

3.2 算术逻辑单元

3.2.1 单元电路

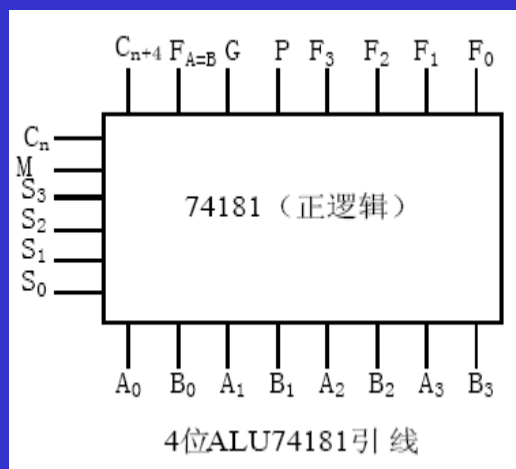
1. 寄存器

- 8D 锁存器
- 三态锁存器

2. 移位寄存器

3.2.2 算术逻辑单元ALU

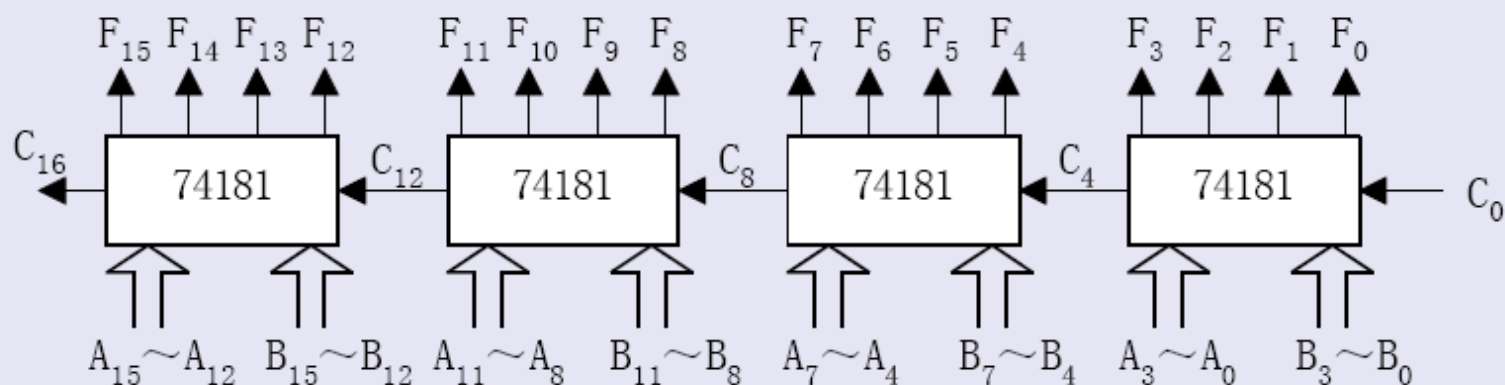
1. 4位ALU——74181



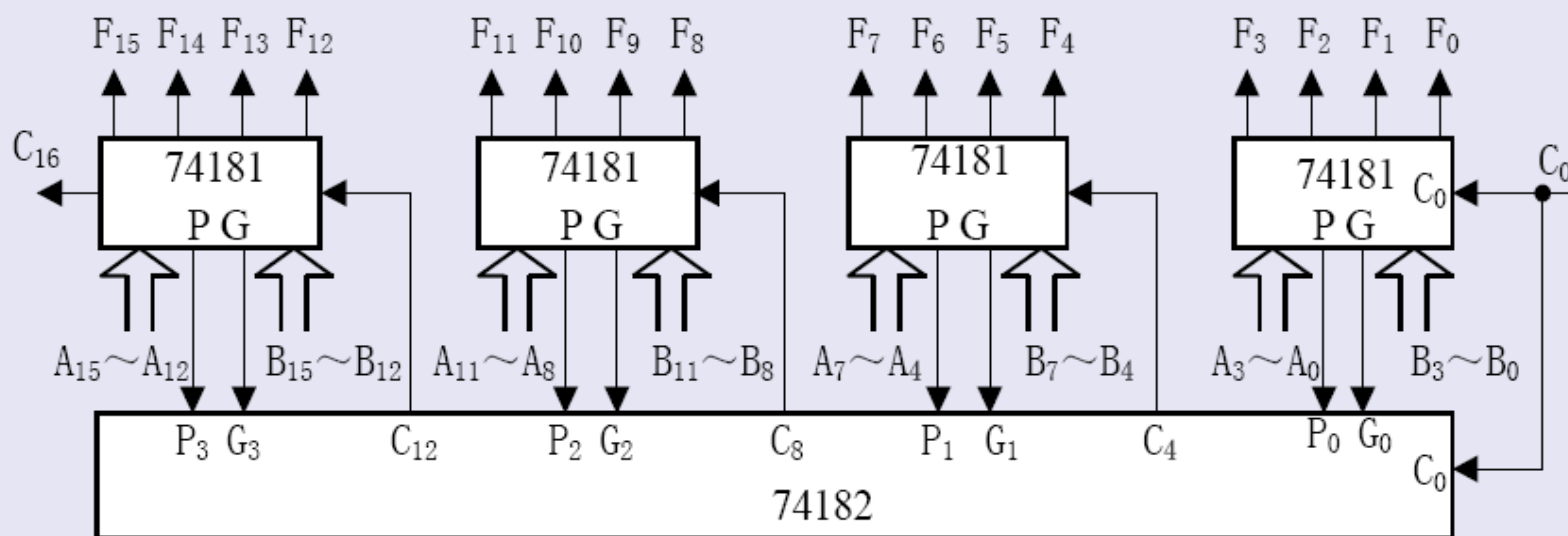
74181 的运算功能

操作选择	运 算 功 能		
	M=1	M=0 算 术 运 算	
$S_3 S_2 S_1 S_0$	逻辑运算	$C_n=1$ 无进位	$C_n=0$ 有进位
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A$ 加 1
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B)$ 加 1
0 0 1 0	$F=\overline{AB}$	$F=A+\overline{B}$	$F=(A+\overline{B})$ 加 1
0 0 1 1	$F=0$	$F=\text{减 } 1$	$F=0$
0 1 0 0	$F=\overline{AB}$	$F=A$ 加 \overline{AB}	$F=A$ 加 AB 加 1
0 1 0 1	$F=\overline{B}$	$F=(A+B)$ 加 \overline{AB}	$F=(A+B)$ 加 \overline{AB} 加 1
0 1 1 0	$F=A \oplus B$	$F=A$ 减 B 减 1	$F=A$ 减 B
0 1 1 1	$F=A\overline{B}$	$F=\overline{AB}$ 减 1	$F=A\overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A$ 加 AB	$F=A$ 加 AB 加 1
1 0 0 1	$F=\overline{A \oplus B}$	$F=A$ 加 B	$F=A$ 加 B 加 1
1 0 1 0	$F=B$	$F=(A+\overline{B})$ 加 AB	$F=(A+\overline{B})$ 加 AB 加 1
1 0 1 1	$F=AB$	$F=AB$ 减 1	$F=AB$
1 1 0 0	$F=1$	$F=2A$	$F=A$ 加 A 加 1
1 1 0 1	$F=A+\overline{B}$	$F=(A+B)$ 加 A	$F=(A+B)$ 加 A 加 1
1 1 1 0	$F=A+B$	$F=(A+\overline{B})$ 加 A	$F=(A+\overline{B})$ 加 A 加 1
1 1 1 1	$F=A$	$F=A$ 减 1	$F=A$

