第四讲 控制器设计

翟象平

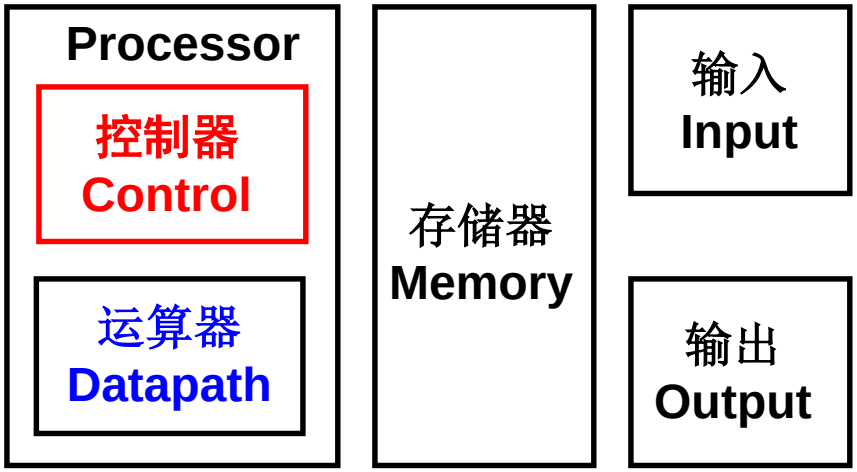
电邮: blueicezhaixp@nuaa.edu.cn

个人主页: http://cyber.nuaa.edu.cn

取指令部件(Instruction Fetch Unit)

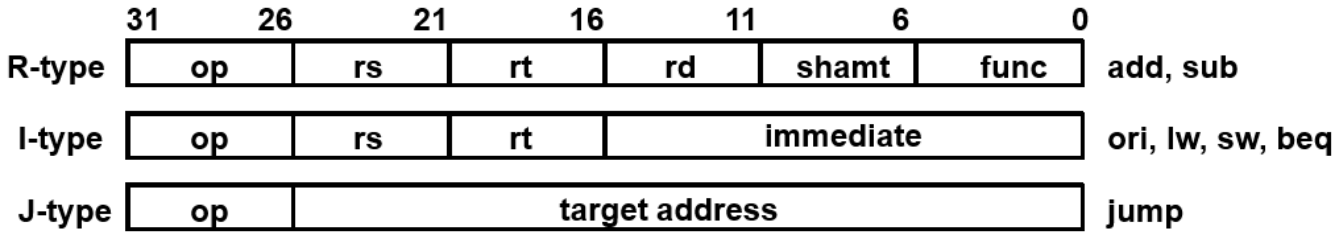
- 计算机五大组成

- 设计控制器
  - 根据每条指令的功能，分析控制信号的取值，并在表中列出
  - 根据列出的指令和控制信号的关系，写出每个控制信号的逻辑表达式



指令与控制信号的关系表

func / op	10 0000 / 00 0000	10 0010 / 00 0000	与func字段无关！				
	add	sub	ori	lw	sw	beq	jump
RegDst	1	1	0	0	x	x	x
ALUSrc	0	0	1	1	1	0	x
MemtoReg	0	0	0	1	x	x	x
RegWrite	1	1	1	1	0	0	0
MemWrite	0	0	0	0	1	0	0
Branch	0	0	0	0	0	1	0
Jump	0	0	0	0	0	0	1
ExtOp	x	x	0	1	1	x	x
ALUctr<2:0>	add	sub	or	addu	addu	subu	xxx



