



微机系统与接口

——CPU的工作原理

王江峰 副研究员
电气工程学院



1 指令系统

● 指令：

- 计算机的工作过程就是计算机执行**程序**的过程
- **程序是一个指令序列**，这个序列明确告诉计算机应该执行什么操作，在什么地方能够找到用来操作的数据
- 一旦把程序装入主存储器，计算机就可以自动执行取出指令和执行指令的任务
- 用来完成此项工作的计算机部件称为**中央处理器（CPU）**
- **指令**是计算机硬件能够识别并直接执行操作的命令
——加、减、乘、除、开方、函数、传送



三地址指令	OP	D1	D2	D3
二地址指令	OP	目的	源	
一地址指令	OP	D	单操作数指令和隐含操作数指令	
零地址指令	OP	无操作数指令和隐含操作数指令		



1 指令系统

● 指令：

一般的计算机运算指令要包含以下这些内容：

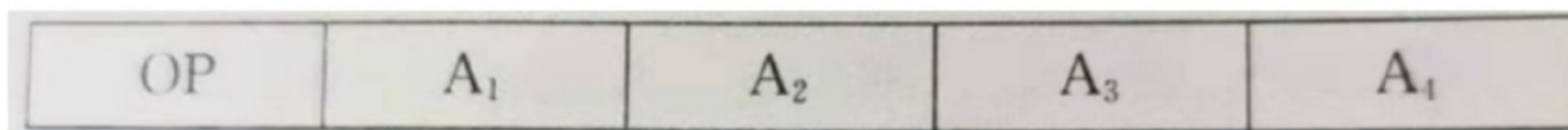
第一个操作数或者其地址 A_1 ；

第二个操作数或者其地址 A_2 ；

需要对操作数进行的操作 OP ；

计算结果的存放地址 A_3 ；

下一条指令的地址 A_4 。



$$(A_1)OP(A_2) \rightarrow A_3$$

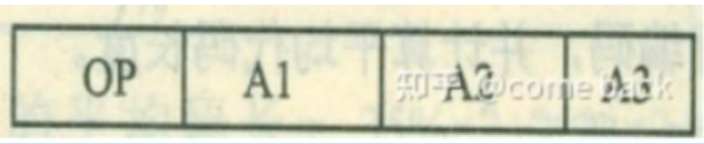
A_4 = 下一条指令的地址



1 指令系统

● 指令：

三地址指令



$$(A_1)OP(A_2) \rightarrow A_3$$

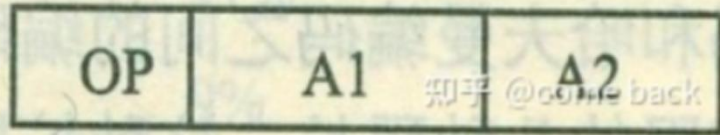
$$(PC) + 1 \rightarrow PC \quad (\text{隐含})$$