2. 级联工作



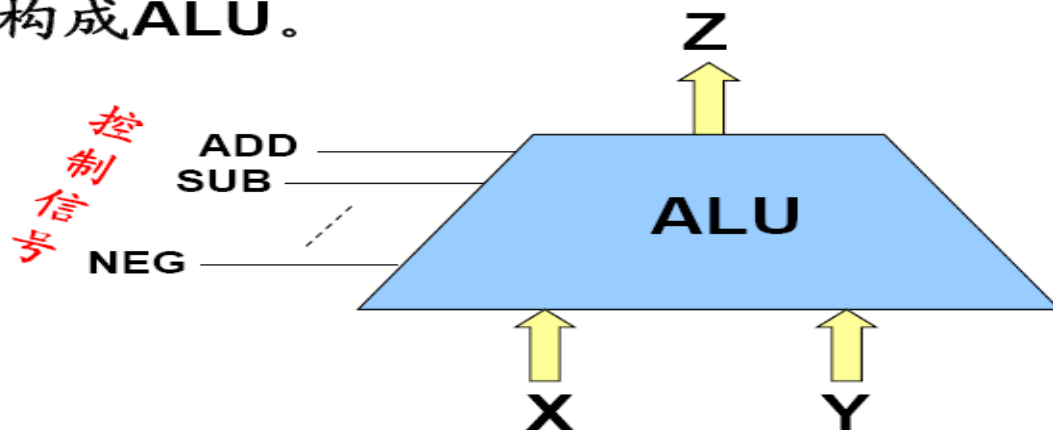
由74181构成组间串行进位的16位ALU



由74181和74182构成组内组间均并行进位的16位ALU

● ALU

将加减器、乘法器、除法器、移位器、与/或/非/异或逻辑部件、计数器、求补器等集合在一起构成ALU。



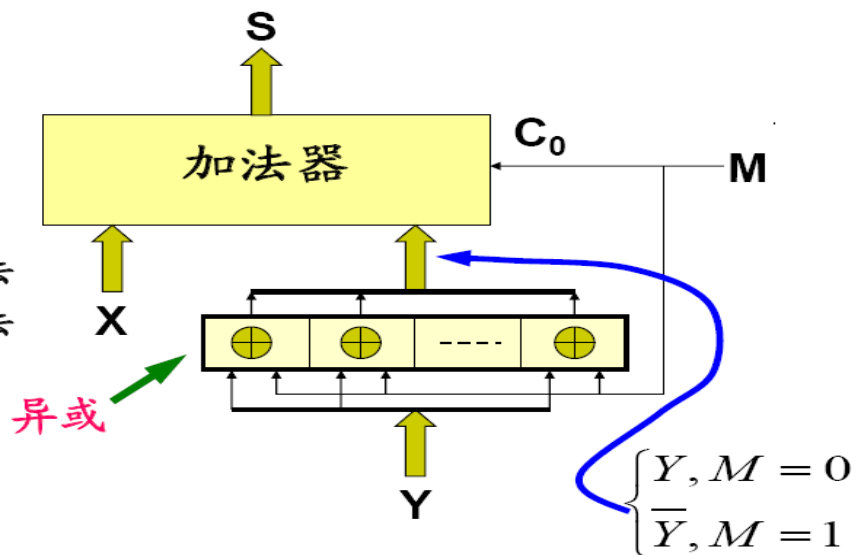
例如:

● 加减器

$$S = X \pm Y$$

M=0, 加法

M=1, 减法



3.2.3 运算器的结构

1. 定点运算器的组成

❖ 基本组成包括：

- **算术逻辑运算单元ALU**：核心部件，实现算术运算和逻辑运算
- **暂存器**：用来存放参与计算的数据及运算结果，它只对硬件设计者可见，即只被控制器硬件逻辑控制或微程序所访问