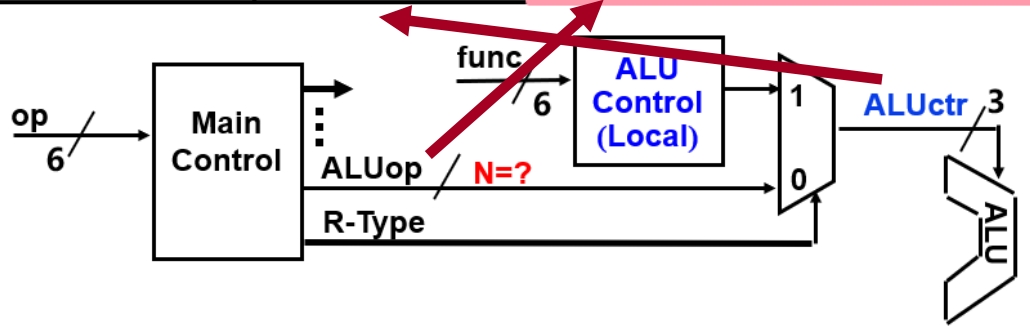
主控单元和ALU局控单元

op	00 0000	00 1101	10 0011	10 1011	00 0100	00 0010
	R-type	ori	lw	sw	beq	jump
RegDst	1	0	0	x	x	x
ALUSrc	0	1	1	1	0	x
MemtoReg	0	0	1	x	x	x
RegWrite	1	1	1	0	0	0
MemWrite	0	0	0	1	0	0
Branch	0	0	0	0	1	0
Jump	0	0	0	0	0	1
ExtOp	x	0	1	1	x	x
ALUctr	Add/Sub	Or	Addu	Addu	Subu	xxx

ALUctr的值：

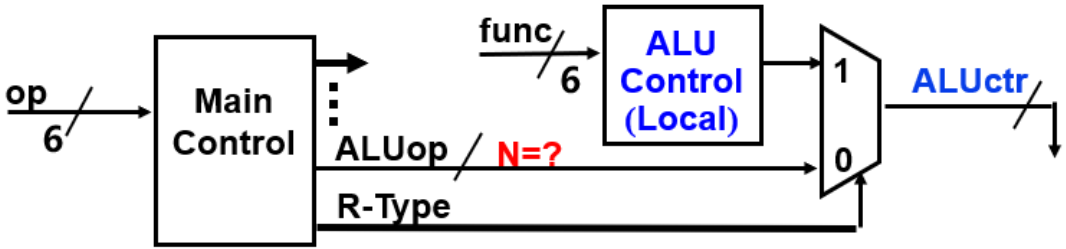
非R型指令时，取决于ALUop；

R型指令时，取决于func。



ALUop的输出位数N应为3！

ALUop的逻辑表达式



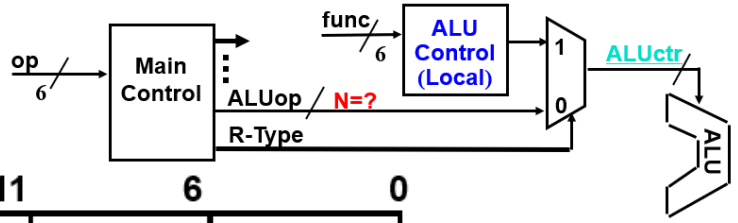
	000000	001101	100011	101011	000100	000010
指令	R-type	ori	lw	sw	beq	jump
运算	由func指定	or	addu	addu	subu	xxx
ALUop<2:0>	xxx 或 xx1	010	000	000	100	xxx

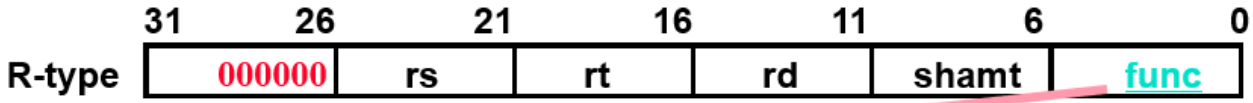
- ALUop<2>=beq=!op<5>&!op<4>&!op<3>&op<2>&!op<1>&!op<0> (op=000100)
- ALUop<1>=ori=!op<5>&!op<4>&op<3>&op<2>&!op<1>&op<0> (op=001101)
- ALUop<0>= R-type=!op<5>&!op<4>&!op<3>&!op<2>&!op<1>&!op<0> (op=000000)

ALU控制信号

ALUctr<2:0>	操作类型	SUBctr	OVctr	SIGctr	OPctr<1:0>	OPctr 的含义
0 0 0	addu	0	0	x	0 0	选择加法器的结果输出
0 0 1	add	0	1	x	0 0	选择加法器的结果输出
0 1 0	or	x	0	x	0 1	选择“按位或”结果输出
0 1 1	（未用）					
1 0 0	subu	1	0	x	0 0	选择加法器的结果输出
1 0 1	sub	1	1	x	0 0	选择加法器的结果输出
1 1 0	sltu	1	0	0	1 0	选择小于置位结果输出
1 1 1	slt	1	0	1	1 0	选择小于置位结果输出

