Beihang
SPINTRONICS
Interdisciplinary Center

数字集成电路基础

张悦

微电子学院

费尔北京研究院 / 自旋电子交叉学科中心

2020-10-2

课程信息

- 教师：张悦、康旺
- 助教：王硕、魏少芊
- 学时安排（56学时）
  - 课堂授课（48学时）
  - 习题/讨论/课堂测试（4学时）
  - 总复习课（2学时）
  - 考试（2学时）

考核方法

- 平时成绩（考勤+交流+作业）+期末考试

课程内容

一、数制与编码（2课时）

二、布尔代数与逻辑函数（6课时）

三、逻辑门电路（5课时）

四、组合逻辑电路（5课时）

五、触发器（6课时）

六、时序逻辑电路（10课时）

七、存储器（4课时）

八、可编程逻辑器件（2学时）

九、脉冲波形的产生与变换（6学时）

十、数模和模数转换（2学时）

学习方法

- 知识结构
  - **逻辑代数**是理论基础，熟练掌握
  - **单元电路**是物理基础，掌握逻辑功能、外部特性、功能扩展和使用方法
  - 掌握数字电路**系统**的**分析方法**和**设计方法**

- 学习方式
  - 课堂讲授
  - 主题研讨与小测试
  - 过程强调交互

- 学习要求
  - **预习**和**复习**
  - 主动**提出问题**，通过讨论加深理解
  - 将每道**习题**都作为设计项目

课内参考教材 （中文）：

1.阎 石 主编：数字电子技术基础（第五版），
高等教育出版社.（面向二十一世纪教材）

课内参考教材 （外文）：

1. Digital Logic Circuit Analysis and Design
Victor P.Nelson 等著
清华大学出版社 （英文影印版）

2. Digital Fundamentals (Seventh Edition)
Thomas L.Floyd 著
科学出版社 （英文影印版）

第一台计算机

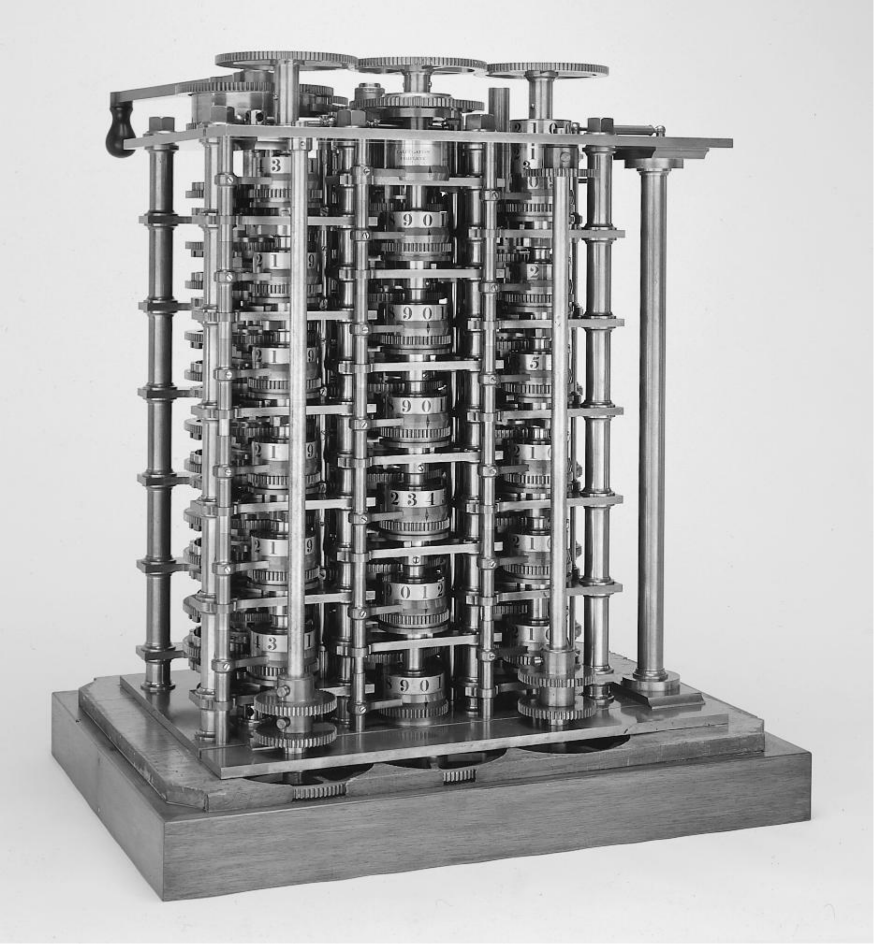
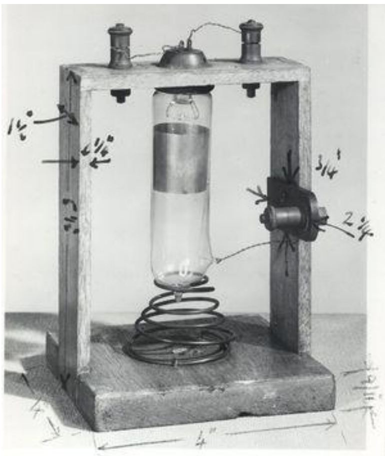
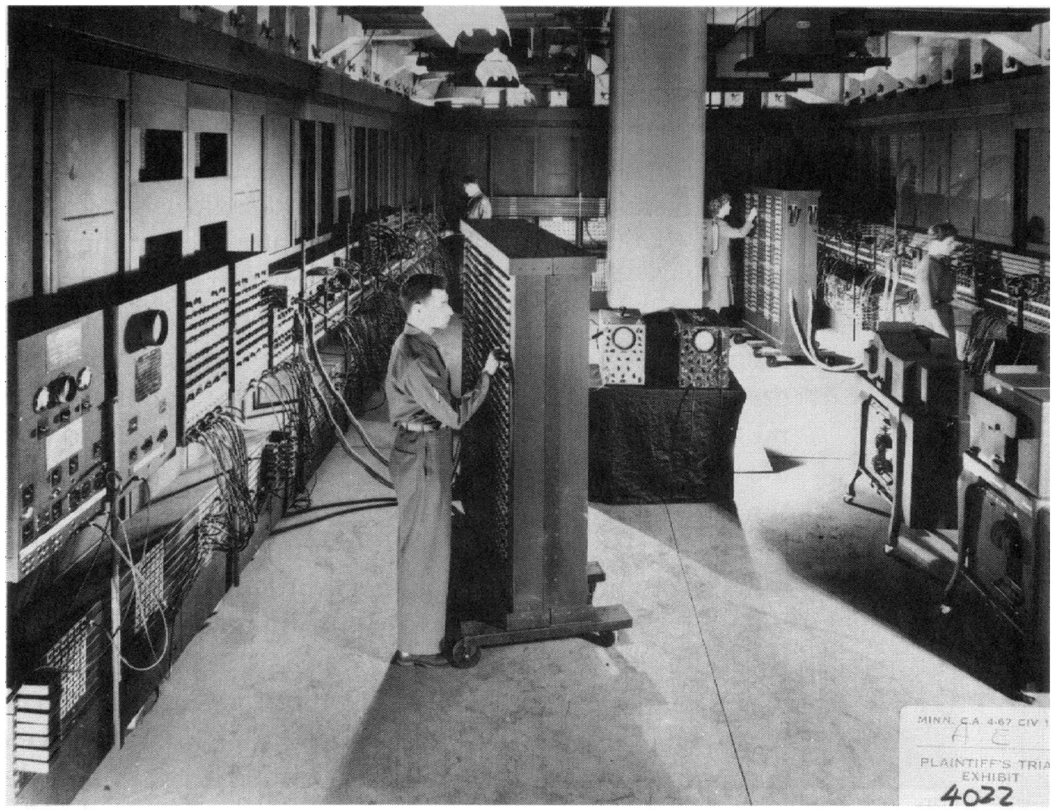