- **通用寄存器堆**：用于存放程序中用到的数据，它可以被软件设计者所访问。
- **内部总线**：用于连接各个部件的信息通道。
- 其他可选电路

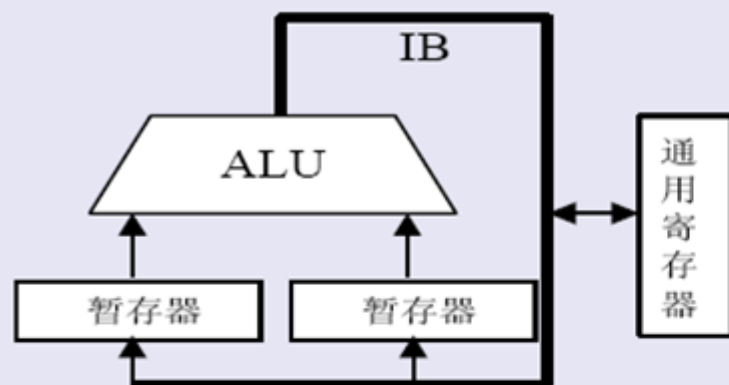
1. 定点运算器的组成

❖ 设计定点运算器，如何确定各部件的功能和组织方式是关键，这取决于以下几个方面：

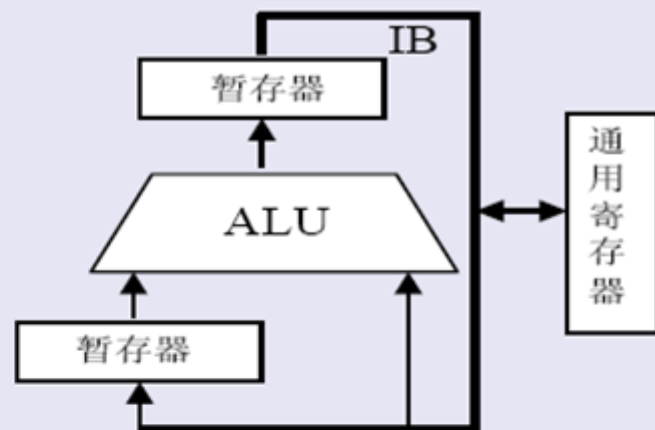
- 指令系统
- 机器字长
- 机器数及其运算原理
- 体系结构

- 单总线结构运算器
- 双总线结构运算器
- 三总线结构运算器

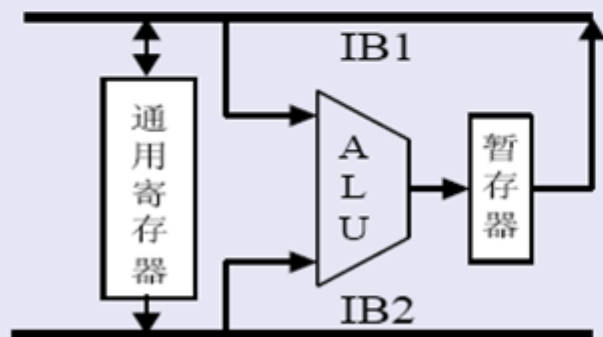
单总线、双总线、三总线结构运算器



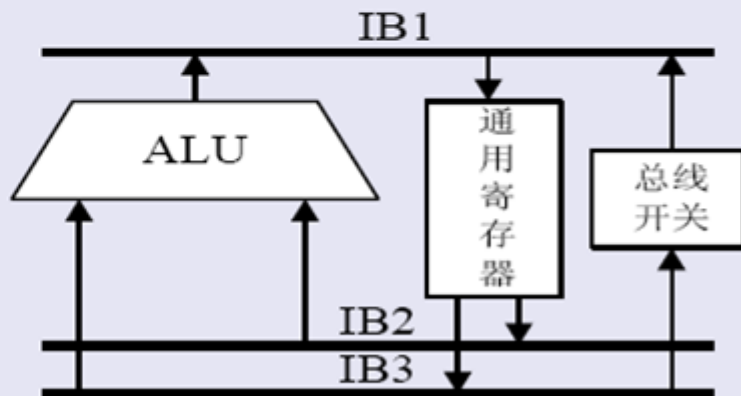
(a) 1



(a) 2



(b)



