



第1章 微型计算机基本结构及信息表示

课程导引

1.1 微型计算机概述

1.2 微机原理的实践选型

1.3 数制及数制之间的转换方法

1.4 计算机中信息的基本表示方式

1.5 文字在计算机中的存储方式—字符编码



课程导引

1. 学什么？

课程主要目的在于从底层开始理解微型计算机是如何工作的，属于计算机、人工智能、电子信息、自动化等理工类本科专业的专业基础课程。

在理论层面，要理解微型计算机的基本工作原理；

在实践层面，要运用直接与硬件打交道的汇编语言进行编程，理解计算机程序基本运行过程。

本课程学习微型计算机的基本结构、信息表示、系统时钟、三总线、硬件系统、指令系统、汇编语言框架和汇编程序设计方法，学习微型计算机的存储器、串行通信接口、中断系统、定时器、模数与数模转换、直接存储器存取等，并可以通过汇编语言编程体会其中的工作过程，为高级语言程序设计、微型计算机应用系统软硬件设计、嵌入式人工智能等提供知识基础。



2. 为什么要学？

由于本课程是运用面向机器的汇编语言，从底层透明理解微型计算机运行的基本原理及其与外界的基本接口方式，属于计算机类、电子类、自动化类、人工智能类等专业的基础性课程，是功底性课程，因此通过本课程的学习，可为计算机应用、软件编程、软硬件协同开发等打下坚实基础。

3. 如何学？——“勤”

- (1) 学习过程：**预习**——花20分钟左右浏览课本；**听课**——准备好笔、纸质笔记本或草稿纸，脑手联动；**课后**——及时仔细看书、梳理笔记、完成作业，当天完成当天任务；**实验**——勤动手；**期中、期末复习**——收拢知识
- (2) 作业及时提交，实验报告及程序及时上传学院教学系统。
- (3) **勤问**：学问学问，一要**学**，二要**问**，要克服“不问”之惯性，问老师、问同学



4. 教学用书

王宜怀等，微型计算机原理及应用—基于RISC-V架构（讲义），2025.

5. 实验套件

AHL-MCP-RISC-V （借给同学们）

6. 网上电子资源

通过群发给同学们



1.1 微型计算机概述

1.1.1 初识微型计算机

1. 世界上第一台电子计算机

1946年，诞生了世界上第一台电子数字计算机（ENIAC），它由**美国宾夕法尼亚大学莫尔电工学院**制造，重达30t，总体积约90m³，占地170m²，耗电140kW，运算速度为**每秒5000次加法**，标志着计算机时代开始，最初主要是军方用于弹道计算。





2. 计算机的种类

早期根据运算速度、体积大小、存储器大小、输入输出能力等指标，分为**巨型机、大型机、小型机、微型机**等四个种类。今天，微型计算机向更广泛的应用拓展。性能介于巨型机（超级计算机）与微型机中间的计算机，其性价比不断提高，目标定位是服务于企业、网络及通信等领域。

1) 超级计算机

超级计算机主要特点是**高速度和大容量**，我国的太湖之光运算速度达到**12亿亿次/秒**，美国的Summit运算速度达到**20亿亿次/秒**。有的需要超过足球场大小的机房，有的耗电超过一个县城用电量。主要用于**科学与工程计算**，服务于中长期天气预报、卫星图像处理、大数据处理等领域。





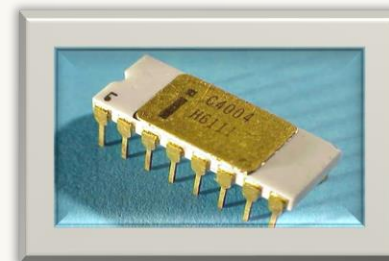
2) 微型计算机

从1946年计算机发明到1971年第一个微处理器的出现，整整过去了25年。

微型计算机是以微处理器为核心，以地址总线、数据总线、控制总线为基础，连接内存储器、输入/输出接口电路，以及相应的辅助电路而形成种类繁多应用广泛的电子计算机。

把微机集成在一个芯片上被称为嵌入式微型计算机，它不以独立的计算机面目出现在人们视野中，而是隐含在各类电子产品中如手机、平板电脑等等。

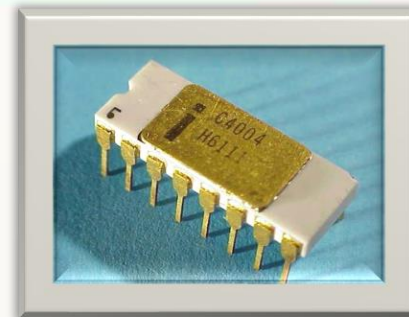
在微型计算机的基础上，配以相应的其他专用电路，以及电源、显示器、机箱，配备操作系统、高级语言和多种软件工具而形成的系统，叫做微型计算机系统，个人计算机（Personal Computer，PC）就是最常见的微型计算机系统，简称PC机。





