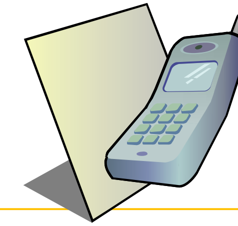
華中科技大學

数字电路与逻辑设计

数字电路与逻辑设计

主讲教师：何云峰 副教授

联系方式



- 电　　话：18672396870
- 电子邮件：yfhe@hust.edu.cn
- QQ：761584990
- 个人主页：http://faculty.hust.edu.cn/heyunfeng

成绩评定办法

- 期末考试：60%
- 平时作业：20%
- 课堂考勤：5%
- MOOC：15%（每章单元测验和讨论）

课程资源

- MOOC
  - https://www.icourse163.org/course/HUST-1207043813

华中科技大学

数字电路与逻辑设计

课程资源

- 作业与实验等提交
  - http://hust.fanya.chaoxing.com

课程资源

- 国家精品资源共享课
  - http://www.icourses.cn/coursestatic/course_5831.html
- 课程网站
  - http://119.97.217.16/2012/szdlyljsj/szdl/index.html

前 言

主讲教师：何云峰

技术革命伴随着大国的崛起

尽管人类的历史写的是帝王将相史或战争史，但人类文明发展的历史实际上是一部科学和技术发展史

技术发展到一个新阶段，人类文明就会上升到一个新的台阶。人类文明进步的根本动力是技术发展驱动；而朝代的更迭、政权的替换，只是给当时的社会进步提供了一个更合理的治理方式。

真正对人类文明进步起根本性作用的是技术的发展

每一次技术的重大发明，都会对人类文明产生重大的改变，同时也给教育带来巨大影响，不仅使教育内容增加，而且使教育思想、教育手段、教育方法更加先进，最终导致物质文明和精神文明的相互促进、共同发展。

石器时代、青铜器时代、航海时代、蒸汽机时代、工业革命、信息技术时代……

备注：引用教育部科技发展中心李志民主任的报告

技术革命

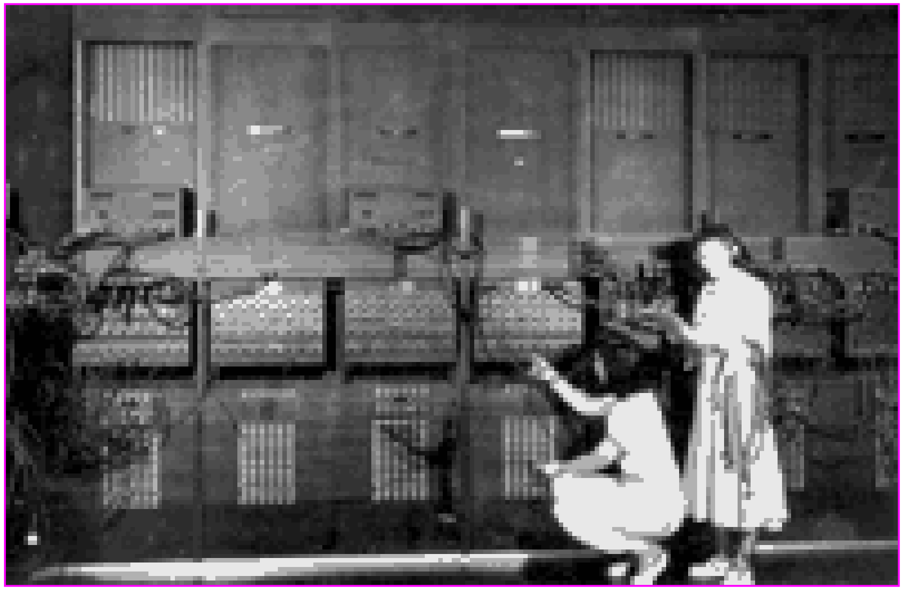
- **第一次工业革命**
  - 18世纪60年代- 19世纪40年代
  - 蒸汽机的广泛应用（即蒸汽时代)

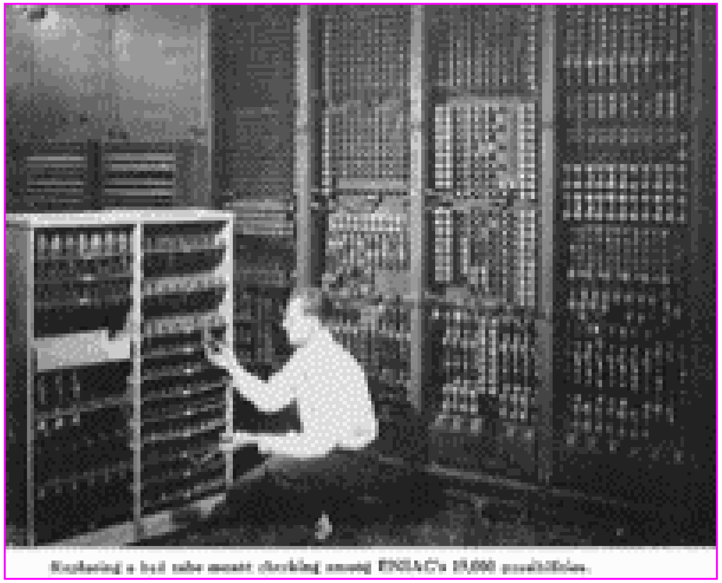
技术革命

- 第二次工业革命
  - 19世纪70年代-20世纪初
  - 电力的广泛应用（即电气时代）、内燃机

技术变革

- 信息革命
  - 1946年，第一台电子计算机ENIAC

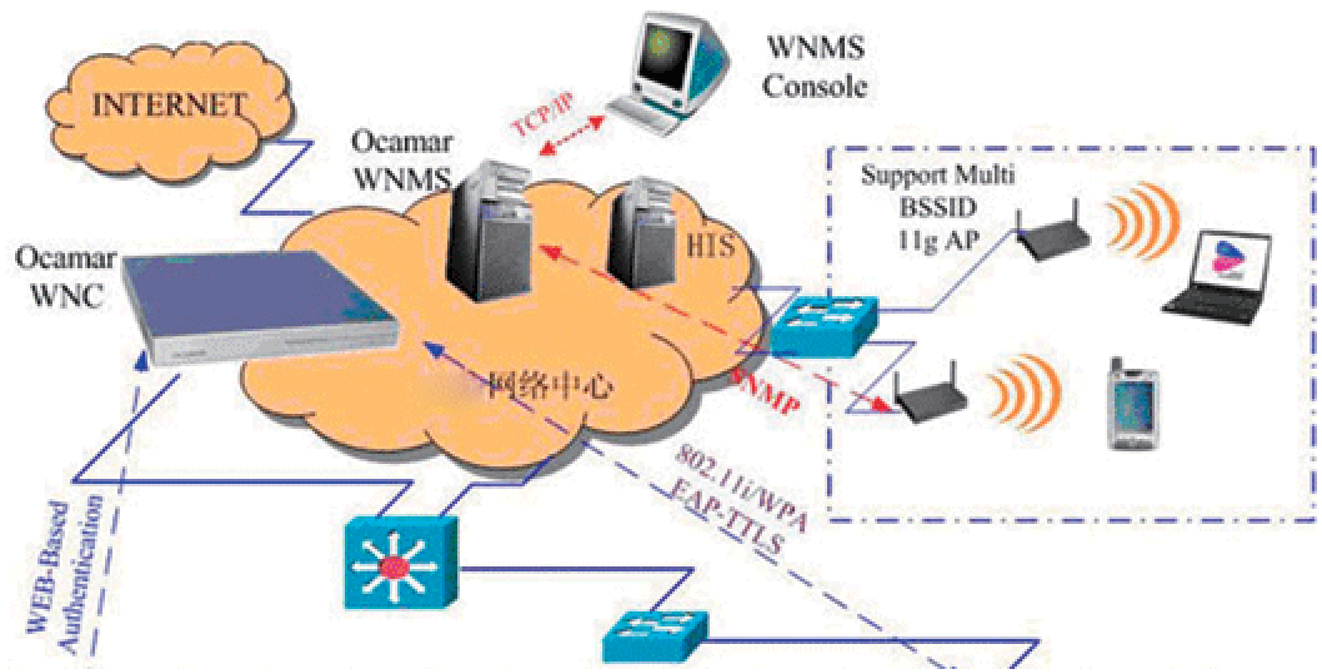




- ENIAC的开发经费几经追加，达到48万美元，相当于现在的1000万美元以上

技术变革

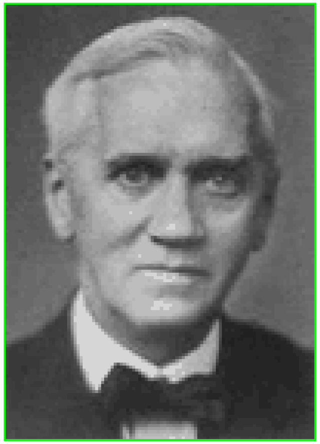
- 信息革命
  - 1969年，第一个计算机网络APPANET

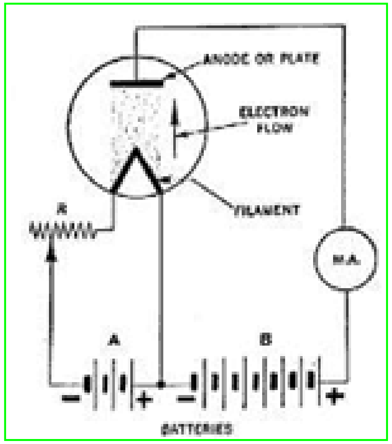


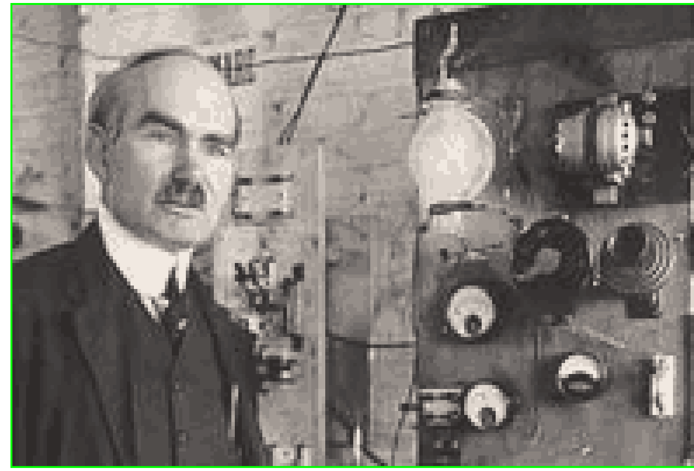
电子管

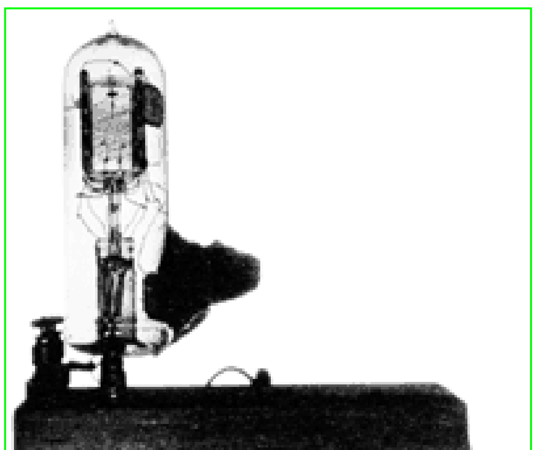
- 电子二极管和三极管在20世纪头几年相继问世
- 真空电子二极管的发明使人类打开了电子文明的大门，而电子三极管的发明及其放大原理的发现，标志着人类科技史进入了一个新的时代：**电子时代**
- 1904年，英国人弗莱明发明真空电子二极管，电子管的诞生，是人类电子文明的起点
- 德弗雷斯特因发明三极管被称为“无线电之父”

