

第 1 章

计算机系统概述

【考纲内容】

(一) 计算机发展历程

(二) 计算机系统层次结构

计算机硬件的基本组成、计算机软件的分类、计算机的工作过程

(三) 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

【复习提示】

本章是组成原理的概述，考查时易针对有关概念或性能指标出选择题，也可能综合后续章节的内容出有关性能分析的综合题。掌握本章的基本概念，是学好后续章节的基础。部分知识点在初学时理解不深刻也无须担忧，相信随着后续章节的学习，一定会有更为深入的理解。本章中读者要重点掌握各个性能指标的计算，这部分内容在历年真题中出现的频率很高。

学习本章时，请读者思考以下问题：

- 1) 计算机由哪几部分组成？以哪部分为中心？
- 2) 主频高的 CPU 一定比主频低的 CPU 快吗？为什么？
- 3) 翻译程序、汇编程序、编译程序、解释程序有什么差别？各自的特性是什么？
- 4) 不同级别的语言编写的程序有什么区别？哪种语言编写的程序能被硬件直接执行？

请读者在学习本章的过程中寻找答案，本章末尾会给出参考答案。

1.1 计算机发展历程

1.1.1 计算机硬件的发展

1. 计算机的四代变化

从 1946 年世界上第一台电子数字计算机 (Electronic Numerical Integrator And Computer, ENIAC) 问世以来，计算机的发展已经经历了四代。

- 1) 第一代计算机 (1946—1957 年) ——电子管时代。特点：逻辑元件采用电子管；使用机器语言进行编程；主存用延迟线或磁鼓存储信息，容量极小；体积庞大，成本高；运算速度较低，一般只有几千次到几万次每秒。
- 2) 第二代计算机 (1958—1964 年) ——晶体管时代。特点：逻辑元件采用晶体管；运算速度提高到几万次到几十万次每秒；主存使用磁心存储器；软件开始使用高级语言，如

1.1.4 本节习题精选

单项选择题

1. 电子计算机的发展已经历了 4 代, 这 4 代计算机的主要元件分别是 ()。
 - A. 电子管、晶体管、中小规模集成电路、激光器件
 - B. 晶体管、中小规模集成电路、激光器件、光介质
 - C. 电子管、晶体管、中小规模集成电路、大规模集成电路
 - D. 电子管、数码管、中小规模集成电路、激光器件
2. 微型计算机的发展以 () 技术为标志。
 - A. 操作系统
 - B. 微处理器
 - C. 磁盘
 - D. 软件
3. 可以在计算机中直接执行的语言和用助记符编写的语言分别是 ()。
 - I. 机器语言 II. 汇编语言 III. 高级语言 IV. 操作系统原语 V. 正则语言
 - A. II、III
 - B. II、IV
 - C. I、II
 - D. I、V
4. 只有当程序执行时才将源程序翻译成机器语言, 并且一次只能翻译一行语句, 边翻译边执行的是 () 程序, 把汇编语言源程序转变为机器语言程序的过程是 ()。
 - I. 编译 II. 目标 III. 汇编 IV. 解释
 - A. I、II
 - B. IV、II
 - C. IV、I
 - D. IV、III
5. 到目前为止, 计算机中所有的信息仍以二进制方式表示的理由是 ()。
 - A. 节约元件
 - B. 运算速度快
 - C. 由物理器件的性能决定
 - D. 信息处理方便

1.1.5 答案与解析

单项选择题

1. C

此题也可根据元件的先进程度的升序得出答案。

2. B

微型计算机的发展是以微处理器的技术为标志的。

3. C

机器语言是计算机唯一可以直接执行的语言, 汇编语言用助记符编写, 以便记忆。而正则语言是编译原理中符合正则文法的语言。

4. D

解释程序的特点是翻译一句执行一句, 边翻译边执行; 由高级语言转化为汇编语言的过程称为编译, 把汇编语言源程序翻译成机器语言程序的过程称为汇编。

5. C

二进制只有 1 和 0 两个数字, 刚好和逻辑电路中的高、低电平对应, 实现起来比较方便且简单可靠, 因此由物理器件的性能决定。



1.2 计算机系统层次结构

1.2.1 计算机系统的组成

硬件系统和软件系统共同构成了一个完整的计算机系统。硬件是指有形的物理设备，是计算机系统中实际物理装置的总称。软件是指在硬件上运行的程序和相关的文档及数据。

计算机系统性能的好坏，很大程度上是由软件的效率和作用来表征的，而软件性能的发挥又离不开硬件的支持。对某一功能来说，其既可以用软件实现，又可以用硬件实现，则称为软硬件在逻辑上是等效的。在设计计算机系统时，要进行软/硬件的功能分配。通常来说，一个功能若使用较为频繁且用硬件实现的成本较为理想，则使用硬件解决可以提高效率。而用软件实现可以提高灵活性，但效率往往不如硬件实现高。

1.2.2 计算机硬件的基本组成

1. 早期的冯·诺依曼机

冯·诺依曼在研究 EDVAC 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·诺依曼机，其特点如下：

- 1) 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。
- 2) 指令和数据以同等地位存储在存储器中，并可按地址寻访。
- 3) 指令和数据均用二进制代码表示。
- 4) 指令由操作码和地址码组成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置。
- 5) 指令在存储器内按顺序存放。通常，指令是顺序执行的，在特定条件下可根据运算结果或根据设定的条件改变执行顺序。
- 6) 早期的冯·诺依曼机以运算器为中心，输入/输出设备通过运算器与存储器传送数据。

典型的冯·诺依曼计算机结构如图 1.1 所示。

注意：“存储程序”的概念是指将指令以代码的形式事先输入计算机的主存储器，然后按其在存储器中的首地址执行程序的第一条指令，以后就按该程序的规定顺序执行其他指令，直至程序执行结束。