图1.1 世界上已知的第一个自动计算器
Babbage的Difference Engine I
（1832年）的工作部件
（摘自［Swade93］，由伦敦科学博物馆提供）

ENIAC – 第一台电子计算机 (1946)

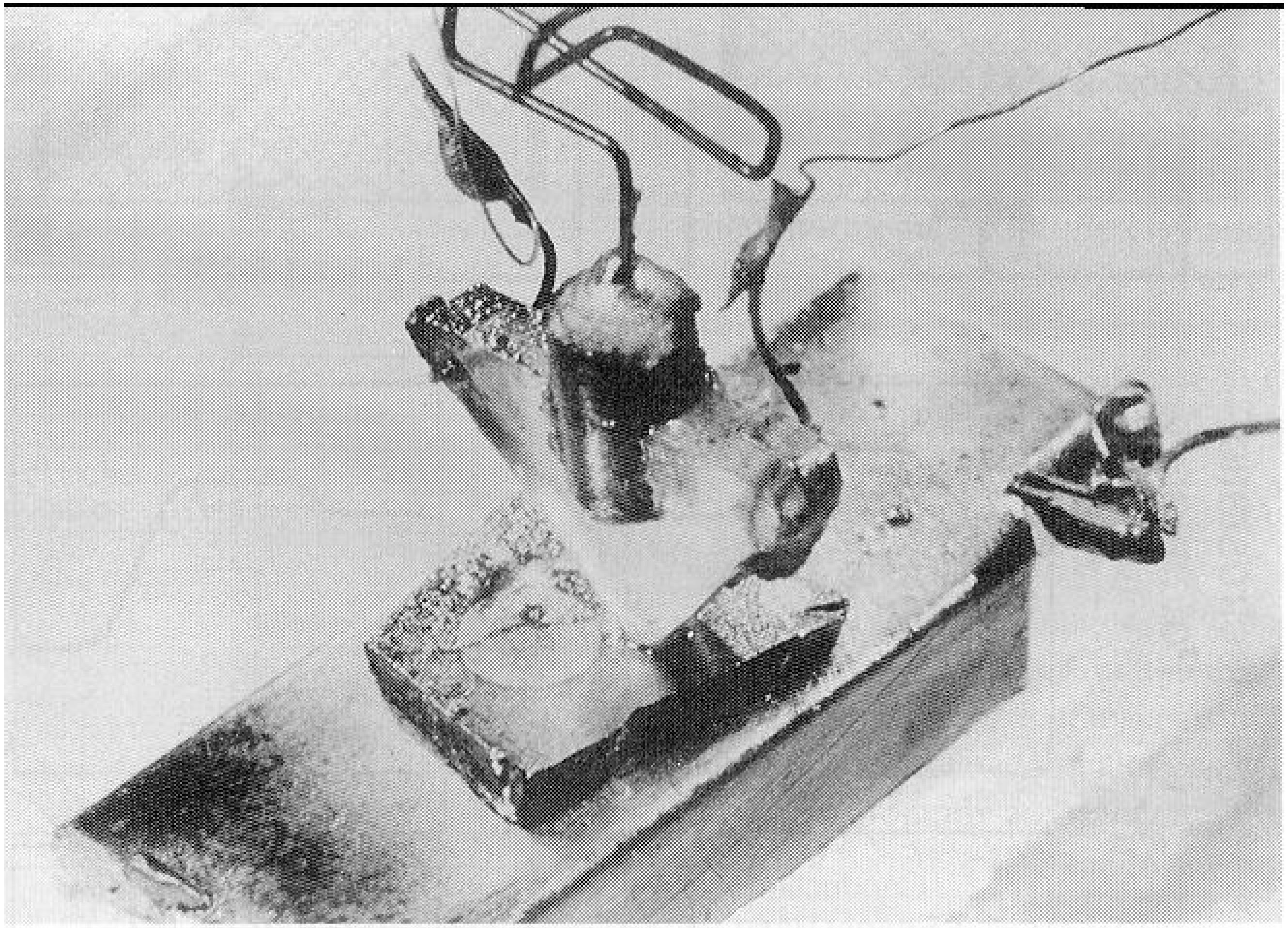


电子管



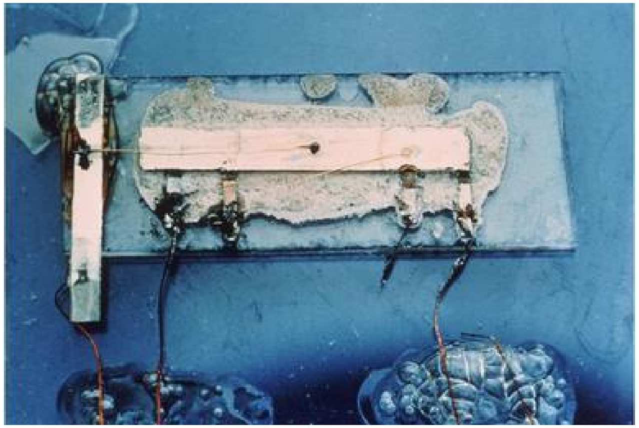
17468个电子管，150KW

晶体管革命

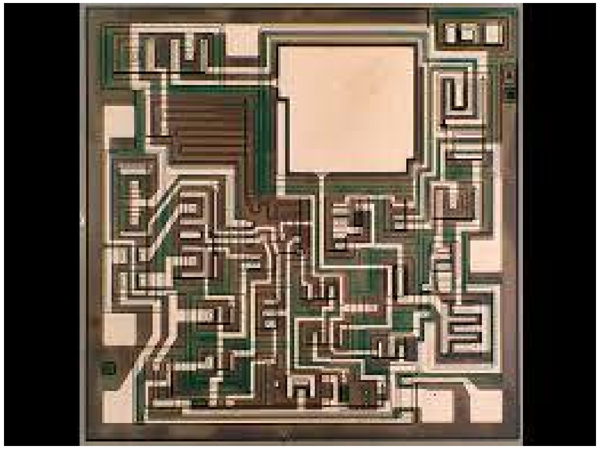


第一支晶体管
贝尔实验室, 1947

第一个集成电路

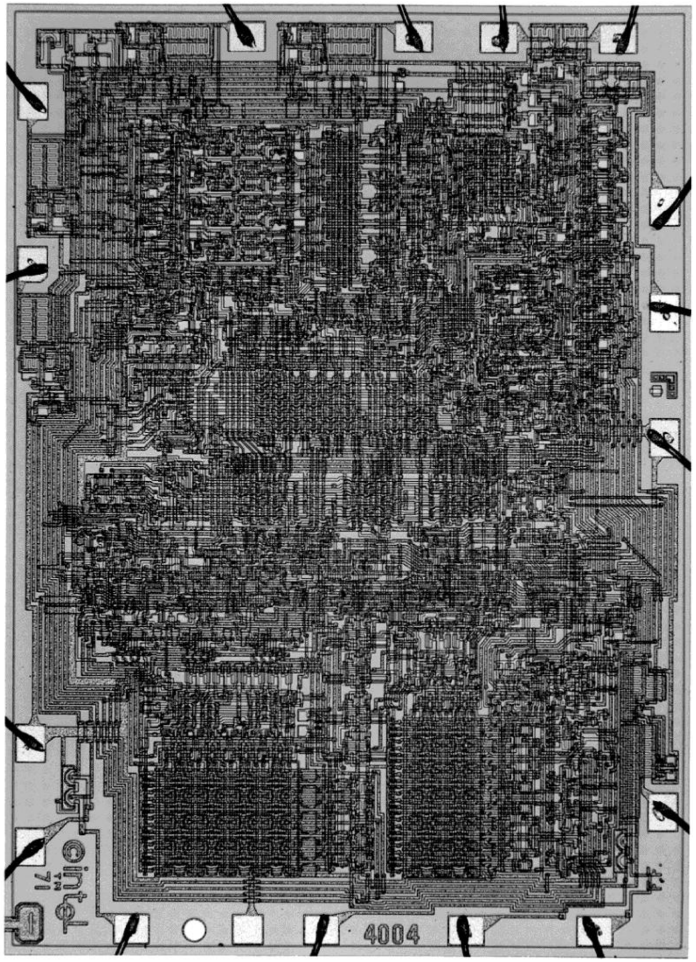


Jack Kilby （2000 Nobel Prize）
Texas Instruments （TI） 1958



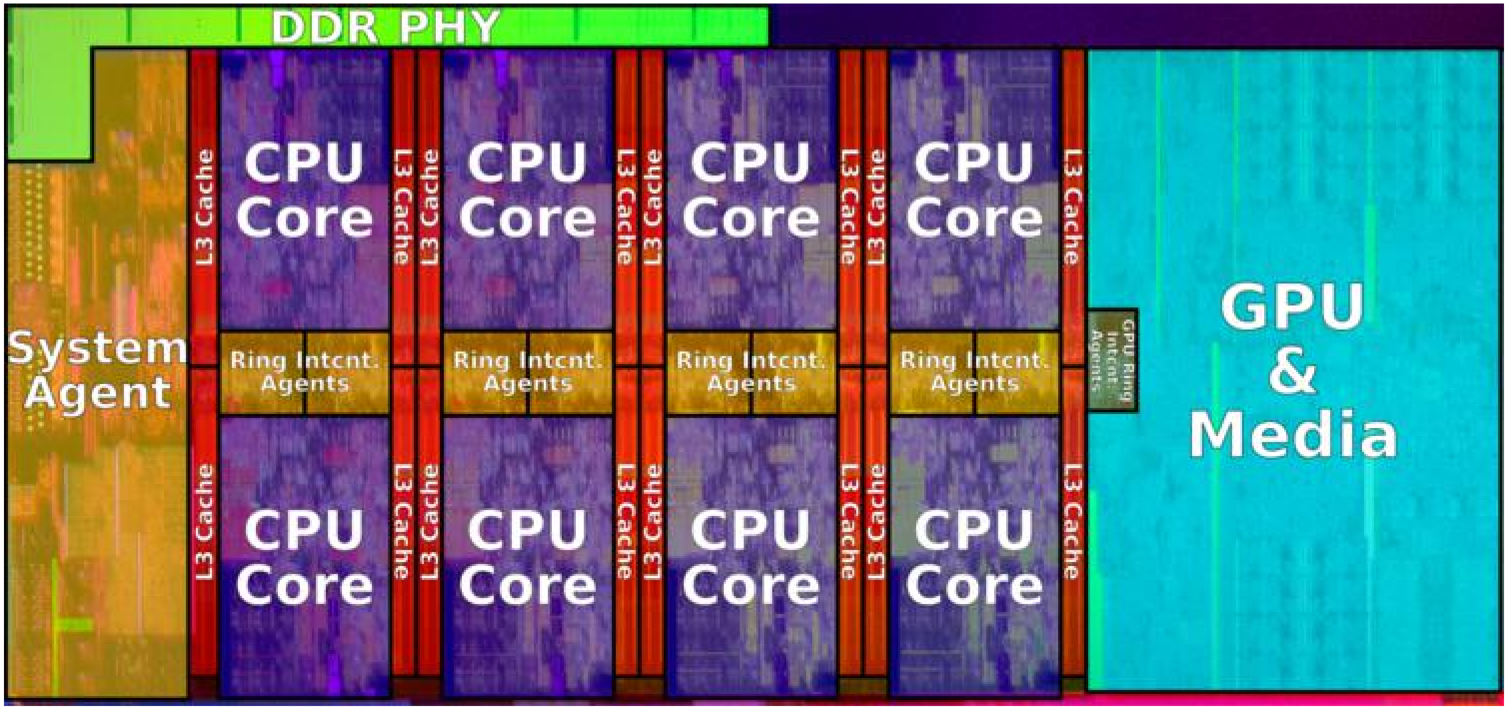
741运算放大器电路
仙童公司 1963

Intel 4004 微处理器

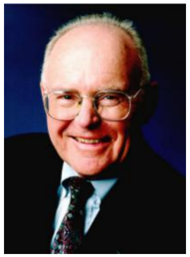


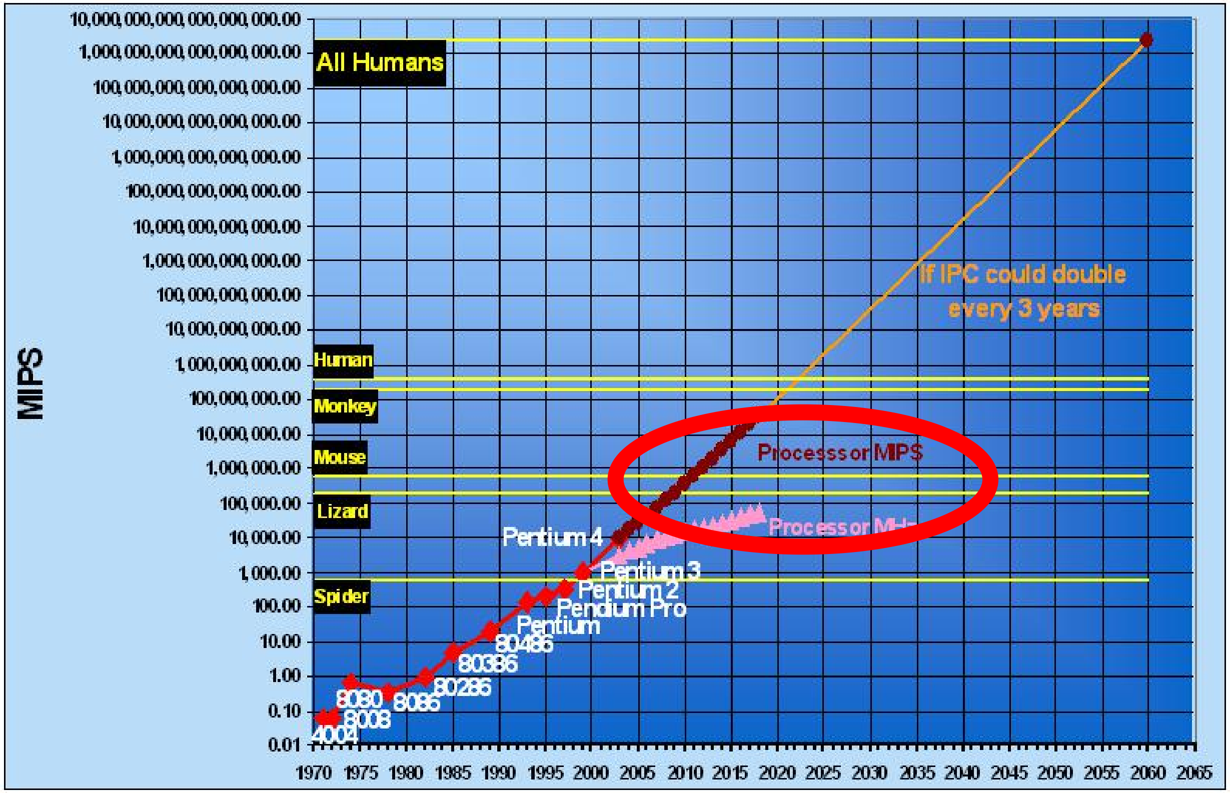
1971
1000个晶体管
1 MHz操作频率

