计算机组成原理

中央处理器（1）

王浩宇,教授

haoyuwang@hust.edu.cn

https://howiepku.github.io/

本节目录

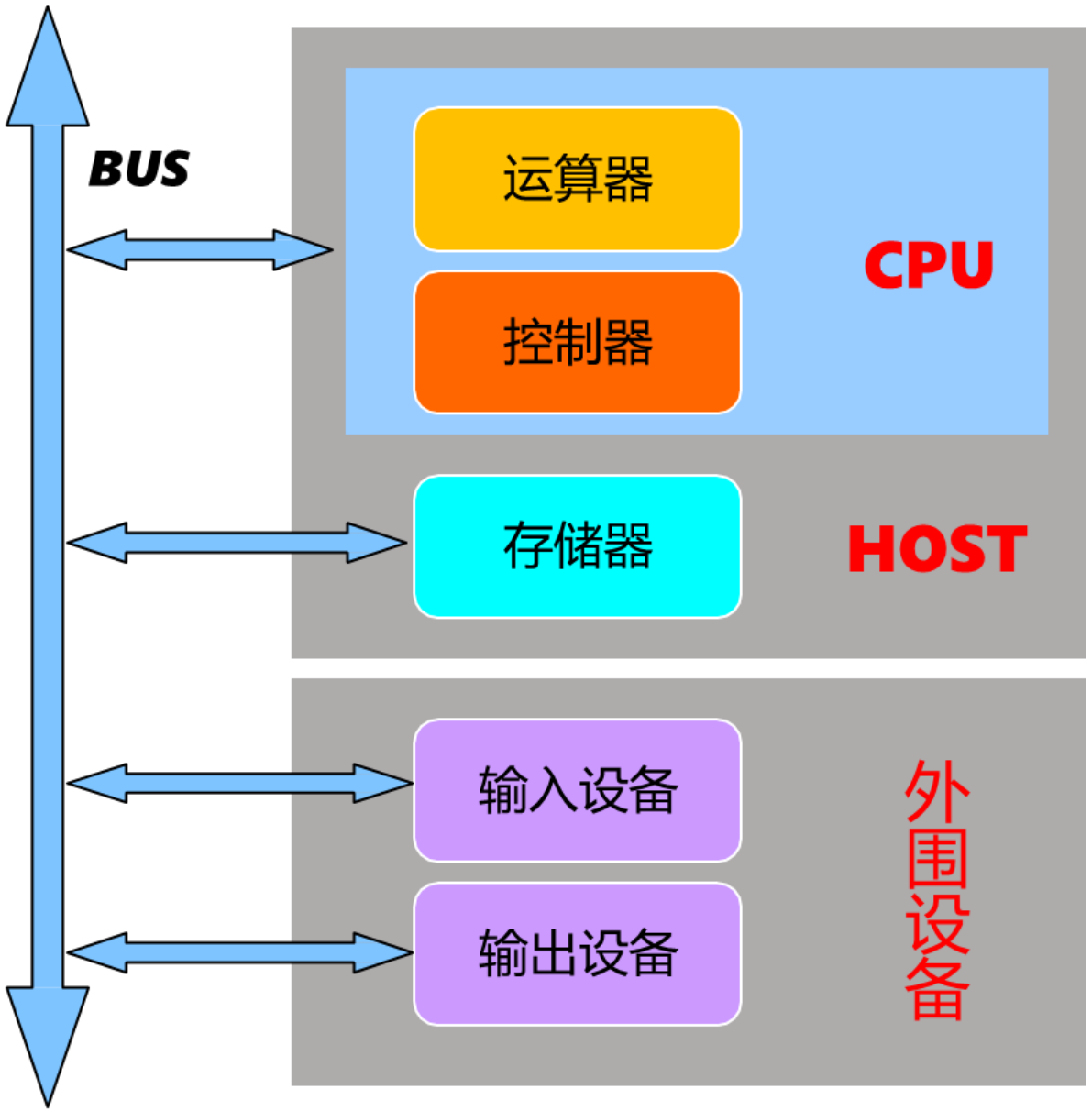
- CPU的组成与功能
- 数据通路与总线结构

CPU的组成与功能

- CPU概述
- CPU中的主要寄存器
- 控制器
- 运算器

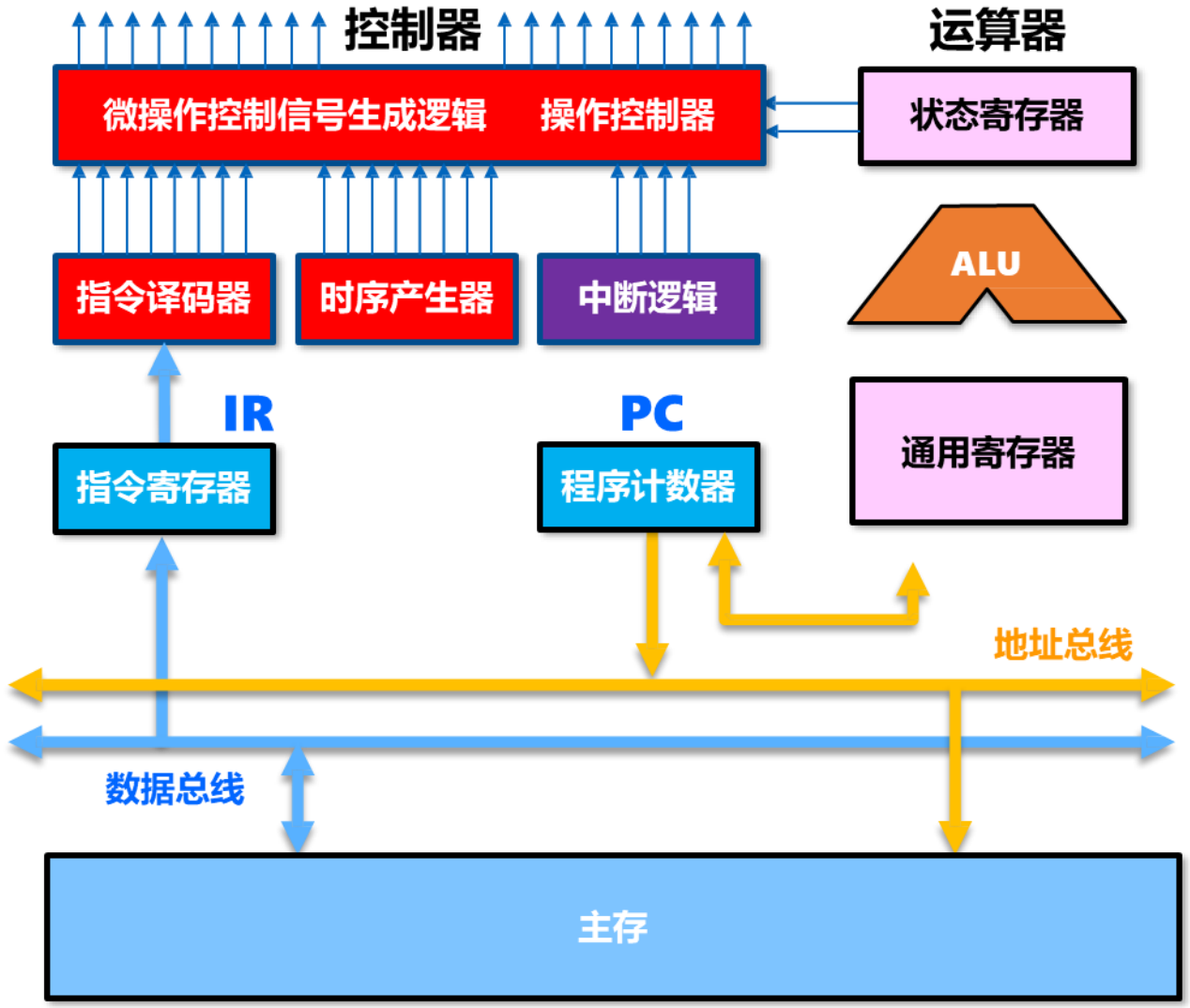
CPU概述

- 冯诺依曼计算机
  - 运算器、控制器
  - 取指令，执行指令
- 差异性
  - ISA，数据通路
  - 控制器实现方式
  - 性能、成本



CPU基本组成

- 运算器　数据加工
- 控制器　程序执行/指令执行
  - ◆ 取指令
    - ◆ Mem[PC++] → IR
  - ◆ 执行指令
    - ◆ 指令字→控制信号序列
    - ◆ 信号序列→数据通路



CPU主要功能

- 取指令并执行指令的部件------CPU

  - 数据加工：算术/逻辑运算　　运算器
  - 控制器
    - 程序控制：程序中指令执行顺序控制
    - 操作控制：将机器指令翻译成执行部件所需的操作控制信号
    - 时序控制：控制操作信号的产生时间、持续时间
    - 异常控制：异常处理，外设交互

CPU执行的过程

- CPU执行指令的过程
  - 取指令
  - PC+1
  - 指令译码
  - 进行操作数地址运算
  - 取操作数
  - 进行算术 / 逻辑运算
  - 存结果
  - 判断和检测“异常”事件
  - 若有异常，则自动切换到异常处理程序
  - 检测是否有“中断”请求，有则转中断处理
- CPU的实现与计算机性能的关系
  - 计算机性能(程序执行快慢)由三个关键因素决定：
    - 指令数目、CPI、时钟周期
      - 指令数目由编译器和指令集决定
      - 时钟周期和CPI由CPU的实现来决定

CPU的基本组成

- 运算器
  - 即执行部件，数据加工处理部件，执行所有的算术运算和逻辑运算
  - ALU
  - 通用寄存器
  - 暂存器
  - 状态条件寄存器PSW
  - 运算器接受控制器的命令而进行动作
    - 运算器所进行的全部操作都是由控制器发出的控制信号来指挥的

CPU的组成

- **控制器：指挥计算机各部件按指令要求进行操作的部件**
  - 控制取出、解释和执行指令
    - 从指令Cache/内存中取指，并提供下一条指令地址
    - 对指令进行译码/测试，产生相应的操作控制信号
    - 产生执行部件的运行所需要的控制信号
  - 指挥并控制CPU，内存和I/O设备之间的数据传送
    - 存储器与控制器之间的信息流动——指令流；
    - 存储器与运算器之间的信息流动——数据流。
  - 中断控制——对异常情况和外部请求的处理

CPU的组成与功能

- CPU概述
- CPU中的主要寄存器
- 控制器
- 运算器

CPU中的主要寄存器

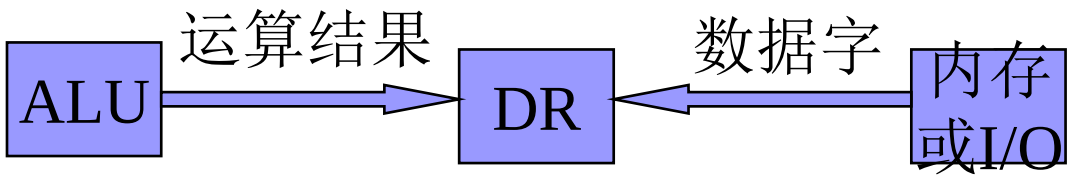
- PC (Program Counter)------程序计数器　X86：EIP　MIPS：PC
- IR (Instruction Register)------指令寄存器　[可选]
- AR (Address Register)------地址寄存器　MAR　[可选]
- DR (Data Register)------数据缓冲寄存器　MDR　[可选]
- AC (Accumulate Count)------累加寄存器　[可选]
- PSW (Program Status Word)------程序状态字　[可选]
  - X86 EFLAGS　MIPS：无

程序计数器（PC）

- 为了保证程序能够连续执行下去
- 程序计数器用于确定下一条指令的地址
- 在程序开始执行前，将它的起始地址送入PC
- 当执行指令时，CPU将自动修改PC的内容，以便使其保持的总是将要执行的下一条指令的地址
  - 顺序执行，PC+1
  - 转移指令例如JMP，后继指令的地址必须从指令寄存器中的地址字段获得

数据缓冲寄存器（DR）

- 数据缓冲寄存器用来暂时存放ALU的运算结果，或由数据存储器读出的一个数据字，或来自外部接口的一个数据字
- 缓冲寄存器的作用是
  - 作为ALU运算结果和通用寄存器之间信息传送中时间上的缓冲
  - 补偿CPU和内存、外围设备之间在操作速度上的差别



