

《编码：隐匿在计算机软硬件背后的语言》

Code: The Hidden Language of Computer Hardware and Software

读书笔记

声明

该文章是阅读《编码：隐匿在计算机软硬件背后的语言》一书之后整理出的读书笔记。若有错误，还需继续修正与增删。

Preface

作者Charles Petzold是Windows编程界的大师，当今世界顶级技术作家。
他是WindowsGDI程序设计首席技术作家，其大作Programming Windows（Windows程序设计）是尽人皆知的Windows编程经典。

What? 本书讲了什么

这是一本讲述**计算机工作原理**的书。读者可以通过本书获得对计算机工作原理较深刻的理解这种理解的深度不逊于“*电气工程师”与“程序员”*的理解。

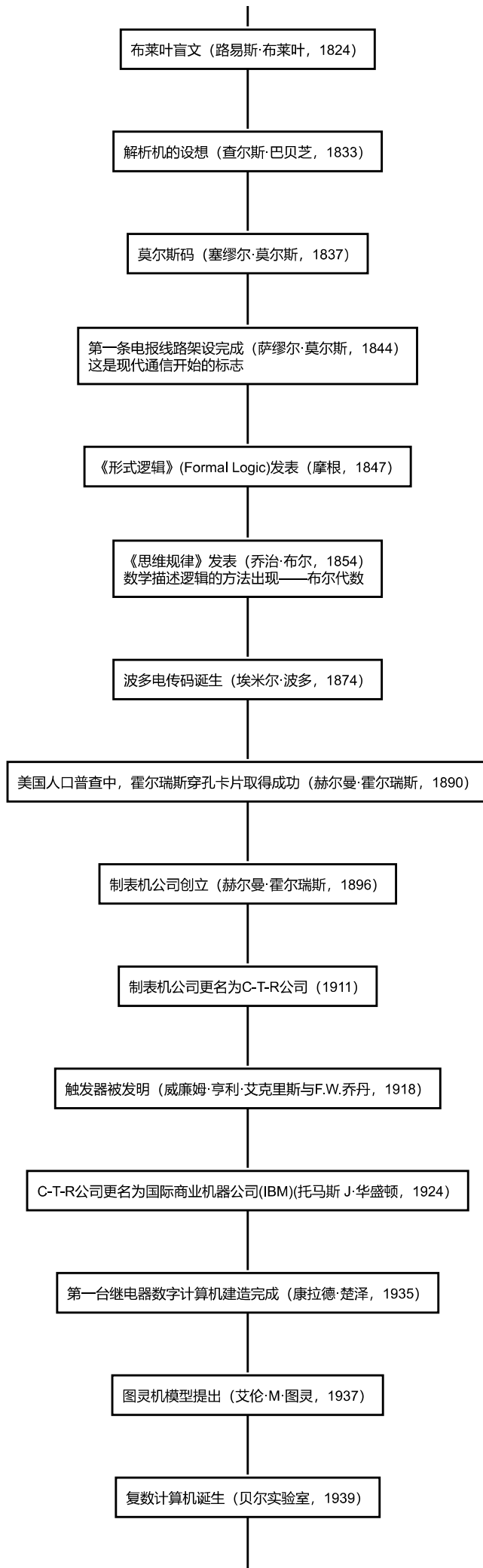
How? 本书是怎样讲的

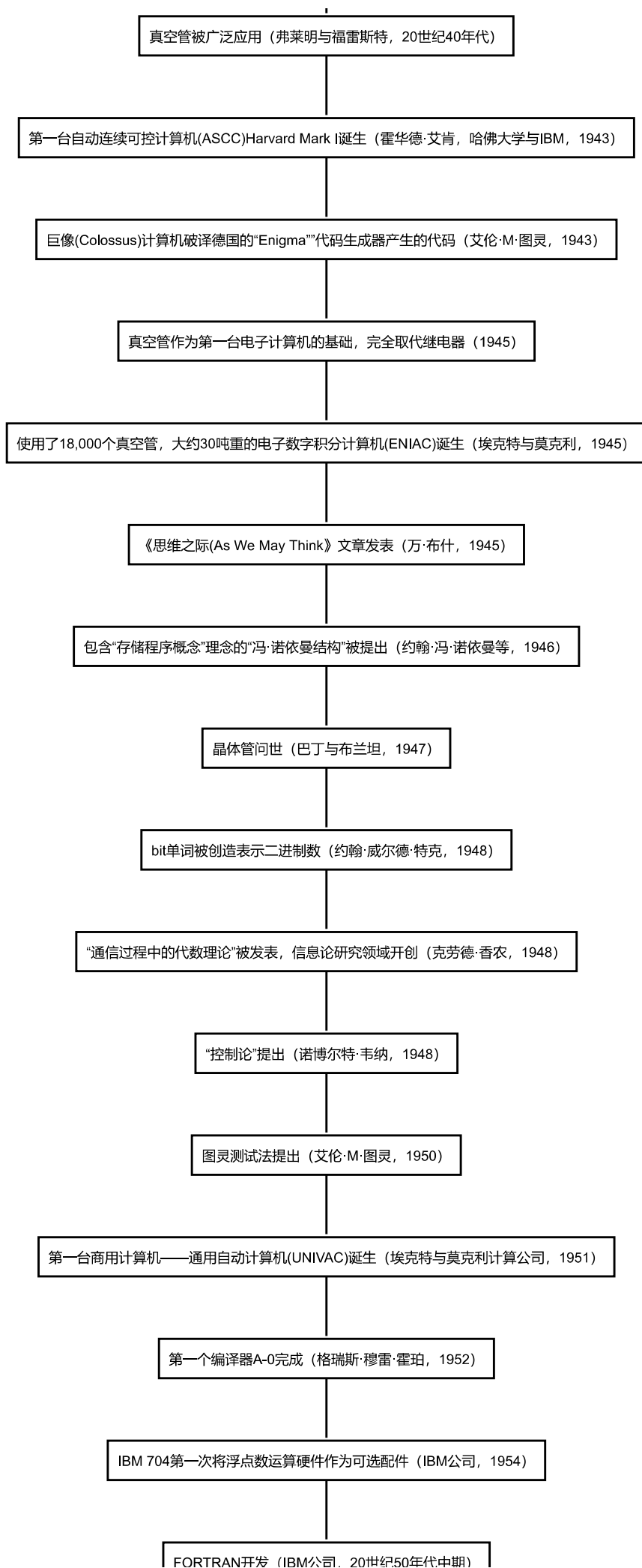
一系列被有机结合起来的零星知识，所涉及的问题包括计算机体系结构、汇编语言及数字电子技术等。

- **历史角度**：审视了计算机技术的发展脉络
- **自底向上**：导线、灯泡以及触发器等简单部件通过精妙地组装，可以构建出一台电子计算机
- **旁征博引**：从美国当红影评人到百年前的布莱叶盲文，从史密斯·索尼娅的滑尺到冯·诺依曼的构想，从石破天惊的论文《思维之实际》到图形化革命

When? 计算机发展时间线

差分机的设想（查尔斯·巴贝芝，1820）





计算机的发明 (1946年)

“磁芯存储器”被开发出 (20世纪50年代中期)

字节(byte)被创造 (IBM公司, 1956)

肖克利半导体实验室在加利福尼亚帕罗奥图市成立, 当今硅谷所在地区 (威廉·肖克利, 1956)

首款磁盘驱动器RAMAC发明 (IBM公司, 1956)

集成电路被发明 (杰克·基尔比与罗伯特·诺伊斯, 1958-1959)

COBOL面世 (1959)

ALGOL改进版面世 (1960)

PL/I面世 (IBM公司, 20世纪60年代中期)

BASIC面世 (约翰·克莫尼与托马斯·克鲁兹, 1964)

摩尔定律提出 (戈登·E·摩尔, 1965)

