

课程介绍——欢迎选修计算机组成

计算机组成这门课重点讲述的是计算机的内部结构和工作原理。适合大学计算机专业2年级的水平。课程共分九讲。现在我们通过选取的一些课程画面来简要了解这门课的内容。[音乐] 第一讲的主题是计算机基本结构。这一讲从电子计算机起源开始。讲述冯诺依曼等人构建现代计算机的历史，并对照当前新型计算机的具体实现，讲解计算机内部结构的基本原理。为了便于大家理解计算机内部的抽象过程，还会通过一个关于餐馆的有趣故事，分析执行指令的过程，从而建立计算机结构的整体概念。这一讲是全课程的开篇，总体难度不大，可以比较轻松地完成学习任务。第二讲的主题是指定系统体系结构。这一讲选取了两种典型的体系结构进行对照讲解。一是X86体系结构，二是MIPS体系结构。首先会介绍这两种体系结构的基本特点和编程模型。然后分析典型的指令格式并结合程序示例进行讲解。这一讲的难度相对较大，尤其是和第一讲相比，有明显的难度提升，如果之前接触过一些汇编语言的知识会更容易学习这一讲。不过因为这一讲的多数内容相对独立，所以即使没有完全听懂也可以继续学习后面的内容。[音乐] 第三讲的主题是算术逻辑单元。这一讲会深入到计算机内部电路的基本单元。首先讲解如何从晶体管构建逻辑门，再分析常见逻辑运算的电路实现，然后重点论述加法器的设计，并探讨如何改进加法器以提高运算性能。这样我们就掌握了简单运算电路的基本原理。这一讲会涉及一些提升电路的知识，但不会很复杂。这一讲和下一讲的关系比较紧密，最好能连在一起学习。[音乐] 第四讲的主题是乘法器和除法器。这一讲会先后分析乘法和除法的运算过程，设计出适合计算机实现的运算方法。这里会用一种动态的方式生动形象的展现乘法器和除法器的运算过程，以便于我们掌握这两种相对复杂的运算部件。然后还会深入分析这两种运算的特点，并对运算部件的性能进行优化。这一讲的难度适中，与上一讲结合在一起就展现了计算机的处理器是如何实现常见的运算了。[音乐] 第五讲的主题是单周期处理器。这一讲会带大家涉及一个简单但是完整的计算机处理器。首先会分析指令的功能，选择合适的部件，构建出数据通路，然后根据每条指令的需求确定控制信号的生成电路，从而完成整个处理器的结构。处理器是计算机中最复杂的部件。这一讲自然也是本课程的核心重点，综合性强，有较高的难度。[音乐] 第六讲的主题是流水线处理器。这一讲会先用浅显易懂的生活示例展现流水线的基本原理，并探讨优化流水线的基本方法，然后结合上一讲的设计，讲解计算机处理器中使用流水线技术会遇到的问题，并分析不同的解决方案。现实中的流水线处理器是非常复杂的，但本课程只是讲解流水线技术的基本原理，所以这一讲的难度并不高。从第三讲到第六讲都在讲述计算机的核心部-处理器。这四讲构成了一个相对独立的版块。[音乐] 第七讲的主题是存储层次结构。这一讲会先介绍计算机中存储层次结构的基本概念，并分析存储器对计算机整体性能的影响，然后以内存为一个重点，讲解其内部结构和运行过程，并分析影响内存性能的要素。本讲的另一个重点是高速缓存。这里会讲解高速缓存的基本结构和运行原理，同时结合高速缓存的发展过程展现其在计算机中的重要作用。这一讲的难度适中。从这一讲开始课程的关注点扩展到了处理器之外的部件。[音乐] 第八讲的主题是中断和异常。如果计算机在运行过程中出现异常情况，该怎么办呢？这一讲先从异常处理的起源说起，并分析中断和异常这两个概念的区别。然后讲解中断处理的基本过程，并以x86体系结构为例，重点讲述内部中断的处理方式。而外部中断则会留到下一讲中。这一讲的知识本身难度并不大，但是由于此前的课程都在讲述正常情况下计算机是如何运行的，因此要用新的视角来学习这一讲。刚开始可能在理解上会有一定的难度。[音乐] 第九讲的主题是输入输出设备。现代计算机的输入输出设备非常丰富。这一讲会先介绍输入输出设备的基本结构和运行过程。然后讲解计算机内部管理输入输出设备的三种主要方式。第一种是最简单的程序控制方式。第二种是中断控制方式，需要结合上一讲的知识。第三种则是复杂的直接存储器访问方式。学到这里我们就能构建出计算机内部结构的整体框架了。这是本课程的最后一讲，总体难度不大，但需要大家结合之前讲述的内容以便联系形成一个整体。欢迎来学习这门课程，希望能够对大家了解计算机内部的事件有所帮助。