2019
> 70亿个晶体管
5 GHz 单核操作频率



摩尔定律

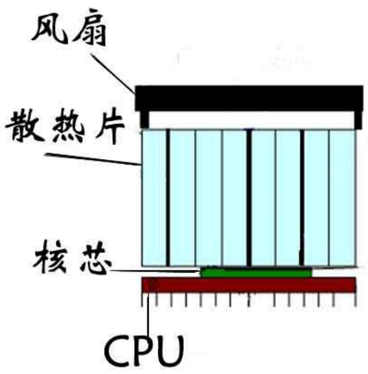




摩尔定律是由英特尔（Intel）创始人之一戈登·摩尔（Gordon Moore）提出来的。其内容为：当价格不变时，集成电路上可容纳的元器件的数目，约每隔18-24个月便会增加一倍，性能也将提升一倍。换言之，每一美元所能买到的电脑性能，将每隔18-24个月翻一倍以上。这一定律揭示了信息技术进步的速度。

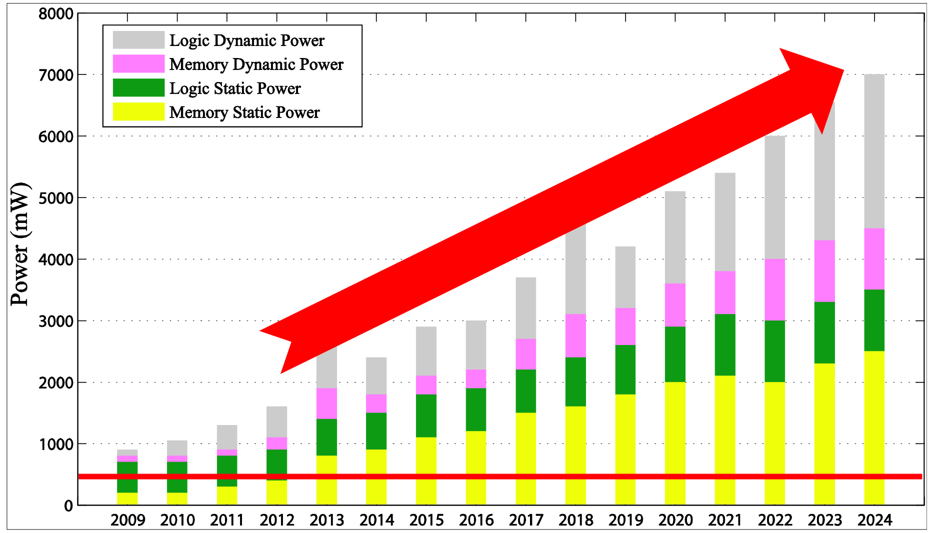
散热是个大问题！

CPU



GPU

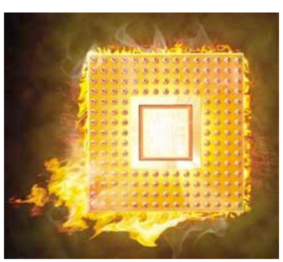
功耗问题

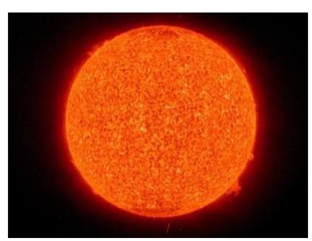


Logic Dynamic Power
Memory Dynamic Power
Logic Static Power
Memory Static Power
Power (mW)
8000
7000
6000
5000
4000
3000
2000
1000
0
2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024

功耗持续增加！！

