3.5.3 单周期MIPS32处理器

（目标指令集与数据通路设计）

※CPU设计的主要任务

①拟定指令集 √

②数据通路设计

③控制器设计

1、目标指令集（共计11条）

序号	类型	指令	功能操作	寻址方式
1	R型运算	add rd, rs, rt	$rs+$rt→$rd	R直接寻址
2		sub rd, rs, rt	$rs-$rt→$rd	
3		and rd, rs, rt	$rs and $rt→$rd	
4		or rd, rs, rt	$rs or $rt→$rd	
5	I型运算	addi rt, rs, imm	$rs+E(imm) →$rt	立即数寻址 R基址寻址
6		andi rt, rs, imm	$rs and E(imm) →$rt	
7		ori rt, rs, imm	$rs or E(imm) →$rt	
8	I型访存	lw rt, imm(rs)	Mem[$rs+E(imm)] →$rt	立即数寻址 R基址寻址
9		sw rt, imm(rs)	$rt→Mem[$rs+E(imm)]	
10	I型分支	beq rs, rt, imm	$rs=$rt: PC+4+E(imm)<<2 →PC $rs≠$rt: PC+4→PC	立即数寻址 PC相对寻址
11	J型跳转	j address	(PC+4)[31:28] ∪ (address<<2)	立即数寻址 伪直接寻址

※CPU执行1条指令的步骤：

①取指令

根据PC，从存储器中取出指令，然后PC +4。

②取操作数

根据指令中操作数字段，选择读取寄存器\存储器或立即数，送ALU(运算器)。

③分析指令

将指令中的操作码送控制器，分析指令的功能，产生相应的控制信号。

④执行指令

ALU根据控制器产生的控制信号完成指令规定的操作，并保存结果、修改PC。

※根据指令执行的所需时钟周期数

① 单周期CPU:

指令固定在1个时钟周期内完成。

- √ 时间效率低，时钟宽度由单指令最长时间决定。
- √ 在指令周期内，功能部件不能共享，冗余度大；

② 多周期CPU:

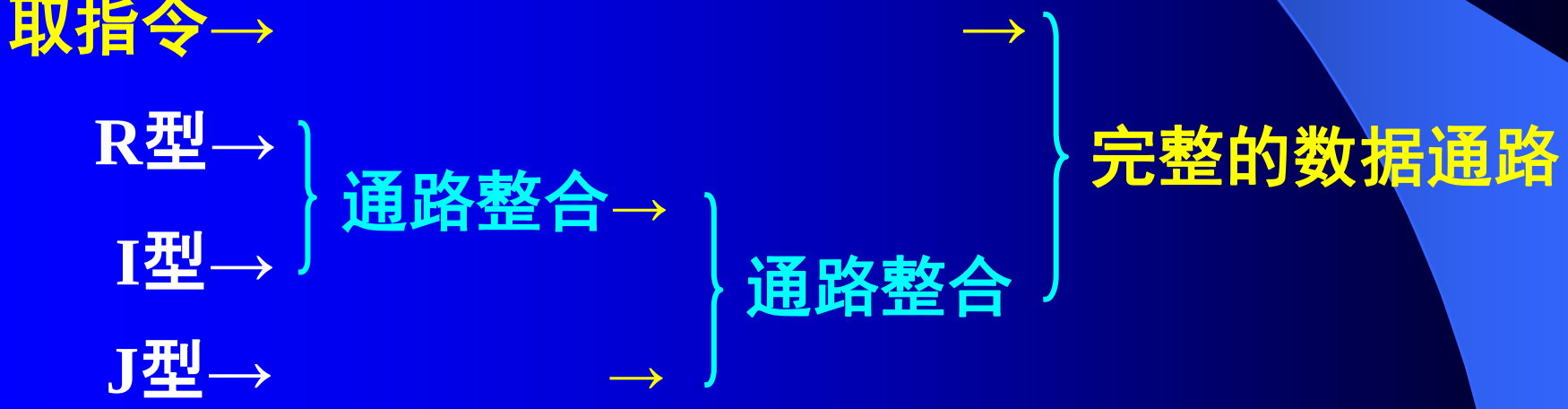
指令分散在多个时钟周期内完成。

- √ 时间效率高，时钟的宽度由单步最长时间决定。
- √ 不同的时钟周期之间，部件可共享，冗余降低。

2、指令的基本数据通路

【基本思路】面向指令功能，逐步扩展、融合

- 分析三类指令的格式和功能
- 选择功能部件，确定部件之间的连接通路
- 整合冗余的部件连线

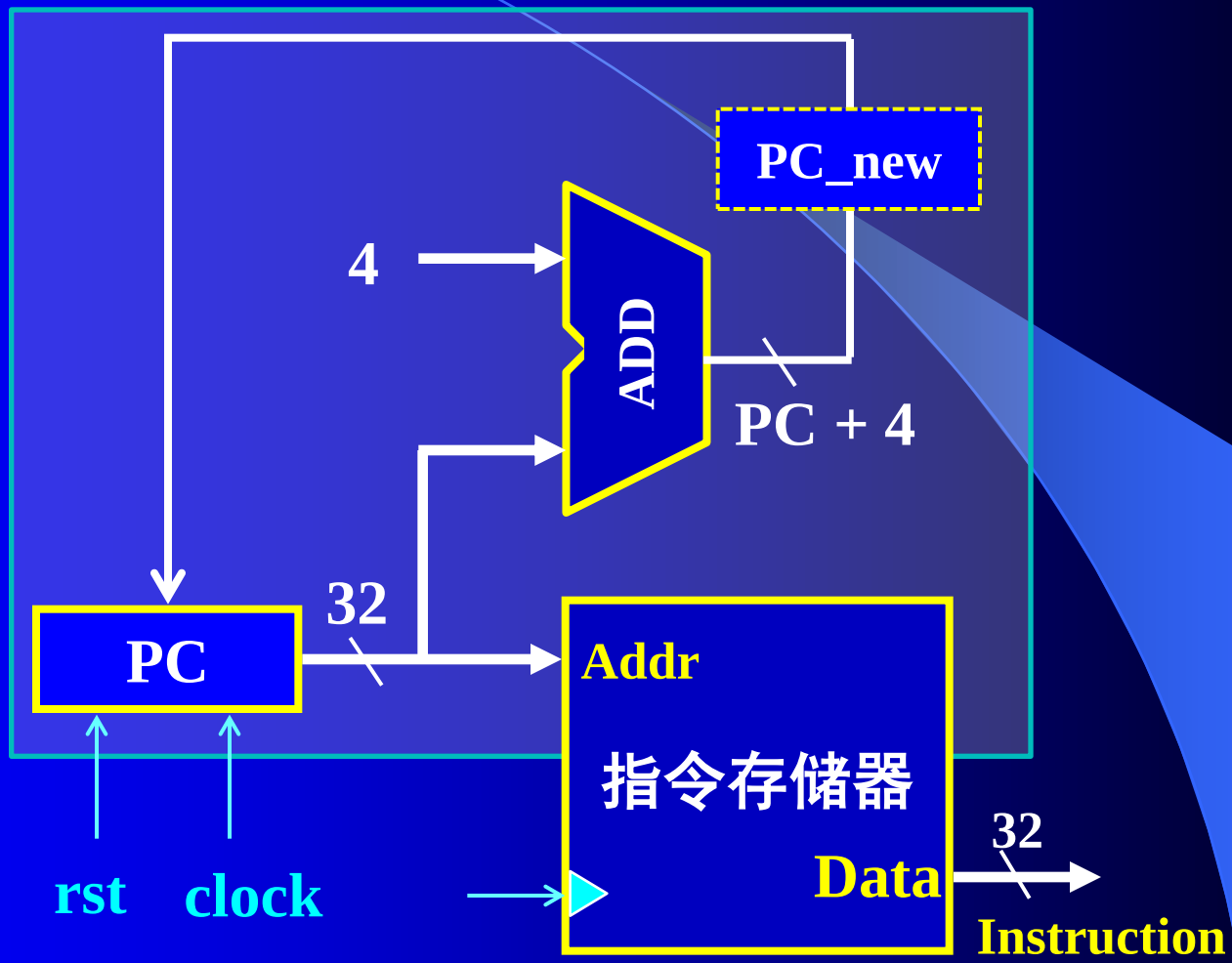


经过多次整合，得到最终的完整数据通路。

（1）取指功能的数据通路（公共）

Instruction ← Mem[PC]