1 指令系统

● 指令：

二地址指令



$$(A_1)OP(A_2) \rightarrow A_1$$

$$(PC) + 1 \rightarrow PC \quad (\text{隐含})$$



1 指令系统

● 指令：

一地址指令



$$(Acc)OP(A_1) \rightarrow Acc$$

$$(PC) + 1 \rightarrow PC \quad (\text{隐含})$$

$$OP(A_1) \rightarrow A_1$$

$$(PC) + 1 \rightarrow PC \quad (\text{隐含})$$



1 指令系统

● 指令：

零地址指令

OP

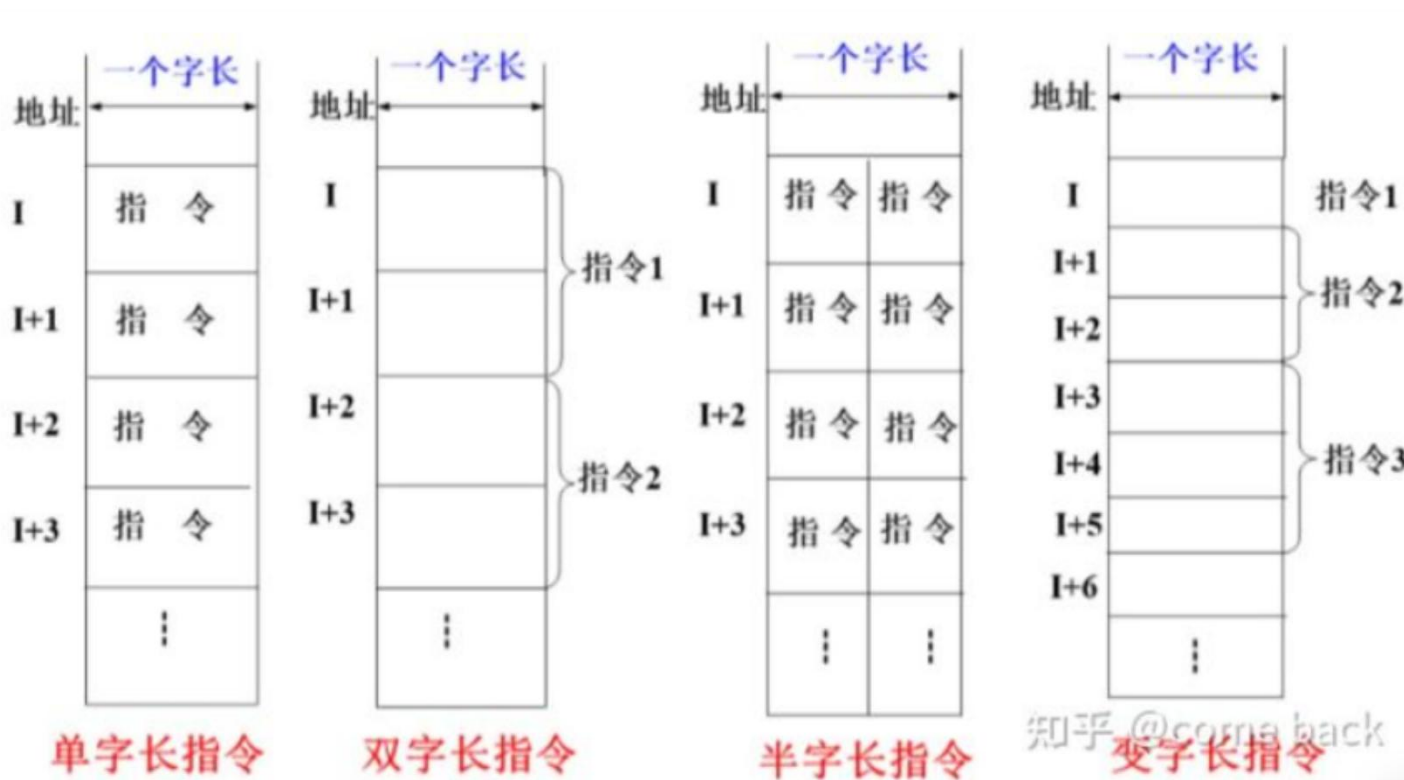
PUSH: 压栈

POP: 出栈



1 指令系统

● 指令：





1 指令系统

● 指令：

	OP	A ₁	A ₂	A ₃	
4 位操作码	0000 0001 ⋮ 1110	A ₁ A ₁ ⋮ A ₁	A ₂ A ₂ ⋮ A ₂	A ₃ A ₃ ⋮ A ₃	15 条 3 地址指令
8 位操作码	1111 1111 ⋮ 1111	0000 0001 ⋮ 1110	A ₂ A ₂ ⋮ A ₂	A ₃ A ₃ ⋮ A ₃	15 条 2 地址指令
12 位操作码	1111 1111 ⋮ 1111	1111 1111 ⋮ 1111	0000 0001 ⋮ 1110	A ₃ A ₃ ⋮ A ₃	15 条 1 地址指令
16 位操作码	1111 1111 ⋮ 1111	1111 1111 ⋮ 1111	1111 1111 ⋮ 1111	0000 0001 ⋮ 1111	16 条 0 地址指令

知乎 @come back



1 指令系统

● 指令系统：

■ 一台计算机中所有指令的集合构成了该计算机的**指令系统**，也称为**指令集**——**复杂指令系统计算机**(Complex Instruction Set Computer)，简称**CISC**

- 有20%的指令使用频度比较高，占据了80%的处理机时间。有80%的指令只在20%的处理机运行时间内才被用到，大量使用频率很低的复杂指令造成硬件资源的浪费
- 增加了硬件的复杂程度，会使指令的执行周期大大加大，从而有可能使整个程序的执行时间反而增加
- 研制周期变长、不易调试维护

——**精简指令系统计算机**(Reduced Instruction Set Computer)，简称**RISC**

- RISC处理器所设计的指令系统应使流水线处理能高效率执行，并使优化编译器能生成优化代码
- 简单而统一格式的指令译码，大部分指令可以单周期执行完成



1 指令系统

● 指令系统:

- 一台计算机中所有指令的集合构成了该计算机的**指令系统**，也称为**指令集**
 - 复杂指令系统计算机(Complex Instruction Set Computer)，简称CISC
 - 精简指令系统计算机(Reduced Instruction Set Computer)，简称RISC

CISC	RISC
以硬件为中心的指令集设计	以软件为中心的指令集设计
通过硬件实现各类程序指令	通过编译器实现简单指令组合，完成复杂功能
更高效地使用内存和寄存器	需要更大的内存和寄存器，并更频繁地使用
可变的指令长度，支持更复杂的指令长度	简单、定长的指令
大量指令数	少量指令数



2 CPU 的功能

● CPU控制整个程序的执行，它具有以下基本功能：

■ 程序控制

——程序控制就是控制指令的执行顺序

- 程序是指令的有序集合，这些指令的相互顺序不能任意颠倒，必须严格按程序规定的顺序执行。保证计算机按一定顺序执行程序是CPU的首要任务

■ 操作控制

——操作控制就是控制指令进行操作

- 一条指令的功能往往由若干个操作信号的组合来实现。因此，CPU管理并产生每条指令的操作信号，把各种操作信号送往相应的部件，从而控制这些部件按指令的要求进行操作

■ 时间控制

——时间控制就是对各种操作实施定时控制

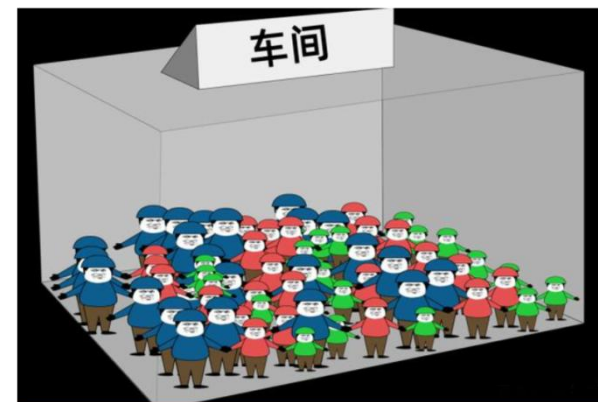
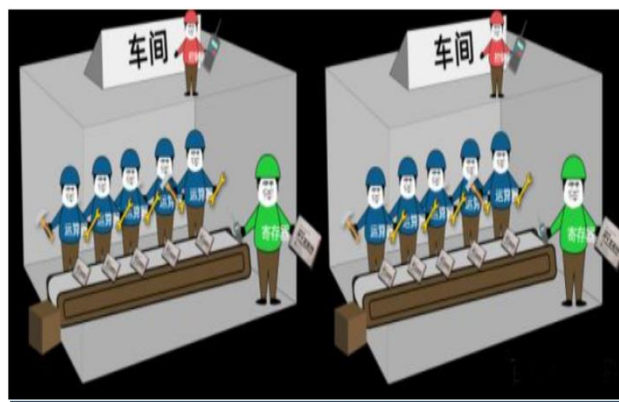
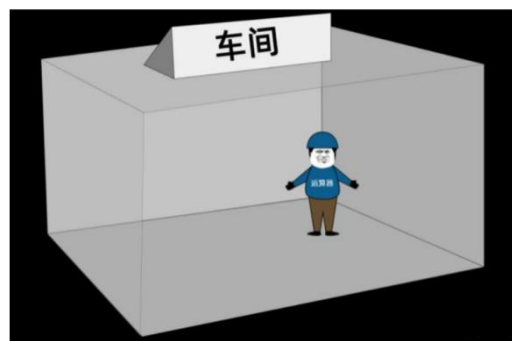
- 在计算机中，各种指令的操作信号和一条指令的整个执行过程都受到严格定时，这样计算机才能有条不紊地工作

■ 数据加工

——数据加工就是对数据进行算术、逻辑运算。完成数据的加工处理，是CPU的根本任务



2 CPU 的功能





2 CPU的组成

● CPU由控制器和运算器组成

■ 控制器

——控制指令的执行顺序控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机

——控制器通常由程序计数器（Program Counter, PC）、指令寄存器（Instruction Register, IR）、地址寄存器（Address Register, AR）、指令译码器（Instruction Decoder, ID）、时序发生器和操作控制器组成

