

# 第10章 中央处理器



## 本章导读：

**中央处理器**（Center Processing Unit, CPU）是计算机的核心部件，它包括运算器和控制器。本章从硬件层面介绍CPU的功能和组成，给出CPU内部运算器、控制器基本原理，给出硬布线控制器及微程序控制器的基本概念，概要介绍指令流水线。

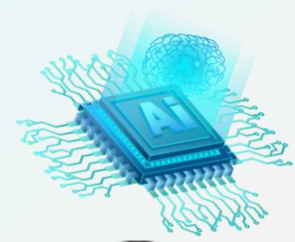


## 10.1 中央处理器概述

中央处理器是计算机硬件中的重要组成部分，它负责执行程序代码执行和数据处理。CPU主要包括控制器和运算器两大部分，还包含高速缓冲存储器以及实现CPU内部组件之间联系的数据通路、控制总线。

### 10.1.1 中央处理器的功能

**CPU具备**运算处理、程序控制、操作控制、时序控制、异常及中断处理等几个方面的功能。



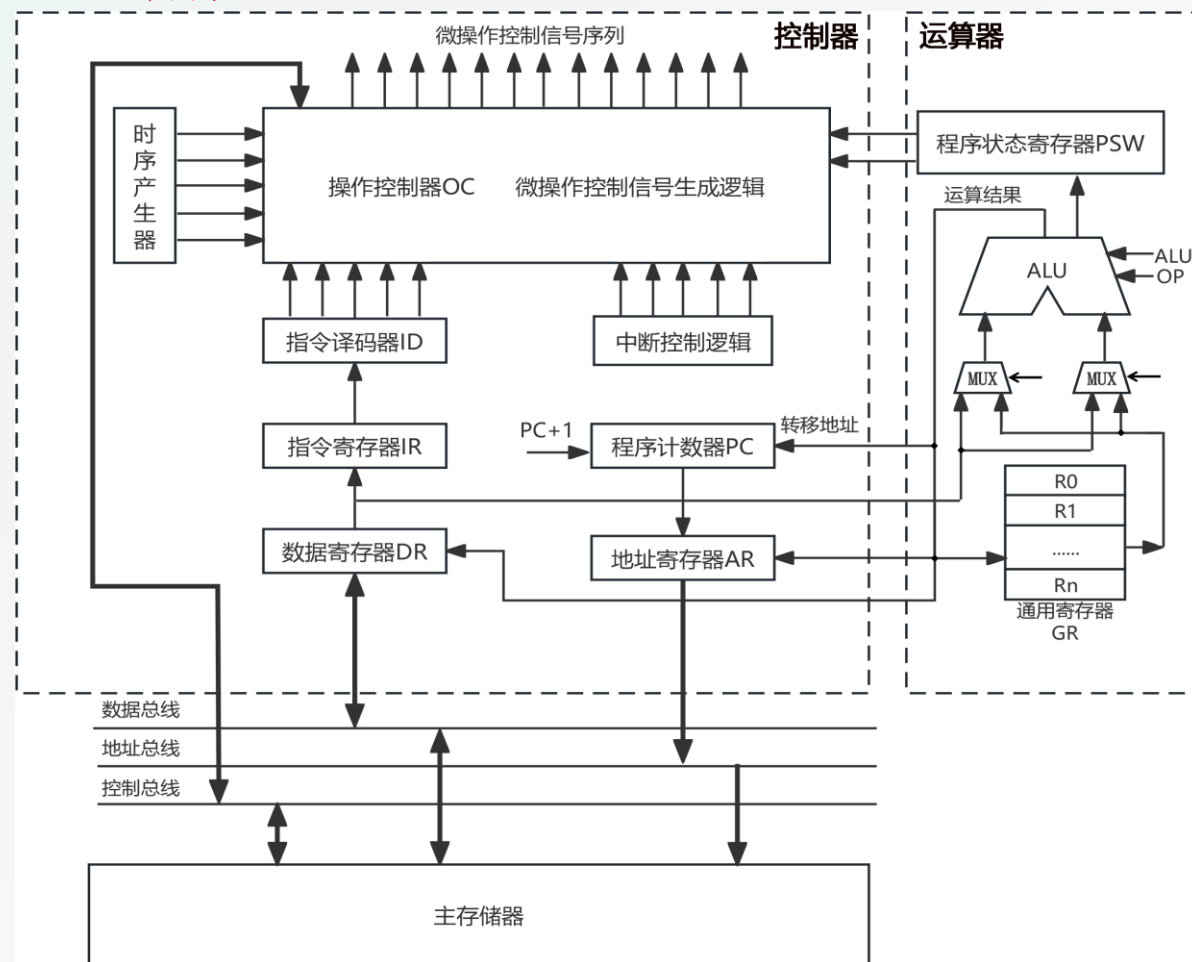
## 10.1.2 中央处理器的基本组成

(重点)

**CPU的基本组成：运算器和控制器**

**运算器：**（1）通用寄存器组（General Registers，GR）、（2）多路选择器（Multiplexer，MUX）、（3）算术逻辑单元（Arithmetic Logic Unit，ALU）、（4）程序状态字寄存器（Program Status Word，PSW）

