

计算机组成原理

引言 考研大纲概述

全国硕士研究生计算机学科专业基础综合（408）考试，包含数据结构、计算机组成原理、操作系统和计算机网络4大专业基础课程，满分150分。其中，计算机组成原理45分，占总分的30%。

计算机组成原理的考察目标为：

1. 掌握单处理器计算机系统中主要部件的工作原理、组成结构以及相互连接方式。
2. 掌握指令集体系结构的基本知识和基本实现方法，对计算机硬件相关问题进行分析，并能够对相关部件进行设计。
3. 理解计算机系统的整机概念，能够综合运用计算机组成的基本原理和基本方法，对高级编程语言（C语言）程序中的相关问题进行分析，具备软硬件协同分析和设计能力。

计算机组成原理，主要讲的就是计算机系统的“组成”，以及其中各部件的具体实现和连接的原理。

第一章 计算机系统概述

1.1 本章大纲要求与核心考点

1.1.1 大纲内容

(一) 计算机系统层次结构

1. 计算机系统的基本组成
2. 计算机硬件的基本结构
3. 计算机软件 and 硬件的关系
4. 计算机系统的工作原理

“存储程序”工作方式，高级语言程序与机器语言程序之间的转换,程序和指令的执行过程。

(二) 计算机性能指标

吞吐量、响应时间；

CPU时钟周期、主频、CPI、CPU执行时间；

MIPS、MFLOPS、GFLOPS、TFLOPS、PFLOPS、EFLOPS、ZFLOPS。

1.1.2 核心考点

本章是计算机组成原理的概述，一般会考察相关概念，以选择题为主，也可能结合后续章节以综合题的形式来考察性能的分析。掌握本章的概念，是整个组成原理课程的基础。

计算机性能指标在历年真题中出现的频次较高，需要重点掌握。

1.1.3 真题分布

考点	考查次数	
	单项选择题	综合应用题
计算机系统层次结构	6	0
计算机性能指标	9	2

1.2 计算机系统简介

1.2.1 计算机系统的概念和分类

计算机系统由“硬件”和“软件”两大部分组成。

- 所谓“硬件”即指计算机的实体部份，它由看得见摸得着的各种电子元器件、各类光、电、机设备的实物组成，如主机、外设等等。
- 所谓“软件”，它是看不见摸不着的，由人们事先编制成具有各类特殊功能的信息组成。通常把这些信息，诸如各类程序寄寓于各类媒体中，如RAM、ROM、磁带、磁盘、光盘、甚至纸带等。它们通常被作为计算机的主存或辅存的内容。

计算机的软件通常又可以分为两大类：系统软件和应用软件。

- 系统软件又称为系统程序，主要用来管理整个计算机系统，监视服务，使系统资源得到合理调度，确保高效运行。它包括：标准程序库、语言处理程序(如将汇编语言翻译成机器语言的汇编程序；将高级语言翻译成机器语言的编译程序)、操作系统(如批处理系统、分时系统、实时系统)、服务性程序(如诊断程序、调试程序、连接程序等)、数据库管理系统、网络软件等等。
- 应用软件又称为应用程序，它是用户根据任务需要所编制的各种程序。如科学计算程序，数据处理程序，过程控制程序，事务管理程序等等。

1.2.2 计算机的发展

(一) 硬件的发展

1943 年，第二次世界大战进入后期，因战争的需要，美国国防部主导建造了第一台计算机 ENIAC(Electronic Numerical Integrator And Computer,ENIAC)，它的全称是“用电子管组成的电子数字积分机和计算机”。

从此以后，计算机的发展经历了电子管、晶体管、集成电路的世代发展，体积越来越小、性能越来越强，并从军事领域迅速扩展应用到生活生产的各个行业，成为了现代信息社会不可或缺的基础设备。

硬件技术对计算机更新换代的影响

代	时间	硬件技术	速度 (次/秒)
一	1946~1957	电子管	40, 000
二	1958~1964	晶体管	200, 000
三	1965~1971	中、小规模集成电路	1, 000, 000
四	1972~1977	大规模集成电路	10, 000, 000
五	1978~现在	超大规模集成电路	100, 000, 000

摩尔定律

集成电路出现之后，芯片集成度不断提高，从在一个芯片上集成成百上千个晶体管的中、小规模集成电路，逐渐发展到能集成成千上万个晶体管的大规模集成电路(LSI)和能容纳百万个以上晶体管的超大规模集成电路(VLSI)。

微芯片集成晶体管的数目增长非常迅速，其规律被总结为“微芯片上集成的晶体管数目每3年翻两番”，这就是所谓的“摩尔定律”。

摩尔定律的另一个常见表述是：每平方英寸电路板上的晶体管数量，每18个月翻一倍。

(二) 软件的发展

(1) 编程语言的发展

软件的编写离不开编程语言。编程语言的发展经历了机器语言、汇编语言和高级语言三个阶段。我们现在使用的编程语言一般都是高级语言。

(2) 操作系统的发展

随着计算机的发展及应用范围的扩大，逐渐形成了软件系统。而其中最重要的一类软件，是为了提高计算机性能和资源利用率而设计的，这就是“操作系统”。

操作系统的发展经历了批处理系统、分时系统、实时系统、PC操作系统、网络操作系统、分布式操作系统等多个阶段。目前我们的个人电脑使用的操作系统主要有：Windows、Mac OS、Linux。