PC ← PC+4



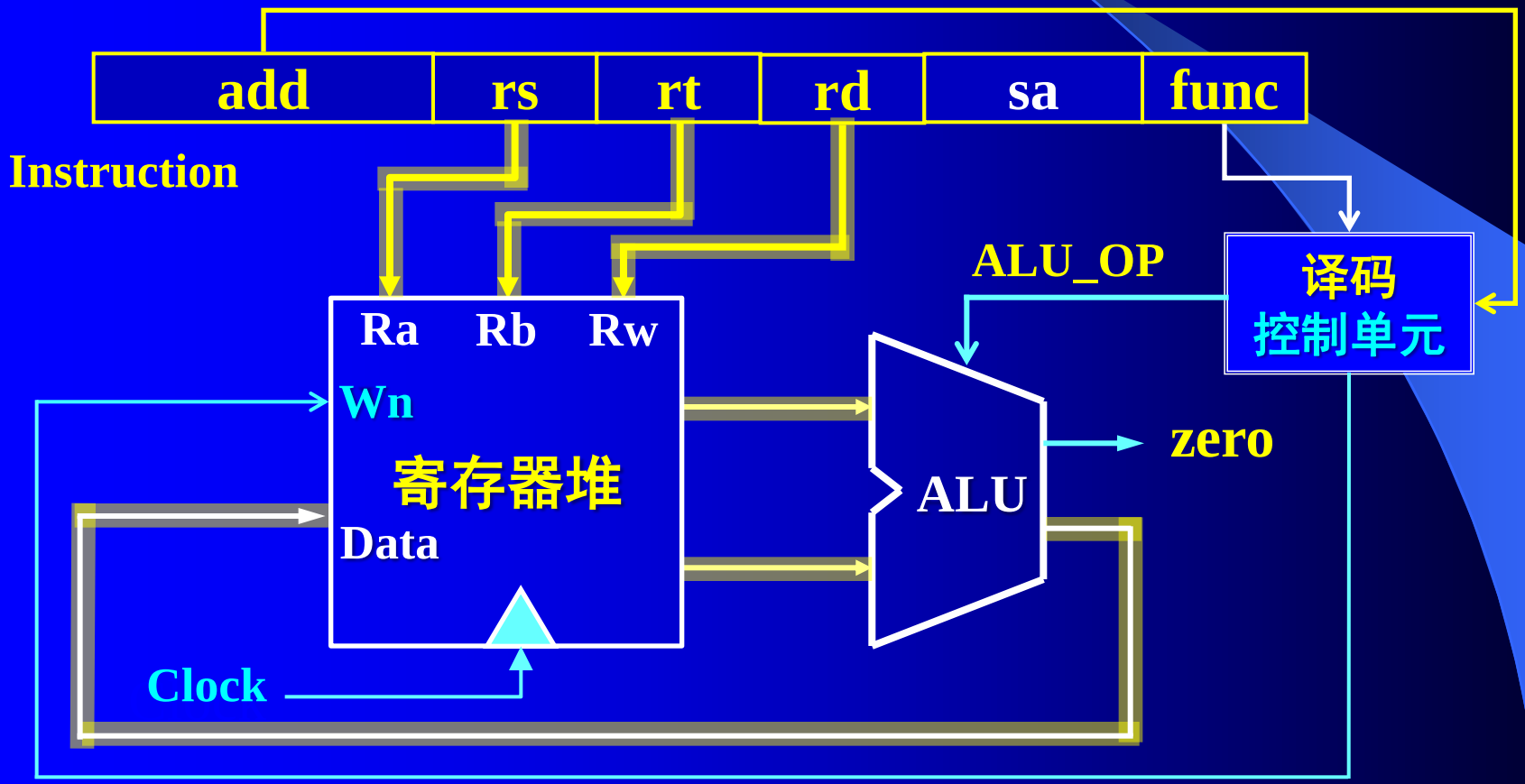
（2）R型运算指令

OP段为6个0，需靠func段确定操作类型；
最多有3个寄存器参与工作；

and rd, rs, rt ?

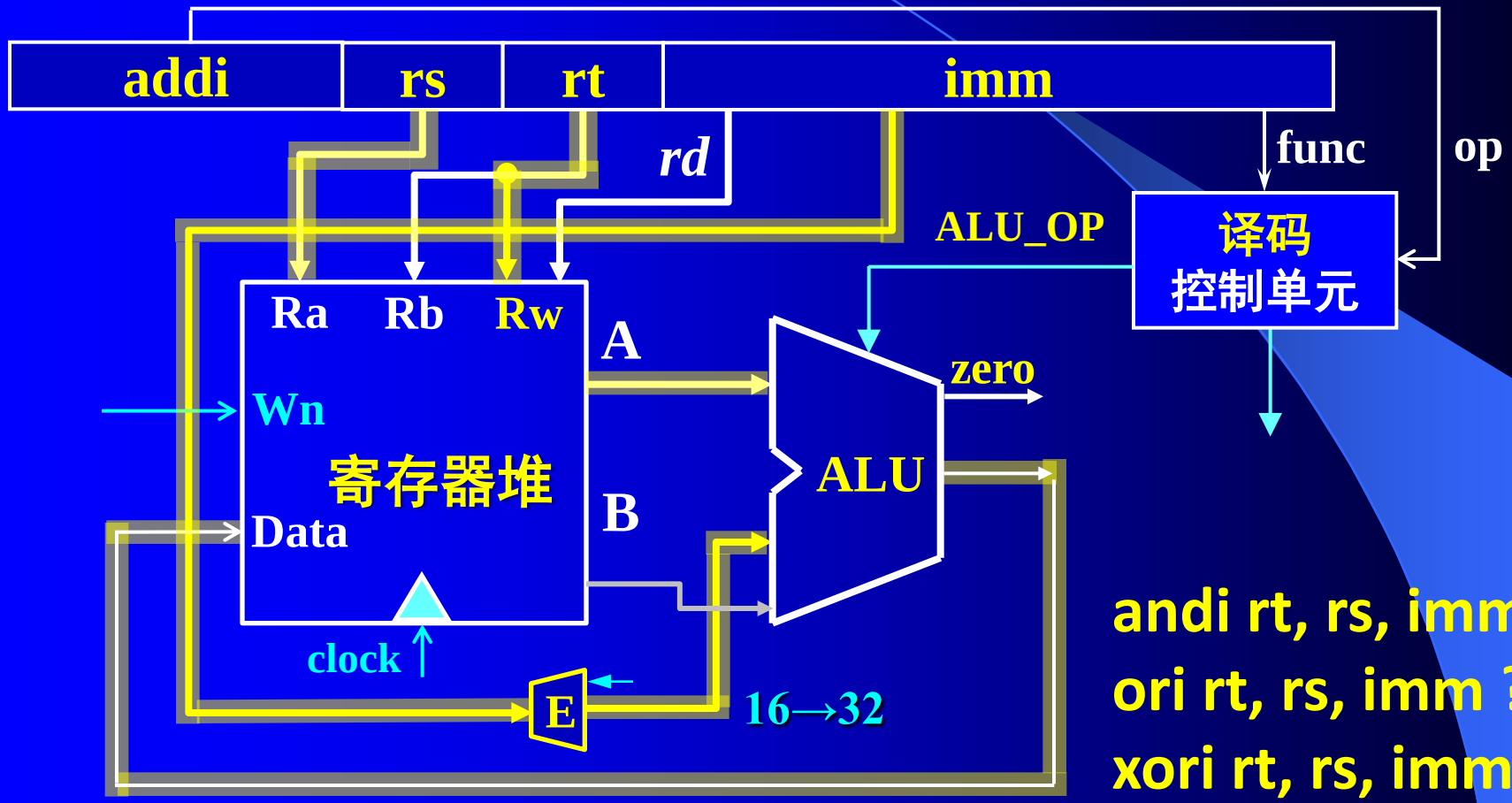
srl rd, rt, sa ?