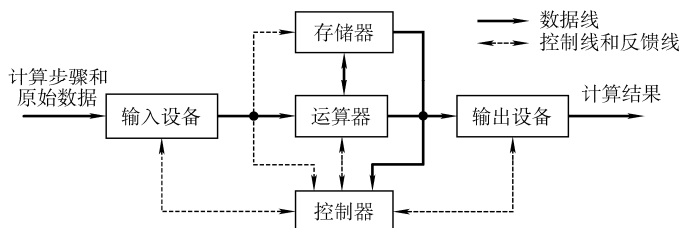


图 1.1 典型的冯·诺依曼计算机结构

2. 现代计算机的组织结构

在微处理器问世之前，运算器和控制器分离，而且存储器的容量很小，因此设计成以运算器为中心的结构，其他部件都通过运算器完成信息的传递，如图 1.1 所示。

而随着微电子技术的进步,同时计算机需要处理、加工的信息量也与日俱增,大量 I/O 设备的速度和 CPU 的速度差距悬殊,因此以运算器为中心的结构不能够满足计算机发展的要求。现代计算机已发展为以存储器为中心,使 I/O 操作尽可能地绕过 CPU,直接在 I/O 设备和存储器之间完成,以提高系统的整体运行效率,其结构如图 1.2 所示。

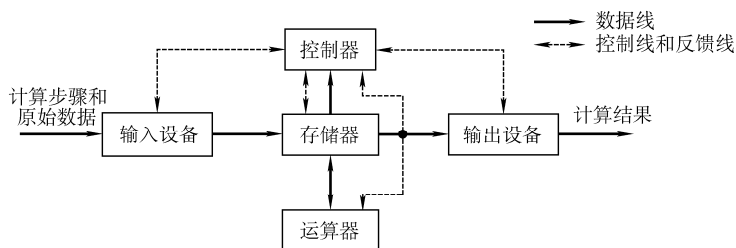


图 1.2 以存储器为中心的计算机结构

目前绝大多数现代计算机仍遵循冯·诺依曼的存储程序的设计思想。

3. 计算机的功能部件

传统冯·诺依曼计算机和现代计算机的结构虽然有所不同,但功能部件是一致的,它们的功能部件包括如下几种。

(1) 输入设备

输入设备的主要功能是将程序和数据以机器所能识别和接受的信息形式输入计算机。最常用也最基本的输入设备是键盘,此外还有鼠标、扫描仪、摄像机等。

(2) 输出设备

输出设备的任务是将计算机处理的结果以人们所能接受的形式或其他系统所要求的信息形式输出。最常用、最基本的输出设备是显示器、打印机。计算机的输入/输出设备(简称 I/O 设备)是计算机与外界联系的桥梁,是计算机中不可缺少的重要组成部分。

(3) 存储器

存储器是计算机的存储部件,用来存放程序和数据。

存储器分为主存储器(简称主存,也称内存储器)和辅助存储器(简称辅存,也称外存储器)。CPU 能够直接访问的存储器是主存储器。辅助存储器用于帮助主存储器记忆更多的信息,辅助存储器中的信息必须调入主存后,才能为 CPU 所访问。

主存储器由许多存储单元组成,每个存储单元包含若干存储元件,每个存储元件存储一位二进制代码“0”或“1”。因此存储单元可存储一串二进制代码,称这串代码为存储字,称这串代码的位数为存储字长,存储字长可以是 1B(8bit)或是字节的偶数倍。

主存储器的工作方式是按存储单元的地址进行存取,这种存取方式称为按地址存取方式(相联存储器是按内容访问的)。

主存储器的最基本组成如图 1.3 所示。存储体存放二进制信息,地址寄存器(MAR)存放访存地址,经过地址译码后找到所选的存储单元。数据寄存器(MDR)用于暂存要从存储器中读或写的信息,时序控制逻辑用于产生存储器操作所需的各种时序信号。

MAR 用于寻址,其位数对应着存储单元的个数,如 MAR 为 10 位,则有 $2^{10} = 1024$ 个存储单元,记为 1K。MAR

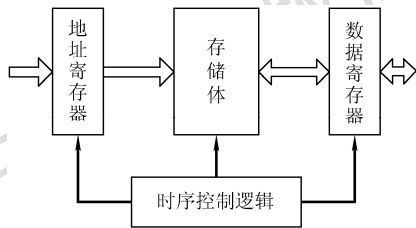


图 1.3 主存储器逻辑图

MDR 的位数和存储字长相等,一般为字节的二次幂的整数倍。

(4) 运算器

运算器的核心是算术逻辑单元 (Arithmetic and Logical Unit, ALU)。运算器包含若干通用寄存器, 用于暂存操作数和中间结果, 如累加器 (ACC)、乘商寄存器 (MQ)、操作数寄存器 (X)、变址寄存器 (IX)、基址寄存器 (BR) 等, 其中前 3 个寄存器是必须有的。

(5) 控制器

PC 用来存放当前欲执行指令的地址，可以自动加 1 以形成下一条指令的地址，它与主存的 MAR 之间有一条直接通路。

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。

图 1.4 所示为冯·诺依曼结构的模型机。CPU 包含 ALU、通用寄存器组 GPRs、标志寄存器、控制器、指令寄存器 IR、程序计数器 PC、存储器地址寄存器 MAR 和存储器数据寄存器 MDR。图中从控制器送出的虚线就是控制信号，可以控制如何修改 PC 以得到下一条指令的地址，可以控制 ALU 执行什么运算，可以控制主存是进行读操作还是写操作（读/写控制信号）。

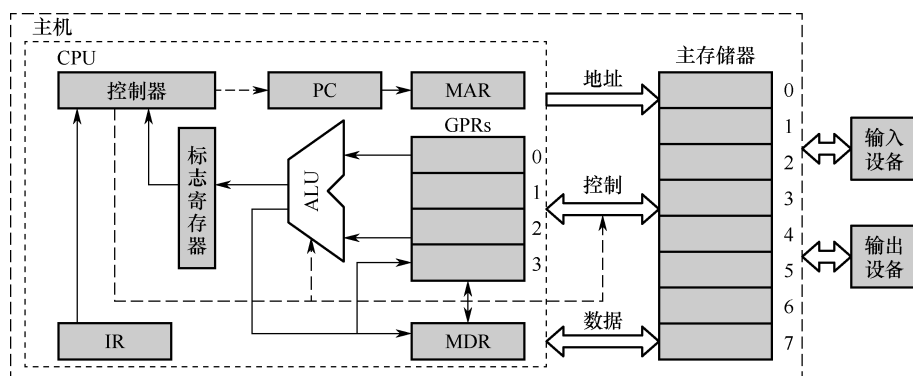


图 1.4 冯·诺依曼结构的模型机

CPU 和主存之间通过一组总线相连，总线中有地址、控制和数据 3 组信号线。MAR 中的地址信息会直接送到地址线上，用于指向读/写操作的主存存储单元；控制线中有读/写信号线，指出数据是从 CPU 写入主存还是从主存读出到 CPU，根据是读操作还是写操作来控制将 MDR 中的数据是直接送到数据线上还是将数据线上的数据接收到 MDR 中。

