

# 信息与电子工程导论

Introduction to Information Science and Electronic Engineering

## 4.3 微处理器和计算机系统

章献民 主编

浙江大学出版社

2023年9月

# 知识图谱

- 2.1 时域和频域
- 2.2 模拟和数字
- 2.3 编码和调制
- 2.4 电磁场与波

## 2 信号与数据

场与波

信号

数据

信息

## 1 信息与信息技术概述

- 1.1 信息
- 1.2 信息科学技术概述
- 1.3 知识图谱

## 3 电子器件与电路

- 3.1 电路模型和基本定律
- 3.2 晶体管和集成电路
- 3.3 集成运算放大器

器件

电路

处理器

计算机

网络

## 4 逻辑与数字系统

- 4.1 数字逻辑和电路
- 4.2 组合逻辑和时序逻辑
- 4.3 微处理器和计算机系统
- 4.4 嵌入式系统
- 4.5 EDA技术

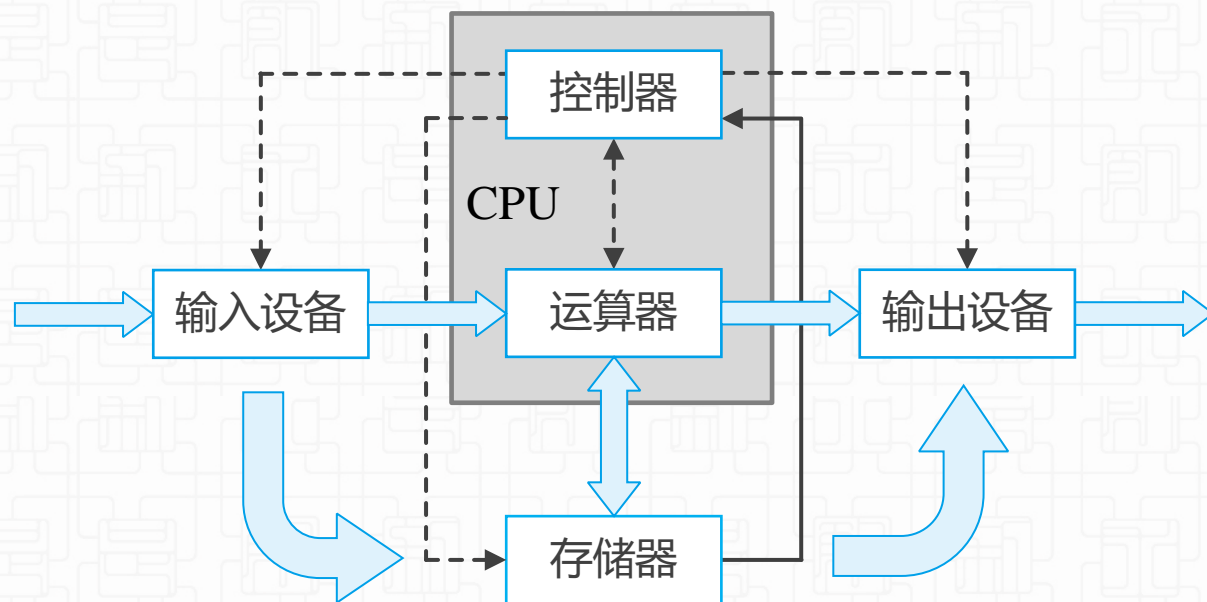
## 5 互联与计算

- 5.1 通信与网络
- 5.2 物联与数联
- 5.3 计算与智能

## 内容提要

❖ 计算机概述

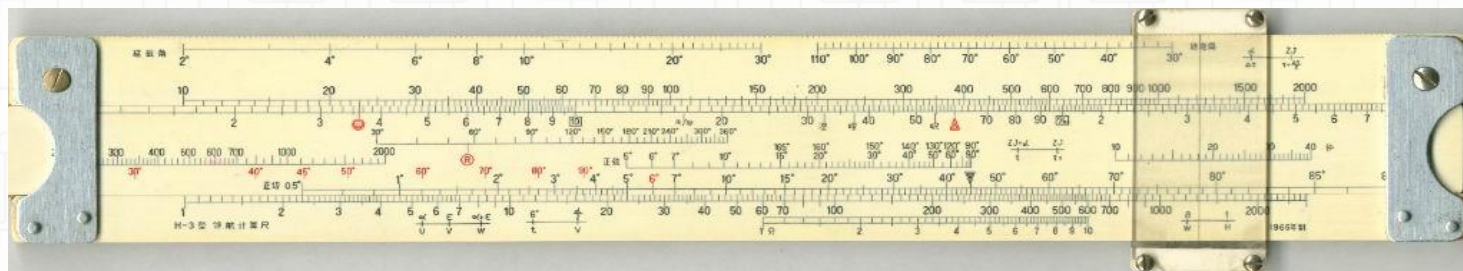
❖ 微处理器





## 早期的计算设备

- ❖ 算盘
- ❖ 纳皮尔算筹 (Napier's Rods)
- ❖ 计算尺



	9	3	4	
2	2	0	1	3
9	0	0	0	1
3	3	1	1	4
	2	7	6	

## 计算机起源

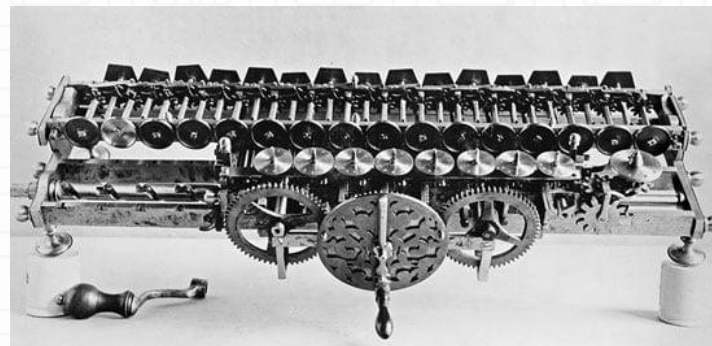
- ❖ 17 世纪，欧洲一批数学家就已开始设计和制造以数字形式进行基本运算的数字计算机。
- ❖ 1642 年，法国数学家**布累斯·巴斯柯**采用与钟表类似的齿轮传动装置，制成了最早的十进制加法器 (**pascaline**)。



Blaise Pascal  
Jun 19, 1623—Aug 19, 1662



Gottfried Wilhelm Leibniz  
Jul 1, 1646—Nov 14, 1716

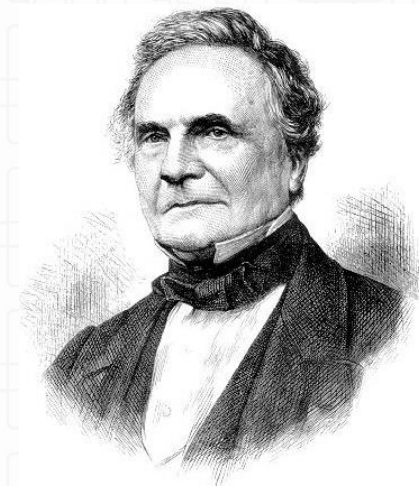


- ❖ 1671 年，德国数学家**戈特弗里德·威廉·莱布尼茨**发明的步进计算器 (**stepped reckoner**)，增添了“步进轮”装置，可通过重复加减法和移位进行乘除运算，这是第一台具有完整的四则运算能力的机械式计算器。

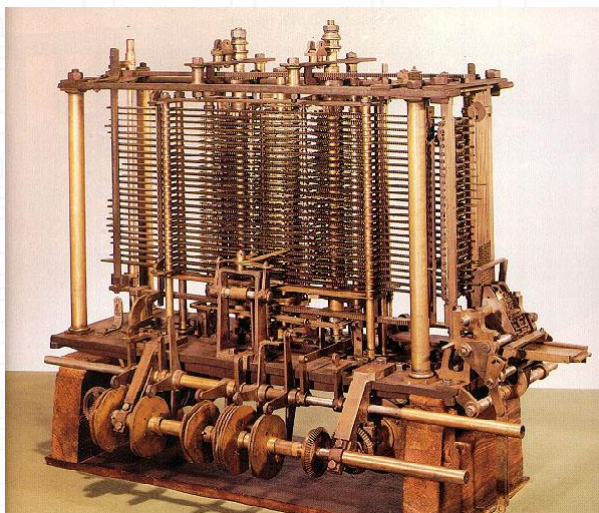
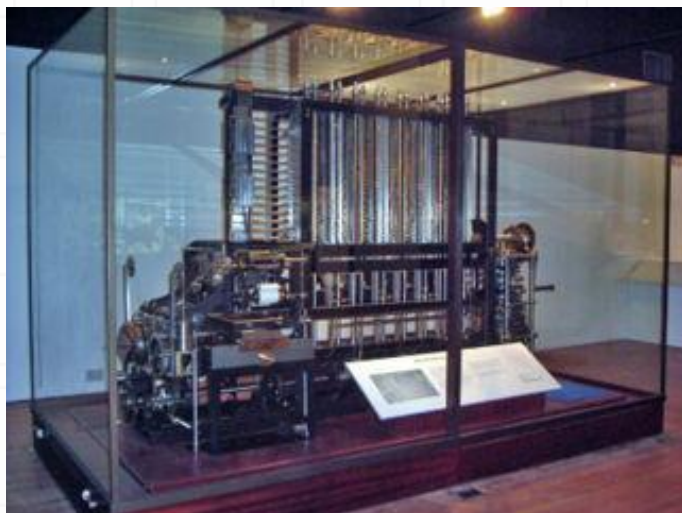


## 差分机 (Difference Engine)、分析仪 (Analytical Engine)

- ❖ 1822年，英国文学家、数学家、哲学家、机械工程师查尔斯·巴贝奇提出了一种新型机械装置“差分机”。
- ❖ 把函数表的复杂算式转化为差分运算，用简单的加法代替平方运算。
- ❖ 1823年开始建造差分机，试图制造和组装25,000个零件，总重近15吨。



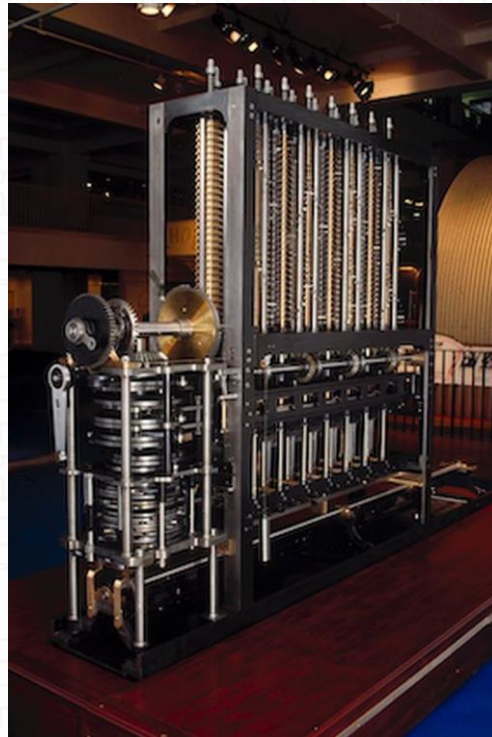
Charles Babbage  
Dec 26, 1791—Oct 18, 1871



- ❖ 在差分机的建造期间巴贝奇构想了一个更复杂的机器——分析机。
- ❖ 分析机是“通用计算机”，体现了现代数字计算机几乎所有重要功能，但限于当时的技术条件而未能实现。

## 现代计算机的鼻祖

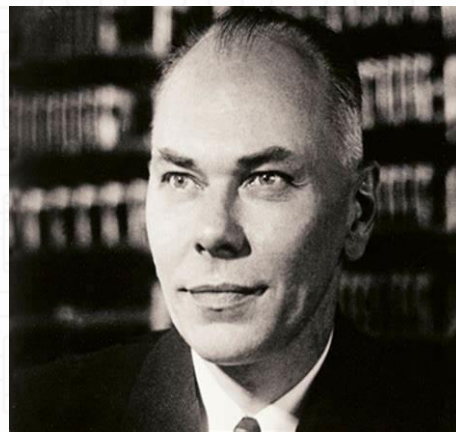
- ❖ 巴贝奇在有生之年虽未完成完整的差分机或分析机，但他那超越时代的思想，在遭受冷遇时仍坚持理想的科学精神却是留给后人的巨大财富。
- ❖ 1991年，在巴贝奇诞辰200周年之际，伦敦科学博物馆使用巴贝奇时代的机械技术建造完成了**差分机2号**的主机。
- ❖ 2002年，在差分机2号开始设计的153年后，一台完整的机器终于面世。它由8000多个部件组成，重5吨，长约335厘米。这台机器工作良好，证明了在巴贝奇的年代使用当时的技术**实现自动计算机器**的可行性。
- ❖ **巴贝奇曾在自传中写到：**“如果有人没有被我的经历吓倒，而且可以基于不同的原则或更简单的机械方法，成功制造出能体现完整数学分析执行过程的机器，那么我将自己的名誉留给他又何妨？因为只有他才能完全领悟我努力的本质以及结果的价值。”