[例] add rd, rs, rt　　*# R[rd]← R[rs]+R[rt]*



（3）在R型上扩展I型运算指令

［例1］addi rt, rs, imm　　*# R[rt]← R[rs]+E(imm)*



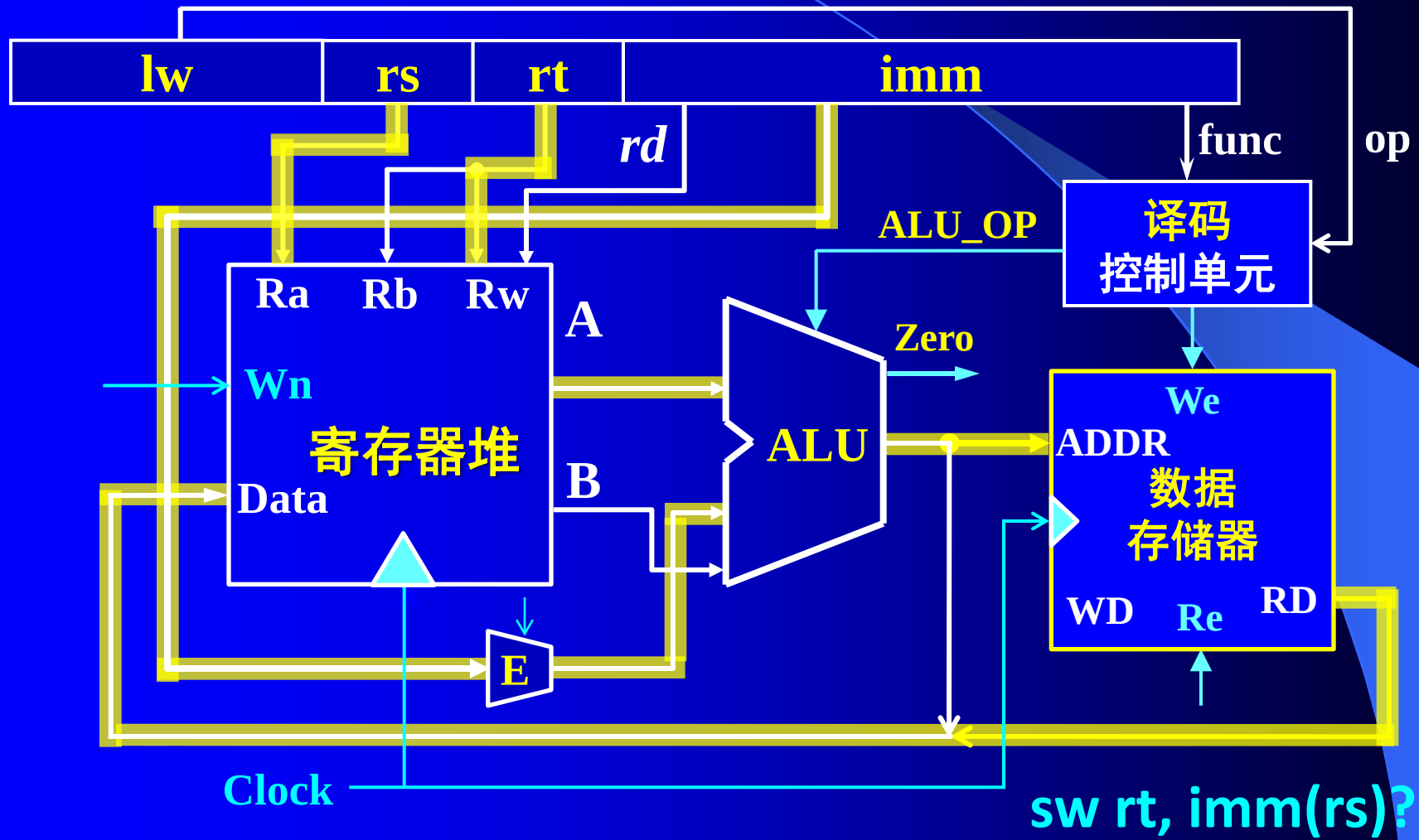
andi rt, rs, imm ?

ori rt, rs, imm ?

xori rt, rs, imm ?