nature International weekly journal of science
2012
Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For A

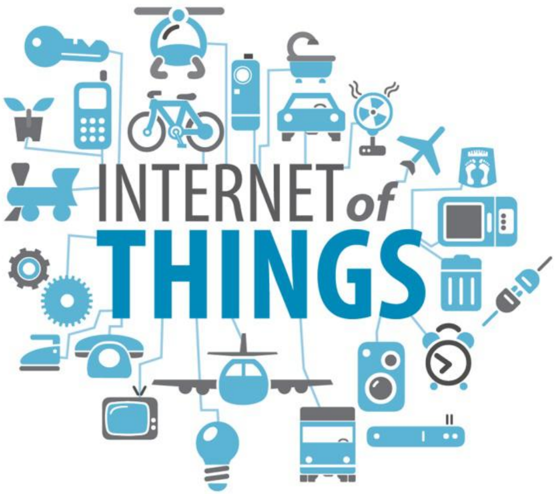


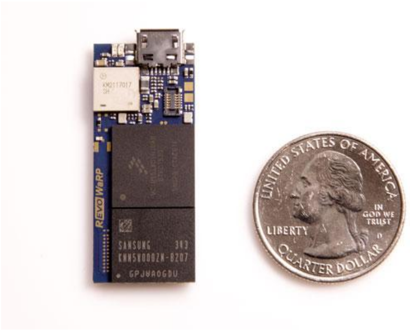


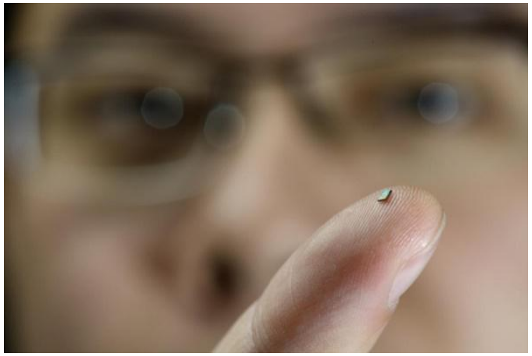
nature International weekly journal of science
2016
Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio &
Archive > Volume 530 > Issue 7589 > News Feature > Article
NATURE | NEWS FEATURE
عربي
The chips are down for Moore's law
The semiconductor industry will soon abandon its pursuit of Moore's law. Now things could get a lot more interesting.