——功能包括：

- 从主存中取出一条指令，并指出下一条指令在主存中的位置；
- 对指令进行译码，并产生相应的操作控制信号，以便启动规定的动作；
- 指挥并控制CPU、主存和输入/输出设备之间数据流动的方向



2 CPU的组成

● CPU由控制器和运算器组成

■ 运算器

——运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)

——运算器接受控制器的命令而进行动作，即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，所以它是执行部件

——运算器由算术逻辑单元 (Arithmetic Logic Unit, ALU)、累加寄存器 (Accumulator, AC)、数据寄存器 (Data Register, DR) 和程序状态字寄存器 (Program Status Word, PSW) 组成

——功能包括：

- 执行所有的算术运算；
- 执行所有的逻辑运算，并进行逻辑测试



2 CPU的组成

● CPU由控制器和运算器组成

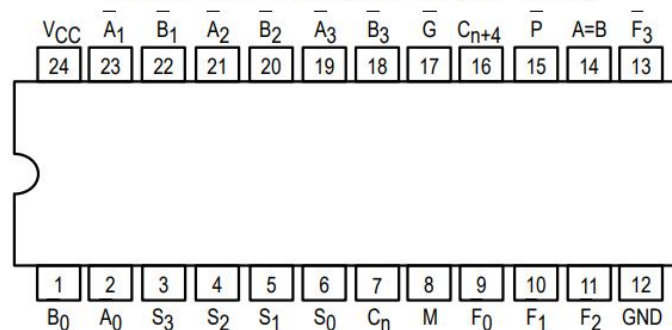
■ 运算器——ALU

4-BIT ARITHMETIC LOGIC UNIT

The SN54/74LS181 is a 4-bit Arithmetic Logic Unit (ALU) which can perform all the possible 16 logic, operations on two variables and a variety of arithmetic operations.

- Provides 16 Arithmetic Operations Add, Subtract, Compare, Double, Plus Twelve Other Arithmetic Operations
- Provides all 16 Logic Operations of Two Variables Exclusive — OR, Compare, AND, NAND, OR, NOR, Plus Ten other Logic Operations
- Full Lookahead for High Speed Arithmetic Operation on Long Words
- Input Clamp Diodes

CONNECTION DIAGRAM DIP (TOP VIEW)





2 CPU的组成

● CPU由控制器和运算器组成

■ 运算器——ALU

SN54/74LS181

FUNCTION TABLE

MODE SELECT INPUTS				ACTIVE LOW INPUTS & OUTPUTS		ACTIVE HIGH INPUTS & OUTPUTS	
S ₃	S ₂	S ₁	S ₀	LOGIC (M = H)	ARITHMETIC** (M = L) (C _n = L)	LOGIC (M = H)	ARITHMETIC** (M = L) (C _n = H)
L	L	L	L	\overline{A}	A minus 1	\overline{A}	A
L	L	L	H	\overline{AB}	\overline{AB} minus 1	$\overline{A + B}$	$A + \overline{B}$
L	L	H	L	$A + B$	\overline{AB} minus 1	\overline{AB}	$A + B$
L	L	H	H	Logical 1 minus 1		Logical 0 minus 1	
L	H	L	L	$A + B$	A plus $(A + \overline{B})$	\overline{AB}	A plus \overline{AB}
L	H	L	H	\overline{B}	\overline{AB} plus $(A + B)$	\overline{B}	$(A + B)$ plus \overline{AB}
L	H	H	L	$A \oplus \overline{B}$	A minus B minus 1	$A \oplus B$	A minus B minus 1
L	H	H	H	$\overline{A + B}$	A + B	\overline{AB}	\overline{AB} minus 1
H	L	L	L	\overline{AB}	A plus $(A + B)$	$\overline{A + B}$	A plus \overline{AB}
H	L	L	H	$A \oplus B$	\overline{A} plus B	$A \oplus B$	A plus \overline{B}
H	L	H	L	\overline{B}	\overline{AB} plus $(A + B)$	\overline{B}	$(A + B)$ plus \overline{AB}
H	L	H	H	$A + B$	A + B	\overline{AB}	\overline{AB} minus 1
H	H	L	L	Logical 0 A plus A*		Logical 1 A plus A*	
H	H	L	H	\overline{AB}	\overline{AB} plus A	$A + B$	$(A + \overline{B})$ plus A
H	H	H	L	\overline{AB}	\overline{AB} plus A	$A + B$	$(A + B)$ Plus A
H	H	H	H	A	A	A	A minus 1