(c)

运算器的三种基本结构

单总线、双总线、三总线结构运算器

除了图中所表示的结构外，还有其他类似的连接形式。

需要强调的是：

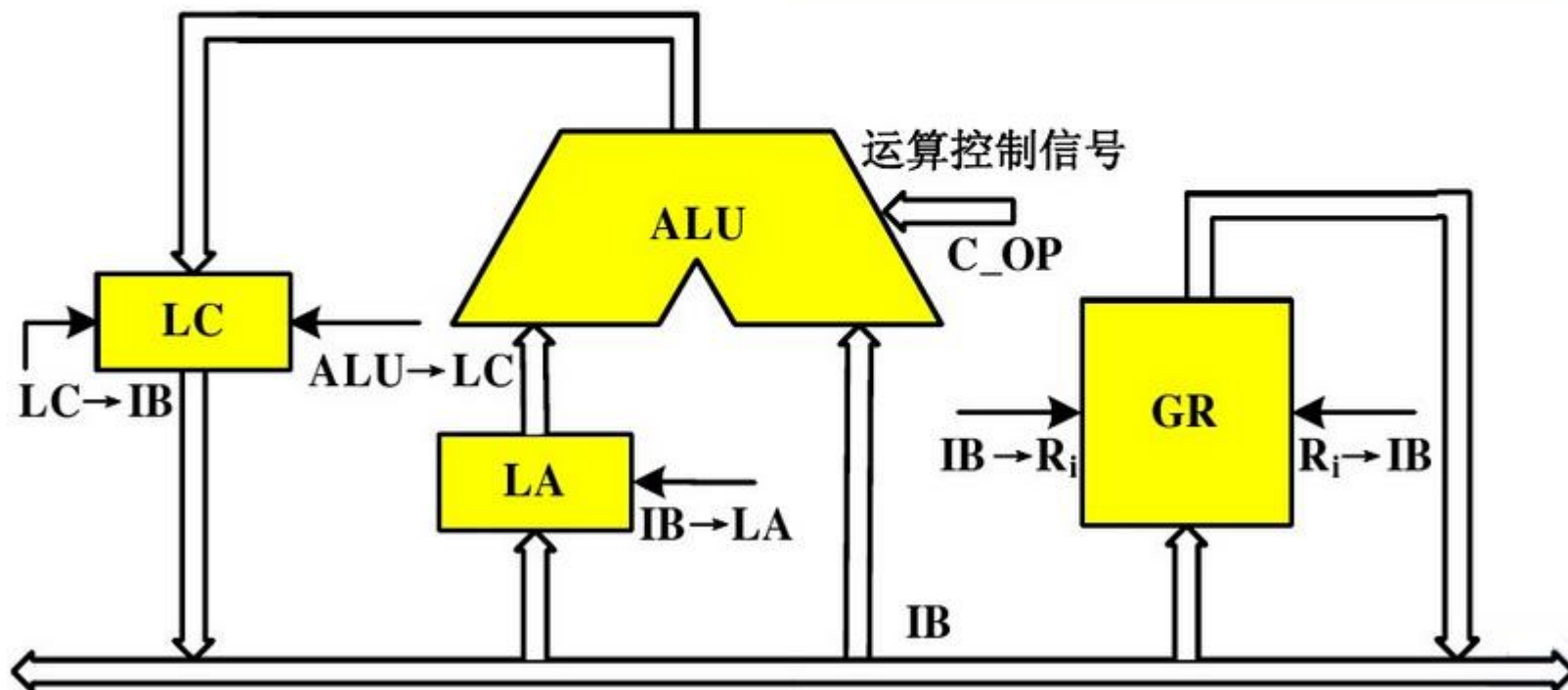
- **总线总是分时工作的**，即在任何时刻只允许传送一个部件的信号。也就是说任何时候只允许一个器件将其信号输出加到总线上。多于一个器件必然引起总线竞争。
- **同一个功能部件一次只能做一件事**。如ALU可以完成加、减、与、或等多种功能。但某一时刻只能完成一种功能，做加法时不可能同时做与运算。
- 在双总线及三总线结构的运算器中需要**多端口器件**。

单总线结构运算器(1)

- 单总线运算器的结构形式2：ALU+2个暂存器

❖ $(R_i) \theta (R_j) \rightarrow R_k$ ：需要3步

- $(R_i) \rightarrow LA$;
- $(R_j) \rightarrow IB$, ALU运算, 结果 $\rightarrow LC$;
- $(LC) \rightarrow R_k$;