1.3 计算机系统的层次结构

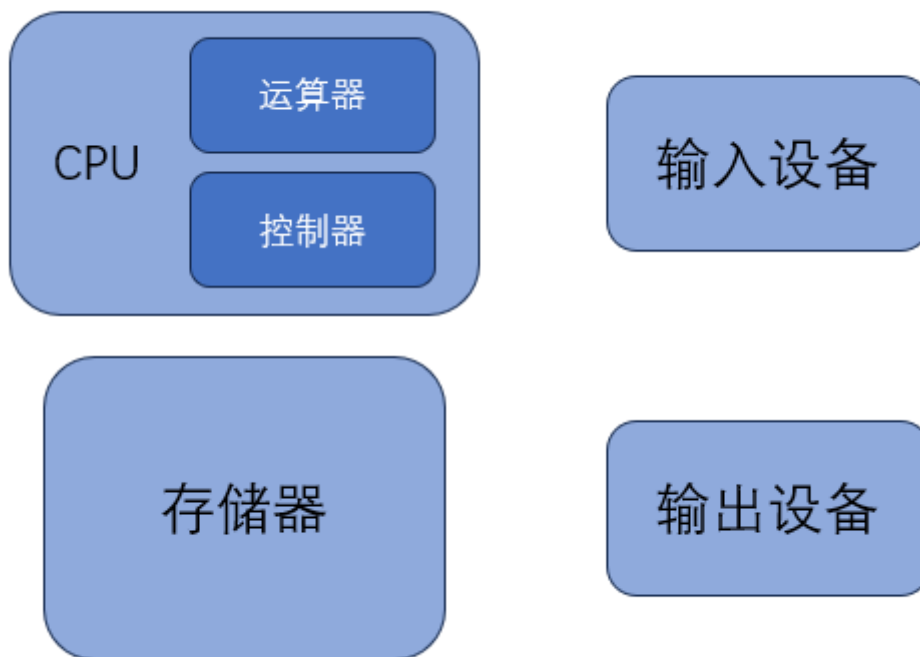
1.3.1 计算机系统的基本组成

(一) 计算机硬件

1. 冯·诺依曼计算机

冯·诺依曼在研究EDVAC计算机时提出了“**存储程序**”的概念，“存储程序”的思想奠定了现代计算机的基本结构，以此概念为基础的各类计算机通称为**冯·诺依曼计算机**，其特点如下：

- 采用“存储程序”的工作方式。
- 计算机硬件系统由运算器、存储器、控制器、输入设备和输出设备5大部件组成。
- **指令和数据以同等地位存储在存储器中，形式上没有区别，但计算机应能区分它们。**
- **指令和数据均用二进制代码表示。**
- 指令由操作码和地址码组成，操作码指出操作的类型，地址码指出操作数的地址。



2. 计算机的功能部件

- 存储器：分为主存和辅存，中央处理器可以直接访问的程序和数据存放在主存中。
- 运算器：完成对信息或数据的处理和运算，如算术和逻辑运算。
- 控制器：完成对计算机各部件协同运行的指挥控制，即保证指令按预定的次序执行，保障每一条指令按规定的执行步骤正确执行，还要处理各项紧急事件。
- 输入设备：用来输入原始数据和程序，如键盘、鼠标。
- 输出设备：用来输出计算机的处理结果，如显示器和打印机。

一般将运算器和控制器集成到同一个芯片上，称为中央处理器(CPU)。CPU和主存储器共同构成主机，而除主机外的其他硬件装置(外存、I/O设备等)统称为外部设备，简称外设。

(二) 计算机软件

1. 软件的分类

软件按其功能分类，可分为系统软件和应用软件。

2. 三个级别的计算机语言



(1) 机器语言

机器语言由二进制编码组成，它是计算机唯一可以直接识别和执行的语言。

(2) 汇编语言

汇编语言是用英文单词或其缩写代替二进制的指令代码，更容易为人们记忆和理解。汇编语言程序必须经过汇编操作，转换为机器语言后，才能在计算机硬件上执行。

(3) 高级语言

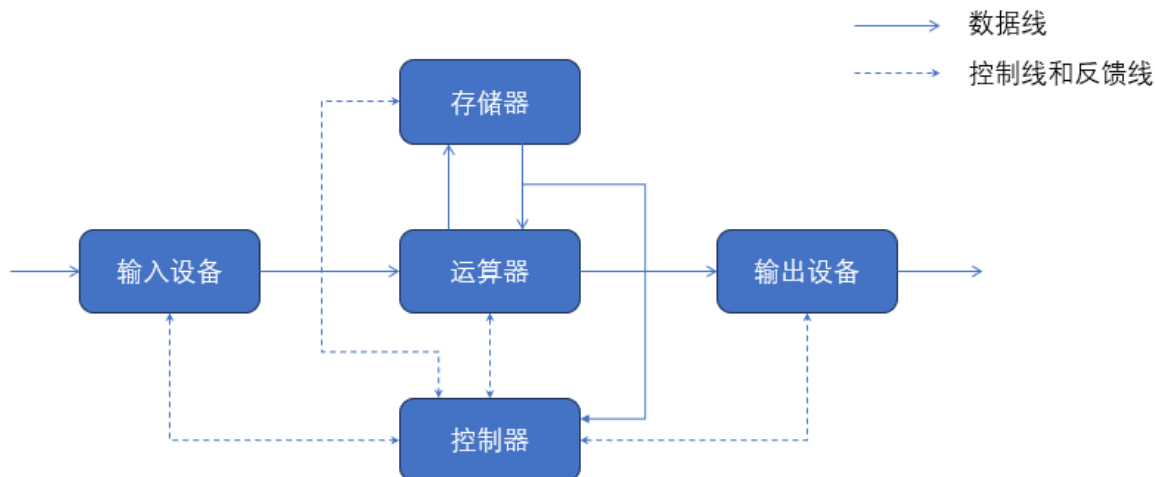
高级语言（如C、C++、Java等）程序需要先经过编译程序编译成汇编语言程序，再经过汇编操作成为机器语言程序。高级语言程序也可直接通过解释的方式“翻译”成机器语言程序。

