

数据通路简介

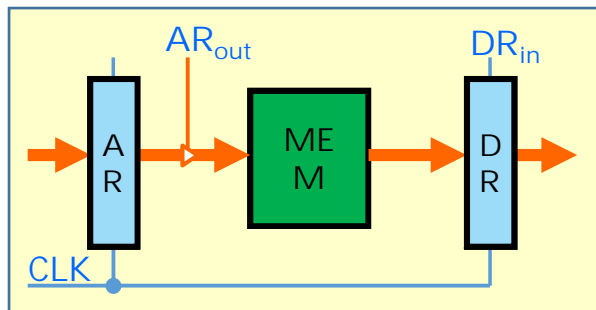
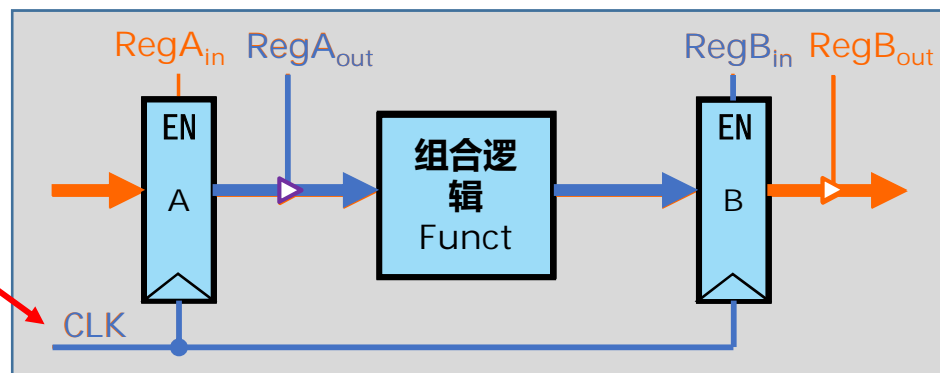
(参考谭志虎老师课件)

数据通路 DataPath

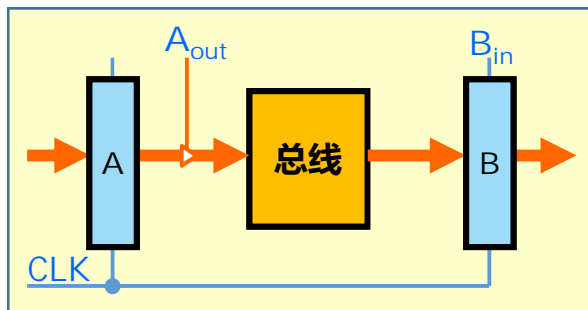
- 数据通路-----执行部件间传送信息的路径 (数据流)
 - ◆通路的建立由控制信号控制，受时钟驱动 (控制流)
 - ◆不同指令、同一指令在执行的不同阶段的数据通路不同
 - ◆分类：共享通路（总线）、专用通路
 - 指令执行流程、执行效率
 - 微操作控制信号的时序安排

数据通路抽象模型（寄存器传输）

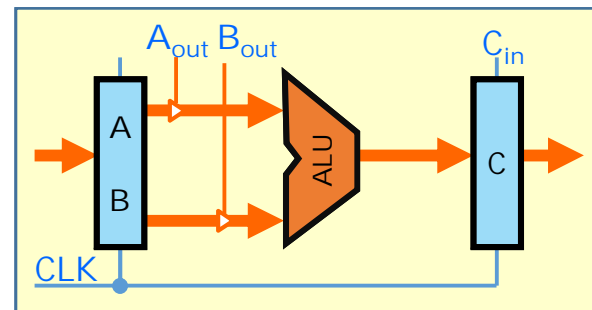
时钟频率?



访存通路

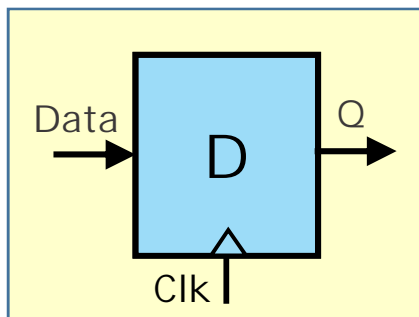


总线传输

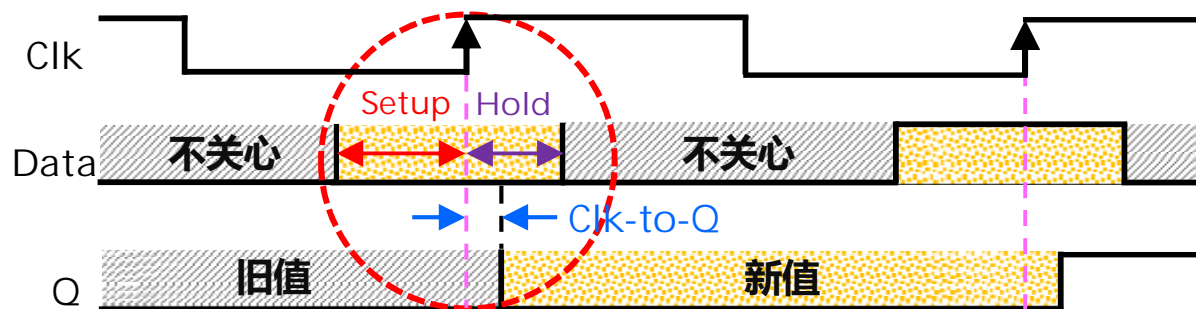


运算通路

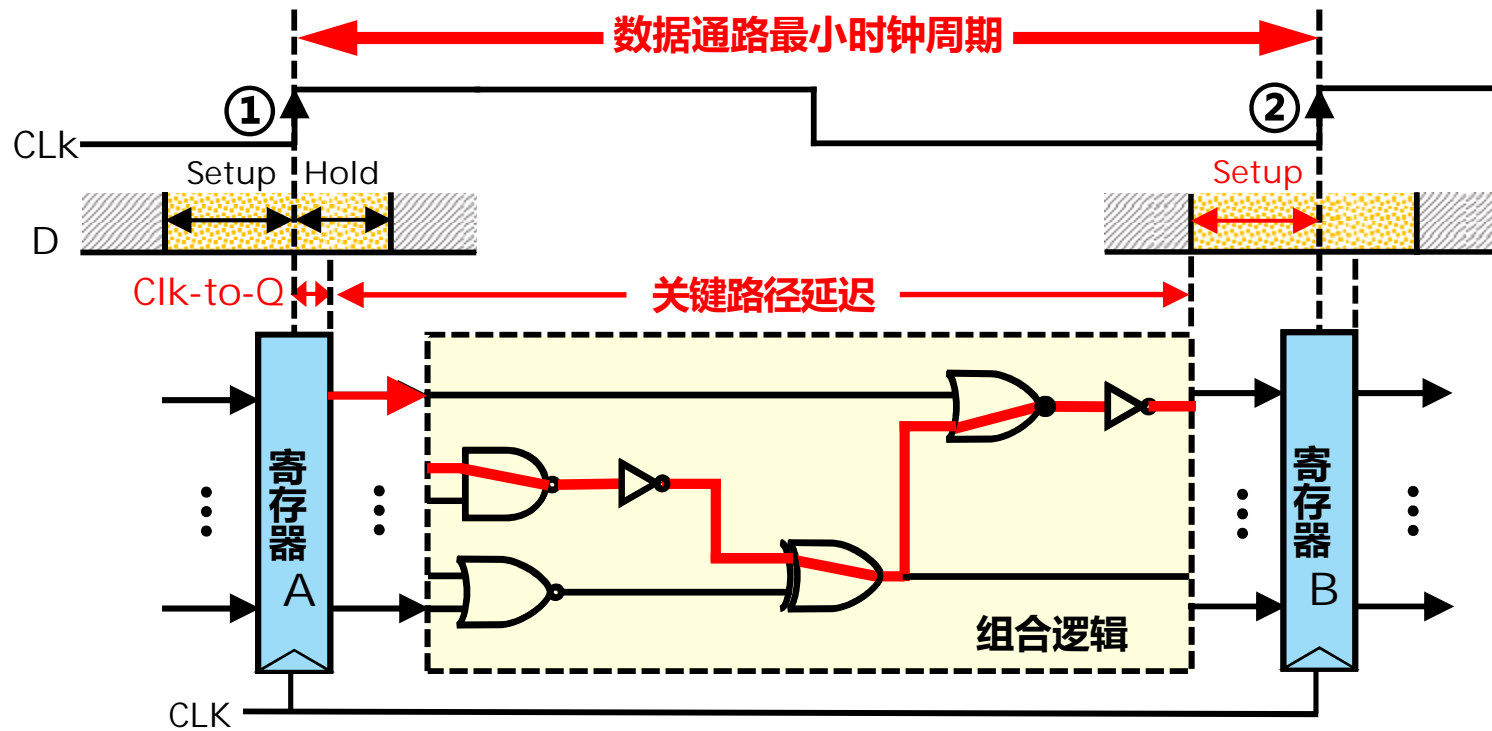
D触发器定时模型



- 时钟触发前输入须稳定一段 **建立时间 (Setup Time)**
- 时钟触发后输入须稳定一段 **保持时间 (Hold Time)**
- 时钟触发到输出稳定的时间 **触发器延迟 Clk_to_Q**



