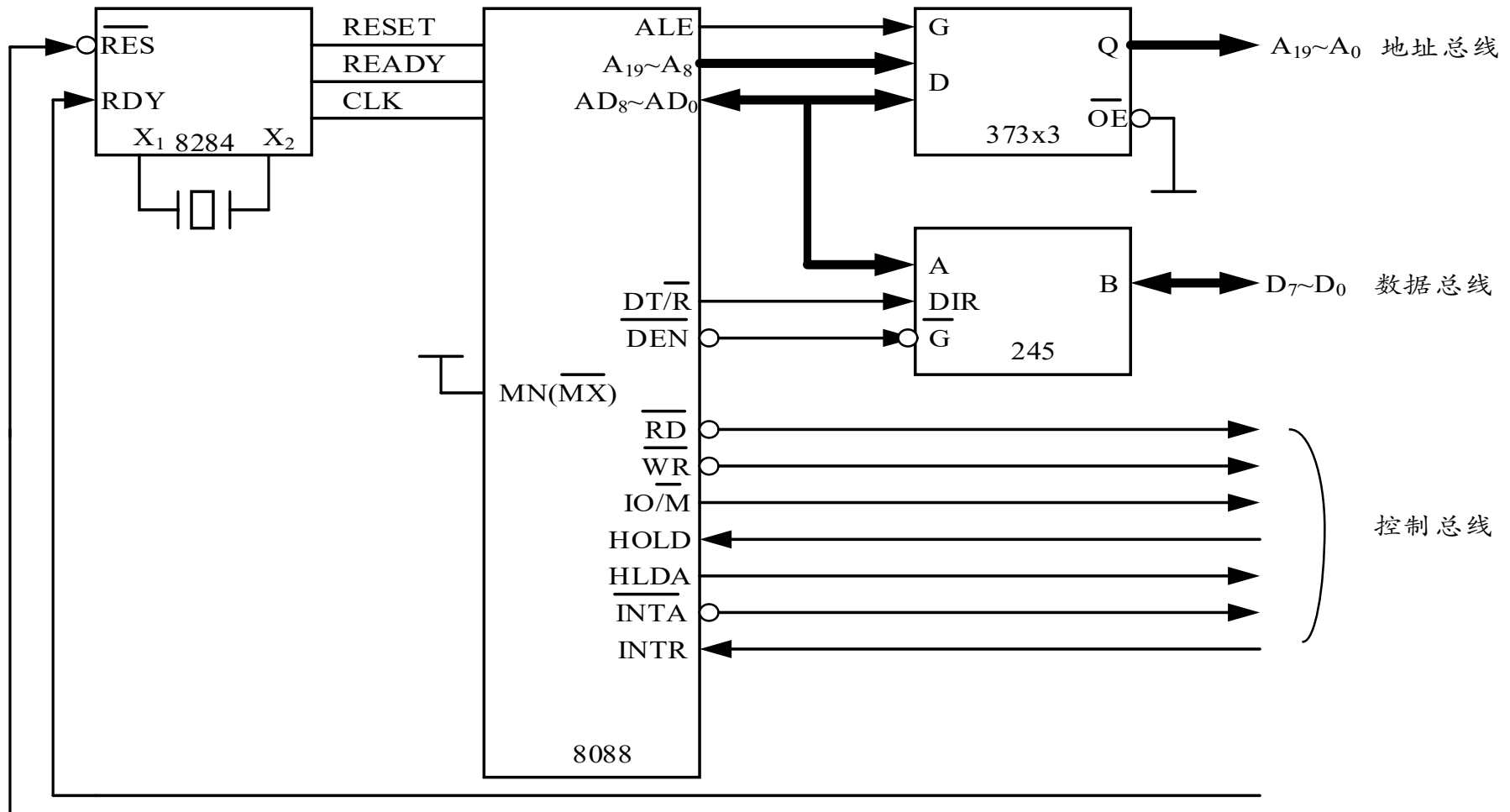
华中科技大学 左冬红



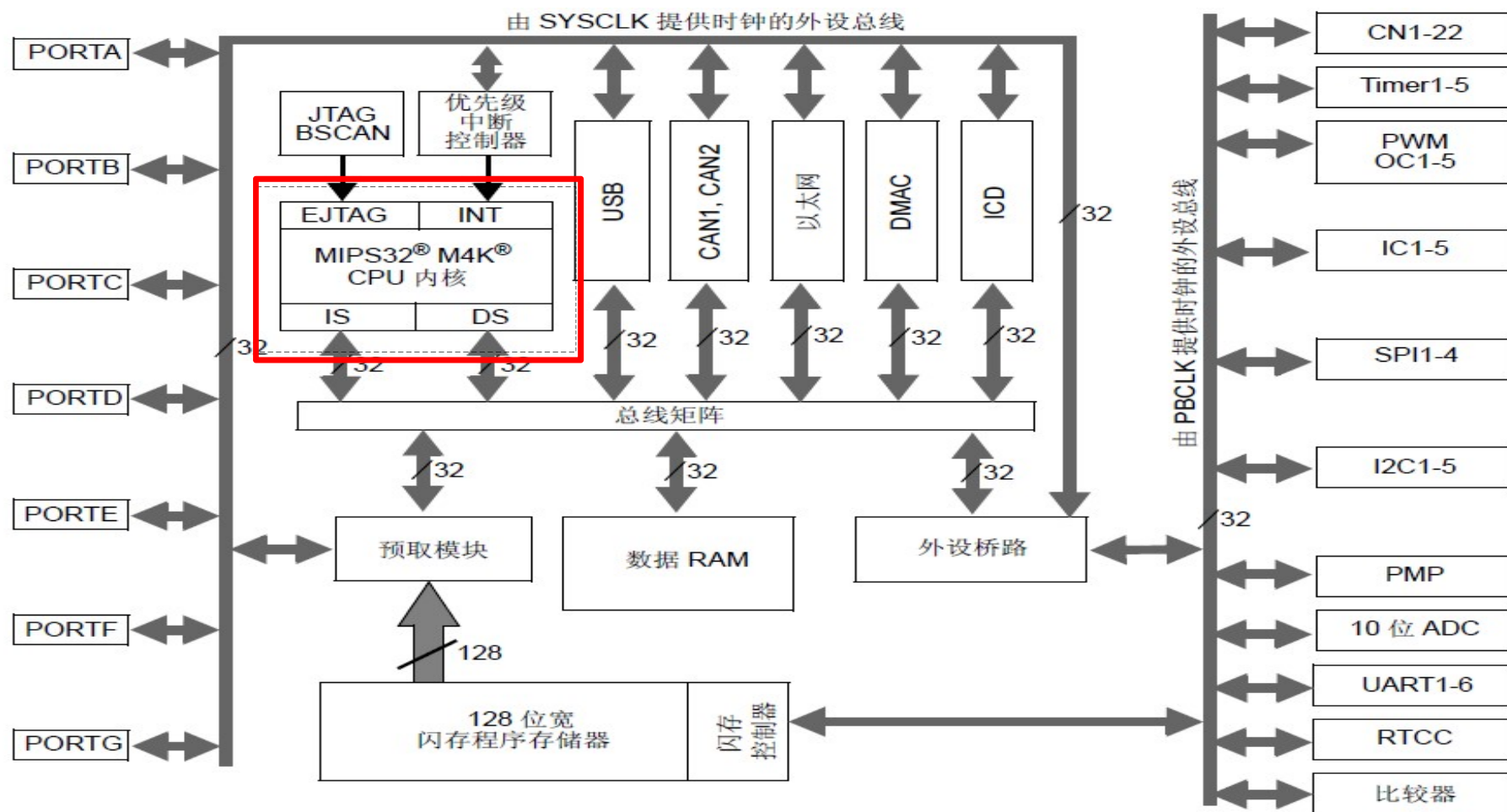
通用微处理器-Intel 8088



Intel 8088 总线



嵌入式微控制器-PLC (Microchip单片机)



