

课程介绍+第一章 概论

time: 2025.09

王冬青

第一周一课后思考：课程介绍+第一章 概论（1）

导言：为什么要学习“计算机组成”？

1. 学习途径：学习过程与已学编程知识相融合，做到不仅知其然还知其所以然。
2. 什么是 ISA？了解它和 API 和 ABI 的关系。

教材：不要只看封面，注意作者：张晨曦等老师编著的（新旧版本都可以）

辅助教材：

1. 入门级教材：《程序是怎样跑起来的》（群共享）
2. 深入级教材：《计算机组成与设计：硬件 / 软件接口》、《深入理解计算机系统》（CSAPP）、《计算机组成：结构化方法》等

推荐课程：

1. 北大 陆俊林 [计算机组成](#)（入门级：浅显易懂）
2. 南大 袁春风 [计算机系统基础](#)；（深入级：有一定难度）

课程实验 (PA)

<https://nju-projectn.github.io/ics-pa-gitbook/>

3. MIT“计算机教育中缺失的一课”
<https://missing-semester-cn.github.io/>
4. 唐朔飞 计算机组成原理

课程介绍：

1. 课程涵盖内容（自学 2-5 章数字逻辑，PPT 和自学重点见群共享“数字逻辑”文件夹）；
2. 课程考核方式：10%考勤 + 30%平时作业（提交两次）+ 60%期末考试；
3. 了解本课程（计算机组成原理）与先修课程（逻辑代数、数字电路）及后续课程（计算机系统结构、操作系统、编译原理等）的关系，确定本课程的学习重点和内容。
4. 模拟量和数字量的关系和区别，数电的优势，模电在哪些地方不可替代？
5. 逻辑分析和逻辑综合的关系。
6. 你计划如何学好这门课？

第一章 概述（1）

1. 什么是计算机系统，日常见到的哪些产品是计算机系统？
2. 计算机硬件和软件的关系，如何理解硬件和软件在逻辑功能上是等效的？

3. 你觉得固件（firmware）是硬件范畴还是软件范畴的？固件和设备驱动（Driver）有什么不同？
4. 依据什么来确定某一功能是由硬件实现还是软件实现？普通用户是否有话语权？
5. 了解计算机系统结构和组成原理之间的关系。（下学期学习系统结构，这两门课关联度比较大）

预习：

- 1、冯诺依曼提出的**以运算器为中心的**计算机中包含哪些功能部件？分别起什么作用？有哪些总线类型？分别起什么作用？能不看资料画出结构图，描述计算机工作流程。（例如如何从外界获取信息，内部处理信息又反馈给外界的？理解计算机的本质就是**接受、处理和传输**信息）
- 2、冯诺依曼机和图灵机有什么关系？
- 3、冯诺依曼在什么背景下提出的**存储程序计算机**（也叫冯诺依曼机），它有什么特点？（可存储+可编程提高了计算机的使用效率和性能）
- 4、了解计算机的发展史，理解为什么会从运算器为中心发展到**以存储器为中心的**？

趣味视频：

【计算机的文艺复兴-冯诺依曼之外】

https://www.bilibili.com/video/BV1Uf4y1o7aj/?share_source=copy_web&vd_source=40a237f36c6fb8a047115a80be6ff80f

【计算机科学速成课 Crash Course Computer Science】

第二周一课后思考：第一章 概述（2）

1. 了解冯诺依曼机提出的历史背景（是对第一台通用电子计算机 ENIAC 的**改进**，ENIAC 的缺点是什么？），ENIAC 是冯诺依曼机吗？为什么传统冯诺依曼机器不适合 GPU、TPU 等处理器，拓展了解**存算一体**或**存内计算**等技术。
2. 理解计算机硬件组成（重点：控制器 CU、运算器 ALU、存储器（尤其是内存部分）），在理解的前提下能画出五部件结构关系图，尤其是各部件间的信号传输类型和方向（**以存储器为中心的图 1.2**）。

理解本课程的核心思想：任何一台计算机的任何一个部件都可以归类为**运算器、控制器、存储器、输入设备和输出设备**中的某一种，目前所有的计算机系统也都是基于这个基础架构来设计实现的。而所有无论高级还是低级语言撰写的计算机程序，都可以抽象为从输入设备读取输入信息到内存，通过运算器和控制器来执行存储在存储器里的程序，最终把结果输出到输出设备中。软件都是基于这样一个抽象基础框架运行的。---冯诺依曼结构的意义

