第一章
1、计算机 ENIAC 是世界上第一台电子计算机。
2、冯·诺依曼计算机的基本特点：

（1）计算机由运算器、控制器、存储器、输入设备和输出设备 5 部分组成。

（2）采用存储程序的方式，程序和数据放在同一个存储器中，并以二进制码表示。

（3）指令由操作码和地址码组成。

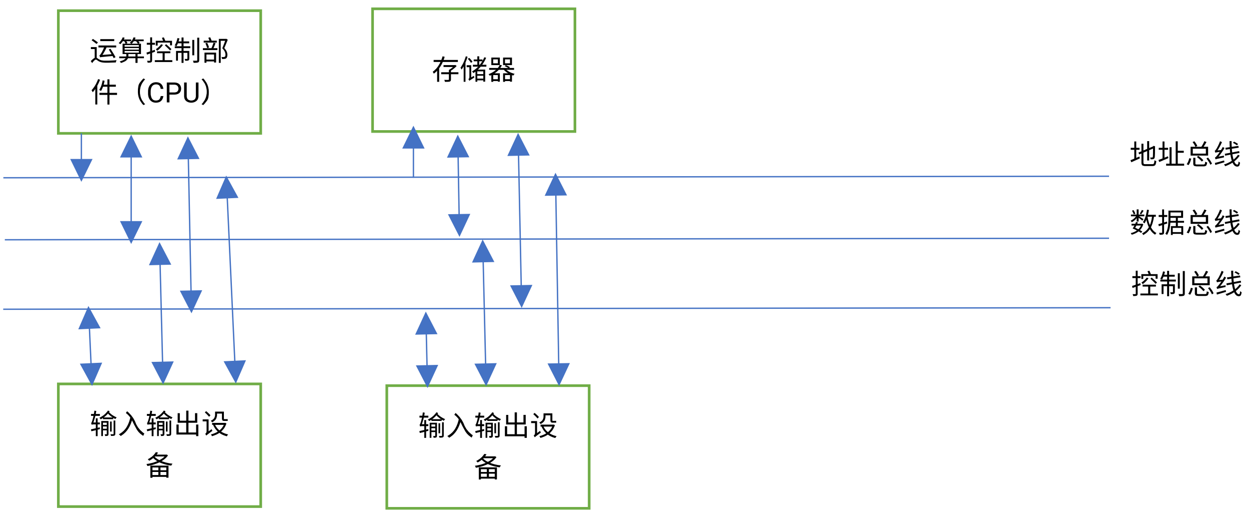
（4）指令在存储器中按执行顺序存放，有指令计数器（即程序计数器 PC）指明要执行指令所在的存储单元地址，一般按顺序递增，但可按运算结果或外界条件而改变。

（5）机器以运算器为中心，输入输出设备与存储器之间的数据传送都通过运算器。

3、电子计算机发展的五个阶段：

第一代：电子管计算机时代
第二代：晶体管计算机时代
第三代：集成电路计算机时代
第四代：大规模集成电路计算机时代
第五代：超大规模集成电路（VLSI,ULSI）计算机时代

4、组成计算机的基本部件有中央处理器（CPU，运算器和控制器）、存储器和输入输出设备。
5、中央处理器又叫 CPU，早期计算机分成运算器和控制器两部分，现在集成在一个芯片中。
6、运算器对信息和数据进行处理和运算，经常进行算术运算和逻辑运算，其内部有一个算数级逻辑运算部件 ALU。
7、



以总线连接的计算机框图

第二章
1、加法器是计算机的基本运算部件之一。
不考虑进位输入时，两数码 Xn，Yn 相加称为半加，表达式如下

$$H_n = X_n \cdot \bar{Y}_n + \bar{X}_n \cdot Y_n = X_n \oplus Y_n$$

功能表

X_n	Y_n	H_n
0	0	0
1	0	1
0	1	1
1	1	0

考虑进位输入时，X_n，Y_n,以及进位输入C_{n-1}相加称为全加，运算结果F_n称为全加和，全加和F_n和进位输出C_n表达式如下

$$F_n = \overline{X}_n\overline{Y}_nC_{n-1} + \overline{X}_nY_n\overline{C}_{n-1} + X_n\overline{Y}_n\overline{C}_{n-1} + X_nY_nC_{n-1}$$

$$C_n = \overline{X}_nY_nC_{n-1} + X_n\overline{Y}_nC_{n-1} + X_nY_n\overline{C}_{n-1} + X_nY_nC_{n-1}$$