电子管

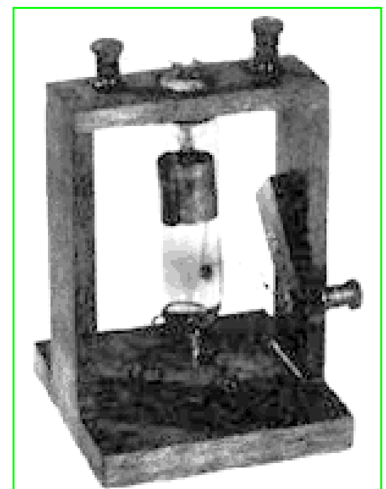








二极管



三极管

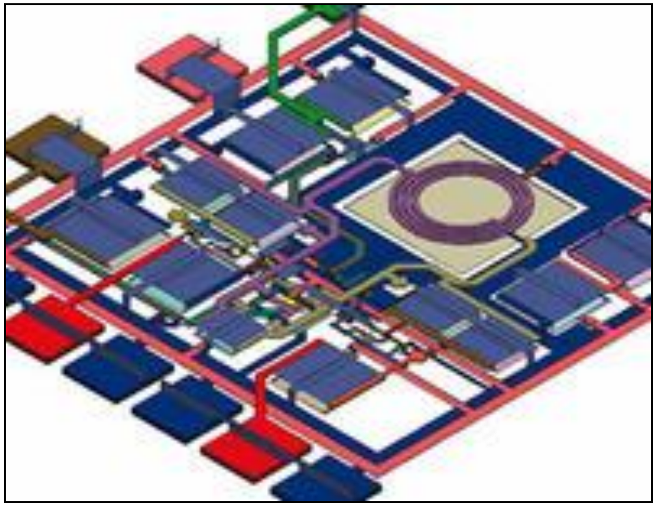
晶体管

- 1947年，贝尔实验室的肖克莱、巴丁、布拉顿发明点触型晶体管
- 1950年又发明了面结型晶体管
- 相比电子管，晶体管体积小、重量轻、寿命长、发热少、功耗低，电子线路的结构大大改观，运算速度则大幅度提高

晶体管

- 肖克莱（左）、巴丁（中）、布拉顿（右）于1956年共同获得诺贝尔物理学奖
- 发明晶体管的肖克莱在加利福尼亚创立了当地第一家半导体公司，这一地区后来被称为硅谷

集成电路（Integrated Circuit-IC）



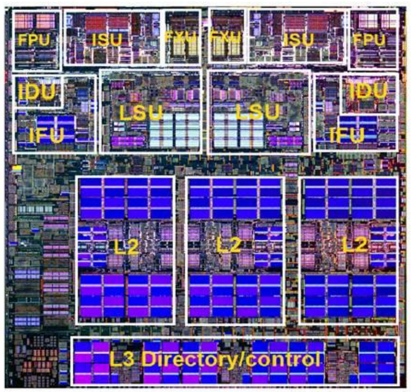
- 一个电路中所需的晶体管、二极管、电阻、电容和电感等元件及布线互连一起，制作在一小块或几小块半导体晶片或介质基片上
- 通过引脚与外部联系

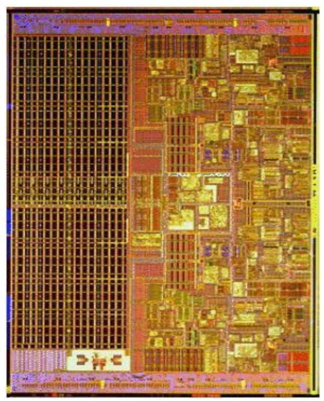

intel
i486 SX
AMD
Athlon
AMD
intel
pentium
C80286-6
S40172
14370040
INTEL '84
AMD
K6
MMX ENHANCED PROCESSOR
50G3796
ESO
IBM
9314
19314000KS
P
intel
pentium III
MADE IN U.S.A.