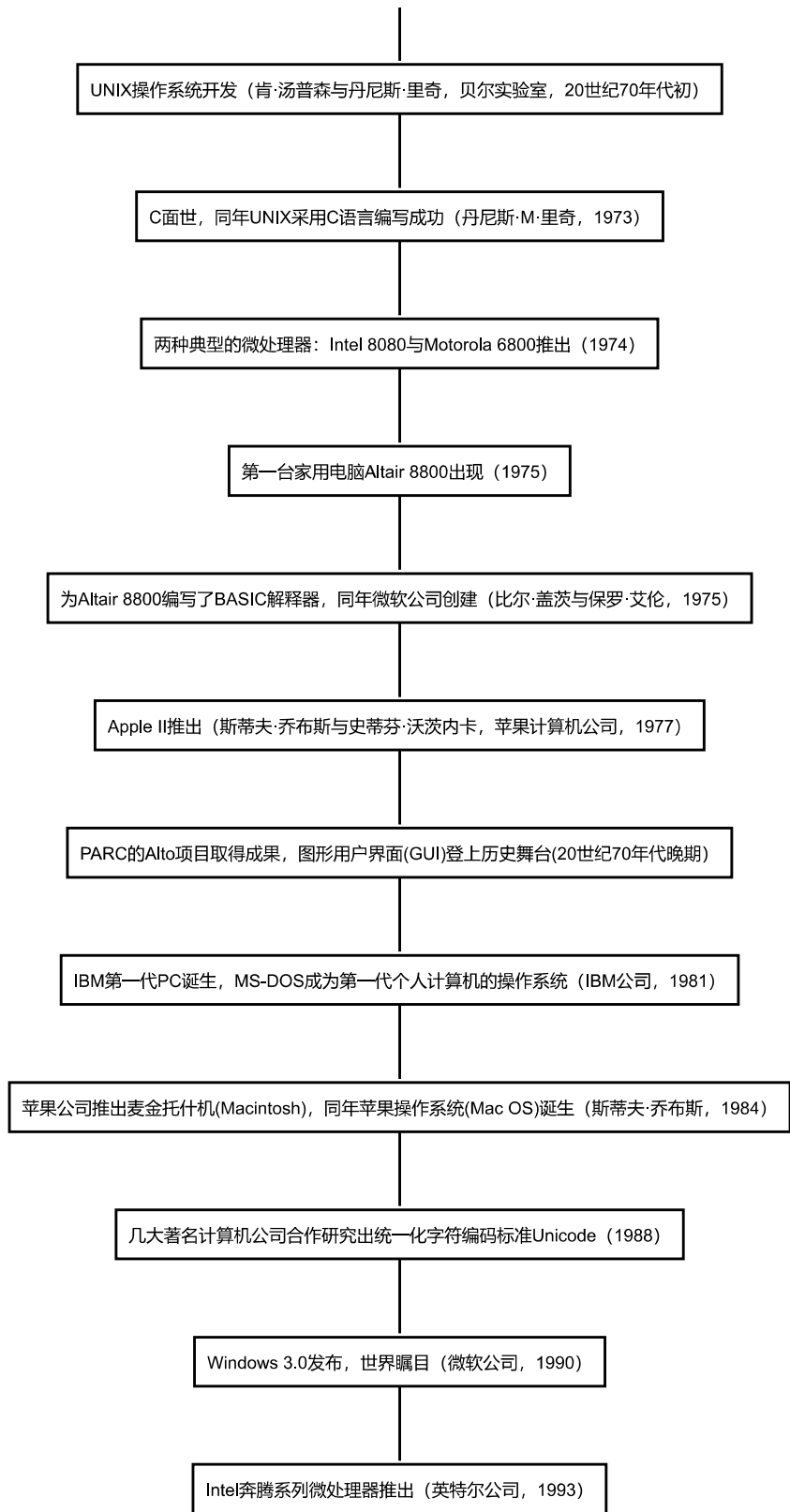
ASCII码正式公布 (1967)

Pascal面世 (尼尔莱斯·沃思, 20世纪60年代末)