由于计算机无法直接理解和执行高级语言程序，因此需要将高级语言程序转换为机器语言程序，通常把进行这种转换的软件系统称为翻译程序。翻译程序有以下三类：

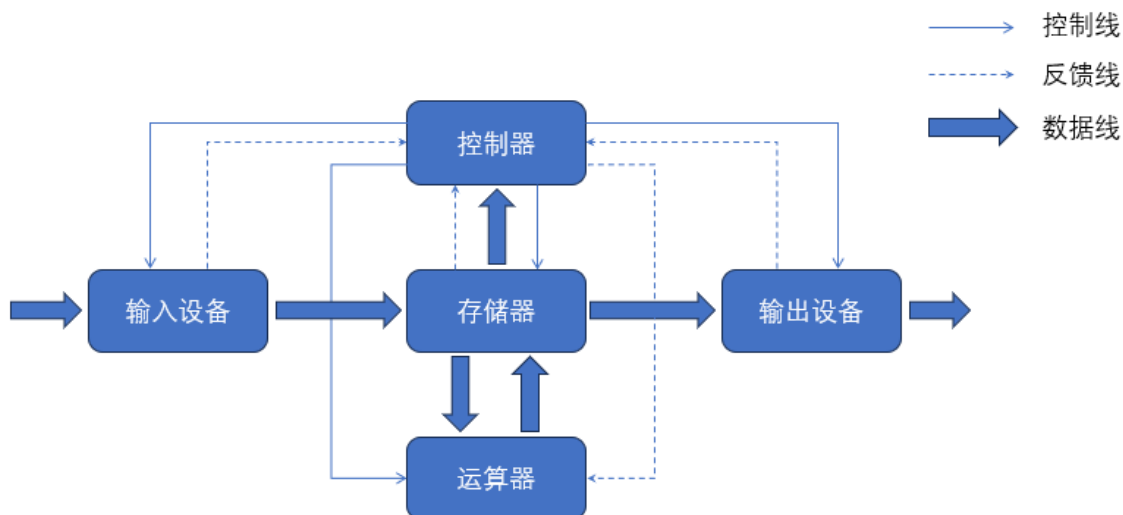
- 汇编程序（汇编器）：将汇编语言程序翻译成机器语言程序。
- 解释程序（解释器）：将高级语言源程序中的语句按执行顺序逐条翻译成机器指令并立即执行。
- 编译程序（编译器）：将高级语言源程序翻译成汇编语言程序或机器语言程序。

1.3.2 计算机硬件的基本结构

典型的冯·诺依曼计算机是以运算器为中心的，如下图所示。其中，输入、输出设备与存储器之间的数据传送都需通过运算器。图中实线为数据线，虚线为控制线和反馈线。



现代的计算机已转化为以存储器为中心，如下图所示。图中实线为控制线，虚线为反馈线，双线为数据线。

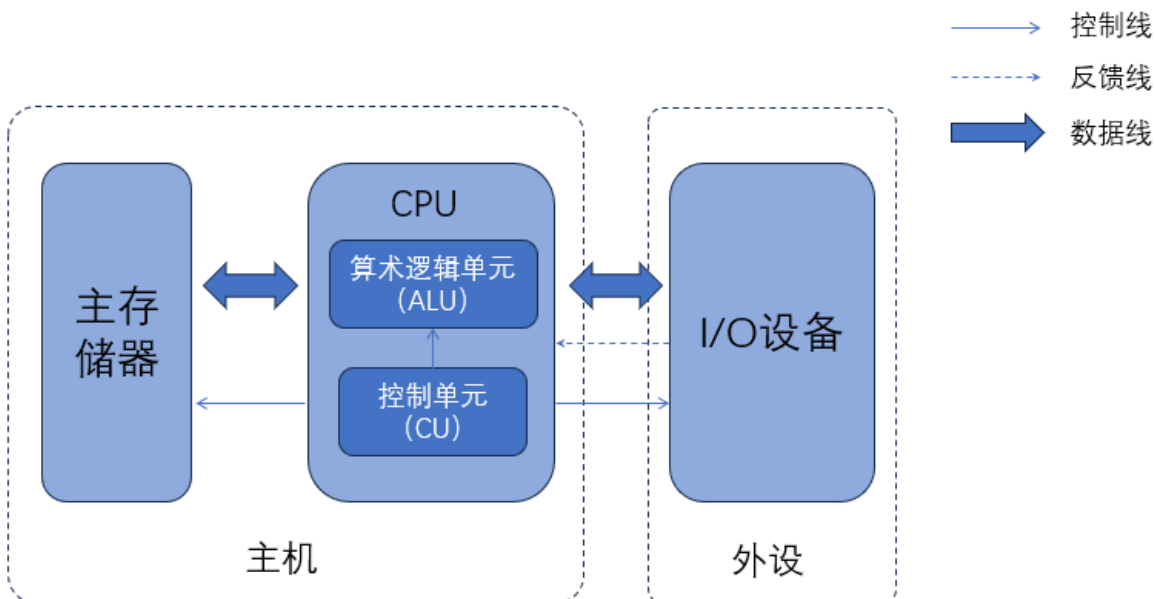


图中各部件的功能是：

- 运算器用来完成算术运算和逻辑运算，并将运算的中间结果暂存在运算器内；
- 存储器用来存放数据和程序；
- 控制器用来控制、指挥程序和数据的输入、运行以及处理运算结果；
- 输入设备用来将人们熟悉的信息形式转换为机器能识别的信息形式，常见的有键盘、鼠标等。
- 输出设备可将机器运算结果转换为人们熟悉的信息形式，如打印机输出、显示器输出等。

由于运算器和控制器在逻辑关系和电路结构上联系十分紧密，尤其在大规模集成电路制作工艺出现后，这两大部件往往制作在同一芯片上，因此，通常将它们合起来统称为中央处理器(Central Processing Unit)，简称CPU。把输入设备与输出设备简称为I/O设备(Input/Output equipment)。

这样，现代计算机可认为由三大部分组成：CPU、IO设备及主存储器(Main Memory, MM)。CPU与主存储器合起来又可称为主机，I/O设备叫作外部设备。



- 主存储器是存储器子系统的一类，用来存放程序和 data，它可以直接与CPU交换信息。另一类叫辅助存储器，简称辅存，又叫外存。
- ALU(Arithmetic Logic Unit) 算术逻辑运算单元，用来完成算术逻辑运算。
- CU(Control Unit) 控制单元，用来解释存储器中的指令，并发出各种操作命令来执行指令。