集成电路的应用

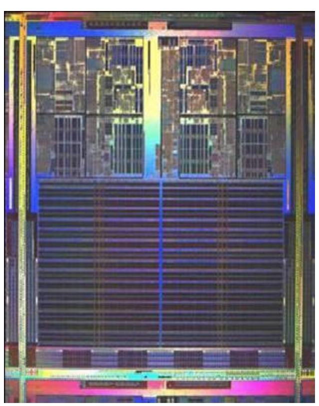
同构多核

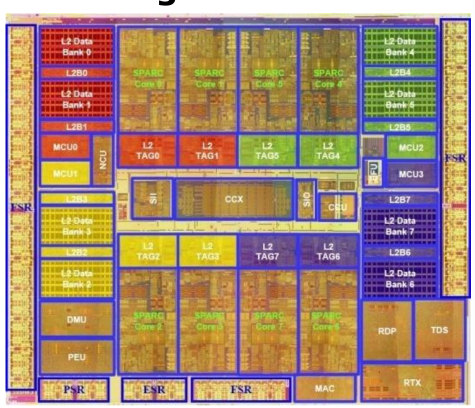
Intel Core Duo Dual Core





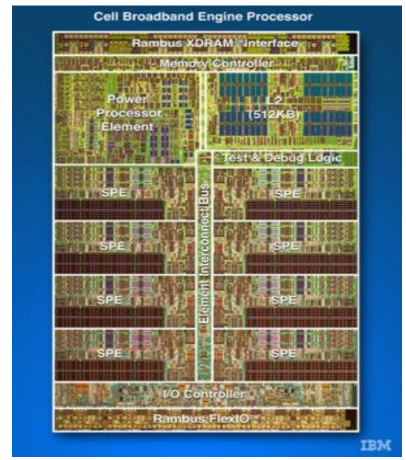
SUN Niagara2 8-core





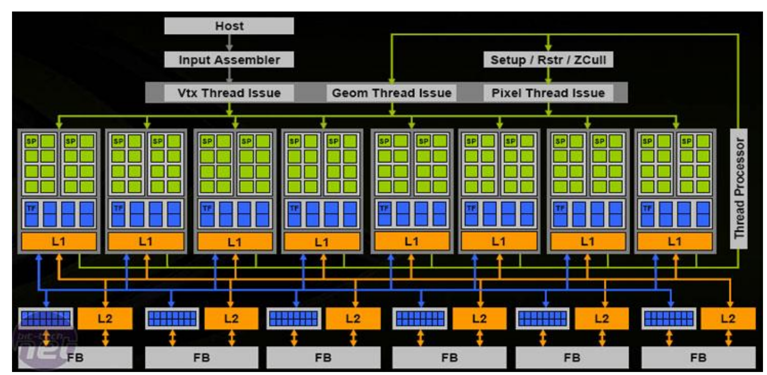
异构多核

IBM CELL



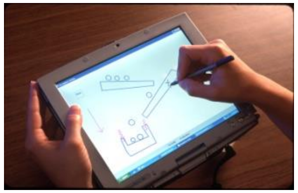
ATI R600

Nvidia G80



集成电路的应用









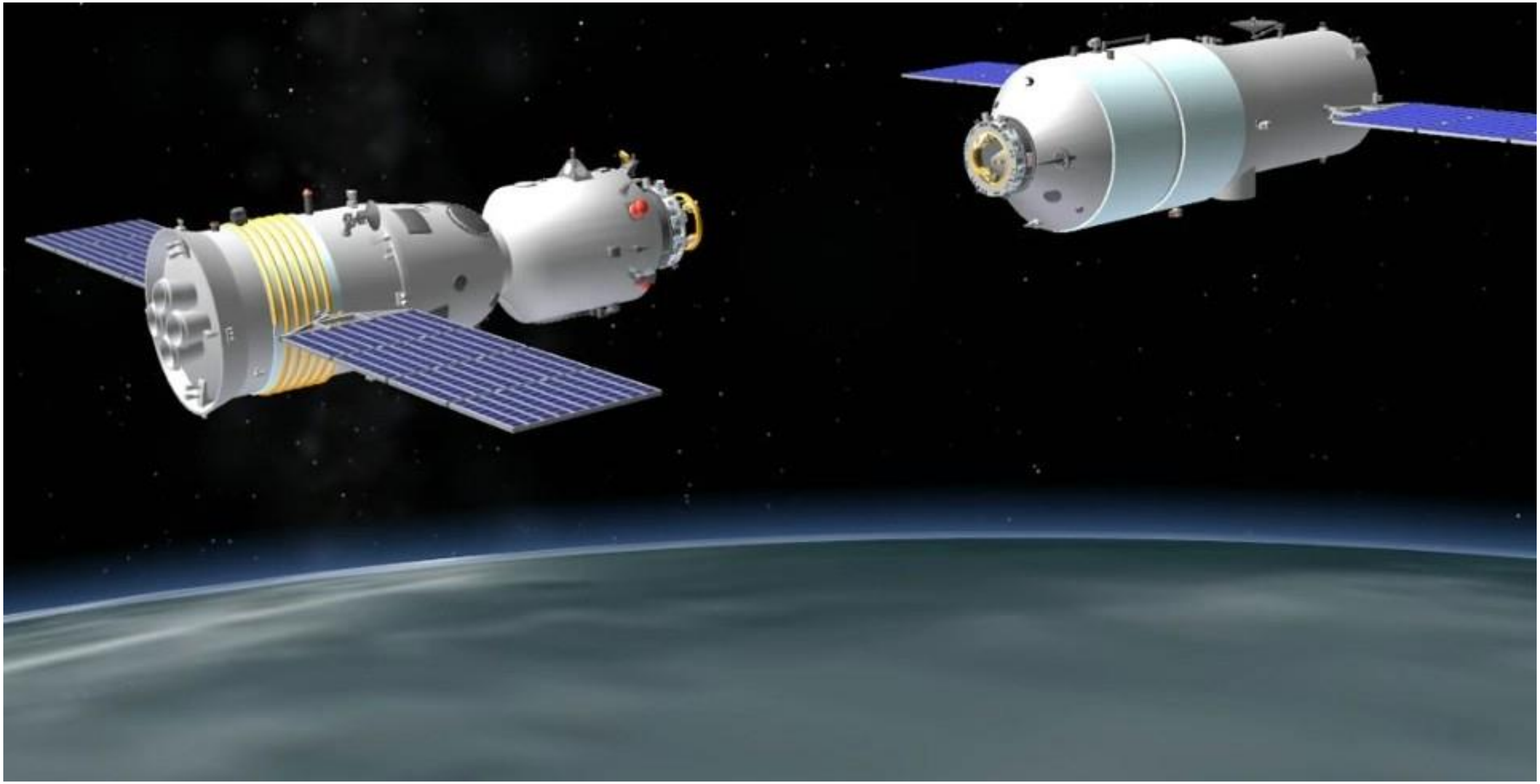




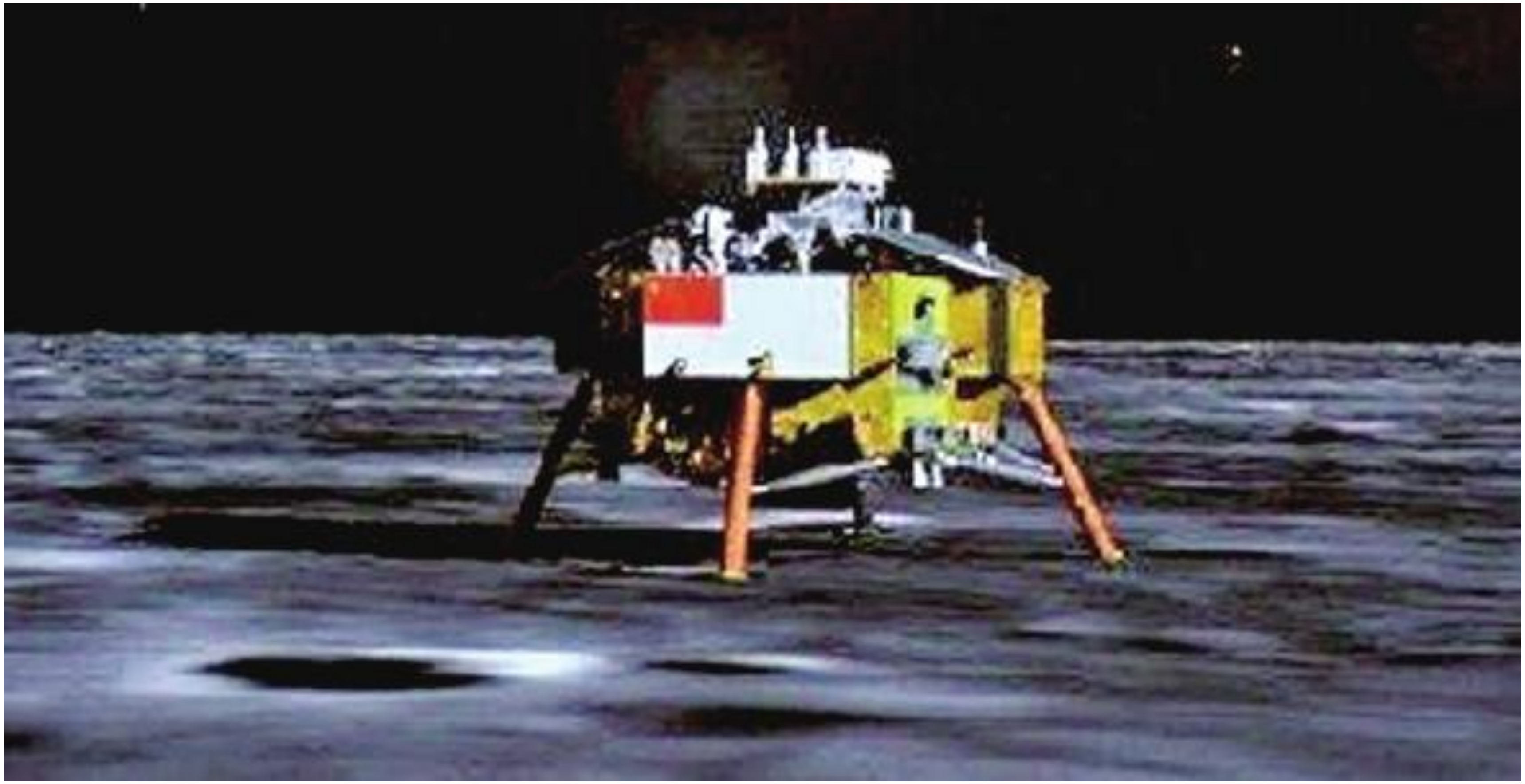




集成电路的应用



集成电路的应用



集成电路的应用

集成电路的应用

集成电路的应用

互联网时代到来……

互联网对人类的影响将不断提升

第一阶段为
信息互联
主要是解决人类知情权的平等

第二阶段为
消费互联
为人类的物质生活需求提供方便

第三阶段为
生产互联
服务于人类就业和事业发展

第四阶段为
智慧互联
帮助人类实现对知识和精神生活的追求

最终发展成为生命互联，满足人类健康长寿的愿望。

互联网将改变行业形态：工业生产将是大协作的时代；农业生产将是按需供应的时代；消费将成为私人定制的时代。

互联网将改变市场结构：市场将点与点的竞争变成链与链的竞争。

备注：引用教育部科技发展中心李志民主任的报告

互联网时代到来……

互联网精神

合作共赢
共建共享

互联网规律

无穷大、无穷小

互联网特征

开放、海量
天量

- **文学进入无经典的时代，艺术成为雅俗共赏的时代，教育成为互为师生的时代，学术将迎来开放存取的时代，新闻真正自由，政治充分民主，历史将会趋于真实。**

备注：引用教育部科技发展中心李志民主任的报告

总 结

- 未来的世界更为精彩，还有很多惊喜……
- 微电子技术、计算机技术是实现这些惊喜的物质基础
- 《数字电路与逻辑设计》是理论基础，必须掌握

课程性质

- 计算机专业必修的一门重要专业基础课
- 在介绍有关数字系统基本知识、基本理论、及常用数字集成电路的基础上，<u>重点讨论数字逻辑电路分析与设计的基本方法</u>
- <u>从计算机的层次结构上讲，“数字逻辑”是深入了解计算机“内核”的一门最关键的基础课程</u>

课程目标

- 掌握数字逻辑的基本概念、基本原理和数字逻辑电路的基本工程知识，理解逻辑电路的基本内涵，能用逻辑语言和工具表达逻辑电路能熟练地运用基本知识和理论对各类电路进行分析
- 掌握逻辑电路的建模方法，具备一般数字逻辑电路的建模与分析求解能力；
- 掌握数字逻辑电路的一般设计方法，具备一般数字逻辑电路设计能力。

课程目标对毕业要求的支撑关系

- 1.1能够将数学、自然科学和信息科学的语言工具用于计算机复杂工程问题的表述
- 1.2能针对计算机复杂工程问题的具体对象进行建模和求解
- 3.1掌握与计算机复杂工程问题有关的工程设计和软硬件产品开发全周期、全流程的基本设计/开发方法和技术，了解影响设计目标和技术方案的多种因素

教材及参考书

- 欧阳星明主编，于俊清副主编. 数字逻辑（第4版），华中科技大学出版社，2009

- 欧阳星明主编，数字逻辑学习与解题指南，华中科技大学出版社（第二版），2005

教学内容(授课 48，上机 32)

- 基本知识、 基本理论、 基本器件
  - 数制、代码表示
  - 逻辑代数
  - 集成门电路与触发器
- 小规模集成电路的逻辑电路分析与设计
  - 组合逻辑电路
  - 同步时序逻辑电路
  - 异步时序逻辑电路
- 中规模通用集成电路及应用
- 大规模可编程逻辑器件及应用

学习方法

- 掌握课程特点
  - 一门既抽象又具体的课程
  - 逻辑设计方法十分灵活
  - 理论知识与实际应用结合十分紧密
- 重视课堂学习
  - 认真听课、主动思考

学习方法

- 培养自学能力
  - 认真阅读教材内容
  - 善于总结、归纳
  - 加强课后练习
  - 积极参与学习讨论
  - 广泛阅读，拓宽知识面
- 注重理论联系实际
  - 将书本知识与工程实际统一
  - 将理论知识与实际应用结合