1.1.2 微型计算机发展简史

1. 微型计算机的开端



1971年，Intel推出了世界上第一个商用4位微处理器有45条指令，速度约0.05**MIPS**（**Million Instructions Per Second**，每秒百万条指令）。

2. 微型计算机的初步发展

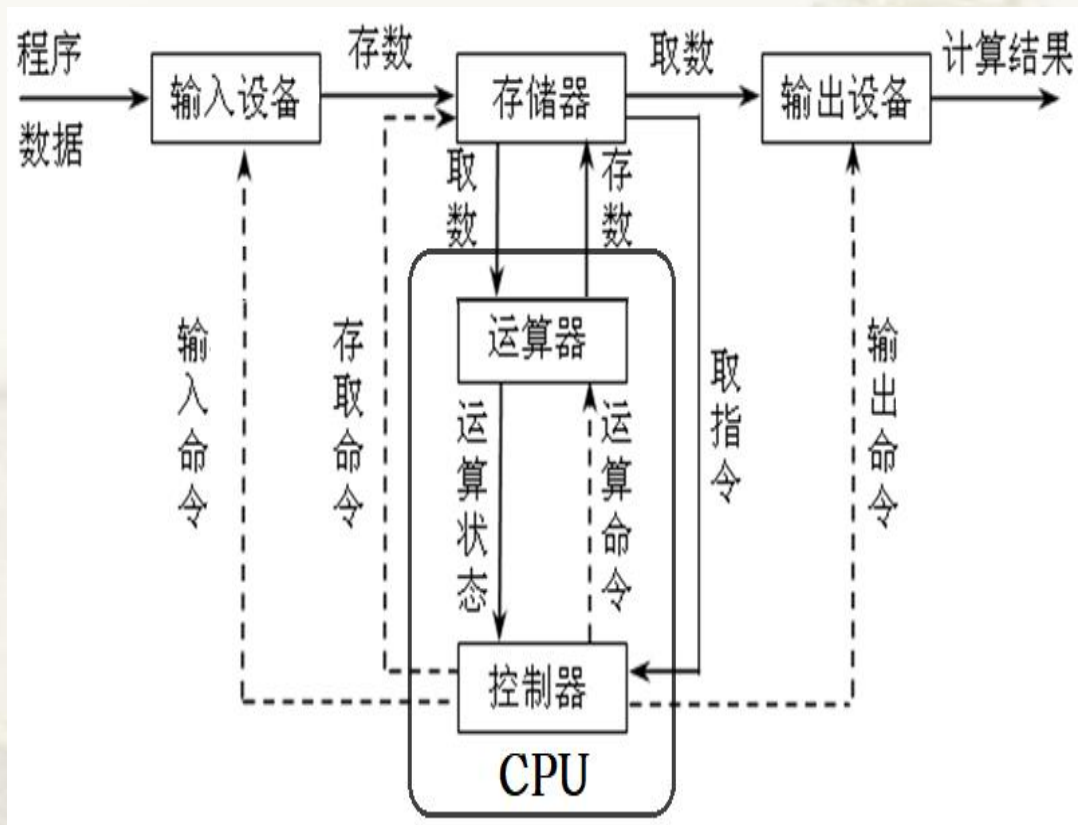
1974-1984，Intel的8080、**8086**，**MCS-51**等，有两条线路：**通用计算机**与**嵌入式计算机**两种模式。Arm出现，改变了嵌入式计算机发展模式。目前**RISC-V**正在逐步推广。