ALU和CU是CPU的核心部件。I/O设备也受CU控制，用来完成相应的输入、输出操作。可见，计算机有条不紊地自动工作，都是在控制器统一指挥下完成的。

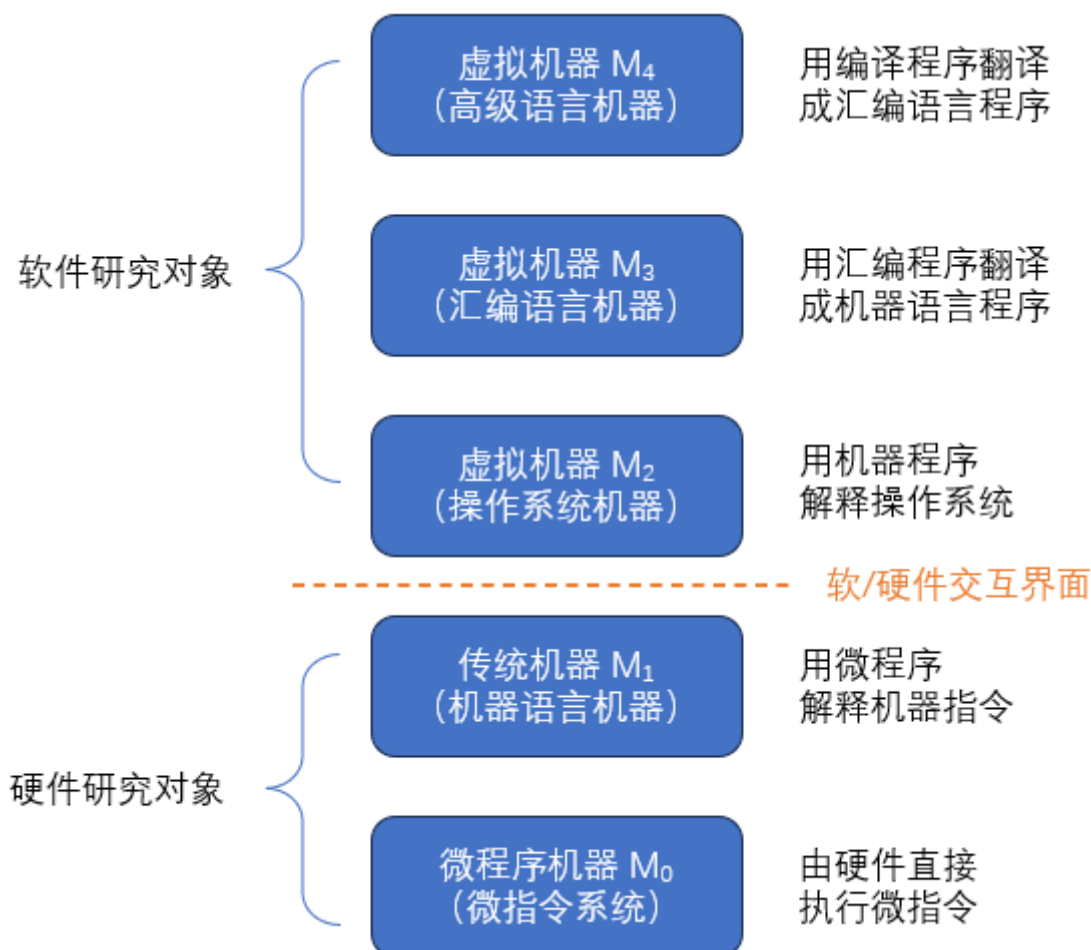
1.3.3 计算机软件 and 硬件的关系

硬件实现的往往是最基本的算术和逻辑运算功能，而其他功能大多通过软件的扩充得以实现。对某一功能来说，既可以由硬件实现，也可以由软件实现。从用户的角度来看，它们在功能上是等价的。这一等价性被称为软、硬件逻辑功能的等价性。

由于“软件”的发展，它不仅可以充分发挥计算机的“硬件”功能，提高计算机的工作效率，而且已经发展到能局部模拟人类的思维活动，因此在整个计算机系统内，“软件”的地位和作用已经成为评价计算机系统性能好坏的重要标志。当然，“软件”性能的发挥，也必须依托“硬件”的支撑。因此，概括而言，计算机性能的好坏，取决于“软”、“硬”件功能的总和。

1.3.4 计算机系统的多级层次结构

从用户的角度看，人们在操作系统提供的运行环境下，首先用高级语言编写程序（称为源程序），然后将源程序翻译成汇编语言程序，再将其翻译成机器能识别的机器语言程序（称为目标程序），最后用微程序解释每条机器指令。这样，就构成一个常见的计算机系统的5级层次结构，如下图所示：



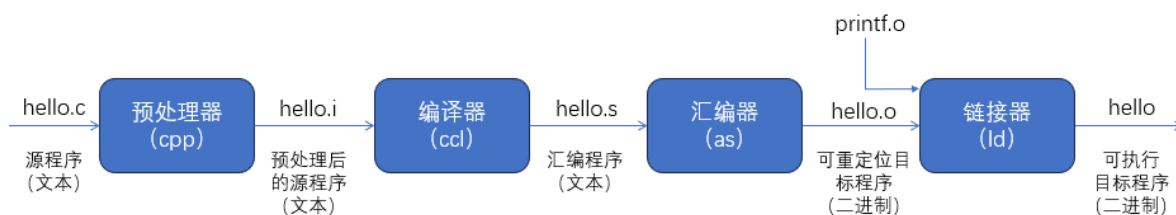
从计算机系统的5级层次结构来看，可以将硬件研究的对象归结为微程序机器 M₀ 与传统机器 M₁，也就是实际机器。而软件研究的对象主要是操作系统及其以上的各级虚拟机器。通常将除硬件系统外的其余层级称为**虚拟机器**，包括操作系统机器 M₂、汇编语言机器 M₃ 和高级语言机器 M₄。简单来说，虚拟机器就是由软件实现的机器。

相邻层级之间的关系，下层是上层的基础，上层是下层的扩展。随着超大规模集成电路技术的不断发展，部分软件功能可以由硬件来实现，所以软/硬件交界面的划分也不是绝对的。

1.3.5 计算机系统的工作原理

1. 从源程序得到可执行程序

用高级语言编写好一段程序之后，需要经过一系列“翻译”过程，才能得到计算机能够执行的机器代码。比如，我们用C语言写了一个简单的 hello world 程序，源程序文件命名为 hello.c，用GCC编译器可以将它翻译成一个可执行目标程序 hello。具体的过程可以分为4个阶段，如下图所示：