第一章　基本知识

主讲教师：何云峰

提 纲

1 数字信号与系统

2 数制及其转换

3 带符号二进制数的代码表示

4 几种常用的编码

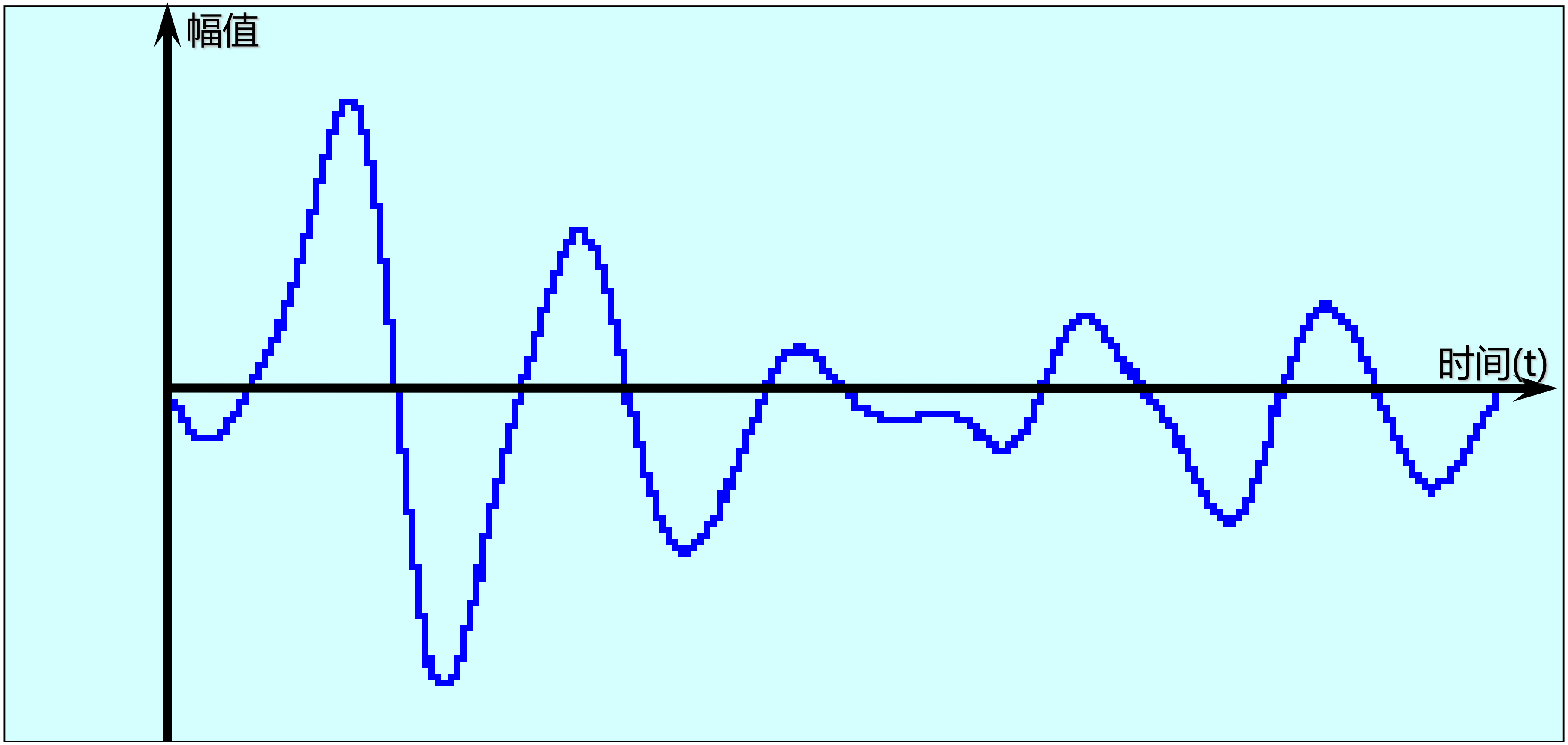
数字信号

- 信号的变化在时间上和数值上都是离散的，或者说断续的，则称为**离散信号**
- 离散信号的变化可以用不同的数字反映，所以又称为**数字信号**，简称为**数字量**
- 举例
  - 学生成绩记录，工厂产品统计，电路开关的状态等

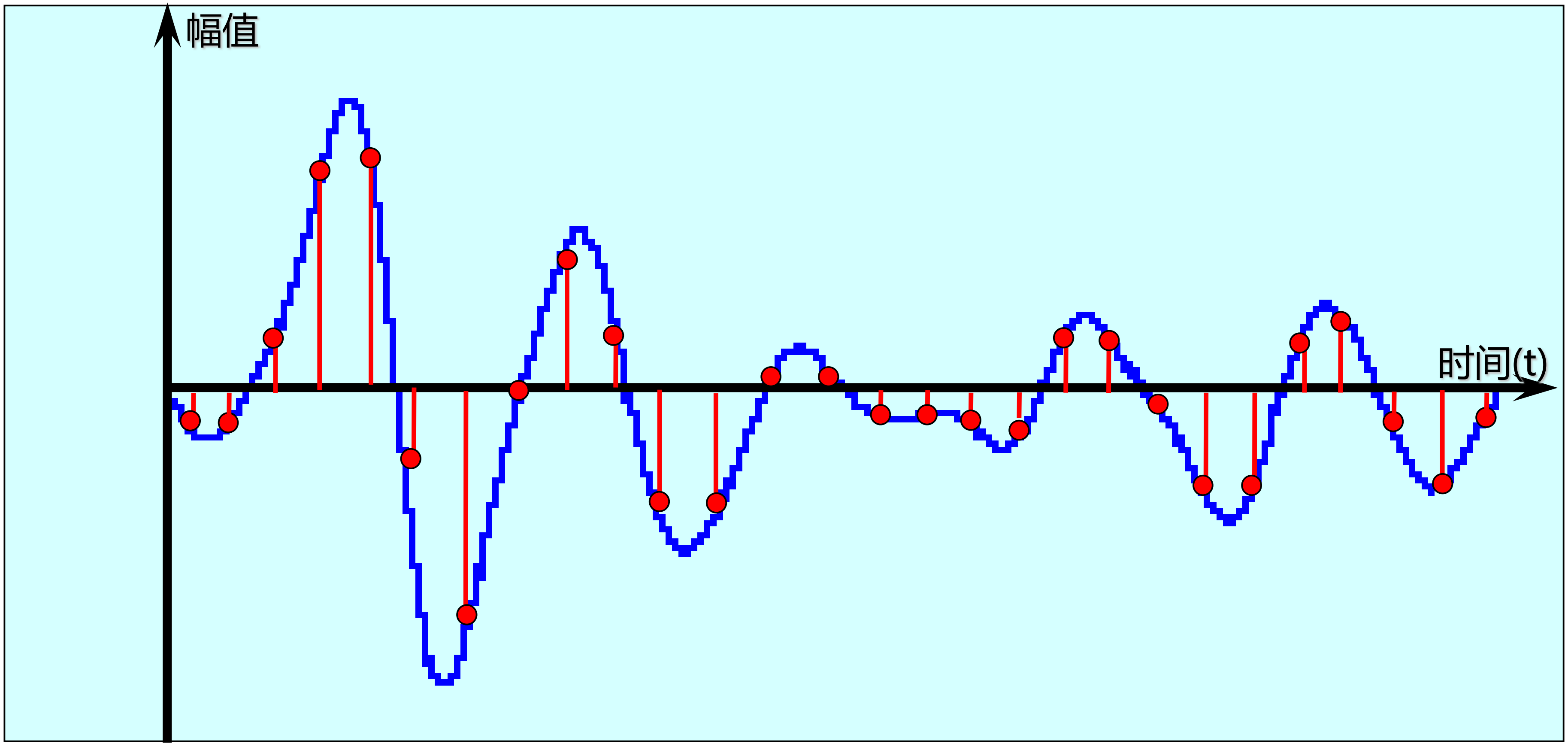
数字系统

- 模拟信号与数字信号的相互转换
  - A/D-模数转换
  - D/A-数模转换
- 何谓“数字系统”？
  - 数字系统是一个能对数字信号进行加工、传递和存储的实体，它由实现各种功能的数字逻辑电路相互连接而成
  - 例如，MP3、手机、数字计算机