## 首台自动按序控制计算器

- ❖ 1937年，哈佛大学研究生霍华德·哈撒韦·艾肯与IBM公司开始合作制造机电计算机。
- ❖ 1944年5月完工并投入使用，命名为ASCC (Automatic Sequence Controlled Calculator, 自动按序控制计算器)。8月，交付给哈佛大学，称之为“哈佛马克一号 (Harvard Mark I)”。



Howard Hathaway Aiken  
Mar 08, 1900—Mar 14, 1973

- ❖ 由开关、继电器、转轴以及离合器等构成，可以按照任何指定的顺序对23位十进制数字执行查表和四则运算，几乎完全实现了巴贝奇的分析引擎原理。但无法执行“条件分支”的操作。



## 可计算性理论

❖ 1936年，英国数学家**阿兰·麦席森·图灵**发表《论可计算数及其在判定性问题上的应用》一文，提出了一种**抽象的计算模型**——图灵机 (Turing machine)，将人们使用纸笔进行数学运算的过程进行抽象，由一个虚拟的机器替代人类进行数学运算。



Alan Mathison Turing  
Jun 23, 1912—Jun 7, 1954

1. 世界上是否所有的数学问题都有明确的答案？
2. 如果有明确的答案，是否可以通过有限步骤的计算得到答案？
3. 对于那些有可能在有限步骤计算出来的学习问题，是否有一种假想的机械，让它不断运行，最后机器停下来的时候，那个数学答案就计算出来了？

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO  
THE ENTSCHIEDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers

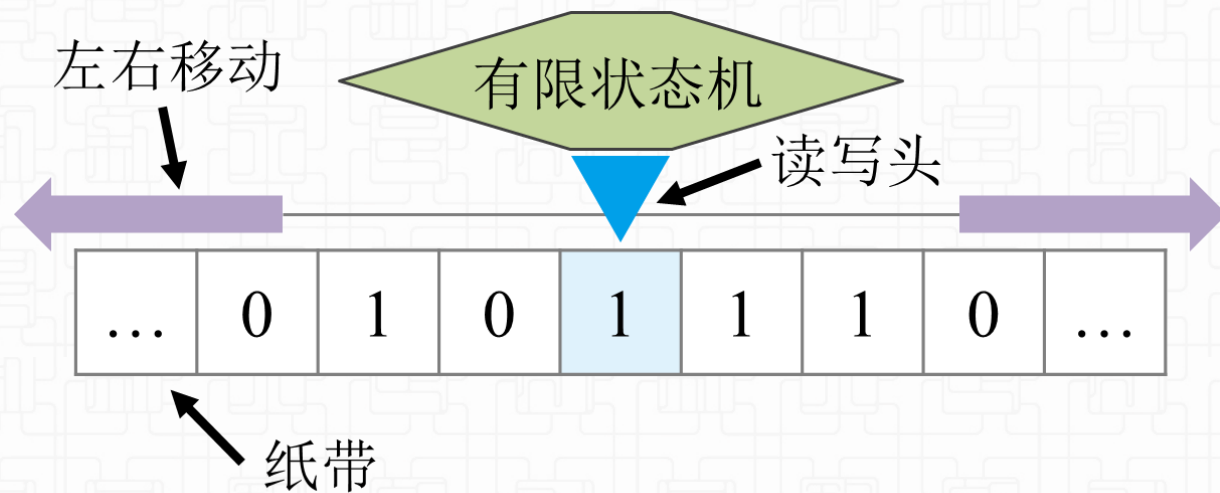
## 图灵机

### ❖ 图灵机主要有以下三部分组成：

- 无限长的纸带，可以比作是计算机的内存。
- 读写头。
- 控制器，可以根据程序中制定的状态转移规则，决定读写头的具体操作。

### ❖ 图灵机可以做下面三个基本的操作：

- 读取指针头指向的符号；
- 修改方框中的字符；
- 将纸带向左或者向右移动，以便修改其临近方格的值。



## 图灵机的意义

- ❖ 证明了**通用计算理论**，肯定了计算机实现的可能性，同时它给出了**计算机应有的主要架构**
- ❖ 图灵机模型引入了**读写与算法、程序语言的概念**，极大的突破了过去计算机器的设计理念
- ❖ 图灵机模型理论是**计算学科最核心理论**，因为计算机的极限计算能力就是通用图灵机的计算能力，很多问题可以转化到图灵机这个简单的模型来考虑。
- ❖ “图灵机”只是**假象的“计算机”**，完全没有考虑硬件状态，考虑的焦点是逻辑结构。
- ❖ 通用图灵机等于向我们展示这样一个过程：
  - 程序和其输入可以先保存到**存储带**上，图灵机就按**程序**一步一步运行直到给出结果，结果也保存在**存储带**上。



## 阿塔纳索夫-贝瑞计算机 (Atanasoff-Berry Computer, 简称ABC计算机)

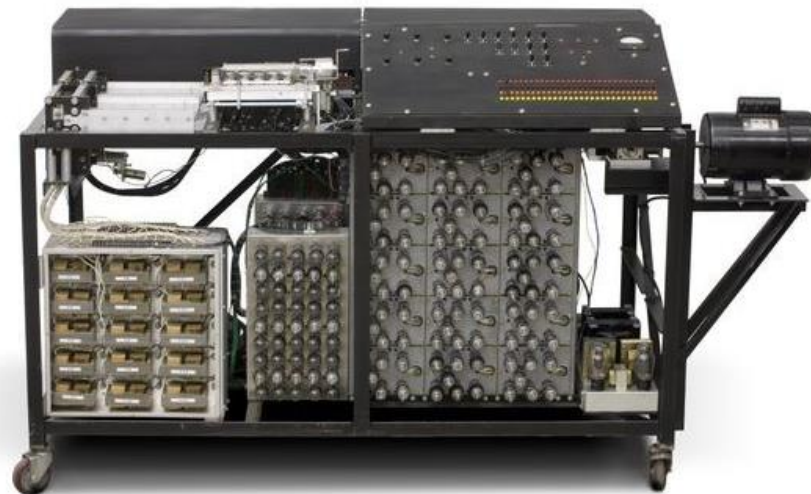
- ❖ 爱荷华州立大学的**约翰·文森特·阿塔纳索夫**和他的研究生**克利福特·贝瑞**在1937—1941年间开发。
- ❖ 不可编程，仅仅设计用于求解线性方程组。ABC计算机开创了现代计算机的重要元素，包括二进制算术和电子开关。但缺乏通用性、可变性与存储程序的机制。



John Vincent Atanasoff  
Oct 04, 1903—Jun 15, 1995



Clifford Edward Berry  
Apr 19, 1918—Oct 30, 1963



- ❖ ABC计算机直到1960年才被发现和广为人知。1973年，美国联邦地方法院注销了埃尼阿克 (ENIAC) 的专利，并得出结论：ENIAC的发明者从阿塔纳索夫那里继承了电子数字计算机的主要构件思想。因此，**ABC被认定为世界上第一台电子计算机。**

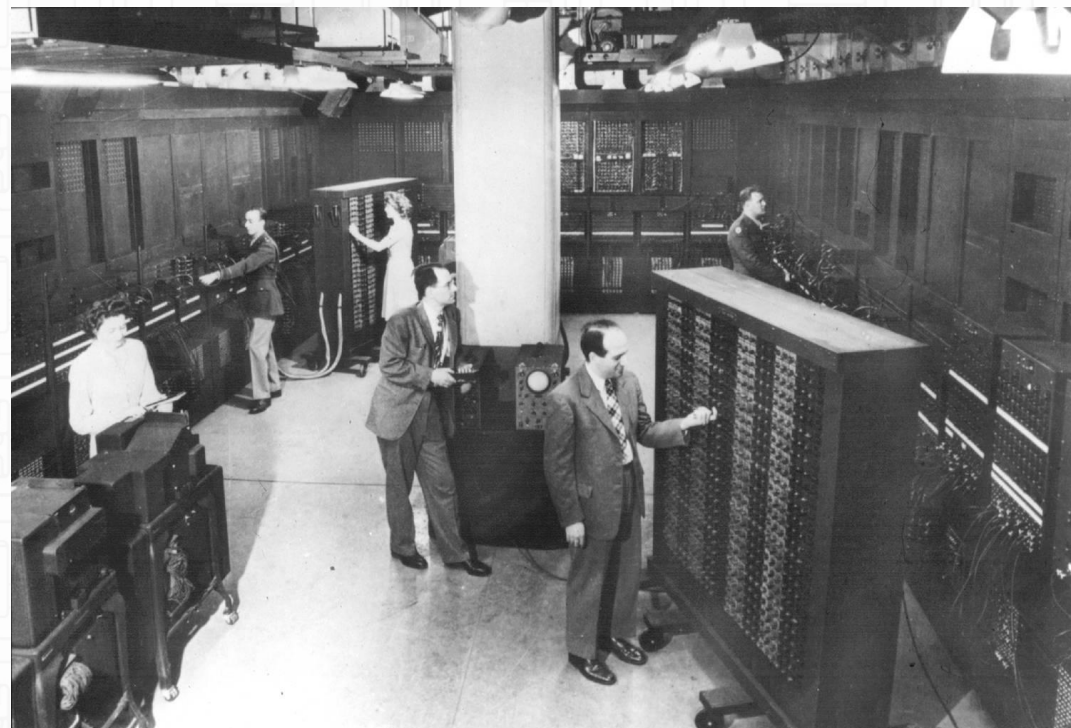
## ENIAC (Electronic Numerical Integrator and Computer, 埃尼阿克)

- ❖ 1946年2月16日, ENIAC在美国宾夕法尼亚大学诞生, 被认为是历史上**第一台通用电子计算机**
- ❖ 每秒5000次加法或500次乘法, 比机电式计算机快1000倍



John Adam Presper Eckert Jr.  
Apr 09, 1919—Jun 03, 1995

John William Mauchly  
Aug 30, 1907—Jan 08, 1980



- ❖ 由18000个电子管, 70000个电阻器, 10000个电容器、1500多个继电器和6000个开关组成
- ❖ 占地140m<sup>2</sup>, 重量约30t, 耗电100kW