**控制器：**（1）程序计数器（Program Counter，PC）、（2）指令寄存器（Instruction Register，IR）、（3）地址寄存器（Address Register，AR）、（4）数据寄存器（Data Register，DR）、（5）指令译码器（Instruction Decoder，ID）、（6）操作控制器（Operation Controller，OC）、（7）时序产生器。

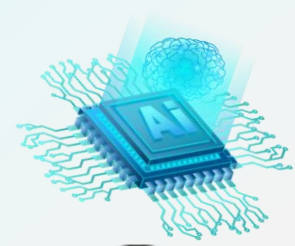




## 10.2 运算器

运算器是计算机进行算术运算和逻辑运算的主要部件，运算器的逻辑结构取决于机器的指令系统、数据表示方法和运算方法等。

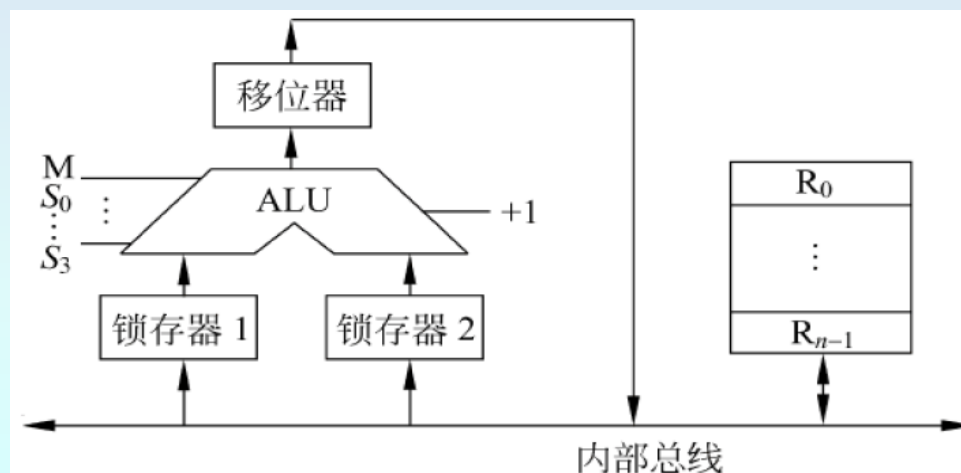
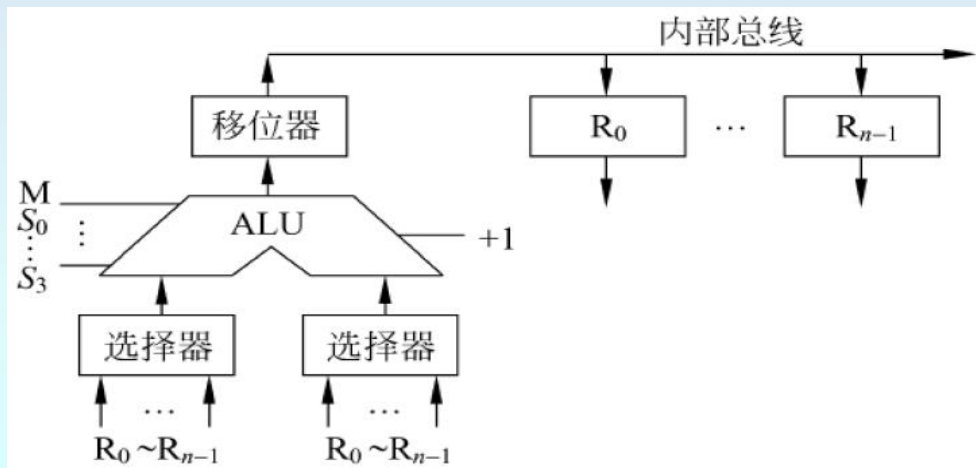
计算机中最基本的算术运算是加法运算，减法、乘法、除法运算都可以使用加法运算实现，因此**加法器是运算器的最基本的运算部件。**



## 10.2.1 运算器的基本组成（了解）

基本的运算器包含以下几个部分：实现基本算术、逻辑运算功能的ALU，提供操作数与暂存结果的寄存器组，有关的判别逻辑和控制电路等。将这些功能模块连接成一个整体时，需要解决一个问题，就是如何向ALU提供操作数。一种方法是在ALU输入端加多路选择器，另一种方法是在ALU输入端加一级锁存器（暂存器）。