3. 了解计算机中的三类总线：**数据总线**、**控制总线**、**地址总线**（图 1.2 中没标出来地址线）的作用。
4. 本课程存储器包含**寄存器（register）**、**内存（memory）**、**外存（storage）**，理解它们的相同点和不同点是什么？它们和 CPU 的数据交互方式和定位方式有什么不同？（计算机中另一种存储器--**缓存（Cache）**工作原理会在下学期系统结构学习）
5. 了解冯诺依曼提出的**存储程序原理**的特点（5 部件、二进制、存储程序、自动取指等），核心是**可存储+可编程**。
6. 理解图 1.4（重点--它是本课程抽象的模型机，它是**通用寄存器型**计算机）中各部件的关系，目前重点是运算部件和内存之间的**数据交互**，控制器部分在后面章节细讲。
说明：有的教材使用**累加器型**计算机学习，不影响对基本工作原理的理解
7. 寄存器分**通用寄存器**（GPR，主要在运算部件里，就是图 1.4 的 R0-R31）和**专用寄存器**（分布在计算机不同部位，如模型机中的 PC、AR、IR 等寄存器），通用寄存器放用户数据，配合运算单元 ALU 工作，专用寄存器有特殊用途，存放特殊意义的信息，寄存器有两个重要参数：**数量**和**位宽**。
8. 理解 CPU 和内存之间“**读-Load**”、“**写-Store**”两个动作是如何完成的？（地址、数据、控制总线的作用，数据总线位数的意义，地址总线位数的意义，“字”在不同机器里的定义，“字长”的含义详见 <https://www.bilibili.com/read/cv19069811/>）。

预习：

- 1、了解 CPU 中控制器的组成及各部件基本功能，目前仅了解即可，细节在 8,9 章学习；
- 2、了解 CPU 时钟周期、节拍、微操作等基本概念；
- 3、了解常用的几个硬件性能指标：主频（时钟周期）、运算速度、**字长**（很多资料里提的字就是这个概念）、存储容量（包括地址线位数和数据线位数的含义）、内存存取周期（访存速度，制约 **CPU 周期（机器周期）** 的主要因素）等；
- 4、计算机发展史、分类及应用当故事看看。

第三周国庆放假

第四周一课后思考：第一章 概述（3）（前半节课）

说明：为了更好理解第 6 和 7 章相关内容，提前学习或拓展了部分第 11 章内容。

- 1、理解**内存编址**和**寻址**的关系；<https://www.bilibili.com/read/cv19071875/>
- 2、**字节编址**和**字编址**各自优缺点；
- 3、掌握**内存编址**和**字节顺序**对于**内存寻址**的影响；

- 4、字节序是针对多字节数据而言的，多字节数据其内存地址是哪个字节对应的地址什么？
常用字节序中小端字节顺序和大端字节顺序的优缺点是什么？
- 5、控制器中主要包含哪些器件，目前重点了解 **IR** 和 **PC** 这两个专用寄存器的作用；
- 6、为什么说 **PC** 指向当前正在执行的指令或下一条指令都对，分别适用什么场合？
- 7、了解**微操作**是什么？了解时钟周期、节拍和微操作的关系；（细节在第 8、9 章学习）
说明：教材中的节拍的叫法严格来说不规范，详见下面补充资料。
- 8、掌握教材目前介绍的几个**硬件性能指标**：主频（时钟周期）、运算速度、**字长**（指机器字长）、存储容量（包括地址线位数和数据线位数的含义）、内存存取周期（访存速度，为什么说它是制约 **CPU 周期（机器周期）** 的主要因素，这也是 **Cache** 出现的历史背景）等；
- 9、主频越快机器性能是不是越好？
- 10、IO 系统既解决计算机与外界的信息交互问题，你用的设备中哪些属于输入设备？哪些属于输出设备？哪些既是输入又是输出设备？

第一章作业：（要提交）

第一章作业 1.1（选 5 个）、1.2、1.3（提交时间待定）。

课外尝试在 linux 下调用 **time** 查看程序运行消耗的 **real time**、**user time**、**sys time**。（在一台或多台电脑上运行看数据有什么变化，理解这些参数的含义）