## 冯·诺依曼计算机体系结构

### ❖ 冯·诺伊曼发现ENIAC存在的主要问题：

- 用大量的开关和插头来编程
- 用10个电子管表示一位十进制数

### ❖ 改进措施：

- 针对用10个电子管表示一位十进制数，提出应该用二进制来替代
- 针对ENIAC用开关和插头来编程，提出了程序可以以编码形式和数据一起在内存中存储

### ❖ 冯·诺伊曼的工作成为如今广泛采用的冯·诺伊曼体系结构的基础

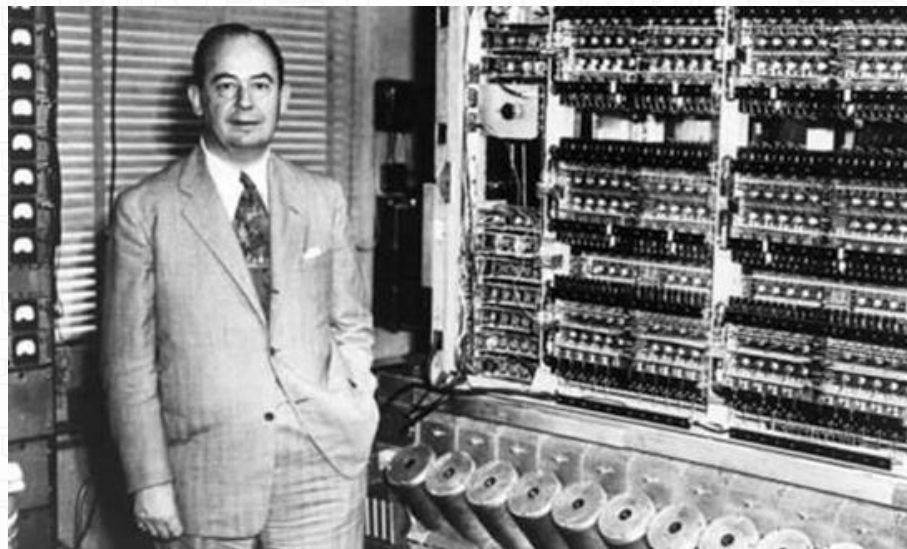
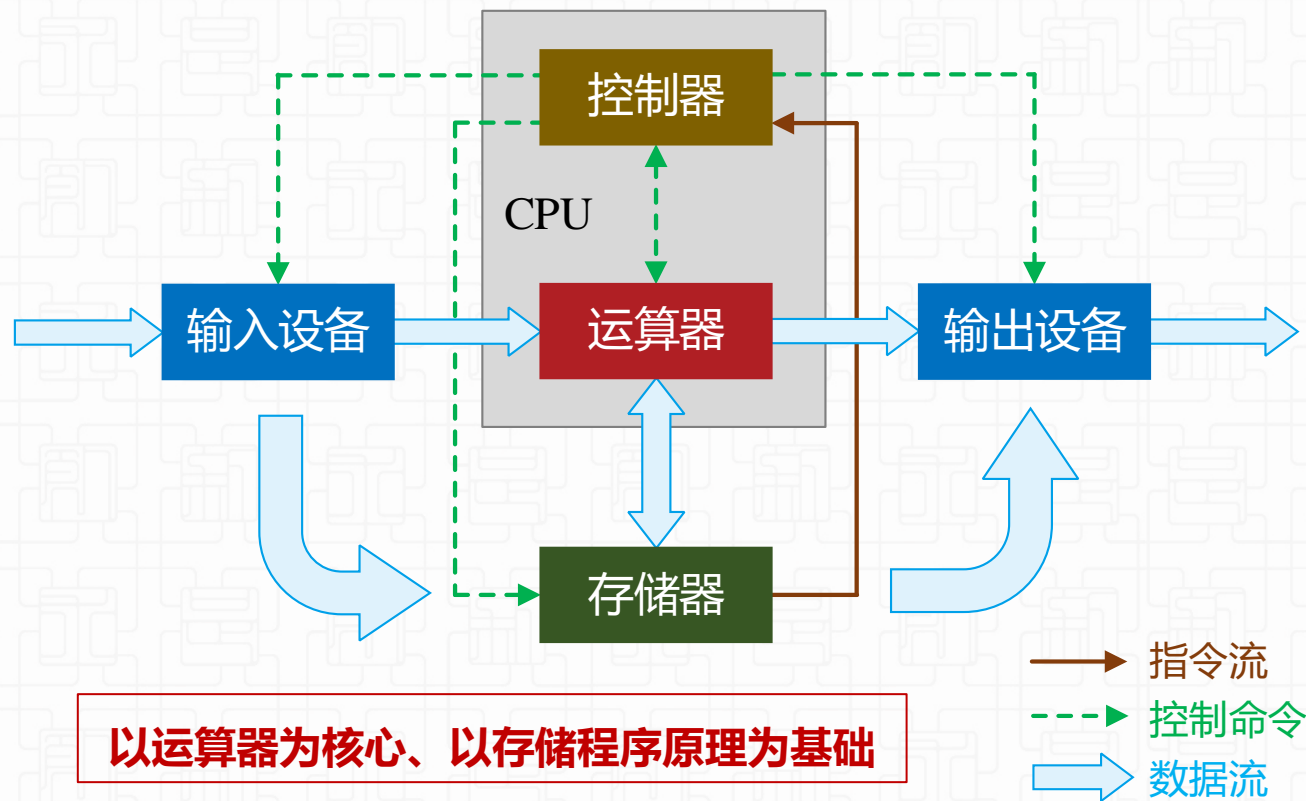
- 将程序与数据存储在了一起
- 依次逐条执行命令
- 用二进制表示信息



John von Neumann  
Dec 28, 1903—Feb 8, 1957



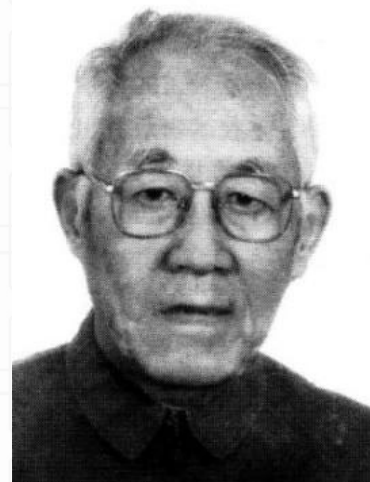
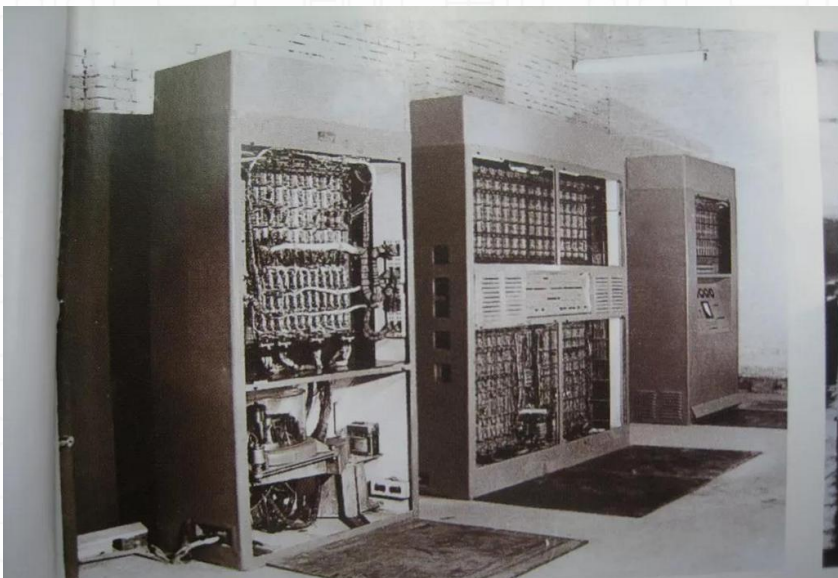
## 冯·诺依曼机



EDVAC于1949年8月交付，1951年才开始运行。是第一台现代意义的通用计算机。6000个电子管和12000个二极管，功率56kW，占地45.5m<sup>2</sup>，重7850kg。

- ❖ 1945年6月，冯·诺依曼提交了他著名的101页的“关于EDVAC的报告草案”，里面描述了计算机的逻辑结构，尤为重要的一点是提出了“存储程序”的思想。
- ❖ “**存储程序 (stored-program)**”的意思就是将程序存储到计算机内部，计算机自动执行。

## 中国第一台数字电子计算机 (103机)



张梓昌

Oct 03, 1921—Sep 21, 2013

- ❖ 1956年制定的《十二年科学技术发展规划》中，就把计算机列为发展科学技术的重点之一，并在1957年筹建了中国第一个计算技术研究所。
- ❖ 在前苏联专家的帮助下，中国科学院计算技术研究所，由七机部张梓昌高级工程师领衔研制的中国第一台数字电子计算机103机（定点32二进制位，每秒2500次）在1958年交付使用。

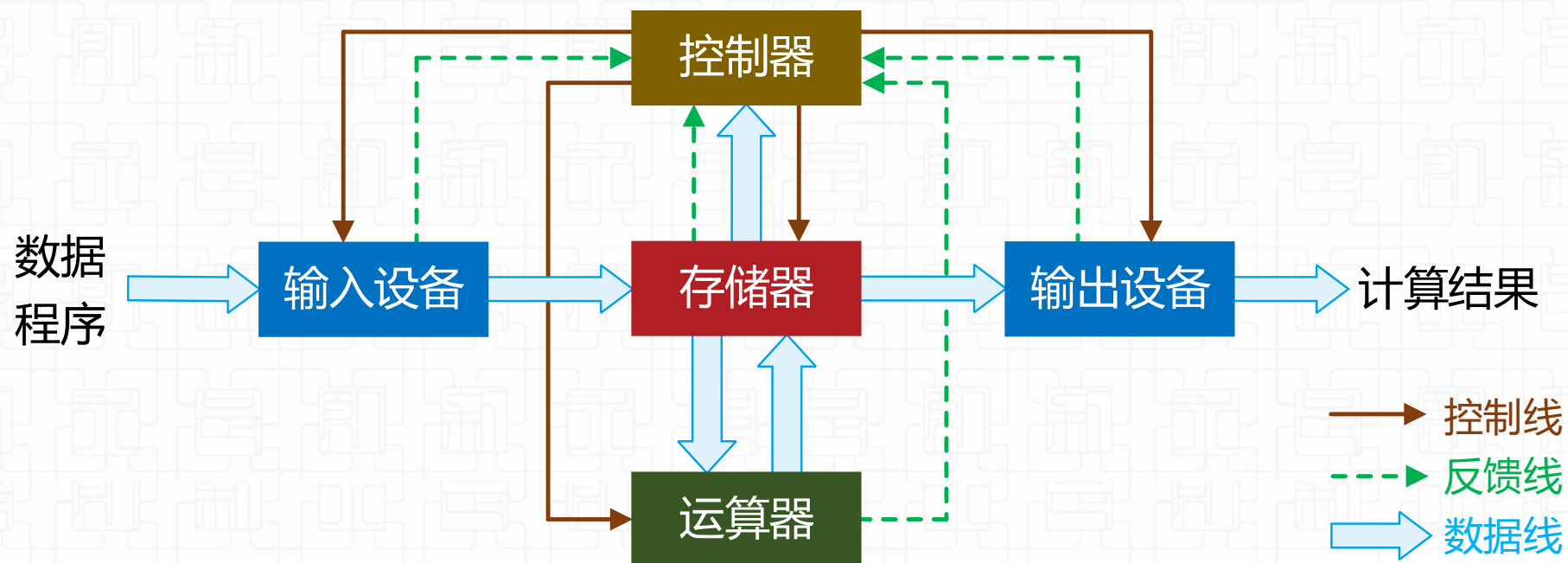
## 电子计算机的发展

发展阶段	逻辑元件	主存储器	运算速度（每秒）	软件	应用
第一代 (1946-1958)	电子管	电子射线管	几千次到几万次	机器语言、汇编语言	军事研究、科学计算
第二代 (1958-1964)	晶体管	磁芯	几十万次	监控程序、高级语言	数据处理、事务处理
第三代 (1964-1971)	中小规模集成电路	半导体	几十万次到几百万次	操作系统、编辑系统、应用程序	有较大发展开始广泛应用
第四代 (1971-至今)	大规模超大规模集成电路	集成度更高的半导体	上千万次到上亿次	操作系统完善、数据库系统、高级语言发展、应用程序发展	渗入社会各级领域



## 现代计算机结构

- ❖ 现代计算机结构以存储器为核心，所有输入的数据/程序以及输出的计算结果，均先存入存储器，然后再送往CPU 进行执行或送至输出设备。



- ❖ 现代计算机结构有效地为CPU 减负，让CPU 更加专注地进行指令执行，大幅提高了效率。

# 计算机语言

## ❖ 高级语言

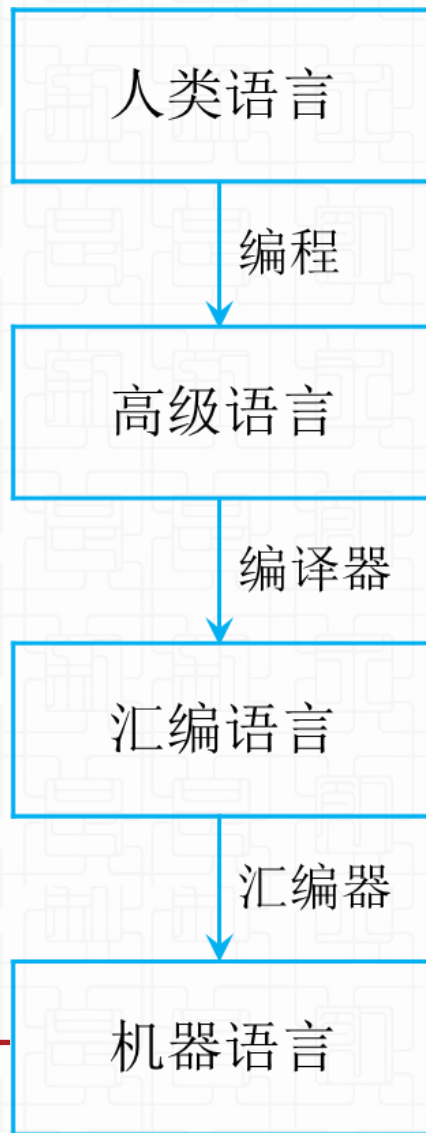
- 更加符合人的习惯，与机器结构无关。

## ❖ 汇编语言

- 用有意义的助记符代替难以理解和记忆的指令字。

## ❖ 机器语言

- 由0和1组成的指令序列，不同型号的计算机的机器语言是不相通的。



交换  $v[k]$  和  $v[k+1]$  的值

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw R15, 0(R2)
lw R16, 4(R2)
sw R16, 0(R2)
sw R15, 4(R2)
```

