



计算机组成与体系结构

第5章 指令系统

赵庆行、张骏鹏

(qhzhao[at]xidian[dot]edu[dot]cn)

人工智能学院

本章内容

- 指令系统设计属于计算机体系结构设计内容。
- 指令系统设计得好坏，不但关系着计算机的**程序设计**，也关系着其**硬件设计**。
- 本章讨论指令系统设计问题，包括：
 - 指令格式
 - 指令类型
 - 寻址方式
 - 数据类型
 - 存储模式
 - 寄存器等

本次课内容

- 指令及指令系统作用
- 指令系统结构层及定义（即设计）
 - 存储模式
 - 寄存器组织
 - 数据类型
 - I/O模式
 - 指令系统

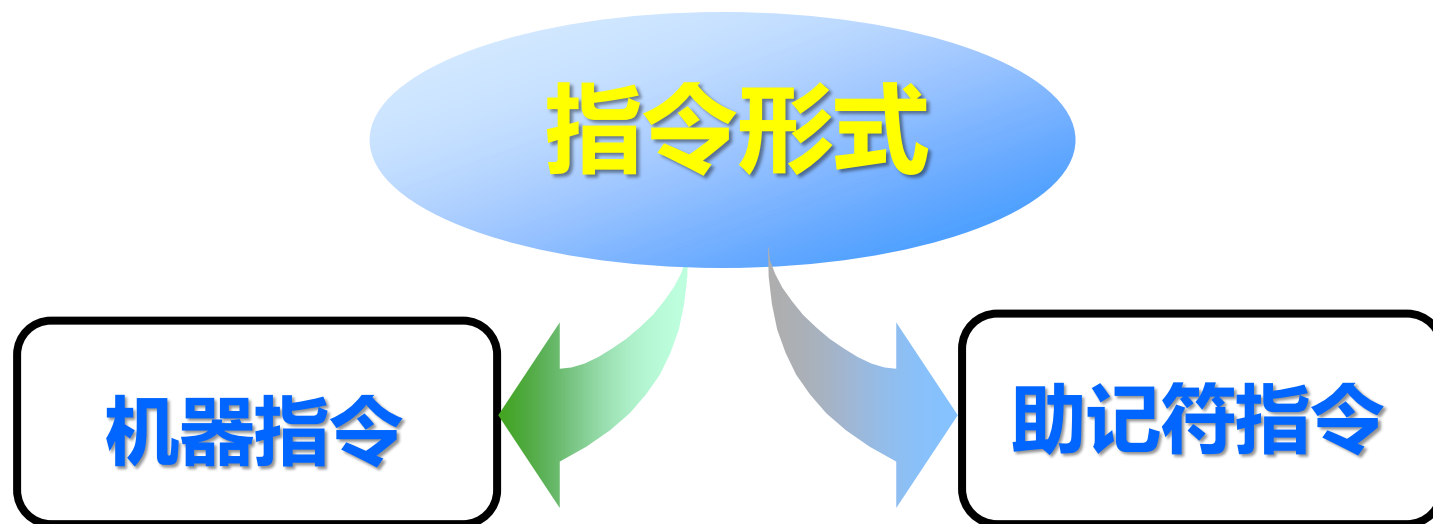


指令系统概述

5.1 指令系统概述

- 计算机工作通过执行计算机**程序**来体现。
- **程序**是由一系列有**时间顺序**、有**逻辑关系**的**指令**构成。
- **指令**是控制计算机硬件完成指定的基本操作（如加、减、移动等）的**命令**，是用户使用计算机和计算机本身运行的**最小功能单位**。
- 能被一台计算机执行的**全部指令的集合**称为该机的**指令系统**（也称**指令集**，Instruction Set）

5.1 指令系统概述



- 机器指令：**二进制数**表示，**CPU**能够识别
- 助记符(汇编)指令：**符号**表示，便于人识别

	机器指令						助记符指令	指令长度
Intel x86	00000101		00110100		00010010		add ax,1234h	3字节
RISC-V	0000000	00001	00010	000	00011	0110011	add x3,x2,x1	4字节