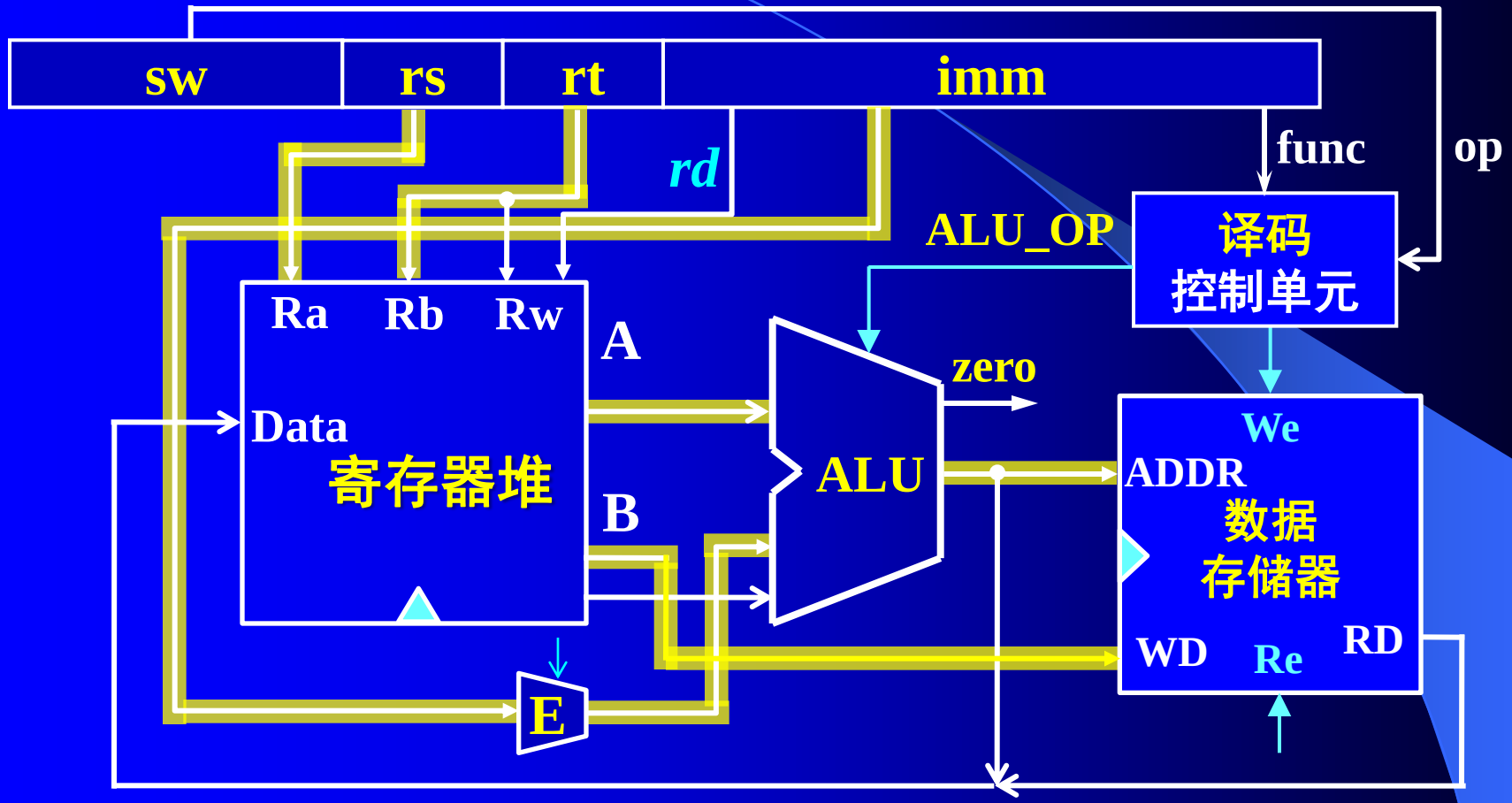
（4）继续扩展I型访存指令

[例2] lw rt, imm(rs)　　*# R[rt]← Mem[R[rs]+E(imm)]*



sw rt, imm(rs)?

[例3] sw rt, imm(rs) # M[R[rs]+E(imm)] ← R[rt]