1.2.3 计算机软件的分类

1. 系统软件和应用软件

软件按其功能分类，可分为系统软件和应用软件。

系统软件是一组保证计算机系统高效、正确运行的基础软件，通常作为系统资源提供给用户使用。系统软件主要有操作系统（OS）、数据库管理系统（DBMS）、语言处理程序、分布式软件系统、网络软件系统、标准库程序、服务性程序等。

应用软件是指用户为解决某个应用领域中的各类问题而编制的程序，如各种科学计算类程序、工程设计类程序、数据统计与处理程序等。

注意：数据库管理系统（DBMS）和数据库系统（DBS）是有区别的。DBMS 是位于用户和操作系统之间的一层数据管理软件，是系统软件；而 DBS 是指计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统、数据库管理员（DBA）和应用系统构成。

2. 三个级别的语言

- 1) 机器语言。又称二进制代码语言，需要编程人员记忆每条指令的二进制编码。机器语言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。使用汇编语言编辑的程序，必须经过一个称为汇编程序的系统软件的翻译，将其转换为计算机的机器语言后，才能在计算机的硬件系统上执行。
- 3) 高级语言。高级语言（如 C、C++、Java 等）是为方便程序设计人员写出解决问题的处理方案和解题过程的程序。通常高级语言需要经过编译程序编译成汇编语言程序，然后经过汇编操作得到机器语言程序，或直接由高级语言程序翻译成机器语言程序。

1.2.4 计算机的工作过程

计算机的工作过程分为以下三个步骤：

- 1) 把程序和数据装入主存储器。
- 2) 将源程序转换成可执行文件。
- 3) 从可执行文件的首地址开始逐条执行指令。

1. 从源程序到可执行文件

在计算机中编写的 C 语言程序，都必须被转换为一系列的低级机器指令，这些指令按照一种称为可执行目标文件的格式打好包，并以二进制磁盘文件的形式存放起来。

以 UNIX 系统中的 GCC 编译器程序为例，读取源程序文件 `hello.c`，并把它翻译成一个可执行目标文件 `hello`，整个翻译过程可分为 4 个阶段完成，如图 1.5 所示。

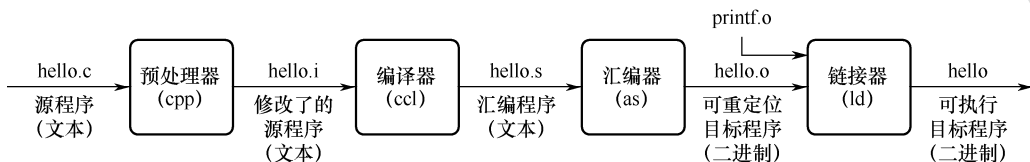


图 1.5 源程序转换为可执行文件的过程

1) 预处理阶段：预处理器（cpp）对源程序中以字符#开头的命令进行处理，例如将#include命令后面的.h 文件内容插入程序文件。输出结果是一个以.i 为扩展名的源文件 `hello.i`。

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以一种文本格式描述了一条低级机器语言指令。

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称为可重定位目标文件的 `hello.o`, 它是一种二进制文件, 因此在文本编辑器中打开它会显示乱码。

4) 链接阶段: 链接器 (ld) 将多个可重定位目标文件和标准库函数合并为一个可执行目标文件, 或简称可执行文件。本例中, 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, 生成可执行文件 `hello`。最终生成的可执行文件被保存在磁盘上。

2. 指令执行过程的描述

程序中第一条指令的地址置于 PC 中，根据 PC 取出第一条指令，经过译码、执行步骤等，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。

指令的执行过程在第 5 章中详细描述。下面以取数指令（即将指令地址码指示的存储单元中的操作数取出后送至运算器的 ACC 中）为例进行说明，其信息流程如下：

1) 取指令: $PC \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow IR$

根据 PC 取指令到 IR。将 PC 的内容送 MAR，MAR 中的内容直接送地址线，同时控制器将读信号送读/写信号线，主存根据地址线上的地址和读信号，从指定存储单元读出指令，送到数据线上，MDR 从数据线接收指令信息，并传送到 IR 中。

2) 分析指令: $OP(IR) \rightarrow CU$

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。在本例中, IR 中是取数指令, 因此读控制信号被送到总线的控制线上。

3) 执行指令: $Ad(IR) \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow ACC$

取数操作。将 IR 中指令的地址码送 MAR，MAR 中的内容送地址线，同时控制器将读信号送读/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。

此外,每取完一条指令,还须为取下一条指令做准备,形成下一条指令的地址,即 $(PC)+1 \rightarrow PC$ 。

注意：(PC)指程序计数器 PC 中存放的内容。PC→MAR 应理解为(PC)→MAR，即程序计数器中的值经过数据通路送到 MAR，也表示数据通路时括号可省略（因为只是表示数据流经的途径，而不强调数据本身的流动）。但运算时括号不能省略，即(PC)+1→PC 不能写为 PC+1→PC。当题目中(PC)→MAR 的括号未省略时，考生最好也不要省略。

1.2.5 计算机系统的多级层次结构

现代计算机是一个硬件与软件组成的综合体。由于面对的应用范围越来越广,因此必须有复杂的系统软件和硬件的支持。由于软/硬件的设计者和使用者都从不同的角度并用不同的语言来对待同一个计算机系统,因此他们各自看到的计算机系统的属性及对计算机系统提出的要求也就不同。

计算机系统的多级层次结构的作用，就是针对上述情况，根据从各种角度所看到的机器之间的有机联系，来分清彼此之间的界面，明确各自的功能，以便构成合理、高效的计算机系统。