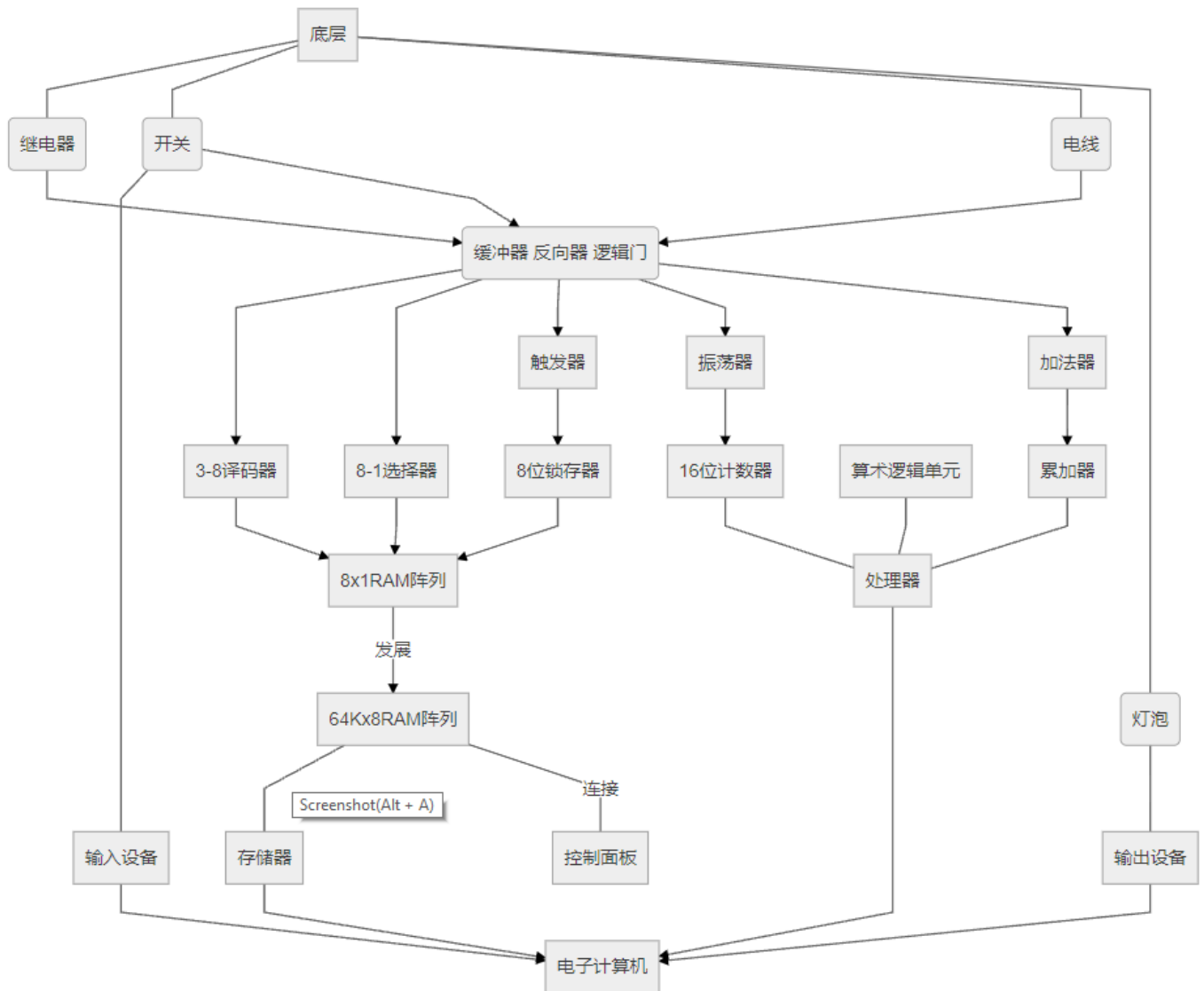
使用集成电路在一块电路板上制造一个完整的计算机处理器变得可能 (20世纪70年代早期)

第一块芯片诞生 (英特尔公司, 1970)

第一块“计算机芯片 (微处理器)”Intel 4004诞生 (英特尔公司, 1971)



How? 自底向上的计算机结构



文章目录

- ○ 声明
- Preface
- ■ What? 本书讲了什么
 - How? 本书是怎样讲的
 - When? 计算机发展时间线
 - How? 自底向上的计算机结构
- @[toc]
- Body
- ■ 1. 至亲密友
 - ■ What? 各类编码
 - AD&DAD: 莫尔斯编码
- 2. 编码与组合
- 3. 布莱叶盲文与二进制码
 - ■ DIF: 瓦伦丁·霍伊与路易斯·布莱叶
 - AD: 布莱叶盲文
- 4. 手电筒的剖析
 - ■ What? 电学知识