(1) 预处理阶段：预处理器 (cpp) 对源程序中以 “ # ” 开头的命令进行处理，输出结果是一个以 “.i” 为扩展名的文件 hello.i。例如 “ #include ” 就会将后面的头文件内容插入程序文件中。

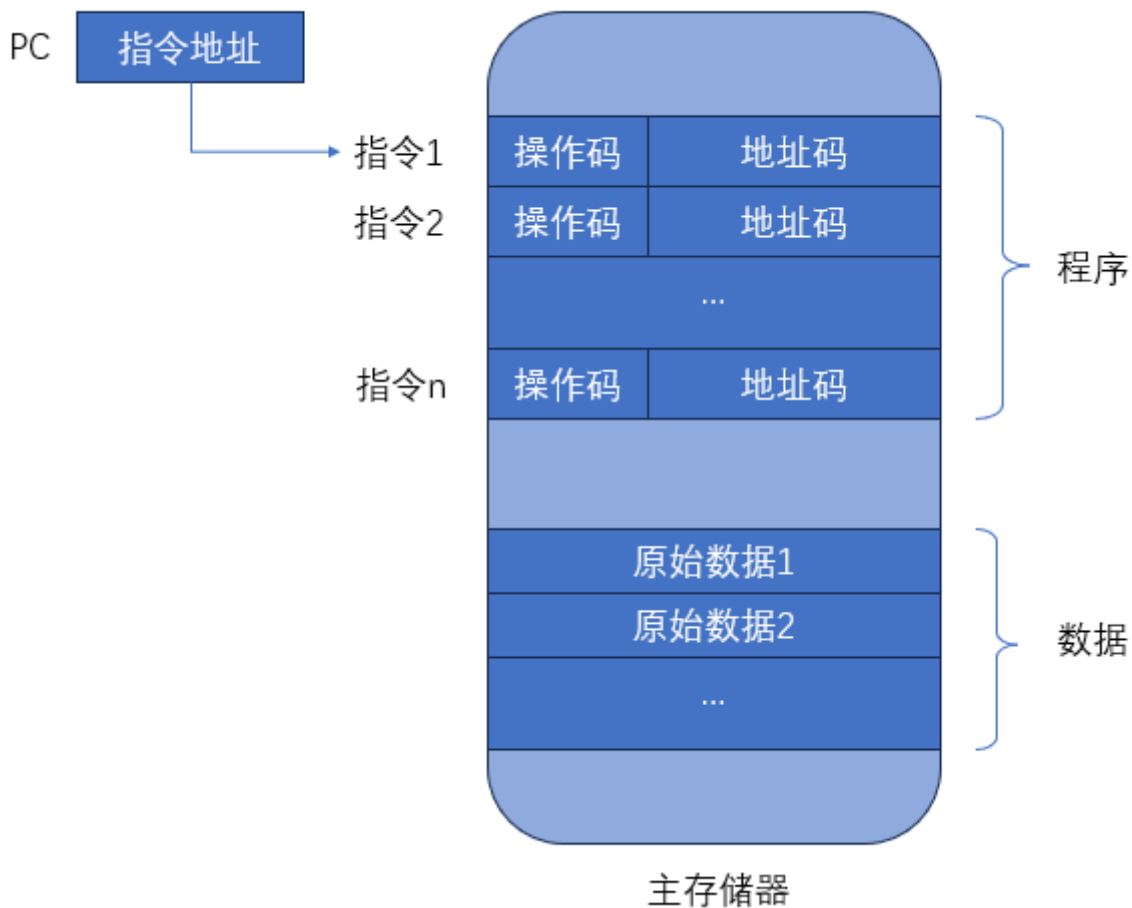
(2) 编译阶段：编译器 (ccl) 对预处理后的源程序进行编译，生成一个汇编语言源程序 hello.s。汇编语言源程序中的每条语句，都用文本格式描述了一条机器语言指令。

(3) 汇编阶段：汇编器 (as) 将 hello.s 翻译成机器语言指令，把这些指令打包成一个“可重定位目标程序” hello.o，它是一个二进制文件，用文本编辑器打开会显示乱码。

(4) 链接阶段：链接器 (ld) 将多个可重定位目标程序和标准库函数合并成一个可执行目标程序。上面的例子中，链接器将 hello.o 和标准库函数 printf 所在的可重定位目标模块 printf.o 合并，生成可执行程序 hello。最终生成的可执行程序被保存在磁盘上。