关于计算机系统层次结构的分层方式,目前尚无统一的标准,这里采用如图 1.6 所示的层次结构。

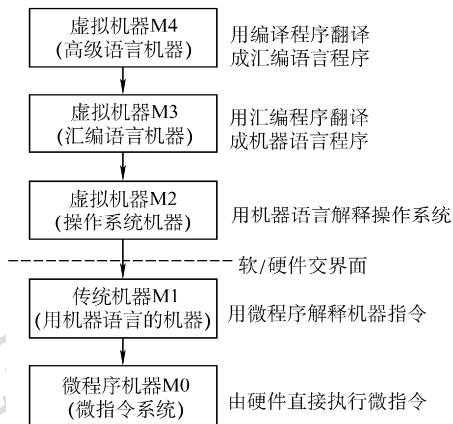


图 1.6 计算机系统的多级层次结构

第1级是微程序机器层，这是一个实在的硬件层，它由机器硬件直接执行微指令。

第2级是传统机器语言层，它也是一个实际的机器层，由微程序解释机器指令系统。

第3级是操作系统层，它由操作系统程序实现。操作系统程序是由机器指令和广义指令组成的，这些广义指令是为了扩展机器功能而设置的，是由操作系统定义和解释的软件指令，所以这一层也称混合层。

第4级是汇编语言层，它为用户提供一种符号化的语言，借此可编写汇编语言源程序。这一层由汇编程序支持和执行。

第5级是高级语言层，它是面向用户的，是为方便用户编写应用程序而设置的。该层由各种高级语言编译程序支持和执行。

在高级语言层之上，还可以有应用层，它由解决实际问题 and 应用问题的处理程序组成，如文字处理软件、数据库软件、多媒体处理软件和办公自动化软件等。

通常把没有配备软件的纯硬件系统称为“裸机”。第3层~第5层称为虚拟机，简单来说就是软件实现的机器。虚拟机只对该层的观察者存在，这里的分层和计算机网络的分层类似，对于某层的观察者来说，只能通过该层次的语言来了解和使用计算机，而不必关心下层是如何工作的。

层次之间的关系紧密，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，部分软件功能将由硬件来实现，因而软/硬件交界面的划分也不是绝对的。

本课程主要讨论传统机器 M1 和微程序机器 M0 的组成原理及设计思想。

1.2.6 本节习题精选

一、单项选择题

- 完整的计算机系统应包括()。
A. 运算器、存储器、控制器
B. 外部设备和主机
C. 主机和应用程序
D. 配套的硬件设备和软件系统
- 冯·诺依曼机的基本工作方式是()。
A. 控制流驱动方式
B. 多指令多数据流方式
C. 微程序控制方式
D. 数据流驱动方式
- 下列()是冯·诺依曼机工作方式的基本特点。
A. 多指令流单数据流
B. 按地址访问并顺序执行指令
C. 堆栈操作
D. 存储器按内容选择地址
- 【2009 统考真题】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. 指令和数据所在的存储单元
- 以下说法错误的是()。
A. 硬盘是外部设备
B. 软件的功能与硬件的功能在逻辑上是等效的
C. 硬件实现的功能一般比软件实现具有更高的执行速度
D. 软件的功能不能用硬件取代
- 存放欲执行指令的寄存器是()。
A. MAR
B. PC
C. MDR
D. IR





- ## 二、综合应用题

- ### 1.2.7 答案与解析

1. D

A 是计算机主机的组成部分，而 B、C 只涉及计算机系统的部分内容，都不完整。

2. A

早期的冯·诺依曼机以运算器为中心，且是单处理机，B是多处理机。冯·诺依曼机最根本

的特征是采用“存储程序”原理，基本工作方式是控制流驱动方式。

3. B

A 是不存在的机器，B 是对“存储程序”的阐述，因此正确。C 是与题干无关的选项。D 是相联存储器的特点。

4. C

虽然指令和数据都以二进制形式存放在存储器中，但 CPU 可以根据指令周期的不同阶段来区分是指令还是数据，通常在取指阶段取出的是指令，在执行阶段取出的是数据。本题容易误选 A，需要清楚的是，CPU 只有在确定取出的是指令后，才会将其操作码送去译码，因此不可能依据译码的结果来区分指令和数据。

5. D

软件和硬件具有逻辑上的等效性，硬件实现具有更高的执行速度，软件实现具有更好的灵活性。执行频繁、硬件实现代价不是很高的功能通常由硬件实现。因此选 D。

6. D

IR 存放当前欲执行的指令，PC 存放下一条指令的地址，不要将它们混淆。此外，MAR 用来存放欲访问的存储单元地址，MDR 存放从存储单元取来的数据。

7. A

在 CPU 中，PC 用来跟踪下一条要执行的指令在主存储器中的地址。

8. C

地址译码器是主存的构成部分，不属于 CPU。地址寄存器虽然一般属于主存，但现代计算机中绝大多数 CPU 内集成了地址寄存器。

9. A

地址寄存器（MAR）存放访存地址，因此位数与地址码长度相同。数据寄存器（MDR）用于暂存要从存储器中读或写的信息，因此位数与存储字长相同。

10. D

运算器的核心部分是算术逻辑运算单元（ALU）。地址寄存器位于 CPU 内，但并未集成到运算器与控制器中。地址寄存器用来保存当前 CPU 所访问的内存单元的地址。由于内存和 CPU 之间存在着操作速度上的差别，所以必须使用地址寄存器来保持地址信息，直到内存的读/写操作完成为止。

11. C

寄存器在 CPU 内部，速度最快。Cache 采用高速的 SRAM 制作，而内存常用 DRAM 制作，其速度较 Cache 慢。本题也可根据存储器层次结构的速度关系得出答案。

12. C

8 位计算机表明计算机字长为 8 位，即一次可以处理 8 位的数据；而 16 位表示地址码的长度，因此该机器有 $2^{16} = 65536$ 个地址空间。

13. B

计算机只能从主存中取指令与操作数，不能直接与外存交换数据。

14. D

操作系统属于大型系统软件；编译程序属于语言处理程序；连接程序属于服务性程序，因此选 D。

15. C

编译程序是先完整编译后运行的程序，如 C、C++ 等；解释程序是一句一句翻译且边翻译边

执行的程序，如 JavaScript、Python 等。由于解释程序要边翻译成机器语言边执行，因此一般速度较编译程序慢。为增加对该过程的理解，附 C 语言编译链接的过程：

源程序(.c) $\xrightarrow{\text{C编译器}}$ 汇编源程序 $\xrightarrow{\text{汇编程序}}$ 目标程序 $\xrightarrow{\text{链接程序}}$ 可执行程序