```
10001100011000100000000000000000
100011001111000000000000000000100
10101100111100100000000000000000
101011000110001000000000000000100
```

## 通用微处理器示例程序

❖ 首先输入一个数据到A，并输出。然后递减这个数据，并输出，直到该数据递减到0。

汇编代码:

	IN A	-- 输入一个数据至A寄存器
Loop:	OUT A	-- 从A寄存器输出
	DEC A	-- A寄存器数据减1
	JNZ loop	-- 如果A不为零转至loop
	HALT	-- 停止



把每个指令换成对应的二进制编码，即完成了编译的过程。

二进制可执行代码:

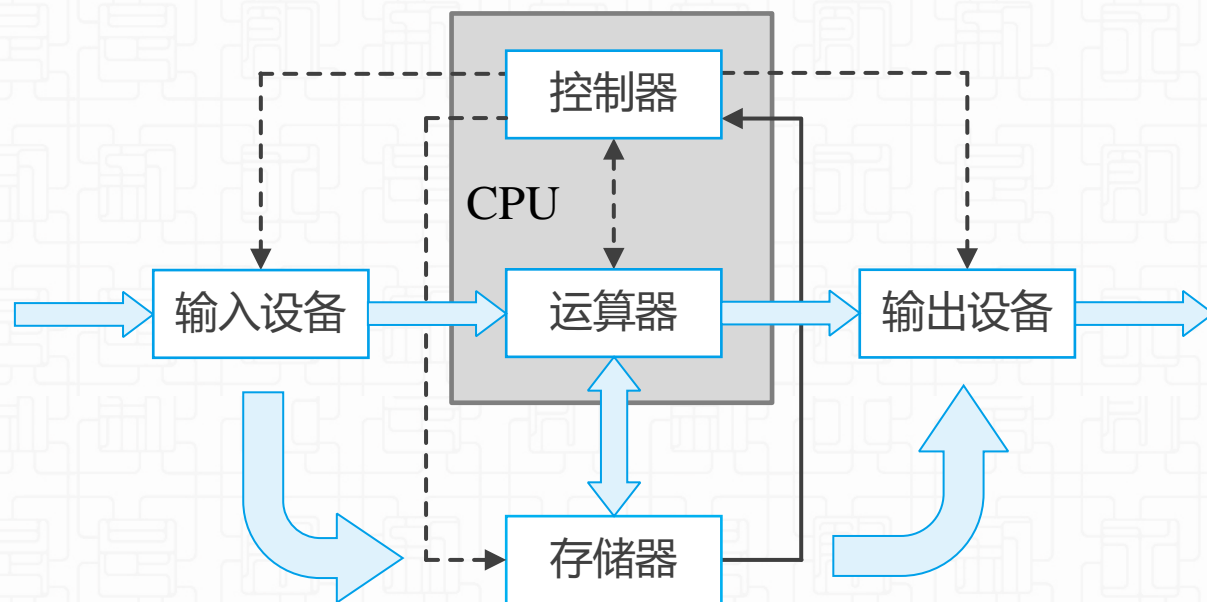
<u>memory</u> <u>address</u>	<u>instruction</u> <u>encoding</u>	
0000	01100000;	-- IN A
0001	10000000;	-- OUT A
0010	10100000;	-- DEC A
0011	11000001;	-- JNZ loop
0100	11111111;	-- HALT



## 内容提要

❖ 计算机概述

❖ 微处理器



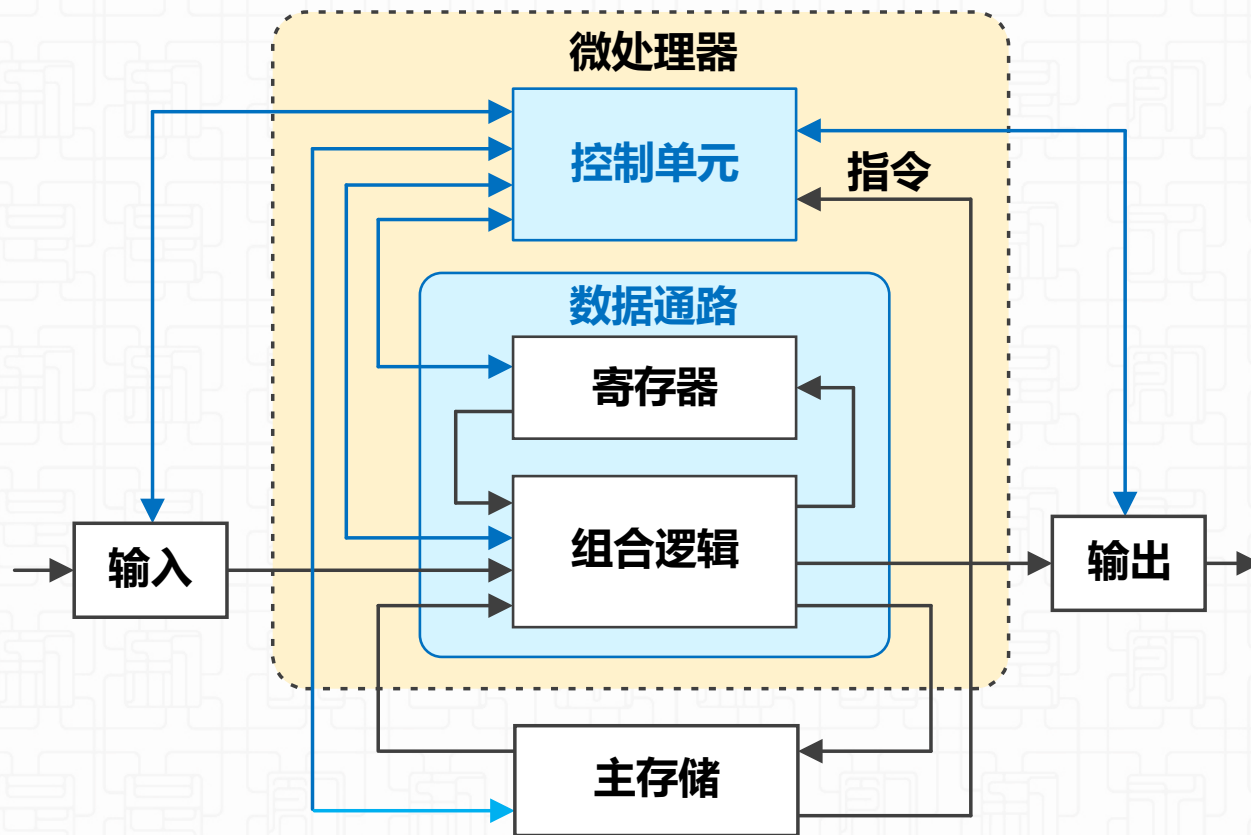
## 微处理器结构

### ❖ 数据通路

- 负责微处理器对数据操作的执行。主要包括三部分：
- 运算功能单元，如加法器，比较器等
- 寄存器和存储单元
- 总线、三态门、选择器等连接单元

### ❖ 控制单元

- 通过控制数据通路中各个功能单位的使能，控制整个微处理器的操作，通常由一个有限状态机构成。
- 在某一个时钟周期内完成哪一个操作，由控制单元的输出信号决定。



## 总线

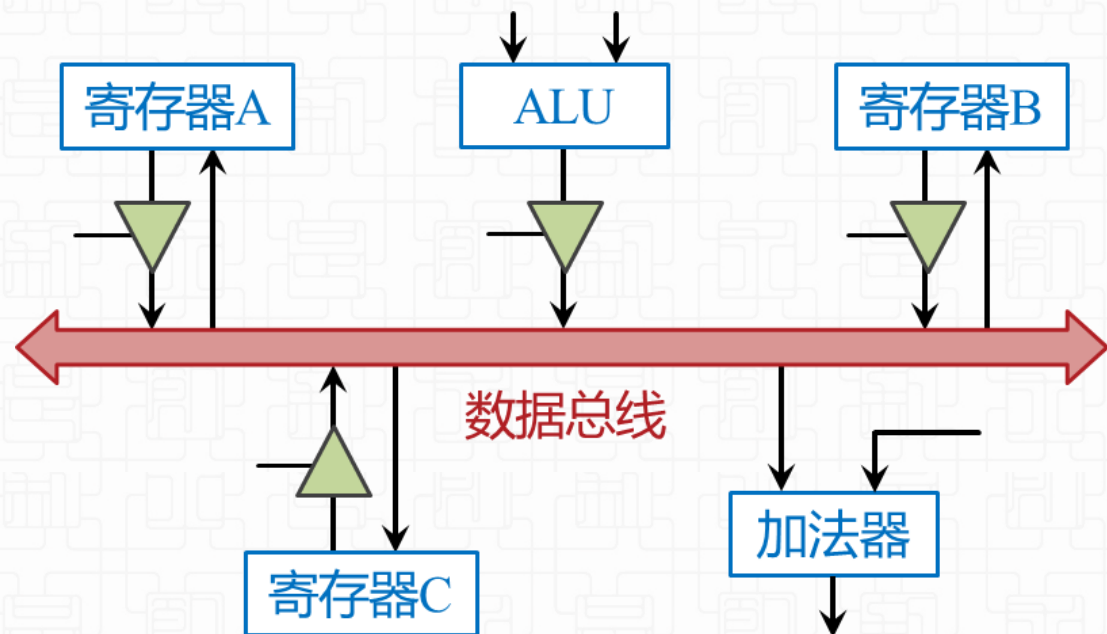
❖ 总线 (Bus) 是一种内部结构，是由导线组成的传输线束，是各功能部件之间传送数据的公共通道，各个部件通过总线相连接。

❖ 总线分为数据总线、地址总线和控制总线三类

- 数据总线用于传送数据
- 控制总线用于传送控制信号和时序信号
- 地址总线则用于选择存储单元或外设

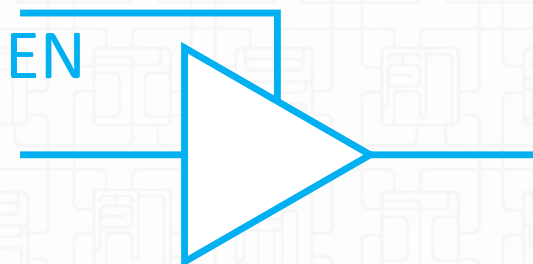
❖ 总线按功能又可分为

- 片内总线
- 系统总线
- 通信总线

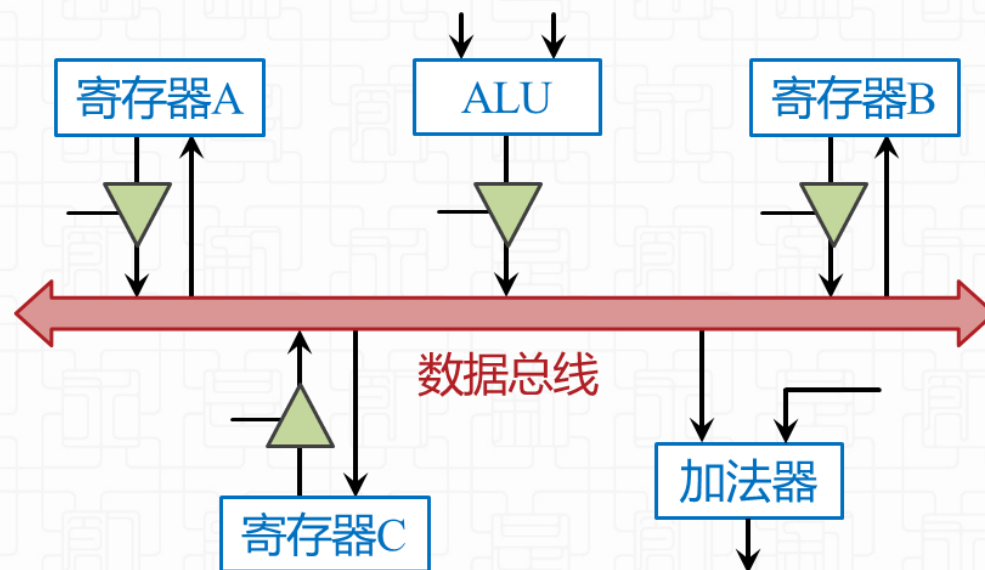




## 三态门



- ❖ 三态门是一种重要的总线接口电路。
- ❖ 三态门都有一个控制使能端EN
  - 当EN有效时，三态电路呈现正常高电平或低电平的输出。
  - 当EN无效时，三态电路呈现高阻态，输出电阻很大，相当于开路，主要用来将逻辑门和系统的其他部分加以隔离。
- ❖ 通常在数据总线上接有多个器件，因为总线只允许同时只有一个使用者，每个器件通过EN 之类的信号选通。如器件没有被选通的话它就处于高阻态，相当于没有接在总线上，这样就不会影响其他器件的工作。



## 专用微处理器数据通路设计示例

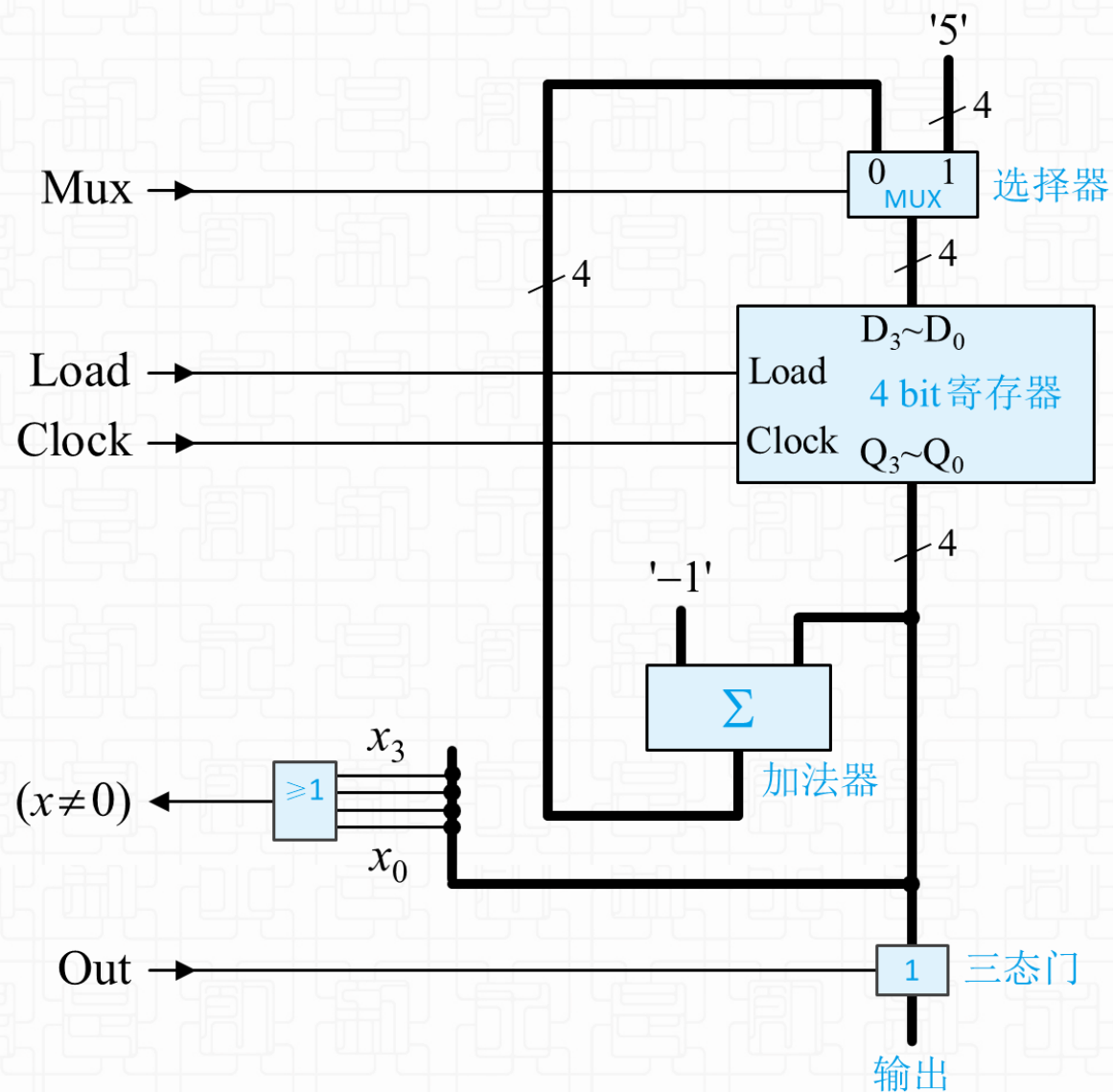
```
x = 5  
WHILE (x ≠ 0) {  
    OUTPUT x  
    x = x - 1  
}
```

- ❖ 设计一个可以完成所示一段伪代码的数据通路。
- ❖ 由于只有  $x$  一个变量，所以只需要一个寄存器，选择 4 bit 的寄存器。
- ❖ 由于有数据运算，因此需要一个加法器。
- ❖ 考虑到要进行输入输出，需要有输入输出总线。要控制是否输出，需要有一个三态门。
- ❖ 程序有循环，需要一个数据选择器。
- ❖ 判断条件 “ $x \neq 0$ ” 是否满足，需要一个判断电路，并有反馈给控制单元的信号。

## 一段伪代码的数据通路