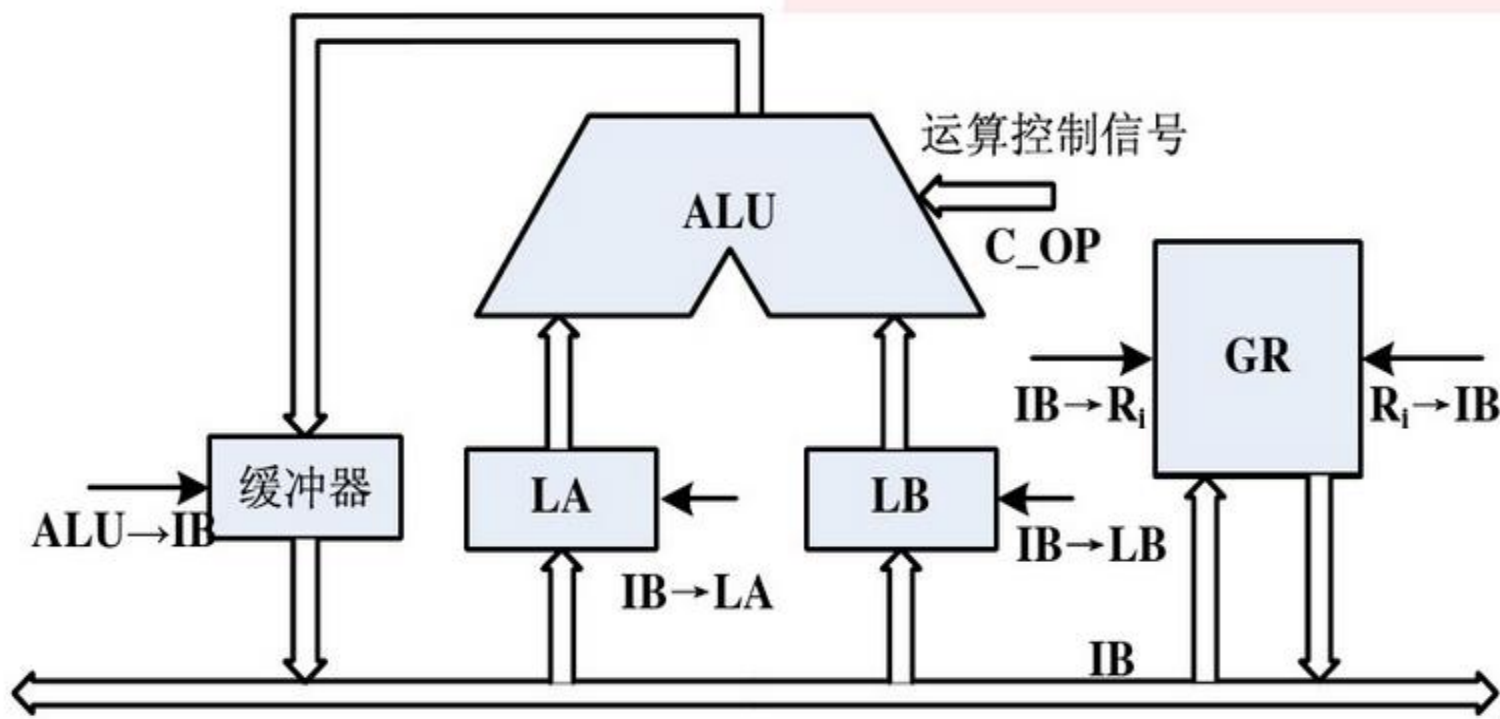


单总线结构运算器(2)

- 单总线运算器的结构形式1: ALU+2个暂存器

❖ $(R_i) \theta (R_j) \rightarrow R_k$: 需要3步

- $(R_i) \rightarrow LA$;
- $(R_j) \rightarrow LB$;
- ALU运算, 结果 $\rightarrow R_k$