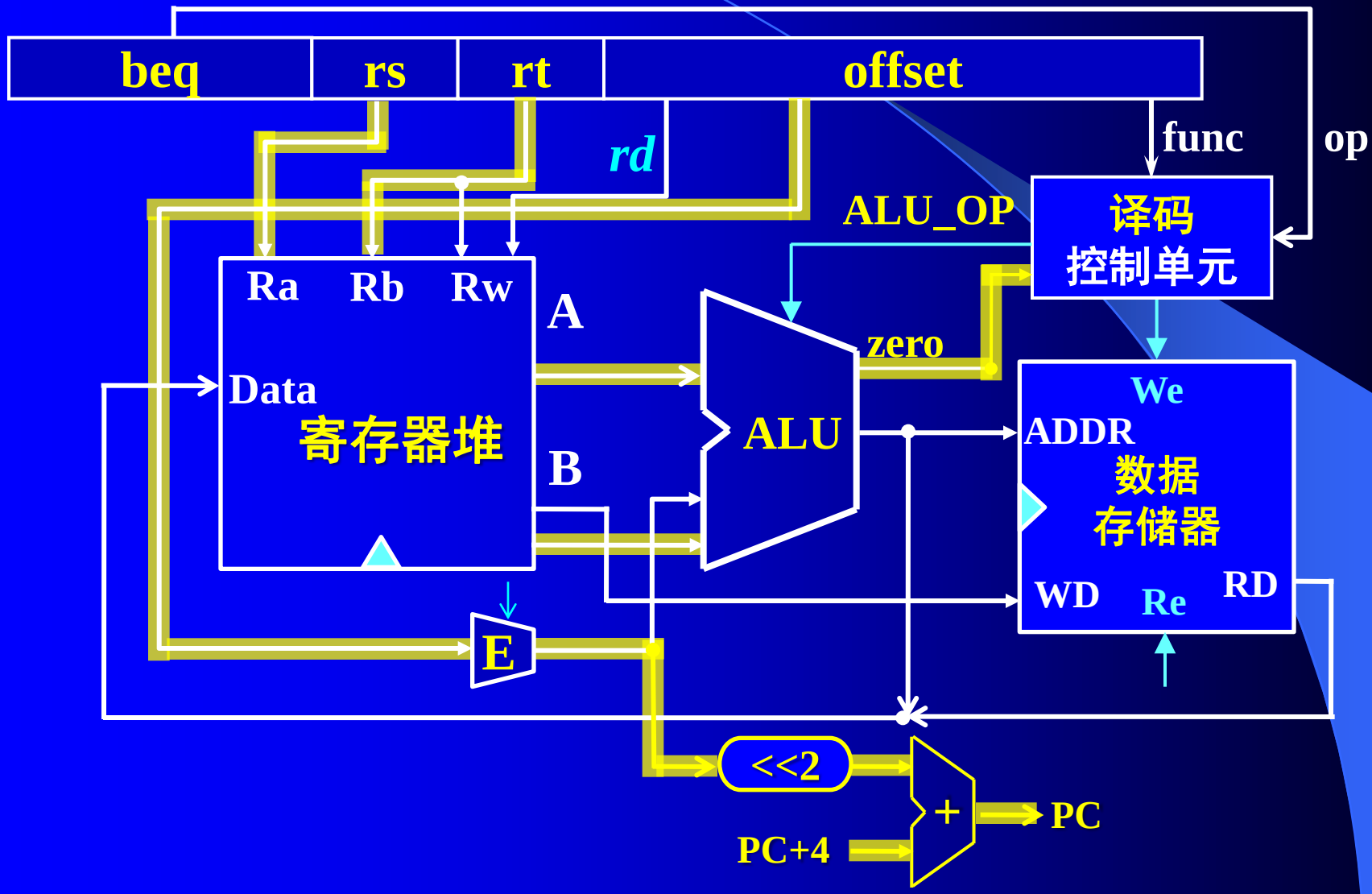


扩展了lw指令后的数据通路

（5）扩展I型分支指令

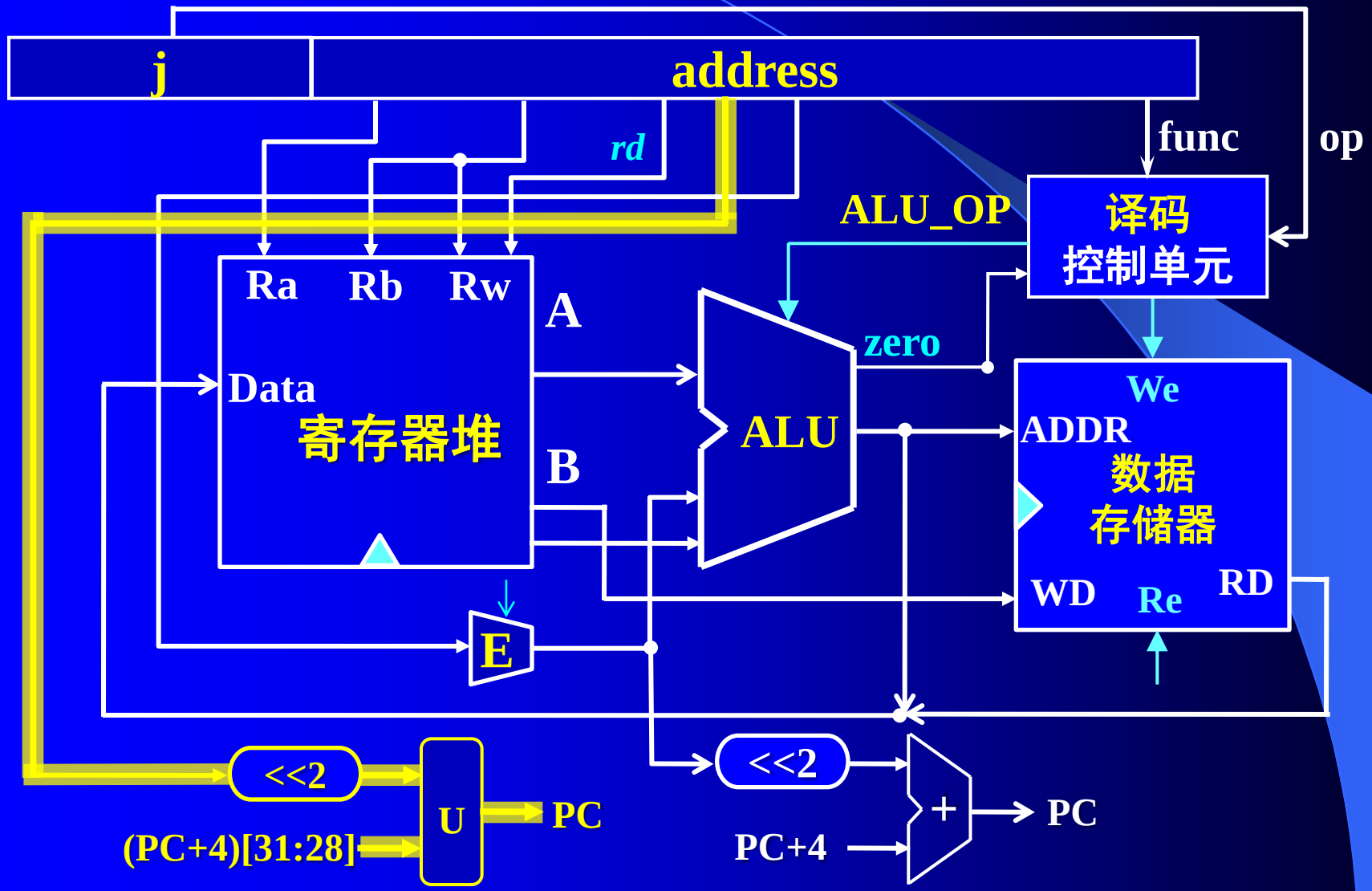
[例4] beq, rs, rt, offset

 # if R[rs]==R[rt]
 # then PC←PC+4+E(offset)<<2
 # else PC←PC+4

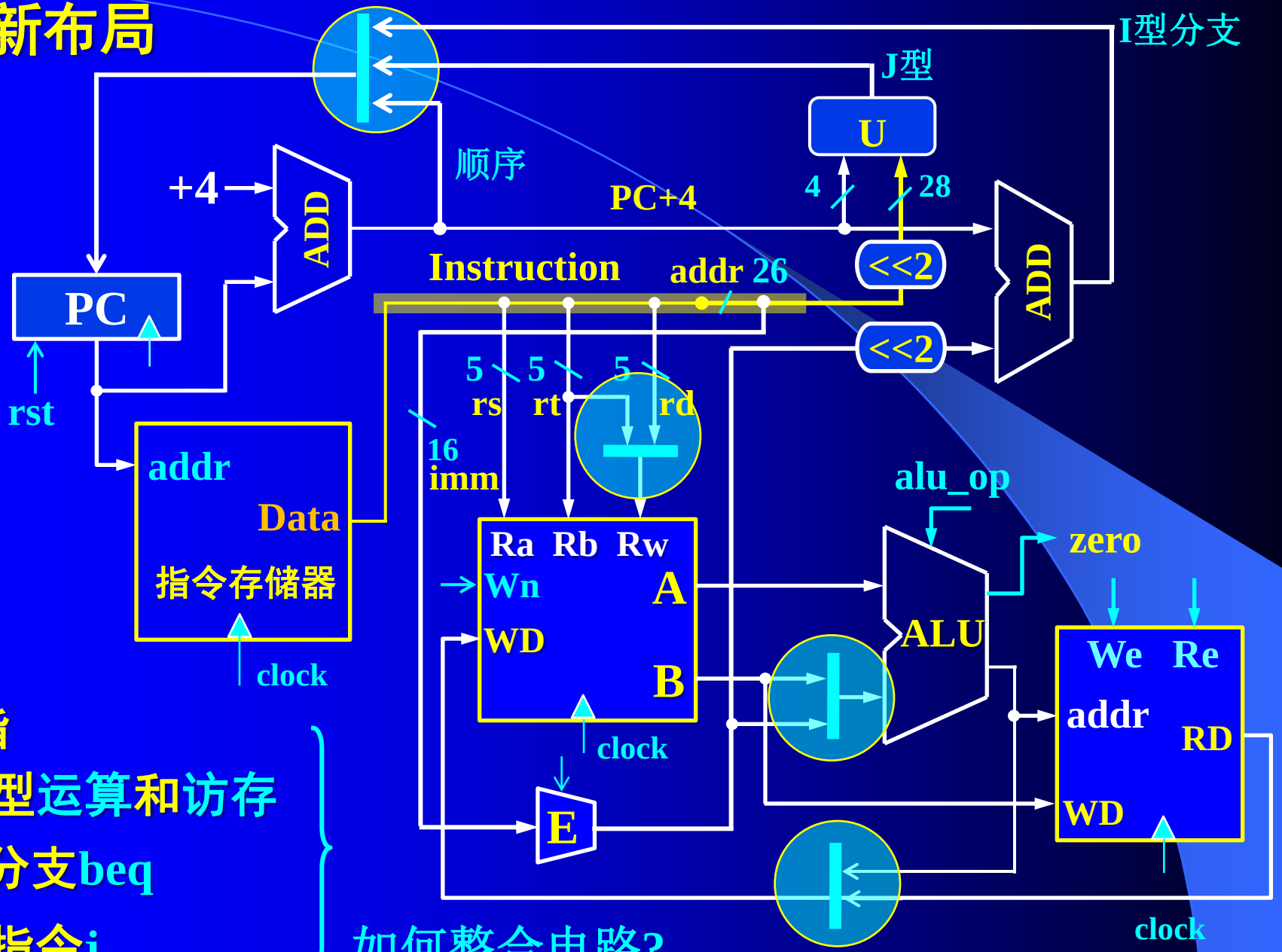


（6）最后扩展J型j指令

[例]j, address　*# PC ←(PC+4)[31:28]∪ (address<<2)*



#重新布局



取指

R/I型运算和访存

I型分支beq

J型指令j

如何整合电路?