第一讲 计算机基本结构

101-电子计算机的兴起

随着人类文明的进步，人们对计算能力的需求也越来越大。

与此同时，科技的迅速发展也为强大计算能力的诞生奠定了基础。

在这样的环境下，电子计算机就应运而生了。

在这个过程中很多科学家、工程师 和各行和业的人投入了大量的智慧和辛劳。

而其中一个人 是我们在谈论到计算机的组成结构时必须提及的，

因为我们今天所使用的计算机几乎全部采用了相同的结构，

这个结构就是以他的名字命名的，而这个人

就是冯·诺依曼。

冯·诺依曼被誉为现代电子计算机之父。

实际上他主要是一位数学家 在布达佩斯大学获得了数学博士，

不久之后就来到了美国，长期在普林斯顿大学和普林斯顿高等研究院工作

二战期间冯诺依曼参与了原子弹的研究工作。

当时他遇到了一个很大的困难，是运算量太庞大。

于是他就致力于寻找具有强大运算能力的设备 其实困惑于这个问题的不仅只有他

除了原子弹， 二战在许多领域都对计算能力提出了很高的要求。

例如在战争中不可或缺的火炮，炮弹要打得准 就必须中和考虑很多的因素。

比如炮壳的仰角比如风向，比如炮弹的种类

实际上这就是些数学函数的运算，但你不能让炮兵在战场上现场去做这些计算。

必须提前把各种情况下炮弹的运行轨迹计算出来 并绘制成一张表，这就是火炮射击表。

当时美国军方要求弹道研究院每天提供六张火炮射击表， 每张表大约要计算数千个炮弹的运行轨迹。

这给弹道研究院的技术人员带来了很大的压力。

大约二百多名计算人员花两个多月才能完成一张表的计算。

即使动用当时最好的计算设备，可能也需要一个月才能完成一张表。

这与军方的要求相距甚远。

在项目不断拖延的压力下，弹道研究院的项目负责人也在寻找速度更快的计算设备。

后来他终于得到了一个好消息，他听说了一个人

叫约翰莫克利，他是一位科学家，在宾夕法尼亚大学摩尔电机工程学院工作。

他当时正和一个学生叫约翰埃克特 在研制一种高速的电子计算设备

双方沟通之后很快就立项开始了工作。

这台计算机被命名为 ENIAC， 在一九四三年正式开始设计，由莫克利任首席顾问。

当时正是硕士一年级的埃克特担任了首席工程师。

这张照片中中间的两人就是他们。

经过了两年多的艰苦设计，ENIAC 终于完成了。

1946 年的 2 月 14 日 ENIAC 在宾夕法尼亚大学完成。

ENIAC 是一个庞然大物，这张照片中几乎所有的物品都是他的组成部分。

ENIAC 共有 180000 个电子管组成。

每秒钟可以完成 5000 次的加法，功率 150 千瓦，占地 170 平方米，重量 30 吨。

总共花费了大约 50 万美元。

从性能上看每秒钟 5000 次加法

远远不如现在的个人计算机，至于说它个头很大，倒也不能这么看。

因为每一个时代性能最好的计算机都是庞然大物。

现在的高性能超级计算机也动则占地数百平方米，重量上百吨。

ENIAC 跟他们比起来也只能算是一个小个子。

不管怎样，ENIAC 在那个时候已经带来了计算能力的飞跃增长。

当然，他实际是在二战之后才完成的，并没有为二战的火炮射击表做出什么贡献。

但是他在后来的弹道计算，原子弹研究、气象预报等很多方面都发挥了重要的作用。

关于他的头衔，我们要注意一下。

普遍认为，它是世界上第一台通用电子计算机。

如果不加通用这两个字，恐怕还会有不小的争议。

因为在他之前就已经有人设计出了电子计算机。

这台计算机叫 ABC，是由美国艾尔瓦州立大学 物理系的阿塔纳索夫副教授和他的学生贝里早在 1939 年设计完成的。

更重要的是莫克利在设计 ENIAC 之前，

就了解到了阿塔纳索夫 的设计成果并多次找阿塔纳索夫探讨其设计原理。

在 ENIAC 的设计过程中不可避免的借鉴了很多 ABC 的设计思想，

但遗憾的是莫克利并没有把这件事情公之于众，这也最终导致了二十多年后的一场专利纠纷，

最后法院判决，因为设计思想来源于 ABC，所以 ENIAC 的专利被判无效，当然这是后话。

我们再回来看 ENIAC 在

ENIAC 的设计过程中，正在寻找高性能设备的冯诺依曼也加入了进来，

作为研制小组的顾问，冯诺依曼参与了很多讨论，并提出了自己的意见和建议。

其中有一点非常的重要，我们先来看左边这张照片，这是 ENIAC 原有四十块面板当中的四块。

最左边那是 ENIAC 三块函数表当中的一块。

我们可以看到 ENIAC 的面板上有很多开关

连线，实际上这些开关连线就是控制 ENIAC 应该如何工作的。

但是插拔这些开关是一件非常费时费力的工作，往往经过几天的配置连线才能让

ENIAC 重新进行一项计算。

冯诺依曼指出 ENIAC 的开关定位和转插线只不过代表了一些数字信息。

完全可以像受程序管理的数据一样存放于主存储器中。

这就是后来的存储程序的概念，由于 ENIAC 已经开始制造，无法进行重大的修改，因此，在 ENIAC 完成之前研制小组就开始讨论下一代计算机的研制工作。后来，冯诺依曼形成了一份报告，这就是著名的关于 EDVAC 的报告草案。这份报告完成于 1945 年，并被分发给一些政府机关和科研机构的人阅读，这份报告详细的描述了一台名为 EDVAC 的计算机的设计方案。它所描述的计算机结构后来就被称为冯诺依曼计算机结构，简称冯诺依曼结构。今天我们所使用的计算机几乎全都是冯诺依曼结构。然而这份重要的研究报告却引起了 埃克特、莫克利等研究小组成员的强烈不满。一方面因为设计方案的公开发表 使得埃克特和莫克利无法就 EDVAC 申请专利。另一方面，应该说这个设计方案是研究小组共同讨论的成果。虽然冯诺依曼在其中做出了重要的贡献，但是他只署了自己的名字，独自享受了这份巨大的荣誉。这也成为了这个研究小组最终决裂的重要原因。

但是不管怎样，在 ENIAC 完成之前，EDVAC 的研制工作就已经开始了。

EDVAC

仍然是为美国弹道研究院设计，它实现了存储程序的概念，大幅提升了任务效率所以它也被称为存储程序式计算机，也就是冯诺依曼结构式计算机。同时 EDVAC 还采用了二进制，而不是 ENIAC 的十进制，极大的简化了逻辑线路。此外 EDVAC 还清晰的分成了五个基本部分，运算器、控制器、存储器、输入设备和输出设备。如果说 ENIAC 是世界上第一台电子计算机还存在争议的话，那关于谁是世界上第一台冯诺依曼结构计算机 人们对 EDVAC 倒是毫无疑问，因为它肯定不是。虽然 EDVAC 在 ENIAC 完成之前就开始了设计工作，但是很快这个研究小组就闹翻了，莫克利和埃克特 离开了宾夕法尼亚大学。并带走了大部分高级工程师，自己成立了一个电子计算机公司。EDVAC 的研究工作就此停滞，作为顾问的冯·诺依曼也回到了普林斯顿高等研究院。在这段时间，他还设计了另一台计算机，叫 IAS。后来 EDVAC 的研制工作重新开始。直到 1949 年 8 月才交赴弹道研究院。然而有很多的问题，导致无法使用。直到 1951 年才解决了全部问题正式运行。当然，此后 EDVAC 的表现非常出色，完成了很多重要的任务，直到 1962 年退役。那究竟是谁抢了先，研究了第一台冯·诺依曼结构计算机呢？这个人，在大西洋的彼岸。他是英国剑桥大学的莫里斯·威尔克斯。他以 EDVAC 为蓝本，设计制造了这台叫做 EDSAC 的计算机。这是一台冯·诺依曼结构的计算机。我们可以看到，它也有五个组成部分。这张较小的照片就是他的存储器。

那么大家可能要问，当时EDVAC都还没有设计完成，威尔克斯是怎么照着EDVAC进行设计的呢？其实我们刚才说过，

冯·诺依曼在完成了EDVAC的报告之后，这份报告就被送给了很多人阅读，引起了人们广泛的兴趣。

很多机构都开始参照这个方案进行设计。

当然。我们也得注意到，EDSAC在第一台的后面，还加了一个定语，叫做“实用的”。

因为它之前，1948年也有一台存储程序式计算机被研制出来。

这是曼彻斯特的Sam.