功能表

X_n	Y_n	C_{n-1}	F_n	C_n
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
1	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1

F_n还可用两个半加器来形成，其表达式为：

$$F_n = X_n \oplus Y_n \oplus C_{n-1}$$

超前进位产生电路是根据各位进位的形成条件来实现的。只要满足下述两个条件中的任一个，就可形成C_1：

（1） X_1、Y_1均为 1；

（2） X_1、Y_1任一个为 1 且进位C_0为 1.

由此，可写得C_1的表达式为

$$C_1 = X_1Y_1 + (X_1 + Y_1)C_0$$

只要满足下述三个条件中任一个即可形成C_2：

（1）X_2、Y_2均为 1；

（2）X_2、Y_2任一个为 1，且X_1、Y_1均为 1

（3）X_2、Y_2任一个为 1，且X_1、Y_1任一个为 1，且C_0为 1

由此可得C_2表达式为

$$C_2 = X_2Y_2 + (X_2+Y_2)X_1Y_1 + (X_2+Y_2)(X_1 + Y_1)C_0$$

同理，有C_3、C_4的表达式如下

$$C_3 = X_3Y_3 + (X_3+Y_3)X_2Y_2 + (X_3+Y_3)(X_2+Y_2)X_1Y_1 + (X_3+Y_3)(X_2+Y_2)(X_1 + Y_1)C_0$$

C_4

$$= X_4Y_4 + (X_4+Y_4)X_3Y_3 + (X_4+Y_4)(X_3+Y_3)X_2Y_2$$
$$+(X_4+Y_4)(X_3+Y_3)(X_2+Y_2)X_1Y_1 + (X_4+Y_4)(X_3+Y_3)(X_2+Y_2)(X_1 + Y_1)C_0$$

进位传递函数P_i和进位产生函数G_i，他们的定义为：

$$\begin{cases}P_i=X_i+Y_i\\G_i=X_i\cdot Y_i\end{cases}$$

以上两式可代入之前推导的表达式中。

第三章

1、二、八、十六和十进制数的对应关系（需要会使用，不会直接考）

二进制数	八进制数	十六进制数	十进制数
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

2、十进制数的编码与运算

（1）有权码，即 BCD 码，注意要对运算结果进行修正，即所加结果超过$(1001)_2$时，要进行加 6 修正，即加$(0110)_2$，并向高位进位。

（2）无权码，即余三码和格雷码等。

余三码是在 8421BCD 的基础上，把每个编码都加上 0011 形成，当两个余三码相加不产生进位时，应从结果中减去 0011，产生进位时，应将进位信号送入高位，本位加 0011。

格雷码的编码规则：任何两个相邻编码只有一个二进制位不同，而其余 3 个二进制位相同。举两组常用的编码为例：

十进制数	余三码	格雷码（1）	格雷码（2）
0	0011	0000	0000
1	0100	0001	0100
2	0101	0011	0110
3	0110	0010	0010
4	0111	0110	1010
5	1000	1110	1011
6	1001	1010	0011

7	1010	1000	0001
8	1011	1100	1001
9	1100	0100	1000

3、对小数

原码表示法

$$[X]_{原}=\begin{cases}X & 0\le X<1\\ 1-X=1+|X| & -1<X\le 0\end{cases}$$

补码表示法

$$[X]_{补}=\begin{cases}X & 0\le X<1\\ 2+X=2-|X| & -1\le X<0\end{cases}$$

反码表示法

$$[X]_{反}=\begin{cases}X & 0\le X<1\\ 2-2^{-n}+X & -1<X\le 0\end{cases}$$

对整数

原码表示法

$$[X]_{原}=\begin{cases}X & 0\le X<2^{n}\\ 2^{n}-X=2^{n}+|X| & -2^{n}<X\le 0\end{cases}$$