```
x = 5
WHILE (x ≠ 0) {
  OUTPUT x
  x = x - 1
}
```

指令	Mux	Load	Out
$x = 5$	1	1	0
OUTPUT $x$	×	0	1
$x = x - 1$	0	1	0

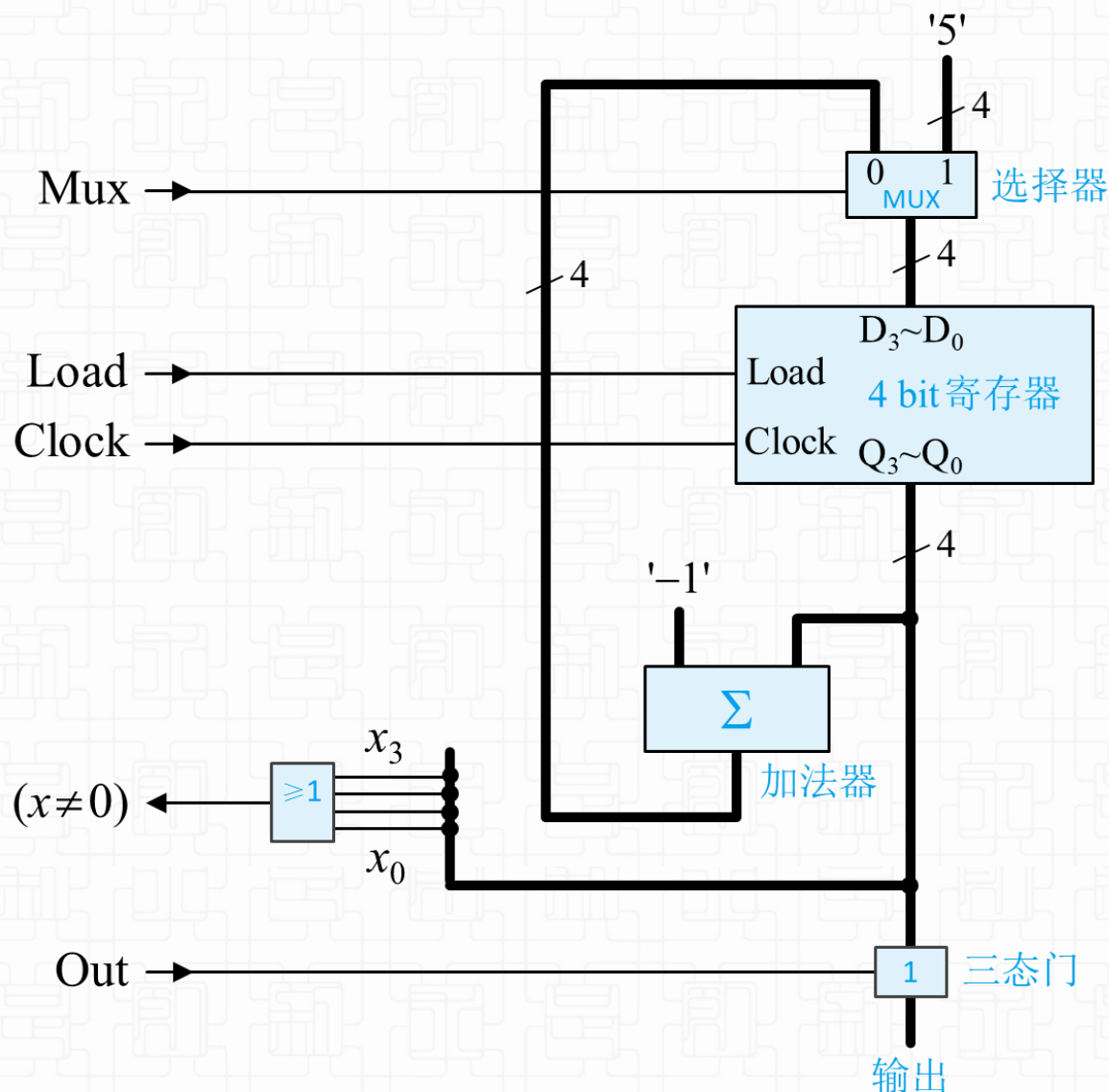




## 控制单元设计

```
x = 5
WHILE (x ≠ 0) {
  OUTPUT x
  x = x - 1
}
```

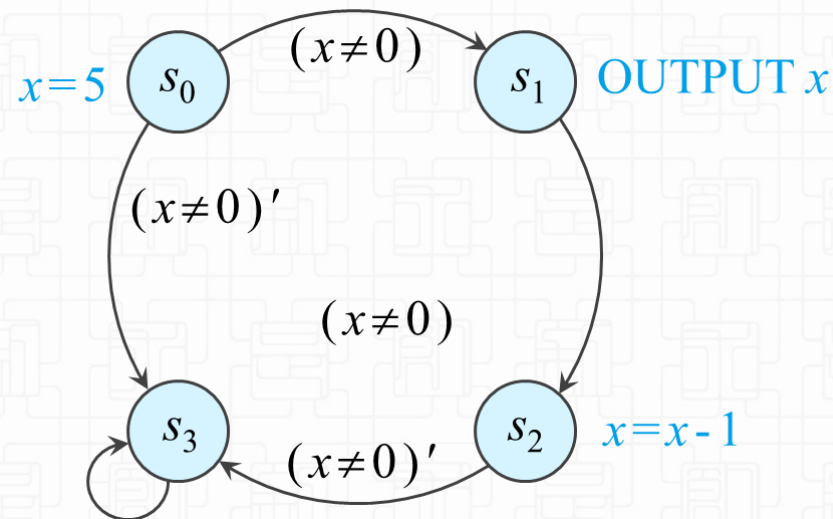
- ❖ 控制单元是一个状态机
- ❖ 状态机每个不同状态时候的输出，决定了数据通路的操作。
- ❖ 在数据通路中，**Mux**、**load**、**Out**等控制信号就是由状态机每个状态输出决定的。
- ❖ 每个状态下输出控制信号的组合，就是**控制字**。



## 次态逻辑方程

- ❖ 根据伪代码画出状态图
- ❖ 列出对应于每个操作（某个状态）的控制字（状态机输出信号）
- ❖ 由状态图得到状态机次态逻辑电路表达式