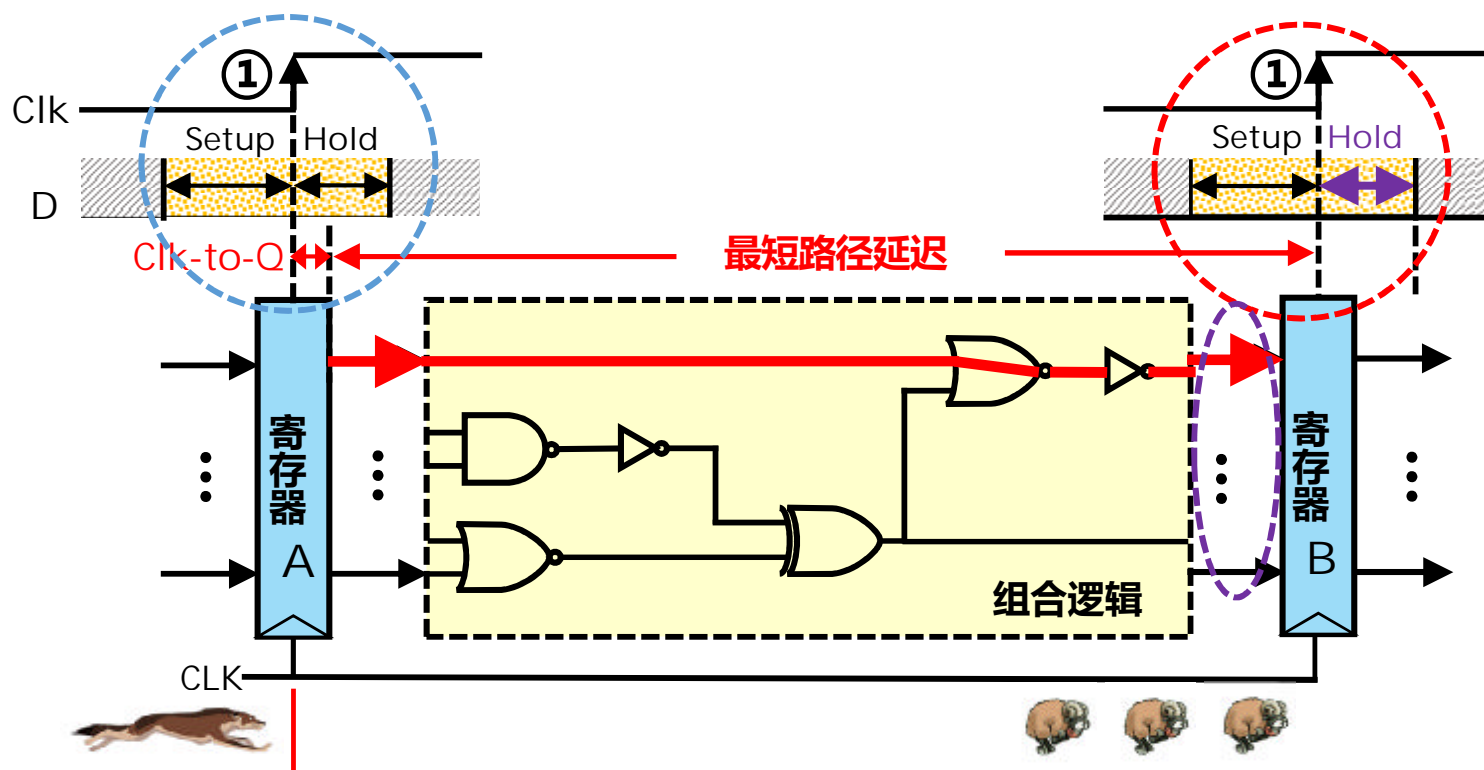
数据通路与时钟周期



■ 时钟周期 > Clk_to_Q + 关键路径时延 + Setup Time

保持时间违例

■ $\text{Clk_to_Q} + \text{最短路径延迟} > \text{Hold_Time}$



数据通路分类

- 共享通路（总线型）

- ◆主要部件都连接在公共总线上，各部件间通过总线进行数据传输

- ◆结构简单，实现容易，但并发性较差，需分时使用总线，效率低

- 专用通路

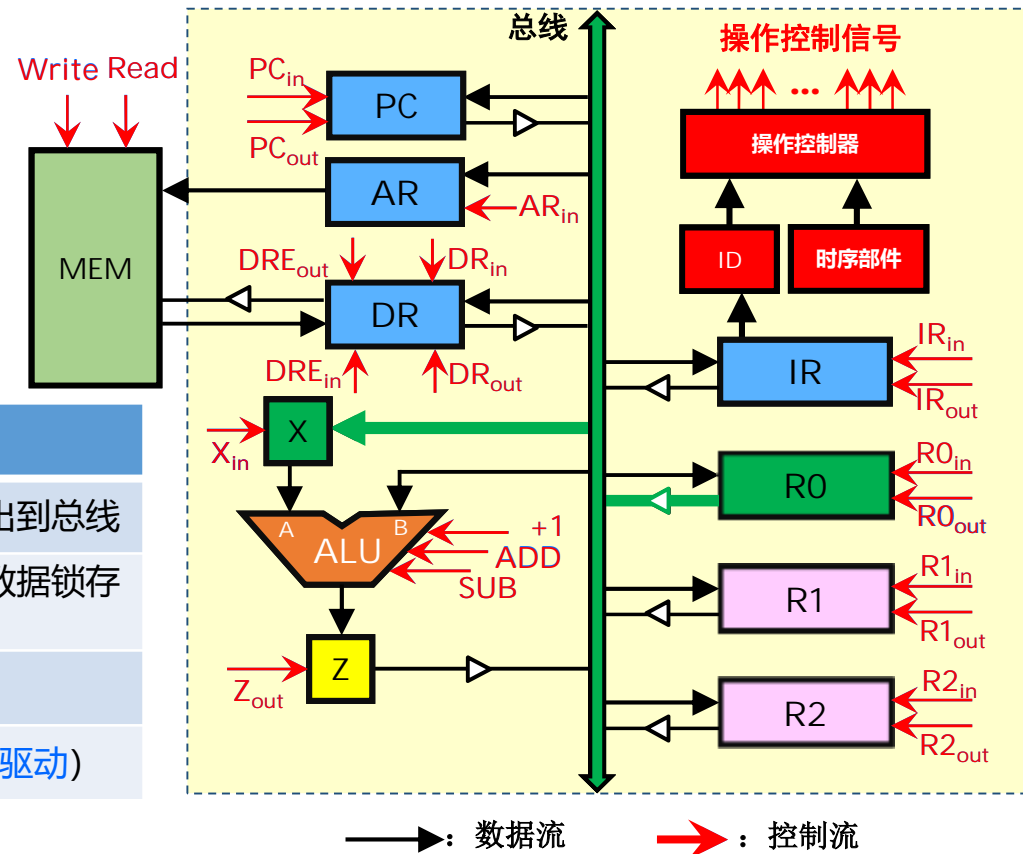
- ◆并发度高，性能佳，设计复杂，成本高