指令系统作用

- 指令系统**位于**指令系统结构层
- 指令系统结构层也称为**机器语言层**，是**机器语言程序员**看到的计算机。
- 指令系统结构层是计算机硬件和编译器都能理解的语言系统，定义了**编译器和硬件之间的接口**
- **指令系统**既为软件设计者提供最底层的程序设计语言，也为硬件设计者提供最基本的硬件设计依据
- 指令系统与计算机系统的**性能、硬件结构复杂度、制造成本、使用方便性**等密切相关，是**软件和硬件的主要界面**。

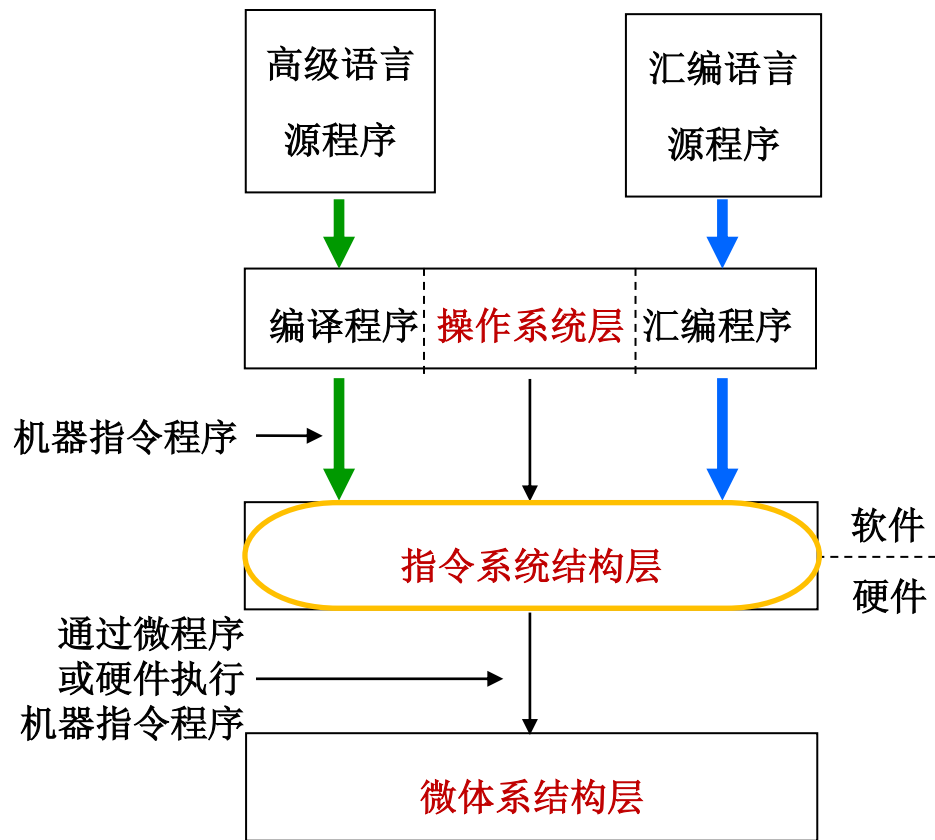


图5.1 指令系统的分层抽象结构

5.1 指令系统概述

- ▶ 一个好指令系统，对于**硬件**设计：容易被高效率实现；对于**软件**设计：容易用它生成代码。
- ▶ 设计、评价指令系统一般从以下方面考虑：
 - **完备性**：常用指令齐全，编程方便。
 - **高效性**：程序占主存空间少，运行速度快。
 - **规整性**：指令和数据使用规则统一简单，易学易记。
 - **兼容性**：即向后兼容（backward compatible）。
- ▶ 设计指令系统的核心问题：设计指令的功能和格式
 - **指令功能**由计算机的功能确定
 - **指令格式**则与**计算机字长、存储器容量和存储模式、寄存器组织、数据类型、硬件结构复杂度、运算性能**等有关。
（为什么？）