```
x = 5
WHILE (x ≠ 0) {
    OUTPUT x
    x = x - 1
}
```

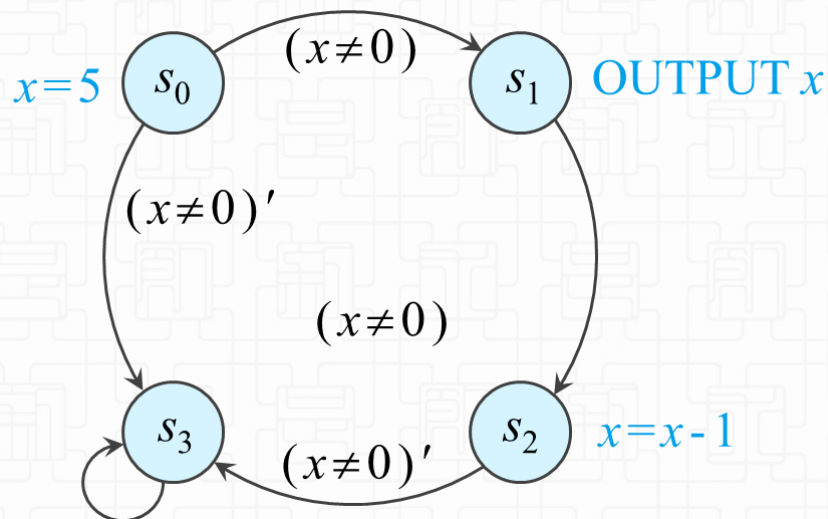


当前状态		次态 $Q_{1next} Q_{0next}$	
名称	$Q_1 Q_0$	$(x \neq 0)$	$(x \neq 0)'$
$s_0$	00	01	11
$s_1$	01	10	10
$s_2$	10	01	11
$s_3$	11	11	11

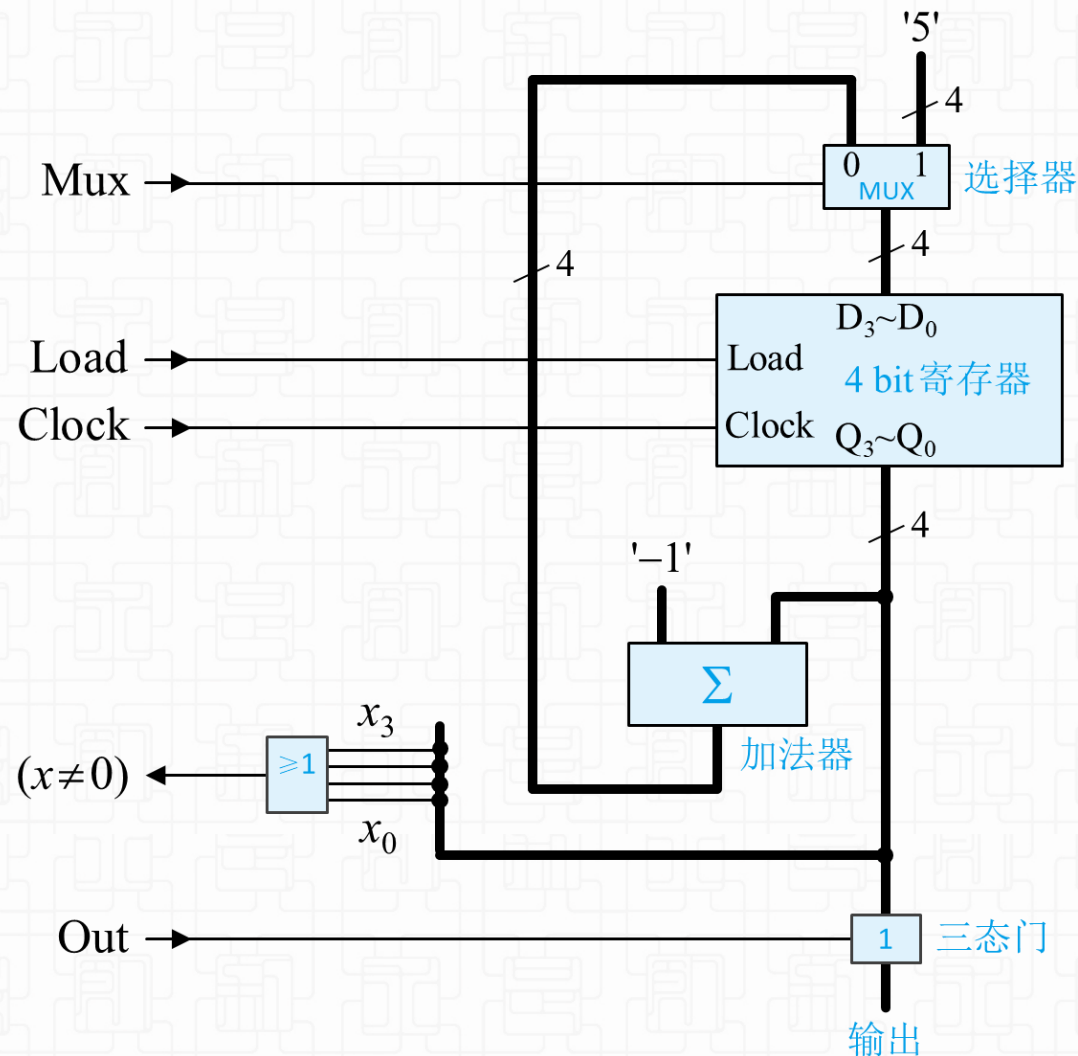
$$Q_{1next} = D_1 = Q_0(x \neq 0) + (x \neq 0)'$$

$$Q_{0next} = D_0 = Q_0' + Q_1 Q_0$$

## 控制字 (状态机的输出是一个需要重点关注的问题)



状态	控制字	$Q_1Q_0$	指令	Mux	Load	Out
$s_1$	1	00	$x = 5$	1	1	0
$s_2$	2	01	OUTPUT $x$	×	0	1
$s_3$	3	10	$x = x - 1$	0	1	0
$s_4$	4	11	无操作	×	0	0





## 输出逻辑方程

❖ 根据每个状态的控制字写出对应的输出逻辑真值表和表达式

状态	控制字	$Q_1Q_0$	指令	Mux	Load	Out
$s_1$	1	00	$x = 5$	1	1	0
$s_2$	2	01	OUTPUT $x$	×	0	1
$s_3$	3	10	$x = x - 1$	0	1	0
$s_4$	4	11	无操作	×	0	0

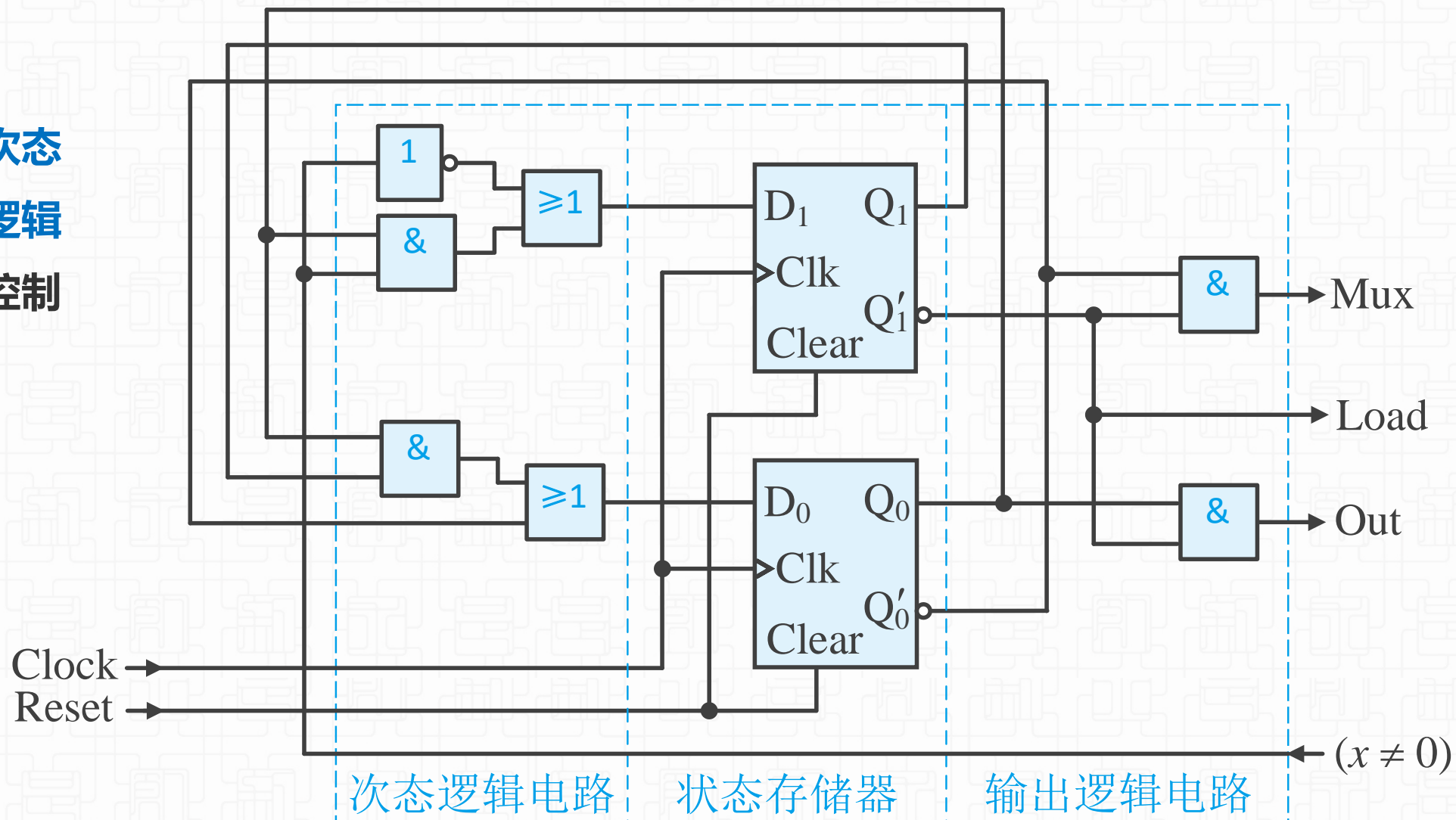
$$\text{Mux} = Q_1'Q_0'$$

$$\text{Load} = Q_0'$$

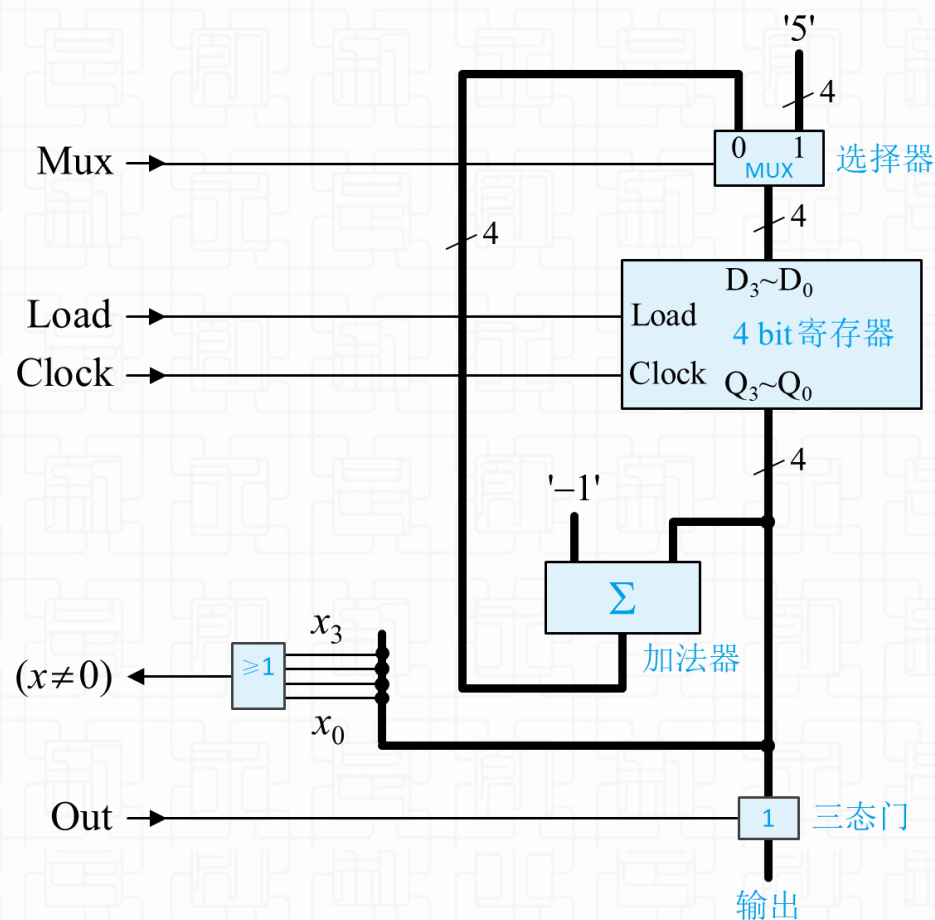
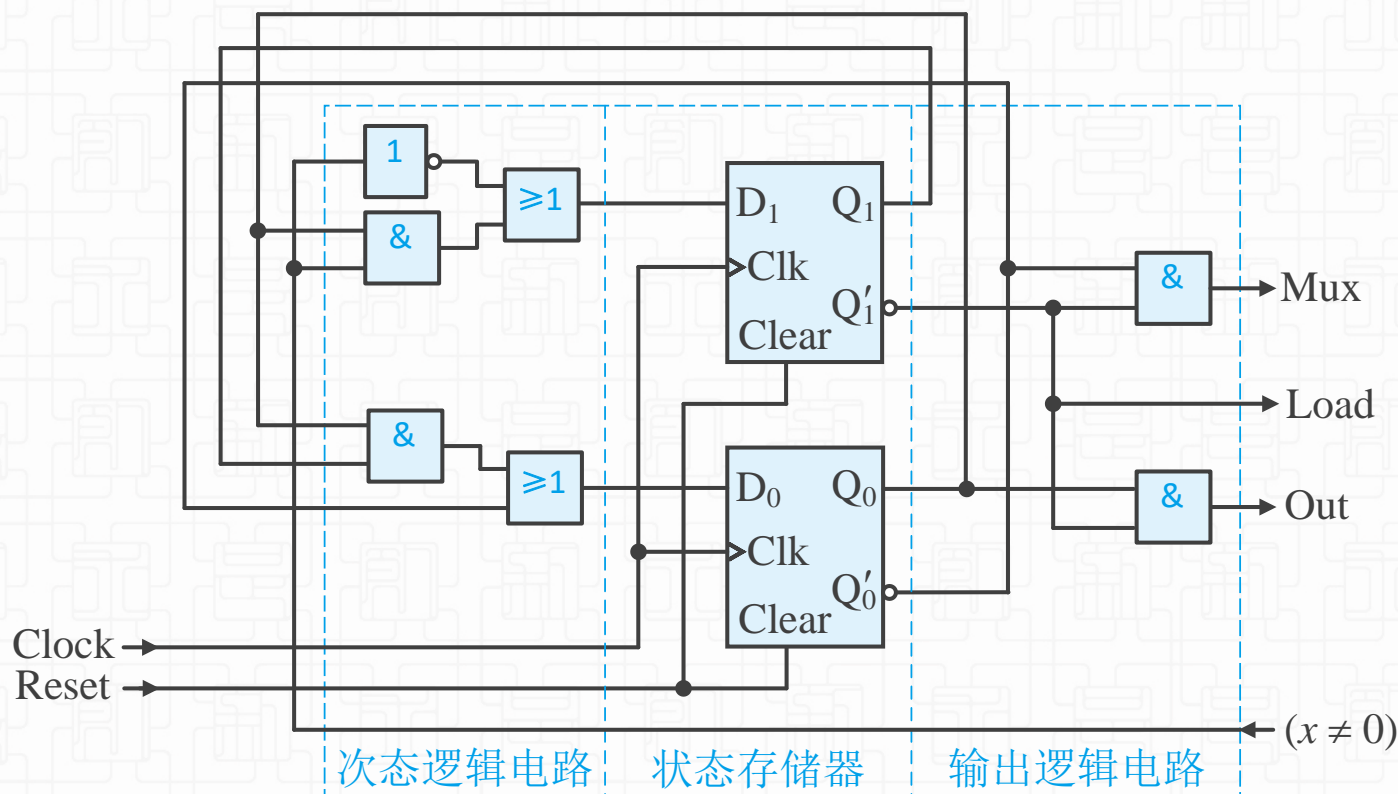
$$\text{Out} = Q_1'Q_0$$

## 控制单元

❖ 根据**状态数目、次态逻辑方程、输出逻辑方程**，可以得到控制单元的电路。



## 完整的专用微处理器电路



❖ 将控制单元和数据通路合起来，就能得到完整的电路。

❖ 微处理器中最关键的是控制单元给数据通路的控制信号和数据通路给控制单元反馈的状态信号。



# 指令执行

❖ 如何知道从哪里取出程序的第一条指令？

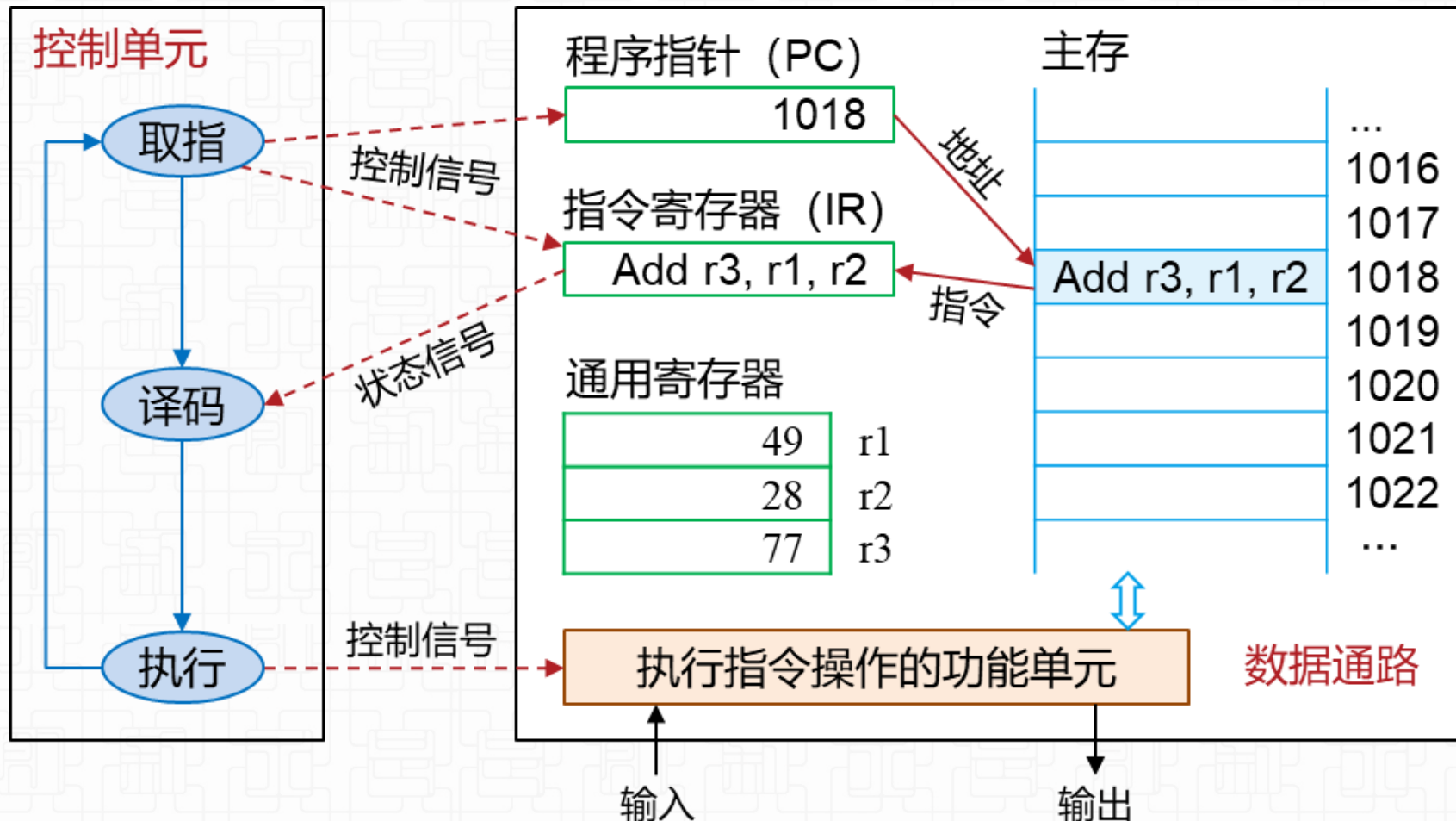
— 操作系统