- ◆可以看作多总线结构

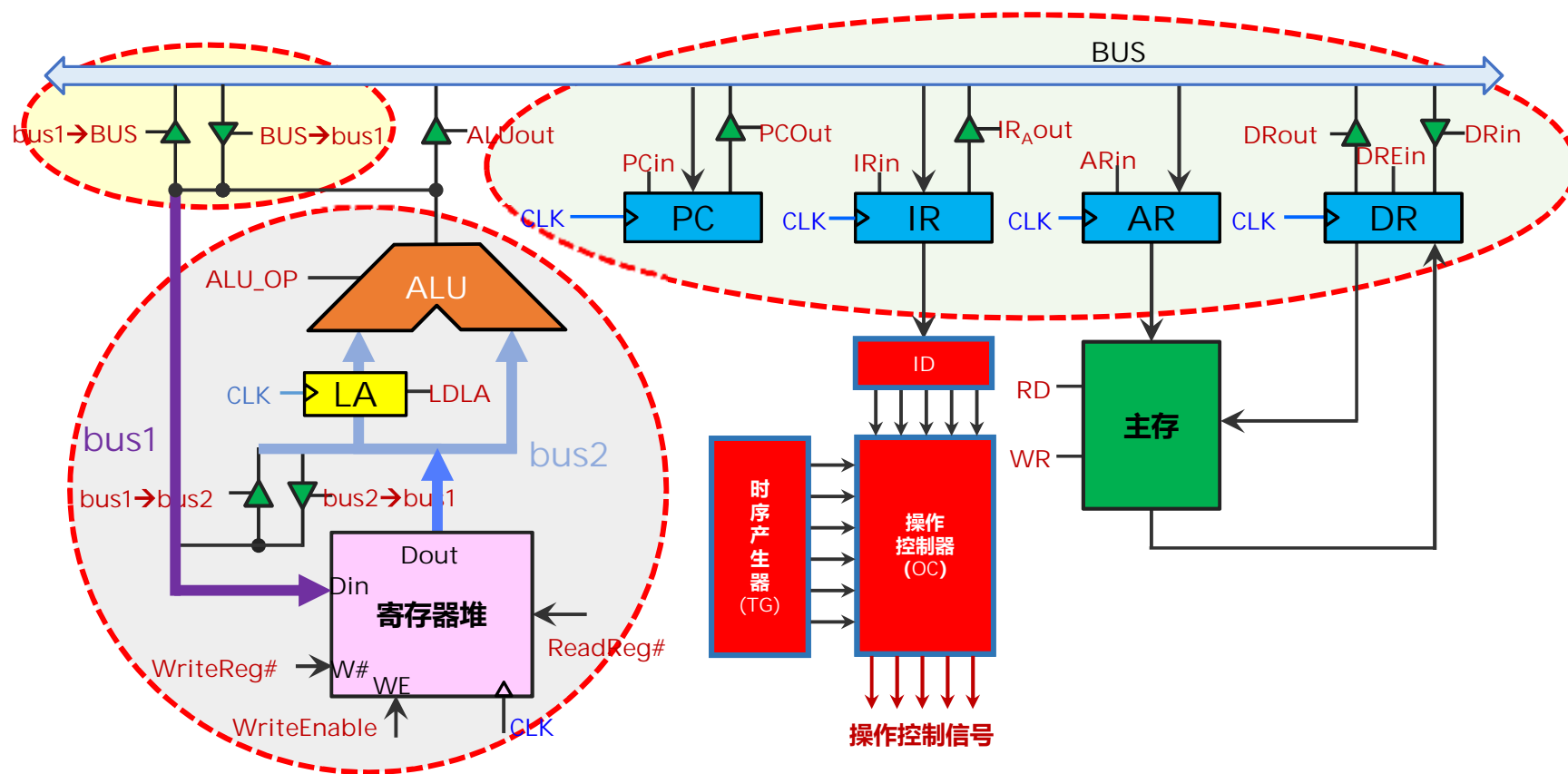
单总线结构CPU实例

- 主要部件都连接在总线上
- 各部件间通过总线进行传输

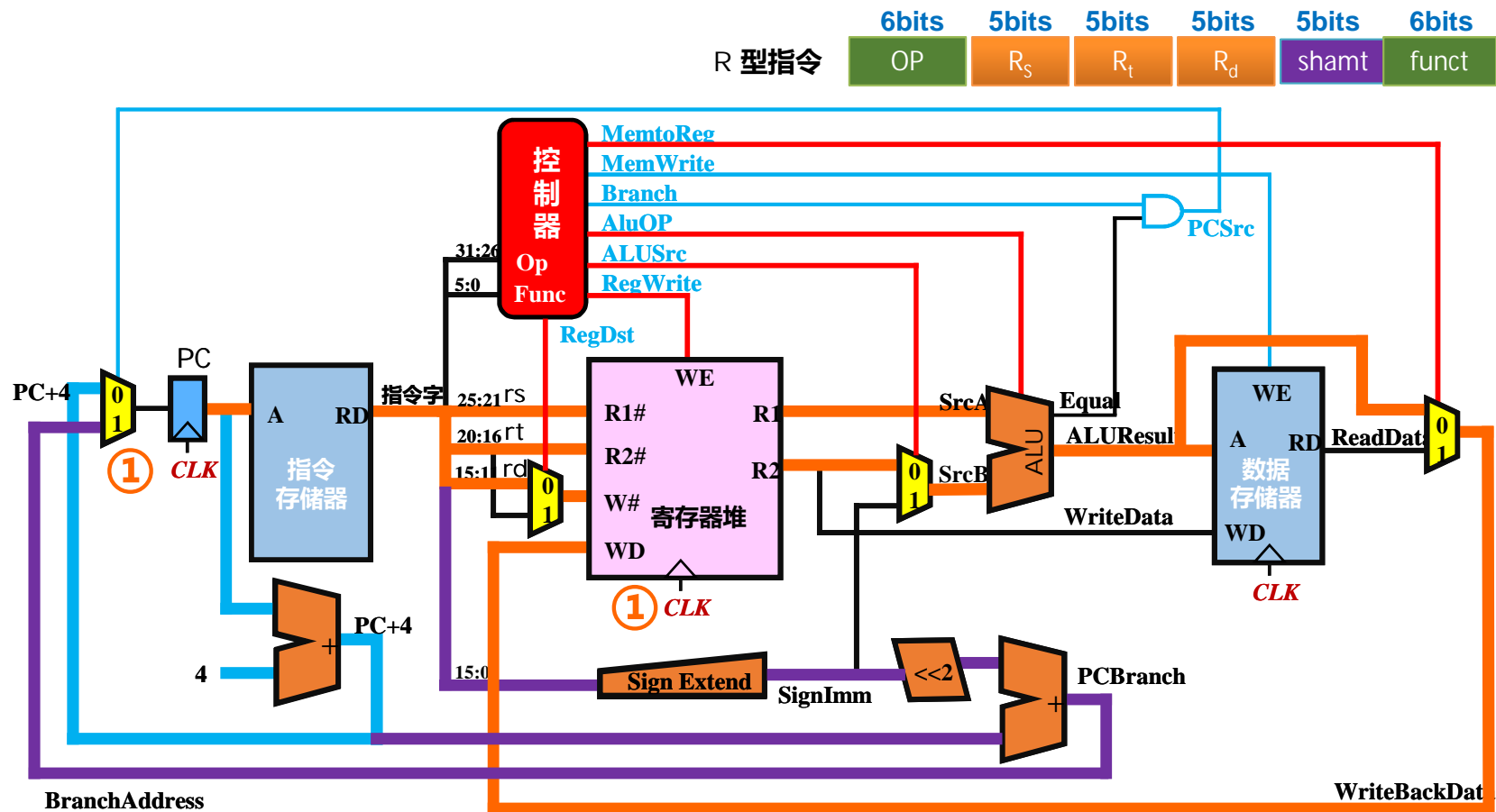
控制信号	作用说明
IR_{out} 、 PC_{out} 、... $R1_{out}$	控制三态门将寄存器值输出到总线
IR_{in} 、 PC_{in} 、... $R1_{in}$	控制寄存器使能端将总线数据锁存 (时钟驱动)
+1、ADD、SUB	运算控制信号
Write、Read	内存读写控制信号 (时钟驱动)