L = LOW Voltage Level

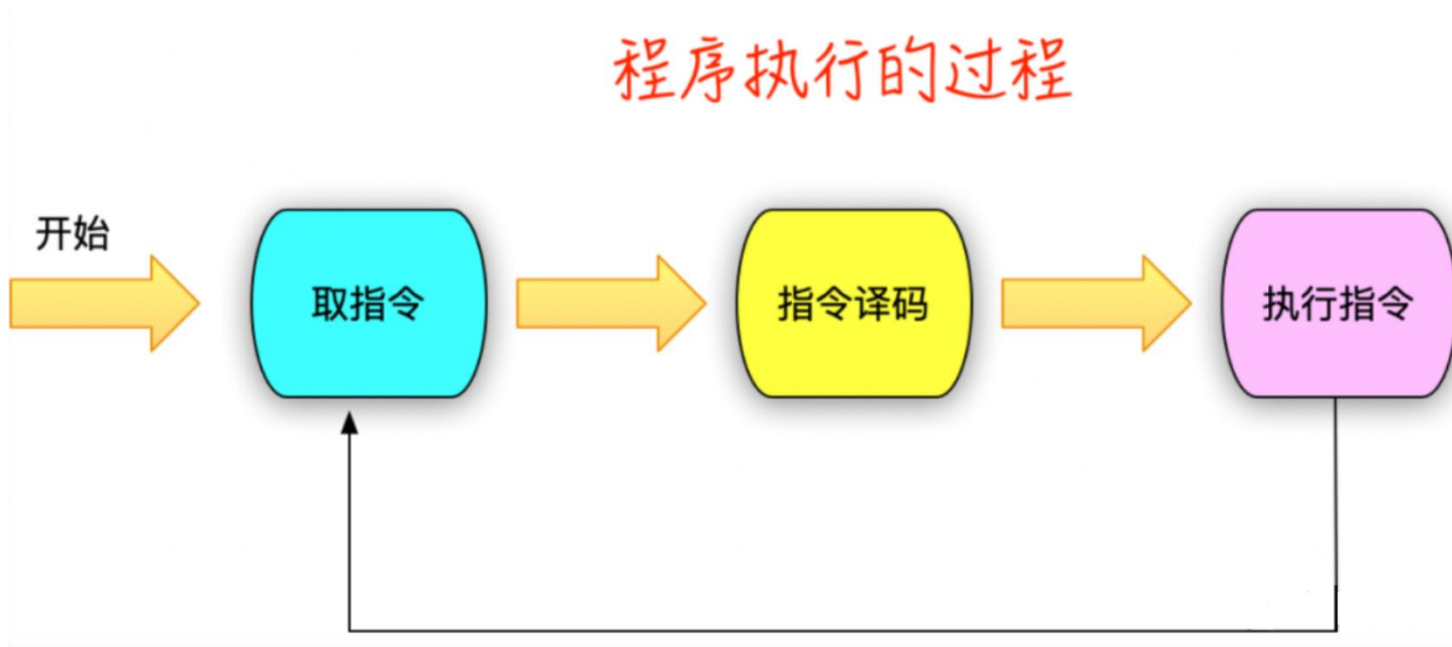
H = HIGH Voltage Level

*Each bit is shifted to the next more significant position

**Arithmetic operations expressed