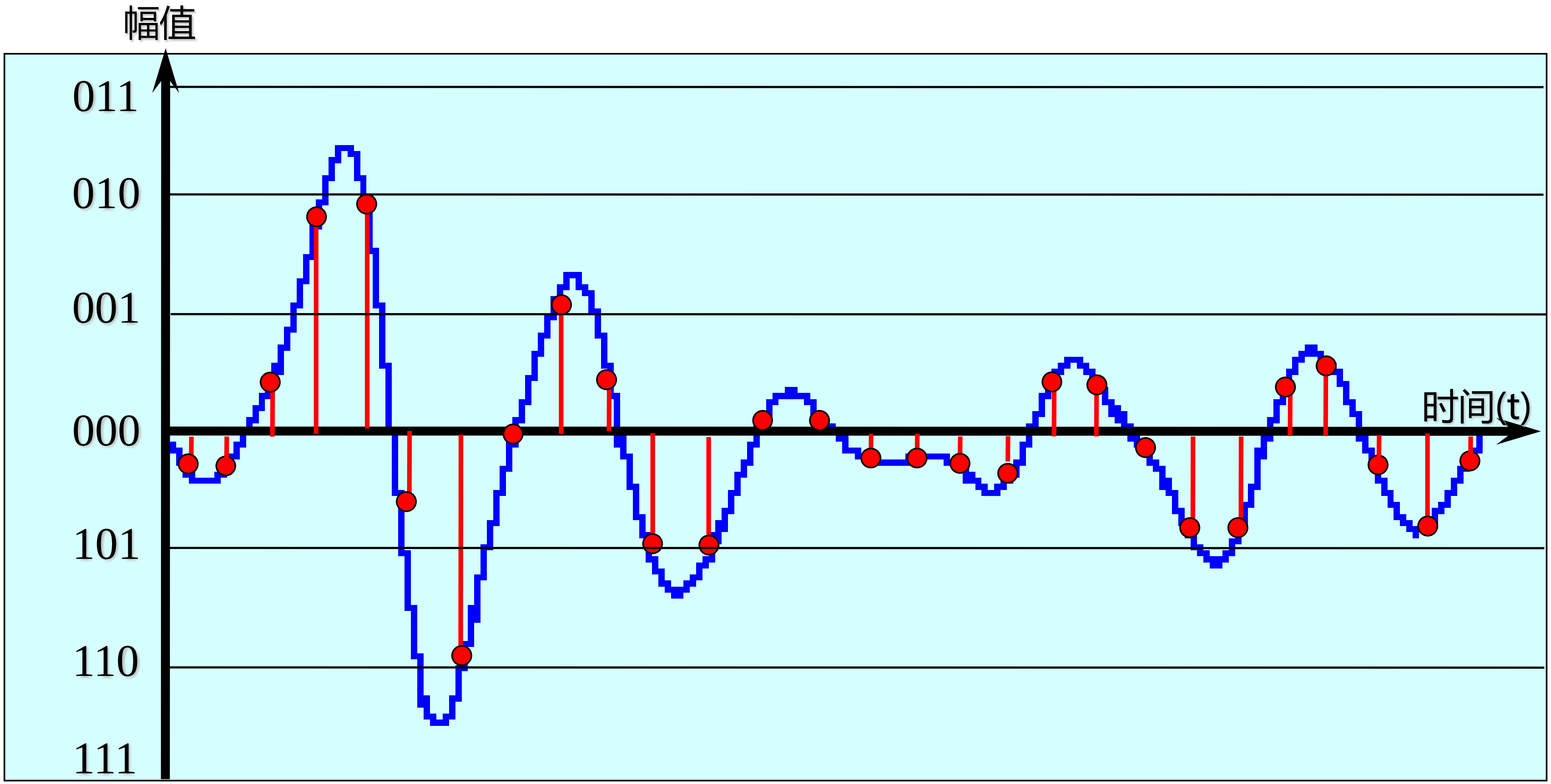
模拟信号数字化



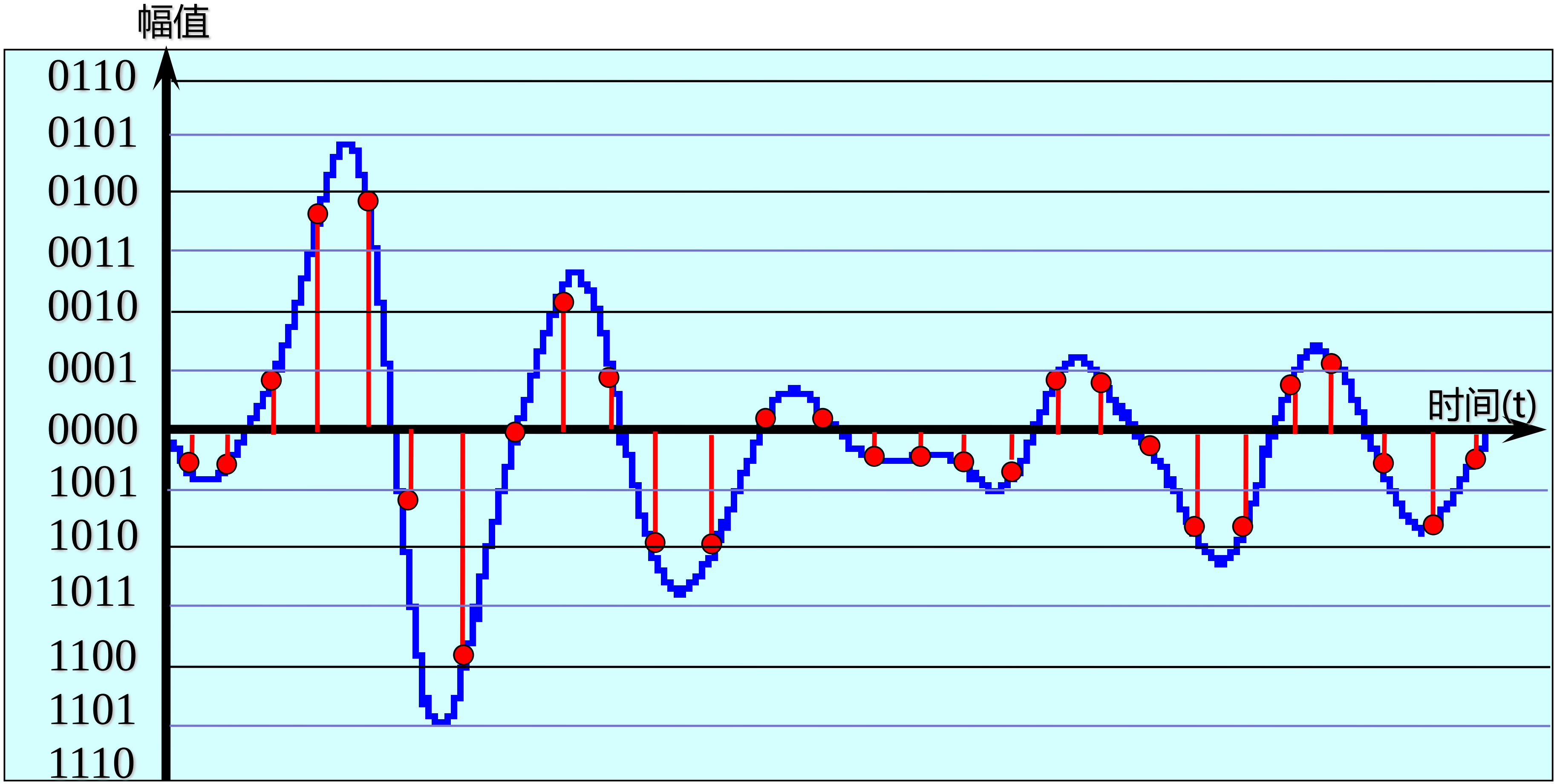
采样(Sampling)



量化(Quantization) 与编码 (Encoding)

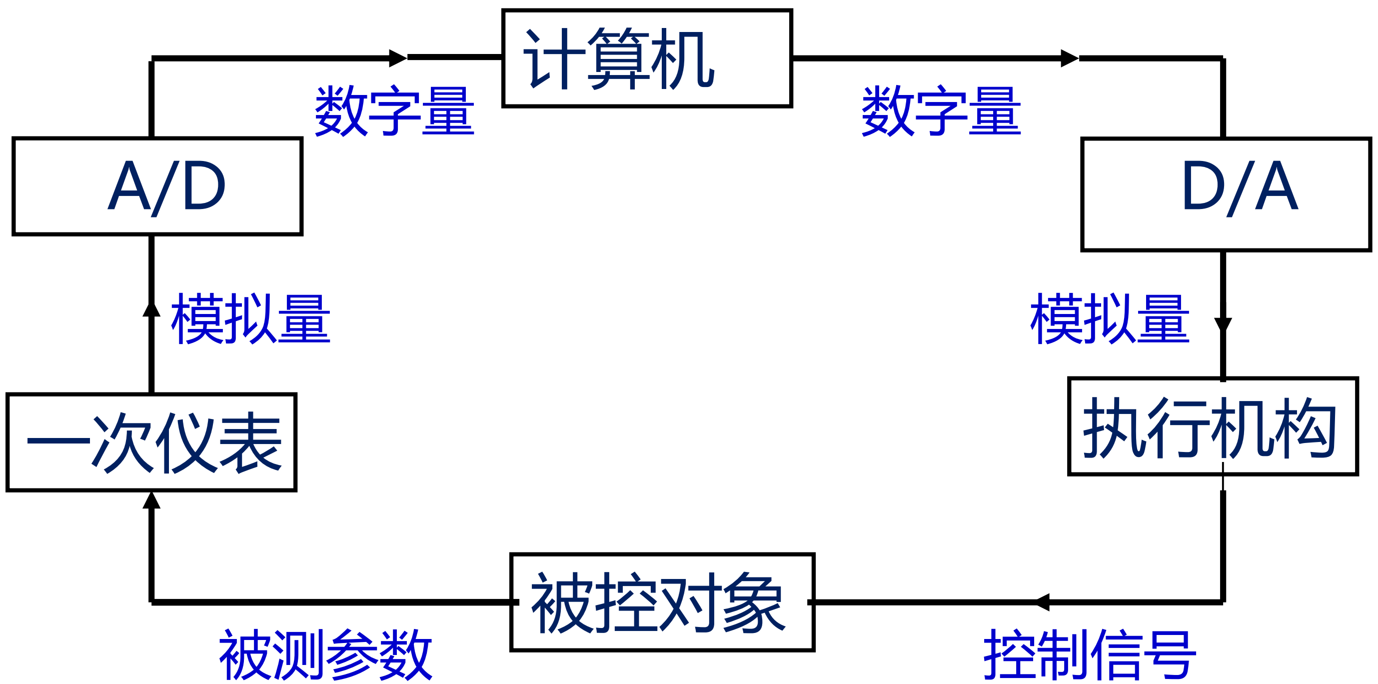


量化(Quantization) 与编码 (Encoding)



数字系统

- 例如，某控制系统框图如下图所示



数字逻辑电路

- 数字电路
  - 用来处理数字信号的电子线路
- 由于数字电路的各种功能是通过逻辑运算和逻辑判断来实现的，所以数字电路又称为数字逻辑电路或者逻辑电路
- 随着半导体技术和工艺的发展，出现了数字集成电路，集成电路发展十分迅速