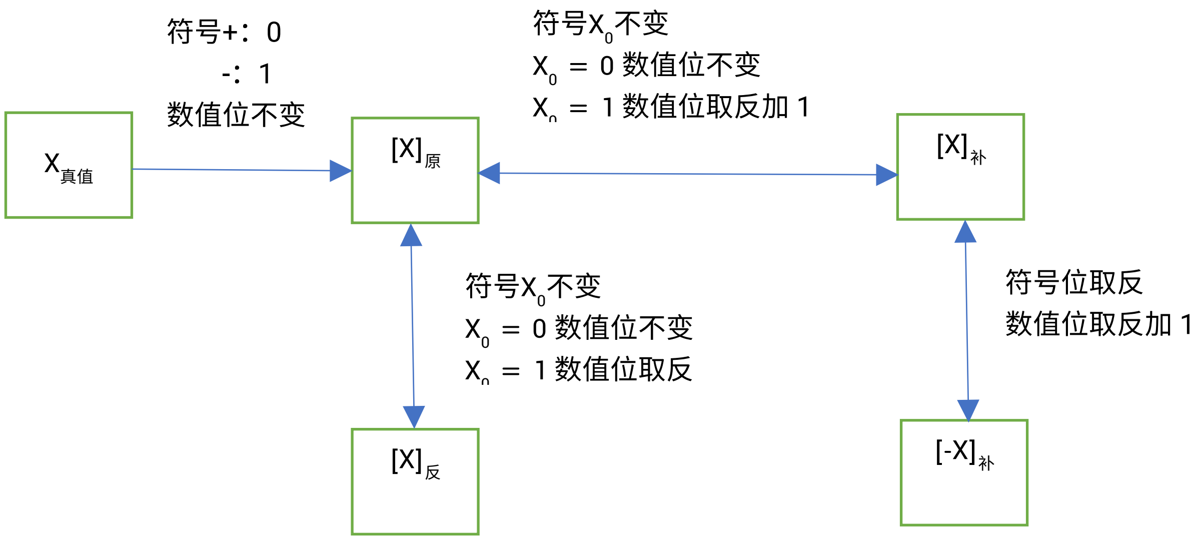
补码表示法

$$[X]_{补}=\begin{cases}X & 0\le X<2^{n}\\ 2^{n+1}+X=2^{n+1}-|X| & -2^{n}\le X<0\end{cases}$$

反码表示法

$$[X]_{反}=\begin{cases}X & 0\le X<2^{n}\\ (2^{n+1}-1)+X & -2^{n}<X\le 0\end{cases}$$

真值，原码，补码，反码以及$[-X]_{补}$的转换规律:



4、当运算结果超出机器数（机器数：通常，机器数是把符号"数字化"的数,是数字在计算机中的二进制表示形式）所能表示的范围时，称为溢出。

两个异号数相加或两个同号数相减，其结果是不会溢出的。仅当两个同号数相加或者两个异号数相减时，才有可能发生溢出的情况。

补码顶点数加减运算溢出判断的方法有三种：

（1） 采用一位符号位

由于减法运算在机器中是用加法器实现的，因此无论是加法还是减法，只要参加操作的两个数符号相同，结果又与原操作数符号不同，则表示结果溢出。

设 A 的符号为A_s，B 的符号为B_s，运算结果的符号为S_s，则溢出逻辑表达式为

$$V = A_s B_s \overline{S_s} + \overline{A_s}\,\overline{B_s} S_s$$

若 V=0，表示无溢出，若 V=1，表示有溢出。

（2） 采用双符号位

双符号位法也称模 4 补码，运算结果的两个符号位$S_{s1}S_{s2}$相同，表示未溢出；运算结果的两个符号位$S_{s1}S_{s2}$不同，表示溢出，此时最高位富豪代表真正的符号。