数据地址寄存器（AR）

- 用来保存当前CPU所要访问的内存单元或I/O设备的地址
- 由于内存和CPU之间存在着速度上的差别，所以必须使用地址寄存器来保存地址信息，直到内存读/写操作完成为止
- 在对主存或I/O端口进行访问时，地址寄存器存放当前访问的地址，数据缓冲器实现数据的缓冲
- CPU通过修改地址寄存器中的值，就可访问不同的存储器单元及不同的I/O端口

指令寄存器（IR）

- 指令寄存器用来保存当前正在执行的一条指令。
- 当执行一条指令时，先把它从指令cache存储器（简称指存）读出，然后再传送至指令寄存器。
- 对操作码进行测试，以便识别所要求的操作。（指令译码器的工作）指令寄存器中操作码字段的输出就是指令译码器的输入。
- 操作码一经译码之后，即可向操作控制器发出具体操作的特定信号。

状态字寄存器（PSW）

- 保存由算术指令和逻辑指令运算或测试结果建立的各种条件代码。
  - 如：运算结果进位状态（C），运算结果溢出标志（V），运算结果为零标志（Z），运算结果为负标志（N）。这些标志通常为1位触发器保存。
- 保存中断和系统工作状态等信息，以便使CPU和系统能及时了解机器运行状态和程序运行状态
- 因此，状态条件寄存器是一个由各种状态条件标志拼凑而成的寄存器

通用寄存器

- 当算术逻辑单元（ALU）执行算术或逻辑运算时，为ALU提供一个工作区。
  - 例如：在执行一次加法运算时，选择两个操作数（分别放在两个寄存器）相加，所得的结果送回其中一个寄存器（如R2）中，而R2中原有的内容即被替换。
- 在众多通用寄存器中，其中任何一个可存放源操作数，也可存放结果操作数

思考题

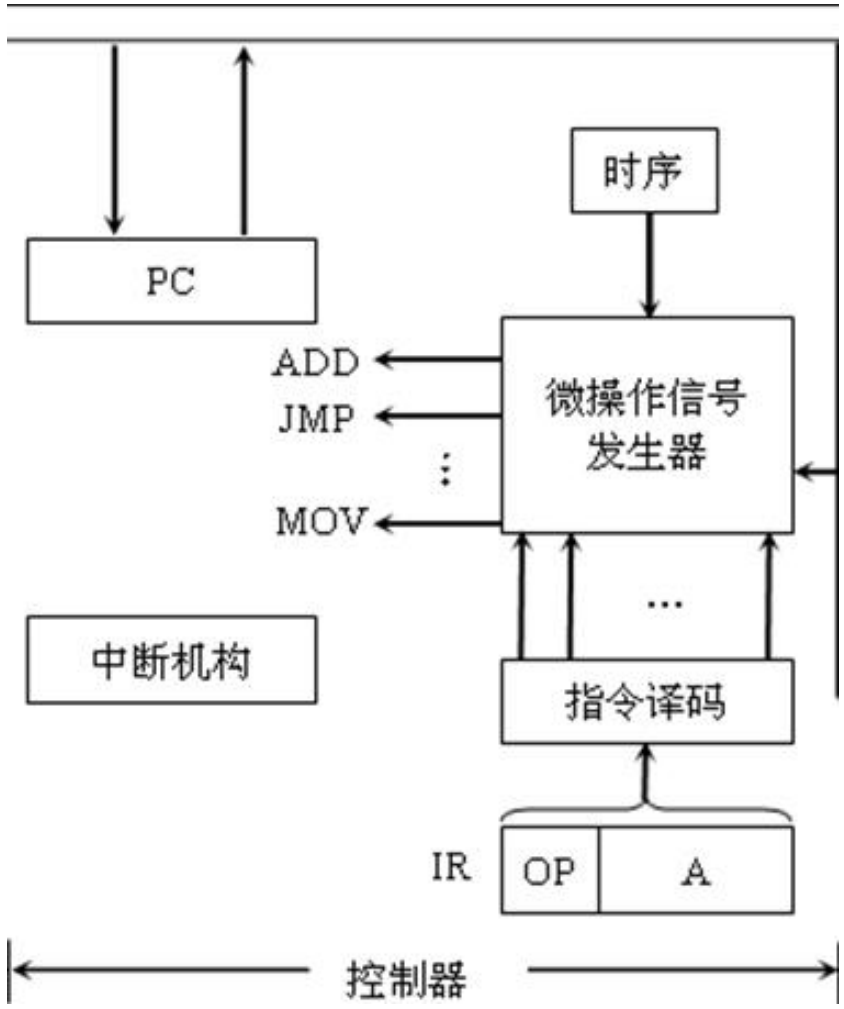
- 下列寄存器中，汇编语言程序员可见的是（）。
  - A. 存储器地址寄存器（MAR）
  - B. 程序计数器（PC）
  - C. 存储器数据寄存器（MDR）
  - D. 指令寄存器（IR）

CPU的组成与功能

- CPU概述
- CPU中的主要寄存器
- 控制器
- 运算器

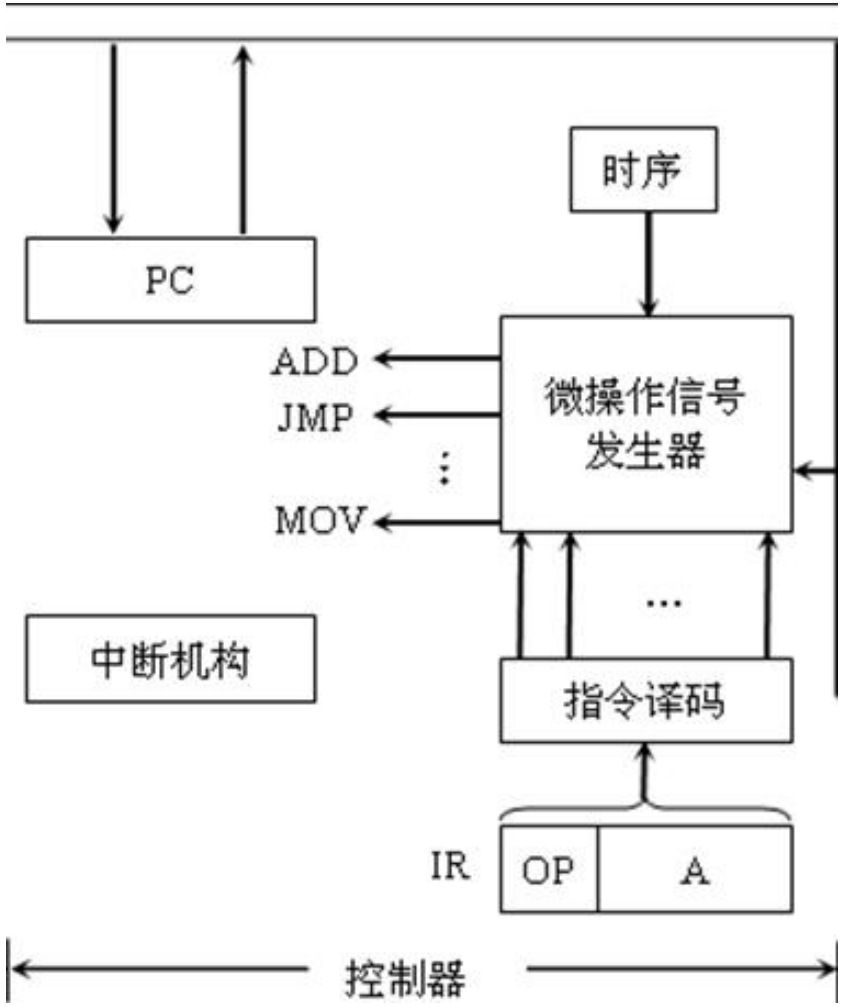
控制器的基本组成

- PC (Program Counter)
  程序计数器
- IR (Instruction Register)指令寄存器
- ID (Instruction Decoder)指令译码器
- OC (Operate Controller)操作控制器
- TG (Timer Generator)
  时序发生器



指令译码器 (Instruction Decoder)

- 对指令进行分段（操作码、地址码）译码，指出指令的操作方式、寻址方式
- 为操作控制器提供输入信号



操作控制器 (Operate Controller)

- 根据指令操作码和地址码、时序信号，产生各种控制信号序列，建立正确的数据通路，从而完成取指令和执行指令的控制。
  - 硬布线控制器　（时序逻辑型）（硬件实现）
  - 微程序控制器　（存储程序型）（软件实现）
- **数据通路—执行部件间（寄存器）传送信息的通路**
  - 指令执行过程中，数据所经过的路径，包括路径中部件
- **操作控制器为数据通路的建立提供各种操作信号**
  - 操作控制器的功能，就是根据指令操作码和时序信号，产生各种操作控制信号，以便正确地选择数据通路，把有关数据打入到一个寄存器，从而完成取指令和执行指令的控制

操作控制器

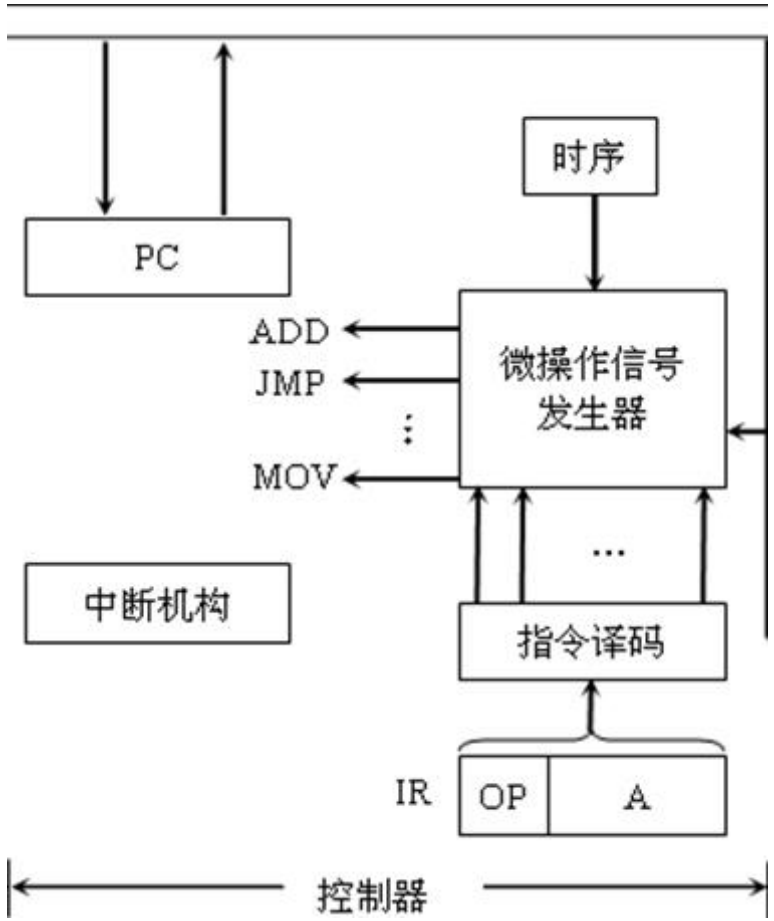
- 根据设计方法不同，操作控制器可分为时序逻辑型与存储逻辑型两种。第一种称为**硬布线控制器**，它是采用时序逻辑技术来实现 ；第二种称为**微程序控制器**，它是采用存储逻辑来实现的。
- 硬布线控制器
  - 硬布线控制器，又称组合逻辑控制器，直接由各种类型的逻辑门电路和触发器等构成。
  - 时序逻辑控制器的最大优点是速度快，但是时序控制信号形成部件的结构不规整，使得设计、调试、维修较困难，难以实现设计自动化。

操作控制器

- **微程序控制器**
  - 微程序控制器是采用存储逻辑来实现的，也就是把微操作信号代码化，使每条机器指令转化成为一段微程序并存入一个专门的存储器(控制存储器)中，微操作控制信号由微指令产生。
  - 微程序控制器的设计思想和组合逻辑设计思想截然不同。它具有设计规整、调试、维修以及更改、扩充指令方便的优点，易于实现自动化设计，已成为当前控制器的主流。但是，由于它增加了一级控制存储器，所以指令执行速度比组合逻辑控制器慢。

时序产生器 (Timer Generator)