16. C

翻译程序是指把高级语言源程序转换成机器语言程序（目标代码）的软件。翻译程序有两种：一种是编译程序，它将高级语言源程序一次全部翻译成目标程序，每次执行程序时，只需执行目标程序，因此只要源程序不变，就无须重新编译。另一种是解释程序，它将源程序的一条语句翻译成对应的机器目标代码，并立即执行，然后翻译下一条源程序语句并执行，直至所有源程序语句全部被翻译并执行完。所以解释程序的执行过程是翻译一句执行一句，并且不会生成目标程序。汇编程序也是一种语言翻译程序，它把汇编语言源程序翻译为机器语言程序。汇编语言是一种面向机器的低级语言，是机器语言的符号表示，与机器语言一一对应。

17. C

寄存器的设置对汇编语言不透明，汇编程序员要对寄存器进行直接操作。全面代表计算机性能的是实际软件的运行情况。软件和硬件在逻辑上是等效的，但不是等价的。向后兼容指的是时间上向后兼容，即新机器兼容使用以前机器的指令系统。

18. B

CPU 由运算器和控制器两个部件组成，而运算器和控制器中都含有寄存器。存储器是一个独立的部件。

19. A

数据库系统是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统、应用系统、数据库管理员构成，其中数据库管理系统是系统程序。

20. C

相联存储器既可以按地址寻址又可以按内容（通常是某些字段）寻址，为与传统存储器区别，又称按内容寻址的存储器。

21. B

在计算机多层次结构中，上下层是可以分割的，且上层是下层的功能实现。此外，上层在下层的基础上实现了更加丰富的功能，仅有下层而没有上层也是可以的。

22. A

硬件能直接执行的只能是机器语言（二进制编码），汇编语言是增强机器语言的可读性和记忆性的语言，经过汇编后才能被执行。

23. D

对于 I，二进制由于只有 0 和 1 两种数值，运算规则较简单，都通过 ALU 部件转换成加法运算。对于 II，二进制只需要高电平和低电平两个状态就可表示，这样的物理器件很容易制造。对于 III，二进制与逻辑量相吻合。二进制的 0 和 1 正好与逻辑量的“真”和“假”相对应，因此用二进制数表示二值逻辑显得十分自然，采用逻辑门电路很容易实现运算。

24. C

冯·诺依曼结构计算机的功能部件包括输入设备、输出设备、存储器、运算器和控制器，程序的功能都通过中央处理器（运算器和控制器）执行指令，A 正确。指令和数据以同等地位存于存储器内，形式上无差别，只在程序执行时具有不同的含义，B 正确。指令按地址访问，数据由指令的地址码指出，除立即寻址外，数据均存放在存储器内，C 错误。在程序执行前，指令和数据需预先存放在存储器中，中央处理器可以从存储器存取代码，D 正确。

1. 解答:

计算机按照此原理应该具有 5 大功能：数据传送功能、数据存储功能、数据处理功能、操作控制功能、操作判断功能。

1.3.1 计算机的主要性能指标

1. 机器字长

注意：机器字长、指令字长和存储字长的关系（见本章常见问题和易混淆知识点4）。

2. 数据通路带宽

注意：各个子系统通过数据总线连接形成的数据传送路径称为数据通路。

3. 主存容量

例如，MAR 为 16 位，表示 $2^{16}=65536$ ，即此存储体内有 65536 个存储单元（可称为 64K 内存，1K=1024），若 MDR 为 32 位，表示存储容量为 64K×32 位。

4. 运算速度

(1) 吞吐量和响应时间。

- **吞吐量。**指系统在单位时间内处理请求的数量。它取决于信息能多快地输入内存，CPU 能多快地取指令，数据能多快地从内存取出或存入，以及所得结果能多快地从内存送给一台外部设备。几乎每步都关系到主存，因此系统吞吐量主要取决于主存的存取周期。
- **响应时间。**指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等待时间。通常包括 CPU 时间（运行一个程序所花费的时间）与等待时间（用于磁盘访问、存储器访问、I/O 操作、操作系统开销等的时间）。

(2) 主频和 CPU 时钟周期。

- **CPU 时钟周期。**通常为节拍脉冲或 T 周期，即主频的倒数，它是 CPU 中最小的时间单位，每个动作至少需要 1 个时钟周期。
- **主频（CPU 时钟频率）。**机器内部主时钟的频率，是衡量机器速度的重要参数。对于同一个型号的计算机，其主频越高，完成指令的一个执行步骤所用的时间越短，执行

指令的速度越快。例如，常用 CPU 的主频有 1.8GHz、2.4GHz、2.8GHz 等。

注意：CPU 时钟周期 = 1/主频，主频通常以 Hz（赫兹）为单位，1Hz 表示每秒 1 次。

(3) CPI (Clock cycle Per Instruction)，即执行一条指令所需的时钟周期数。

(4) CPU 执行时间，指运行一个程序所花费的时间。

$\text{CPU 执行时间} = \text{CPU 时钟周期数} / \text{主频} = (\text{指令条数} \times \text{CPI}) / \text{主频}$

上式表明，CPU 的性能（CPU 执行时间）取决于三个要素：① 主频（时钟频率）；② 每条指令执行所用的时钟周期数（CPI）；③ 指令条数。

主频、CPI 和指令条数是相互制约的。例如，更改指令集可以减少程序所含指令的条数，但同时可能引起 CPU 结构的调整，从而可能会增加时钟周期的宽度（降低主频）。有关主频、CPI 和指令条数的相互制约关系，相信读者在学完指令系统、数据通路设计后会有更深刻的认识。

(5) MIPS、MFLOPS、GFLOPS 和 TFLOPS。

- MIPS (Million Instructions Per Second)，即每秒执行多少百万条指令。 $\text{MIPS} = \text{指令条数} / (\text{执行时间} \times 10^6) = \text{主频} / \text{CPI}$ 。
- MFLOPS (Mega Floating-point Operations Per Second)，即每秒执行多少百万次浮点运算。 $\text{MFLOPS} = \text{浮点操作次数} / (\text{执行时间} \times 10^6)$ 。
- GFLOPS (Giga Floating-point Operations Per Second)，即每秒执行多少十亿次浮点运算。 $\text{GFLOPS} = \text{浮点操作次数} / (\text{执行时间} \times 10^9)$ 。
- TFLOPS (Tera Floating-point Operations Per Second)，即每秒执行多少万亿次浮点运算。 $\text{TFLOPS} = \text{浮点操作次数} / (\text{执行时间} \times 10^{12})$ 。