运算器内的各功能模块之间的连接也采用总线结构，这个总线称为**运算器的内部总线**，ALU和各寄存器都挂在这上面，但它们仅是CPU的内部数据通路，只有数据线。



## 10.2.2 基本运算部件

### 1. 一位全加器 (重点)

一位全加器是构成加法器的基本单元。一位全加器实现两位本地二进制数以及低位的进位相加，求得本地和以及向高位的进位。

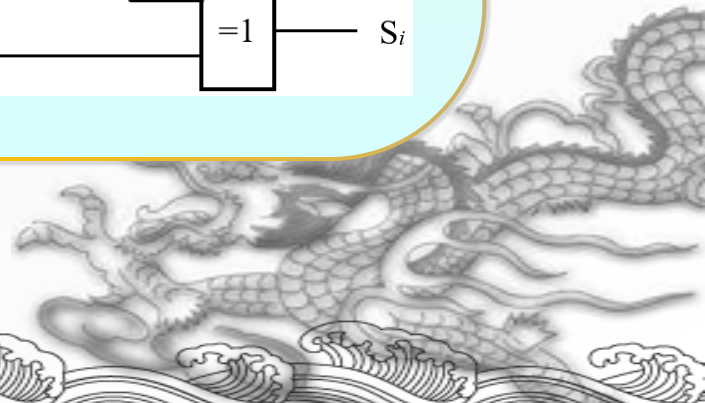
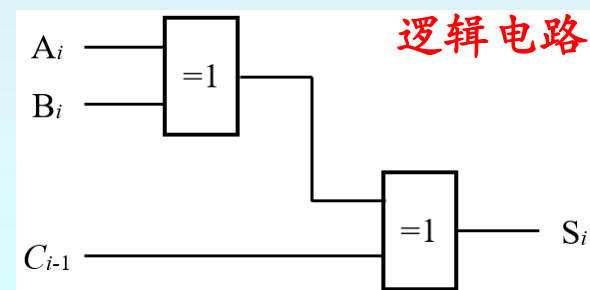
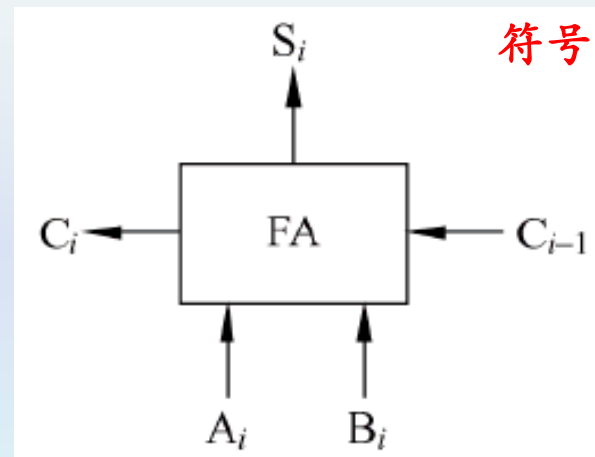
表10-1 一位全加器真值表

$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

一位全加器的逻辑表达式

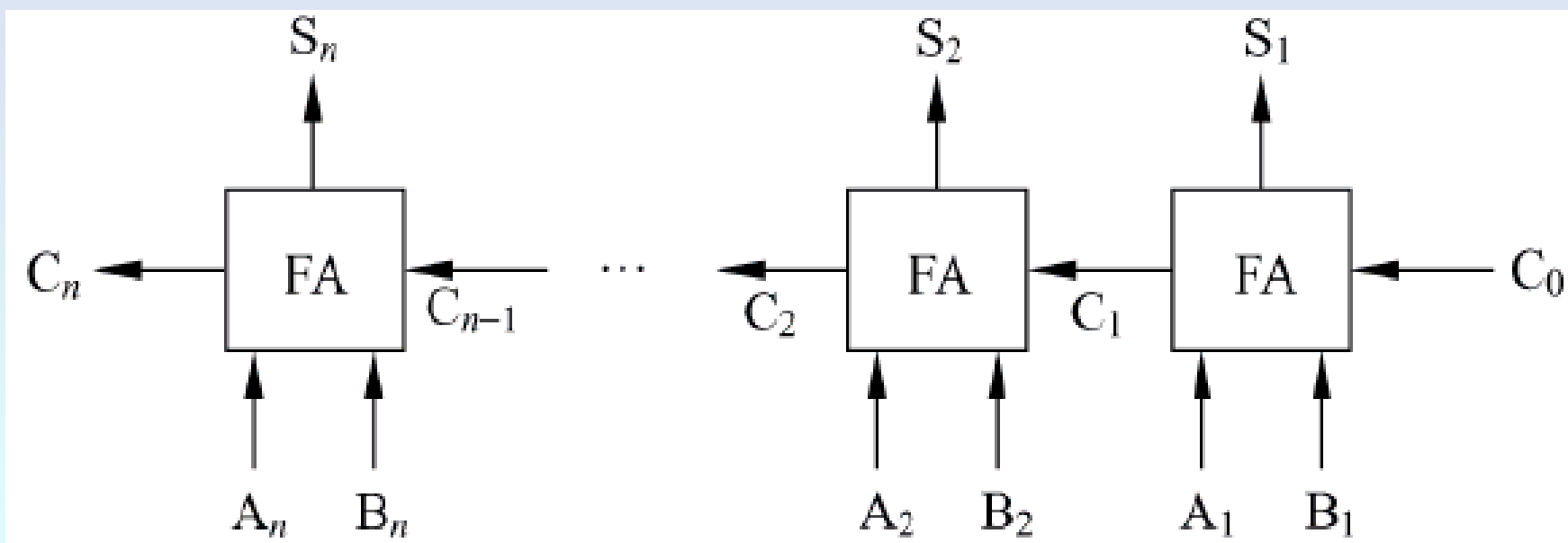
$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$