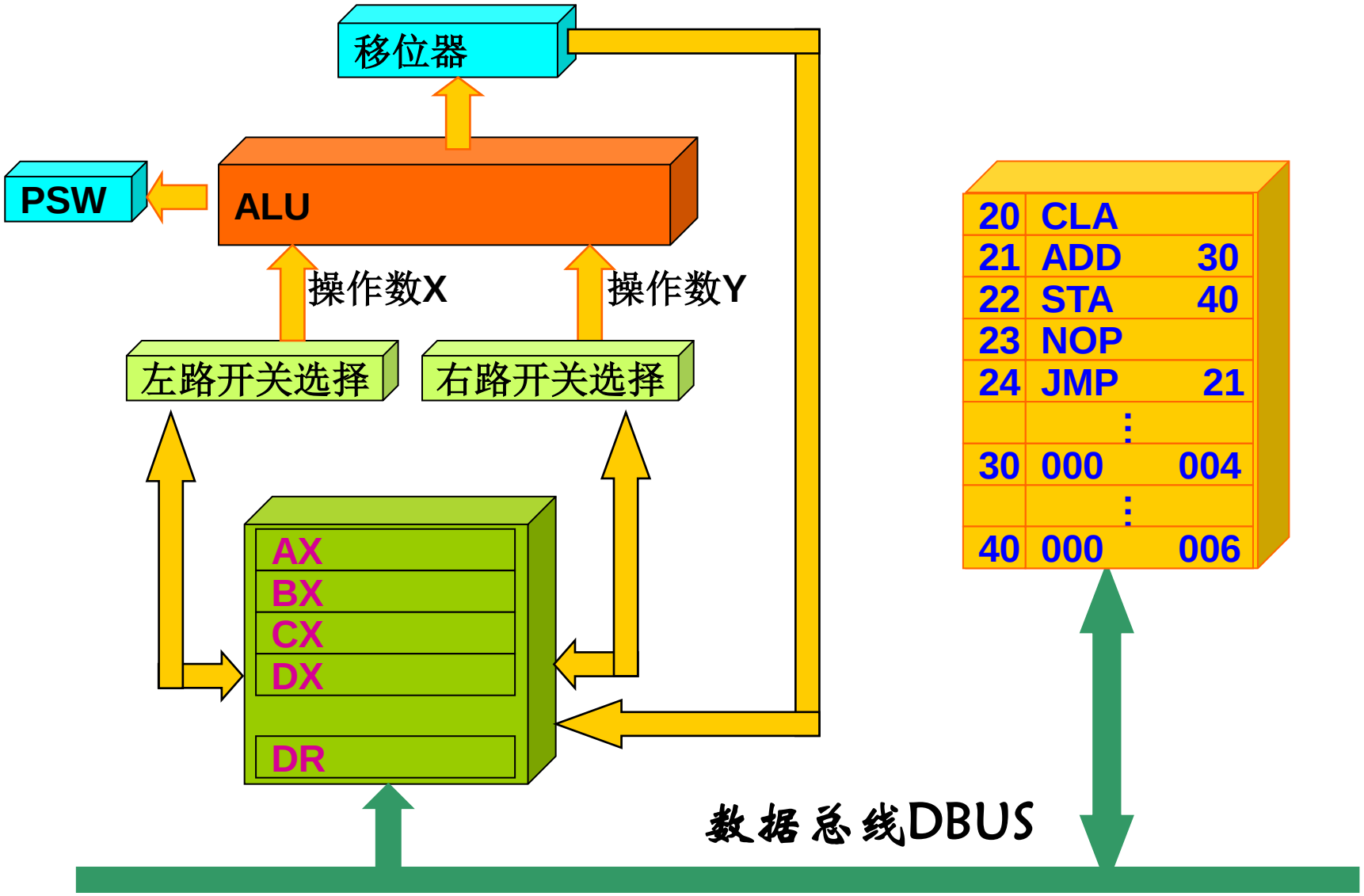
- 产生各种时序信号 (电位, 脉冲)
- 对各种操作实施时间上的控制
  - 产生时间、持续时间

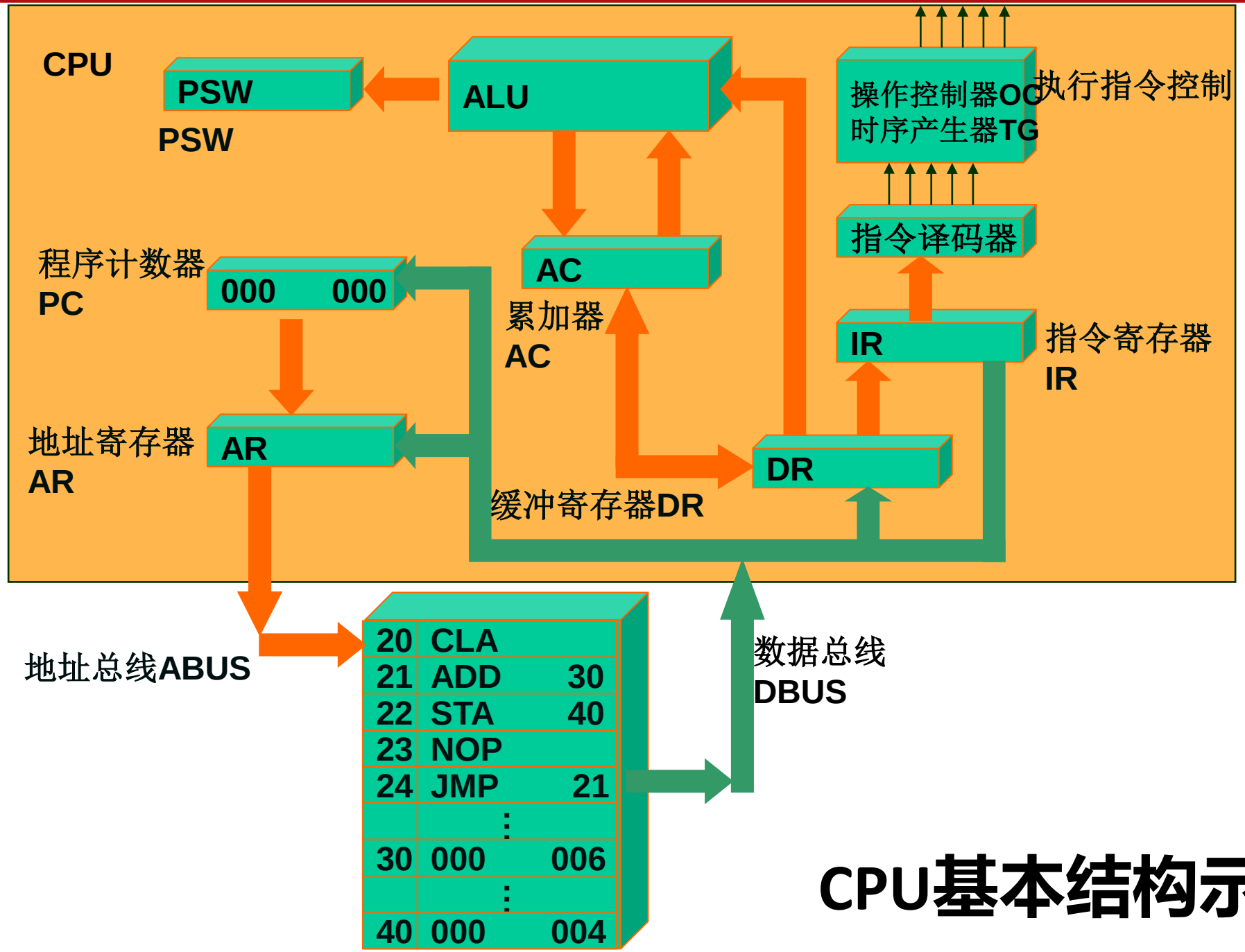


CPU的组成与功能

- CPU概述
- CPU中的主要寄存器
- 控制器
- **运算器**

运算器的基本组成


移位器
PSW
ALU
操作数X
操作数Y
左路开关选择
右路开关选择
AX
BX
CX
DX
DR
20 CLA
21 ADD 30
22 STA 40
23 NOP
24 JMP 21
30 000 004
40 000 006
数据总线DBUS


CPU
PSW
PSW
ALU
操作控制器OC
时序产生器TG
执行指令控制
指令译码器
AC
累加器
AC
程序计数器
PC
000 000
IR
指令寄存器
IR
地址寄存器
AR
AR
DR
缓冲寄存器DR
地址总线ABUS
数据总线
DBUS
20 CLA
21 ADD 30
22 STA 40
23 NOP
24 JMP 21
30 000 006
40 000 004

CPU基本结构示意

本节目录

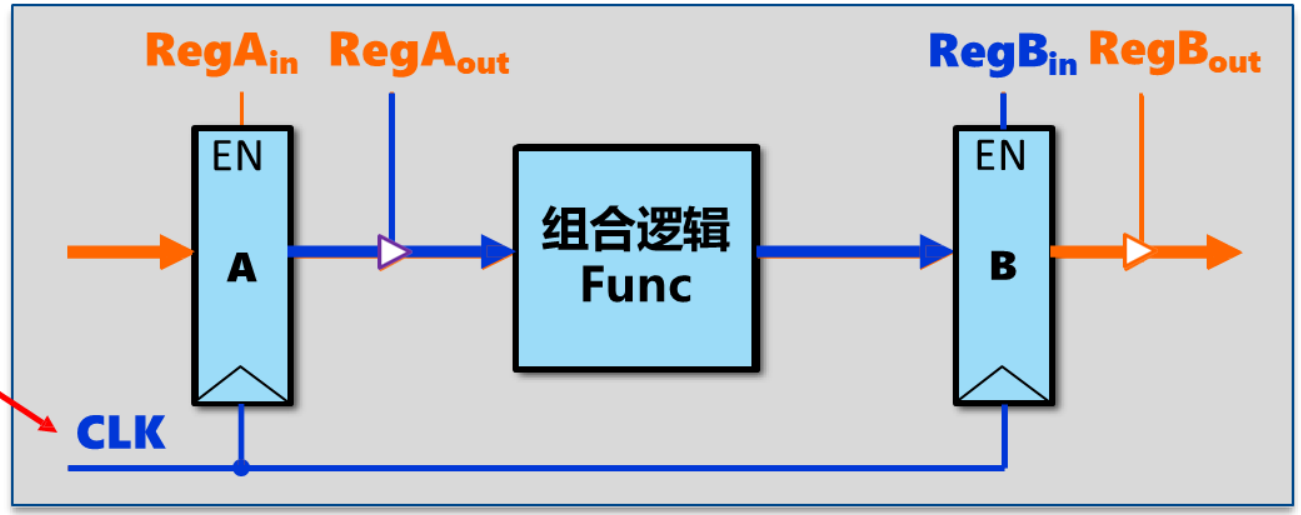
- CPU的组成与功能
- 数据通路与总线结构

数据通路（DataPath）基本概念

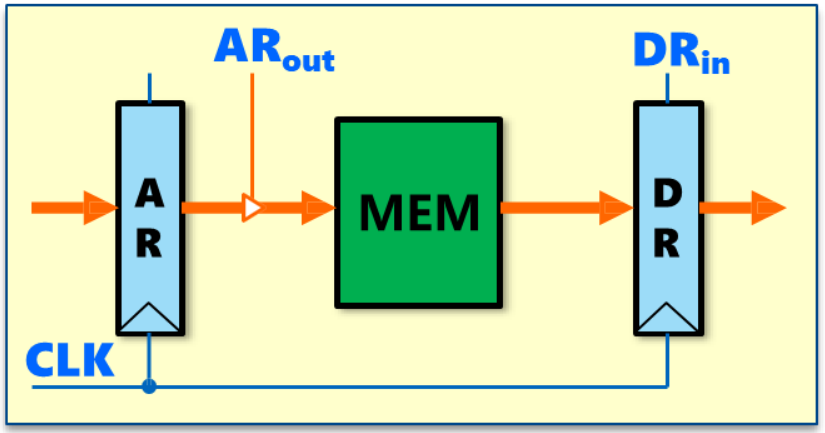
- 数据通路------执行部件间传送信息的路径。
  - 通路的建立由控制信号控制，受时钟驱动；
  - 不同指令、同一指令在执行的不同阶段的数据通路不同；
  - 数据通路分类：共享通路（总线）、专用通路
    - 指令执行流程、执行效率
    - 微操作控制信号的时序安排

数据通路抽象模型

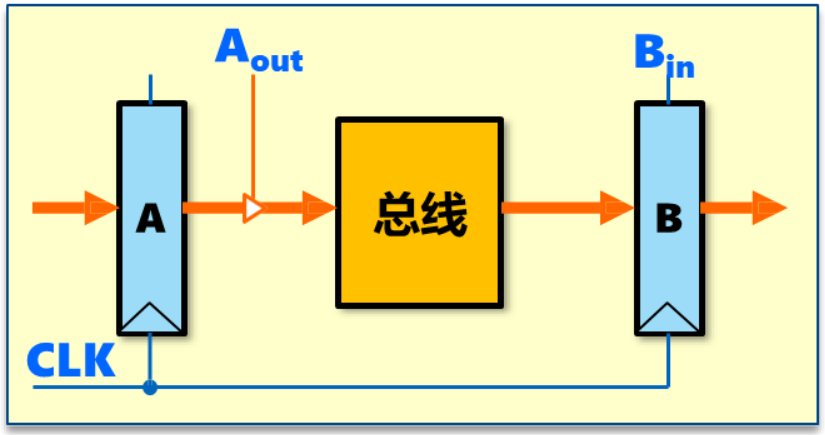
时钟频率?