ALU局部控制表达式

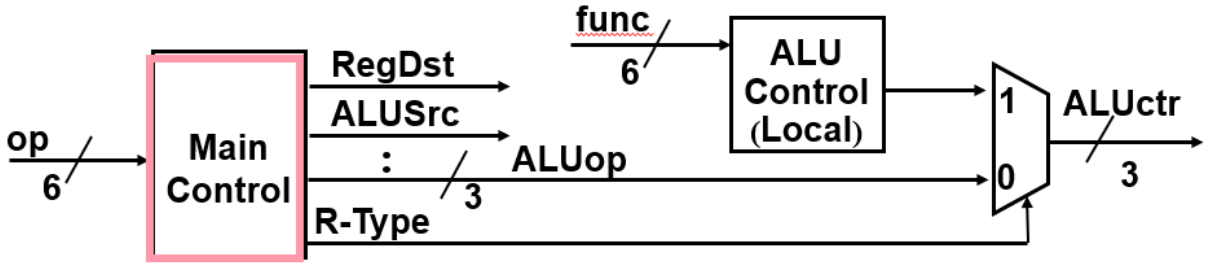




func<5:0>	Instruction Operation	ALUctr<2:0>	ALU Operation
10 0000	add	001	Add
10 0010	sub	101	Sub
10 0100	and	-----	And
10 0101	or	010	Or
10 1010	set-on-less-than	111	Slt

- ALUctr<0> = !func<3> & !func<2> & !func<1> & !func<0> + !func<2> & func<1> & !func<0>
- ALUctr<1> = !func<3> & func<2> & !func<1> & func<0> +func<3> & !func<2> & func<1> & !func<0>
- ALUctr<2> = !func<2> & func<1> & !func<0>

ALU局部控制表达式



op	00 0000	00 1101	10 0011	10 1011	00 0100	00 0010
	R-type	ori	lw	sw	beq	jump
RegDst	1	0	0	x	x	x
ALUSrc	0	1	1	1	0	x
MemtoReg	0	0	1	x	x	x
RegWrite	1	1	1	0	0	0
MemWrite	0	0	0	1	0	0
Branch	0	0	0	0	1	0
Jump	0	0	0	0	0	1
ExtOp	x	0	1	1	x	x
ALU运算	“R-type”	Or	Addu	Addu	Subu	xxx
ALUop <2>	x	0	0	0	1	x
ALUop <1>	x	1	0	0	0	x
ALUop <0>	1	0	0	0	0	x
R-type	1	0	0	0	0	0

控制信号逻辑方程（RegWr）

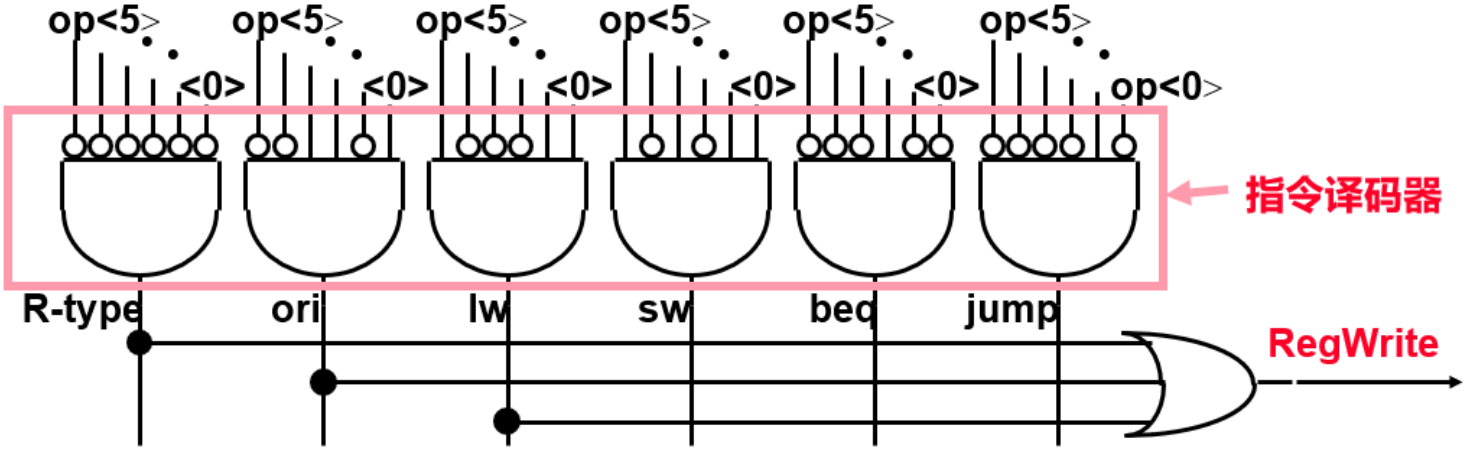
op	00 0000	00 1101	10 0011	10 1011	00 0100	00 0010
	R-type	ori	lw	sw	beq	jump
RegWrite	1	1	1	0	0	0