in 2s complement notation

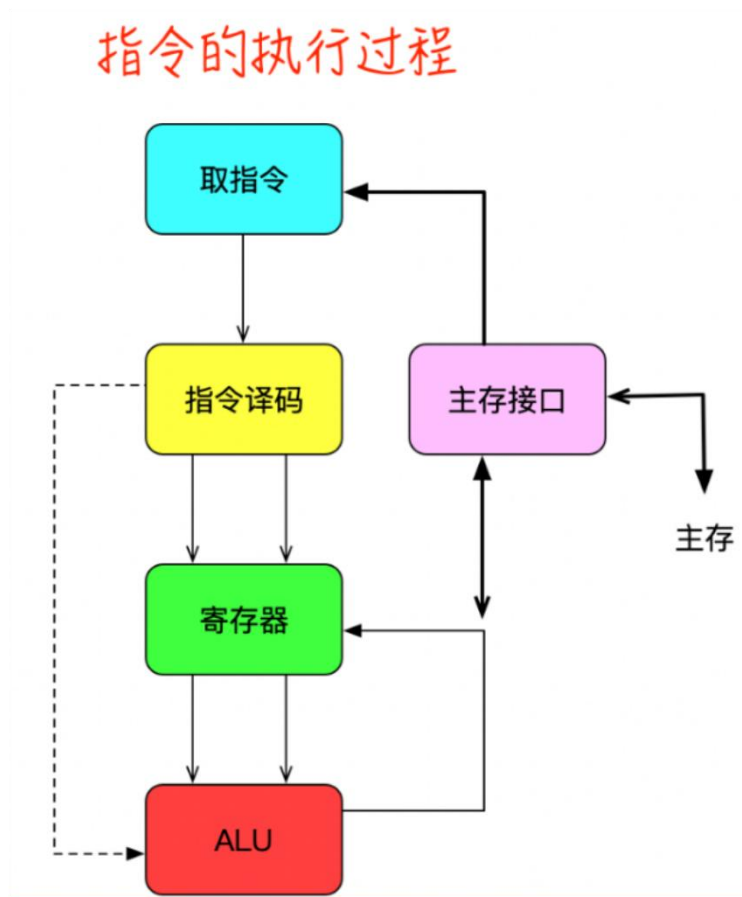


3 CPU 的工作原理



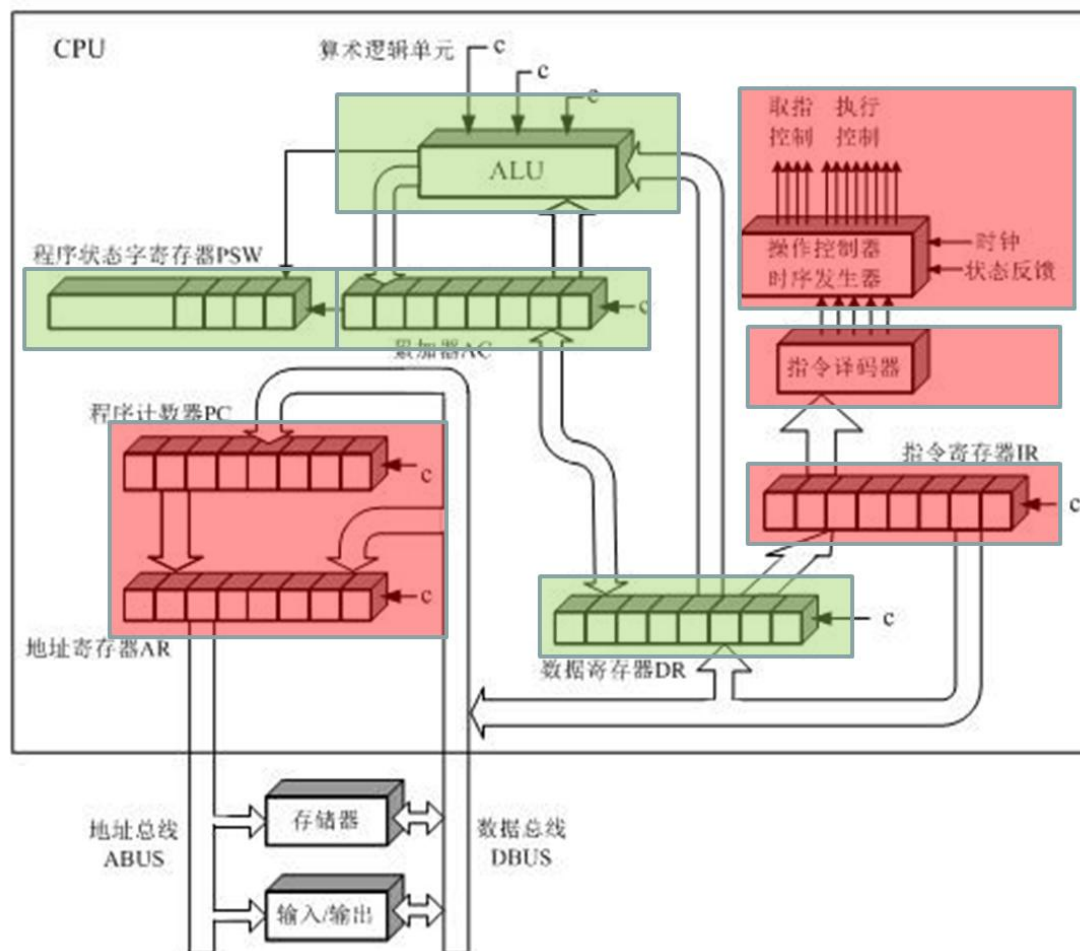


3 CPU 的工作原理





3 CPU 的工作原理





3 CPU 的工作原理