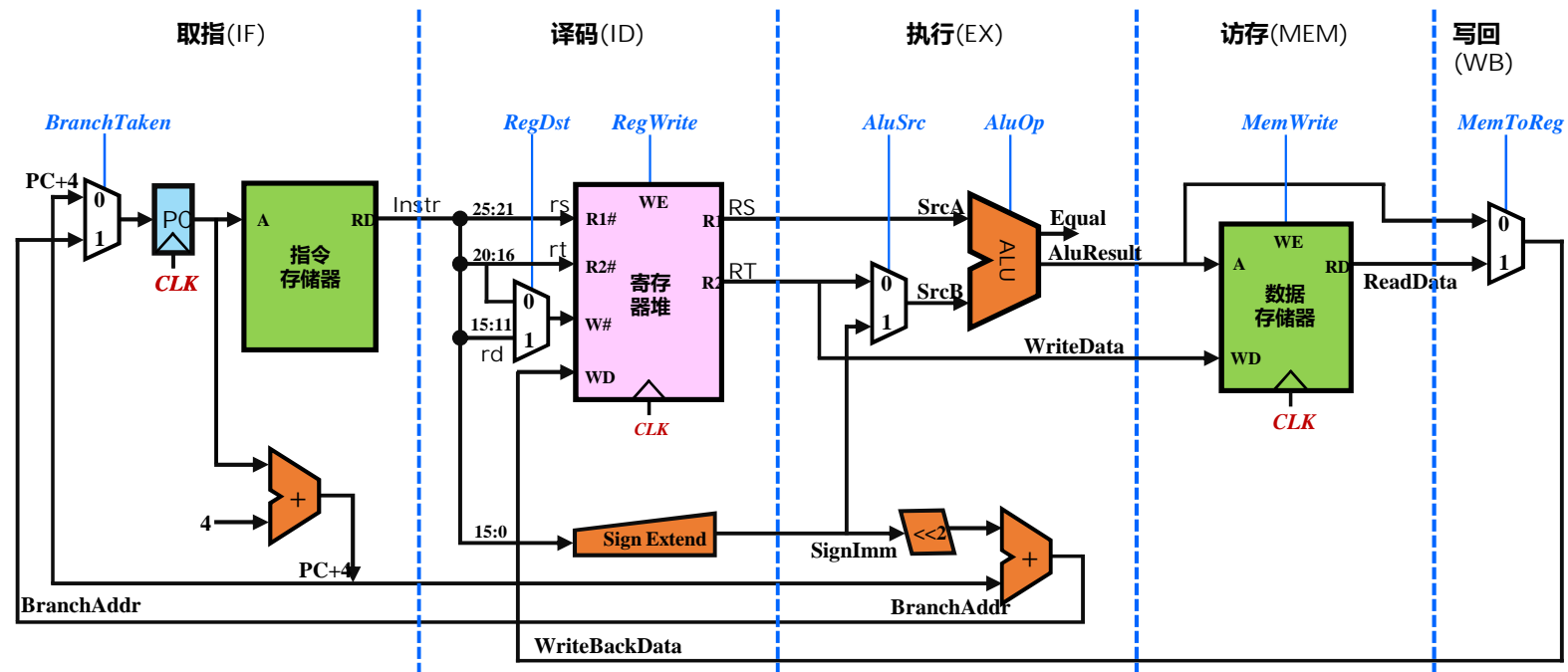
多总线架构数据通路



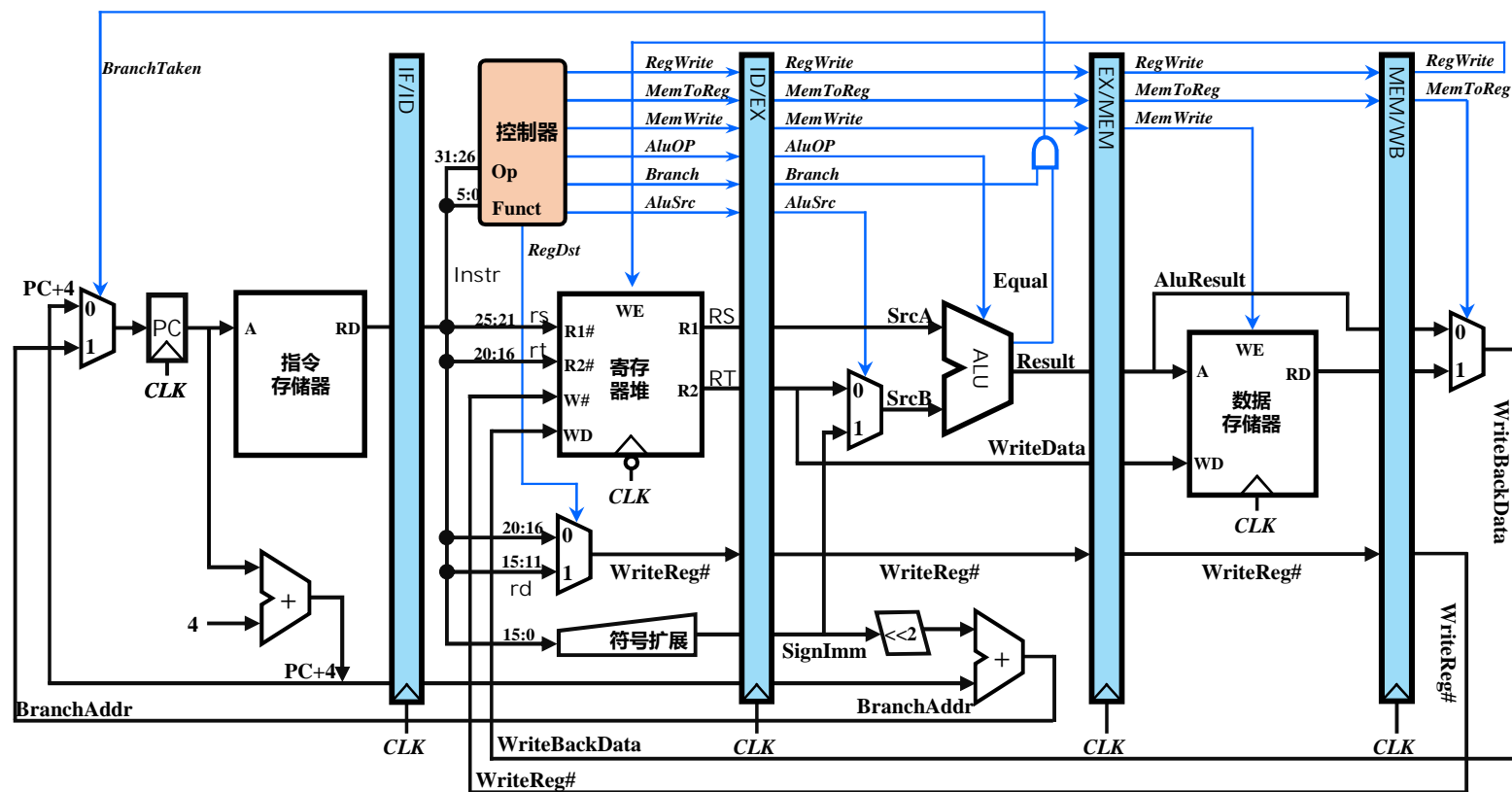
专用通路 单周期MIPS CPU



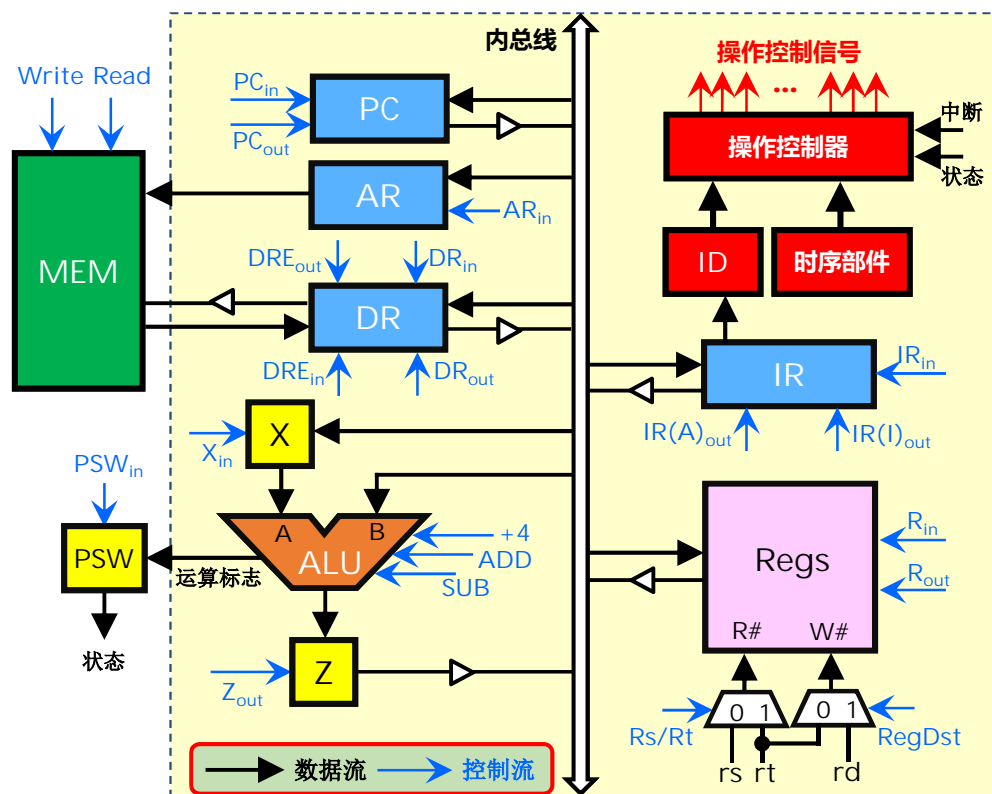
单周期MIPS处理器数据通路



5段流水线控制信号与传递



单总线结构CPU MIPS 指令周期



■ *lw* *rt*, *i mm(rs)*

■ *sw* *rt*, *i mm(rs)*

■ *beq* *rs*, *rt*, *i mm*

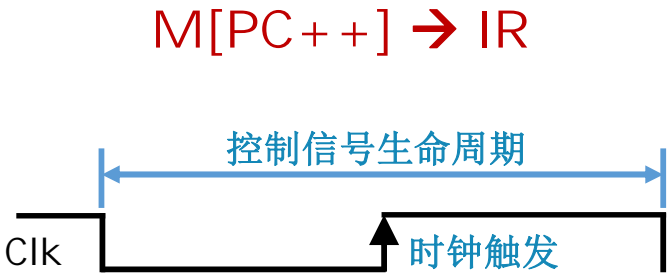
■ *add* *rd*, *rs*, *rt*

■ *addi* *rt*, *rs*, *i mm*

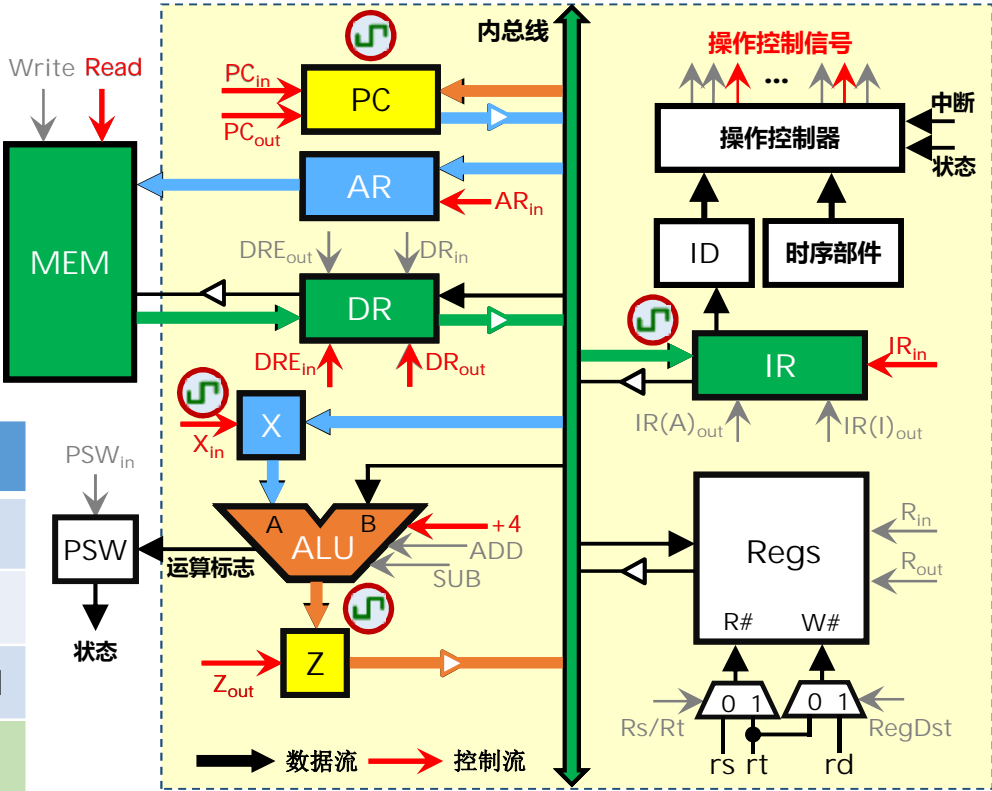
单总线结构MIPS CPU 典型指令

#	指令	指令功能 (RTL描述)
1	<code>lw rt,imm(rs)</code>	$R[rt] \leftarrow M[R[rs] + \text{SignExt}(imm)]$
2	<code>sw rt,imm(rs)</code>	$M[R[rs] + \text{SignExt}(imm)] \leftarrow R[rt]$
3	<code>beq rs,rt,imm</code>	$\text{if}(R[rs] == R[rt]) \text{ PC} \leftarrow \text{PC} + 4 + \text{SignExt}(imm) \ll 2$