## 2. 串行进位加法器

把 $n$ 个一位全加器串接起来，就可以进行两个 $n$ 位二进制数的相加。**缺点**：速度慢

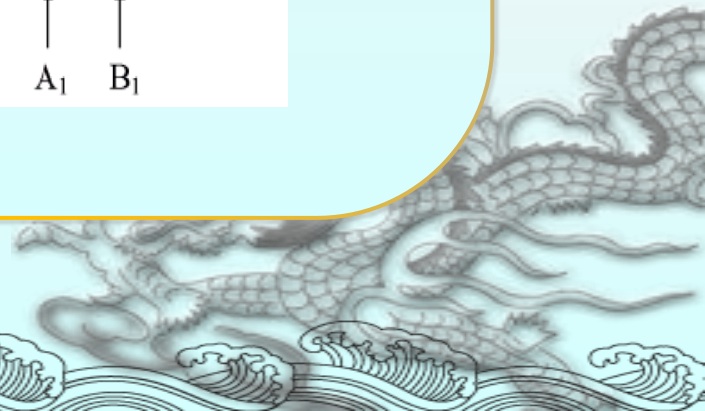
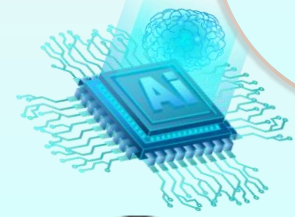
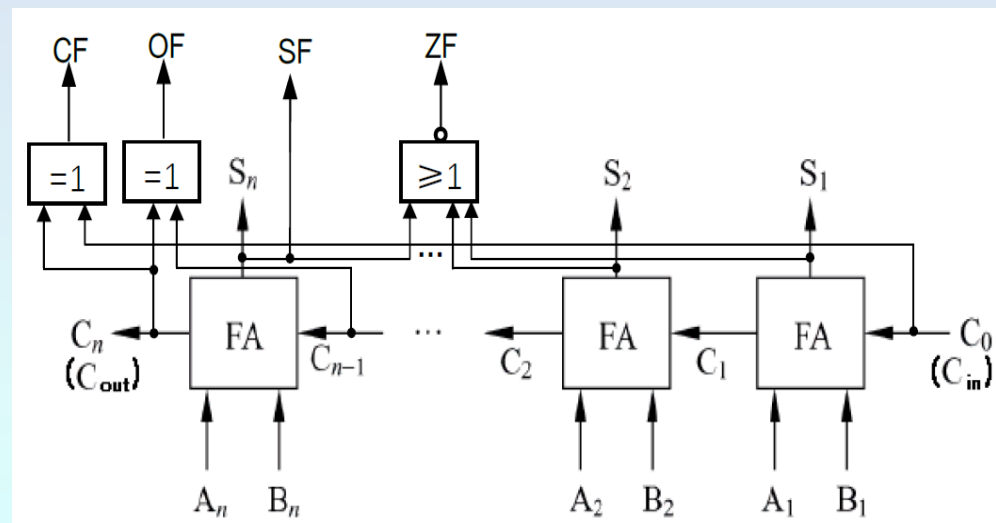
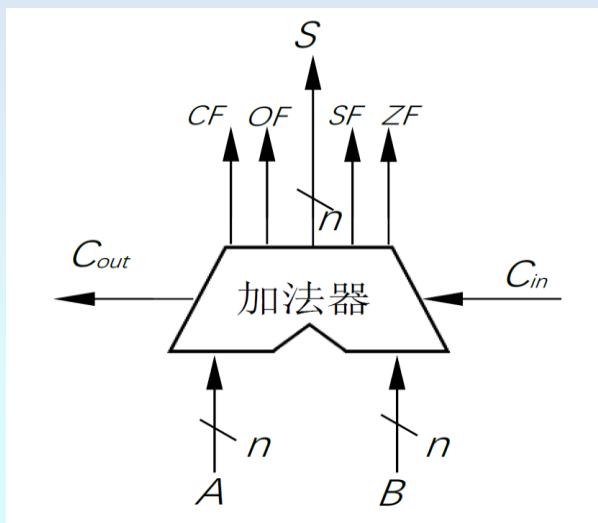




## 3. 带标志位加法器 (了解)

加法器在运算过程中需要产生运算结果，同时需要给出运算状态，即运算标志位，通常的运算标志位包含溢出标志OF、运算结果的符号标志SF，运算结果的零标志ZF以及进位/借位标志CF。4个标志的逻辑表达式为：

$$OF = C_n \oplus C_{n-1} \quad SF = S_n \quad CF = C_n \oplus C_0 \quad ZF = \overline{S_1 \oplus S_2 \oplus \dots \oplus S_n}$$



## 4. 并行进位加法器 (重点掌握)

已知一位全加器的进位 $C_i$ 的逻辑表达式为:

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$

其中,  $A_i B_i$  取决于本地参加运算的两个数, 而与低位进位无关, 因此称  $A_i B_i$  为进位产生函数 (本次进位产生), 用  $G_i$  表示, 其含义是: 若本位的两个输入均为1, 必然要向高位产生进位。  $(A_i \oplus B_i) C_{i-1}$  不但与本位的两个数有关, 还依赖于低位送来的进位, 因此称  $A_i \oplus B_i$  为进位传递函数 (低位进位传递), 用  $P_i$  表示, 其含义是: 当两个输入中有一个为1, 低位传来的进位  $C_{i-1}$ , 将向更高位传送, 所以进位表达式又可以写成:

$$C_i = G_i + P_i C_{i-1}$$

因此可以实现各级进位信号同时形成。

$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 C_1 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

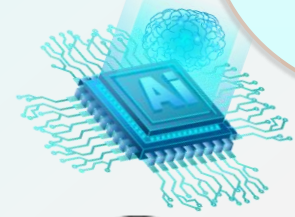
$$C_3 = G_3 + P_3 C_2 = G_3 + P_3 G_2 + P_3 P_2 C_1 + P_3 P_2 P_1 C_0$$

$$C_4 = G_4 + P_4 C_3 = G_4 + P_4 G_3 + P_4 P_3 C_2 + P_4 P_3 P_2 C_1 + P_4 P_3 P_2 P_1 C_0$$

.....