指令系统结构层

--存储模式

5.2.1 存储模式

- 存储模式定义包括存储器结构、特殊存储区（如堆栈等）、数据存储顺序、边界对齐等相关事宜的确定
- 按字节编址存储
- 数据存储顺序（字的地址都是低地址）
 - 大端存储：数据的低字节存储在高地址单元中
 - 小端存储：数据的低字节存储在低地址单元中，如x86、RISC-V

例5.1 一个32位的十六进制数据12345678H，存储在1000H地址开始的以字节编址的主存空间中，

地址	大端存储	小端存储
1000H	12H	78H
1001H	34H	56H
1002H	56H	34H
1003H	78H	12H

5.2.1 存储模式

- **边界对齐**：存储地址是 2 的整数倍(16位字长数据)，4 的整数倍(32位字长)或8的整数倍(64位字长)。即字长是几个字节，存储地址是几的整数倍。

8字节								地址
								24
								16
08H	07H	06H	05H	04H	03H	02H	01H	8
								0

边界对齐

8字节								地址
								24
				08H	07H	06H	05H	16
04H	03H	02H	01H					8
								0

边界未对齐

- 边界对齐可以加快读写数据的速度。为什么？

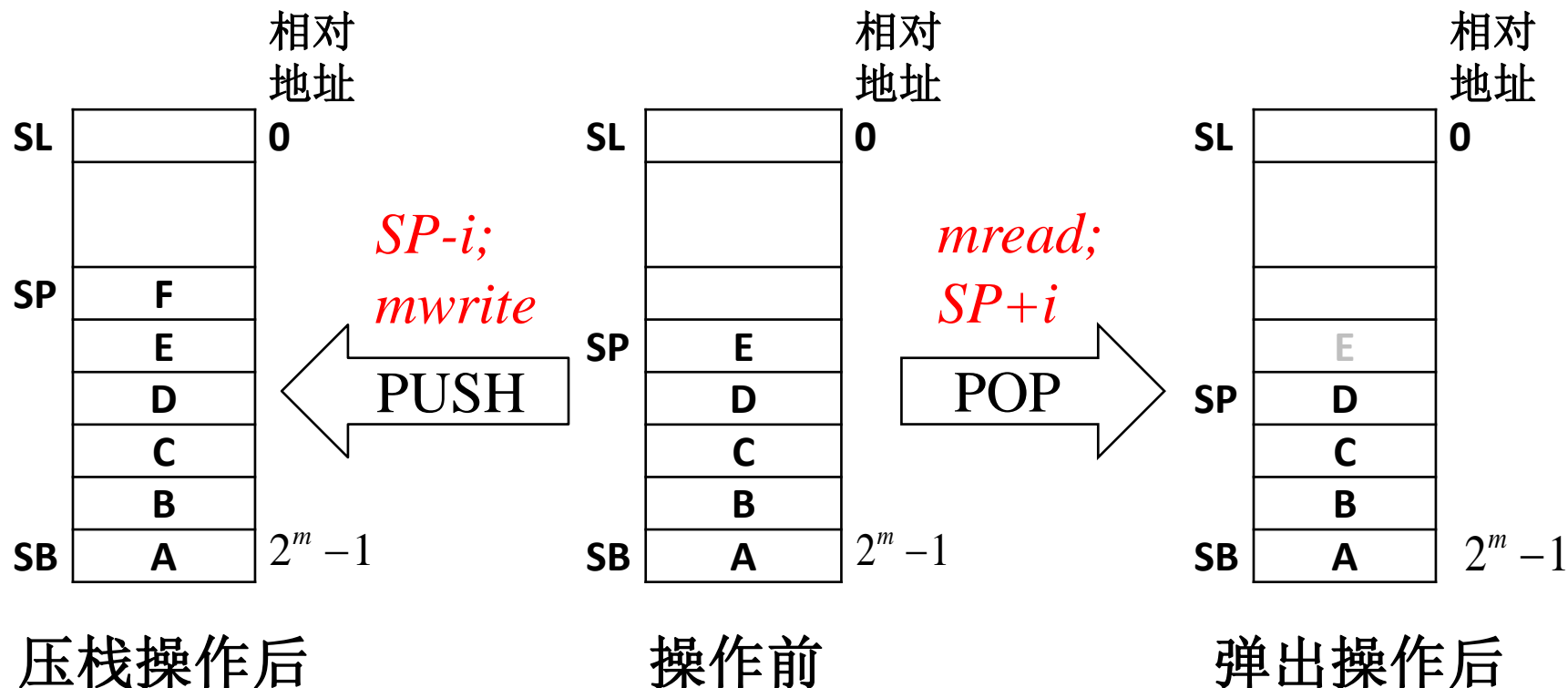
5.2.1 存储模式