注意：在描述存储容量、文件大小等时，K、M、G、T 通常用 2 的幂次表示，如 1Kb = 2^{10} b；在描述速率、频率等时，k、M、G、T 通常用 10 的幂次表示，如 1kb/s = 10^3 b/s。通常前者用大写的 K，后者用小写的 k，但其他前缀均为大写，表示的含义取决于所用的场景。

1.3.2 几个专业术语

- 1) 系列机。具有基本相同的体系结构，使用相同基本指令系统的多个不同型号的计算机组成的一个产品系列。
- 2) 兼容。指计算机软件或硬件的通用性，即使用或运行在某个型号的计算机系统上的硬件/软件也能应用于另一个型号的计算机系统时，称这两台计算机在硬件或软件上存在兼容性。
- 3) 软件可移植性。指把使用在某个系列计算机中的软件直接或进行很少的修改就能运行在另一个系列计算机中的可能性。
- 4) 固件。将程序固定在 ROM 中组成的部件称为固件。固件是一种具有软件特性的硬件，固件的性能指标介于硬件与软件之间，吸收了软/硬件各自的优点，其执行速度快于软件，灵活性优于硬件，是软/硬件结合的产物。例如，目前操作系统已实现了部分固化（把软件永恒地存储于只读存储器中）。

1.3.3 本节习题精选

一、单项选择题

1. 【2010 统考真题】下列选项中，能缩短程序执行时间的措施是（ ）。
 - 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

- 

- 

- 

- D. I, III, IV

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

D. 600

- D. 70s

- 17 •

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

20. 【2017 统考真题】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 1.2GHz。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

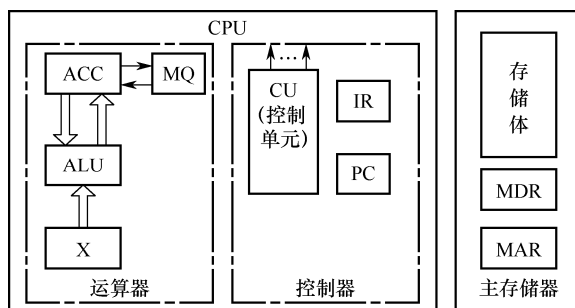
B. 0.625

C. 1.6

D. 2.5

二、综合应用题

1. 设主存储器容量为 $64K \times 32$ 位，且指令字长、存储字长、机器字长三者相等。写出如下图所示各寄存器的位数，并指出哪些寄存器之间有信息通路。



2. 用一台 40MHz 的处理器执行标准测试程序，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 速率和程序的执行时间 (I 为程序的指令条数)。

指令类型	CPI	指令混合比	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%

3. 微机 A 和 B 是采用不同主频的 CPU 芯片，片内逻辑电路完全相同。

- 1) 若 A 机的 CPU 主频为 8MHz，B 机为 12MHz，则 A 机的 CPU 时钟周期为多少？
- 2) 若 A 机的平均指令执行速度为 0.4MIPS，则 A 机的平均指令周期为多少？
- 3) B 机的平均指令执行速度为多少？

4. 某台计算机只有 Load/Store 指令能对存储器进行读/写操作, 其他指令只对寄存器进行操作。根据程序跟踪试验结果, 已知每条指令所占的比例及 CPI 数如下表所示。

指令类型	指令所占比例	CPI	指令类型	指令所占比例	CPI
算术逻辑指令	43%	1	Store 指令	12%	2
Load 指令	21%	2	转移指令	24%	2

求上述情况的平均 CPI。

假设程序由 M 条指令组成。算术逻辑运算中 25% 的指令的两个操作数中的一个已在寄存器中, 另一个必须在算术逻辑指令执行前用 Load 指令从存储器中取到寄存器中。因此有人建议增加另一种算术逻辑指令, 其特点是一个操作数取自寄存器, 另一个操作数取自存储器, 即寄存器-存储器类型, 假设这种指令的 CPI 等于 2。同时, 转移指令的 CPI 变为 3。求新指令系统的平均 CPI。

1.3.4 答案与解析

一、单项选择题

1. D

CPU 时钟频率 (主频) 越高, 完成指令的一个执行步骤所用的时间就越短, 执行指令的速度就越快, I 正确。数据通路的功能是实现 CPU 内部的运算器和寄存器及寄存器之间的数据交换, 优化数据通路结构, 可以有效提高计算机系统的吞吐量, 从而加快程序的执行, II 正确。计算机程序需要先转化成机器指令序列才能最终得到执行, 通过对程序进行编译优化可以得到更优的指令序列, 从而使得程序的执行时间也越短, III 正确。

2. D

MIPS 是每秒执行多少百万条指令, 适用于衡量标量机的性能。CPI 是平均每条指令的时钟周期数。IPC 是 CPI 的倒数, 即每个时钟周期执行的指令数。MFLOPS 是每秒执行多少百万条浮点数运算, 用来描述浮点数运算速度, 适用于衡量向量机的性能。

3. D

CPU 主频指 CPU 的时钟脉冲频率, CPI 是执行一条指令平均使用的 CPU 时钟数。

4. B

存储体由许多存储单元组成, 每个存储单元又包含若干存储元件, 每个存储元件能寄存一位二进制代码 “0” 或 “1”。可见, 一个存储单元可存储一串二进制代码, 称这串二进制代码为一个存储字, 称这串二进制代码的位数为存储字长。

5. C

计算机中一个字的长度可以是 16、32、64 位等, 一般是 8 的整数倍, 不一定是 32 位。

6. C

机器字长、指令字长和存储字长, 三者数值上可以相等也可以不等, 视不同机器而定。一个存储单元中的二进制代码的位数称为存储字长。存储字长等于 MDR 的位数, 而数据字长是数据总线一次能并行传送信息的位数, 它可以不等于 MDR 的位数。

7. В

计算机的位数，即机器字长，也就是计算机一次能处理的二进制数的长度。要注意的是，操作系统的位数是操作系统可寻址的位数，它与机器字长不同。一般情况下可通过寄存器的位数来判断机器字长。