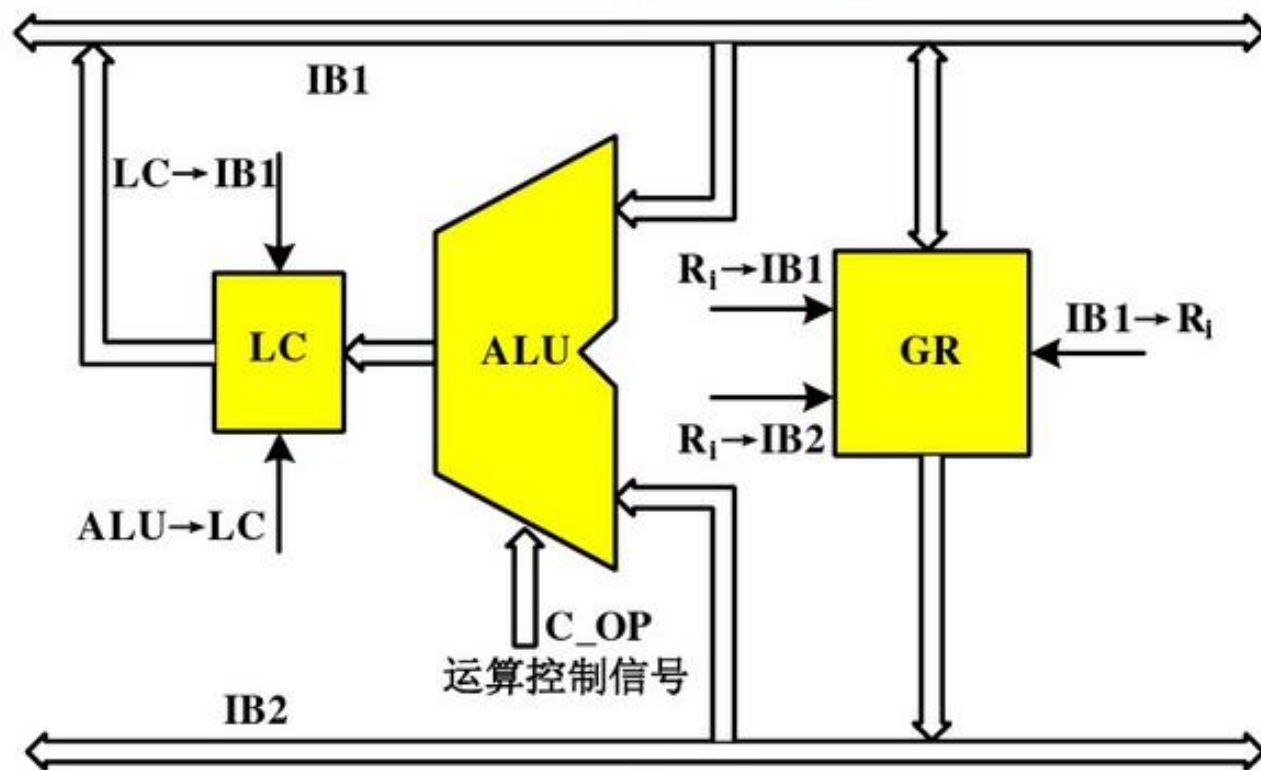


双总线结构运算器(1)

双总线运算器的结构形式1: ALU+1个暂存器

❖ $(R_i) \theta (R_j) \rightarrow R_k$: 需要2步

- $(R_i) \rightarrow IB1, (R_j) \rightarrow IB2, \text{ALU}$ 运算, 结果 $\rightarrow LC$;
- $(LC) \rightarrow R_k$;