小结

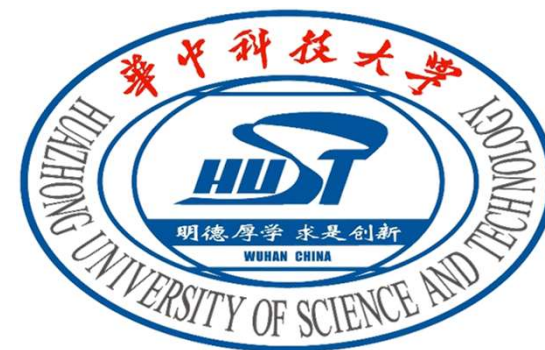
- PC机的CPU通过总线与计算机系统其他模块连接
- 嵌入式微控制器（单片机）
 - 集成了一个计算机的硬件系统
 - 内部通过总线互联
 - 与外设之间通过外部总线互联

下一讲：MicroBlaze软核微处理器

微机原理与接口技术

MicroBlaze软核微处理器 及最小系统

华中科技大学 左冬红

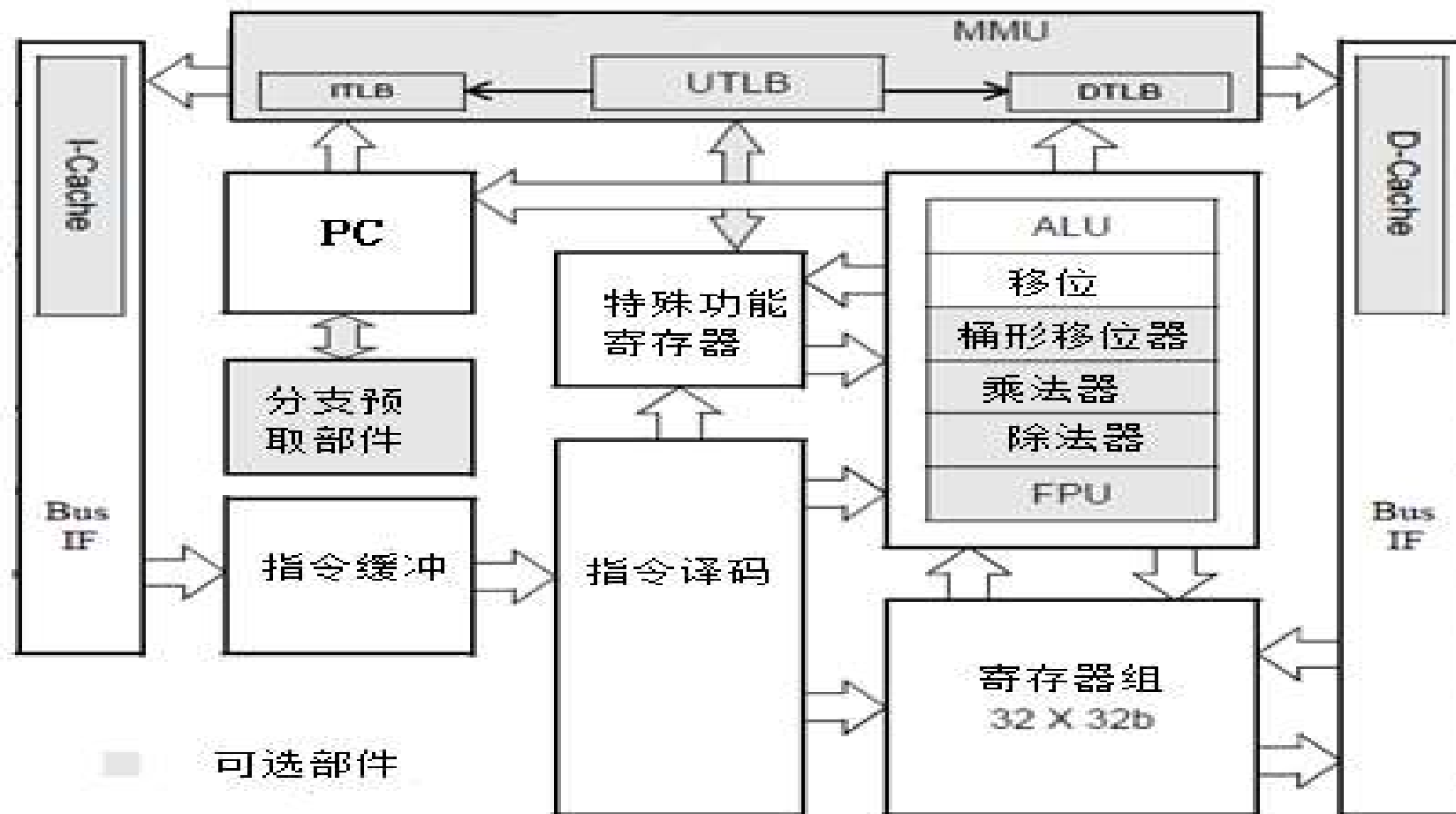


术语

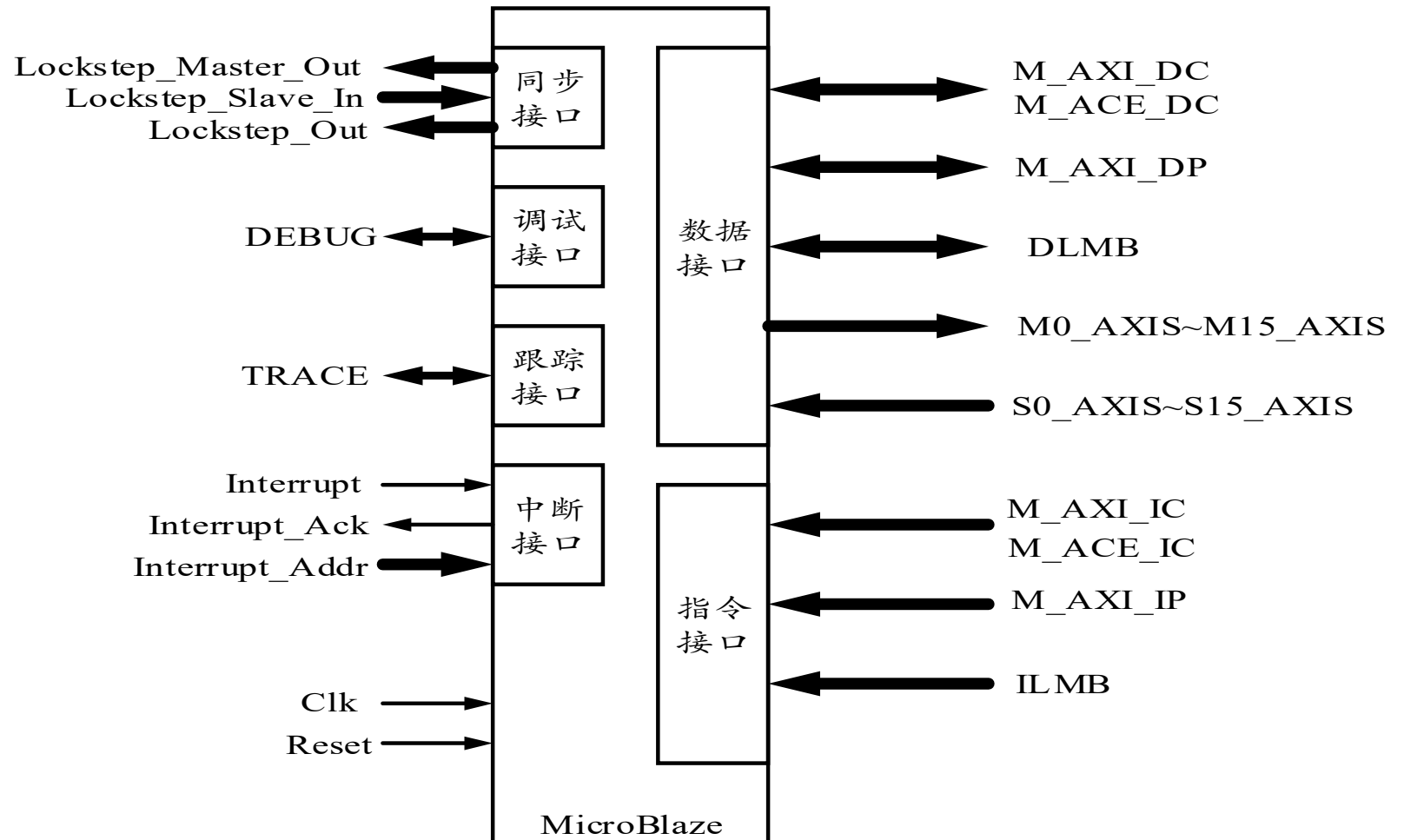
软核 采用硬件描述语言设计，在可编程逻辑器件上实现，具有知识产权的硬件电路模块

硬核 由硬布线完成的硬件电路模块，一旦布线完成，模块内布线不能修改

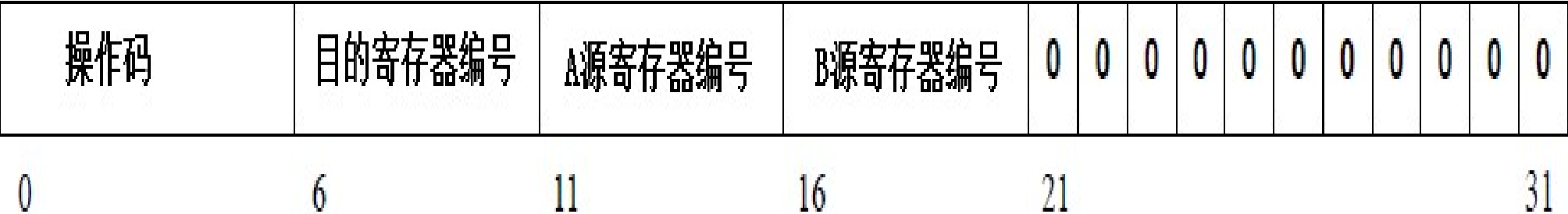