上述各式中所有进位输出仅由  $G_i$ 、 $P_i$  及最低进位输入  $C_0$  决定, 而不依赖于其低位的进位输入  $C_{i-1}$ , 因此各级进位输出可以同时产生。

**速度快, 但电路复杂, 可采用分组并行进位方式**



## 5. 算术运算单元 (了解)

算术逻辑运算单元 (ALU) 是一种能进行多种算术运算与逻辑运算的组合逻辑电路，其核心部件是带标志加法器，多采用先行进位方式实现。

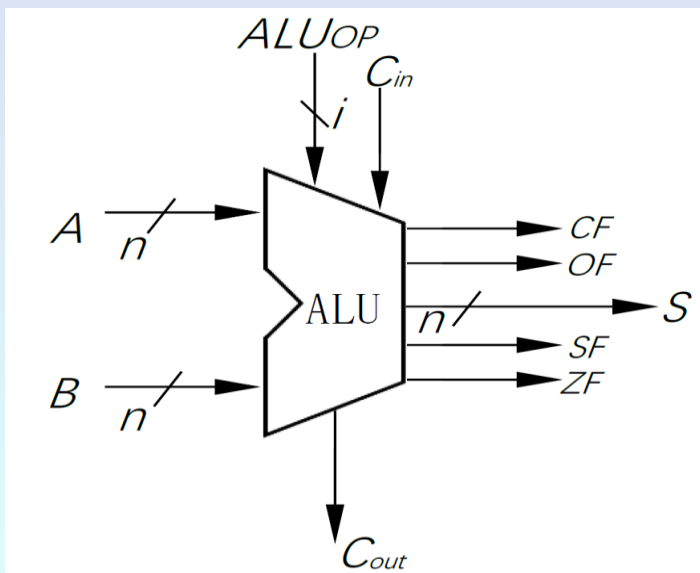


图10-10 ALU逻辑符号

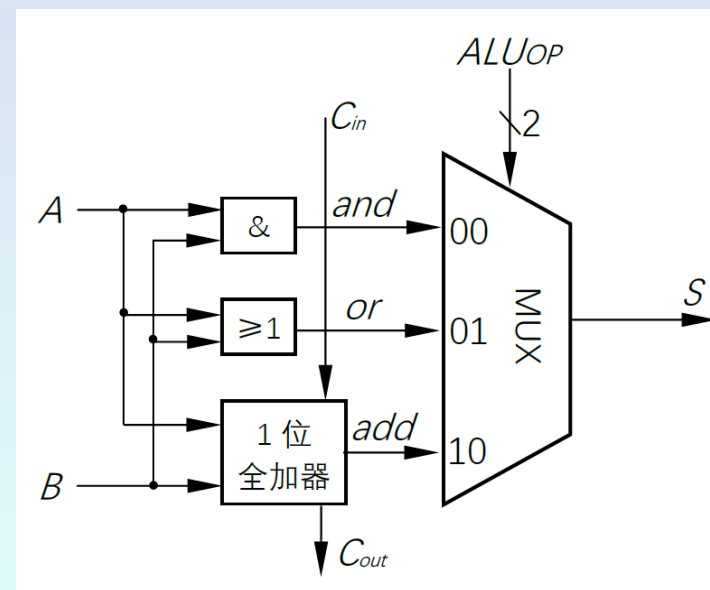
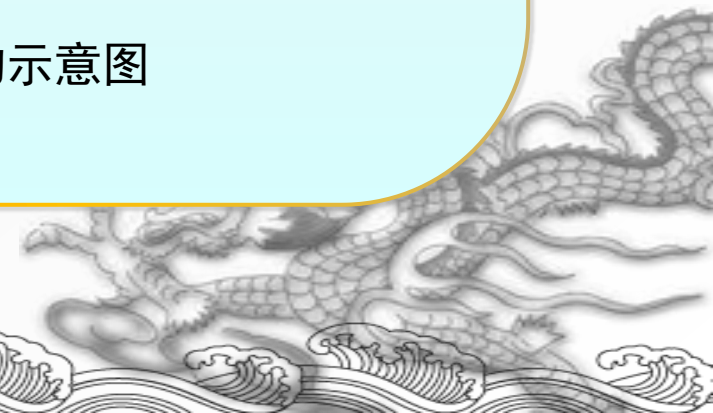


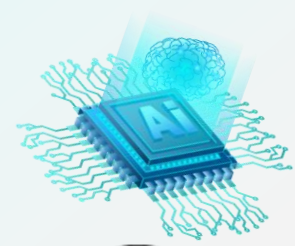
图10-11 1位ALU结构示意图



10.2.3 定点数补码加减运算及实现 (略)