只不过那是一个实验性的设备，并不能实用。当然，如果较真的话，

ENIAC在1948年也被改造成了存储程序式的计算机。

这可是一台实用性的计算机。

所以，在这个电子计算机刚刚兴起的时代，想争个第一也不是一件容易的事情。

我们再来看看，莫克利和埃克特到底去做了什么。

他们离开宾夕法尼亚大学，倒不全是因为跟冯·诺依曼闹了别扭。

最重要的原因，是宾夕法尼亚大学要求他们将发明专利的所有权转让给学校。

在协商无效的情况下，他们只能离开。

后来他们创办了埃克特·莫克利计算机公司。并着手研究一台计算机，叫“UNIVAC”。

UNIVAC的设计开始于他们自己的计算机公司。

但是这两位科学的天才却不善于经商。

埃克特·莫克利计算机公司很快因为严重的亏损，

而被雷明顿·兰德公司收购。不过UNIVAC的研究得以继续。

最终，在1951年，第一台UNIVAC交给美国人口普查局使用。

这就是UNIVAC。

这，是UNIVAC的控制台。

但是当时，电子计算机还没有得到社会的广泛认可。

普通民众，更是很少有人知道。但是机会很快就来了。

1952年的总统选举，艾森豪威尔对史蒂文森。

在选举之后，统计选票是一项非常繁重，而且容易出错的工作。

当时UNIVAC也参与了此项工作。

结果，UNIVAC在统计了部分选票之后，最先准确的预测了艾森豪威尔将赢得选举。

这，就是UNIVAC当时打印在纸上的预测结果。

然而，很多人都不相信这个结果。包括全国性的电视媒体都不敢播报。

最终证实，UNIVAC的计算结果是正确的。

由此，UNIVAC一举成名，电子计算机也获得了人们的广泛关注。

此后，很多公司都进入了电子计算机的领域。

其中，原先主要在治疗机领域的IBM公司也转入电子计算机领域，并成功推出了大型计算机IBM701。

当时，冯·诺伊曼也正是IBM公司的科学顾问。他也和原子弹之父，奥本·海墨一起出席了- IBM701的揭幕仪式。

从此，IBM在这个新兴的市场上迅速扩张。

到五十年代中期，据称IBM占领了这个新兴市场的70%。

剩下的30%，有七家其他的公司所占据。

因此，媒体也戏称，当时的电子计算机市场，是“IBM和七个小矮人”的故事。

而在这七个小矮人中，也有已经并购了UNIVAC研制部门的斯普利·兰德公司。

所以我们可以说，是UNIVAC真正开启了商用计算机的时代。

从此，一批又一批的明星产品、明星公司，和明星人物登上了历史的舞台。

现在，电子计算机已经成为了推动社会进步的重要动力。

也是我们生活中，不可或缺的组成部分。而现在计算机，基本上都采用了冯·诺伊曼结构。

因此我们从下一节开始，将一起来探索，究竟什么才是“冯·诺伊曼结构”？

102-冯·诺依曼结构的要点

现在的计算机形态各异

但是究其本质，几乎全部采用了冯诺依曼结构 要了解计算机，首先要知道什么是冯诺依曼结构

当然，冯诺依曼结构的内涵是非常丰富的。

在这一节，我们先来探究其中的一些要点

要说冯诺依曼结构，我们就要从《EDVAC的报告草案》中说起

在冯诺依曼署名编写的这份报告中 详细描述了EDVAC这台计算机的设计方案。

从那之后一直到现在，绝大多数计算机都基于这个结构进行设计。

这样的结构就被称为冯诺依曼计算机结构

简称冯诺依曼结构。这是这份报告当中的一页。

我们可以看到下方还有EDVAC的字样。这份报告长达101页

虽然最终并未完稿，但其内容已经非常的丰富。

主要论述了两个重要的思想。第一，

控制计算机的程序应该存放在存储器中，而不是有开关连线来实现。

开关连线会大大降低计算机的运行效率。第二，计算机应该采用二进制，而不是十进制。

十进制的的方式会导致计算机的内部结构变得异常的复杂。

这份报告还明确了计算机的内部结构应该有五个部分组成。

包括运算器、控制器、存储器、输入设备和输出设备。

我们想要了解这五个部分的含义，不妨先来看一个小例子。

让我们跨越到70年代。

在1971年，诞生了世界上第一个商业微处理器。Intel4004

图中这颗金色的芯片就是4004.

它的面积很小。采用了10微米的制造工艺。内部晶体管2250个

主频最高740KHz，字长是4位 那么跟今天相比，它的各项指标都显得非常落后了。

现在的微处理器制造工艺已经达到了10纳米这个量级 晶体管数量甚至可以达到数十亿。而主频也提高了几千倍。

字长现在常用的是64位或者32位。

但是如果跟冯诺依曼所处的时代相比4004已经拥有了非常不错的性能。

实际上4004拥有与ENIAC相当的性能

那4004是用来做什么的呢？其实他最初的目的并不是用来制造计算机的。

这就是4004最初的产品。

大家看是什么？其实他就是一个计算器。和我们现在经常拿在手上用的是同一个类型。

那么当时，正是应日本一家生产厂商的要求

intel才设计了4004.那我们把这台机器打开看一看。

这是这台机器的主板。我们把一个局部放大。

中间这块金色的芯片自然就是4004 他旁边这块芯片是4001

是一块只读的存储器。它里面存放着一些指令。

但是只能读，不能写。它的特点是断电以后

内容不会丢失。4004左边这块是4002.