- 5. 绕过拐角的通信
 - ■ How? 铺设长电路时处理大电阻
 - PRO: 由于电阻的存在电路距离跨度不能实现无限 (如电报系统)
- 6. 电报与继电器
 - ■ How? 电报机(telegraph)的相关问题
- 7. 我们的十个数字
 - ■ AD&DAD: 罗马数字系统 (古希腊人并不以代数著称)
 - AD: 印度-阿拉伯数字系统 (位置计数系统)
- 8. 十的替代品
- 9. 二进制数
- 10. 逻辑与开关
 - ■ DIF: 布尔代数与传统代数
 - What? 布尔测试
 - What? 重要的概念突破——将布尔代数与电路相融合
- 11. 门
 - ■ What? 继电器的连接——逻辑门的关键
 - What? 摩根定律
- 12. 二进制加法器
 - ■ What? 逻辑门与二进制加法的相关性
- 13. 如何实现减法
 - ■ How? 如何解决减法中的借位问题
 - How? 求2的补数
- 14. 反馈与触发器
 - ■ What? 振荡器
 - What? 反馈(feedback)
- 15. 字节与十六进制
- 16. 存储器组织
 - ■ How? 用一个灯泡确定锁存器的数据输出信号
 - How? 用3个选择输入端输入8个数据
 - AD: RAM相比顺序型存储器
 - EXP: 8x1RAM阵列的扩充
 - What? 字节的单位换算
 - PRO: 当前存储器存在的隐患
- 17. 自动操作
 - ■ EXP: 自动加法器除了可以累加一组数字, 还可以自主确定累加多少数字, 并记住RAM中计算结果的个数
 - PRO: 代码与数据存储器同步、顺序
 - Solution: 根本性的修改
 - DIF: 计算机(computer)与计算器(calculator)
 - What? 处理器的若干组件
 - What? 硬件与软件
 - Who? 程序设计人员=编码员(coders)=软件工程师(software engineers)

- What? 汇编语言(assembly language)
- 18. 从算盘到芯片
 - ■ How? 计算工具的发展史
 - How? 比较微处理器性能的三个衡量标准
- 19. 两种典型的微处理器
 - ■ What? 计算机的指令集 (Intel 8080)
 - What? 标志位(flag)
 - CMP: Motorola 6800相比于Intel 8080
 - Why? 根据摩尔定律增加的大量晶体管
- 20. ASCII码和字符转换
- 21. 总线
 - ■ DIF: 随机访问存储器与磁介质存储器
- 22. 操作系统
 - ■ What? 内存空间
 - What? 控制台可识别命令
- 23. 定点数和浮点数
 - ■ What? 定点数与浮点数
- 24. 高级语言与低级语言
- 25. 图形化革命
 - ■ What? 计算机图形的两个分支
 - What? 数据压缩(Data Compression)
 - AD: 数字化声音相比于模拟声音
 - How? 实现计算机与人对话
 - Why? Web站点不为计算机提供一个可执行程序
- Additions
- ■ 计算机技术发展史上的杰出人物简列
- 计算机技术发展史上的著名组织与公司简列

Body

1. 至亲密友

编码是一种用来在机器和人之间传递信息的方式。编码就是交流。

What?各类编码

- 莫尔斯电码(Morse Code)
- 口头话语(the spoken word)或言辞(speech)
- 书面语言(the written word)或文本(text)
- 手语
- 布莱叶盲文(Braille)

AD&DAD: 莫尔斯编码

- DAD:
 1. 莫尔斯编码没有区分大写字母和小写字母。
- AD:
 1. 证明了两个不同的事物只要经过适当的组合，即可表示所有类型的信息。

2. 编码与组合

莫尔斯码又称二进制码(Binary Code)，它的组成元素只有两个——“点”和“划”。点、划的组合可以表示你想要的任意数目的码字。

码字的数目 = 2 “点”和“划”的数目 = 2 编码的位数 码字的数目 = $2^{\{ \text{“点”和“划”的数目} \}} = 2^{\{ \text{编码的位数} \}}$ 码字的数目 = $2^{\{ \text{“点”和“划”的数目} \}} = 2^{\text{编码的位数}}$

3. 布莱叶盲文与二进制码

布莱叶盲文的工作方式 ——> 编码的一些性质

DIF: 瓦伦丁·霍伊与路易斯·布莱叶

- 视力健全的人被自身的感知模式所禁锢。（类似于手电筒交流在空中比划字母的形状）
- 完全不同于印刷字母的文字系统更适合盲人阅读。

AD: 布莱叶盲文

1. 缩写: 二级布莱叶盲文
2. **优先码(precedence codes)或换档码(shift codes)**: 数字标识符与字母标识符（作用域内，直到作用域结束）
3. **逃逸码**: 大写标识符（可区分大小写，紧随其后的字母）