符号位$S_{s1}S_{s2}$的各种情况如下：

① $S_{s1}S_{s2}$=00，表示结果为正数，无溢出

② $S_{s1}S_{s2}$=01，表示结果为正溢出。

③ $S_{s1}S_{s2}$=10，表示结果为负溢出。

④ $S_{s1}S_{s2}$=11，表示结果为负数，无溢出

溢出逻辑判断表达式为 $V=S_{s1} \oplus S_{s2}$，若 V=0，表示无溢出；若 V=1，表示有溢出。

（3） 采用一位符号位根据数据位的进位情况判断溢出

若符号位的进位C_s与最高数位的进位C_1相同，则说明没有溢出，否则表示发生溢出。溢出逻辑判断表达式为 $V=C_s \oplus C_1$，若 V=0，表示无溢出；若 V=1，表示有溢出。

5、定点数补码一位乘法（Booth 算法）

设$[X]_补 = X_sX_1X_2…X_n$，$[Y]_补 = Y_sY_1Y_2…Y_n$，则运算规则如下：

① 符号位参与运算，运算的数均以补码表示。

② 被乘数一般取双符号位参与运算，部分积取双符号位，初值为 0，乘数可取单符号位。

③ 乘数末尾增设附加位Y_{n+1}，且初值为 0。

④ 根据（Y_n，Y_{n+1}）的取值来确定操作，附 Booth 算法的移位规则：

Y_n（高位）	Y_{n+1}（低位）	操作
0	0	部分积右移一位
0	1	部分积加$[X]_补$，右移一位
1	0	部分积加$[-X]_补$，右移一位
1	1	部分积右移一位

⑤ 移位按补码右移规则进行。

⑥ 按照上述算法进行 n+1 步操作，但第 n+1 步不再位移（共进行 n+1 次累加和 n 次右移），仅根据Y_n与Y_{n+1}的比较结果做相应的运算。

例子可看书上 P46-48 的例 3.32-3.34。

6、浮点数的表示

表示格式

$$N = r^E \times M$$

式中，r 是浮点数阶码的底（隐含），与尾数的基数相同，通常 r=2，E 和 M 都是有符号的定点数，E 称为阶码，M 称为尾数。

J_r	$J_1J_2…J_M$	S_f	$S_1S_2…S_n$
阶符	阶码的数值部分	数符	尾数的数值部分

浮点数的一般格式

规格化浮点数

左规：当浮点数运算的结果为非规格化时要进行规格化处理，将尾数算数左移一位，阶码减 1（基数为 2 时）的方法称为左规，左规可能要进行多次。

右规：当浮点数运算的结果为输出现溢出（双符号位为 01 或 10）时，将尾数算数右移一位，阶码加 1（基数为 2 时）的方法称为右规。需要右规时，只需进行一次。

规格化浮点数的尾数 M 的绝对值应满足条件 1/R≤|M|≤1。

浮点数的加减法运算

① 对阶

对阶的目的是使两个操作数的小数点位置对齐，即使得两个数的阶码相等。为此，先求阶码差，然后以小阶向大阶看齐的原则，将阶码小的尾数右移一位（基数为 2）阶加 1，知道两个数的阶码相等为止。尾数右移时，舍弃掉有效位会产生误差，影响精度

② 尾数求和

将对阶后的尾数按定点数加减运算规则运算。

③ 规格化

④ 舍入

在对接和右规的过程中，可能会将尾数的低位丢失，引起误差，影响精度。常见的舍入方法有："0"舍"1"入法和恒置"1"法。

"0"舍"1"入法：类似于十进制数运算中的"四舍五入"法，即在尾数右移时，被移去的最高数值位为 0，则舍去；被移去的最高数值位为 1，则在尾数的末位加 1.这样做可能会使尾数又溢出，此时需再做一次右规。