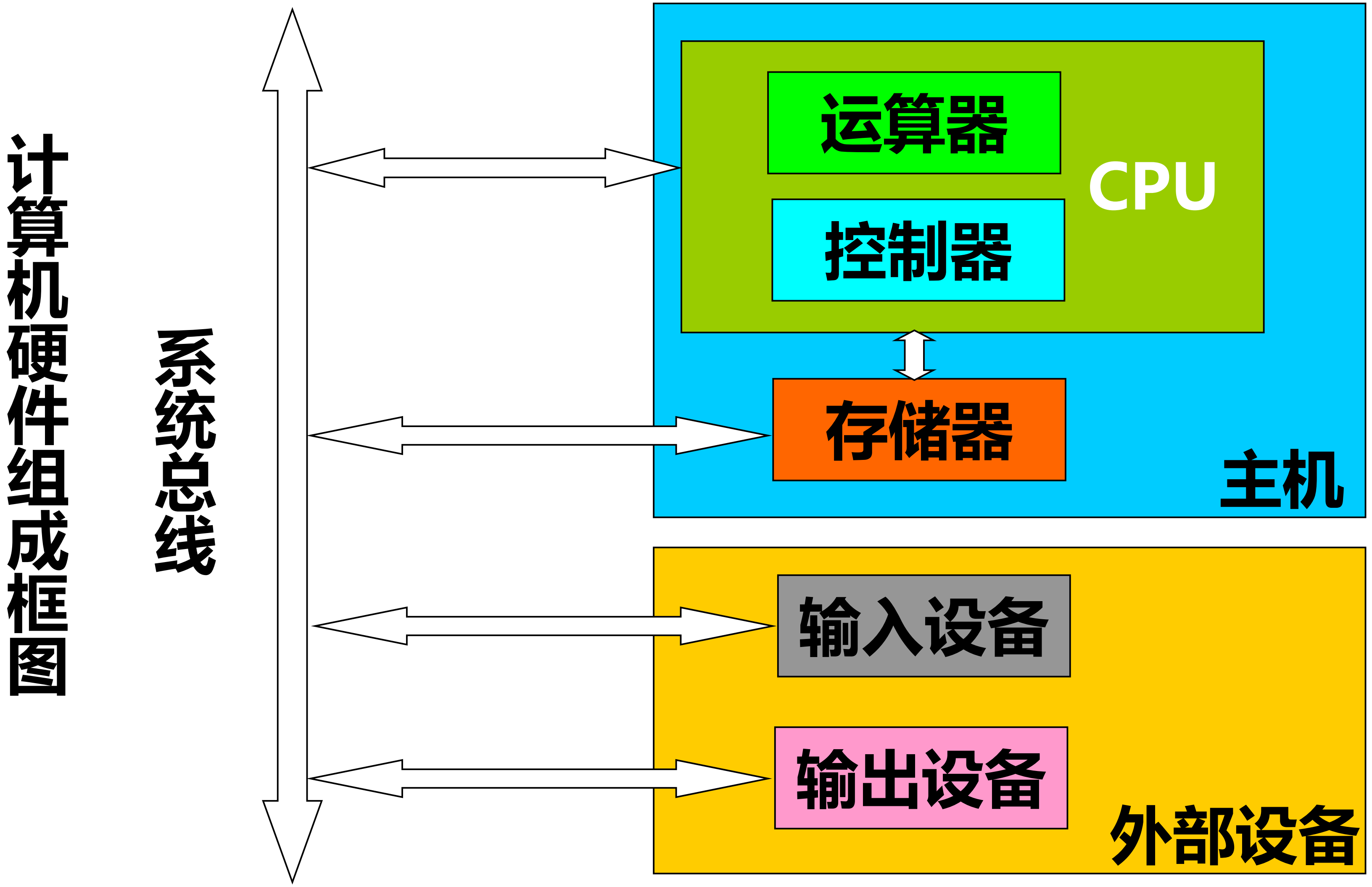
数字逻辑电路的特点

- 电路的基本工作信号是二值信号
- 电路中的半导体器件一般都工作在开、关状态
- 电路结构简单、功耗低、便于集成制造和系列化生产，产品价格低廉、使用方便、通用性好
- 工作速度快、精度高、功能强、可靠性好

数字系统的层次结构

第 5级	复杂数字系统，含第2到第4级的逻辑部件	
第 4 级	复杂逻辑功能部件，如微处理器	VLSI级
第 3 级	逻辑功能部件，如加法器、计数器、乘法器	MSI、LSI级
第 2 级	基本逻辑器件，如逻辑门、触发器等	SSI级
第 1 级	电子元件，如二极管、三极管、电阻、电容	

典型的数字系统——数字计算机



数字计算机的发展

表1.1 数字计算机的发展

划 代	主要元器件	生产时间	国 家
第一代	电子管	1946年	美 国
第二代	晶体管	1958年	美 国
第三代	小规模集成电路	1964年	美 国
第四代	中、大规模集成电路	1971年	美 国

计算机的发展趋势：速度↑、功能↑、可靠性↑、体积↓、价格↓、功耗↓?

数字逻辑电路的类型

- 根据一个电路有无记忆功能，可以分为：
  - 组合逻辑电路（Combinational Logic Circuit）
  - 时序逻辑电路（Sequential Logic Circuit）
- 组合逻辑电路
  - 在任何时刻的稳定输出仅取决于该时刻的输入，而与电路过去的输入无关

数字逻辑电路的类型

□时序逻辑电路

– 在任何时刻的稳定输出不仅取决于该时刻的输入，而且与过去的输入相关

数字逻辑电路的类型

- 时序逻辑电路按照是否有统一的时钟信号进行同步，可分为：
  - 同步时序逻辑电路
  - 异步时序逻辑电路
- 组合逻辑电路、同步时序逻辑电路和异步时序逻辑电路将分别在第4、5和6章讲述

数字逻辑电路的研究方法

- 逻辑电路的研究有两个主要任务
  - 一是分析
  - 二是设计
- 逻辑分析
  - 对一个已有的数字逻辑电路，研究它的工作性能和逻辑功能
- 逻辑设计
  - 根据提出的逻辑功能，在给定条件下构造并实现预定功能，又称为逻辑综合

数字逻辑电路的研究方法

- 传统方法
  - 以逻辑代数作为基本理论，从逻辑抽象到功能实现
  - 建立在小规模集成电路基础之上
  - 以技术经济指标作为评价一个设计方案优劣的主要性能指标
  - 设计时追求的是如何使一个电路达到最简
- 注意
  - 一个最简的方案并不等于一个最佳的方案

数字逻辑电路的研究方法

- 非传统方法
  - 用中、大规模集成组件进行逻辑设计
  - 用可编程逻辑器件（PLD）进行逻辑设计
  - 用计算机进行辅助逻辑设计

提 纲

1 数字信号与系统

2 数制及其转换

3 带符号二进制数的代码表示

4 几种常用的编码

提 纲

1 数字信号与系统

2 数制及其转换

3 带符号二进制数的代码表示

4 几种常用的编码

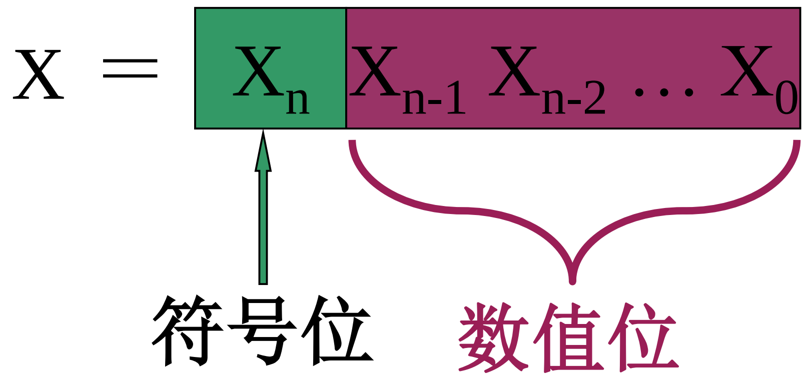
带符号二进制数的代码表示

- 机器数
- 原码表示法
- 补码表示法
- 反码表示法

机器数

- 两种表示方法：真值法和机器数（机器码）
  - 真值：一般书写形式表示的数，通常用“+”、“-”表示正负
  - 机器数：正负符号数码化后的数据
- 机器数实际上是数据在机器中的表示形式，是由数据的符号位和数值部分一起编码而成
- 常用机器码
  - 原码、补码、反码

符号位的表示



$$\begin{cases} X_n=0，\quad X \geq 0 \\ X_n=1，\quad X \leq 0 \end{cases}$$

原码表示法

- 原码又称之为符号数值表示法，是一种比较直观的编码表示法
- 符号位表示了数据的正或负
- 数码 0 表示正号，数码 1表示负号
- 数值部分保留了真值的特征，为真值的绝对值
- 简言之：符号位+真值绝对值

原码的定义

- 整数原码的定义

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

- 小数原码的定义

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

整数原码举例

例1： $x=+1101$

$[x]_{原}=01101$

整数原码举例

例2：　$x=-1101$

$$[x]_{原}=2^4-(-1101)$$
$$=10000+1101$$
$$=11101$$

小数原码表示举例

例1： $x=0.1011$

$[x]_{原}=0.1011$

小数原码表示举例

例2：　$x = -0.1011$

$$[x]_{原} = 1 - x$$
$$= 1 - (-0.1011)$$
$$= 1 + 0.1011$$
$$= 1.1011$$

十进制数的原码求法举例

例：　$x=25$

$=16+8+1$

$=10000+1000+1$

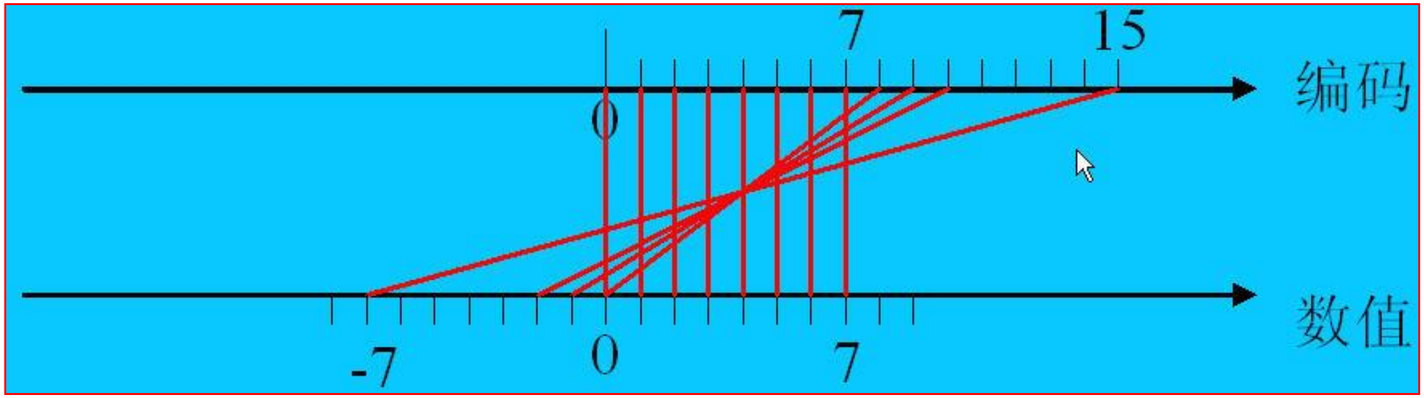
$=11001$

$[x]_{原}=011001$

带符号数的原码表示

原　码	真值	原　码	真值
0000	+0	1000	-0
0001	+1	1001	-1
0010	+2	1010	-2
0011	+3	1011	-3
0100	+4	1100	-4
0101	+5	1101	-5
0110	+6	1110	-6
0111	+7	1111	-7

原码在数轴上的表示



原码表示法的特点

- 零的表示有“+0”和“-0”之分
  - $[+0.00\ldots0]_{原}=0.00\ldots0$
  - $[-0.00\ldots0]_{原}=1.00\ldots0$

※ 正数的原码是其本身，负数的原码的符号位为1，数值位不变

原码的加法

- 假设字长为8bits
- 十进制运算：$(1)_{10} + (1)_{10} = 2$
- 二进制运算

$$(1)_2 + (1)_2$$
$$= 00000001 + 00000001$$
$$= 00000010$$
$$= 2$$

正确

原码的加法

- 假设字长为8bits
- 十进制运算：$(-1)_{10}+(-1)_{10}=-2$
- 二进制运算

$(-1)_2+(-1)_2$

$=10000001+10000001$

$=100000010$

$=?$

数值位：2，符号位如何确定呢?