- **堆栈**：具有**先进后出(FILO)**操作规则的被特殊定义的主存区域。
- 在使用堆栈之前，要对堆栈的大小、在主存中的位置做出定义。使用堆栈时用三个专用地址寄存器来管理
 - 堆栈指针 (Stack Pointer, **SP**)：指示当前可操作的堆栈单元。
 - 堆栈基址 (Stack Base, **SB**)：指示堆栈的底部。
 - 堆栈界限 (Stack Limit, **SL**)：指示堆栈的最顶端

堆栈界限=堆栈基址**±**堆栈大小。
- 堆栈设计方案
 - 堆栈界限**SL**>堆栈基址**SB**
 - 堆栈界限**SL**<堆栈基址**SB**

5.2.1 存储模式

- ▶ 对堆栈有两种基本操作：压栈（**PUSH**）和弹出（**POP**），也称为入栈和出栈。



(假设栈底地址SB>栈顶地址SL)

5.2.1 存储模式

- 为了防止堆栈操作错误，
 - 堆栈大小要按需求设置的**足够大**（预留裕度）；
 - 用户对堆栈的压栈和弹出**操作要成对**进行，以防止堆栈溢出；
 - 最好将**系统堆栈和用户堆栈分开**，以免用户破坏系统对堆栈的正常使用。

5.2.1 存储模式

冯·诺依曼结构和哈佛结构

— **冯·诺依曼结构**：指令与数据混合存储的主存架构

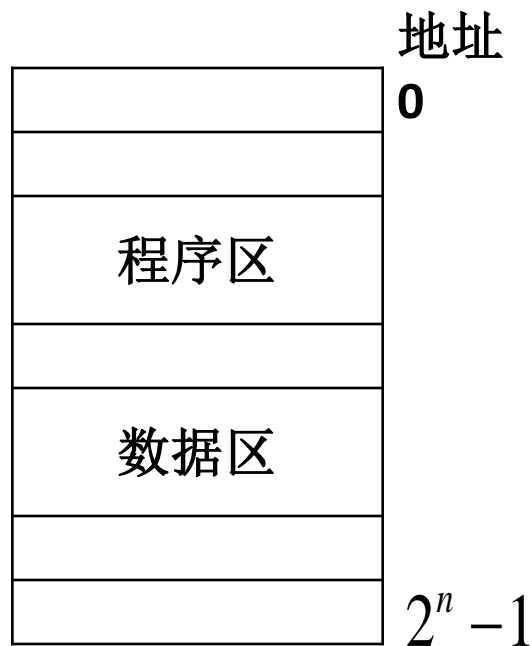
- » 优点：指令和数据可以共享并充分利用主存资源
- » 缺点：不正确的数据操作可能会破坏到指令，造成程序可能无法正常运行。