- RegWrite = R-type + ori + lw

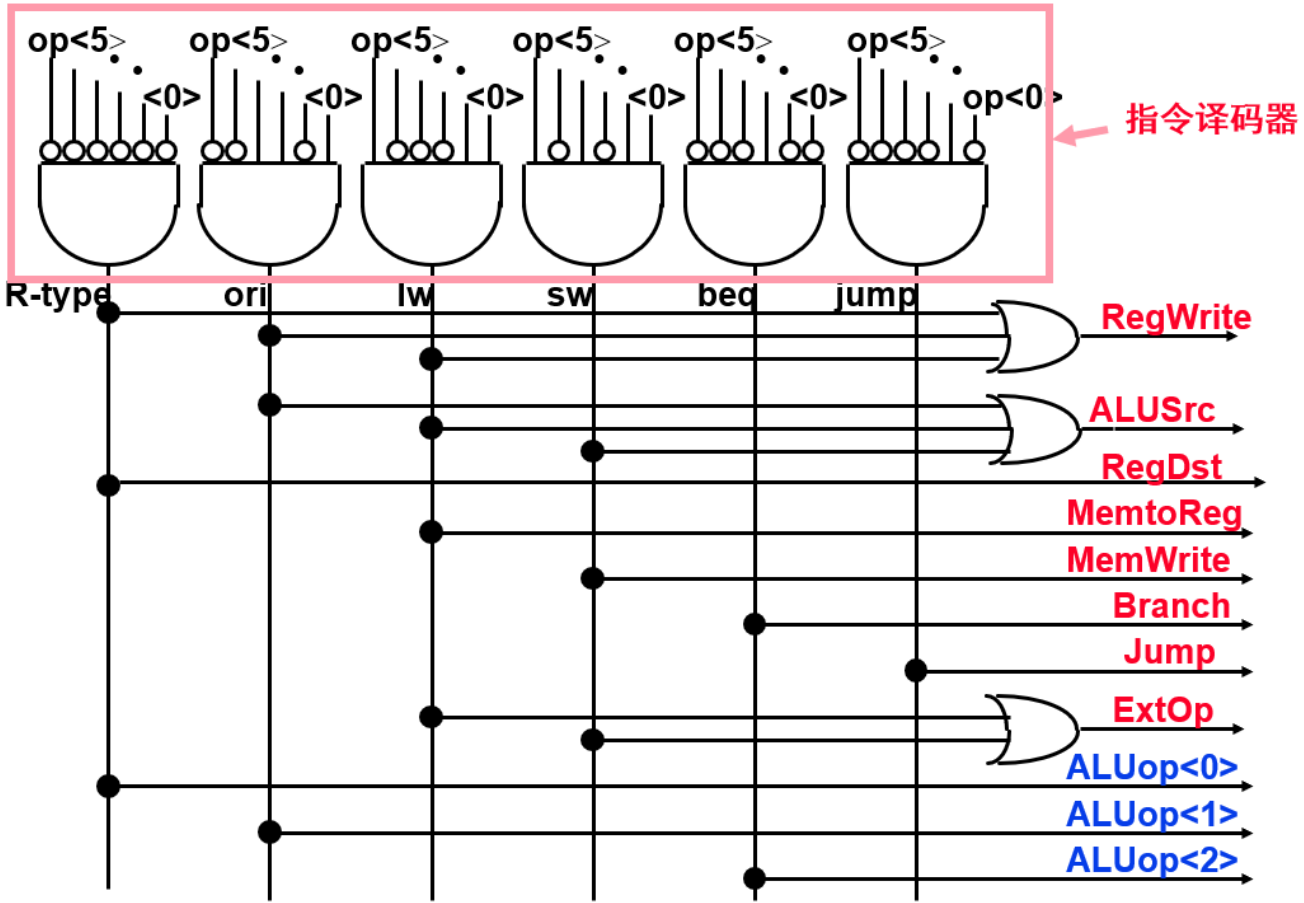
  = !op<5> & !op<4> & !op<3> & !op<2> & !op<1> & !op<0> (R-type)