原码的减法

- 十进制运算：$(1)_{10} - (1)_{10} = 0$
- 二进制运算

$$(1)_2 - (1)_2$$

$$= (1)_2 + (-1)_2$$

$$= 00000001 + 10000001$$

$$= 10000010$$

$$= (-2)$$

正确吗？

原码的减法

- 十进制运算：$(1)_{10} - (2)_{10} = -1$
- 二进制运算

$(1)_2 - (2)_2$

$= (1)_2 + (-2)_2$

$= 00000001 + 10000010$

$= 10000011$

$= (-3)$

正确吗?

结论

- 原码不能直接进行减法运算
- 正确的做法
  - 当对两个数求和时，如果符号相异，则需要先比较两个数的绝对值的大小，然后做减法
  - 绝对值大的符号是结果的符号
  - 绝对值的差值是结果的数值位
- 缺点
  - 利用它进行加减法运算较为麻烦

问题

- 如何简化问题呢?
- 解决办法
  - 减法变加法，符号位直接参与运算
- 原码不行，有其他编码吗?

反码表示法

- 符号位与原码相同
- 数值位与符号位相关
  - 正数的反码是正数本身，与原码形式相同
  - 负数的反码符号位为1，其数值部分由原码的数值部分按位取反得到

反码表示法

整数反码的定义

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

小数反码的定义

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

举例说明

- $X=+0.1011$
- $X=-0.1011$,
  - $[X]_{反}=(2-2^{-4})-0.1011$

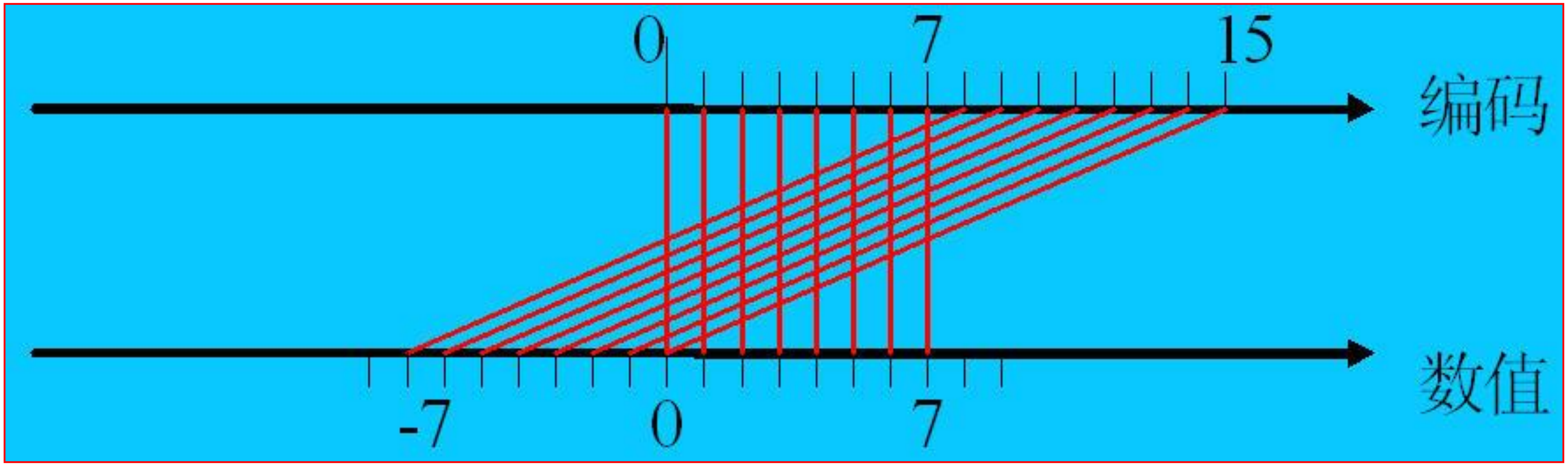
    $=1.1111-0.1011$

    $=1.0100$
- 求$X=-13/16$的二进制反码?
  - $[X]_{反}=[-0.1101]_{反}=1.0010$

带符号数的反码表示

反 码	真 值	反 码	真 值
0000	+0	1000	-7
0001	+1	1001	-6
0010	+2	1010	-5
0011	+3	1011	-4
0100	+4	1100	-3
0101	+5	1101	-2
0110	+6	1110	-1
0111	+7	1111	-0

反码在数轴上的表示



负数的反码加法

- 假设字长为8bits
- 十进制运算：$(-1)_{10} + (-1)_{10} = -2$
- 二进制运算

  $(-1)_2 + (-1)_2$

  $= 10000001 + 10000001$【原码】

  $= 11111110 + 11111110$【反码】

  $= 111111100$【反码】

  $= 11111101 = -2$

符号位进位加到结果最低位

反码的减法

- 十进制运算：$(1)_{10}-(2)_{10}=-1$
- 二进制运算

$$
\begin{aligned}
&(1)_2-(2)_2\\
=\ &(1)_2+(-2)_2\\
=\ &00000001+10000010\text{【原码】}\\
=\ &00000001+11111101\text{【反码】}\\
=\ &11111110\text{【反码】}\\
=\ &(-1)
\end{aligned}
$$

正确吗?

反码的减法

- 十进制运算：$(1)_{10}-(1)_{10}=0$
- 二进制运算

$$
\begin{aligned}
&(1)_2-(1)_2\\
&=(1)_2+(-1)_2\\
&=00000001+10000001 \text{【原码】}\\
&=00000001+11111110 \text{【反码】}\\
&=11111111 \text{【反码】}\\
&=-0
\end{aligned}
$$

合理吗?

问题

- 还有更合理的表示方法吗?

模的概念

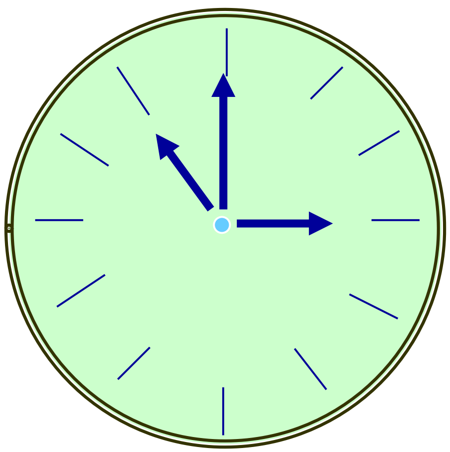
- “模”是指一个计量系统的计数范围
  - 如时钟的计量范围是0～11，模=12
- 计算机也可以看成一个计量机器，它也有一个计量范围，即都存在一个“模”
  - 表示n位的计算机计量范围是$0 \sim 2^n-1$，模$= 2^n$
- “模”实质上是计量器产生“溢出”的量，它的值在计量器上表示不出来，计量器上只能表示出模的余数
- 任何有模的计量器，均可化减法为加法运算

补码的特征

- 标准时间3点整，有一只表的当前时间为11点，如何校准时间？

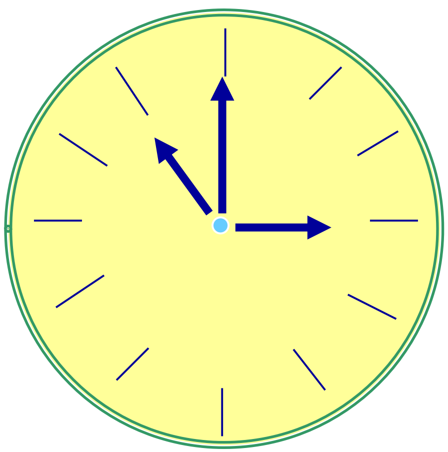
方法一：

顺时针转4个小时



方法二：

逆时针转8个小时



补码的特征

- 结论：两种方法是等效的，这种关系记作：

  -8=4（mod 12）

- 含义：-8与4对模12是互补的，或者说以12为模时-8的补码为4
- 模（或称模数）：一个计量系统的范围，记作mod或M
- 同理，模为12时，-2的补码是10，-5的补码是7

补码表示法

整数补码的定义

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

小数补码的定义

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

补码表示法

- 正数与原码相同
- 负数的补码的符号位为1，数值位为其反码的末位加1
- 如：$[-1010]_{补}=1[1010]_{反}+1=10101+1=10110$
- 对于0，在补码中的定义下只有一种表示形式：

$$[+0.00\ldots0]_{补}=[-0.00\ldots0]_{补}=0.00\ldots0$$

补码的意义

- 计算机中的数据受字长的限制，数据的运算属于有模运算
- 计算结果能够方便地按模丢掉
- 可以将减法转为加法运算
- 计算机中可只设置加法器，从而可简化设计、降低成本

求补码举例

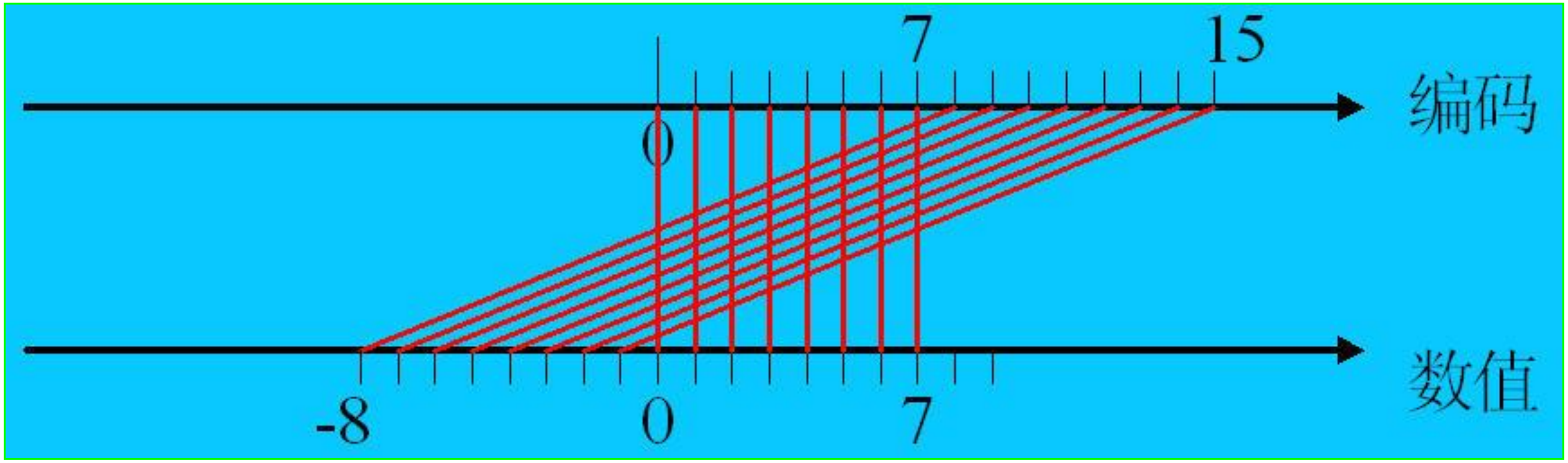
- 若X=0.1010, 则$[X]_{补}$=0.1010
- 若X=-0.1010, 则$[X]_{补}$=2-0.1010=1.0110
- 若X=1010, 则$[X]_{补}$=1010
- 若X=-1010, 则$[X]_{补}$=2^5-1010

  =10000-1010=10110

带符号数的原码、反码和补码表示

	原码	反码	补码		原码	反码	补码
0000	+0	+0	0	1000	-0	-7	-8
0001	1	1	1	1001	-1	-6	-7
0010	2	2	2	1010	-2	-5	-6
0011	3	3	3	1011	-3	-4	-5
0100	4	4	4	1100	-4	-3	-4
0101	5	5	5	1101	-5	-2	-3
0110	6	6	6	1110	-6	-1	-2
0111	7	7	7	1111	-7	-0	-1

补码在数轴上的表示



求补码?