补充：

<https://zhuanlan.zhihu.com/p/90829922>

【浅析】CPU 中的指令周期、CPU 周期和时钟周期

计算机中我们常常会混淆**指令周期**、**CPU 周期**和**时钟周期**，要区分这些并不难，但要想彻底弄懂这些，就得要求我们对 CPU 底层有一定了解。

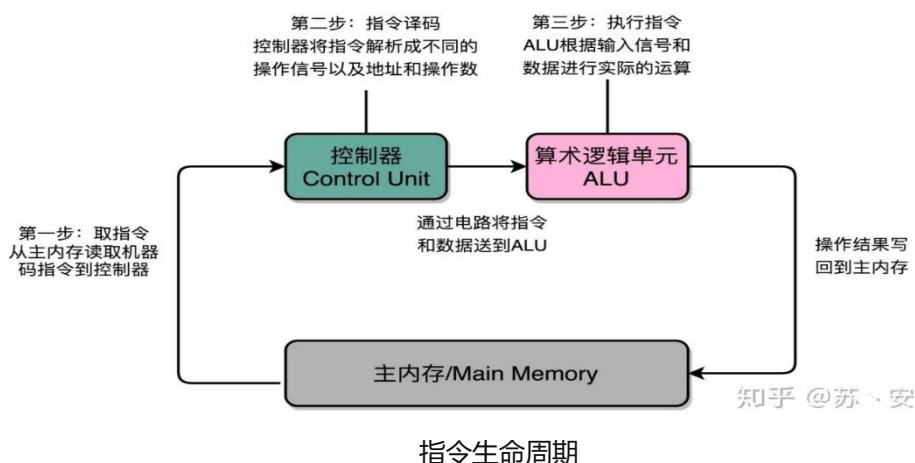
一. 指令周期

指令周期：是指计算机从取指到指令执行完毕的时间

计算机执行指令的过程可以分为以下三个步骤：

1. **Fetch（取指）**，也就是从 **PC** 寄存器里找到对应的指令地址，根据指令地址从内存里把具体的指令，加载到指令寄存器中，然后把 **PC** 寄存器自增，好在未来执行下一条指令。
2. **Decode（译码）**，也就是根据指令寄存器里面的指令，解析成要进行什么样的操作，是 **R、I、J** 中的哪一种指令，具体要操作哪些寄存器、数据或者内存地址。
3. **Execute（执行指令）**，也就是实际运行对应的 **R、I、J** 这些特定的指令，进行算术逻辑操作、数据传输或者直接的地址跳转。

在取指令的阶段，我们的指令是放在**存储器（也就是内存）**里的，实际上，通过**PC 寄存器**和**指令寄存器**取出指令的过程，是由**控制器（Control Unit）**操作的。指令的解码过程，也是由**控制器**进行的。一旦到了执行指令阶段，无论是进行算术操作、逻辑操作的 R 型指令，还是进行数据传输、条件分支的 I 型指令，都是由**算术逻辑单元（ALU）**操作的，也就是由运算器处理的。不过，如果是一个简单的无条件地址跳转，那么我们可以直接在控制器里面完成，不需要用到运算器。



二. CPU周期

CPU 周期亦称**机器周期**，在计算机中，为了便于管理，常把一条指令的执行过程划分为若干个阶段，每一阶段完成一项工作。

例如，取指令、[存储器](#)读、存储器写等，这每一项工作称为一个**基本操作**（注意：每一个基本操作都是由若干 CPU 最基本的动作组成）。完成一个基本操作所需要的时间称为机器周期。通常用内存中读取一个[指令字](#)的最短时间来规定 CPU 周期。

三. 时钟周期

时钟周期也称为[振荡周期](#)，定义为[时钟频率](#)的[倒数](#)。时钟周期是计算机中最基本的、最小的[时间单位](#)。在一个时钟周期内，CPU 仅完成一个**最基本的动作**。