4. 手电筒的剖析

电在计算机中的工作原理——电学

What? 电学知识

- 串联与并联
- 导电性与阻抗性，导体与绝缘体
- 电阻（阻抗）的影响因素
- 电压、电流、电阻和功率
- 开关（**很重要的部分**）

二进制码与电气电路之间的相似性将起到很大作用。

5. 绕过拐角的通信

电路

How? 铺设长电路时处理大电阻

- 使用粗导线（昂贵）
- 增加电压，同时使用更大电阻的灯泡

PRO: 由于电阻的存在电路距离跨度不能实现无限（如电报系统）

Solution: 继电器

6. 电报与继电器

今天的全球性即时通信融合了两点：即时通信与远距离通信。在19世纪早期，为了实现以上两点的融合，电报以及继电器应运而生。

How? 电报机(telegraph)的相关问题

- 在线路的这一段采取一些措施，使线路的另一端发生某种变化
- 电磁现象（电磁铁）
- 长导线所带来的电阻：线路不能无限延长——>中继系统
- **继电器**: 输入弱电流，输出强电流

7. 我们的十个数字

十进制

- Digit: 数字, 手指、脚趾
- five与fist (拳头) 具有相同的词根

AD&DAD: 罗马数字系统 (古希腊人并不以代数著称)

- DAD:
 1. 不易于乘除
- AD:
 1. 易于加减

AD: 印度-阿拉伯数字系统 (位置计数系统)

1. 位置相关
2. 没有10的符号
3. 有0

8. 十的替代品

其它进制, 八进制、四进制、二进制。

9. 二进制数

二进制中的一位(bit)称为1比特, 我们可用它表达简单的信息: 是或不是, 亮或灭, 打开或关闭。而事实上只要信息能转换成两种或多种可能性的选择, 都可以用比特来表示 (胶卷的胶片速度、条形码)。

10. 逻辑与开关

布尔代数: 一种数学形式来描述逻辑

DIF: 布尔代数与传统代数

- 布尔代数中, 加法也可以分配乘法
- **最大区别:**
$$F * F = F \quad F * F = F \quad F * F = F$$
$$F + F = F \quad F + F = F \quad F + F = F$$

What? 布尔测试

AND01

0 00
1 01

OR01

0 01
1 11

NAND01

0 11
1 10

NOR01

0 10
1 00

XOR01

0 01

XOR01

1 10

What? 重要的概念突破——将布尔代数与电路相融合

串联 = A AND 串联=AND 串联 = AND

并联 = O R 并联=OR 并联 = OR

- 乔治·布尔没有机会观察布尔表达式在开关线路与如何实现（布尔去世后15年人类才发明白炽灯）。
- 在19世纪，没有人将布尔代数中的AND与OR同线路中的串联及并联关联到一起。

11. 门

逻辑门(logic gate): 继电器的组合，用电路去模拟基本逻辑任务

What? 继电器的连接——逻辑门的关键

- 四种基本逻辑门
 - 与门 AND
 - 或门 OR
 - 或非门 NOR
 - 与非门 NAND
 - 异或门 XOR
- 复杂电路
 - **2-4译码器**
- 反向器(inverter)
- 缓冲器(buffer) **没什么作用**

What? 摩根定律

- 两组等价关系
- 简化布尔表达式的重要手段（**简化电路**）

12. 二进制加法器

搭建计算两个数加和的器件是搭建一台计算机的基础。

What? 逻辑门与二进制加法的相关性

+加法01

0 01

1 10

+进位01

0 00

1 01

- 进位位与与门的输出结果一致
- 加法位与异或门的输出结果一致

半加器(Half Adder)（没有将之前一次加法可能产生的进位位纳入下一次运算）

全加器(Full Adder)（包括进位输入）

13. 如何实现减法

加法和减法在某些方面互相补充，但是在机制上二者却存在本质区别 ——> 把减法变成加法

How? 如何解决减法中的借位问题

利用**补数**，我们将不会再用减法。（以1开头的每个二进制数都表示一个负数）

How? 求2的补数

- 将每位取反再加1
- 相同方法可还原数值

有符号数与无符号数

14. 反馈与触发器

记忆力：一个能计数的电路必定需要触发器。

What? 振荡器

- 不需要人为干涉，自发工作
- 时钟(clock)
- 循环(cycle)、周期(period)、频率(frequency)