4	<code>addi rt,rs,imm</code>	$R[rt] \leftarrow R[rs] + \text{SignExt}(imm)$
5	<code>add rd,rs,rt</code>	$R[rd] \leftarrow R[rs] + R[rt]$

取指令数据通路



节拍	数据通路 (数据流)	控制信号 (控制流)
T1	PC \rightarrow AR, PC \rightarrow X	PC _{out} , AR _{in} , X _{in}
T2	X+4 \rightarrow Z	+4
T3	Z \rightarrow PC, M[AR] \rightarrow DR	Z _{out} , PC _{in} , DRE _{in} , Read
T4	DR \rightarrow IR	DR _{out} , IR _{in}

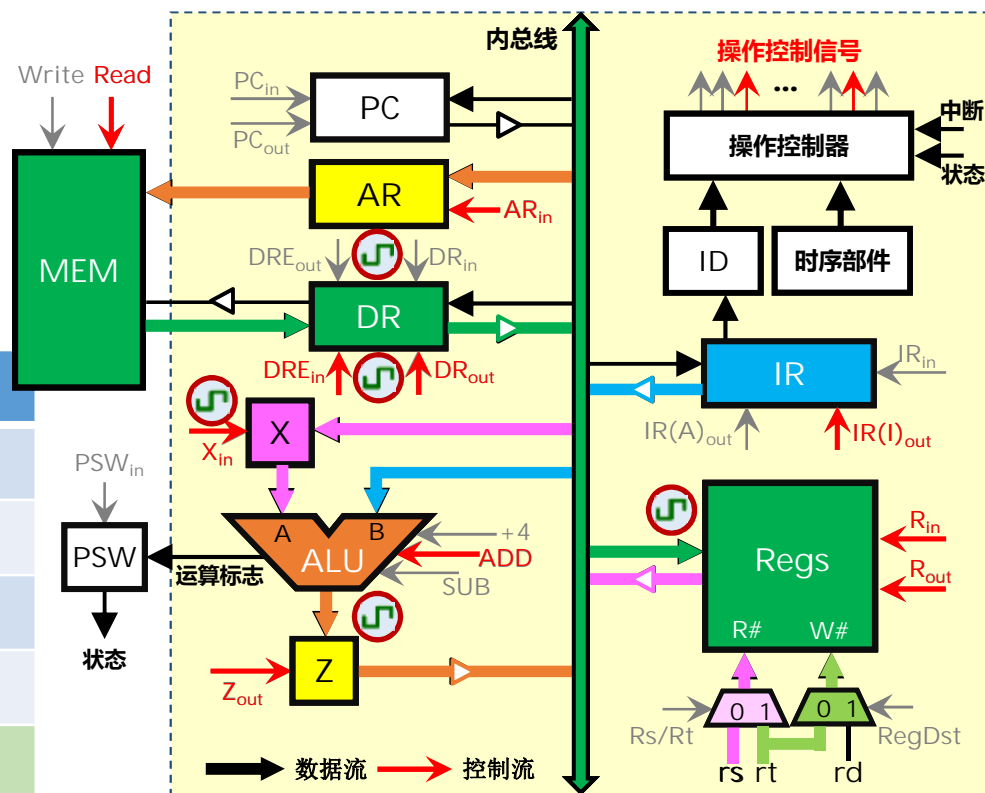


LW 指令执行数据通路 $lw\ rt, imm(rs)$



$$M[R[rs] + imm] \rightarrow R[rt]$$

节拍	数据通路 (数据流)	控制信号(控制流)
T5	$R[rs] \rightarrow X$	R_{out}, X_{in}
T6	$X + imm \rightarrow Z$	$IR(I)_{out}, ADD$
T7	$Z \rightarrow AR$	Z_{out}, AR_{in}
T8	$M[AR] \rightarrow DR$	$DRE_{in}, Read$
T9	$DR \rightarrow R[rt]$	DR_{out}, R_{in}