物联网-IoT时代

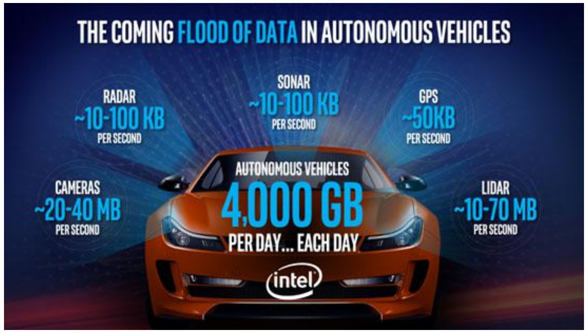
第三次信息革命







5G



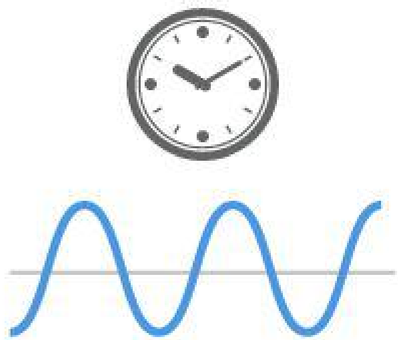
物联网终端——
形形色色的微型电路

模拟电路与数字电路

- 模拟信号
  - 时间上连续 或 数值上连续的信号
  - 来自于自然界客观存在的物理量
- 模拟电路
  - 处理模拟信号的电路

- 数字信号
  - 时间和数值均离散的信号
  - 例如：电子表的计时信号、流水线上的零件数
- 数字电路：
  - 处理数字信号的电路

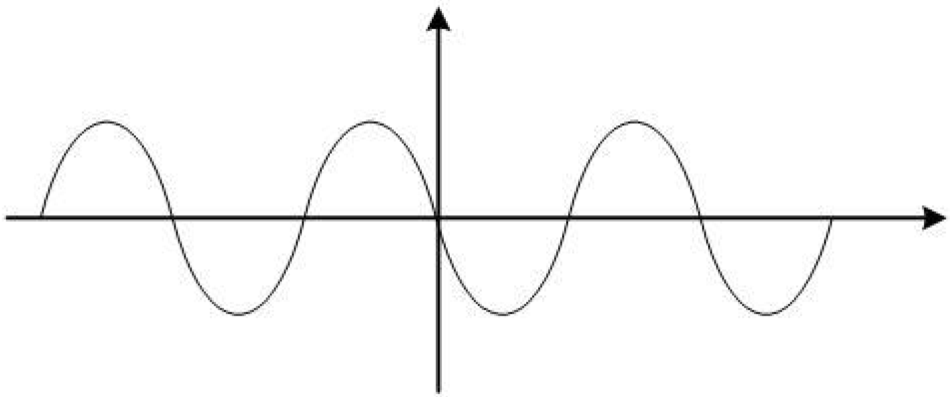
模拟



数字

模拟电路与数字电路

1）信号不同

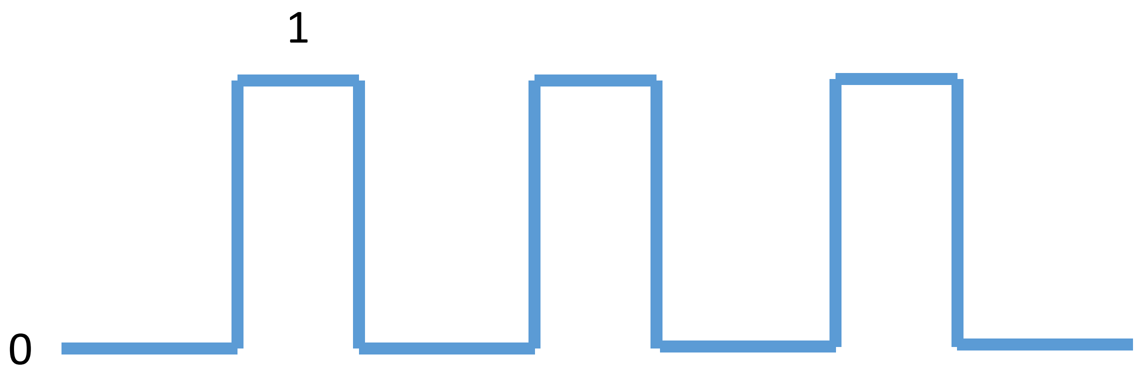


$$y=A\sin(\omega t)$$

正负峰值之间连续取值

模拟电路与数字电路

1）信号不同



在电路中用低、高电平表示0、1两种逻辑状态

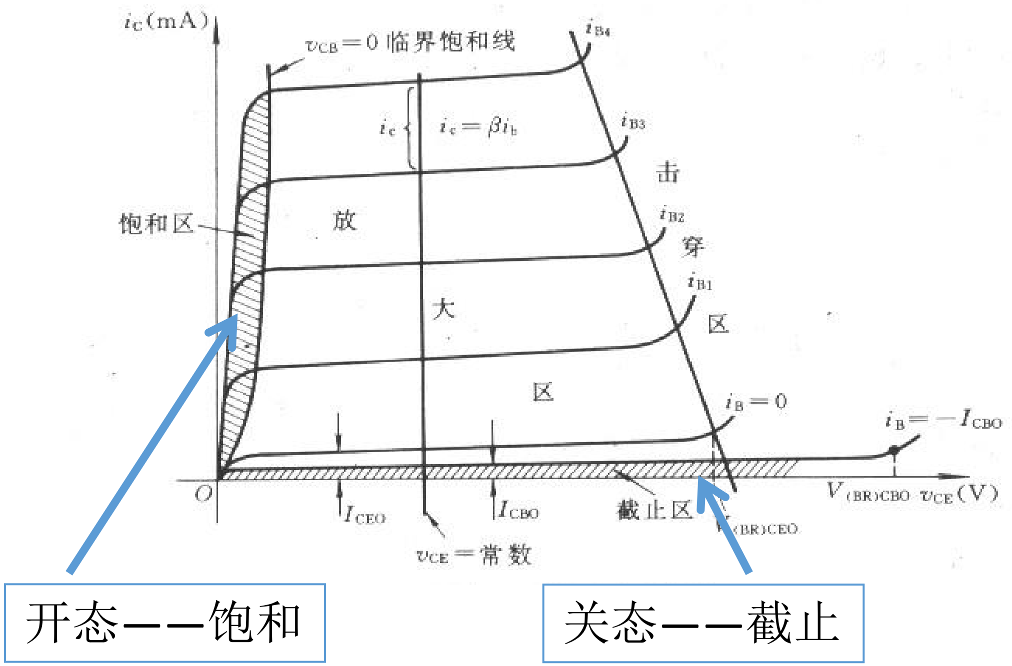
逻辑电平与电压值的关系（正逻辑）

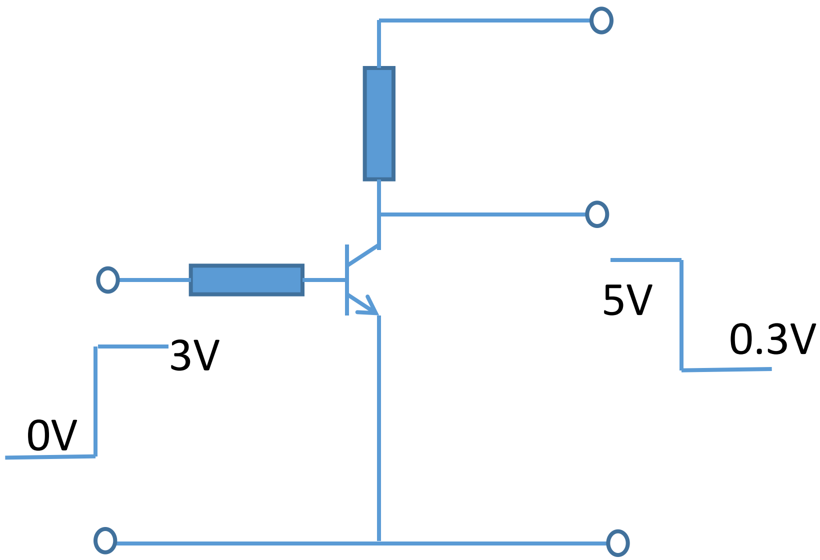
电压(V)	二值逻辑	电　平
+5	1	H(高电平)
0	0	L(低电平)

模拟电路与数字电路

2）晶体管的工作状态不同

在数字电路中，晶体管工作在开关状态

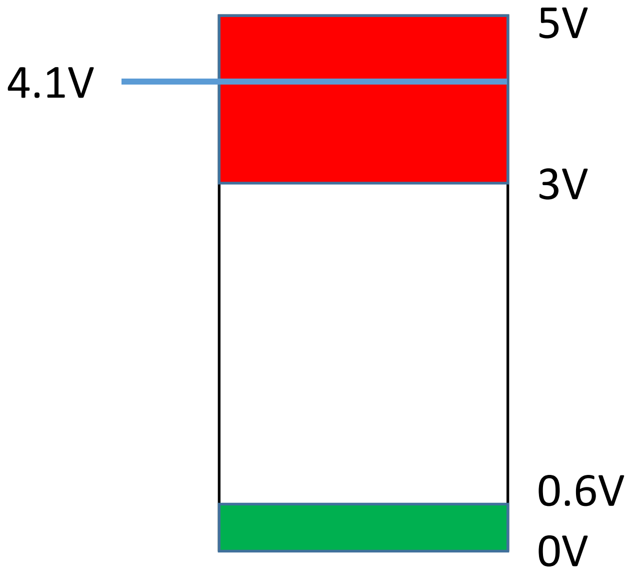




模拟电路与数字电路

3）抗干扰能力不同

数字电路中的高低电平都指的是一定的电压范围，在所受到的干扰不足以改变信号的状态时，不影响电路的正常工作。



模拟电路与数字电路

4）分析方法不同

模拟电路	数字电路
微变等效电路	逻辑分析方法
——电路分析	数学工具： 布尔代数
性能指标	描述方法： 真值表 表达式 功能表等
	功能实现

数字集成电路基础

——数制与编码

张悦
微电子学院
费尔北京研究院 / 自旋电子交叉学科中心

2020-10-2

数制:多位数码中的每一位数的构成及低位向高位进位的规则

1.常用数制

1.1 十进制

（1）计数符号：0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

（2）进位规则：逢十进一.

例：$1983.62 = 1\times10^{3} + 9\times10^{2} + 8\times10^{1} + 3\times10^{0} + 6\times10^{-1} + 2\times10^{-2}$

（3）十进制数按权展开式

$$(N)_{10} = \sum_{i=-m}^{n-1} a_i \times 10^{i}$$

系数　　权

1.2. 二进制

(1) 计数符号：0，1 .

(2) 进位规则：逢二进一.

(3) 二进制数按权展开式

$$(N)_2 = \sum_{i=-m}^{n-1} a_i \times 2^i$$

数字电路中采用二进制的原因：

1）数字装置简单可靠；

2）二进制数运算规则简单；

3）数字电路既可以进行算术运算，也可以进行逻辑运算.

1.3. 十六进制和八进制

（1）十六进制数计数符号：

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