MicroBlaze软核结构



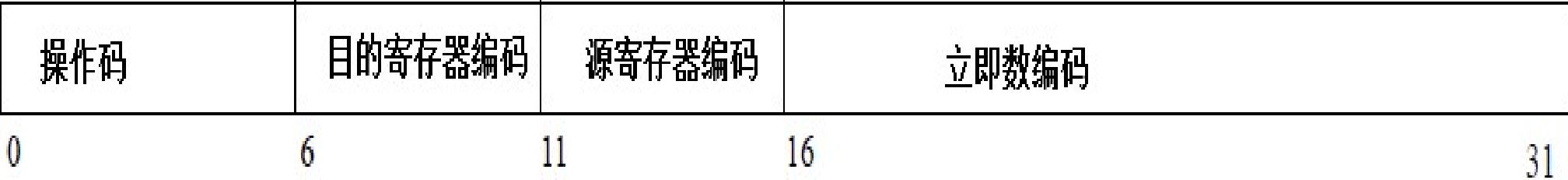
MicroBlaze软核外部接口



MicroBlaze软核指令架构



A型指令 类似R型指令



B型指令 类似I型指令

MicroBlaze软核寄存器

具有32个32位通用寄存器, R0~R31

具有18个32位特殊功能寄存器, PC、MSR等, 是否存在与配置相关

嵌入式计算机系统的基本构成