是一块随机存储器。它的特点在于可以很方便的进行读和写。

但是断电以后内容那个就会丢失。那么4003呢 是移位计算机的芯片。

那么实际用来负责进行输入和输出。

那我们可以看到4004之所以叫做4004

其实它还有一帮兄弟，从4001、4002到4003

那么他们和4004一起呢构成了一个完整的芯片组。

这个芯片的组合被称为MCS

那这张照片是这个芯片组的另一种实现。

同样也连接了这个计算器的键盘和打印机的输出 虽然这只是一个计算器，还算不上一台计算机

但他和冯诺依曼结构有着简洁明了的对应关系。

首先4004作为微处理器 或者叫CPU，其实包含了运算器和控制器的功能。

4002对应着存储器。

而键盘通过一片4003进行连接构成了输入设备。

另一片4003连接了打印机，构成了输出设备。

在这里我们就可以看到 这个计算器其实非常简洁明了的对应了冯诺依曼结构。

至于4001他并不属于我们现在要说的冯诺依曼结构的五大组成部分。

但是在冯诺依曼结构当中也有它的一席之地，这一点我们以后再说好，回来看冯诺依曼结构的要点。

冯诺依曼结构中规定了计算机的五大组成部分。

第一部分，运算器。简称CA 这是英文中央算术运算的缩写。

第二部分控制器。简称CC 这是中央控制的缩写。

存储器，简称M

输入设备，简称I。输出设备，简称O

这五大部分连接形成一体。就构成了冯诺依曼结构的计算机。

那么冯诺依曼结构 除了说明计算机应该由这五大部分组成之外

还指出计算机的数据和程序均已二进制代码的形式，不加区别的存放在存储器中

存放的位置由存储器的地址指定。第三，计算机在工作时能够自动的从存储器中取出指令加以执行。

当然这份报告中描述内容还有很多。我们先来看这几项。

其实第二点和第三点，其核心内容就是我们所说的存储程序的概念。

那我们就围绕着运算器、控制器和存储器这一部分 先来看一看存储程序到底指的是什么。

运算器、控制器和存储器 是冯诺依曼结构的核心

如果要与现代计算机进行对照的话，运算器和控制器应该对应于现在计算机中的CPU，

而存储器应该对应于现在计算机中的主存储器。

我们又常称之为主存或者内存。

CPU和主存储器之间一般通过系统总线进行连接。

我们以个人计算机为例

在这块个人计算机的主板上会有CPU芯片

主存，通过主板上的系统总线进行连接。

首先我们来看主存是如何组织的。

我们要关注两个名词。一个是地址，一个是内容。

地址是指每个存储单元对应的序号。一般是从0开始

0、1、2、3逐个编制。在这里我们是用二进制进行表示的。

而内容是指存储单元中存放的信息。

在现在的计算机当中一般一个存储单元就是一个字节。

也就是8个二进制位。组成的地址和内容的关系我们再来解释一下。

就好比有这么一个货架。这个货架有四层。

那我们为每一格进行一个编号。

那么编为第0格、第1格、第2格、第3格、 这些编号就相当于主存的地址

而货架上存放的物品就是贮存的内容。

主存中存放的就是我们计算机当中所需要使用的程序和数据

那么存储程序的概念又是如何体现的呢？我们一起来看。

如果把冯诺依曼结构的计算机比作一个餐馆的话

其中的CPU就好比是厨房。

主存好比是餐馆当中的仓库 在CPU当中大厨自然是

控制器。运算器呢就好比是厨具。

你给了他不同的原料，经过运算以后就可以得到相应的结果。

那么在仓库当中就放满着我们刚才所说过的那样的货架。

有很多，而且货架是被连续编号的。第一个货架

编了0，1，2，3第二个是4567。后面我们应该还能看到有很多的货架。

那么这些货架上放的是什么呢？我们会看到这里有一些像一张一张纸一样的

东西，它就是这个餐馆厨师会要用到的任务单，上面会写着要你这个厨师接下来该做一些- 什么事。

一张任务单上写着一条指令。呃，可能是要你炒一个菜，也可能要你从厨房搬个东西，

这都有可能。对于计算机来说这就是一条一条的指令。

若干条的指令就可以组合成一个程序。

把这个程序存放在存储器当中，这就是存储程序的一个很重要的概念。

后面这些蔬菜实际上就是餐馆的原料了，就相当于计算机当中的数据。

这里就体现出了冯·诺依曼结构当中所说的 将程序和数据不加区分的存放在存储器当中。

那么不是冯·诺依曼结构又是什么样的呢？

就好比这个餐馆里的库房里存放的全是原料，就是这些蔬菜，原料。

没有这些任务单，厨师的任务呢是通过别的方式发布给厨师的。

比如说要通过餐馆的经理，直接把任务单送到厨师这来，送一张任务单厨师就

完成一项任务，呃，这就是ENIAC的工作方式。

那大家觉得这样有什么不好呢？这似乎更自然一些，我们现实生活中也是这样的，哪有

把厨师要做的任务单跟这些蔬菜放到一起呢。对吧，前提在于现实生活中这样的运转可能

没有问题，餐馆经理给厨师下达的任务单速度也挺快，送来一张 任务单，厨师做一盘菜，过一会儿又送来一张任务单，厨师再做一盘菜。

衔接是非常合适的。对于计算机来说，它的要点在于CPU运转的

很快，即使ENIAC那个时候速度也是相当快的，

而外部人们能给它下指令的速度太慢了，我们想象一下这个餐馆如果开在

火星，我们一次已经把他做菜所需要的原料运到这火星这个餐馆的仓库里了，但是厨师要做的- 事情呢却是

由我们地球上的这个经理一张一张任务单送过去，我们现在到火星送一张任务单过去可能要

花一年的时间，厨师收到这张任务单以后，很快就从仓库里取来蔬菜，然后做了一盘菜可能就- 花十分钟的时间，

然后他怎么办呢？只能坐在那接着等第二张任务单，因为他不知道下面该做什么了。再等一年- 再收到一张任务单，再花十分钟做一盘菜。

这样的效率，大家看到，是非常低的，这正好就是ENIAC的工作方式。

现在我们就明白了冯·诺依曼结构中存储程序的概念。

建基所需要的程序和数据，通过输入设备进入了存储器

控制器发出地址给存储器，获得程序当中的对应的指令。

还可以得到相应的数据，控制器发命令给运算器，指挥运算器对数据进行相应的运算，

运算器可能会返回一些运算的状态，比如说是否正确完成等等。

最后还会将运算的结果存放到存储器中。

最后还需要通过输出设备 将结果输出到计算机外部的某些存储介质上。

这样才能完成整个计算的过程。

现在我们已经知道了冯·诺依曼结构中最重要部分，

那这个结构又是如何运转的呢？

下一节我们将通过一个小故事来了解这一点。

103-冯·诺依曼结构的小故事

如果一开始就深入到计算机内部的复杂结构中，

也许会让人有些迷惘。我们还是用一个轻松愉快的方式来入手。

通过一个餐馆的小故事来了解冯诺依曼结构是如何运转的。

冯诺依曼结构的计算机就好比这个餐馆，CPU就是厨房，

而主存就是仓库，仓库里的货架就好比主存当中的存储单元，

我们为货架的每一格都编上了一个序号，相当于主存当中的地址，货架上

存放的物品相当于存储单元当中的内容，主存中会存放着计算机的指令，

还有数据，对于这个餐馆来说就是厨师需要执行的任务

和做菜所用的原料，那厨房当中 承担主要控制任务的就是这位大厨了，

它也是控制器，而运算器则是出具。

那我们还需要一些附加的设备。

这个厨师记性很不好，他必须要在身边 有一张纸，上面写着下一张任务单的位置，

这里这张纸上写了1，厨师就知道我下一张任务单放在仓库的第一格，那么如果把任务单取回来以后

还需要放在身边随时查看所以他还可以有一个放置当天任务单的位置。

如果取来了任务单就放在这。

那厨师做菜，做完了菜放哪呢，不能直接端着锅冲到

仓库去吧。所以他身边操作台上总得放几个盘子，相当于CPU当中我们就称这些盘子为通用寄存器。

用于临时存放计算器的运算结果或者要送到运算器的操作数。

我们有可能从主存当中事先会取来

一些操作数，当然也需要执行指令去取来一些操作数，先放到通用寄存器当中，

那好这个餐馆是如何运转的呢？我们就让它像计算机的运转一样，

计算机运转的核心内容就是执行指令，计算机执行一条指令的主要步骤呢包括如下四步，

第一步叫做取址，第二部叫做译码，第三部称为执行，第四部是回写。

依次执行完这四部，计算机就完成了了一条指令的执行。

那我们就来看看这个冯诺依曼结构的餐馆是如何像计算机一样执行指令的。

现在我们假设这个餐馆处在这样一个状态，

厨师第一步要做的事情就是查看

下一张任务单的位置在哪，我要去取这张任务单了，我一查，发现是1，

我向主存发送了请求我需要第一格的物品，他并不知道第一格存放的是什么，他只是根据这里的编号向主存

发出申请，哎，主存并不简单是一些货架，

他还是要有一些控制逻辑，那么这类控制逻辑就会响应控制器的请求，需要第一格的物品，他就找到了第一格的物品，把第一格的物品给送回去了，

厨师收到了任务单了，就把他放在身边存放当前任务单的位置，

这就完成了第一步，取任务单的工作。而实际上这一步呢还需要有一项任务。

就是更新下一张任务单的位置。第一格取完了

等这件事做完之后下一次再去取任务单的时候再去取下一格，

就把它更新为2，那现在不会取第二格了，只是先更新，

等一会去执行下一条指令的时候再根据这个位置去向主存发出请求，

好，那这样的话取任务单的这一步就算真正的完成了。

第二步，在计算机当中称之为译码，对于这个厨师来说，就是分析刚才拿到的任务单。