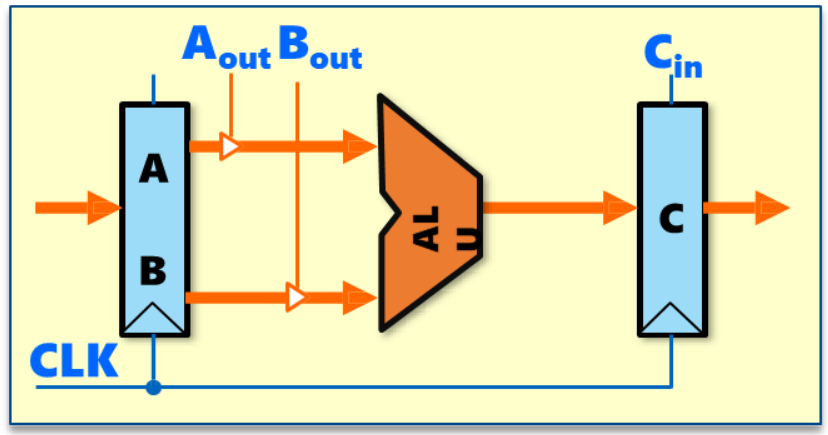
Func (A) → B



访存通路

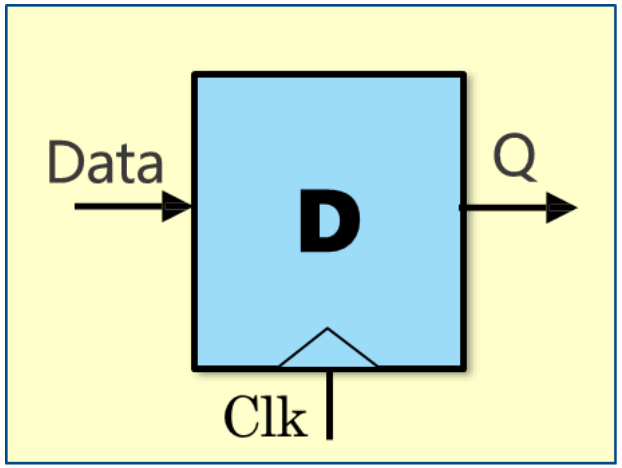


总线传输

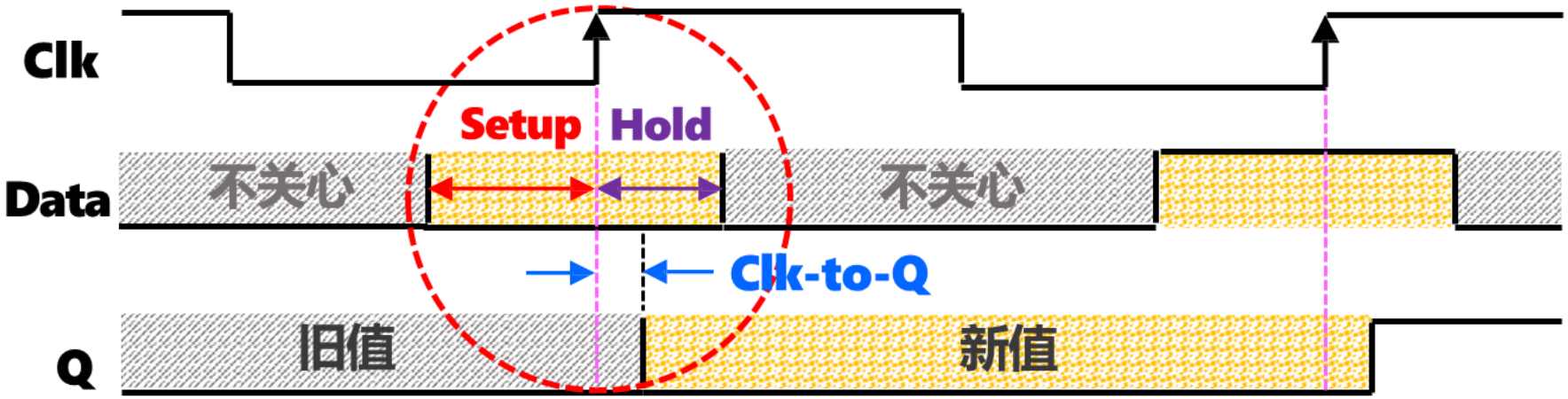


运算通路

D触发器定时模型

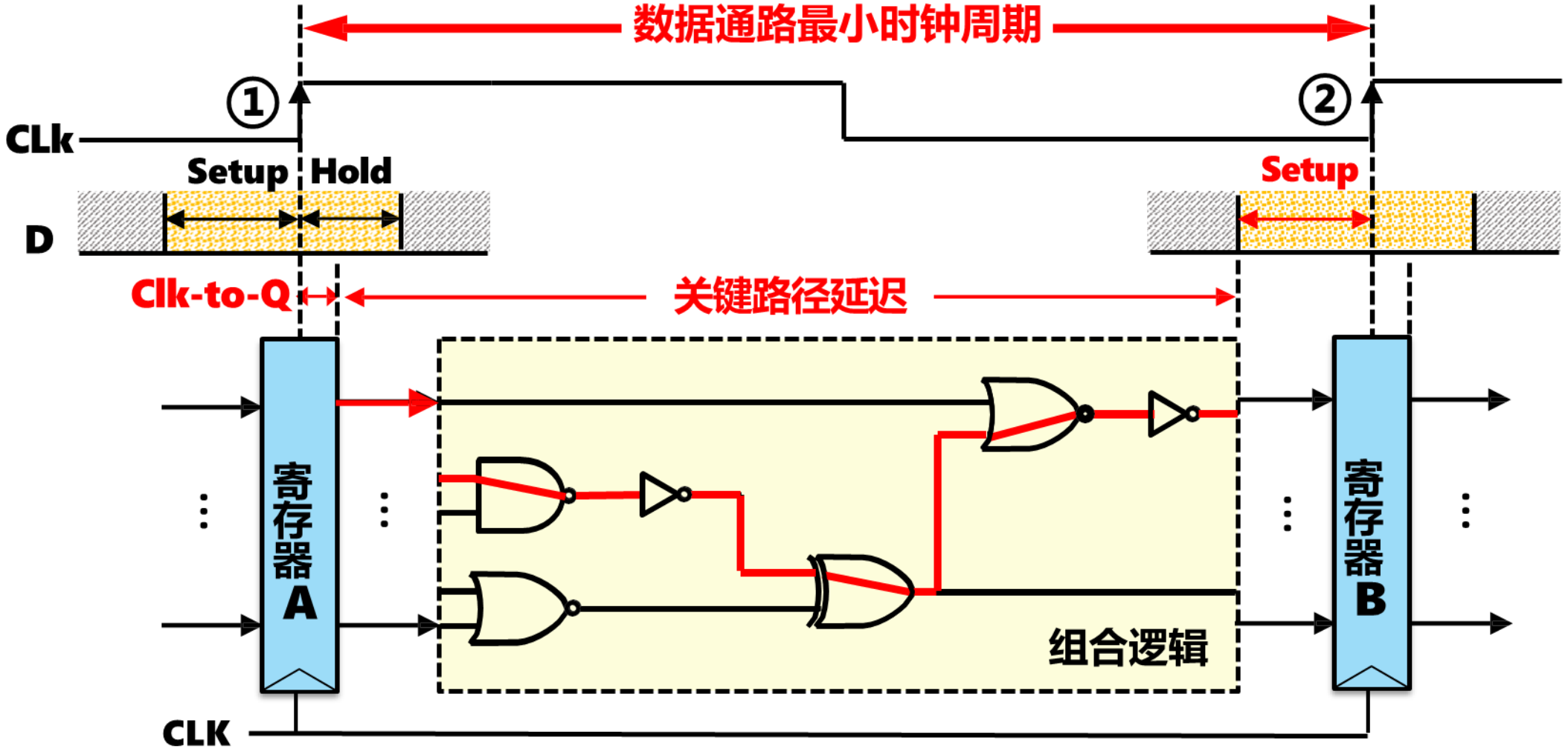


- 时钟触发前输入须稳定一段 **建立时间（Setup Time）**
- 时钟触发后输入须稳定一段 **保持时间（Hold Time）**
- 时钟触发到输出稳定的时间 **触发器延迟 Clk_to_Q**