2. 存储程序的基本思想

“存储程序”的基本思想，就是将程序和数据一样，存放在主存中；运行时通过地址访问到程序的内容，解析出对应的指令进行执行。

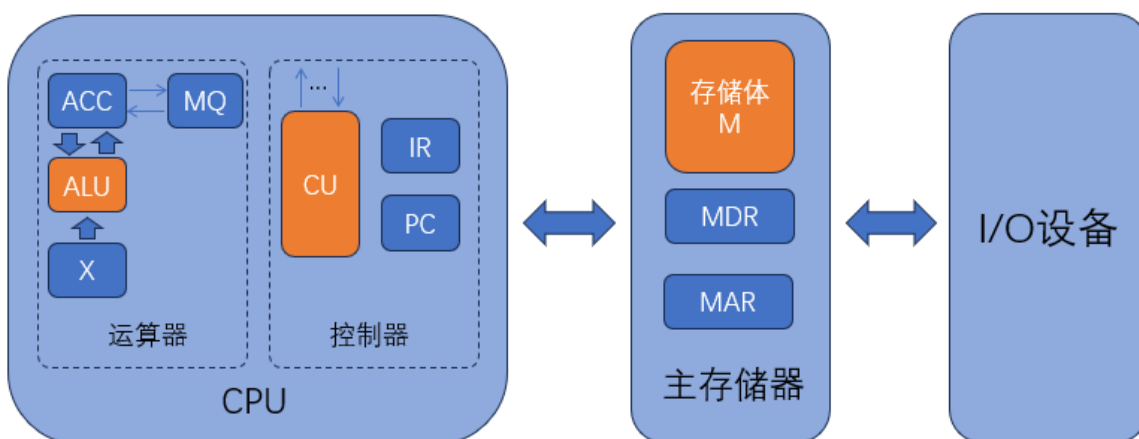


- 程序执行前，先将第一条指令的地址存放在程序计数器（PC）中；
- 将PC的内容作为地址访问主存，取出指令；
- 在每条指令执行过程中，都需要计算下一条将执行指令的地址，并送至PC。如果当前指令是顺序执行的，则下一条指令地址是PC的内容加上当前指令的长度；如果是跳转指令，则下一条指令的地址是指定的目标地址；
- 当前指令执行完毕后，再根据PC的值作为地址访问主存，取出的是下一条将要执行的指令。

这样，计算机就可以周而复始地自动执行程序中的每一条指令了。

3. 计算机硬件组成的细化说明

为了更清楚地了解计算机的工作过程，我们需要将计算机的组成部件进一步细化，如下图所示。



除了之前已经列出的核心部件 ALU、CU 以及主存的存储体M，CPU和主存储器中还必须配置一些寄存器（Register），用来存放特定的信息。下面我们分别进行简单介绍。

(1) 主存储器

主存储器（简称主存或内存）包括了存储体M、各种逻辑部件以及控制电路等。存储体由许多存储单元组成，每个存储单元又包含若干个存储元件；每个存储元件能存放一位二进制代码（0或者1）。这样，每个存储单元可以存储一串二进制代码，这就被称为一个“存储字”；存储字的二进制位数称为“**存储字长**”。

主存中的每个存储单元有一个唯一的编号，叫做存储单元的“**地址**”（Address）。主存的工作方式就是按照存储单元的地址，来实现对存储字各位的存（写入）取（读出）。这种存取方式叫做“按地址访问存储器”。

为了实现按地址访问的方式，主存中还必须配置两个寄存器：MAR 和 MDR。

- **MAR**（Memory Address Register，存储器地址寄存器）：用来存放想要访问的存储单元的地址，它的位数决定了能访问的存储单元的最大个数。
- **MDR**（Memory Data Register，存储器数据寄存器）：用来存放从存储体单元中取出，或者准备向存储体单元存入的数据，它的位数和存储字相等。

当然，如果想要完整地实现一个存取数据的操作，还需要 CPU 给主存加上各种控制信号，比如读命令、写命令以及地址译码驱动信号等。随着硬件技术的发展，主存现在都是统一制作的大规模集成电路芯片，所以一般都将MAR 和 MDR 集成在 CPU 芯片中。

(2) 运算器

运算器包括了一个算术逻辑单元（ALU）和最少三个寄存器。

- **ACC**：Accumulator，累加器；
- **MQ**：Multiplier-Quotient Register，乘商寄存器；
- **X**：操作数寄存器

这三个寄存器在完成不同的算术运算时，所存放的操作数也各不相同。具体的情况如下表所示：

	加法	减法	乘法	除法
ACC	被加数及和	被减数及差	乘积高位	被除数及余数
MQ	——	——	乘数及乘积低位	商
X	加数	减数	被乘数	除数

不同机器的运算器结构也有所不同，有的机器用 MDR 取代 X 寄存器。

(3) 控制器

控制器由控制单元（CU）和程序计数器（PC）、指令寄存器（IR）组成。

PC：Program Counter，程序计数器，用来存放当前将要执行指令的地址。它与主存的 MAR 之间有一条直接通路，且具有自动加1的功能，也就是可以自动形成下一条指令的地址。

IR：Instruction Register，指令寄存器，用来存放当前的指令。IR 的内容来自主存的 MDR，包含了操作码和地址码。IR 中的操作码 OP（IR）会送至 CU，可以记作 $OP(IR) \rightarrow CU$ ，用来分析指令；而地址码 Ad（IR）作为操作数的地址送至存储器的 MAR，可以记作 $Ad(IR) \rightarrow MAR$ 。

CU 是控制器的核心组件，用来分析当前指令所需完成的操作，并发出各种微操作命令序列，从而控制所有被控对象。控制器是计算机的神经中枢，由它指挥各部件自动协调地工作；完成一条指令操作，需要取指、分析和执行3个阶段。

(4) I/O

I/O 系统包括各种 I/O 设备及其相应的接口。每一种 I/O 设备都由 I/O 接口与主机联系，它接收 CU 发出的各种控制命令，并完成相应的操作。

4. 计算机硬件的工作过程

总结一下，当计算机接收到机器语言程序后，硬件的工作过程分为以下几步：

- 把程序和数据装入主存储器；
- 从程序的起始地址运行程序；
- 按照程序的首地址从存储器中取出第一条指令，经过译码等步骤控制计算机各功能部件协同运行，完成这条指令的功能，并计算下一条指令的地址；
- 用新得到的指令地址继续读出第二条指令并执行，直到程序结束为止。每条指令都是在取指、译码和执行的循环过程中完成的。

我们现在以从主存中取数据的指令为例，详细分析一下它的执行过程：

(1) 取指令：PC → MAR → M → MDR → IR

根据 PC 取指令到 IR。将 PC 的内容送至 MAR，将 MAR 的内容送至地址线，同时控制器将读信号送至读/写信号线，从主存指定存储单元读出指令，并通过数据线送至 MDR，再传送到 IR。

(2) 分析指令：OP (IR) → CU

指令译码并送出控制信号。控制器根据 IR 中指令的操作码，生成相应的控制信号，送到不同的执行部件。这里 IR 中是取数指令，所以读控制信号被送到总线的控制线上。

(3) 执行指令：Ad (IR) → MAR → M → MDR → ACC

取数操作。将 IR 中指令的地址码送至 MAR，将 MAR 的内容送至地址线，同时控制器将读信号送至读/写信号线，从主存指定存储单元读出操作数，并通过数据线送至 MDR，再传送到 ACC 中。