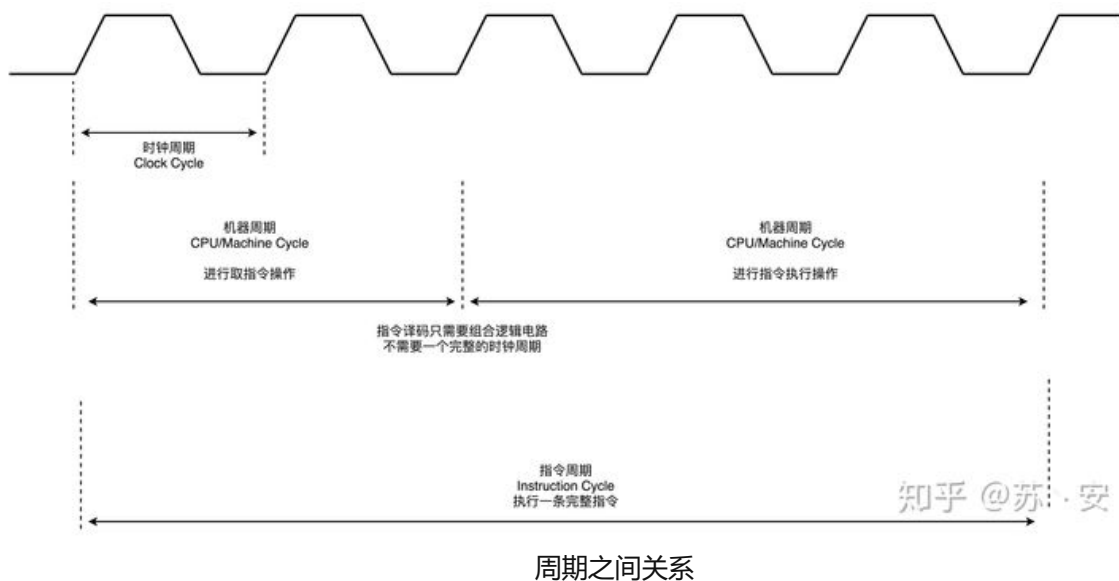
四. 周期之间的关系

[指令周期](#)（Instruction Cycle）：取出并执行一条指令的时间。

[CPU 周期](#)：一条指令执行过程被划分为若干阶段，每一阶段完成所需时间。

[时钟周期](#)（Clock Cycle）：又称振荡周期，是处理操作的最基本单位。

对于一个指令周期来说，我们取出一条指令，然后执行它，至少需要两个 CPU 周期。取出指令至少需要一个 CPU 周期，执行至少也需要一个 CPU 周期，复杂的指令则需要更多的 CPU 周期。而一个 CPU 周期是若干时钟周期之和。



所以，我们说一个指令周期，包含多个 CPU 周期，而一个 CPU 周期包含多个时钟周期。

本文是 [《深入浅出计算机组成原理》](#) 学习笔记

第一章 总结：

1、冯诺依曼的存储程序计算机特点：

- 1) 将硬件划分为 5 部分，以运算器为中心（演变到现在以存储器为中心）；
- 2) 指令和数据都一起存放在内存中，用同一套地址线 and 数据线（这是区别于哈佛结构最主要的点，哈佛结构是将指令和数据分开存放了，这个具体在 OS 课会讲）；
- 3) 可存储；（相较机械计算机）
- 4) 可编程；（相较计算器）
- 5) 二进制的引入；（相较 ENIAC）

2、硬件 = 控制单元 + 运算单元 + 存储器 + IO 设备；

- 3、控制单元功能 CU：对指令进行译码，根据译码结果向计算机其他部件发出控制信号，使各部件协同完成该指令的执行。（目前一条指令的执行可简单理解为：取指、译码、执行）；
- 4、寄存器 R：位置在 CPU 或 IO 控制芯片上（x86 的硬件端口，这部分教材没涉及），CPU 里面的 R 分通用寄存器（协助 ALU 进行运算的，操作数的来源之一）和专用寄存器（协助 CU 工作的，如 PC、IR、MAR、MDR 等）；
- 5、存储器（这里特指内存）：字节顺序、编址方式和是否数据对齐会影响数据在内存中的具体存放方式（宽度和排序）。（这部分有些书上没直接给出，但是后面教材内容会用到，不

理解可能会影响相关内容的学习)；

- 6、系统软件：没明确定义，但一般分平台类（如 OS、运行库、驱动、数据库、网络协议等）和开发工具类（编译器、汇编器、链接器、调试器等）；
- 7、应用软件：除了系统软件以外的（这个大家都熟悉）；
- 8、固件：比较特殊，课堂上没展开说，逻辑上它属于硬件范畴，形式上它是软件，这部分软件是和硬件紧密结合的，主要做板级初始化和 OS 引导的，例如 PC 的 BIOS，或嵌入式系统里的 bootloader。
- 9、计算机性能指标：
 - 1) 主频：也就是时钟周期的倒数；
时钟周期就是我们后面说的节拍；
机器周期（CPU 周期）是微操作的执行时间；（教材里没单独提，但你们实验会用到）；
一般一个机器周期等于若干个时钟周期，我们的教材为了简化学习，一个机器周期假定为 一个时钟周期（节拍），所以教材就没单独定义。
 - 2) 运算速度（吞吐量）
都说自己快，以什么为标准，第三方评测—基准测试套件，如 SPEC 等；
 - 3) 机器字长
注意不同的系统字长往往不一样，但一般和寄存器宽度和数据总线宽度一致；
字是一种数据类型，所以只要是兼容指令集，字都是一样的，例如 x86 的字都是 2B。
 - 4) 主存容量：与存储器的地址线 and 数据线都相关；
 - 5) 主存存取周期：限制了系统总线频率（目前是 KM 级）和机器周期；
其他没涉及的如 CPU 执行时间、CPI（指令周期）都会在下学期系统结构学习。
- 10、计算机发展史（建议看看，对理解技术的发展背景和技术的生命周期有帮助）；
- 11、计算机常见分类方式和应用