3. 微型计算机的无处不在

可以举出无数微型计算机应用的例子



1.1.3 微型计算机的冯·诺依曼结构框图（重点）



1. CPU:运算器与控制器和在一起，称为中央处理器CPU。它是从外部设备获得数据，通过加工、处理，再把处理结果送到CPU的外部世界。从功能角度看，CPU包含运算器、寄存器和控制器。从编程角度看，CPU包含寄存器与可执行的指令系统。

2. 存储器:存储器是用来存储数据和指令的记忆部件。从编程角度看，存储器就是地址单元，从功能上说，存储器分为许多种类。

3. 输入设备与输出设备:输入设备是向计算机输入信息的设备，如键盘、鼠标等。输出设备是计算机把处理结果以人能识别的数字、符号等形式表达出来的设备，如显示器、打印机等。

特别提示：冯·诺依曼结构：指令和数据存储器统一编址，哈佛结构：指令和数据存储器分开编址



1.1.4 微型计算机中的三总线：地址、数据与控制（重点）

与车辆的通行需要道路一样，计算机中各个部件之间的信息传输也需要通过电子线路进行，这些线路被称为计算机中的总线。

总线（Bus）是计算机系统中各个部件之间信息传送的公共通路，是一种物理连接。按照计算机所传输的信息种类，总线主要有地址总线（Address Bus, AB）、数据总线（Data Bus, DB）和控制总线（Control Bus, CB），就是通常意义上的计算机内的三总线，也称为系统总线。CPU与存储器、I/O接口之间的连接需要通过三总线。

计算机中总线也有类似市内道路及城市之间道路之分，性能上也各异，不同类型、不同性能的总线，构成了计算机系统内部的通信连接。



1. 地址总线

CPU通过它寻找存储器单元（房间号码）。

特别提示：地址的计算方法

【练习1-1】4GB是如何算出来的？（地址线条数与寻址空间的计算方法）

【练习1-2】1TB=1KGB，需要多少根地址线？

2. 数据总线

CPU通过它发送数据到存储器单元，或取出存储单元中数据到CPU中。

特别提示：机器字长

3. 控制总线

控制着CPU访问存储器动作过程。



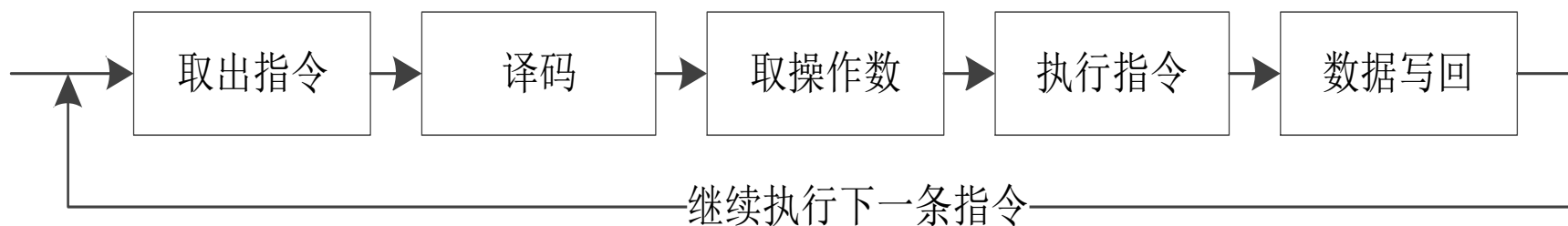
4. 三总线的协同工作

对于三总线协同工作问题，这里举一例加以理解。若要向RAM 中5279地址单元写入一个字节数68，过程如下：（1）CPU通过地址总线发出5279，定位到RAM中的地址单元，相当于找到了那个房间；（2）CPU通过数据总线发出68这个数，但这个数在线上，还没有存到那个房间；（3）CPU发出控制总线，控制着把数据总线上的这个数存入指定房间。于是，RAM中地址为5279这个房间的数据就为68，CPU通过三总线完成了一次数据存储过程。

一般来说，程序执行过程中，CPU对内部寄存器的操作速度比对RAM中的变量操作速度快。



1.1.5 计算机执行指令的简明过程（难点）



- (1) 取出指令。以程序计数器PC中的值为地址，从主存中取出要执行的指令。
- (2) 译码。把指令翻译成CPU内部的微动作序列。
- (3) 取操作数。若指令需要从数据存储器取数，此阶段进行。
- (4) 执行指令。例如，若完成一个加法运算，CPU内的算术逻辑单元ALU将被连接到一组输入和一组输出，输入端提供相加的数值，输出端将含有最后的运算结果输出。
- (5) 数据写回。通常把执行指令阶段的运行结果写到CPU内部寄存器中，以便被后续的指令快速地存取，
- (6) 继续执行下一条指令。