  + !op<5> & !op<4> & op<3> & op<2> & !op<1> & op<0> (ori)

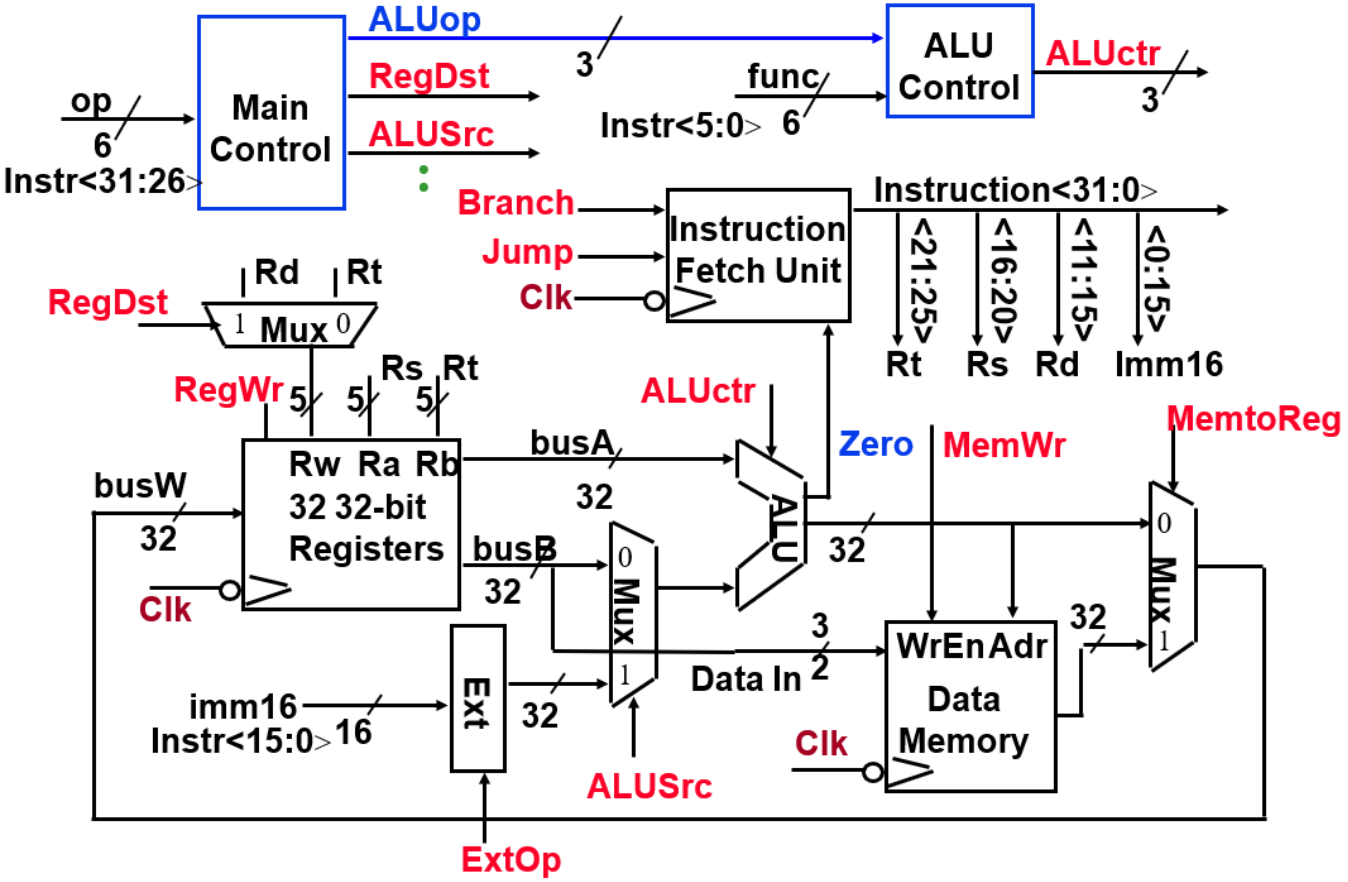
  + op<5> & !op<4> & !op<3> & !op<2> & op<1> & op<0> (lw)



主控的可编程逻辑阵列实现



7条指令的完整单周期处理器



Q & A

THANKS