— **哈佛结构**：将主存分为指令地址空间和数据地址空间，程序必须放在指令存储器中，数据必须放在数据存储器中。

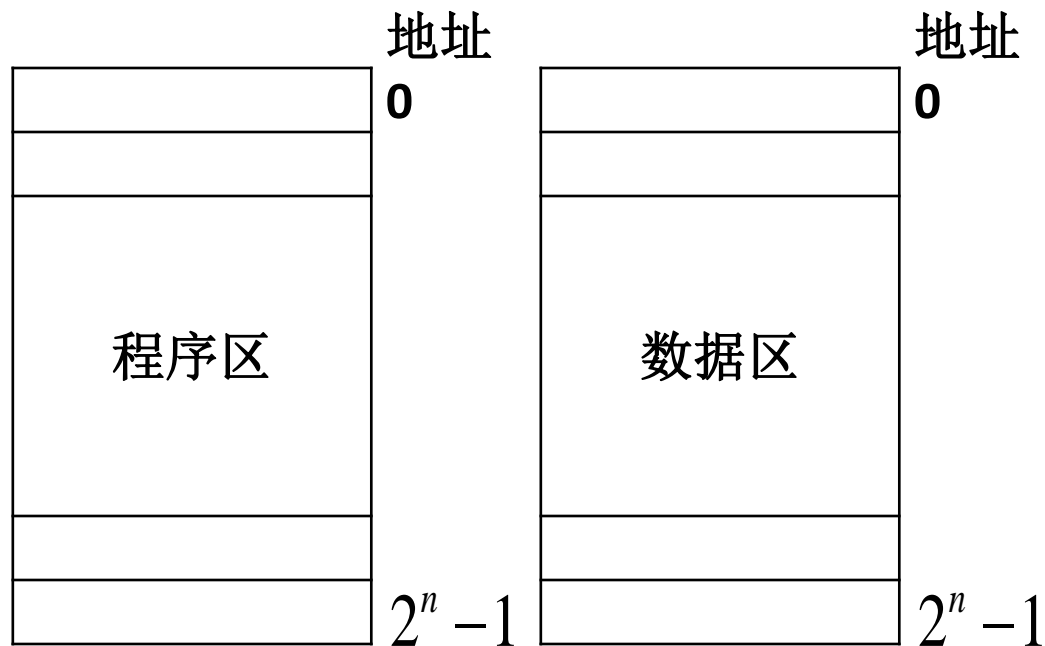
- » 优点：(1)指令存储器设计、控制简单，可以加快读指令的速度；(2)避免数据对程序可能造成的破坏；(3)利用 n 位地址可以获得2个 2^n 大小的地址空间。
- » 缺点：指令存储器和数据存储器有可能不能得到充分利用；设计复杂。

5.2.1 存储模式

冯·诺依曼结构和哈佛结构



(a) 冯·诺依曼结构



(b) 哈佛结构

5.2.1 存储模式

➤ 加载/存储体系结构(Load/Store结构)

- 在加载/存储体系结构中，所有运算的源操作数由**寄存器或指令**提供，而不是直接来自主存；运算结果也必须放在寄存器中，而不是直接写入主存。
- 在许多**RISC(如RISC-V)**系统中采用加载/存储体系结构，指令系统即加载（load）和存储（store）指令，load指令实现将主存的数据传送到寄存器，store指令实现将寄存器的数据传送到主存。
- 减少**访存指令**，进而简化指令系统。

某16位字长计算机的堆栈区域的栈底（SB）位于高地址，当前堆栈指针SP大小为03100H，那么执行2次PUSH操作后，SP值为（ ）

A. 03102H

B. 03104H

C. 030FEH

D. 030FCH



指令系统结构层

--寄存器组织

5.2.2 寄存器组织

- ▶ 寄存器是**存储体系中最上层**（速度最快、容量最小）的存储部件，是**CPU内部**配合控制器、运算器工作的重要部件，也是指令系统结构层定义的重要对象。
- ▶ 从指令系统结构层来看，寄存器是**软件设计者唯一能操作**的**CPU内部资源**。
- ▶ 寄存器的**基本功能**是为**CPU**运行提供所需的信息、保存**CPU**运行产生的结果。寄存器的**主要用途**是对频繁使用的数据进行快速访问。
- ▶ **寄存器定义**包括寄存器功能、寄存器所属层次、寄存器规模（数量）、寄存器字长等信息的确定。