❖ 如何按程序控制流执行指令？

— 程序计数器

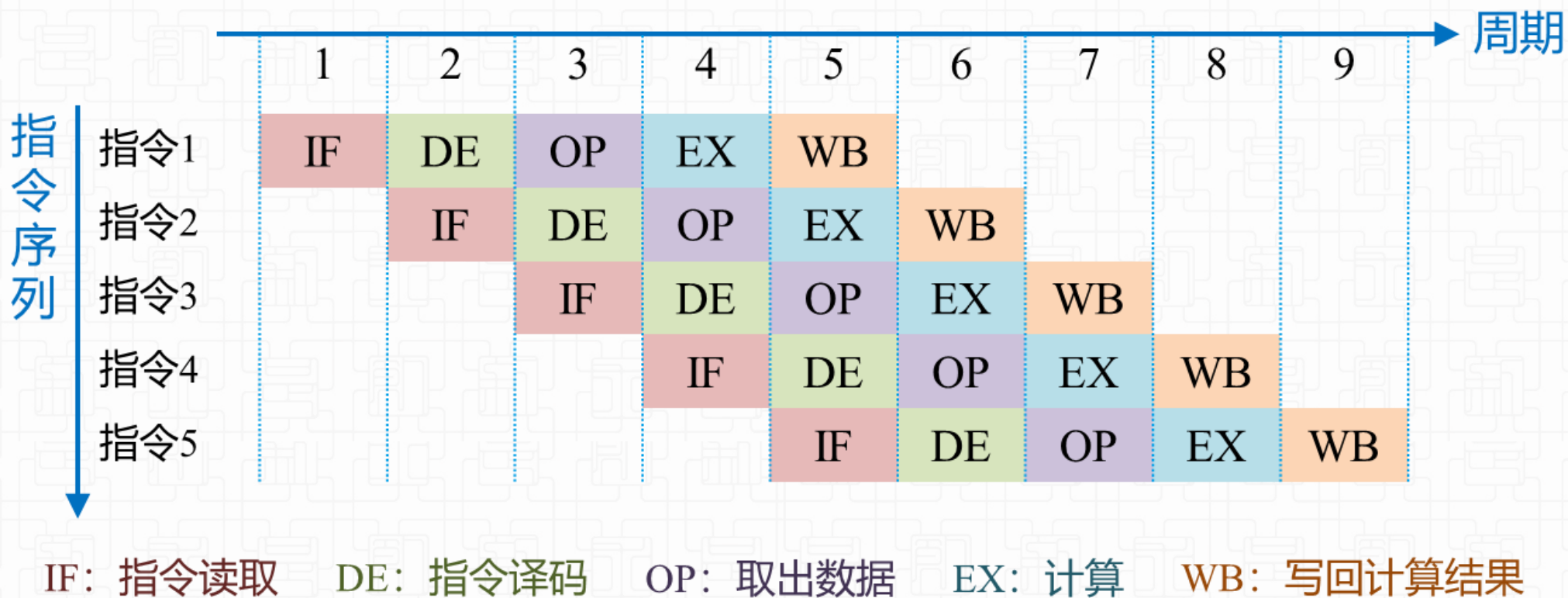
❖ 如何知道从哪里取操作数？

— 地址、寻址方式



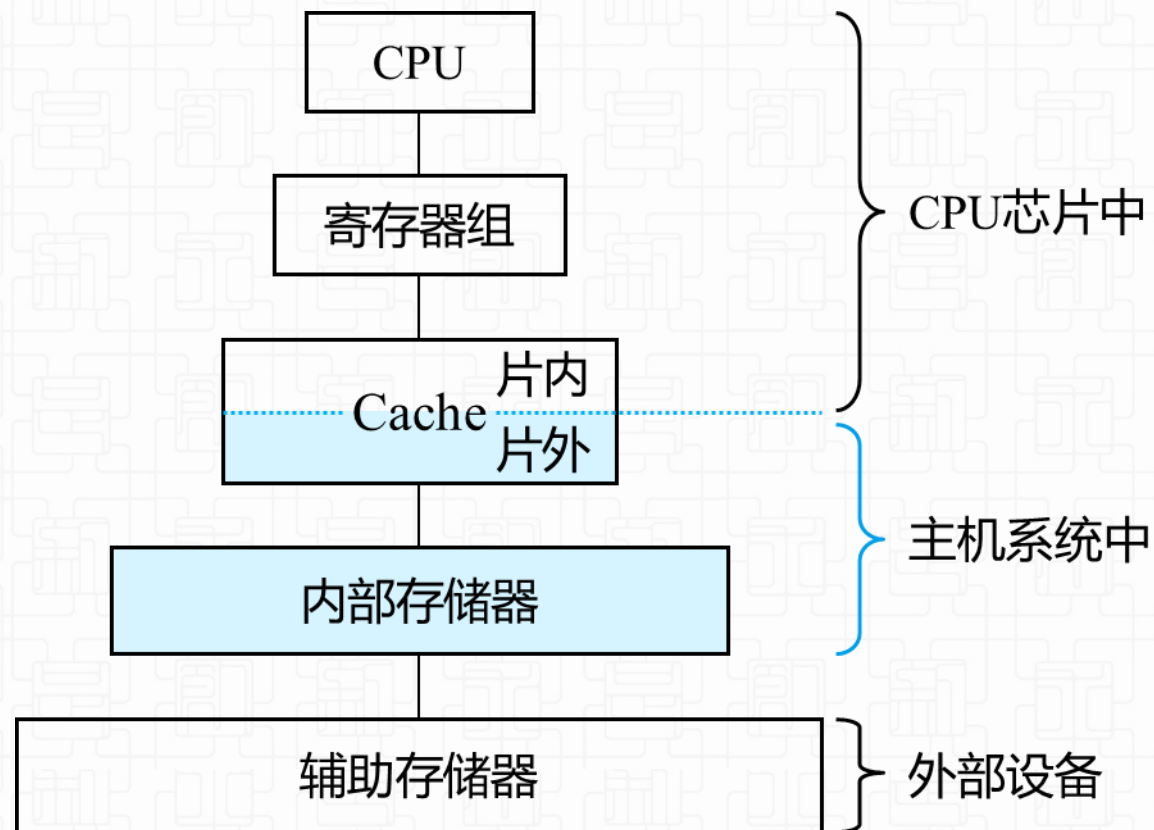
## 指令流水

❖ 改变各条指令按顺序串行执行的规则，使机器在执行上一条指令的同时取出下一条指令，即上一条指令的执行周期和下一条指令的取指周期同时进行。



## 存储器

- ❖ 存储器包含**Cache**、**主存**和**外存**
- ❖ **Cache**是一种容量小、速度快的存储器阵列
- ❖ **主存**是微处理器能直接访问的寄存器，用来存放系统和用户的程序及数据
- ❖ 常用作主存的存储器分为**只读存储器 (ROM)**和**随机存取存储器 (RAM)** 两类。





## 指令集架构 (Instruction Set Architecture, ISA)

❖ 指令集架构定义了微处理器的指令集，即**微处理器能够执行的指令的集合**。

- 指令集架构主要规定了指令格式、寻址访存、数据类型、寄存器等
- 通常包括三大类主要指令类型：运算指令、分支指令和访存指令。

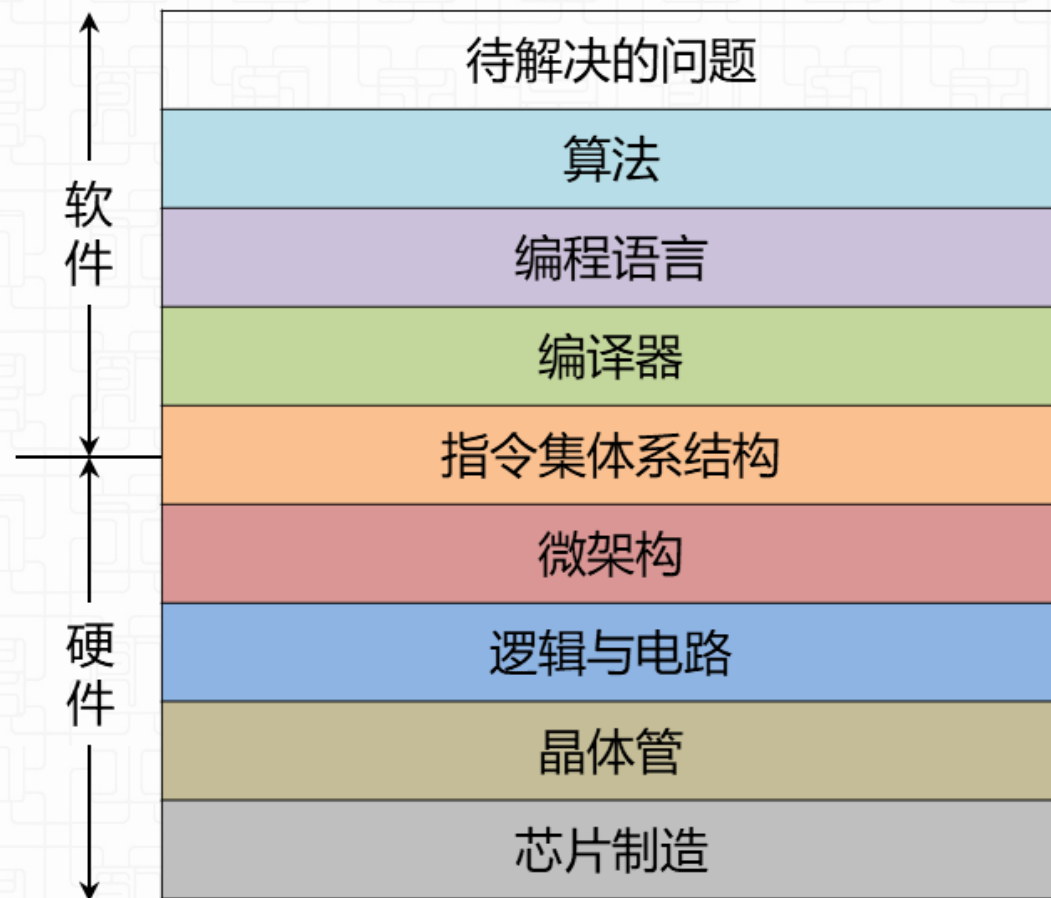
❖ 从现阶段的主流体系结构讲，指令集可分为复杂指令集和精简指令集：

- **复杂指令集** (Complex Instruction Set Computing, **CISC**) 侧重于硬件执行指令的功能性，其对应的硬件结构很复杂。主要应用于计算机的处理器，个人计算机处理器用的是x86。
- **精简指令集** (Reduced Instruction Set Computing, **RISC**) 侧重于结构简单、处理速度更加快速上。主要用于嵌入式处理器上，主要有ARM 指令集、MIPS 指令集和RISC-V 指令集等。



## 指令集架构和微架构

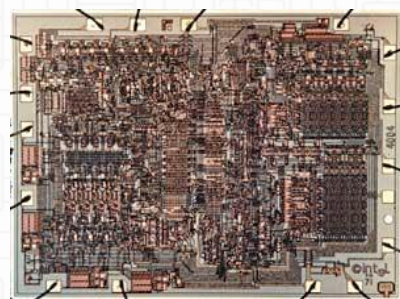
- ❖ **微架构**又称为微体系结构或微处理器体系结构，是实现指令集架构的**硬件电路结构**，包括处理器的组成部分和对指令集架构的连接和操作。
- ❖ **同一指令集可以有不同的微架构**，因为同一指令可以通过不同的电路单元或组合来实现。
- ❖ **指令集架构**为汇编语言的编程者和编译器所见，是**软件与硬件之间的接口**，使得软件可以被编写、编译和运行在计算机上。



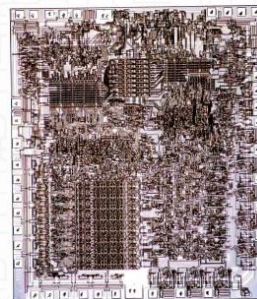