（2）十六进制数进位规则：逢十六进一.

（3）按权展开式：

$$(N)_{16} = \sum_{i=-m}^{n-1} a_i \times 16^i$$

例： $$(6D.4B)_{16} = 6\times16^1 + D\times16^0 + 4\times16^{-1} + B\times16^{-2}$$

$$= 6\times16^1 + 13\times16^0 + 4\times16^{-1} + 11\times16^{-2}$$

数制

（1）八进制数计数符号：0, 1, ...6, 7.

（2）八进制数进位规则：逢八进一.

（3）按权展开式：

$$(N)_8 = \sum_{i=-m}^{n-1} a_i \times 8^i$$

例：

$$(63.45)_8 = 6\times 8^1 + 3\times 8^0 + 4\times 8^{-1} + 5\times 8^{-2}$$

不同进制数的对照表

十进制数	二进制数	八进制数	十六进制数
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	*07*	7
08	1000	*10*	8
09	1001	11	9
10	1010	12	*A*
11	1011	13	*B*
12	1100	14	*C*
13	1101	15	*D*
14	1110	16	*E*
15	1111	*17*	*F*
16	10000	*20*	20
17	10001	21	21
18	10010	22	22
19	10011	23	23
20	10100	24	24

数制转换

- 数值相等，计数方法（数制）不同，
- 本质：权值的转换

- 数制转换 之 任意进制到十进制的转换

利用任意进制数的按权展开式，可以将一个任意进制数转换成等值的十进制数。

例：

$$(1011.101)_2 = 1\times 2^3 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-3}$$

$$= 8 + 2 + 1 + 0.5 + 0.125$$

$$=(11.625)_{10}$$

例： $(8FA.C)_{16} = 8 \times 16^2 + F \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1}$

$= 2048+240+10+0.75=2298.75$

2.1 十进制数转换为二进制数

2.1.1 十进制整数的转换

$$\overbrace{k_{n-1} \times 2^{n-3} + k_{n-2} \times 2^{n-4} + \ldots + k_2 \times 2^0 + \underbrace{k_1/2}_{\text{余数为}k_1}}^{\text{余数为}k_0}$$

依次类推，“**除2取余**”法

例：（173）

2	$(173)_{10}$	………………	余数=1	k_0
2	86	………………	余数=0	k_1
2	43	………………	余数=1	k_2
2	21	………………	余数=1	k_3
2	10	………………	余数=0	k_4
2	5	………………	余数=1	k_5
2	2	………………	余数=0	k_6
2	1	………………	余数=1	k_7
	0			