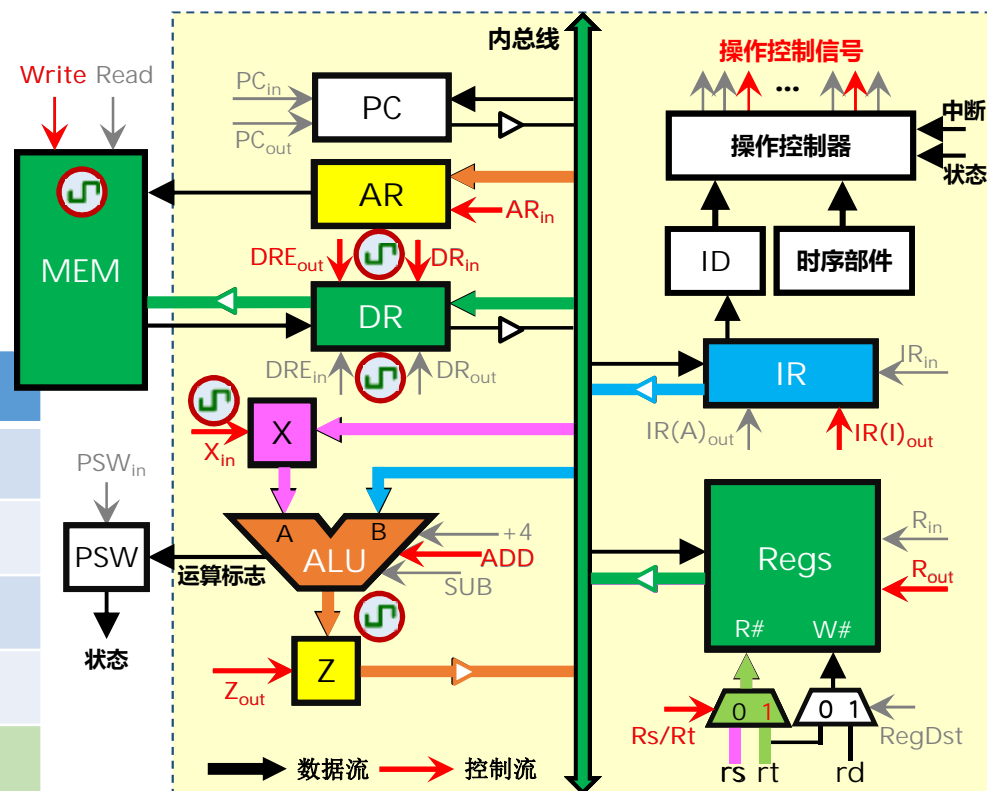


SW 指令执行数据通路 $sw\ rt, imm(rs)$



$$R[rt] \rightarrow M[R[rs] + imm]$$

节拍	数据通路 (数据流)	控制信号 (控制流)
T5	$R[rs] \rightarrow X$	R_{out}, X_{in}
T6	$IR[I] + X \rightarrow Z$	$IR(I)_{out}, ADD$
T7	$Z \rightarrow AR$	Z_{out}, AR_{in}
T8	$R[rt] \rightarrow DR$	$R_{out}, Rs/Rt, DR_{in}$
T9	$DR \rightarrow M[AR]$	$DRE_{out}, Write$



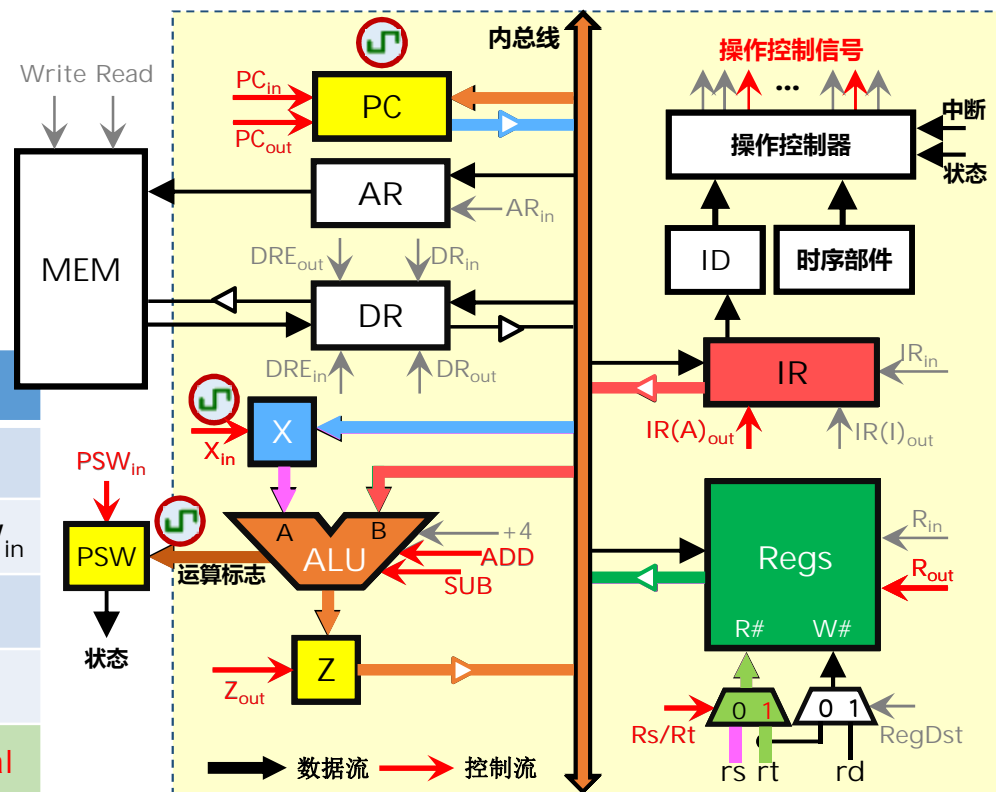
BEQ 指令数据通路 *beq rs, rt, imm*



if ($R[rs] = R[rt]$)

$PC + 4 + imm \ll 2 \rightarrow PC$

节拍	数据通路 (数据流)	控制信号(控制流)
T5	$R[rs] \rightarrow X$	R_{out}, X_{in}
T6	$X - R[rt] \rightarrow PSW$	$R_{out}, Rs/Rt, SUB, PSW_{in}$
T7	$PC \rightarrow X$	PC_{out}, X_{in}
T8	$IR(A) + X \rightarrow Z$	$IR(A)_{out}, ADD$
T9	If ($PSW.equal$) $Z \rightarrow PC$	$Z_{out}, PC_{in} = PSW.equal$

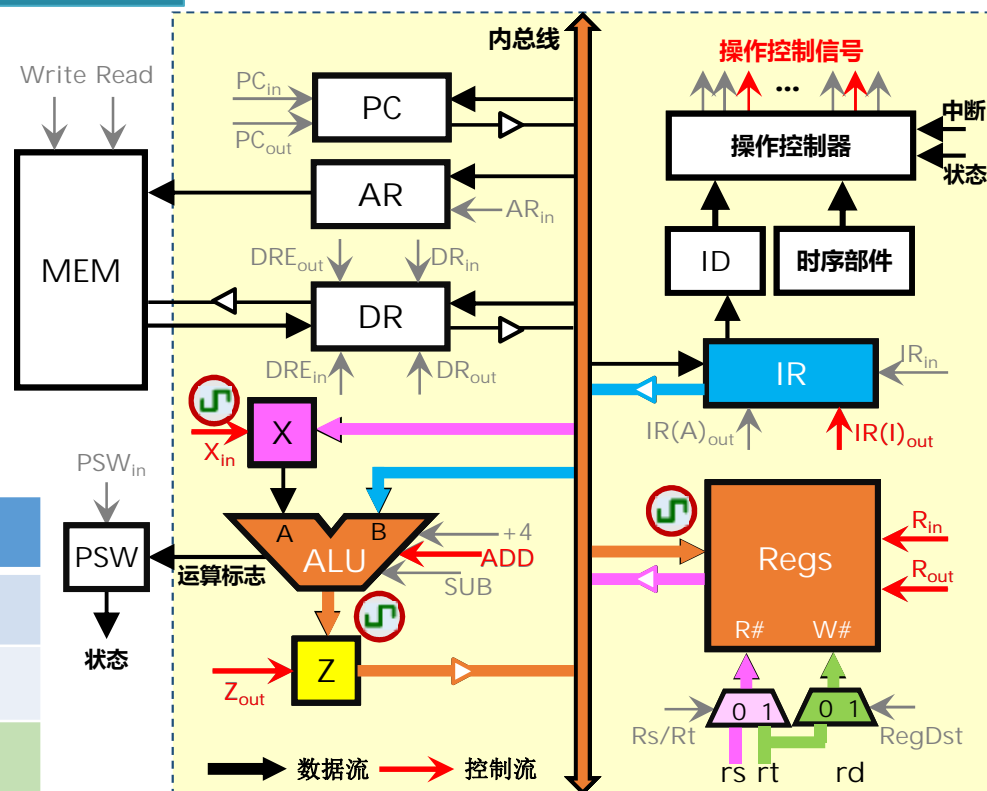


I型运算类指令执行数据通路 *addi rt, rs, imm*



$$R[rs] + imm \rightarrow R[rt]$$

节拍	数据通路 (数据流)	控制信号(控制流)
T5	$R[rs] \rightarrow X$	R_{out}, X_{in}
T6	$X + imm \rightarrow Z$	$IR(I)_{out}, ADD$
T7	$Z \rightarrow R[rt]$	Z_{out}, R_{in}