- $X=0.11111111$, $[X]_{补}=?$
- $[X]_{补}=0.11111111$
- $X=-0.11111111$, $[X]_{补}=?$
- $[X]_{补}=1.00000000+0.00000001=1.00000001$
- $X=-0.10101001$, $[X]_{补}=?$
- $[X]_{补}=1.01010110+0.00000001=1.01010111$

负数的补码加法

- 假设字长为8bits
- 十进制运算：$(-1)_{10}+(-1)_{10}=-2$
- 二进制运算

  $(-1)_2+(-1)_2$

  $=10000001+10000001$

  $=11111111+11111111$【补码】

  $=111111110$【补码】

  $=11111110=-2$

符号位进位直接丢掉

补码的减法

- 十进制运算：$(1)_{10}-(2)_{10}=-1$
- 二进制运算

$$
\begin{aligned}
&(1)_2-(2)_2\\
&=(1)_2+(-2)_2\\
&=00000001+10000010\text{【原码】}\\
&=00000001+11111110\text{【补码】}\\
&=11111111\text{【补码】}\\
&=(-1)
\end{aligned}
$$

正确吗?

补码的减法

- 十进制运算：$(1)_{10} - (1)_{10} = 0$
- 二进制运算

$(1)_2 - (1)_2$

$= (1)_2 + (-1)_2$

= 00000001 + 10000001 【原码】

= 00000001 + 11111111 【补码】

= 100000000【补码】

= 00000000=0 符号位进位丢弃

合理

机器数的应用

- 补码的加减法比较方便，得到了广泛应用
- 目前计算机中广泛采用补码表示
- 少数机器采用原码进行存储和传输，计算时用补码表示

机器码的求法对比

机器码	真值为正数	真值为负数
原　码	符号位为0，等于真值	符号为1，等于真值
反　码	符号位为0，等于真值	符号为1，逐位取反
补　码	符号位为0，等于真值	符号为1，逐位取反，末位加1

例 子

真 值	+10001111	-10001111	+0.10011111	-0.10011111
原 码	010001111	110001111	0.10011111	1.10011111
反 码	010001111	101110000	0.10011111	1.01100000
补 码	010001111	101110001	0.10011111	1.01100001
三种表示方法均有符号位和数值位两部分，符号位都是用1表示“负”，用0表示“正”，而数值位各不相同				

提 纲

1 数字信号与系统

2 数制及其转换

3 带符号二进制数的代码表示

4 几种常用的编码

几种常用的编码

- 十进数的二进制编码
- 可靠性编码
- 字符编码

十进制数的二进制编码

- 十进制的二进制表示（BCD码）
  - 8421码
  - 2421码
  - 5421码
  - 余3码
- 十进制数串在机器中的表示

十进制数的二进制编码

十进制数	8421 码	2421 码	余 3 码
0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0
末	1 0 1 0	0 1 0 1	0 0 0 0
选	1 0 1 1	0 1 1 0	0 0 0 1
用	1 1 0 0	0 1 1 1	0 0 1 0
的	1 1 0 1	1 0 0 0	1 1 0 1
编	1 1 1 0	1 0 0 1	1 1 1 0
码	1 1 1 1	1 0 1 0	1 1 1 1

十进制数串在机器中的表示

- 字符串形式

  一个字节存放一个十进制位（8421）

- 压缩的十进制数串

  一个字节存放两个十进制位，节省一半的存储空间。

符号位用8421码中未选用的编码：
正号用1100，负号用1101

-3 5 6

0011
0101
0110
1101（符号位）

-3 5 6

0011	0101
0110	1101

可靠性编码

- 为了减少或者发现代码在形成和传送过程中都可能发生的错误
- 可靠性编码的作用是为了提高系统的可靠性
- 下面介绍几种：
  - 奇偶校验码
  - CRC校验码
  - 格雷码

奇偶校验

- 奇偶校验包含着奇校验和偶校验两种校验
  - 奇校验：让整个校验码（包含有效信息和校验位）中1的个数为奇数
  - 偶校验：是让整个校验码中1的个数为偶数
- 有效信息（被校验的信息）部分可能是奇性（1的个数为奇数）也可能是偶性
- 奇偶两种校验都只需配一个校验位，就可以使整个校验码满足指定的奇偶性要求

奇偶校验

校验码

$$B = b_1b_2b_3b_4b_5b_6b_7b_8 \quad P$$

被校验信息 **校验位**

奇校验

$$P = \overline{b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_8}$$

偶校验

$$P = b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_8$$

奇偶校验

校验位的取值

被校验信息	奇校验位取值	偶校验位取值
10 10 10 10	1	0
11 00 11 01	0	1
11 01 00 11	0	1
10 01 10 01	1	0
10 10 11 00	1	0
11 10 11 00	0	1

奇偶校验的特点

- 奇偶校验是一种常见的简单校验，只需要1位校验码
- 奇偶校验只具有发现错误的能力，不具备对错误定位继而纠正错误的能力
- 奇偶校验，只具有发现一串二进制代码中，同时出现奇数个代码出错的能力
- 如果同时发生偶数个代码出错，这种校验就不具备发现错误的能力了

奇偶校验的性能

00001	00001	00001	00001
00001	00000	01111	11111
正确传输	正常检错	正常检错	检错失效

循环冗余校验码（CRC）

- 一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。
- 特征
  - 信息字段和校验字段的长度可以任意选定

循环冗余校验码（CRC）

原理

- 增加附加信息
- 使所生成的数据能与发送端和接收端共同选定的某个特定数整除
- **模2除法：**每一位除的结果不影响其它位，即不向上一位借位

循环冗余校验码（CRC）

- 实例：已知多项式G（X）$=x^4 + x + 1$（CRC-4/ITU），发送数据为111010

（1）多项式二进制化 10011（k+1位）

（2）在发送数据后加k个0，1110100000

（3）用初始数用模2除法除以10011

（4）得到余数0010

（5）将初始传输码加上余数得到最终传输的数据1110100010

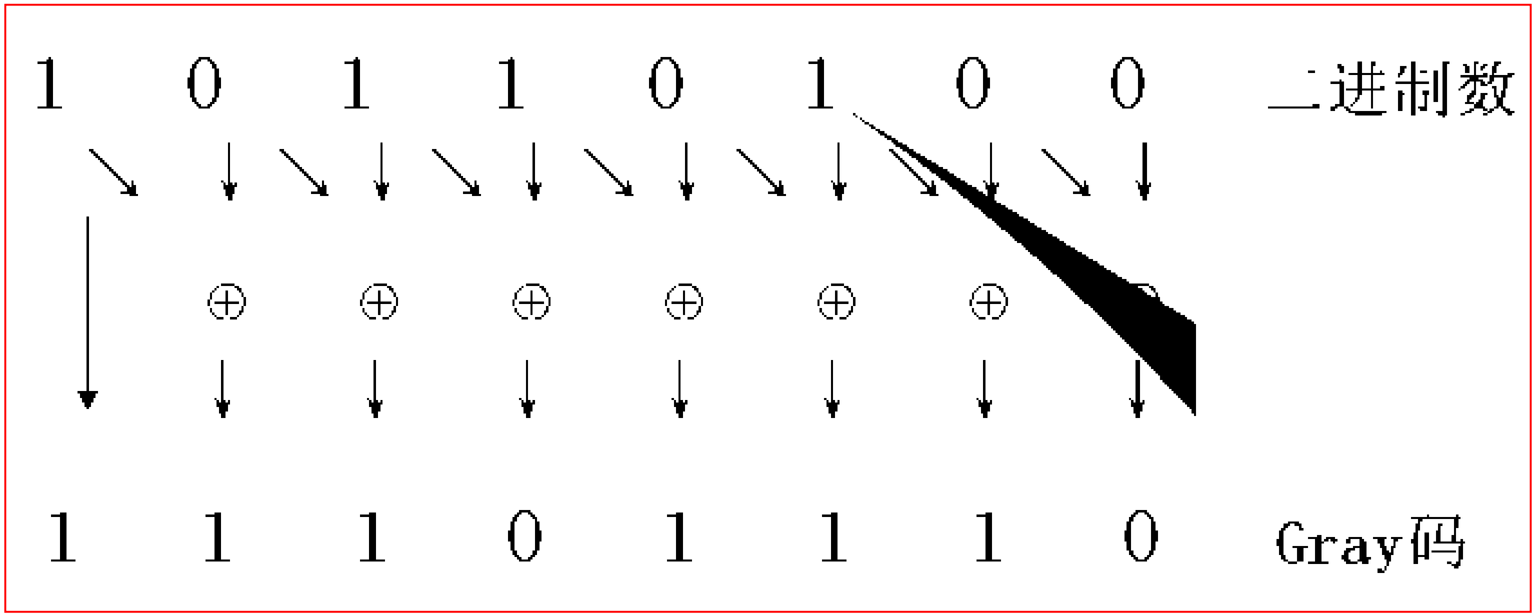
格雷码（Gray Code）

- **特点**
  - 任意两个相邻的数，其格雷码仅有一位不同
- **作用**
  - 避免代码形成或者变换过程中产生的错误

4位二进制码对应的典型格雷码

十进制数	4位二进制码	典型格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

格雷码的转换



问题解答

Thank You!