10.2.4 机器数的移位运算 (略)

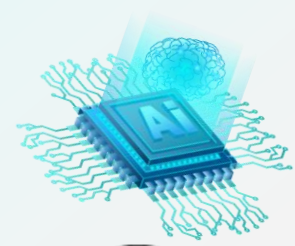
10.2.5 定点数乘法运算及实现 (略)



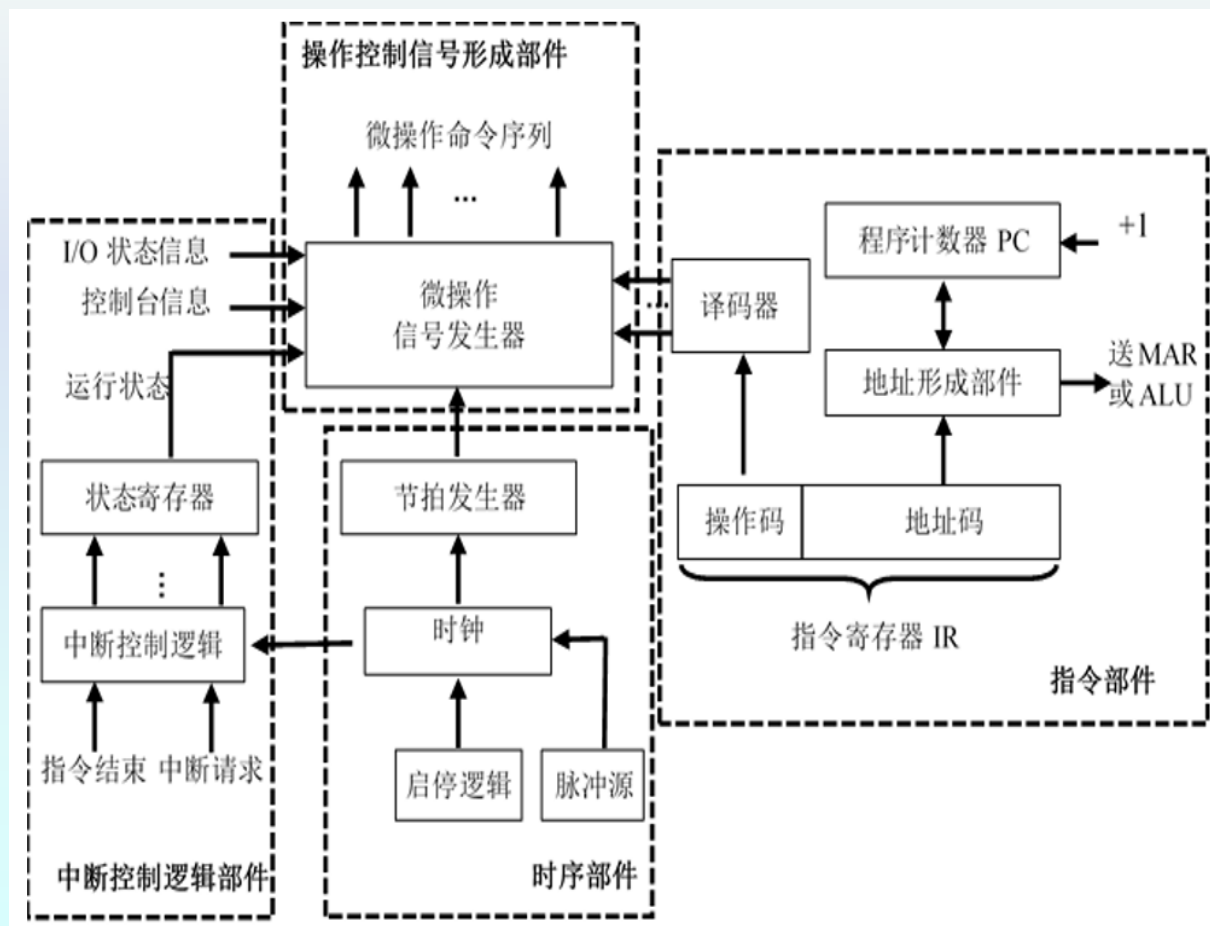


## 10.3 控制器的组成和实现方法

控制器是计算机系统的指挥中心，它把运算器、存储器、输入输出设备等部件连接为一个有机的整体，同时根据指令的要求指挥整个计算机的工作。



## 10.3.1 控制器的基本组成

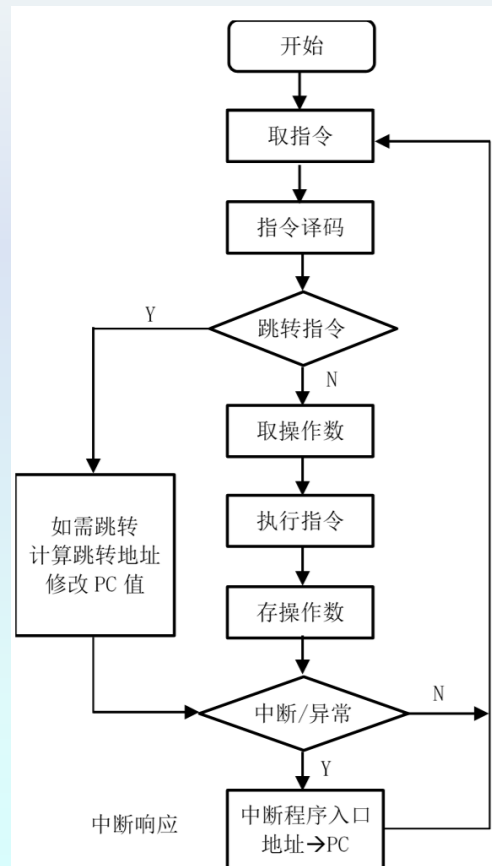


## 10.3.2 指令执行过程

### 1. 指令执行过程描述（重点）

计算机工作的过程就是运行程序的过程，控制器依据事先编制好并存放在主存中的程序控制各功能部件协调工作。计算机运行程序的过程，实质上就是由控制器根据程序对应的机器指令序列逐条执行指令的过程。CPU上电复位后即开始不断地进行取指令、执行指令的“死循环”控制器执行指令的一般流程如图所示。

指令执行时，首先是以PC为地址访问主存取指令，同时更新PC的值作为后续顺序指令的地址。通常指令译码会根据指令功能进入不同的执行路径，这里仅仅区分了分支指令和其他指令如果是分支指令且满足分支条件，则只需重新修改PC的值作为分支目标地址即可。对于其他指令可能要经历取操作数、执行指令、存操作数的基本过程。指令执行完毕后还需要进行中断异常判断，如果不存在中断异常，直接进入取指令、执行指令的循环。如果控制器检测到中断异常处理器就会进入中断响应阶段，此时需要关中断、保存断点、修改PC值作为中断服务程序入口地址，然后转去执行中断服务例程。