1.2 微机原理的实践选型

1.2.1 微机原理实践选型的困惑

微机原理教学的主要目的是运用面向机器的汇编语言，从底层透明理解微型计算机运行的基本原理及其与外界的基本接口方式。要达到这一目标，必须进行具备可以实际动手的基本实验设备。

长期以来，我国的大部分微机原理教学选用1978年Intel开始推出16位8086微处理器，一些学校配有专门“微机原理实验箱”，内含串行通信接口芯片8251、8位模数接口芯片ADC089、8位数模转换芯片DAC0832、并行接口芯片8255、定时器接口芯片8253、键盘接口芯片8279等等。今天，主芯片和所有接口芯片早已不生产了，一些学校改用模拟方式进行教学。在芯片选型、器件更新、教材更新、实验体系的更新、教学素材的更新等等，成为实践选型的困惑。



1.2.2 微机原理实践选型的基本原则

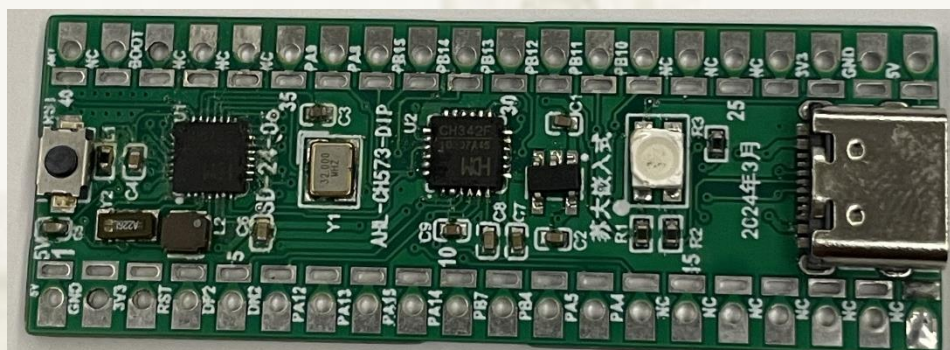
微机原理教学具有基础属性、实践属性与发展属性。满足这三个属性，是微机原理教学芯片选型的基本原则。

1.2.3 AHL-MCP微机原理实践平台概述

1. 选择RISC-V架构微处理器作为微机原理教学蓝本的缘由

RISC-V架构微控制器满足微机原理教学的基础属性、实践属性与发展属性。

2. AHL-CH573微机原理实验套件简介





1.3 数制及数制之间的转换方法

1.3.1 数制

1. 数制的概念

通俗地说，数制（**Number system**）就是计数的法则，它用一组固定的数码和一套统一的规则来表示数字的大小。例如，人们日常生活中使用的数制是十进制（**Decimal system**），它使用0、1、2、3、4、5、6、7、8、9这十个数码，并定义以下规则：自然界中所有的数字都用这十个数码表达，满十进一，且规定同一个数码在从左到右不同的位置上所表示的数值大小不同。人类普遍使用十进制，可能与远古时代用十指记数这个习惯有关。



2. 基数计数法

基数计数法（**Radix notation**），也称按位计数法或进位计数法，该计数方法是**以基数和位权来表示的计数方法**，任何一个数制都包含基数和位权这两个基本要素。

数制中的基数（Radix number）表示基本符号的个数。例如，十进制的基数就是10，二进制的基数就是2，十六进制的基数为16。

数制中的位权（Position weight）表示某一位上的1所表示数值的大小（所处位置重要性的度量），一般简称权（weight）。例如，十进制数693.85，该数中最左边的6代表600，而 $600 = 6 * 10^2$ ，这里的 10^2 就是6所处位置的“权”

有了基数与权概念，**任意一个数 x 可表示成按权展开**，例如： $26.38 = \sum_{i=1}^{-2} \alpha_i R^i = 2 * 10^1 + 6 * 10^0 + 3 * 10^{-1} + 8 * 10^{-2}$



3. 计算机中常用的数制

表 1-1 计算机中常用的二进制、十进制、十六进制

数制	数码	数码个数	基数	进位规则	借位规则	书写前缀	书写后缀
二进制	0、1	2	2	逢二进一	借一当二	0b	B
十进制	0、1、2、3、4、5、6、7、8、9	10	10	逢十进一	借一当十	(无)	D
十六进制	0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F	16	16	逢十六进一	借一当十六	0x	H