3、数据通路综合

不能为各型指令单独设置通路，因为：

→数据通路太繁杂;
→硬件冗余度太大;

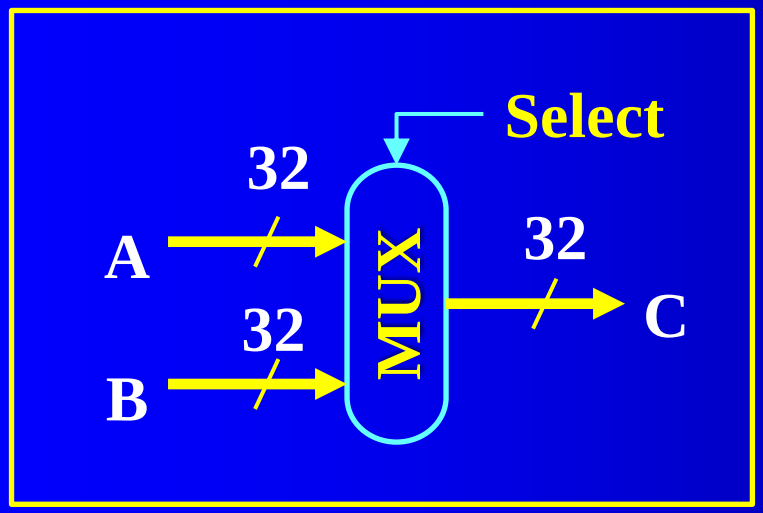
必须将各型指令的数据通路综合、化简

→提高处理器的集成度;
→方便半导体工艺实现;
→提高处理器可靠性。

目标：把各种指令的数据路径合并

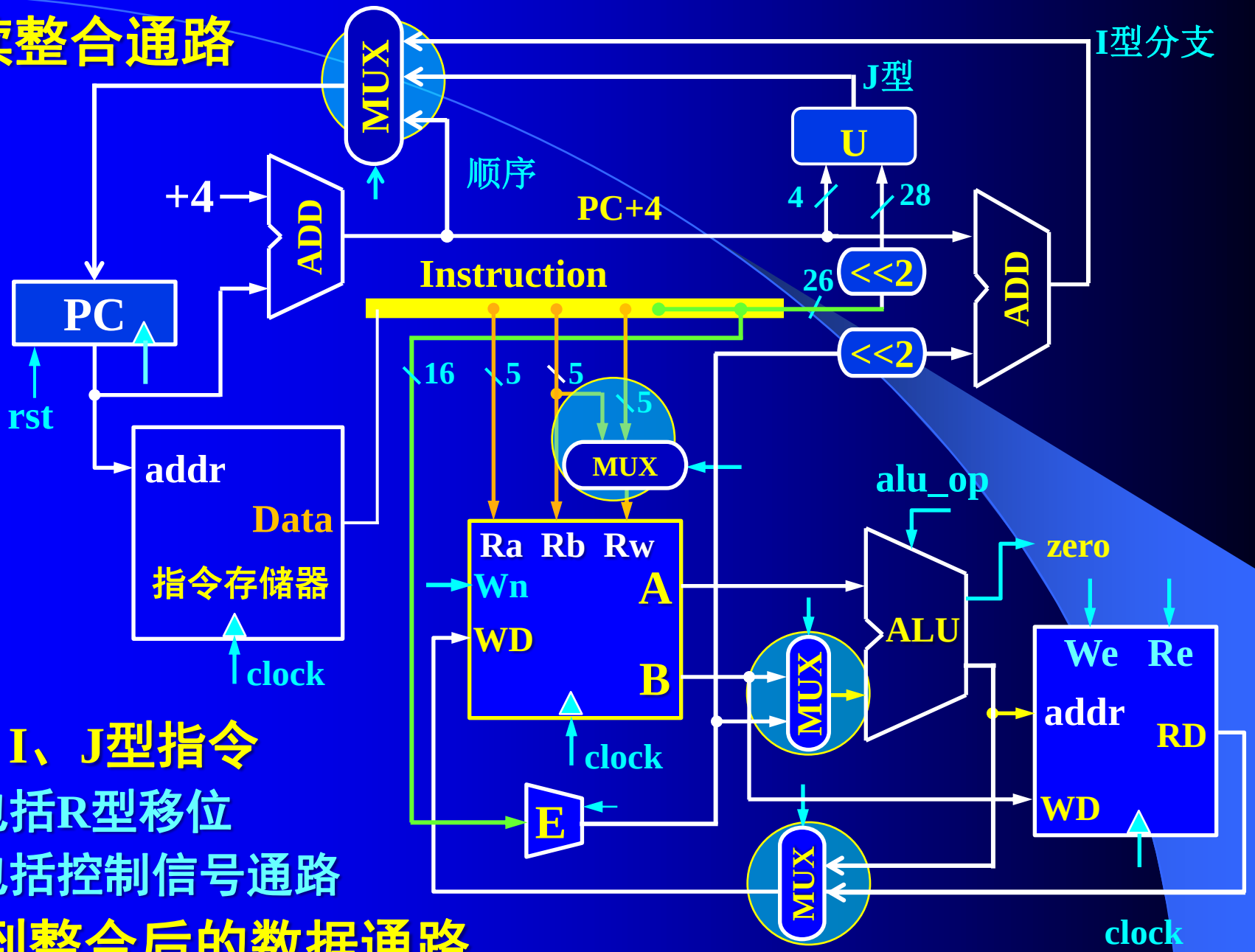
- 取指令（各指令共享）
- R型指令
- I型指令（Load/Store、分支）
- J型指令（转移）指令

■ 基本思路：用多路选择器，整合冗余通路。



利用不同的选通信号，控制选通（切换）不同的数据通路。

继续整合通路



R、I、J型指令

未包括R型移位

未包括控制信号通路

得到整合后的数据通路