## 2. 指令周期（重点）

一条指令从取出到执行完成所需要的时间称为指令周期，包括取指令、分析指令、执行指令三个过程，可能还有存数周期。

### 1) 取指周期

取指周期中，CPU根据PC的内容为地址从主存取出指令，并计算后续指令的地址。

### 2) 译码/取操作数周期

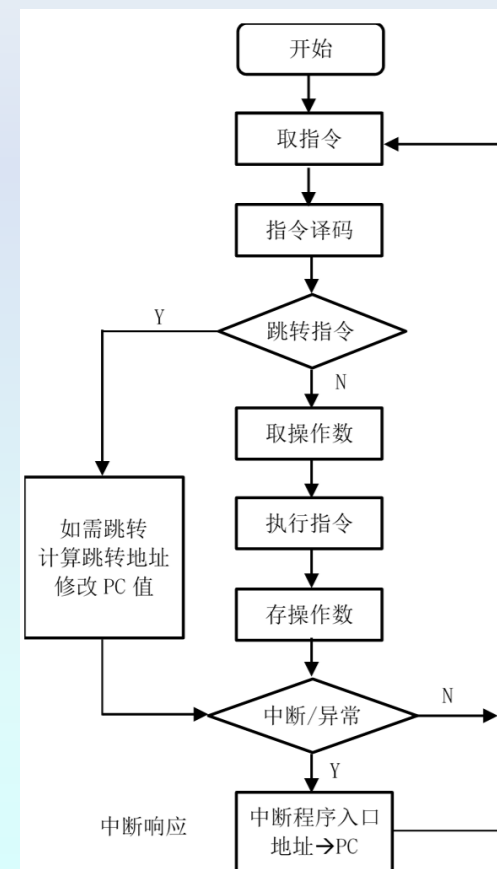
译码/取操作数周期中，对指令寄存器中的指令字进行指令译码，识别指令类型。

### 3) 执行周期

控制器向运算器ALU及数据通路中的其他相关部件发送操作控制命令，对已取出的操作进行加工处理，并将处理的状态信息记录到程序状态寄存器PSW中。

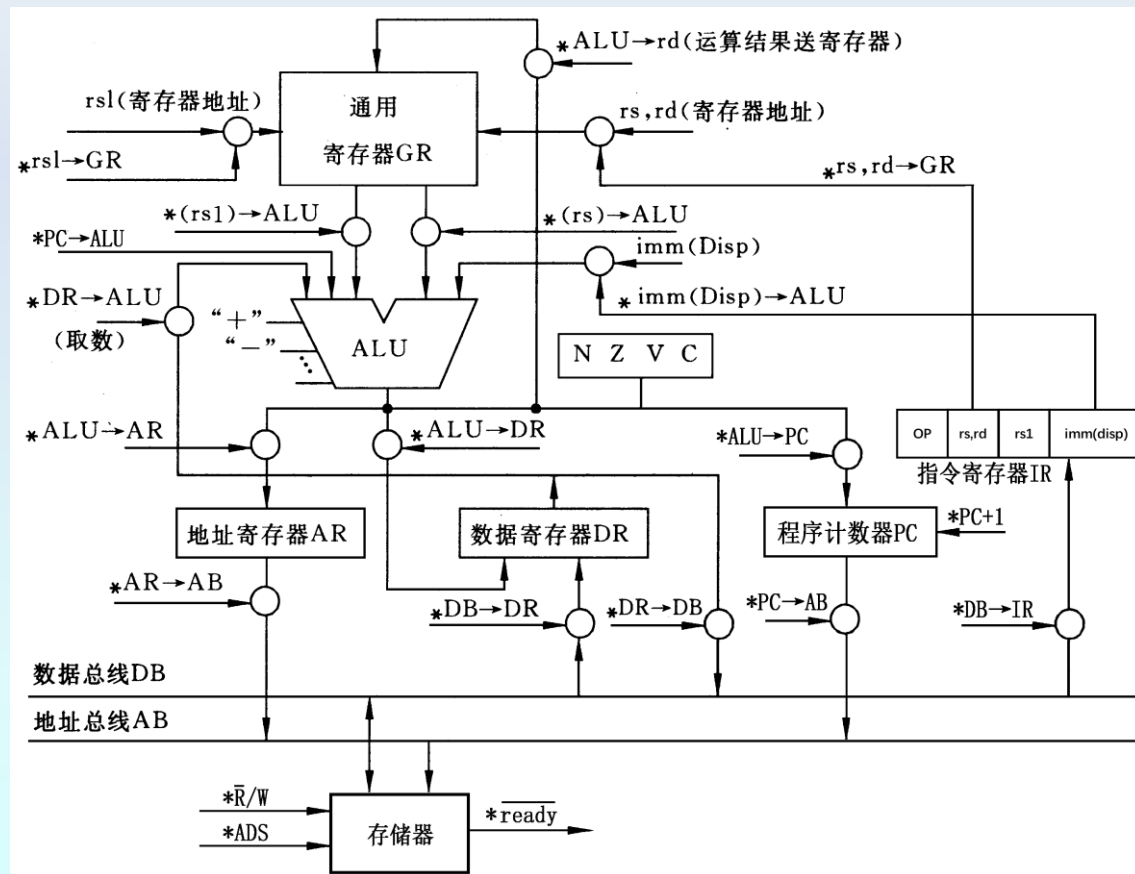
### 4) 存数周期

存数周期将运算结果写回到目的寄存器或存储器中，将运算结果写回存储器时，该过程间较长，可能需要多个时钟周期。



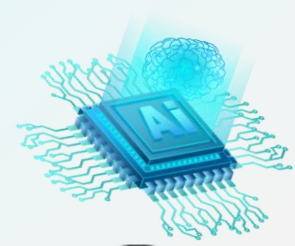


一条指令在译码后，控制器在不同机器周期产生不同的控制信号，完成该周期所需要的完成的功能。



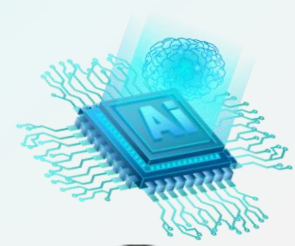