说明：十六进制数码中的 A、B、C、D、E、F 分别对应十进制的 10、11、12、13、14、15

【练习1-3】将二进制数101.1101及十六进制数8BD. A6F按权形式展开。

【练习1-4】写出八进制数码个数、基数、进位规则、借位规则。



1.3.2 数制之间的转换方法

1. 其他进制数与十进制数之间的转换

1) 其他进制数转为十进制数：“按权展开求和”

【练习1-5】把十六进制数0x6A8转为十进制数。

2) 十进制数转为其他进制数：一般采用“乘除法”

【练习1-6】把十进制数56.23转为二进制数和十六进制数。



2. 二进制数与十六进制数之间的转换

表 1-2 十六进制数与二进制数的对应关系

十六进制数	二进制数	十六进制数	二进制数
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

二进制数转换为十六进制数的基本方法：以小数点为界，整数部分向左，每4 位二进制数为一组，不足4位的，高位补0，然后用1位十六进制数码表示对应的二进制数即可；小数部分向右，每4 位二进制数为一组，不足4 位的，低位补0，然后用1位十六进制数码表示对应的二进制数即可。

十六进制数转换为二进制数的基本方法：把每位十六进制数码用4位二进制数表示，书写时根据具体情况去除不影响结果的整数部分的前置0与小数部分的后置0，使之符合平时书写习惯即可。

【练习1-7】 将二进制数0b101001.110101转为十六进制数，将十六进制数0x27B5.3D转为二进制数。

3. 利用工具查看进制转换结果





1.4 计算机中信息的基本表示方式

1.4.1 计算机中信息表示的相关基本概念（重点）

1. 位、字节、机器字长

硬件上，计算机中的所有数据均表现为二进制。“位”（bit）是单个二进制数码的简称，是可以拥有两种状态的最小二进制值，分别用“0”和“1”表示。计算机中信息单位是8位二进制数，即“字节”（byte），是计算机中信息基本度量单位。

机器字长是指计算机在运算过程中一次能吞吐的二进制数据位数，表示了CPU内部数据通路的宽度，它等于数据总线条数，与CPU内数据寄存器的宽度是一致的。

计算机中使用二进制，可做如下理解：第一，二进制只取两个数码0和1，物理上可以用两个不同的稳定状态的元器件来表示；第二，它的运算规则简单，基数为2，进位规则是“逢二进一”，借位规则是“借一当二”；第三，计算机的理论基础是逻辑和代数，当二进制与只使用“真”和“假”两个值与逻辑代数建立联系后，就为计算机的逻辑设计提供了便利的工具，如集成电路中门电路的设计。