(4) 每取完一条指令，还必须计算下一条指令的地址，为取下一条指令做准备：(PC) + 1 → PC

1.4 计算机性能指标

衡量一台计算机的性能是由多项技术指标综合确定的，既包含硬件的各类性能，又包括软件的各种功能，这里主要讨论硬件的技术指标。

1.4.1 主要性能指标

1. 机器字长

机器字长，是指 **CPU 一次能处理数据的位数**，也就是 CPU 内部用于整数运算的数据通路的宽度。**字长通常就等于 CPU 的通用寄存器宽度**，也就是 **CPU 内用于整数运算的运算器位数**，它反映了计算机处理信息的能力。我们平常所说“一台16位或32位的机器”，这里的16、32就是指字长。

字长越长，数的表示范围也越大，精度也越高。机器的字长也会影响机器的运算速度。倘若CPU字长较短，又要运算位数较多的数据，那么需要经过两次或多次的运算才能完成，这样势必影响整机的运行速度。当然，机器字长对硬件的造价也有较大的影响。它将直接影响加法器(或ALU)、数据总线以及存储字长的位数。

- 指令字长：一条指令中包含的二进制代码的位数。
- 存储字长：一个存储单元中存储的二进制代码的长度。

指令字长和存储字长，都必须是字节 (Byte) 的整数倍。指令字长一般取存储字长的整数倍：如果指令字长等于存储字长的2倍，那么取一条指令就需要2个机器周期；如果指令字长等于存储字长，那么取指周期就等于机器周期。

2. 数据通路带宽

数据通路带宽，是指数据总线一次所能并行传送信息的位数，它关系到数据的传送能力。这里所说的数据通路带宽是指外部数据总线的宽度，它与 CPU 内部的数据总线宽度（机器字长）可能不同。

3. 存储容量

存储器的容量，包括主存容量和辅存容量。我们一般主要关心主存容量。

主存容量是指主存中所能存储信息（二进制代码）的最大容量，通常以字节数来衡量。

$$\text{存储容量} = \text{存储单元个数} \times \text{存储字长}$$

在主存储器中，MAR 的位数反映了存储单元的个数，MDR 的位数则反映了存储字的长度。例如，MAR 为16位，表示 $2^{16} = 65536$ ，也就是说对应的存储体内有65536个存储单元（一般称为64K内存，1K=1024）；而如果 MDR 为32位，那么主存的存储容量为 $2^{16} \times 32 = 2^{21} = 2\text{M 位}$ ($1\text{M}=2^{20}$)。

现代计算机中常以字节的个数来描述容量的大小，一个**字节**（Byte）被定义位8位二进制代码。所以上述存储容量是 2M 位（bit），也可用 2^{18} 字节（Byte，简称为 B）表示，记作 2^{18}B 或 256KB。

同理，辅存容量也可用字节数来表示，例如，某机辅存（如硬盘）容量为 128 GB ($1\text{G} = 1\text{KM} = 2^{30}$)。

4. 运算速度

计算机的运算速度与许多因素有关，如机器的主频、CPU 的结构、执行什么样的操作、主存本身的速度（主存速度快，取指、取数就快）等等都有关。

1.5.2 专业术语解释

吞吐量：系统在单位时间内处理请求的数量，主要取决于主存的存取周期。

响应时间：从用户向计算机发送一个请求，到系统对该请求做出响应并获得所需结果的时间。通常包括 CPU 时间（**计算机执行程序的时间**）和**等待时间**（用于磁盘访问、存储器访问、I/O操作等的时间）。

主频（CPU 时钟频率）：机器内部主时钟的频率，它是衡量机器速度的重要参数。对统一型号的计算机，主频越高，完成指令的一个步骤所用的时间越短，执行指令的速度越快。通常以赫兹（Hz）为单位。

CPU 时钟周期：节拍脉冲的宽度或周期，也就是主频的倒数，它是 **CPU 中最小的时间单位**。

$$\text{CPU 时钟周期} = 1 / \text{主频}$$

CPI：Clock cycle Per Instruction，**执行一条指令所需的时钟周期数**。对一个程序或一台机器来说，CPI 指的是该程序或该机器指令集中所有指令执行所需要的平均时钟周期数。

CPU 执行时间：运行一个程序所花费的时间。

$$\text{CPU 执行时间} = (\text{指令数} \times \text{CPI}) / \text{主频}$$

对于同一个程序，CPU 的执行时间就代表了 CPU 的性能，它主要取决于三个要素：主频、CPI 和指令数。这三者是相互制约的。**不同的机器可以有不同指令集，更改指令集可以让程序的指令数更少，但 CPI 可能就会增大；同时可能引起 CPU 结构的调整，从而造成主频的降低。**

现在机器的运算速度，普遍采用单位时间内执行指令的平均条数来衡量，并用 MIPS（Million Instruction Per Second）作为计量单位，即每秒执行百万条指令。比如，某机每秒能执行200万条指令，则记作2 MIPS。

MIPS：Million Instructions Per Second，每秒执行百万条指令的数目。

$$\text{MIPS} = \text{主频} / (\text{CPI} \times 10^6)$$

FLOPS: Floating-point Operations Per Second, 每秒执行浮点运算的次数。

MFLOPS: 百万次浮点运算每秒。 $\text{MFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^6)$

GFLOPS: 十亿次浮点运算每秒。 $\text{GFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^9)$

TFLOPS: 万亿次浮点运算每秒。 $\text{TFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^{12})$

PFLOPS: 千万亿次浮点运算每秒。 $\text{PFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^{15})$

EFLOPS: 百京次浮点运算每秒。 $\text{EFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^{18})$

ZFLOPS: 十万京次浮点运算每秒。 $\text{ZFLOPS} = \text{浮点操作次数} / (\text{执行时间} * 10^{21})$

需要注意, 在计算机中, 描述存储容量、文件大小时, K 、 M 、 G 、 T 等数量单位通常用2的幂次表示, 比如 $1 \text{ KB} = 2^{10} \text{ B}$; 而在描述速率、频率等概念时, 通常用10的幂次表示, 比如 $1 \text{ kb/s} = 10^3 \text{ b/s}$ 。