你看，看这个任务单上写了这么几条。

第一呢说明了这项任务所需要用的方法。

第二呢说明了这项任务所需要的原料。

看来这个原料有两个位置，一个字仓库，一个在他身边的盘子里。最后这项任务完成了以后，看起来是炒一盘菜，那么炒完之后呢

还得把成果存放在某个地方，

这个任务单上也指定了，存放在A号的盘子里。好，这个任务就分析完了。

控制器就把他转换成若干组的控制信号——完成，

这样译码这个阶段就算完成了。

第三步，对计算机执行指令是其核心的步骤，

就叫做执行。对程式来说一样，执行这个已经分析完的任务，这个任务一共有4步，你先完成第一步，

第一步是要去取第六格的物品。好，与刚才一样，

向主存发出请求说我需要第六格的物品，主存会响应，把第六格的物品送过来。

由于这次是取运算要用的操作数，

所以返回的物品呢控制器就会把它放在运算器的其中一个入口，

然后第二部是取这个运算的第二个操作数。其实

这是放在A号盘子里的，就在厨师身边，很快，马上就可以拿到。

那么第三步是执行这项运算，那么控制器

会给出对应的信号给运算器，让它执行对应的运算，经过短暂的时间之后呢运算就完成了，运算结果也产生了。

好，执行阶段就到此结束了。

但是我们发现现在运算结果还放在运算器的这个输出端口上。

我们必须要把它转移到另外的地方，根据这条指令的要求，

运算的结果应该是存放在A号的通用计算器里。

好，那我们在最后一步回写也就是保存结果的时候呢，用控制器

把这个运算结果存放在A号的通用器当中，这样我们就完成了执行指令的全部过程。

做完了这条指令，控制器也不会闲着，借着来执行下一条指令。

它就会去查看下一张任务单的位置，跟刚才执行前一条指令的第一步是一样的，先去查看下一张任务单的位置，发现是2，

那么就向主存发出所需要第二格的物品，然后主存就会把第二格的任务单返回来，

注意我们取回来以后呢要自动更新成下一张任务单的位置，之后再做13，这个

更新完以后我们就像刚才一样，继续分析当前取回来这张任务单并完成它所指定的任务，最后保存结果，

然后再去取下一条指令。不断的重复，计算机就会自动的运转起来了。

这个餐馆的工作的过程其实就是我们计算机执行指令的过程。

CPU从主存中取出对应的指令，

那么刚才厨师做的那项任务呢对于计算机当中很可能是这样的，

CPU当中寄存器A有一个内容，然后主存的地址为6的这个单元里有这样的内容，

CPU根据这条指令把

主存中地址为6的单元的内容取出来，并把CPU内部寄存器A的内容取出来，可能执行一个加法，

然后把运算的结果更新到寄存器A当中，这样就完成了一条加法指令的运算。

这个冯诺依曼结构的餐馆

向我们展示了计算机运行的基本原理，不过餐馆仅仅是餐馆，

从下一节我们就要开始探索真正的计算机的内部结构。

不用担心，其实没有那么复杂。

104-计算机结构的简化模型

为了掌握计算机的内部结构，

我们要先来构建一个计算机的简化模型。

这个模型能够体现计算机内部的关键之处。

又足够的简单，便于理解。

这个冯诺依曼结构的餐馆，我们已经非常熟悉了。

我们都可以根据这个餐馆的结构几乎一一对应的来构造出我们的计算机结构的模型。这个计算机结构的简化模型我们又称之为模型机。

目前包含了CPU和存储器两大布局，中间通过系统总线相连。

系统总线又被细化为控制总线、地址总线 and 数据总线三个部分。

CPU和存储器内部各有响应的结构。

我们现在就来逐个部件的说明这个模型机是如何构建的。

首先来看存储器。

在设计计算机时要确定存储器的编制方式。

如果存储器按字节编制，那每一个存储单元就能存放八位的二进制数。

而且每个存储单元的地址是唯一的。

CPU要想访问某个存储单元，必须要通过地址总线给出相应的地址，

因此如果地址总线的宽度为N，则CPU能管理的存储单元数最多为 2^N 个，

假设地址总线的宽度为32位，那么CPU能管理的存储单元数就是 2^{32} ，就是iii。

CPU发来的地址将会存放在存储器地址寄存器，简称为MAR。

那CPU要读写的数据则会通过数据总线传送，数据总线的宽度一般为存储单元位宽的整数倍，

对应的数据则会放在

一个被称为存储器数据iii，简称为MDR

的结构中，最后

控制总线与存储器当中的控制总线相连，用于接受来自于CPU的读写信号或者向CPU反馈传输已经完成的信号。

我们再来看模型机的CPU。

首先来看其中的控制器。

控制器是用于控制计算机各个部件完成取指令分析指令执行指令等功能。

在我们的模型机中主要包含如下的部分我们逐个来进行分析。

首先是指令寄存器，

简称为IR。用于存放正在执行或者即将执行的指令。

这个指令是从存储器中取来的。

那么暂时存放在这个寄存器里头。程序计数器PC

用于存放下一条指令的存储单元地址，

我们还记得在餐馆那个例子中厨师要去取下一张任务单的时候就会检查他下一张任务单的位置，

那么其实对应的就是这个计算机的功能。

那么在取完下一条指令之后呢这个程序计数器需要自动递增以指向接下来那一条指令的地址，

那么CPU当中也有访存相关的寄存器，

访问存储器的地址，存放在MAR寄存器中，

用于写到存储器或者从存储器读回的数据在MDR

指令译码部件则是用于对IR也就是

指令寄存器当中的指令进行译码，

以确定其中存放的是哪一条指令。当确定了是哪条指令之后，

控制电路就可以产生对应的控制信号，这些控制信号在持续脉冲的同步下去控制CPU当中各个控制部件的动作。

然后我们来看模型机的运算器。

运算器顾名思义是用来完成运算的。

运算主要可分为算术运算和逻辑运算，

常见最基本的算术运算就是加法和减法。稍微复杂一些还有乘法和除法等等。

那常见的逻辑运算包括非、与、或等等。

运算器主要有如下的部件构成。

其核心部分是LU，也就是 算数逻辑单元的简称。它用于完成算数运算和逻辑运算，

我们从图中可以看出这个LU有两个输入，一个输出，比如我们要完成一个

加法，那么加法的两个操作数也就是被加数和加数非标从A端口和B端口

输入，经过LU计算后，将加法的运算结果送到输出端口，

为了便与描述，我们将LU的输入和输出

分别设置了暂时保存数据用的计时器。

分别记为X,Y和Z。

也就说LU将会对X和Y所保存的数据 进行控制器所指定的运算并将结果

输送到Z，此外，LU在运算时

还会产生相应的状态，比如这个运算的结果是否产生进位，

是否会有溢出等等。那这样的运算结果的状态会保存在F这个寄存器，F是标志的简称。

那运算器所要运算的数据

实际是从存储器来的。但是不能每一次运算都去存储器取数。

这样的话效率就太低了。

所以有一些常用的数需要提前从存储器中

取出来。那我们就需要在运算器中有临时存放这些数的部件。

这就是通用寄存器。在我们的模型机中设置了N个通用寄存器，

其编号从R0一直到RN-1，

通用寄存器当中的数据可以来自于存储器，

也可以来自于其他通用寄存器。或者来自于LU的

输出，也就说我们可以在两个不同的通用寄存器之间传递数据，
也可以在LU和通用寄存器之间
传递数据。在通用寄存器和LU
已经刚才提到的控制器当中的若干寄存器之间要传递数据的话，就必须经过CPU的内部总线。
内部总线是用于在CPU内部跟各个部件之间传递数据的装置。
我们举一个例子。例如我们需要将数据从R0传送到RX，
也就是可能为下一次运算进行准备，那么数据会通过内部总线从R0
传到了X，那么所谓这个传送的过程指的是什么呢？在之后我们具体讲电路实现的时候会为大家
详细解释，
在这里为了便于理解，我先为大家打一个比方，如果我们把内部总线看作是
输水的水管的话，那我们可以把这些通用寄存器等部件看作是蓄水的水池，
比如在这个例子中我们要从R0传送数据到X，R0和X都可以看做两个蓄水池，
它们分别有自己的进水口和出水口，那么R0我们看它有个双向
的箭头，证明它既可以输入又可以输出，所以它既有进水口
又有出水口。而X就一个单向的箭头，说明它只有进水口没有出水口，
因为我们知道X和Y作为LU的数据输入的临时寄存器的，
所以它不需要输出。从R0传送数据到X，
我们可以理解成在控制电路的控制下，将R0的出水管打开，
再将X的入水管打开，与此同时将其他所有部件的出水管和入水管都关闭，
这样R0里面保存的数据就像水流一样，经过了内部总线会留到了X当中，
那么经过一段时间之后，X中所保存的数就跟R0当中的数一样了。
这个我们就可以用来形象的理解，从R0传送输入到X，这就是内部总线的作用。
那这样，我们这个计算机结构的简化模型就构造完成了。

现在我们已经构建完成了这个模型机的核心部分，
想要了解这个静态的结构，一个有效的方法是观察其动态的运行过程。
下一节我们就来一起分析模型机是如何运行的。

105-计算机执行指令的过程

执行指令是计算机过程的核心，
理解了指令如何执行呢，也就把握住了计算机运行原理的关键。
在这一节，我们就一起来看这个模型机是如何执行指令的。
现在我们已经有了这样一个模型
而且我们也知道了计算机执行一条指令的主要步骤 包括取指、译码、执行、回写。
那我们就像在这个模型机上尝试执行一条计算机的指令。
这里给出了一个例子，我们想执行的这条指令是ADD R0,