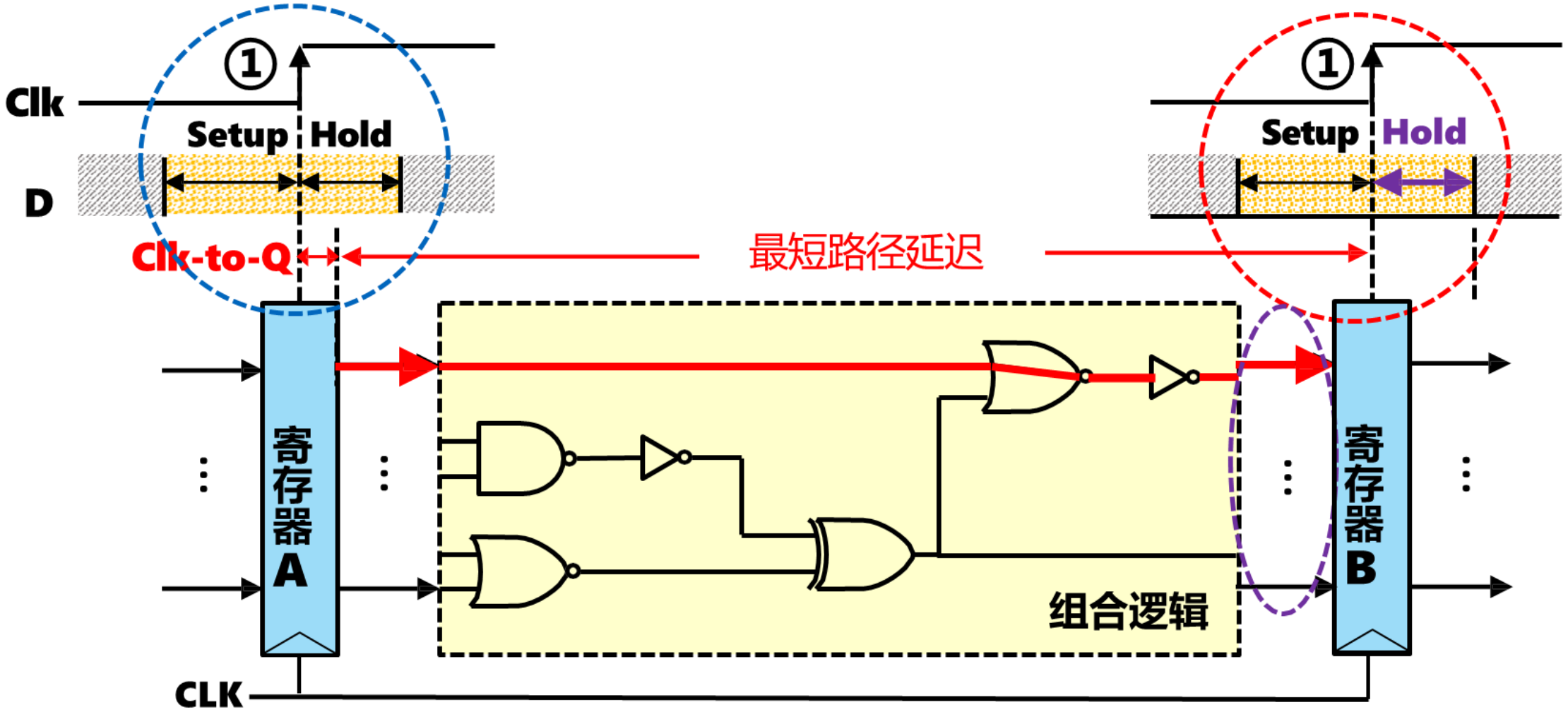
数据通路与时钟周期

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



保持时间违例

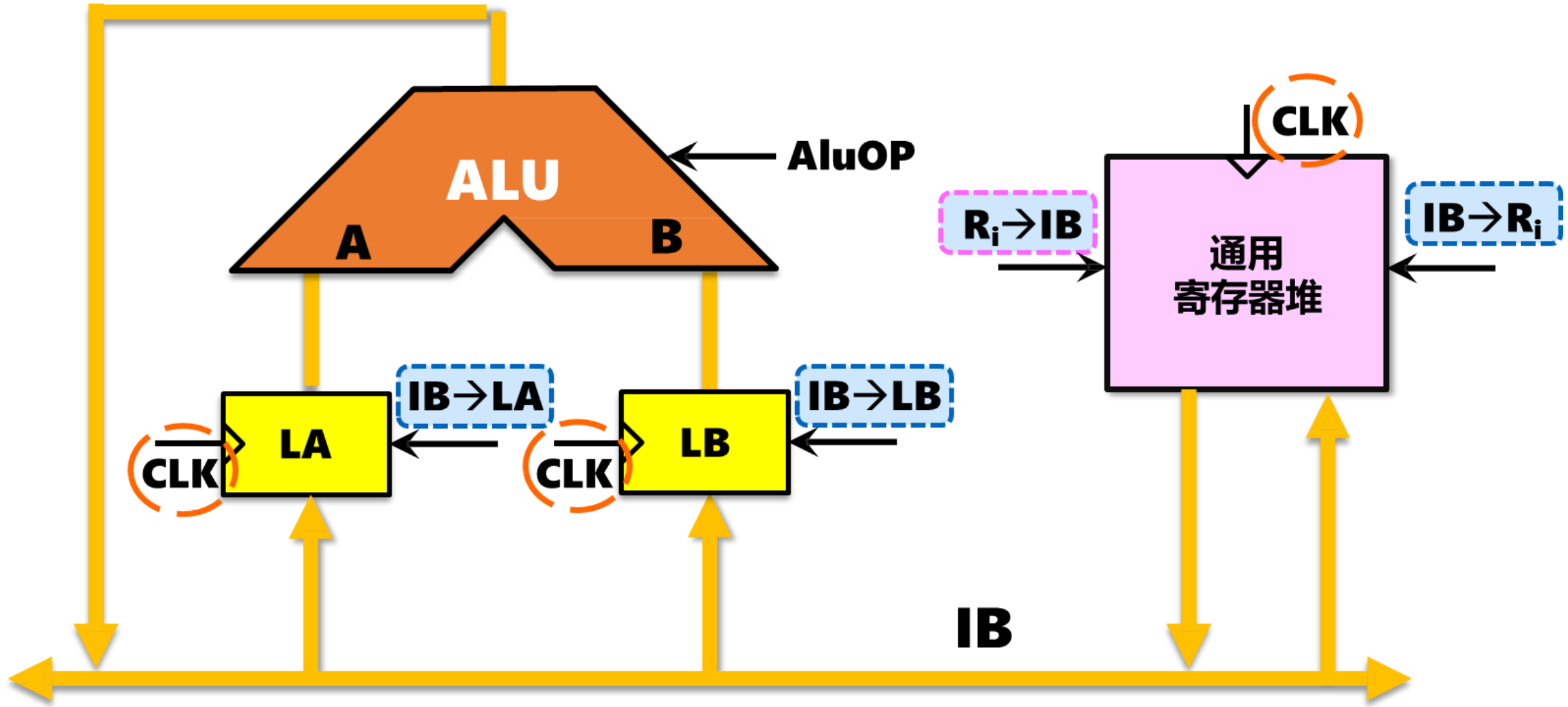
- Clk_to_Q + 最短路径时延 > Hold_Time



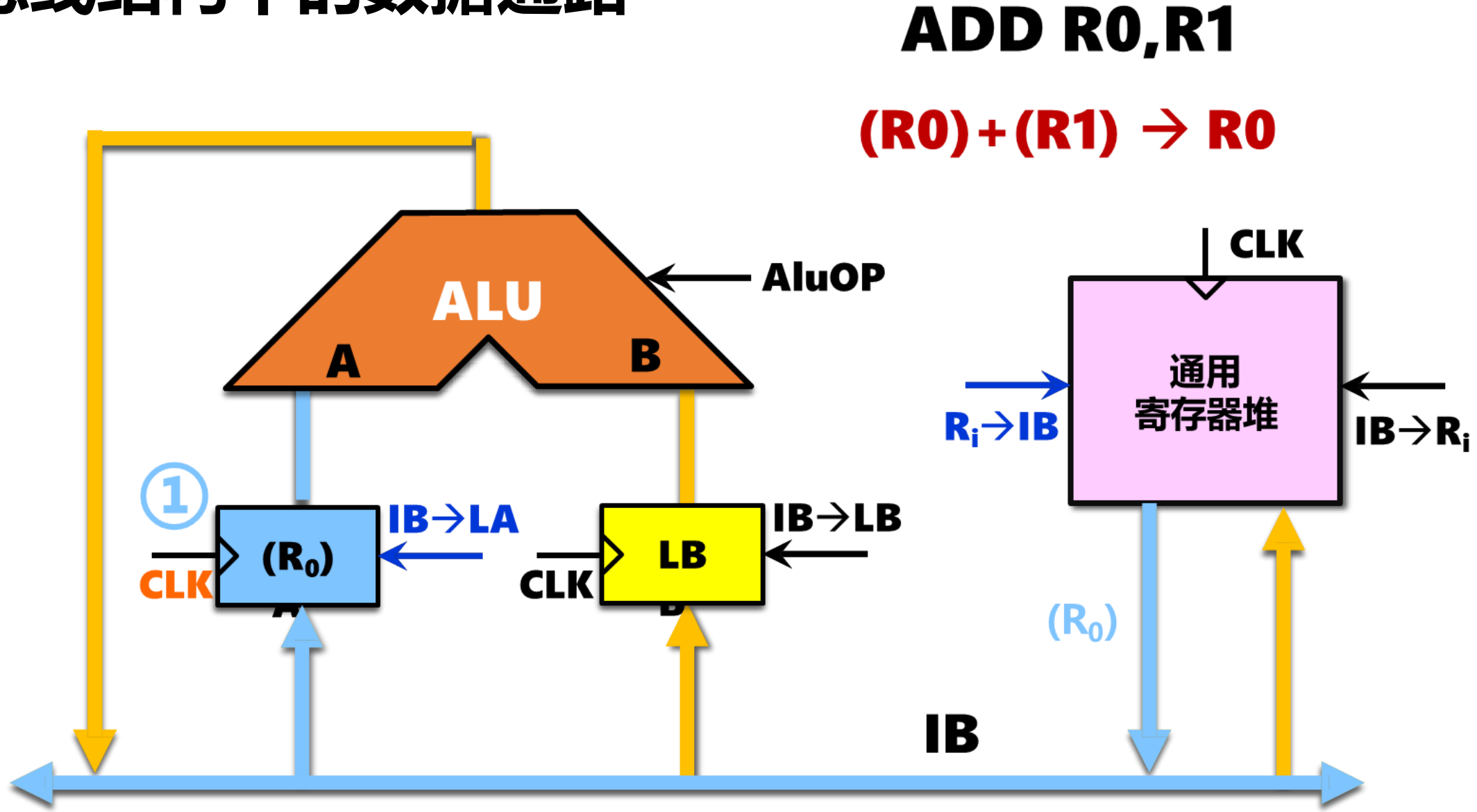
数据通路分类

- 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
  - 可以看做多总线结构

单总线结构中的数据通路



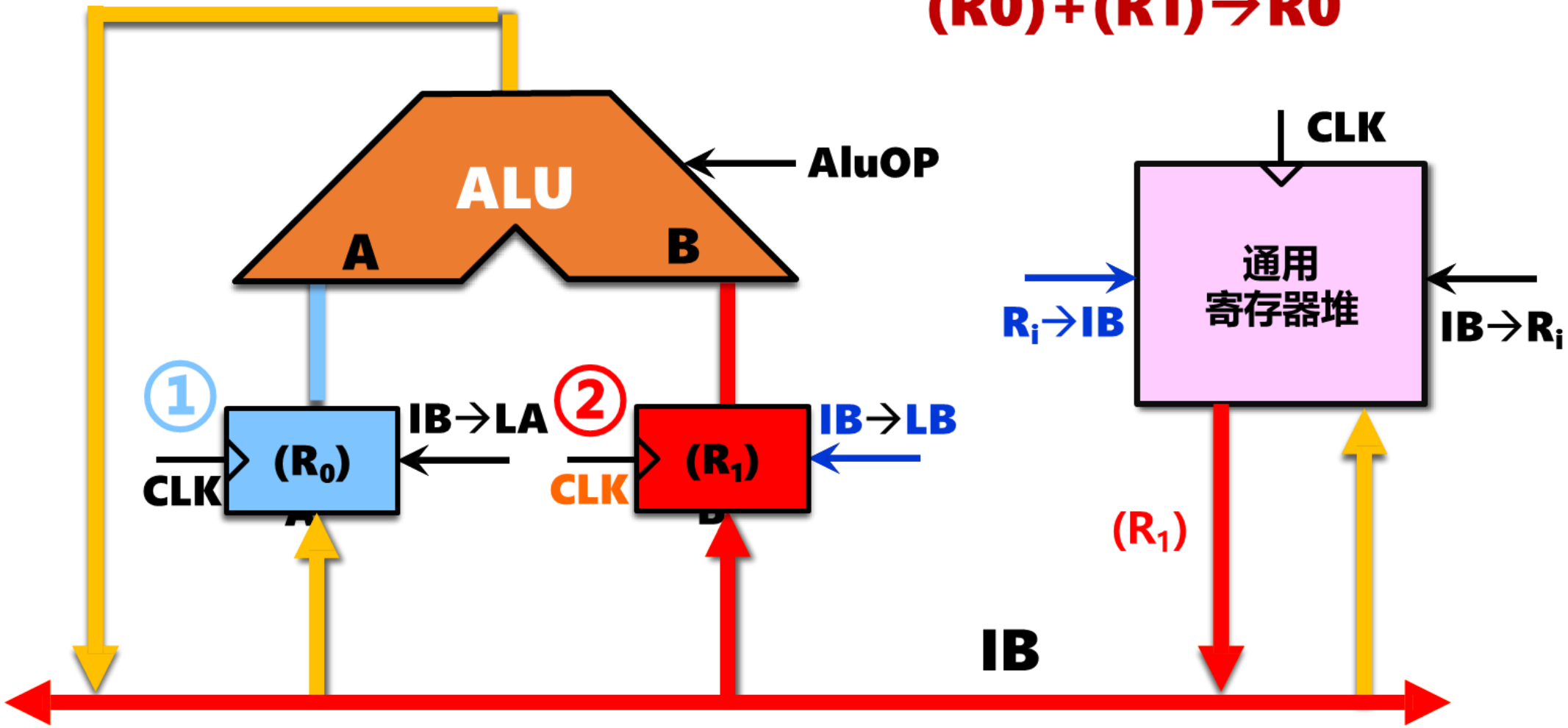
单总线结构中的数据通路



单总线结构中的数据通路

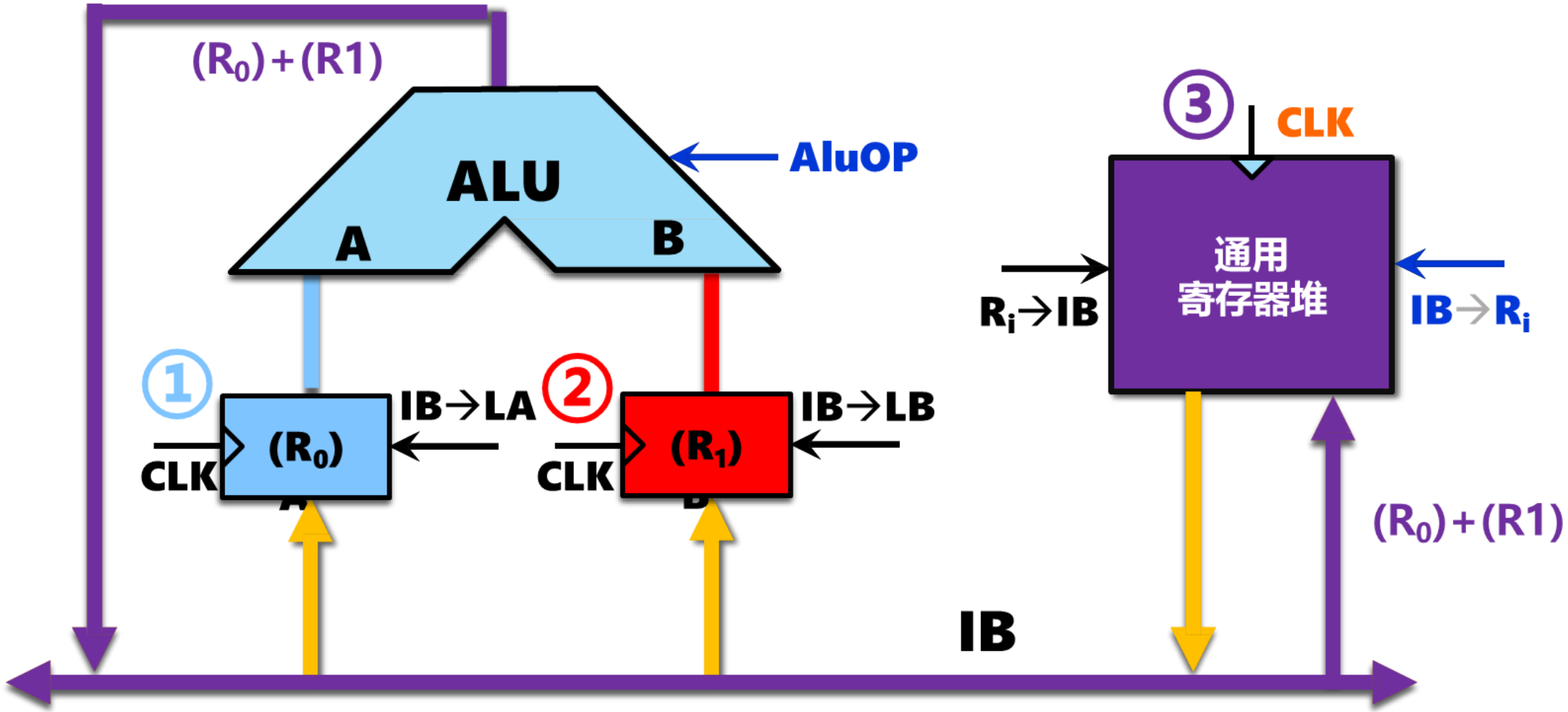