# Intel 微处理器

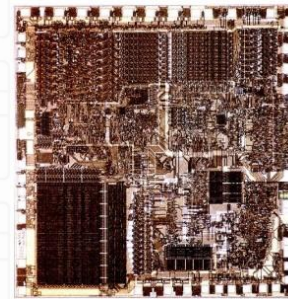
**INTEL 4004**  
The first microprocessor



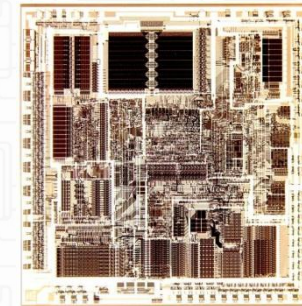
1971年, 4004  
10 $\mu$ m, 2300个MOS



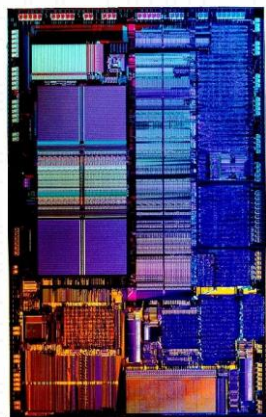
1974年, 8080  
6 $\mu$ m, 6000个MOS



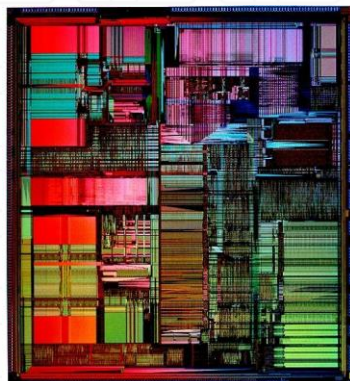
1978年, 8086  
3 $\mu$ m, 2.9万个MOS



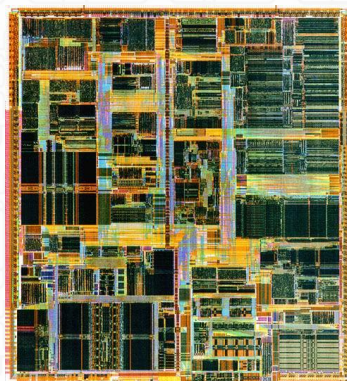
1982年, 80286  
1.5  $\mu$ m, 13.4万个MOS



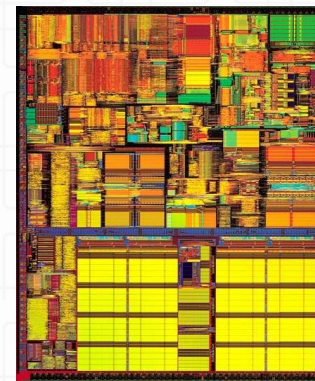
1989年, 80486  
1  $\mu$ m, 120万个MOS



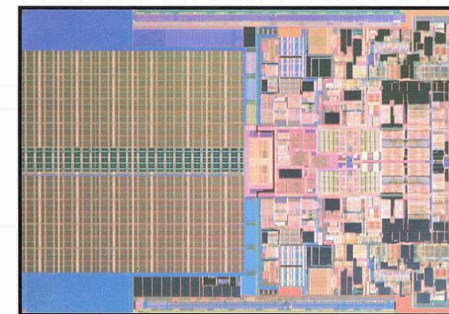
1993年, Pentium  
0.8  $\mu$ m, 310万个MOS



1997年, Pentium II  
0.35  $\mu$ m, 750万个MOS



1999年, Pentium III  
0.18  $\mu$ m, 2800万个MOS



2008年, Core 2 Quad  
45nm, 8.2亿个MOS

## 龙芯一号

- ❖ 2002年8月10日诞生的“**龙芯一号**”，我国首枚拥有自主知识产权的通用高性能微处理芯片。
- ❖ IP核是兼顾通用及嵌入式CPU特点的32位处理器内核，主频**266 MHz**。采用**0.18 $\mu$ m CMOS**工艺制造。
- ❖ 龙芯的诞生，打破了国外的长期技术垄断，结束了中国近二十年无“芯”的历史。
- ❖ 龙芯从2001年至今共开发了**1号、2号、3号**三个系列处理器和龙芯桥片系列，在政企、安全、金融、能源等应用场景得到了广泛的应用。
  - 龙芯1号系列为32位低功耗、低成本处理器，主要面向低端嵌入式和专用应用领域；
  - 龙芯2号系列为64位低功耗单核或双核系列处理器，主要面向工控和终端等领域；
  - 龙芯3号系列为64位多核系列处理器，主要面向桌面和服务器等领域。





# The End.



中国大学MOOC

章献民

zhangxm@zju.edu.cn