[6]，这样的指令格式代表什么含义呢？从外边，我应该能看出来这是一条加法指令。

那么这个加法应该有两个原操作数，其中一个就是通用寄存器R0的内容；

另一个[6]，中括号就代表是存储单元的内容。

那么，中括号里面带一个6说明是地址为6的存储单元的内容。

我们把这两个原操作数相加，就可以得到一个运算结果。

那么，在我们这种格式指令当中，默认第一个操作数即是原操作数又是目的操作的。

这个运算结果就会更新到通用寄存器R0中。

现在我们就来看这条指令是如何在模型机上执行的。

假设模型机的当前状态已经有了一些初始值，

我们需要注意的是CPU中的R0

所保存的值是00000011，这等于10进制的3。

而存储单元地址为6的存储单元的内容是00000010，

相当于10进制的2。

那么我们所要做的，就是把这个存储单元的数和R0当中的数相加，最后再放到R0中。

我们还看到PC寄存器也就是指向下一条指令地址的寄存器，保存的值是

0001，那么，他所指向的内层单元，这里，所以这个内层单元中其实是

保存了当前所需要执行的那条指令。

我就用10101010这么一个值来指代，这仅仅是指代而已，

那我们就来看在这个模型机上是如何运行的。

指令执行的第一步-，取指。

取指这一步所要做的

是控制器将致力的地址送往存储器，存储器按给定的地址读出指令内容，送回控制器。

那我们来看具体的操作步骤。

首先请注意 控制器会发出控制信号将PC

寄存器中的内容通过内部总线传送到MAR中。

好，现在MAR寄存器中也保存了0001。

下一步，MAR

将会把这个地址送到地址总线上，

与此同时，控制电路

会在控制总线上发出相应的控制信号，代表这次访问存储器的操作是要读数据。

这样存储器的MAR寄存器 就会收到地址总线上传送来的地址并把它保存下来。

存储器中的控制逻辑，也会受到控制总线中传送来的控制信号 得知这次仿真操作是一次读操作。

这样存储器通过地址译码器 就可以查找到对应地址001的存储单元的内容；

并将该存储单元的内容送到MDR寄存器中。

然后存储器的控制逻辑，会通过控制总线将CPU反馈 当前的传输已经准备好了，同时，MDR之中的内容也会送数据总线上。

随后，CPU中的控制电路检测到来自控制总线的准备好的ready信号，就知道当前数据总线上已经准备好了数据。

因此MDR寄存器就会将当前数据总线上传送来的数值保存下来，这就获得了我们所要取的指令。当然，仅到如此是不够的。

MDR中的内容还不需要传送到智力寄存器中，也就是IR寄存器。

当指令的编码已经保存到IR寄存器中时，取指这部分工作就已经完成了。

真的已经完成了吗？请再仔细想一想。

其实我们还需要再做一件事，就是把PC寄存器更新为下一条指令所需要访问的地址。

刚才才是0001，现在变成0010，也就是10进制的2

。更新完以后我们才可以做效益不的工作。

那么，取指阶段到此正式完成。

第二部是译码。

在一步控制器会分析指令的操作性质并向相关的部件发出指令所需的控制信号。

当前的指令计算器IR中指令编码会送到指令译码部件。

指令译码部件根据指令编码

很快会发现这是一条加法指令。

而且，是要把R0和存储器中地址为6的单元的内容相加，并把结果存放到R0中。

然後，控制电路就据此产生对应的控制信号，发到相关部件中

将译码阶段就到此完成了。

第三步是执行。

控制器会从通用寄存器或者存储器中取出操作数，并命令运算器对操作数进行指令规定的运算。

首先，

根据这条指令我们会发现，还是需要去取操作数，其中一个操作数在存储器里中。因此，

这一步会在MAR中放置要访问的存储器的地址，0110。

就是10进制的6。那么，随后的过程类似于取指阶段的操作。

MAR将地址发到地址总线，同时，控制电路在控制总线发出读操作的控制信号。

存储器的MAR控制逻辑会接收到相应的信号，

然后查找到对应的地址，这次要查找的地址是0110，对应的内容

会送到MDR这个存储器。

然後，控制逻辑会向CPU反馈当前数据已经准备好了的信号。

然後，MDR的内容也会被放置到数据总线上，

CPU会接收数据并保存在MDR寄存器中。

由于这个数据要进行加法运算，所以控制器会进一步将MDR中的数据传送到ALU的输入端目前会暂存到Y寄存器中。

这一个操作数现在就准备好了，那么另一个操作数是放在R0

中；因此，控制器还会将R0中的数据传送到ALU的另一个输入端，也就是x寄存器。

现在，两个操作数都准备好了，

在控制电路的控制下ALU就会进行运算

再将x、Y中的内容执行加法并计算出结果。

结果就是00000101也就是10进制的5。

到此，执行阶段就完成了。

第四步是回写。

也就是将运算结果写入到通用寄存器或者寄存器。

现在，运算结果还在ALU的输出端，也就是Z寄存器中，

控制电路会给出相应的控制信号

将Z寄存器当中的内容传送到R0中。

R0目前的内容是之前的原操作数，

也就是00000011。随后，会被新的结果所覆盖。

这样，我们这个加法运算的结果就已经保存到了R0寄存器中，

回写阶段到此完成。这条指令你就执行完毕了。

然后，CPU就会自动执行下一条指令。

它们是和刚才一样，先将PC中的内容传送到MAR寄存器当中，

然后MAR会将这个地址传送到地址总线上依次执行下一条指令。

现在，我们通过学习计算机制度和执行指令 应掌握弗洛伊曼结构的核心内容。

但是，人们是如何向计算机下达指令的呢？计算机的运行结果又是如何让外界得知？

这是我们下一节将关注的内容。

106-计算机输入和输出

如果无法与外界进行联系，计算机的CPU，存储器再强大也没有任何的用处。

因此我们需要在我们的模型机上加上输入与输出

这样才能构成一个完整的计算机模型。

这就是冯·诺依曼计算机的五大组成部分

分别是运算器、控制器、存储器、输入设备和输出设备。

现在我们就主要来介绍输入设备和输出设备。

根据冯诺依曼结构的要求，计算机要执行的程序首先需要通过输入设备传送到存储器中

那么这个过程是如何完成的呢？就让我们来看一看冯诺依曼这个时代所用的计算机。

当时编写程序并不像我们现在这样方便 程序员需要在长长的纸带上打上小孔
若干个小孔就代表一条指令，那么将这样穿孔的纸带送入计算机后
计算机可以通过有孔或者没孔的区别转换成0和1的二进制信息。