ADD R0,R1

(R0)+(R1)→R0

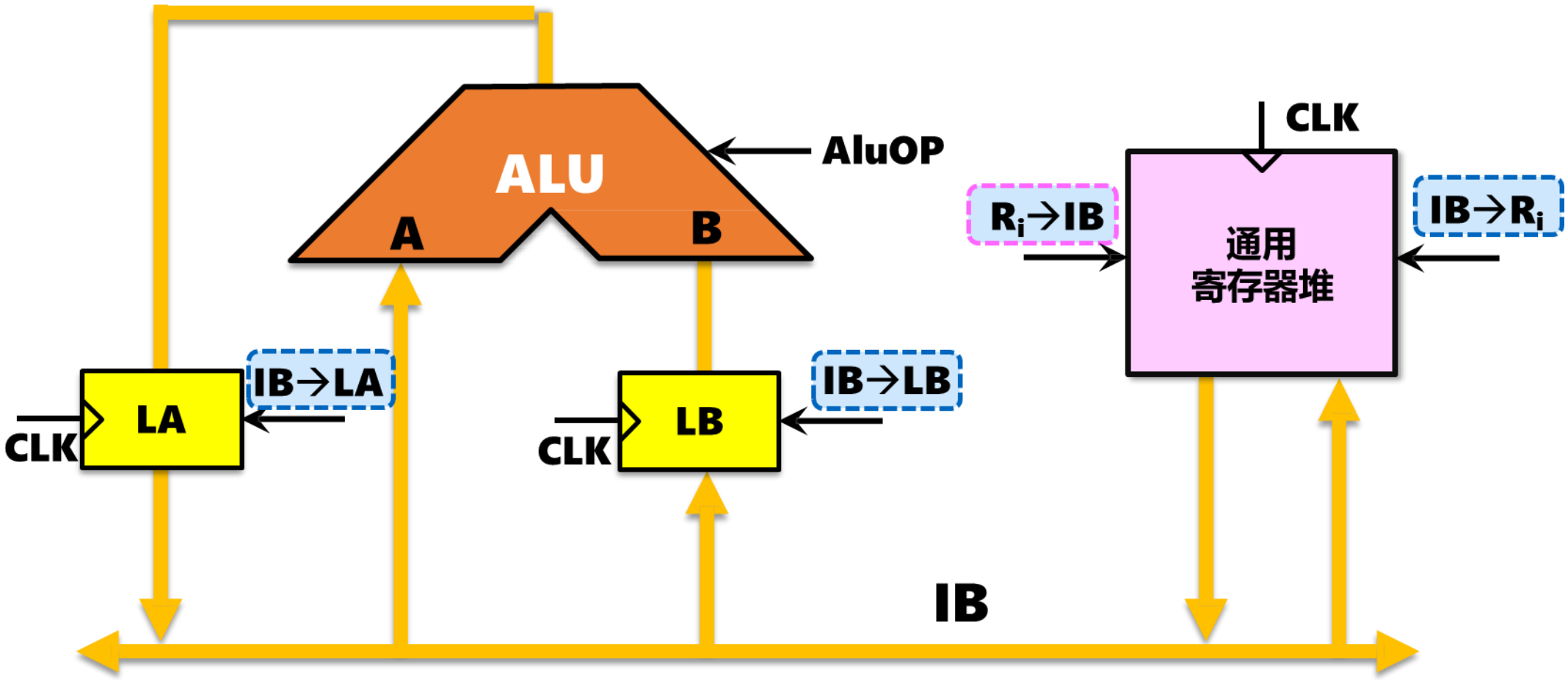


单总线结构中的数据通路

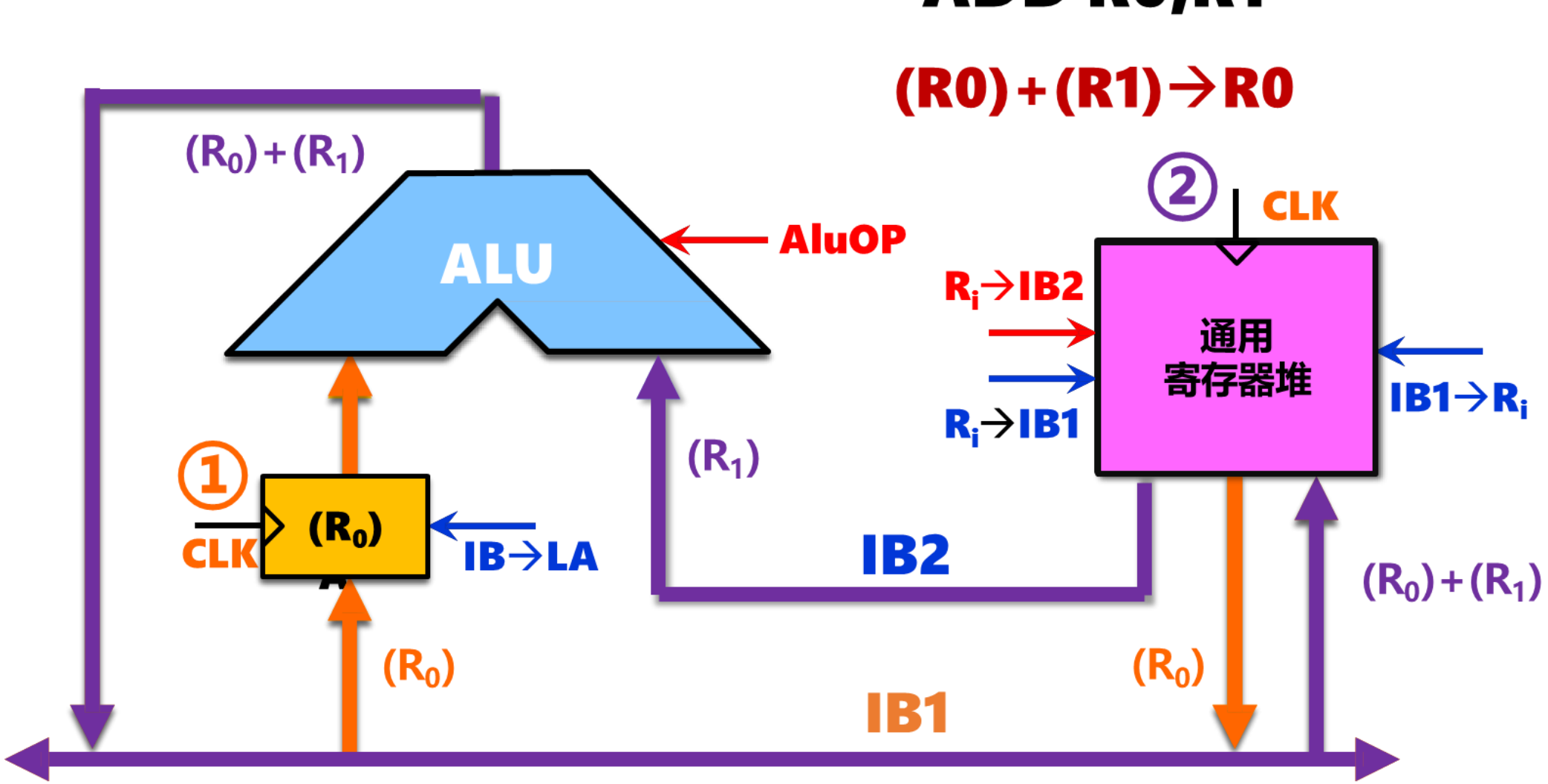
- 单总线，两个锁存器，3个时钟周期



单总线结构中的数据通路



双总线结构与运算通路

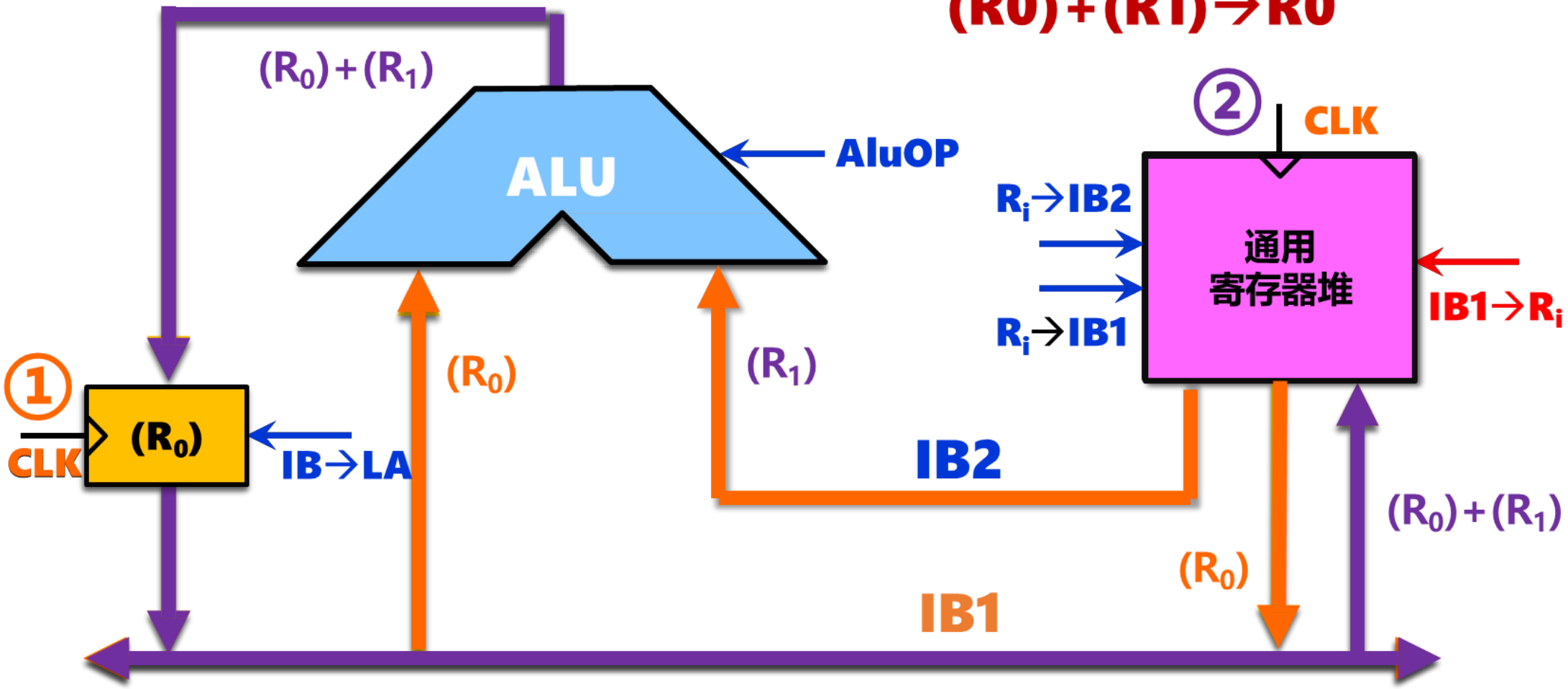


- 双总线，1个锁存器，2个时钟周期

双总线结构与运算通路

ADD R0,R1

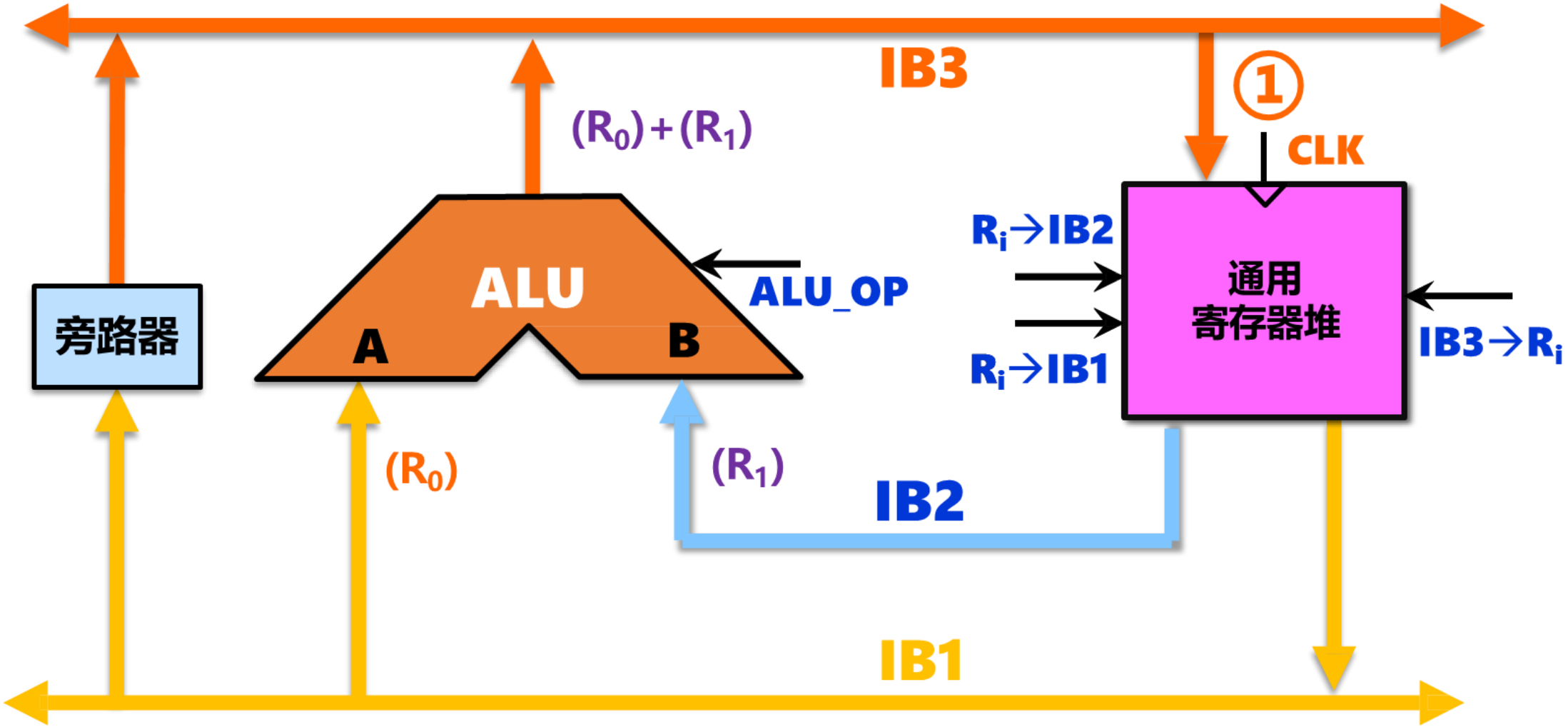
(R0)+(R1)→R0



- 双总线，1个锁存器，2个时钟周期

三总线结构与运算通路

- 三总线，0个锁存器，1个时钟周期



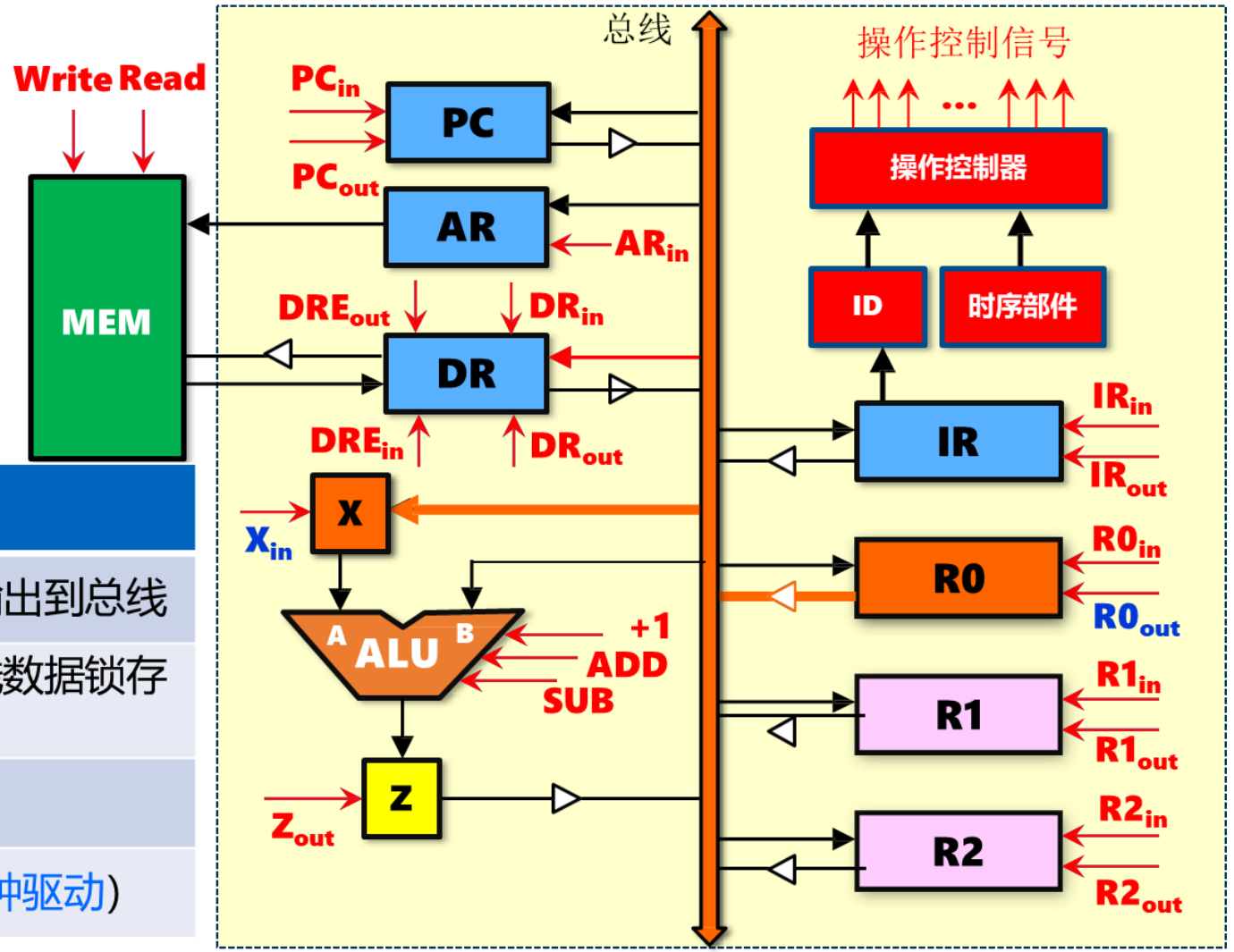
数据通路与总线结构小结