低位

$$(173)_{10} = (1010\ 1101)_2$$

高位

2.1.2十进制小数的转换

考虑：

$$(D)_{10}=k_{-1}\times 2^{-1}+k_{-2}\times 2^{-2}+\ldots+k_{-(m-1)}\times 2^{-(m-1)}+k_{-m}\times 2^{-m}$$

$$2\times(D)_{10}=\underbrace{k_{-1}}_{\text{整数部分为}k_{-1}}+k_{-2}\times 2^{-1}+\ldots+k_{-(m-1)}\times 2^{-(m-2)}+k_{-m}\times 2^{-(m-1)}$$

依次类推，“**乘2取整**”法

例：（0.6875）

$$\begin{array}{llcr} & & & 0.6875 \\ & & & \times\quad 2 \\ \hline k_{-1} & 1 & \cdots\cdots\cdots & 1.3750 \\ & & & 0.3750 \\ & & & \times\quad 2 \\ \hline k_{-2} & 0 & \cdots\cdots\cdots & 0.7500 \end{array}\qquad\begin{array}{llcr} & & & 0.7500 \\ & & & \times\quad 2 \\ \hline k_{-3} & 1 & \cdots\cdots\cdots & 1.5000 \\ & & & 0.5000 \\ & & & \times\quad 2 \\ \hline k_{-4} & 1 & \cdots\cdots\cdots & 1.0000 \end{array}$$

$$(0.6875)_{10}=(0.1011)_2$$

2.2 二进制与十六进制数间转换

- 数制转换 之 “二－十六” 进制的转换
  - “分组对应”法
    - 由于4位二进制数恰好有16个状态，而将这4位二进制数看作一个整体时，它的进位输出又正好是逢十六进一。

$$(1011101.101001)_2 = (\underline{0101}\ \underline{1101}.\underline{1010}\ \underline{0100})_2$$

$$= (5D.A4)_{16}$$

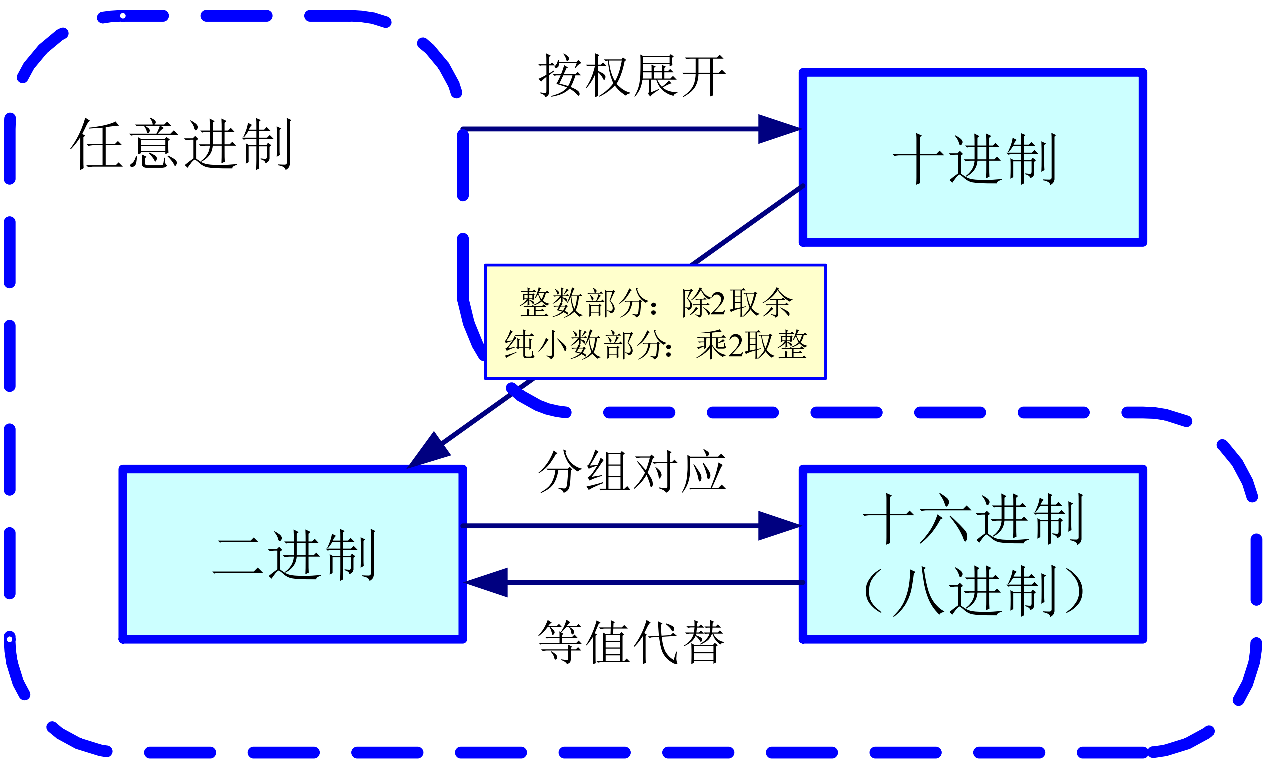
- 数制转换 之 “十六－二” 进制的转换
  - 将十六进制数的每一位用等值的4位二进制数代替

例： $(8\,FA.C6)_{16}$

$$(\ 8\quad F\quad A\ .\ C\quad 6\ )_{16}$$

$$(\ \underline{1000}\ \ \underline{1111}\ \ \underline{1010}\ .\ \underline{1100}\ \ \underline{0110}\ )_{16}$$

2.3 小结：进制之间的转换方法



3.1 二进制正负数的表示及运算

- 二进制四则运算

加

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ +\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 0 \end{array}$$

减

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ -\ 0\ 1\ 0\ 1 \\ \hline 0\ 1\ 0\ 0 \end{array}$$

乘

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ *\ 0\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ \ \ \\ 1\ 0\ 0\ 1\ \ \ \ \ \ \\ 0\ 0\ 0\ 0\ \ \ \ \ \ \ \ \ \\ \hline 0\ 1\ 0\ 1\ 1\ 0\ 1 \end{array}$$

移位与加法

除

$$\begin{array}{r} 1.1\ 1\ \ldots \\ 0\ 1\ 0\ 1 \,\big)\, 1\ 0\ 0\ 1\ \ \ \ \ \ \ \ \\ 0\ 1\ 0\ 1\ \ \ \ \ \ \ \ \\ \hline 1\ 0\ 0\ 0\ \ \ \ \\ 0\ 1\ 0\ 1\ \ \ \ \\ \hline 0\ 1\ 1\ 0 \\ 0\ 1\ 0\ 1 \\ \hline 0\ 0\ 1\ 0 \end{array}$$

移位与减法

二进制正负数的表示

- 二进制的原码、反码及补码（有符号二进制数）

✓正数的三种表示法一样：
符号位为0，随后是二进制的绝对值，即（正数）“原码”。

原码

- 最高位表示正、负号
- 0表示正，1表示负
- 其余各位表示数的绝对值

例：

（设：为8-bit有符号数）

$$[(+43)_{10}]_{原} = 00101011$$

$$[(-43)_{10}]_{原} = 10101011$$

负数的原码、反码和补码表示方法

- 原码 例：$[-25]_{原}=10011001$
- 反码 例：$[-25]_{反}=11100110$
  - 负数的反码是对正数的编码（正数原码）取反；
  - 注：绝对值位域取反，符号位为“1”；也可以认为是对整个码字逐位取反。
- 补码 例：$[-25]_{补}=11100111$
  - 计算方法
    - “反码加1”