8. C

MFLOPS 是指每秒执行多少百万次浮点运算, 该参数用来描述计算机的浮点运算性能, 而用于科学计算的计算机主要评估浮点运算的性能。

9. C

机器字长是计算机内部一次可以处理的二进制数的位数，因此该计算机一次可处理 $4 \times 8 = 32$ 位的二进制代码。

10. B

汇编程序员可以通过指定待执行指令的地址来设置 PC 的值，状态寄存器、通用寄存器只有为汇编程序员可见，才能实现编程，而 IR、MAR、MDR 是 CPU 的内部工作寄存器，对程序员均不可见。

11. A

时钟周期即 CPU 频率的倒数，是最基本的时间单位，其余选项均大于时钟周期。另外，CPU 周期又称机器周期，它由多个时钟周期组成。

12. A

CPI 是执行一条指令所需的时钟周期数，系统结构、指令集、计算机组织都会影响 CPI，而时钟频率并不会影响 CPI，但可加快指令的执行速度。例如，执行一条指令需要 10 个时钟周期，则一台主频为 1GHz 的 CPU，执行这条指令要比一台主频为 100MHz 的 CPU 快。

13. B

主频、主存容量和指令系统（间接影响 CPI）并不是综合性能的体现。吞吐率指系统在单位时间内处理请求的数量，是评价计算机系统性能的综合参数。

14. D

提高 CPU 主频、扩大主存容量对性能的提升是有限度的。采用并行技术是实现高性能计算的重要途径, 现今超级计算机均采用多处理器来增强并行处理能力。

15. C

兼容指计算机软件或硬件的通用性，因此选项 A、D 错。选项 B 中，它们在任何计算机间可以通用，错误。选项 C 中，兼容通常在同一系列的不同型号计算机间，正确。

16. C

会计电算化属于计算机数据处理方面的应用, I 错误。II 显然正确。计算机“运算速度”指标的含义是每秒能执行多少条指令, III 错误。这样集成的芯片称为 CPU, IV 错误。

17. C

基准程序的 $CPI = 2 \times 0.5 + 3 \times 0.2 + 4 \times 0.1 + 5 \times 0.2 = 3$ 。计算机的主频为 1.2GHz，即 1200MHz，因此该机器的 $MIPS = 1200/3 = 400$ 。

18. D

程序 A 的运行时间为 100s, 除去 CPU 时间 90s, 剩余 10s 为 I/O 时间。CPU 提速后运行基准程序 A 所耗费的时间是 $T = 90/1.5 + 10 = 70\text{s}$ 。

误区：CPU 速度提高 50%，则 CPU 时间减少一半，而误选 A。

19. D

假设原来的指令条数为 x ，则原 CPI 为 $20f/x$ (f 为 CPU 的时钟频率)，经过编译优化后，指令条数减少到原来的 70%，即指令条数为 $0.7x$ ，而 CPI 增加到原来的 1.2 倍，即 $24f/x$ ，则现在 P 在 M 上的执行时间就为：(指令条数×CPI)/ $f = (0.7x \times 24 \times f/x)/f = 24 \times 0.7 = 16.8s$ ，选 D。

20. C

运行时间 = 指令数×CPI/主频。M1 的时间 = 指令数×2/1.5，M2 的时间 = 指令数×1/1.2，两者之比为 $(2/1.5):(1/1.2) = 1.6$ 。因此选 C。

二、综合应用题

1. 解答：

由于主存容量为 $64K \times 32$ 位，因 $2^{16} = 64K$ ，则地址总线宽度为 16 位，32 位表示数据总线宽度，因此 MAR 为 16 位，PC 为 16 位，MDR 为 32 位。

因指令字长 = 存储字长 = 机器字长，则 IR、ACC、MQ、X 均为 32 位。

寄存器之间的信息通路有：

PC→MAR

Ad(IR)→MAR

MDR→IR

取数：MDR→ACC，存数：ACC→MDR

MDR→X

2. 解答：

CPI 即执行一条指令所需的时钟周期数。本标准测试程序共包含 4 种指令，则 CPI 就是这 4 种指令的数学期望，即

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

MIPS 即每秒执行的百万条指令数。已知处理器时钟频率为 40MHz，即每秒包含 40M 个时钟周期，因此

$$MIPS = 40/CPI = 40/2.24 = 17.9$$

程序的执行时间 $T = CPI \times T_{IC} \times I$ ，其中 T_{IC} 是一个 CPU 时钟的时间长度，是 CPU 时钟频率 f 的倒数，因此有

$$T = CPI \times T_{IC} \times I = CPI \times (1/f) \times I = 5.6 \times 10^{-8} \times I \text{ 秒}$$