- 单总线，2个锁存器，3个时钟周期
- 双总线，1个锁存器，2个时钟周期
- 三总线，0个锁存器，1个时钟周期
- 总线越多，性能越好

数据通路实例

1 单总线结构CPU

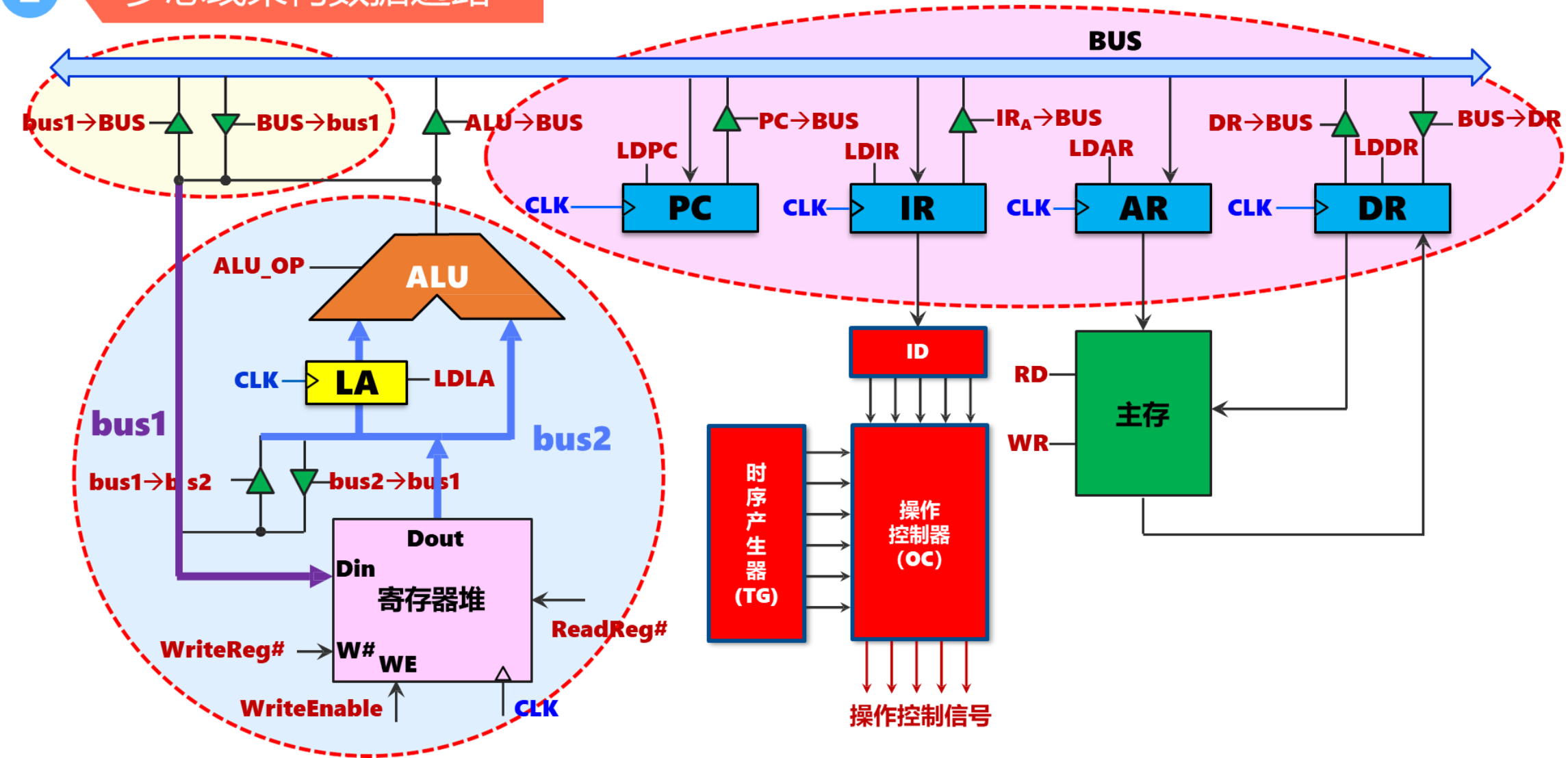
- 主要部件都连接在总线上
- 各部件间通过总线进行传输



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）

数据通路实例

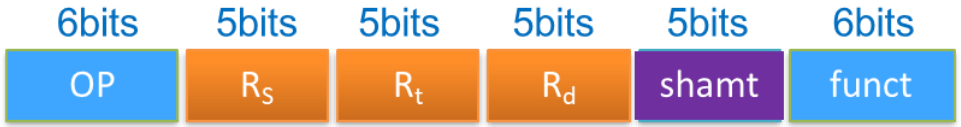
2 多总线架构数据通路

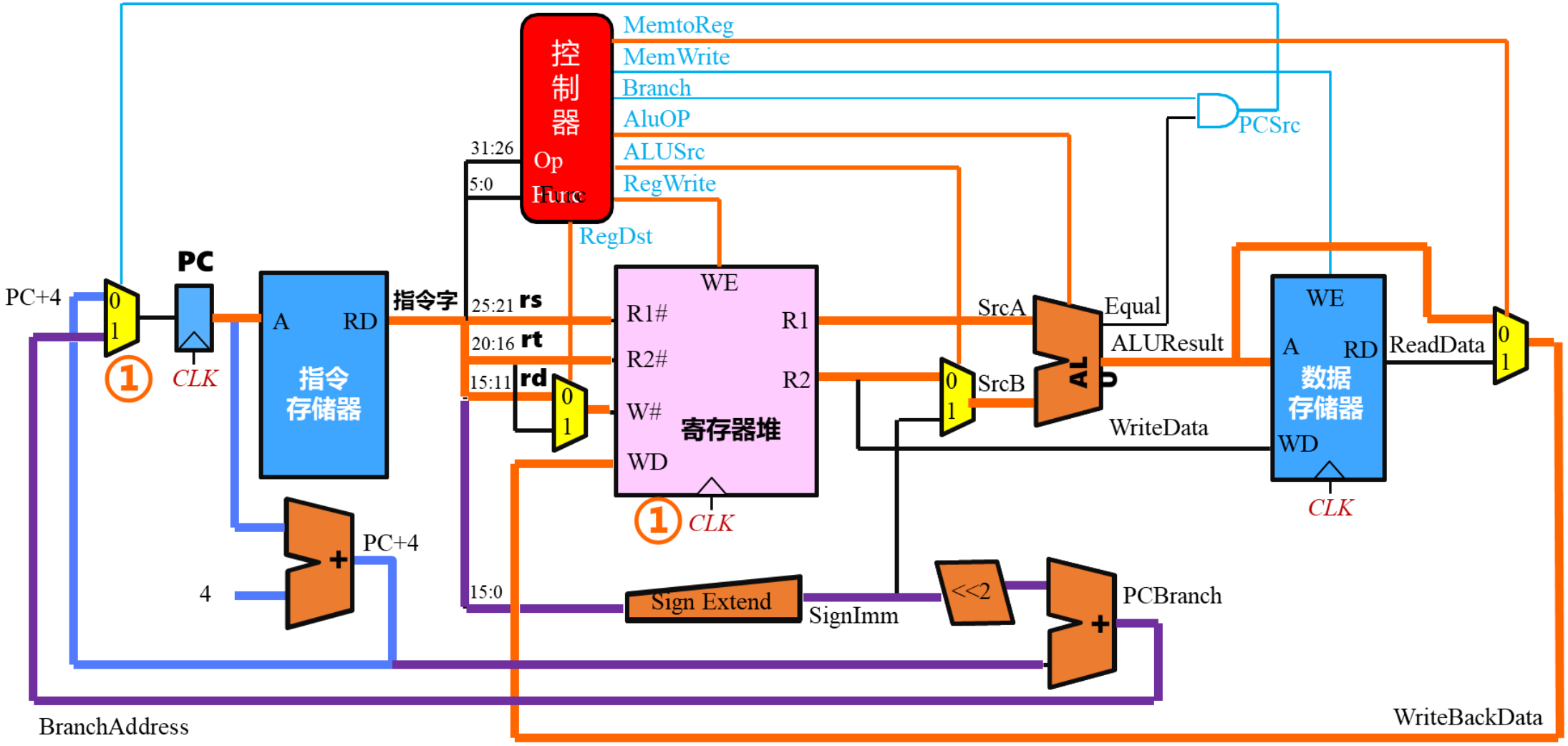


数据通路实例

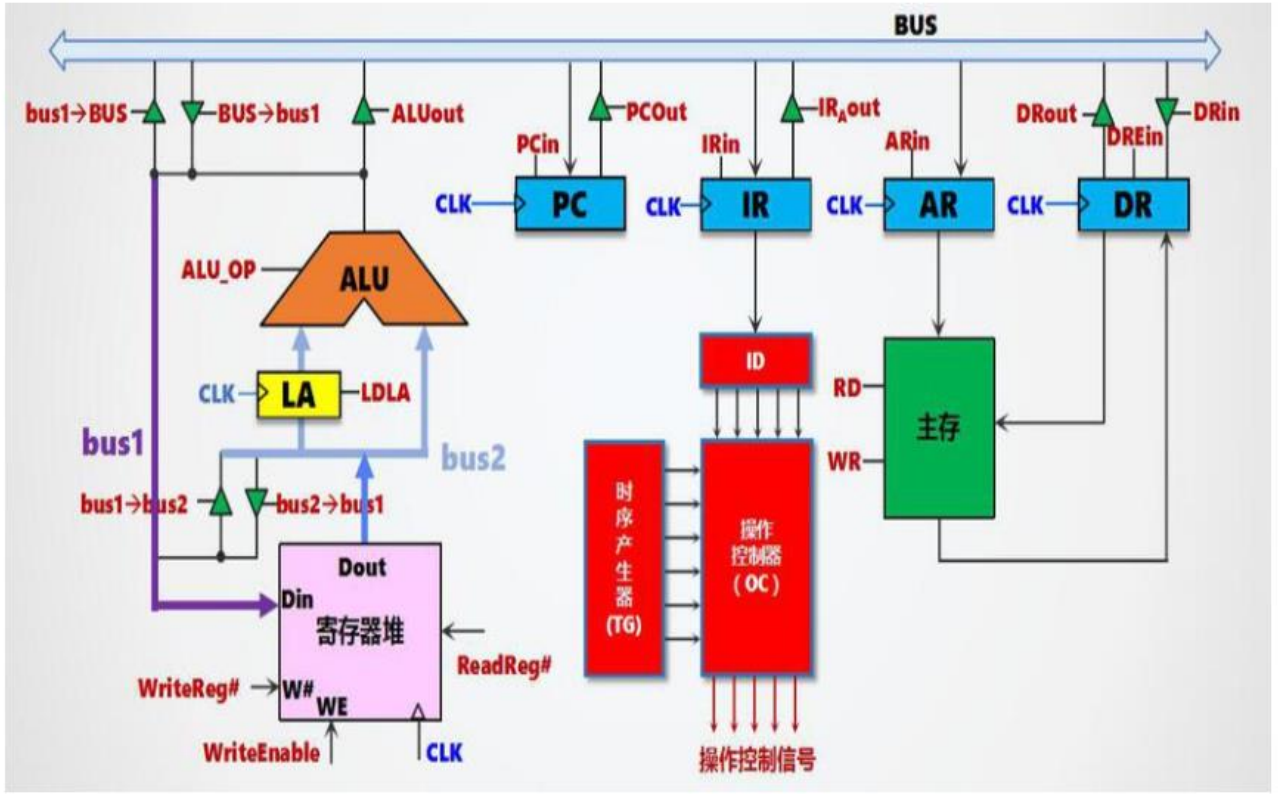
3 专用通路 单周期MIPS

R 型指令





课堂练习



假定PCout 兼有使PC +"1"的功能，围绕该图的下列描述中，正确的是（ ） （多选）