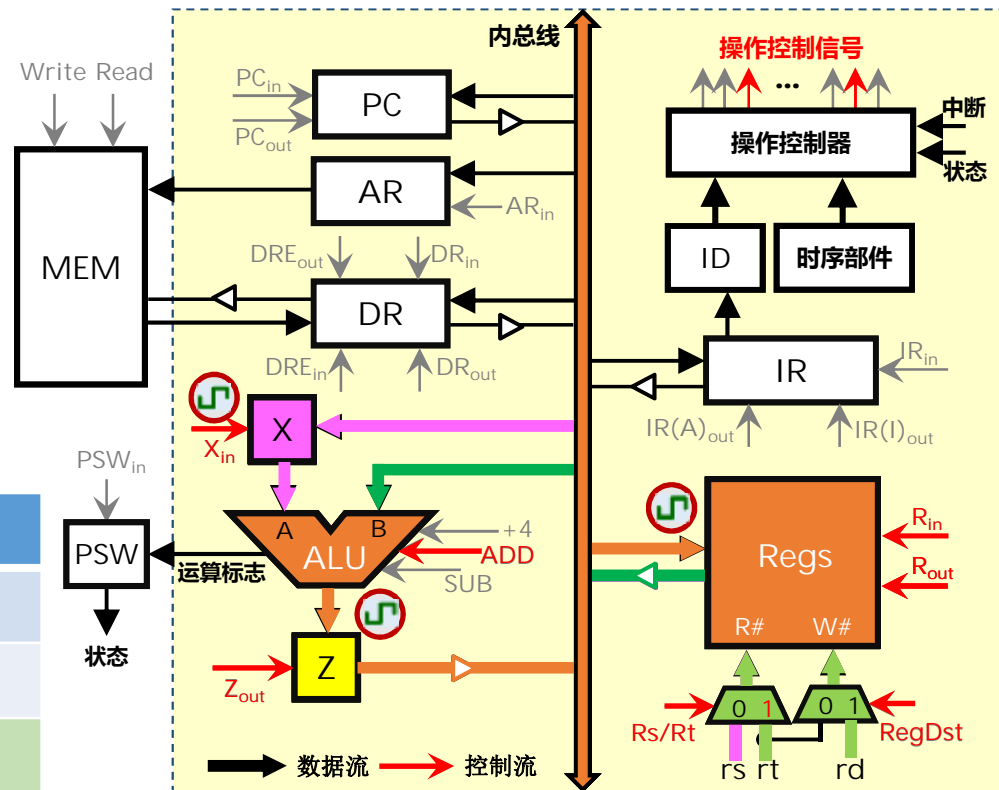


R型运算类指令执行数据通路 *add rd, rs, rt*

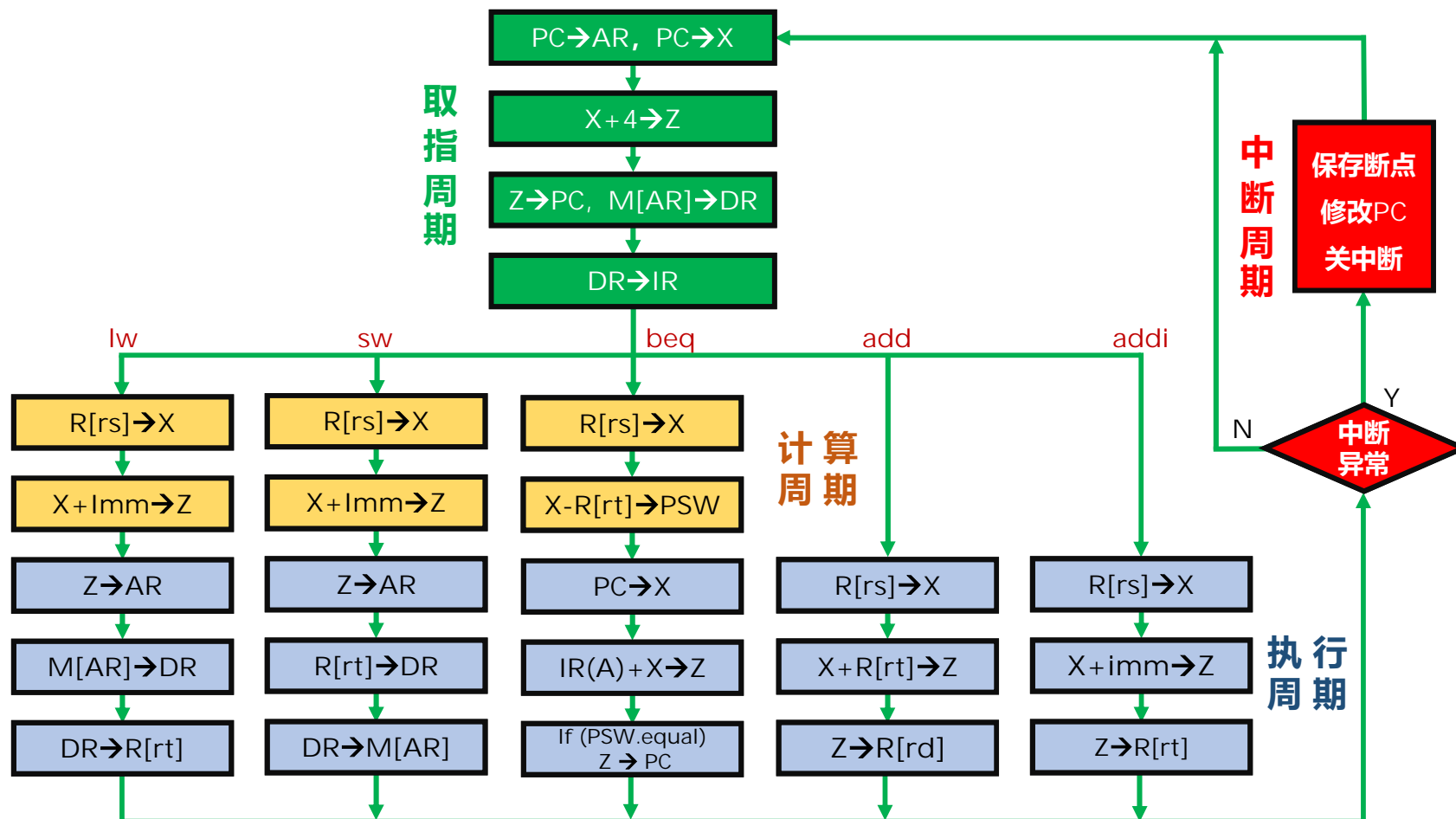


$$R[rs] + R[rt] \rightarrow R[rd]$$

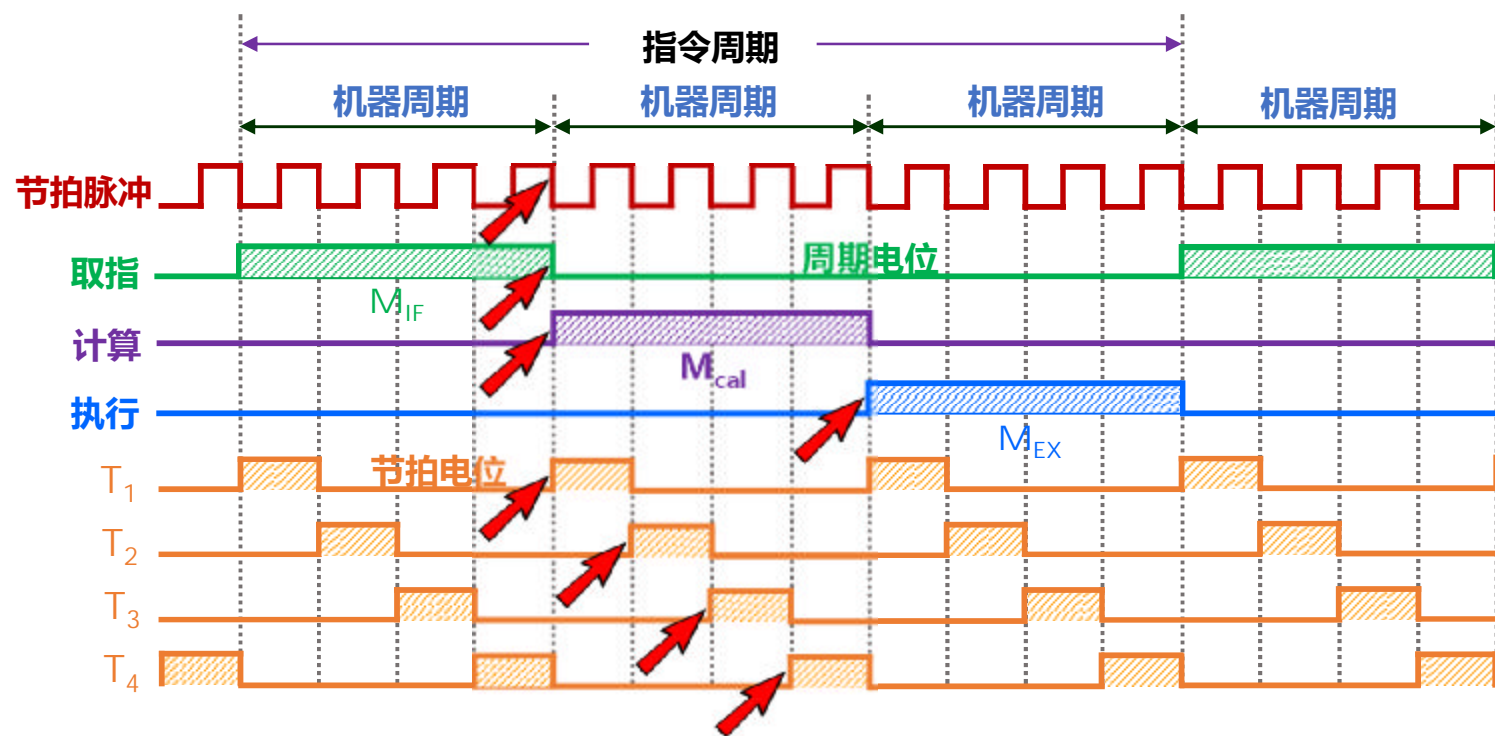
节拍	数据通路 (数据流)	控制信号 (控制流)
T5	$R[rs] \rightarrow X$	R_{out}, X_{in}
T6	$X + R[rt] \rightarrow Z$	$Rs/Rt, R_{out}, ADD$
T7	$Z \rightarrow R[rd]$	$Z_{out}, R_{in}, RegDst$