1.5 章节练习

一、单项选择题

1. 【2009真题】冯·诺依曼计算机中指令和数据均以二进制形式存放在存储器中, CPU 区分它们的依据是 ()。
- A. 指令操作码的译码结果 B. 指令和数据的寻址方式
- C. 指令周期的不同阶段 D. 指令和数据所在的数据单元

答案: C

2. 【2015真题】计算机硬件能够直接执行的是 ()。
- I. 机器语言程序 II. 汇编语言程序 III. 硬件描述语言程序

A. 仅 I B. 仅 I、II C. 仅 I、III D. I、II、III

答案: A

3. 【2016真题】将高级语言源程序转换为机器级目标代码文件的程序是 ()。
- A. 汇编程序 B. 链接程序 C. 编译程序 D. 解释程序

答案: C

4. 【2019真题】下列关于冯·诺依曼结构计算机基本思想的叙述中, 错误的是 ()。
- A. 程序的功能都通过中央处理器执行指令实现 B. 指令和数据都用二进制表示, 形式上无差别
- C. 指令按地址访问, 数据都在指令中直接给出 D. 程序执行前, 指令和数据需预先存放在存储器中

答案: C

5. 【2020真题】下列给出的部件中, 其位数(宽度)一定与机器字长相同的是 ()。
- I. ALU II. 指令寄存器 III. 通用寄存器 IV. 浮点寄存器

A. 仅 I、II B. 仅 I、III C. 仅 II、III D. 仅 II、III、IV

答案: B

6. 【2010真题】下列选项中，能缩短程序执行时间的是（ ）。

I. 提高 CPU 时钟频率 II. 优化数据通路结构 III. 对程序进行编译优化

A. 仅 I 和 II B. 仅 I 和 III C. 仅 II 和 III D. I、II、III

答案：D

7. 【2011真题】下列选项中，描述浮点数操作速度的是（ ）。

A. MIPS (**MIPS**: Million Instructions Per Second, 每秒执行百万条指令的数目)

B. CPI (**CPI**: Clock cycle Per Instruction, **执行一条指令所需的时钟周期数**。对一个程序或一台机器来说, CPI 指的是该程序或该机器指令集中所有指令执行所需要的平均时钟周期数。)

C. IPC (**1/CPI**)

D. MFLOPS 百万次浮点运算每秒。MFLOPS = 浮点操作次数 / (执行时间 * 10^6)。

答案：D

8. 【2012真题】假定基准程序 A 在某计算机上的运行时间为 100s, 其中 90s 为 CPU 时间, 其余为 I/O 时间。若 CPU 速度提高 50%, I/O 速度不变, 则运行基准程序 A 所耗费的时间是（ ）。

A. 55s B. 60s C. 65s D. 70s

答案：D

9. 【2013真题】某计算机的主频为 1.2 GHz, 其指令分为4类, 它们在基准程序中所占比例及CPI如下表所示。

指令类型	所占比例	CPI
A	50%	2
B	20%	3
C	10%	4
D	20%	5

该机的 MIPS 是（ ）。

MIPS = 主频 / (CPI × 10^6)

A. 100 B. 200 C. 400 D. 600

答案：C

10. 【2014真题】程序 P 在机器 M 上的执行时间是 20s, 编译优化后, P 执行的指令数减少到原来的 70%, 而 CPI 增加到原来的1.2倍, 则 P 在 M 上的执行时间是（ ）。

A. 8.4s B. 11.7s C. 14s D. 16.8s

答案：D

11. 【2017真题】假定计算机 M1 和 M2 具有相同的指令集体系结构 (ISA), 主频分别为 1.5GHz 和 1.2 GHz。在 M1 和 M2 上运行某基准程序 P, 平均 CPI 分别为 2 和 1, 则程序 P 在 M1 和 M2 上运行时间的比值是（ ）。

A. 0.4 B. 0.625 C. 1.6 D. 2.5

答案： C

二、综合应用题

1. 说明机器字长、指令字长、存储字长的区别和联系。

答案：

机器字长：计算机能直接处理的二进制数据的位数，机器字长一般等于内部寄存器的大小，它决定了计算机的运算精度。

指令字长：一个指令字中包含二进制代码的位数。

存储字长：一个存储单元存储二进制代码的长度。

它们都必须是字节的整数倍。指令字长一般取存储字长的整数倍，如果指令字长等于存储字长的2倍，就需要2次访存来取出一条指令，因此，取指周期为机器周期的2倍；如果指令字长等于存储字长，则取指周期等于机器周期。早期的计算机存储字长一般和机器的指令字长与数据字长相等，故访问一次主存便可以取出一条指令或一个数据。随着计算机的发展，指令字长可变，数据字长也可变，但它们都必须是字节的整数倍。

2. 用一台 40MHz 的处理器执行基准程序，它所包含的混合指令数和响应所需的时钟周期见下表。求有效的 CPI、MIPS 和程序的执行时间（程序的指令条数为 I ）。

指令类型	CPI	指令混合比
算术和逻辑	1	60%
转移	4	12%
高速缓存命中的访存	2	18%
高速缓存失效的访存	8	10%

答案：

CPI 是执行一条指令所需的平均时钟周期数。本程序中包含4种指令，根据它们不同的占比，CPI 就是这4种指令的数学期望：

$$CPI = 1 \times 60\% + 4 \times 12\% + 2 \times 18\% + 8 \times 10\% = 0.6 + 0.48 + 0.36 + 0.8 = 2.24$$

MIPS 是每秒执行的百万条指令数。已知时钟频率为 40MHz，也就是每秒有 40M 个时钟周期，所以：

$$MIPS = 40 \times 10^6 / (2.24 \times 10^6) \approx 17.9$$

程序的执行时间 T = 平均每条指令执行时间 \times 指令条数，而平均每条指令执行的时间，就是 CPI \times 时钟周期：

$$T = CPI \times \text{时钟周期} \times \text{指令条数} = 2.24 \times (1 / 40\text{MHz}) \times I = 5.6 \times 10^{-8} \times I \text{ 秒}$$