A. 实现 PC + "1"的数据通路是 PC -> PC

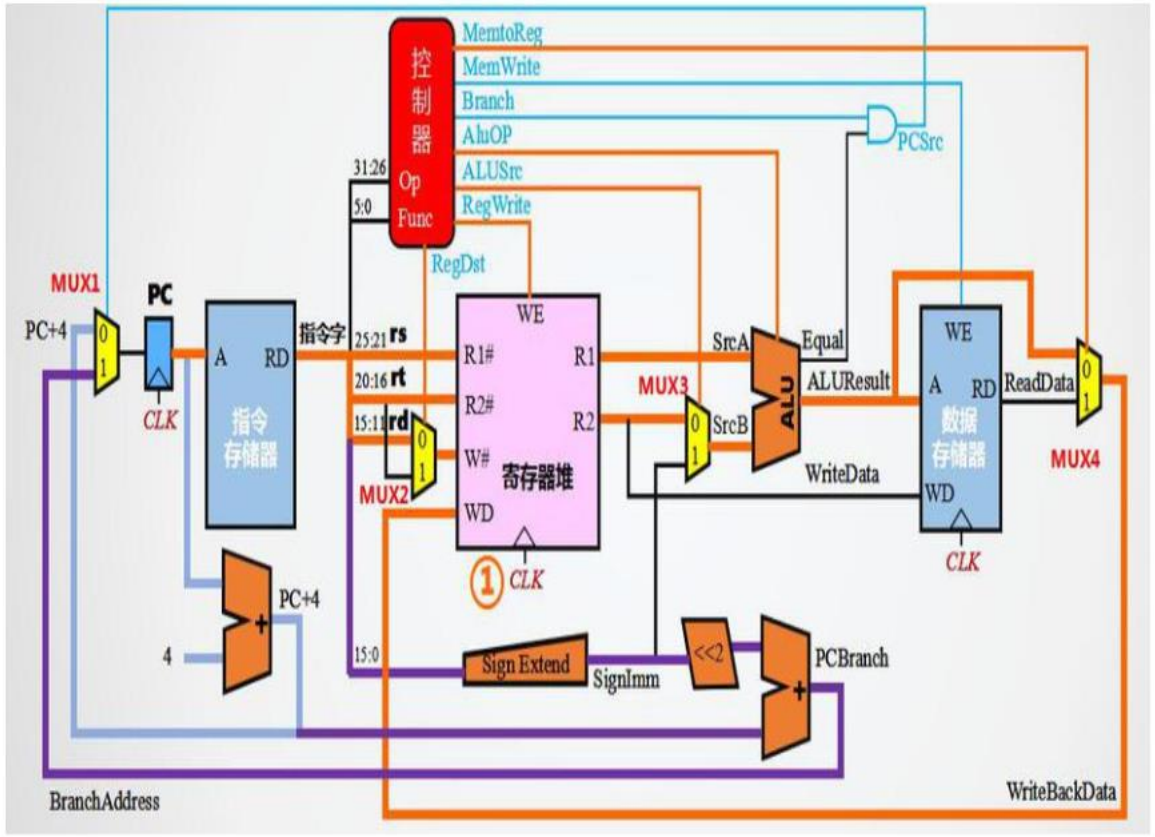
B. 取指令的数据通路为： PC-> AR -> 主存 -> DR -> IR

C. PCout、DRout 、 IRA out 及 bus1->BUS 都一定不能同时有效

D. 某采用间接寻址的指令其执行阶段的数据通路一定包含:
IR -> AR ->主存 ->DR->AR->主存->DR

课堂练习

下图为基于专用通路的 CPU结构



围绕该图的下列描述中，正确的是（ ）（多选）

A. 取指令的数据通路为 PC-> 指令寄存器

B. 完成PC 增量操作的数据通路为： PC-> PC+1 部件-> PCBranch ->MUX1->PC

C. 由于使用了专题通路结构和多路选择器，所有功能部件的输出将不再有数据冲突

D. 图中PC 既是指令地址寄存器也是数据地址寄存器