5.2.2 寄存器组织

➤ 寄存器按功能可分为两类

- **通用寄存器**：在多种场合都可以使用、各种信息都可以存储其中的寄存器。大多数系统的通用寄存器数量可以从几个到几十个，某些嵌入系统会达到上百个，具有完全对称、可互换使用的特点。通常用 $R_0 \sim R_{n-1}$ 表示。
- **专用寄存器**：一般只在特定场合下使用，每个专用寄存器只存特定的信息，如标志寄存器。专用寄存器的数量根据系统需要可以从几个到几十个不等。

5.2.2 寄存器组织

- ▶ 寄存器按可操作性分为可见与不可见寄存器。
 - 有些寄存器是指令系统结构层可见的，有些寄存器是微体系结构层可见的。一般而言，微体系结构层可见的寄存器在指令系统结构层可能是不可见的，而指令系统结构层可见的寄存器在微体系结构层一定可见。如程序计数器PC、堆栈指针SP均为微体系结构层和指令系统结构层均可见的寄存器。
 - 指令系统结构层可见的寄存器是可以程序访问的，其中一部分为用户程序可见的，如通用寄存器；另一部分寄存器只允许在内核模式下由操作系统使用。通常将用户程序可见的一组寄存器称为寄存器组(Register Set)或寄存器文件(Register File, RF)。
 - 仅微体系结构层可见的寄存器都是专用的，是不可以程序访问的，如指令寄存器IR。

5.2.2 寄存器组织

- ▶ **典型寄存器**有地址寄存器(AR)、数据寄存器(DR)、指令寄存器(IR)、程序计数器(PC)、堆栈指针寄存器(SP)、标志寄存器(FR)等。
 - **地址寄存器**：提供CPU访问主存或外设资源时的主存地址或I/O接口地址(外设地址)，或提供生成地址的相关信息。
 - » 地址寄存器可以有一到多个。
 - » 变址寄存器是一种特殊的地址寄存器，它具有每次使用完其内部地址之后能自动对其修改的功能。
 - **数据寄存器**：用来存放原始数据和处理结果、作为数据传输的来源和目的地。
 - » 在某些计算机系统中，数据寄存器除了存储数据之外还赋予了特殊功能，如作为累加器、计数器、地址寄存器、隐含寄存器等使用。
 - » 数据寄存器可以有几到几十个，一般为通用寄存器。

5.2.2 寄存器组织

- **指令寄存器**：暂存从主存中获得的CPU要执行的当前指令，并将该指令输出给控制器，由控制器译码执行它。
- **程序计数器**：指示当前指令在主存中的存放地址，它具有自动计数功能。
 - » 对于长度可变的指令，当依据程序计数器的内容从主存中取得一个**指令字节**后，程序计数器**自动加“1”**，指向下一个指令字节；
 - » 对于长度固定的指令，当依据程序计数器的内容从主存中取得一条完整指令后，程序计数器自动**加指令长度**，指向下一条指令，
 - » 不断地依据程序计数器的内容取得程序中所有要执行的指令并加以执行。
- **堆栈指针寄存器**：指示堆栈的当前操作单元，具体功能见5.2.1节。

5.2.2 寄存器组织

— **标志寄存器**也称为程序状态字，是内核模式和用户模式下均可使用的寄存器，它既用于保存**CPU**的当前工作状态，又用于为**CPU**运行提供必要的控制信息。

表明**CPU**工作状态的信息称为条件码或**状态标志**：

- » **Z**——零标志
- » **S**——符号标志
- » **C**——进位标志
- » **O**——溢出标志
- » **P**——奇偶标志
- » **A**——半加进位标志