计算机硬件系统的基本构成：

CPU

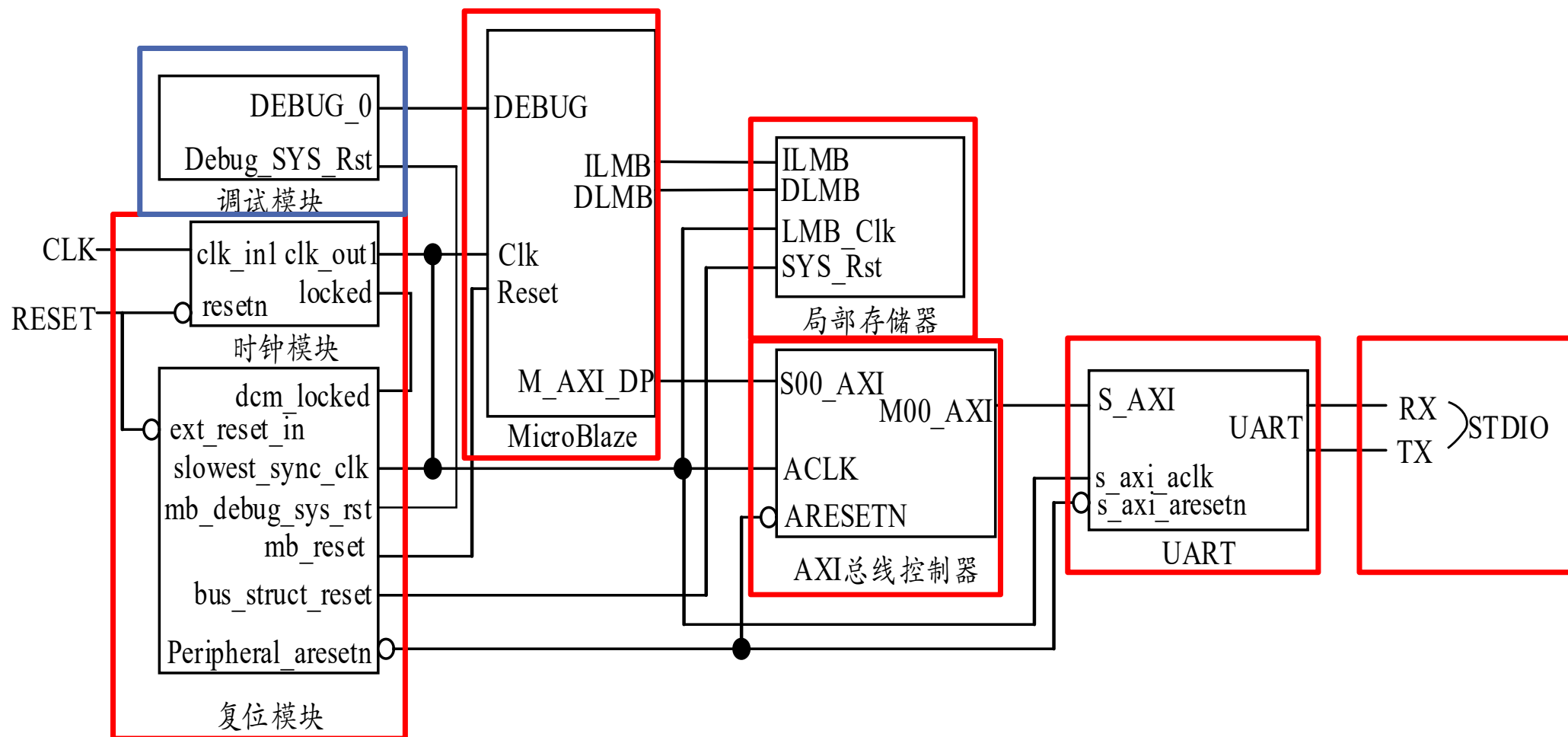
存储器

总线

IO接口

IO设备

嵌入式计算机系统的基本构成



小结

- 软核
- MicroBlaze软核微处理器结构
- 嵌入式计算机系统硬件基本构成
 - 最小系统构成
 - 调试接口
 - UART作为STDIO
 - 片内系统

下一讲：计算机存储系统结构