## 10.4 硬布线控制器（了解）

硬布线控制器又称为组合逻辑控制器，这种控制器的电路直接由各种类型的逻辑门电路和触发器等构成，其内部结构相对复杂但速度较快，目前RISC指令集处理器普遍采用硬布线控制器提升性能。



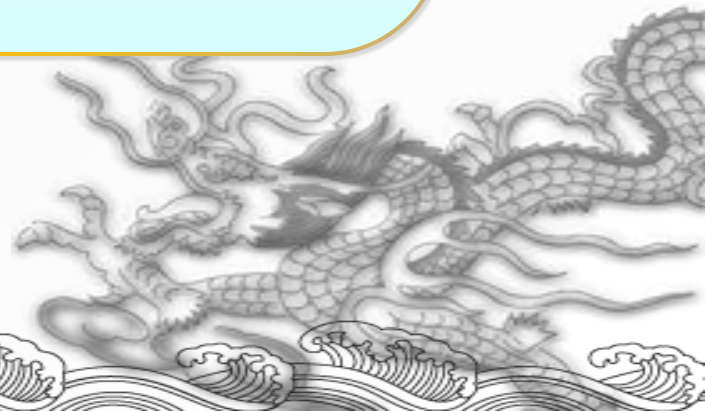
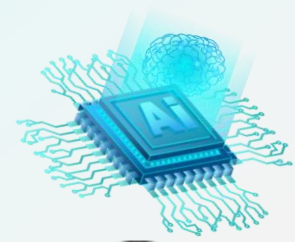
## 10.5 微程序控制器（了解）

前节描述的硬布线控制器，其控制信号的生成速度较快，适合于RISC这种简单、规整的指令系统。不过，由于硬连线控制器是一个多输入、多输出的巨大逻辑网络，对于复杂指令系统来说，对应的硬连线控制器结构庞杂，实现困难，维护不易，扩充和修改指令相当困难。因此，对于复杂指令系统，大多采用微程序方式来设计控制器。



## 10.6 指令流水线工作原理（了解）

CPU上电后就循环往复地进行取指令、执行指令，在执行完一条指令后才取出下一条指令执行，这是一种串行工作方式。显然，这种串行方式没有充分利用执行部件，在指令的执行过程，部件在某些周期内在进行操作，而在某些周期内是空闲的。如果控制器调度恰当，让各个部件持续工作，就可提高各个部件的工作效率，进而提高计算机运行速度。与现代工厂流水线生产一样，指令的执行也可以采用流水线方式，将指令的执行划分成若干个独立的阶段，多条指令的执行相互重叠起来，多条指令就可以并行执行，以提高CPU执行指令的效率。







*Thank you*