指令周期方框图 (数据流)

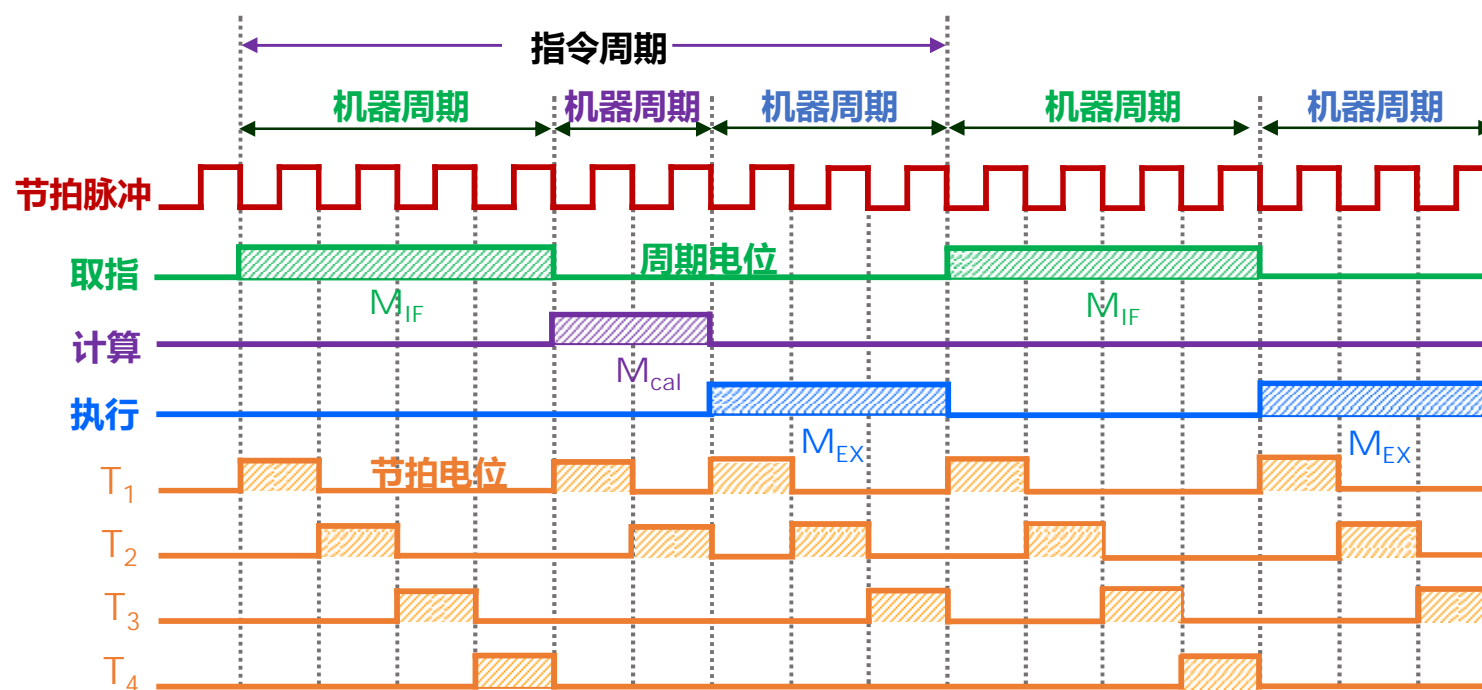


定长指令周期的三级时序发生器



构建时序发生器? 输入: 节拍脉冲 输出: M_{IF} , M_{cal} , M_{EX} , $T_1 \sim T_4$

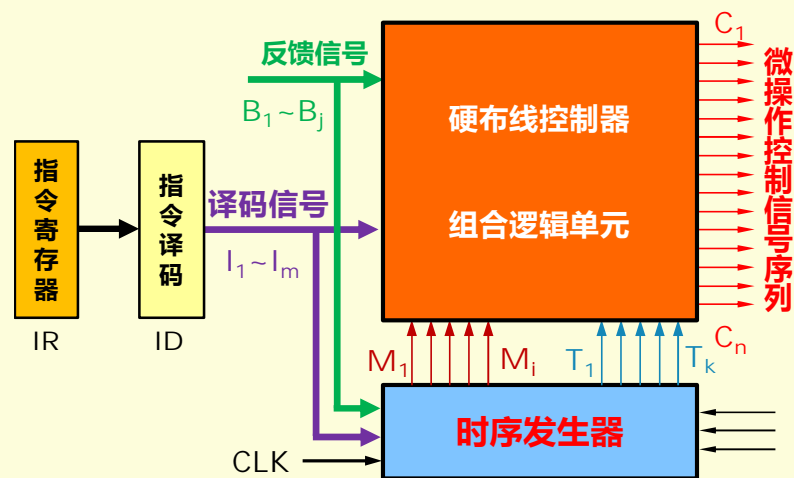
变长指令周期三级时序时序发生器



机器周期数可变、节拍数可变，无周期浪费，更加灵活

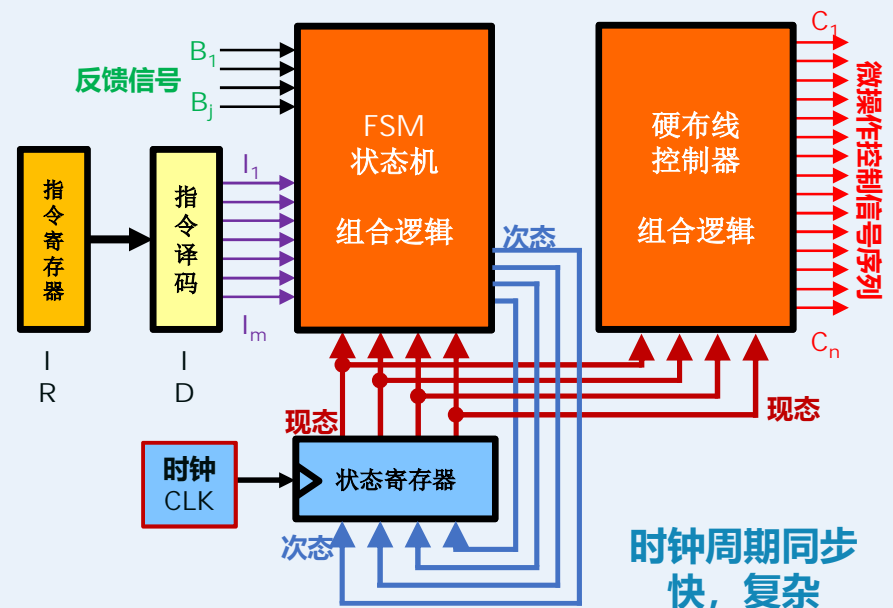
传统三级时序与现代时序对比

传统三级时序



机器周期同步，慢，简单

现代时序



时钟周期同步
快，复杂