2. 机器数与真值

数的符号书写用“ \pm ”号表达，称为真值。在规定了用0表示正数、1表示负数之后，以二进制形式形式存储于计算机内部，称为机器数。机器数有不同的编码表示。

例如，整数通常采用补码表示方式，下面将阐述其编码方法及缘由。



1.4.2 整数在计算机中的补码表示方法（难点）

1. 补码的定义与求法

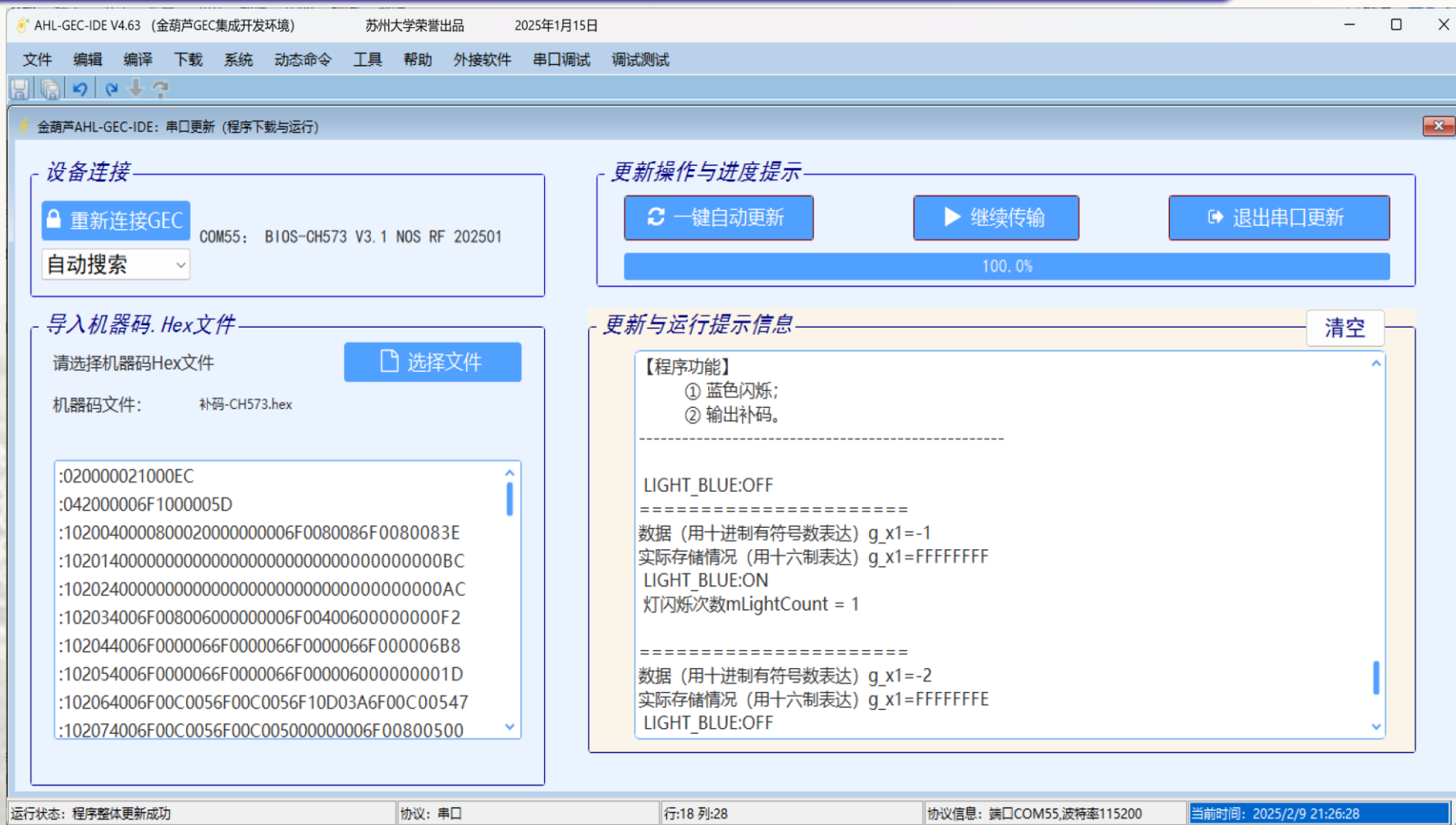
要给出补码概念，首先简要介绍模（modulo）的含义：对于n位机器字长，由于每一位只能存储0或1，对应二进制表达，能接受的最大值是 2^n-1 ，这个 2^n 就称为模。也就是说到 2^n 就溢出啦。就好比一个12小时的时钟，12点就是0点。

有了模的概念，可以把由真值求补码的方法可以作为补码的定义：补码主要用于解决负整数在计算机中的存储问题，对应n位字长，模 $m=2^n$ ，整数表达范围是： $-2^{n-1} \sim (2^{n-1}-1)$ ，设真值记为 x_z ，对于在此范围内的任意正数， $x_z \geq 0$ ，其补码： $x_b = x_z$ ，对于在此范围内的任意负数， $x_z \leq 0$ ，其补码： $x_b = 2^n - |x_z|$ 。简单地说，一个正整数的补码就是自身，一个负整数的补码是模减去这个数的绝对值。

由此可以给出八位、十六位例子（书本P13）



2. 利用程序直观了解补码





3. 在计算机中整数使用补码表示的缘由（重点）

1) 原码与反码表示不合适

为了容易理解，下面以8位计算机为例阐述这个问题。

(1) 所谓原码表示，就是用最高位表示整数的±号，0表是正，1表示负，数值位不变。这种表示出现了-0（1000 000）问题，0就是0，哪还有+0、-0如何理解？如何参与运算？