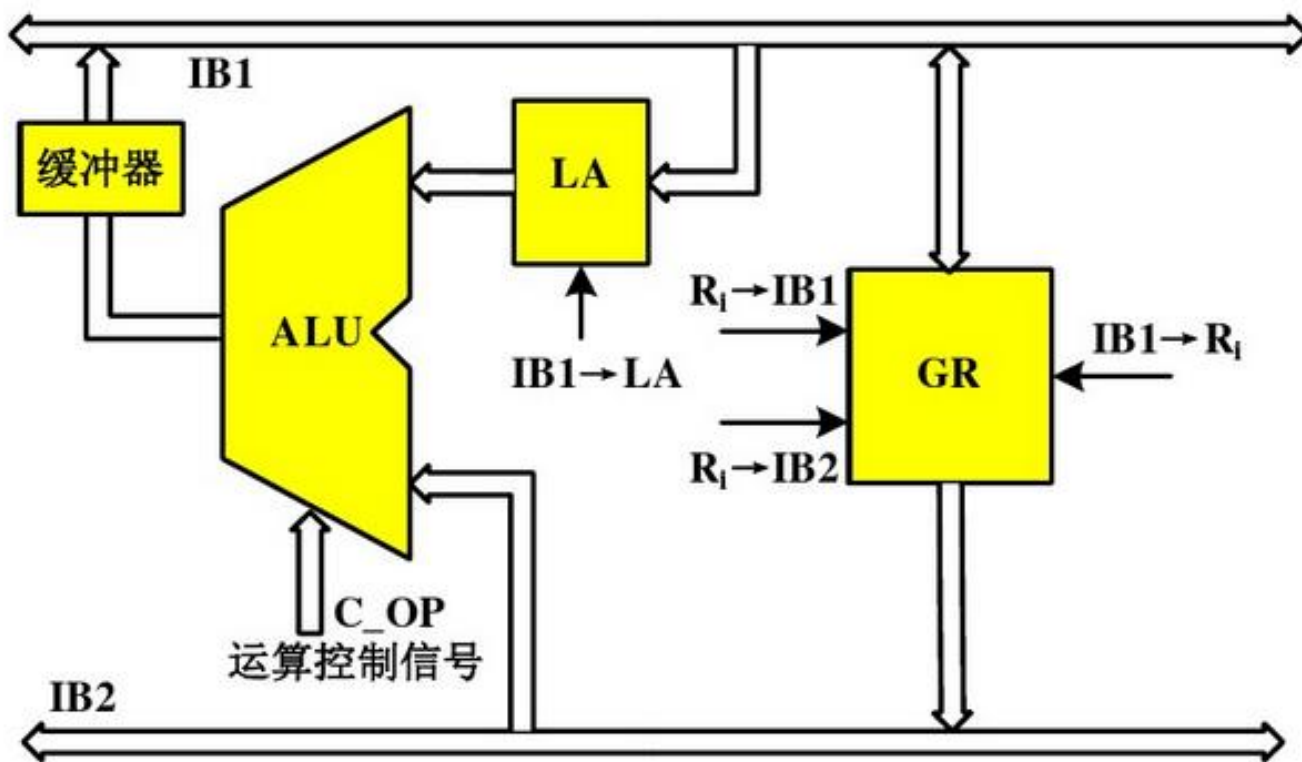


双总线结构运算器(2)

- 双总线运算器的结构形式2：ALU+1个暂存器

❖ $(R_i) \theta (R_j) \rightarrow R_k$ ：需要2步

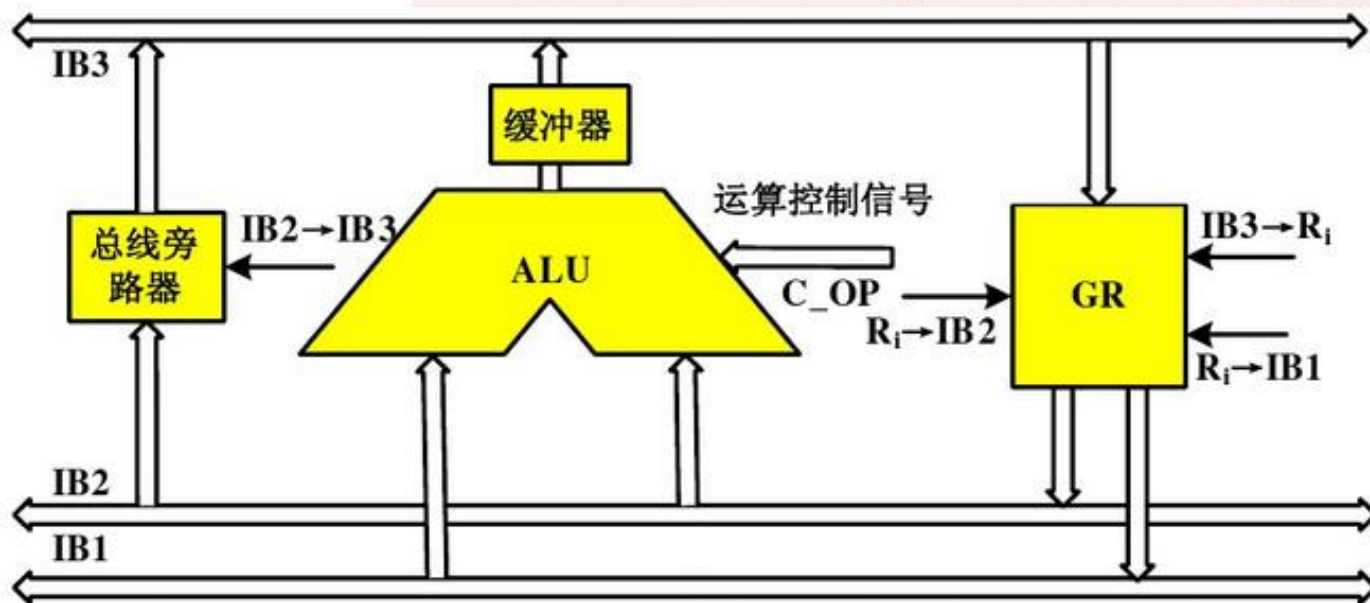
- $(R_i) \rightarrow LA$
- $(R_j) \rightarrow IB2$, ALU运算, $IB1 \rightarrow R_k$;



三总线结构运算器

❖ $(R_i) \theta (R_j) \rightarrow R_k$: 需要1步

■ $(R_i) \rightarrow IB1$, $(R_j) \rightarrow IB2$, ALU运算, $IB3 \rightarrow R_k$;



必须指出的是，在分析某一种运算器的运算过程和通路时，一个**基本的原则**就是在一个CPU周期（一步）内，**某条总线上的数据必须是唯一的**，且不能保留（至下一个CPU周期）。