确切地说：将相反符号数（这里指“正数原码”）的码字含符号位逐位取反，然后从最低位加1。

- **补码的算例：**

设以8-bit存贮有符号整数，最高位为符号位

1. 求 $[(-39)_{10}]_{补}$，即 以补码表示$(-39)_{10}$
2. 给定补码为$[11101010]_{补}$，求该数，以十进制表示

解 1 ：

绝对值的补（原）码表示为： $(+39)_{10} = 0\ 010\ 0111$

原码取反： $[(-39)_{10}]_{反} = 1\ 101\ 1000$

"反码加1" $[(-39)_{10}]_{补} = 1\ 101\ 1001$

解 2 ： "符号位"为"1"，说明为负数，则：

求相反符号数（负负为正）的补码表示——"反码加1"

答案：$[\ 1110\ 1010\ ]_{补} = (-\ 001\ 0110\ )_2 = (-22)_{10}$

- 注意：补码加减法运算应在相应位数表示的数值范围内进行。

- 补码——加减法运算
  - 负数采用补码表示后，就可以把减法转换为加法

例：39-22=39+(-22)=17

- 注：$(+39)_{10}$　　$(+22)_{10}$　　$[(-22)_{10}]_{补}$

$(0010\ 0111)_2$　　$(0001\ 0110)_2$　　$(1110\ 1010)_2$

原码：

```
  010 0111
- 001 0110
__________
  001 0001 = (+17)10
```

要保证被减数不小于减数；否则调换次序。

判断得到运算结果的符号。

补码：

```
   0010 0111
+  1110 1010
____________
 1 0001 0001
```

自动丢弃

码制：编制代码所要遵循的一定的规则

4.1 二—十进制代码（BCD码）(Binary Coded Decimal codes)

用四位二进制代码来表示一位十进制数码，这样的代码称为二-十进制码，或BCD码.

四位二进制有16种不同的组合，可以在这16种代码中任选10种表示十进制数的10个不同符号，选择方法很多. 选择方法不同，就能得到不同的编码形式.

码制

8421码又称BCD（ Binary Coded Decimal）码，是十进制代码中最常用的二种。在这种编码方式中，每一位二值代码的1都代表一个固定数值，将每一位的1代表的十进制数加起来，得到的结果就是它所代表的十进制数码。由于代码中从左到右每一位的1分别表示8、4、2、1，所以将这种代码称为8421码。每位的1代表的十进制数称为这一位的权。8421码中每一位的权是固定不变的，它属于恒权代码。

5211码每位的权正好与8421码十进制计数器4个触发器输出脉冲的分频比相对应。这种对应关系在构成某些数字系统时很有用。

2421码是一种恒权代码，它的0和9、1和8、2和7、3和6、4和5也互为反码，这个特点和余3码相仿。

余3码的编码规则与8421码不同，如果把每一个余3码看作4位二进制数，则它的数值要比它所表示的十进制数码多3，故而将这种代码称为余3码。如果将两个余3码相加，所得的和将比十进制数和所对应的二进制数多6。因此，在用余3码做十进制加法运算时，若两数之和为10，正好等于二进制数的16，于是便从高位自动产生进位信号。此外，从表1.5.1中还可以看出，0和9、1和8、2和7、3和6、4和5的余3码互为反码，这对于求取对10的补码是很方便的。余3码不是恒权代码。如果试图将每个代码视为二进制数，并使它等效的十进制数与所表示的代码相等，那么代码中每一位的1所代表的十进制数在各个代码中不能是固定的。

余3循环码是一种变权码，每一位的1在不同代码中并不代表固定的数值。它的主要特点是相邻的两个代码之间仅有一位的状态不同。

码制

十进制	8421码	5211码	2421码	余3码	余3循环码
0	0000	0000	0000	0011	0010
1	0001	0001	0001	0100	0110
2	0010	0100	0010	0101	0111
3	0011	0101	0011	0110	0101
4	0100	0111	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1100	1101	1010	1111
8	1000	1101	1110	1011	1110
9	1001	1111	1111	1100	1010
位权	8421	5211	2421	无权	无权

（1）**有权BCD码**：每位数码都有确定的位权的码，

例如：8421码、5211码、2421码.

如：5211码1011代表5+0+1+1=7;

2421码1100代表2+4+0+0=6.

* 5211BCD码和2421BCD码不唯一.

例：2421BCD码0110也可表示6

* 在表中：

① 8421BCD码和代表0~9的二进制数一一对应；

② 2421BCD码的前5个码和8421BCD码相同，后5个码以中心对称取反，这样的码称为自反代码.

例：　4→0100　5→1011

　　　0→0000　9→1111

（2）无权BCD码：每位数码无确定的位权，例如：余3码.

余3码的编码规律为：在8421BCD码上加0011，

例 6的余3码为： 0110+0011=1001

余3码也是自反代码

4.2 格雷码(Gray码)

格雷码为无权码, 特点为：相邻两个代码之间仅有一位不同, 其余各位均相同. 具有这种特点的代码称为循环码, 格雷码是循环码.

格雷码（ Gray Code）又称循环码。从表中5位格雷码编码表中可以看出格雷码的构成方法，这就是每一位的状态变化都按一定的顺序循环。如果从00000开始，最右边一位的状态按0110顺序循环变化，右边第二位的状态按00111100顺序循环变化，右边第三位按0000111111110000序循环变化。可见，自右向左，每一位状态循环中连续的0、1数目增加一倍。

- 循环码（格雷码）的特性
  1. 单位距离特性
  2. 循环相邻特性
  3. 镜像反射特性

十进制数	b_4	b_3	b_2	b_1	b_0	十进制数	b_4	b_3	b_2	b_1	b_0
0	0	0	0	0	0	16	1	1	0	0	0
1	0	0	0	0	1	17	1	1	0	0	1
2	0	0	0	1	1	18	1	1	0	1	1
3	0	0	0	1	0	19	1	1	0	1	0
4	0	0	1	1	0	20	1	1	1	1	0
5	0	0	1	1	1	21	1	1	1	1	1
6	0	0	1	0	1	22	1	1	1	0	1
7	0	0	1	0	0	23	1	1	1	0	0
8	0	1	1	0	0	24	1	0	1	0	0
9	0	1	1	0	1	25	1	0	1	0	1
10	0	1	1	1	1	26	1	0	1	1	1
11	0	1	1	1	0	27	1	0	1	1	0
12	0	1	0	1	0	28	1	0	0	1	0
13	0	1	0	1	1	29	1	0	0	1	1
14	0	1	0	0	1	30	1	0	0	0	1
15	0	1	0	0	0	31	1	0	0	0	0

作业

第五版

- 1.2-(4)；1.4-(2)；
  1.5-(4)；1.6-(2,3)
  1.7-(1,4)；1.10-(2,4)；
  1.12-(3.5)

第四版

- 1.1-(2,4)；1.2-(2,3)；
  1.3-(1,4)；1.4-(2,4)