(2) 所谓反码表示，就是用最高位表示整数的±号，0表是正，1表示负，数值位逐位取反。这种表示出现了-0（1111 1111）问题。

(3) 原码与反码表示解决不了符号位变成了数字之后参与运算问题。在原码表示中，计算： $1 + (-1) = (0000\ 0001)_{\text{原}} + (1000\ 0001)_{\text{原}} = (1000\ 0010)_{\text{原}} = -2$ ，这是不对的。在反码表示中，计算： $(-1) + (-2) = (1111\ 1110)_{\text{反}} + (1111\ 1101)_{\text{反}} = (1111\ 1011)_{\text{反}} = (1000\ 0100)_{\text{原}} = -4$ ，这也是不对的。



2) 补码表示合适

(1) 补码表示可以解决以上问题。首先，没有+0、-0问题了，而且可以用原码中-0(1000 0000)，在补码中表示为-128，形成了-128, -127, ..., -1, 0, 1, ..., 127，共256个8位有符号数的完整表达。其次，在补码表示中，计算： $1 + (-1) = 0000\ 0001 + 1111\ 1111 = 0000\ 0000 = 0$ ，这是对的。又用补码表示计算： $(-1) + (-2) = (1111\ 1111)_{\text{补}} + (1111\ 1110)_{\text{补}} = (1111\ 1101)_{\text{补}} = (1000\ 0011)_{\text{原}} = -3$ ，这也是对的。

(2) 使用补码表示，可以将真值的减法运算变为机器中加法运算，使得CPU内部不需要设计减法器。例如， $1 - 2 = 1 + (-2) = (0000\ 0001)_{\text{补}} + (1111\ 1101)_{\text{补}} = (1111\ 1111)_{\text{补}} = (1000\ 0001)_{\text{原}} = -1$ ，正确。

这里从直观角度说明了机器数选用补码的可行性，通过这个说明，可以基本理解为什么设计补码这种表示方式。



3) 补码设计的基本数学原理

生活中具有1~12小时指针的机械闹钟，到12小时后，又从0开始（12就是0），即超过12就溢出了。若说是18点，即6点， $18/12$ 的余数是6，数学上称之为模运算，符号“mod”，即 $18 \bmod 12 = 6$ ，读做“18模12的结果为6”。

接下来看，若机械闹钟指针指向8点，要把它拨到指向5点，有两种方法：

方法一：回拨，即逆时针拨3小时，即用减法： $8-3=5$ ；

方法二：正拨，即顺时针拨9小时，即用加法： $8+9=5$ （不对啊， $8+9$ 怎么等于5？可对于这个闹钟，这样的操作是对的）。看看实际数学过程： $(8+9) \bmod 12 = 5$ ，即 $8-3$ 与 $8+9$ 具有等同效果。减法运算变成了加法运算。同时，注意这个 $9=12-3$ ，给出了顺时针拨多少小时的一个求法，可以表示成： $8-3$ 与 $8+(12-3)$ 是等效的。

类比一下，在计算机中，若用8位表示机器数，超过256就溢出了（256就是0）。类似上面方法：要计算 $1-2$ ，通过类比， $1-2$ 与 $1+(256-2)$ 是等效的，从8位机器数来看，这个254就称为-2的“补码”。利用这种方法，减法不见了，减法变成了用加法替代。这种分析，也给出了负数补码的一种简便求法，例如-2的补码： $256-2=254$ 。特别看看-128的补码： $256-128=128$ ，-128就是8位机器数（补码）能表示的最小数字了。



1.4.3 实数在计算机中的浮点数表示方法

1. 浮点数表示方法

IEEE于1985年制订了二进制浮点运算标准 IEEE 754 (IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985)，后来经修订后，标准号改为 IEC 60559。该标准规定：从逻辑上用三元组{S, E, M}来表示一个数 V 的，其中符号位S决定数是正数 ($S=0$) 还是负数 ($S=1$)，占1位；M是二进制小数，是通过把二进制数中的小数点向左移n位，直到小数点的左边只有一位且为1而得到的，占23位，被称为尾数位；指数位 $E=127+n$ ，占8位，它决定了小数点的实际位置。以单精度浮点数 (32位) 为例，表1-3列出了其存储格式，从中看出存储值展开成二进制形式就是S、E、M三者的组合。