本题中的 I 对于解题应无作用，程序的执行时间应是指令的期望即 CPI 乘以时钟的时间长度，即 $T = CPI \times T_{IC}$ 。

3. 解答：

1) A 机的 CPU 主频为 8MHz，所以 A 机的 CPU 时钟周期 = $1/8MHz = 0.125\mu s$ 。

2) A 机的平均指令周期 = $1/0.4MIPS = 2.5\mu s$ 。

3) A 机平均每条指令的时钟周期数 = $2.5\mu s / 0.125\mu s = 20$ 。

因微机 A 和 B 的片内逻辑电路完全相同，所以 B 机平均每条指令的时钟周期数也为 20。

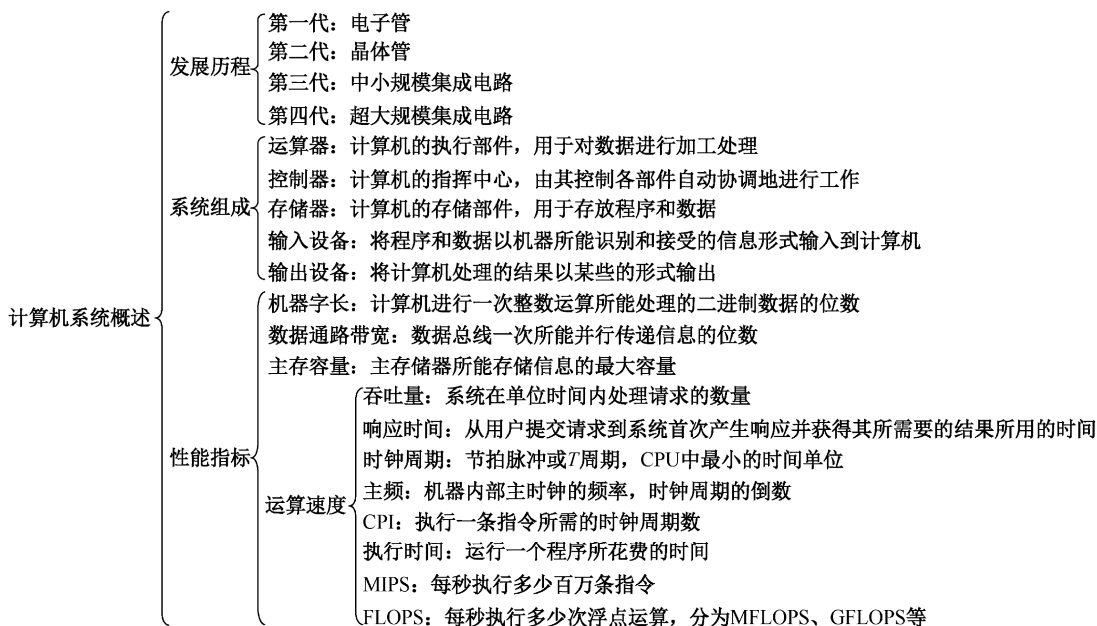
由于 B 机的 CPU 主频为 12MHz，所以 B 机的 CPU 时钟周期 = $1/12MHz = 1/12\mu s$ 。

B 机的平均指令周期 = $20 \times (1/12) = 5/3\mu s$ 。

B 机的平均指令执行速度 = $1/(5/3)\mu s = 0.6MIPS$ 。

另解：B 机的平均指令执行速度 = A 机的平均指令执行速度 $\times (12/8) = 0.4MIPS \times (12/8) = 0.6MIPS$ 。

本章的知识架构图如下：



1.5 常见问题和易混淆知识点

1. 同一个功能既可以由软件实现又可以由硬件实现吗？

软件和硬件是两种完全不同的形态，硬件是实体，是物质基础；软件是一种信息，看不见、摸不到。但在逻辑功能上，软件和硬件是等效的。因此，在计算机系统中，许多功能既可以由硬件直接实现，又可以在硬件的配合下由软件实现。

例如，乘法运算既可用专门的乘法器（主要由加法器和移位器组成）实现，也可用乘法子程序（主要由加法指令和移位指令等组成）来实现。

2. 翻译程序、汇编程序、编译程序、解释程序的区别和联系是什么？

翻译程序是指把高级语言源程序翻译成机器语言程序（目标代码）的软件。

翻译程序有两种：一种是编译程序，它将高级语言源程序一次全部翻译成目标程序，每次执行程序时，只需执行目标程序，因此只要源程序不变，就无须重新翻译，请注意同一种高级语言在不同体系结构下，编译成目标程序是不一样的，目标程序与体系结构相关，但仍不是计算机硬件能够直接执行的程序。另一种是解释程序，它将源程序的一条语句翻译成对应的机器目标代码，并立即执行，然后翻译下一条源程序语句并执行，直至所有源程序语句全部被翻译并执行完。所以解释程序的执行过程是翻译一句执行一句，并且不会生成目标程序。

汇编程序也是一种语言翻译程序，它把汇编语言源程序翻译为机器语言程序。汇编语言是一种面向机器的低级语言，是机器语言的符号表示，与机器语言一一对应。

编译程序与汇编程序的区别：若源语言是诸如 C、C++、Java 等“高级语言”，而目标语言是诸如汇编语言或机器语言之类的“低级语言”，则这样的—个翻译程序称为编译程序。若源语言是汇编语言，而目标语言是机器语言，则这样的—个翻译程序称为汇编程序。

3. 什么是透明性？透明是指什么都能看见吗？

在计算机领域中，站在某类用户的角度，若感觉不到某个事物或属性的存在，即“看”不到某个事物或属性，则称为“对该用户而言，某个事物或属性是透明的”。这与日常生活中的“透明”概念（公开、看得见）正好相反。

例如，对于高级语言程序员来说，浮点数格式、乘法指令等这些指令的格式、数据如何在运算器中运算等都是透明的；而对于机器语言或汇编语言程序员来说，指令的格式、机器结构、数据格式等则不是透明的。

在 CPU 中，IR、MAR 和 MDR 对各类程序员都是透明的。

4. 机器字长、指令字长、存储字长的区别和联系是什么？

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：一个指令字中包含的二进制代码的位数。

存储字长：一个存储单元存储的二进制代码的长度。

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，若指令字长等于存储字长的 2 倍，则需要 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；若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，因此访问一次主存便可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，但它们必须都是字节的整数倍。

请注意 64 位操作系统是指特别为 64 位架构的计算机而设计的操作系统，它能够利用 64 位处理器的优势。但 64 位机器既可以使用 64 位操作系统，又可以使用 32 位操作系统。而 32 位处理器是无法使用 64 位操作系统的。

5. 计算机体系结构和计算机组成的区别和联系是什么？

计算机体系结构是指机器语言或汇编语言程序员所看得到的传统机器的属性，包括指令集、数据类型、存储器寻址技术等，大都属于抽象的属性。

计算机组成是指如何实现计算机体系结构所体现的属性，它包含对许多对程序员来说透明的硬件细节。例如，指令系统属于结构的问题，但指令的实现即如何取指令、分析指令、取操作数、如何运算等都属于组成的问题。因此，当两台机器指令系统相同时，只能认为它们具有相同的结构，至于这两台机器如何实现其指令，完全可以不同，即可以认为它们的组成方式是不同的。例如，一台机器是否具备乘法指令是一个结构的问题，但实现乘法指令采用什么方式则是一个组成的问题。

许多计算机厂商提供一系列体系结构相同的计算机，而它们的组成却有相当大的差别，即使是同一系列的不同型号机器，其性能和价格差异也很大。例如，IBM System/370 结构就包含了多种价位和性能的机型。

6. 基准程序执行得越快说明机器的性能越好吗？

一般情况下，基准测试程序能够反映机器性能的好坏。但是，由于基准程序中的语句存在频度的差异，因此运行结果并不能完全说明问题。