3.2.3 运算器的结构

2. 标志寄存器、标志位

❖ **标志寄存器**：又称为**状态寄存器**。

- 用来**保存ALU操作结果的某些状态**。
- 不同CPU，标志寄存器中包含的标志也不尽相同。

❖ **最基本的5种运算结果标志**：

- **ZF**：结果为零标志，
 - 运算结果为全0，ZF置1
 - 运算结果不全为0，ZF置0。

$$ZF = \overline{F_n + F_{n-1} + \dots + F_0}$$

- **CF**：进位/借位标志位，**CF标志只对无符号数运算有意义**
 - 加法运算时：C=1则CF置1（表示有进位），否则置0；
 - 减法运算时：C=0则CF置1（表示不够减，有借位），否则置0。

$$CF = ADD \cdot C + SUB \cdot \overline{C}$$

2. 标志寄存器、标志位

- **OF**: 溢出标志, 反映有符号数加减运算所得结果是否溢出; OF标志只对带符号数运算有意义。

- 运算溢出: $OF=1$
- 运算没有溢出: $OF=0$

$$OF = C \oplus C'$$

- **SF**: 符号标志, 记录运算结果的符号, =运算结果的最高位。

- 在现代微机中, 有符号数采用补码表示法
- 运算结果为正数时, $SF=0$, 为负数 $SF=1$ 。

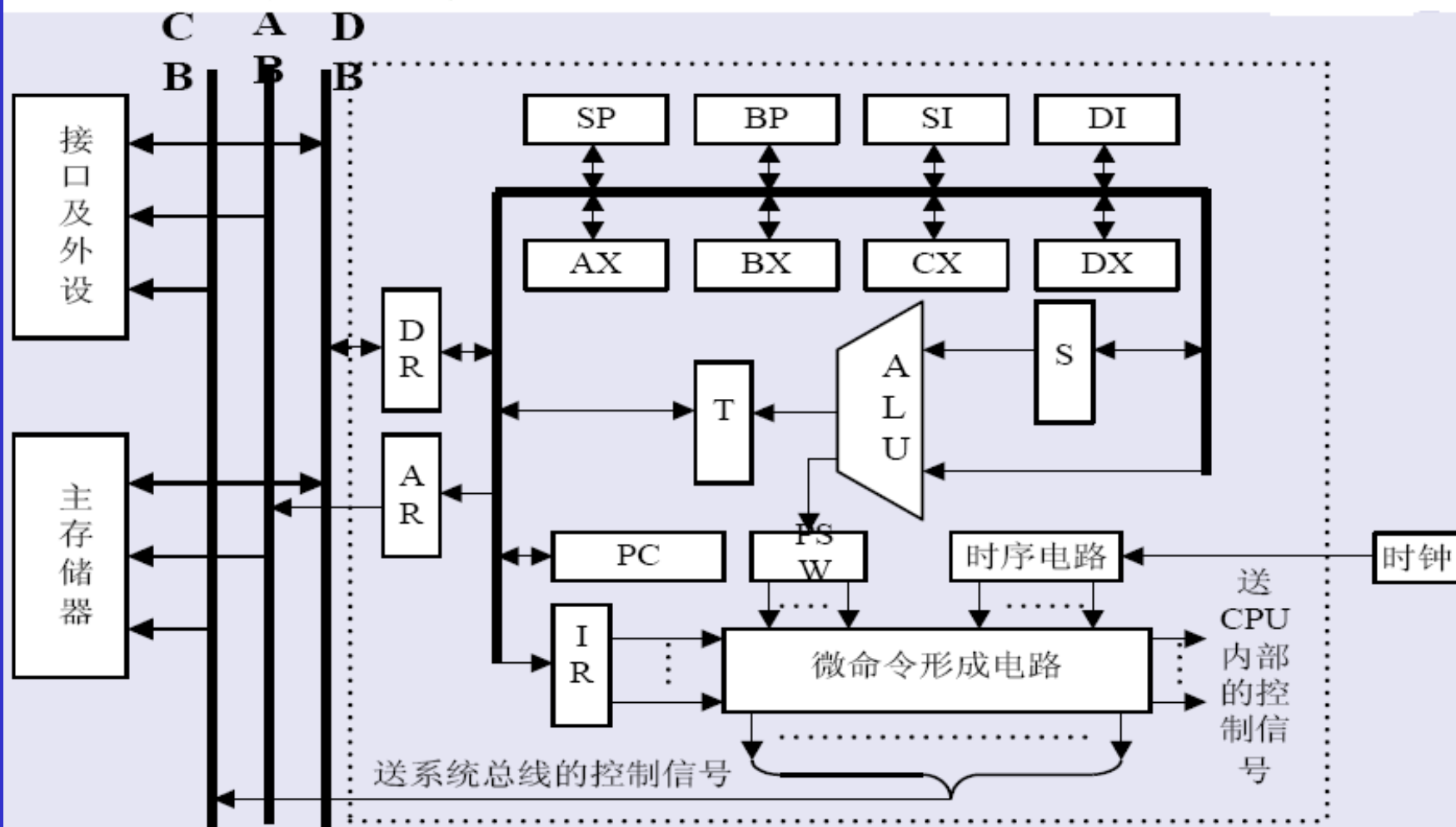
$$SF = F_n$$

- **PF**奇偶标志: 反映运算结果中“1”的个数的奇偶性
- 当结果操作数中“1”的个数为偶数: $PF=1$
- 当结果操作数中“1”的个数为奇数: $PF=0$ 。

$$PF = \overline{F_n \oplus F_{n-1} \oplus \dots \oplus F_0}$$

一种简化的计算机结构图

● 计算机简单框图



某计算机简化框图

本章作业-3

第13题

注：本次作业与下次作业一起交