表1-3 单精度浮点数（32位）存储格式

字段	字段 1（1 位）	字段 2（8 位）	字段 3（23 位）	实际存储值
数据位	D31	D30 D23	D22 D0	
用途	符号位	指数位	<u>尾数位</u>	
记号	S	E	M	
例 1： 32.125	0	10000100	000000010000000000000000	0x42008000
例 2： -64.75	1	10000101	000000110000000000000000	0xC2818000
例 3： 0.015625	0	01111001	000000000000000000000000	0x3C800000
例 4： -0.0078125	1	01111000	000000000000000000000000	0xBC000000

2. 直观理解浮点数表示方法





1.5 文字在计算机中的存储方式—字符编码（重点）

1.5.1 英文编码—ASCII码

计算机处理的一切信息用“0、1”两个符号存储，但却能处理诸如英文、汉字及其他文字信息。人们把像“a、b、c、你、我、他、……”这类信息称为字符（character）。计算机要能处理它们，必须用二进制表示，给出一些规则，规定“a”用什么二进制表示，“b”用什么二进制表示，等等，这种方式称为字符编码（Character encoding）。因历史发展与应用场合不同，字符编码有许多不同方式，常用的英文编码方式主要有ASCII码，常用的中文编码方式主要有GB2312

1. ASCII码的发布者及发布时间

ASCII码（American Standard Code for Information Interchange），中文翻译为：美国信息交换标准代码。被国际标准化组织（International Organization for Standardization, ISO）定为国际标准，称为ISO 646标准，适用于所有拉丁文字字母。ASCII码由美国国家标准学会（American National Standard Institute, ANSI）于1967年第一次规范发布，1986年为最近一次更新。



2. ASCII码的内容概要

ASCII 码使用一个字节进行编码，分为标准ASCII 码与扩展ASCII 码。标准ASCII 码也叫基础ASCII码，规定最高位为0，其他7位表示数值，其范围为0~127，包括编码32个控制符、10个数字、52个大小写字母及其他符号。

表 1-5 标准 ASCII 概括总结

分类	十六进制值	二进制值	十进制值	符号
32 个控制符	0x00	0000 0000	0	NUL (null) 空字符
	
	0x1F	0001 1111	31	US (unit separator) 单元分隔符
空格及 15 个标点符号	0x20	0010 0000	32	(space) 空格
	0x21~0x2F	0010 0001~0010 1111	33~47	! " # \$ % & (右单引号) () * + , - . /
10 个数字	0x30~0x39	0011 0000 ~ 0011 1001	48~57	0~9
6 个符号	0x3A~0x40	0011 1010~0100 0000	58~64	: ; < = ? @
26 个大写字母	0x41~0x5A	0100 0001~0101 1010	65~90	大写字母: A ~ Z
5 个符号	0x5B~0x60	0101 1010~0110 0000	91~96	\ (反斜杠)] (右中括号) ^ (脱字符) _ (下划线) ' (左单引号)
26 个小写字母	0x61~0x7A	0110 0001~0111 1010	97~122	小写字母: a ~ z
4 个符号	0x7B ~ 0x7E	0111 1011 ~ 0111 1110	123 ~ 126	{ } ~
删除符号	0x7F	0111 1111	127	DEL (delete)



1.5.2 中文编码—GB2312及GBK

1. GB2312及GBK的发布者与发布时间

中文编码《信息交换用汉字编码字符集》是由中国国家标准总局1980年发布，标准号是GB 2312-1980。GB2312标准共收录6763个汉字，为了表示更多的汉字，1995年又颁布了《汉字编码扩展规范》（GBK）GBK与GB2312标准兼容，同时支持ISO/IEC10646-1和GB 13000-1的全部中、日、韩（CJK）汉字，共计20902字。GB 18030-2005《信息技术-中文编码字符集》收录了70244个汉字。



2. GB2312及GBK的内容概要

GB2312基本集共收入汉字6763个和非汉字图形字符682个，每个汉字用两个字节编码，分区进行，区号01-94，每区含有94个位号，这种编码方式也称为区位码。举例来说，“啊”字是GB2312之中的第一个汉字，它的区码为16，位码为01，分别用十六进制表示，分放在高低字节，成为两字节的区位码0x1001，区位码加上0x2020就是国标码0x3021，再加上0x8080就是存储在计算机中的机内码0xB0A1，这就是汉字的计算机编码。

为什么不直接使用国标码将汉字存储在计算机内部呢？

汉字机内码的每个字节都大于128，解决了与西文字符的ASCII码冲突的问题，也给编程判断提供了依据。



本章作业：第1章习题 4、5、9、11