其中，**Z**、**S**、**C**、**O**和**P**标志为程序可见的状态标志；而**A**标志为程序不可见的状态标志。

为**CPU**运行提供控制信息的标志称为**控制标志**：

- » **I**——中断允许标志(外部可屏蔽中断)
- » **T**——单步跟踪标志，也称陷阱标志。

5.2.2 寄存器组织

- 有些计算机系统还设置了运行模式标志（控制CPU运行在用户模式或内核模式下）、方向标志（控制变址寄存器按地址增大或减小的方向自动修改其内的地址）等。
 - 标志寄存器在用户模式下是可读的，但某些标志只能在内核模式下写入。
- 寄存器的长度一般依据计算机字长、运算器处理数据的长度、所存信息的长度等来确定，大多数寄存器的长度仍选为字节的整数倍。



指令系统结构层

--数据类型

5.2.3 数据类型

- ▶ 数据类型不同，处理方法不同，硬件支持也不同。
- ▶ 计算机系统可以处理的数据分为：**数值型数据**和**非数值型数据**。（第2章）
- ▶ 计算机系统要定义哪些数据类型，**与指令系统有关**（指令功能与数据类型、指令格式与数据长度等密切相关），**与硬件体系结构也有关**（如对定点数的处理需要定点运算器，对浮点数的处理需要浮点运算器）。
- ▶ 对计算机系统软硬件功能、性能、成本等做出综合考虑，再为计算机系统选择**最利于其实现**的数据类型。



指令系统结构层

--I/O模式

5.2.4 I/O模式

- 在指令系统结构层对I/O模式的定义主要涉及I/O结构。
- I/O结构的设计类似于主存结构，给I/O系统中的每个I/O设备分配不同的I/O地址，对I/O设备的操作也像对存储单元一样通过I/O地址进行读写操作。
- I/O地址有两种编码方式：存储器映射方式和I/O映射方式。
 - 存储器映射方式也称统一编址方式，就是将主存的一部分地址空间分配给I/O设备。优点：不需要专门的I/O指令，简化控制，编程灵活。缺点：地址相互制约，空间受限。ARM和RISC-V采用此方式。
 - I/O映射方式也称独立编址方式，就是主存与I/O设备使用各自独立的地址空间，使两者操作独立互不干扰。优点：操作明确，便于系统调试维护，地址空间不相互干扰。缺点：对I/O设备操作不够灵活。
- 在I/O映射方式时，I/O地址空间也可以采用按字或字节编址，也可以选择小端或大端模式输入输出数据，也有数据在I/O地址空间边界对齐的问题。



指令系统结构层

--指令

5.2.5 指令类型

- ▶ 指令系统是**指令系统结构层的主要特征**，它的每一条指令可以被**CPU**执行，并规定了计算机的基本功能。
- ▶ 由指令构成的程序控制了计算机系统的运行，并决定了计算机系统要完成的任务。
- ▶ 计算机的指令系统可能简单到只有二三十条指令，也可能多达上百条指令。
- ▶ 无论指令系统规模大小，根据指令功能，都可分为数据传送类、算术运算类、逻辑运算类、移位类、字符串操作类、数据转换类、输入/输出类、程序控制类、系统控制类等若干种基本的指令类型。

5.2.5 指令类型

- ▶ 由于硬件支持不同，I/O操作有不同的实现方式：
 - 设置**专用的I/O指令**（I/O编址采用**独立编址**方式）；
 - 用**传送类指令**实现I/O操作（I/O编址采用存储映射方式，即I/O与存储器**统一编址**）；
 - 非CPU控制实现I/O操作（如有DMA控制器、I/O处理器或通道处理机等支持）。
- ▶ 在**多用户、多任务**计算机系统中，必须设置**特权指令**（只供操作系统或其他系统软件使用）。
 - 特权指令主要用于**系统资源的分配和管理**，如检测用户的访问权限，修改虚拟存储管理的段表、页表，改变系统的工作模式，任务的创建和切换等。
 - 在某些多用户计算机系统中，为了**统一管理外围设备**，I/O指令也被当作特权指令，用户不能直接使用它们，需要输入输出时通过系统调用来实现。

用于指示执行的指令在内存中的地址的寄存器是（）

- A. 地址寄存器AR
- B. 程序计数器PC
- C. 指令寄存器IR
- D. 堆栈指针SP