从而识别出程序员所表达的指令。经过运算以后

计算机将通过电传打字机将输出的结果打印在纸带上。

这样人们就可以通过观察，分析纸带上的穿孔情况来判断输出的结果。

那这样的计算机的输入和输出与冯诺依曼结构又是如何对应呢

我们先来想，这个穿孔纸带应该对应着什么。是对应着输入设备吗？

其实纸带上的信息还是要通过特定设备才能被识别。

所以穿孔纸带应该是作为输入设备的来源。其实在冯诺依曼结构中，对此也进行了规定。

除了输入设备I，输出设备O 冯诺依曼结构中还规定了外部的记录介质，成为R

实际上穿孔纸带就是这种外部记录介质。

它可以静态的存储计算机所需的信息。

不会因为计算机断电而丢失。

那好，刚才提到了电传打字机。有属于哪个类别呢？ 其实电传打字机应该属于输出设备的一种。

电传打字机会将计算机的运行结果打印到纸带上

所以电传打印机输出的纸张也可以认为是外部记录介质R。

除了纸张，还可以用带磁性的材料 存储信息。类似于我们现在使用的硬盘。

那么硬盘应该属于输入设备、输出设备 还是外部记录介质？

首先我们看硬盘是可读又可写的。所以我们至少应该是这么连接的

而硬盘中的信息在断电之后是不会丢失的 所以硬盘也具有外部记录介质R的特性。

所以严格说来，硬盘里记录信息的磁片才是真正的外部记录介质。

而硬盘当中还会有其他的控制芯片 可以认为是输入或者输出设备的一部分。

在通常情况下我们没有必要划分的那么苛刻 可以认为硬盘就是属于外部记录介质R这种类型。

那么现在的计算机当中 很多设备都会既有输入的功能又有输出的功能。

所以通常我们会这么来画这个结构图 将输入设备和输出设备化成一个部件

既有输入的功能又有输出的功能。

现在，我们就来看 如何在我们的模型机上添加这样一个输入输出设备。

如果直接考虑硬盘这样的设备，就太过复杂了。

我们希望现在模型机上添加一个简单的输入输出设备。

说到简单的设备，我们先来看一个小故事。

这是爱德华·罗伯茨

当他60年代从军队退役之后，开了一个小公司， 制造和销售台式的计算器。

但是后来由于德州仪器这样的大公司进军计算机的市场，采用了

低价销售的策略，罗伯茨这样的小公司自然无法抵抗。很快就到了破产的边缘。

为了挽救这样的局面，罗伯茨也苦苦厮守了很多办法。其中一个办法就是设计制造面向个人的计算机。虽然但是计算机已经开始小型化了，

但是那是，所为小型计算机的体积仍然很大。不是个人能够轻易搬动的。

而且动格上万美元的价格，更不是个人所能承受的。

而大众电子杂志的编辑跟罗伯茨说

如果你能做出400美元以下的计算机，我就在杂志封面上给你做推荐。

后来罗伯茨真的造出了这样的计算机。

这就是牛郎星8800

它内部主要有两块集成电路。一是intel8080微处理器

第二是一块256字节的存储器。

那么他在面板上提供了简单的输入输出 靠手来上下拨动一些开关进行编程输入

计算机在经过运算之后，会将计算的结果通过 这些小灯泡进行显示。虽然构造简单，

但这确实是一台设计精巧，而且有实用价值的计算机。

更重要的是它价格非常便宜。真的做到了400美元以下。

于是，大众电子杂志就在1975年1月 真的为他做了封面的推荐。

此后，订单像雪片一样地飞来。挽救了罗伯茨的公司。

当时，很多电子计算机的爱好者和团体都争相采购

牛郎星8800型计算机以进行分析研究。

其中就有乔布斯和他的搭档沃兹尼亚克。

他们一方面认真学习了这台计算机精巧的内部结构，

另一方面，他们也对如此简陋的输入输出感到非常不满意。

在这样的影响下，同一年，他们就很快设计完成了自己的产品。

就是苹果I型个人计算机。这台计算机有键盘的输入

并且可以连接电视机作为显示器输出。这也是苹果公司的第一个产品。

那好，我们回来看如何在模型机上添加输入输出设备。

我们不需要那么复杂的输入输出。

现在，我们只想添加一个类似于牛郎星8800型计算机的输入和输出

也就是手动的开关和小灯泡的显示。

我们可以设想有一个类似于存储器结构的输入输出芯片。它内部有

两个单元，每个单元都有自己的地址。我们这里设定的是1110和1111这两个地址。

他们和存储器里面的地址都不相同。因此可以相互区分。

那这个输入输出如何与计算机的其他部分相连呢？

我们知道CPU是通过控制总线，地址总线和数据总线与存储器相连的。

现在我们就来对此做一个简单的改造。

将控制总线分别连接到存储器和输入输出芯片的控制逻辑

地址总线也同样分别连到这两个部件。

数据总线也是如此。然后将输入输出设备中其中一个单元

连接到手动的开关，这个单元中有8个比特 每个比特连接一个开关。

当开关拨到上方时，对应的位就会被设置为1。开关拨到下方时

对应的位就会被设置为0。另外一个存储单元接到8个小灯泡

当这个单元中的比特为高电平时，对应的小灯泡就会亮起。

如果这个比特是低电平时，对应的小灯泡就会熄灭。

那现在我们就有一个非常简单的输入输出。

那这个输入输出在计算机上是如何工作的呢？我们通过一个简单的输入的场景来进行学习。

我们假设这时候CPU和存储器已经开始工作了。

其中一条指令就是要读取这8个手动开关的状态。

假设现在8个手动开关的状态 是前三个开关被拨到高位，后五个开关被拨到低位

那地址为1111这个单元中所保存的2进制信息就是这样的。

那CPU如何能读取到这个输入信息呢？

我们首先来看地址总线。CPU会将地址总线

第一总线上所发的地址是1111，那么在总线上会有一些简单的电路进行地址的识别，

可以发现这个地址应该发到输入输出部件，而不是存储器。

与此同时，我们来看控制总线。

CPU会发出读的信号，

从而告知输入输出设备目前是要进行一个读操作，

那输入输出设备的控制逻辑和地址译码器 就会从地址为1111的单元中读出相关的信息，

送到数据总线上。

数据总线会将这个数据 最终送回到CPU中，

这样就完成了输入信息的读取工作。

此后，CPU可以对这个输入的数据

进行运算，或者根据这个数据调整后面的执行的行为，也可以把这个数

暂时保存到存储器中以备后来的使用。

那这样就是一个输入的简单场景。然后我们再来看一个输出的简单场景。

既然是输出，我们就要给出输出单元对应的地址，在这里是1110，与刚才一样，

地址总线会将这个地址传送到输入输出设备，

同时，控制总线上回给出写的控制信号，

然后在数据总线上，

会给出我们想输出的数据，这里设为1100,1100。

然后输入输出设备就会根据来自总线的信息判断出

要向地址为1110的单元写入数据1100,1100，

此后就可以完成写入的操作。

当写入操作完成后，对应的小灯泡就接收到了

高低不同的电压，我们就会发现1所对应的灯泡亮起，

0所对应的灯泡熄灭。这就是一个输出的简单场景。

这样我们就将CPU,存储器 输入输出设备通过系统总线连接了起来。

但仅仅如此肯定是不够的。我们还会想增加更多的输入输出设备。

比如说打印机，比如说 硬盘，有了这些其实就很像早期计算机的构造。

计算机作为一个整体已经可以开始运行了。

当然我们还可以通过网络将计算机内部的信息 与其他设备或其他计算机进行交互，

我们还可以增加更多的输入输出设备，进行更为丰富的人机交互，

现在看到的这些就是在个人计算机中常见的输入输出设备了。

早期的个人计算机中几乎每一个设备都需要通过

对应的输入输出芯片或者板卡进行连接，例如我们会需要单独的网卡，单独的声卡等等。

在现在的个人计算机当中，大多数的输入输出设备的控制芯片都会集中在

主板上的南桥芯片中，例如我们刚才提到的 这些设备。

那么对于一些比较复杂的输入输出的需求，

比如高性能的显示，还是需要独立的板卡与外部的设备进行相连。

这就是以个人计算机为例所展示的输入输出设备相关的情况。

运算器，控制器，存储器再加上输入和输出，

我们通过模型机的构建已经掌握了iii结构的基本原理。

但是这个基本原理和真实的计算机之间是否能够一一对应呢？

我们下一节来探讨这个问题。

107-冯诺依曼结构和具体实现

我们现在日常所接触的计算机，几乎全部都是冯·诺依曼结构，都遵循着存储程序的基本原理。

那这些计算机的具体实现，和当年冯·诺依曼的设计方案有哪些区别和联系呢。

这就需要我们对比分析了。

冯·诺依曼计算机的主要构成，我们现在已经非常熟悉了，包括运算器CA，控制器CC，存储器M，

输入设备I，和输出设备O。

此外，还有在计算机外部的记录介质R。那这样 原理性的框图和现在计算机的具体实现是如何来对应的呢。

我们还是拿个人计算机作为例子来进行讲解，在这块个人计算机的主板上，

可以插上CPU芯片，存储，我们也常称为内存条，
再接上硬盘，连上键盘，插上显卡，连上显示器，
最后再接上电源，这样就构成了一台可以正常运行的计算机。

那CPU对应的是运算器CA和控制器CC，主存对应的是存储器M。

这些我们已经比较清楚了，那CPU是如何获取指令，开始执行的呢。

这块主板上采用的是个人计算机上长期占据主导地位的南北桥结构。

所以CPU对外会连接的一块芯片，叫做北桥。

这是在主板上非常明显的一块芯片，CPU想要访问主存，

就得通过北桥芯片。但是我们还要注意，计算机刚启动的时候，主存里面是没有信息的，因为当计算机断电之后，

主存的信息都会丢失。那在系统启动之后，CPU从哪里获得指令呢？

我们也可以看到北桥下方，还有一块比较大的芯片。

它就是南桥。

我们已经知道南桥内部，集成了很多输入输出设备的控制器，其中就包括硬盘的控制器。