恒置"1"法：尾数右移时，不论丢掉的最高数值位是"1"还是"0"都使有以后的尾数末尾恒置"1"。这种方法同样有使尾数变大和变小的两种可能。

⑤ 判断溢出

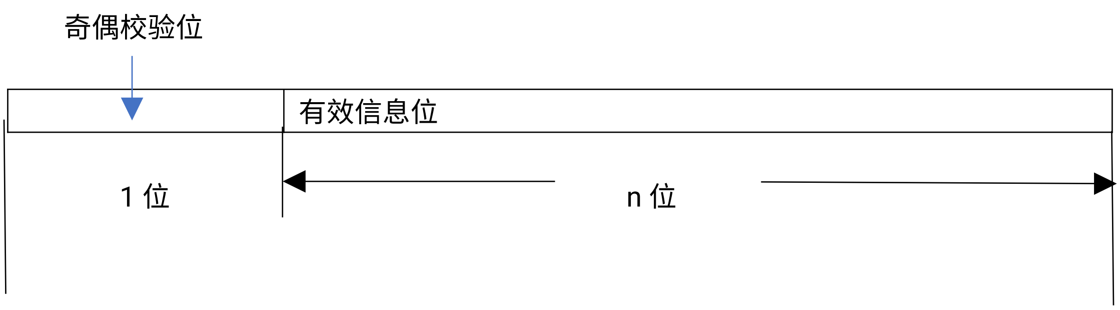
在浮点数规格化中已指出，当尾数之和（差）出现 01.xx 或 10.xx 时，并不表示溢出，只能将此数右规后，再根据阶码来判断浮点数运算结果是否溢出。

7、奇偶校验码

在原编码上加一个校验位，它的码距等于 2，可以检测出一位错误（或奇数位错误），但不能确定出错的位置，也不能够检测出偶数位错误，增加的冗余位称为奇偶校验位。

奇偶校验实现的方法：

有若干位有效信息再加上一个二进制位（校验位）组成校验码，校验位的取值（0 或 1）将使整个校验码中"1"的个数为奇数或偶数，所以有两种可供选择的校验规律。



奇偶校验码

奇校验码：整个校验码（有效信息位和校验位）中“1”的个数为奇数。
偶校验码：整个校验码（有效信息位和校验位）中“1”的个数为偶数。

第四章

1、CPU 通过使用 AP（地址寄存器）和 DR（数码寄存器）和主存进行数据传递。若 AR 为 K 位字长，DR 为 n 位字长，则允许主存包含2^K个可寻址单位（字节或字)。

2、动态存储器 DRAM 工作原理

DRAM 是利用存储元电路中栅极电容上的电荷来存储信息的，DPAM 的基本存储元通常只使用一个晶体管，所以它比 SRAM 的密度高很多。DPAM 采用地址复用技术，地址线是原来 1/2，地址信号分行、列两次传送。

相对于 SRAM，DPAM 具有容易集成，价位低，容量达和功耗低等优点，但 DRAM 的存取速度比 SRAM 的慢，一半用来组成大容量主存系统。

DRAM 电容上的电荷一般只能维持 1~2ms，因此即使电源不断电，心系也会自动消失，为此必须每隔一定时间刷新，通常取 2ms，称为刷新周期。刷新方式一般为 3 种：

（1） 集中刷新　存在死区（集中刷新期间)，死区不能访问存储器。
（2） 分散刷新　不存在死区，工作周期前半部分正常用于读、写或保持；后半部分用于刷新。缺点加长了系统存取周期，降低了整机速度。
（3） 异步刷新　既可缩短死时间，又能充分利用最大刷新间隔为 2ms 的特点。

DRAM 刷新需要注意问题：

（1） 刷新对 CPU 是透明的，刷新不依赖与外部的访问
（2） 动态 RAM 的刷新单位是行，由芯片内部自行生成行地址
（3） 刷新操作类似于读操作，但有所不同

DRAM 需要刷新，SRAM 不需要刷新，但两者都满足断电后数据丢失

DRAM 发展

（1） 同步 DRAM（SDRAM）
（2） DDR SDRAM
（3） DDR2 SDRAM
（4） DDR3
（5） Rambus DRAM（RDRAM）

3、非易失性半导体存储器

（1）只读存储器（ROM）
　　内容不会改变
（2）可编程序的只读存储器（PROM）
　　一次性写入存储器
（3）可擦可编程序的只读存储器（EPROM）
　　可实现整体擦除，编程次数不受限制
（4）可电擦可编程序只读存储器（E^2PROM）
　　可用电擦除，但重复改写的次数有限制

（6）　快速擦除读写存储器（Flash Memory）

4、存储器的组成与控制（P81,408 P104）
（1）位扩展法
（2）字扩展法
（3）字位同时扩展法
存储器地址寄存器 MAR 的位数决定了主存地址空间大小（主存容量不能代表 MAR 位数）
寻址范围与主存地址空间大小有关。