所以硬盘实际上是和南桥芯片相连的。

而硬盘，我们也已经知道了，它是属于外部记录介质，它内部保存着计算机系统所需的程序和数据。

那CPU所需要的指令是否就直接从硬盘取得呢。

其实也不是。因为要从硬盘中访问数据，相对比较复杂。

对硬盘进行配置和访问，本身还需要程序来控制。

这也就是所谓的硬盘驱动程序。

所以在计算机刚启动时，CPU是无法从硬盘中，获得自己想要的第一条指令的。那第一条指令在哪呢？

实际上在主板上，还会有一个很小的芯片，它会与南桥相连。

一般我们称之为BIOS芯片，这个芯片存储容量很小，

会保存的一段比较简单但十分重要的程序，包括检查主板上都有哪些设备，

这些设备工作是否正常。这个芯片实际上是一颗只读存储器。

当系统断电之后，只读存储器中保存的信息是不会丢失的。

所以当计算机启动后，CPU可以依次通过北桥，南桥，访问到BIOS芯片，从里面取得指令。

所以这颗BIOS芯片，虽然是一颗只读存储器，但是从冯·诺依曼结构原理上来看，它并不属于存储

器的范畴，而是和硬盘一样，属于外部记录介质，它和早年计算机的穿孔纸带也属于同一类别。

严格地说，BIOS是指这颗只读存储器当中存储的软件。

它是基本输入输出系统的缩写。

那通常我们把保存了BIOS这个软件的只读存储器 成为BIOS芯片。现在计算机主板上的BIOS芯片，

有很多种封装形式，位置也有不同。

但大多会在芯片旁边的主板上，标出BIOS的字样。大家有兴趣可以找一找。

那CPU从BIOS中取出指令后，会依次地检查主板上的各个设备，包括有显卡，通过显卡点亮显示器，检查键盘的存在，之后我们就可以通过键盘进行输入。

这些都是与I/O设备。此外，现在的显卡功能越来越复杂，

性能也越来越强大，尤其是其中的图形处理单元。

我们也常称为GPU。GPU往往支持很多复杂的数学运算，拥有很强的运算能力。

所以有时候，也会让GPU承担一些运算的工作。

实际上在现在高性能的超级计算机中，经常会采用大量的GPU来进行运算。所以从这个意义上讲，

GPU也承担了运算器和控制器的一些功能。

那在这个结构当中，只有北桥芯片的特征比较特殊一些。

实际上，它提供了各个芯片之间互相访问的通道。

当然，我们也不用太纠结，北桥到底对应着哪个结构。

因为整个南北桥的结构也在迅速地发生着变化。

这还是刚才那块主板，我们先来看右侧这张图，最上方是微处理器，也就是CPU。

这颗芯片中，往往包含着多个CPU核。

往下，它连接着北桥。

北桥中，主要会有主存控制器，对外连接着主存，那

现在的显卡，一般采用PCIE接口，所以在北桥中，会集成PCIE控制器，

对外通过PCIE的插槽，连接PCIE的显卡。当然如果你对显示的性能要求不高，

也可以通过北桥内部自带的集成显卡，直接连接显示器。

那北桥还连接着南桥，南桥中，集中了大多数I/O设备的控制器，那我们再来看左边这张图。

这是CPU。

它对外的访问都需要通过北桥，

而其中对传输的性能要求最高的就是与主存的这条

通道。主存通道的性能也成为了整个计算机性能的关键瓶颈。

除此之外，另一条对性能影响很大的通道就是显示通道。

那么这就是PCIE的插槽，这上面可以插上PCIE显卡。再有一条重要的通道，

就是北桥与南桥的连接，通过这条通道，外部的输入输出设备可以对主存进行访问。那在这里，

红颜色的箭头表示传输压力很大的通道，而南北桥之间的绿色箭头表示这条通道传输压力相对较小。

那在这种情况下，首先要解决对性能影响最大的关键问题。

这就是CPU访问主存的通道。

所以在南北桥架构上，发生的第一个重要的变化，就是北桥中的主存控制器被移到了CPU芯片当中。

通过左边这张图，我们也可以看出来，CPU的插槽与主存的插槽变得更近了，

这就是因为现在CPU是直接访问主存，而不需要再通过北桥了。

这样可以大大地提高数据的传输率，进而提升系统的性能。

现在，显示的这条通道，就成为了下一个要解决的问题。

而南桥中连接的这些设备，暂时还不关键。

因此，在最新的个人计算机的CPU设计中，不仅包括了

主存控制器，还包含了PCIE控制器，在外部可以直接连接显卡。

这样北桥当中最重要的一些部件，都被集成到了CPU芯片当中，那北桥就没有存在的必要了。

于是将北桥中剩余的一些功能，和南桥芯片整合到一起之后，所谓南北桥的架构，

其实已经消失了。整个个人计算机的主板由三片式的架构缩减成了两片式。

由于没有了北桥，南桥这个名字也显得有些奇怪了。

所以它现在往往有一些不同的名字。那我们再看左边这张图，

现在CPU直接连接着主存，还直接连接着显卡，

还与原先的南桥直接相连，

这样就将原先通过主板的复杂的通路，大多移到了芯片的内部。

而芯片内部的数据传输率，是远远高于主板的。

这样就大大地提升了系统的性能。

那既然从三片缩减成了两片，我们可不可以进一步缩减呢？

其实已经有人这么做了，这就是系统芯片的概念。

系统芯片，可以认为是将计算机系统，或其它电子系统集成到一个单一的芯片上。

我们来看一个例子，这是一颗硬币大小的芯片，我们来看它内部

包含了哪些功能呢。这个框图就是这颗芯片中所包含的功能模块。

这部分是一个四核的CPU。

这一部分 是显卡的功能模块，

而这些是南桥中数量繁多的 各种接口的控制器，

而且这个芯片还采用了一些特殊的方法，把主存也集中在了一起。

这样，就在这一颗芯片内部，集中了iii结构当中的所有组件。

所以我们可以用这样的芯片制造出非常小巧的计算机。你看这个计算机的主板

大约只有手掌大小，却拥有非常丰富的外部接口，与一个普通的计算机并没有什么差别。

那这样高集成度的优势就让系统芯片的技术 广泛应用于智能手机，平板电脑等移动计算设备上。

那么原先的计算机中那么多的芯片和板卡 为什么能够集中到越来越少的芯片当中呢？

那我们就得说一说一个非常著名的定律。

这就是摩尔定律。摩尔定律最初是由摩尔提出的。

早年，摩尔加入了晶体管的设计者肖克利的实验室，但是由于肖克利的性格问题和这些青年人产生了很大的矛盾，那么包括摩尔在内有8个年轻人一起密谋脱离了肖克利实验室。

创办了仙童公司，而肖克利非常生气，把他们称为8个叛徒，而仙童公司

迅速聚集了大批有才干的人，创造出了属于自己的辉煌，

后来由于投资方的一些原因，仙童公司的这些精英们又逐渐的离去，

他们当中的很多人都在硅谷创办了公司，因此人们常将仙童公司比作硅谷的人才摇篮，

说仙童公司就好比一颗成熟的蒲公英，风吹之下 飘散到硅谷各处，生根发芽，

而其中就有摩尔和当年 八叛逆之一的罗伯特诺伊斯共同创办的英特尔公司。

而在1965年，当时摩尔还在仙童公司，

他应邀为电子学杂志35周年的专刊

撰写文章，他在文中做出了这样一个预测，在最低原件价格下， 集成电路的复杂度每年大约增加一倍，

这一增长率至少可以维持10年。

这个预测就是摩尔定律的原型。

这十年，集成电路的增长果然非常的迅速，不过增长的速度和摩尔的预测略有差别。

所以十年后，摩尔在一次报告中调整了他的预测，将每年增加一倍

调整为每两年增加一倍。所以我们要注意 摩尔进行的预测是指集成电路的复杂度，

而且他认为这个增长速度应该是每两年增加一倍，

但是后来被称为摩尔定律的这一预测又发生了很多的演变。

现在我们常见的摩尔定律的描述是这样的。

当价格不变时， 集成电路上可容纳的晶体管数目约每隔18个月便会增加一倍，

性能也将提升一倍。我们需要注意的是，这和摩尔原本的预测是有些差别的。

首先时间不一样。第二，摩尔也没有提及性能的事情。

此外摩尔定律还有一些其他的描述。

比如每18个月，计算机产品的性能会翻一番，

或者说相同性能的计算机产品每18个月价格会

降一半，在这样描述的基础上还引出了反摩尔定律等其他的描述。

既然说到了性能，我们不妨来做个有趣的对比。左边是1946的ENIAC，

它每秒钟能执行5000次定点的加法运算。

右边是现在全世界最快的超级计算机， 中国研制的天河2号，从2013年到2014年

连续3次占据世界超级计算机500强的首位。

它每秒钟能执行3.3亿亿次浮点运算。

我们不妨就来一个非常粗略的比较，就以运算次数作为指标，

那么从ENIAC到天河2号，67年的时间，性能增长超过6万亿倍，

恰好相当于每18个月翻了一番。

当然这只是一个很有趣的对比，它并不严谨。

所以我们还是回来看一看摩尔定律它原本想描述的内容。

这张图展示了微处理器，芯片的晶体管数量的增长情况。

在集成电路芯片上，晶体管数量的增长速度才是摩尔最初预测的内容。

最早的是1971年的4004，大约有2300个晶体管。

现在来看整体的发展趋势。

从1971年到2011年总共40年的时间，

微处理器晶体管的数量从2300个迅速增长到26亿个，

增长幅度超过100万倍。

如此算来，不到两年就翻了一番。与当年摩尔的预测是非常贴近的。

虽然在这之中也有一些人为控制的因素，

而且摩尔定律是否已走到尽头也是这些年来不断被提及的话题。

但是无论怎样摩尔定律所展示的集成电路的快速发展

的确是计算机性能快速提升的关键动力。

当然在摩尔定律推动下产生的这些计算机其实都是冯诺依曼结构的计算机。

我们需要牢记的是冯诺依曼结构存储程序的概念，和冯诺依曼计算机各大组成部分之间的关系。

由此看来，私人计算机的具体实现方式发展变化的很快，

但是其内在的结构特点却长期保持着稳定。

因此想要真正掌握计算机的结构和运行的原理，就要不断的学习其最新发展的变化，

也要深入的了解它内部不变的原理，二者不可偏废。