What? 反馈(feedback)

- 当两个开关都断开时，两个稳定态——**触发器(Flip-Flop)**可以保持信息
- 一个能计数的电路必定需要触发器——**计数器**
- R-S触发器
- 电平触发与边沿触发时钟

15. 字节与十六进制

8位的数据流，即数据路径的位宽为8。

16. 存储器组织

存储器：记录信息

- **D型电平触发器**——>**锁存器**：具备对1位信息存储的能力

How? 用一个灯泡确定锁存器的数据输出信号

- **8-1选择器**

How? 用3个选择输入端输入8个数据

- **3-8译码器**
- 读/写存储器(read/write memory)或**随机访问存储器(Random Access Memory, RAM)**
 - 地址端口(Address)
 - 3-8译码器（输入端）
 - 8位锁存器
 - 8-1选择器（输出端）

AD：RAM相比顺序型存储器

- 读写自由

EXP：8x1RAM阵列的扩充

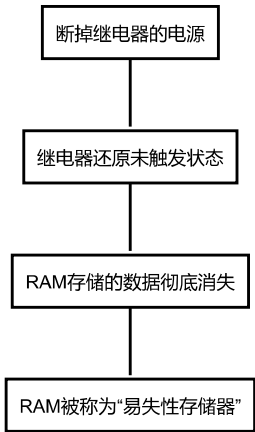
- 增加1个数据输入：8x2RAM阵列
- 增加1个地址端口：16x1RAM阵列
- RAM 阵列 的 存 储 容 量 = 2 地 址 端 口 的 数 量 RAM阵列的存储容量= $2^{\text{地址端口的数量}}$ RAM阵列的存储容量 = 2地址端口的数量
- 10个地址端口，8个Data In和Data Out：1024x8RAM阵列
- **控制面板**

What? 字节的单位换算

1 安字节 = 2 10 批字节 = 2 20 太字节 (T) = 2 30 吉字节 (G) = 2 40 兆字节 (M) = 2 50 千字节 (K) = 2 60 字节
1安字节=2^{10}批字节=2^{20}太字节(T)=2^{30}吉字节(G)=2^{40}兆字节(M)=2^{50}千字节(K)=2^{60}字节
1安字节 = 210批字节 = 220太字节(T) = 230吉字节(G) = 240兆字节(M) = 250千字节(K) = 260字节
1 安字节 ≈ 1 0 3 批字节 ≈ 1 0 6 太字节 (T) ≈ 1 0 9 吉字节 (G) ≈ 1 0 12 兆字节 (M) ≈ 1 0 15 千字节 (K) ≈ 1 0 18 字节
1安字节 ≈ 10^3批字节 ≈ 10^6太字节(T) ≈ 10^9吉字节(G) ≈ 10^12兆字节(M) ≈ 10^15千字节(K) ≈ 10^18字节

- 讨论存储器时，通常使用字节数；
- 描述在线路中流动的数据时，使用比特（千比特每秒kbps与兆比特每秒mbps）

PRO：当前存储器存在的隐患

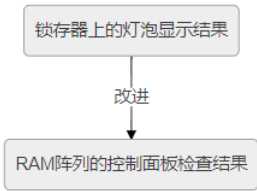


- 易失性存储器**需要恒定的电流。

17. 自动操作

怎样让加法器自动地完成数据输入和计算？
编写程序。

- 累加器



EXP：自动加法器除了可以累加一组数字，还可以自主确定累加多少数字，并记住RAM中计算结果的个数

- 指令码或操作码(operation code,opcode)，即**机器码(machine codes)**
 - Load（加载）
 - Store（保存）
 - Add（加法）
 - Subtract（减法）
 - Add with Carry（进位加法）
 - Subtract with Borrow（借位减法）
 - Halt（停止）
 - Jump（跳转）
 - Conditional Jump（**条件跳转**）
 - Jump If Zero（零转移）
 - Jump If Carry（进位转移）
 - Jump If Not Zero（非零转移）

- Jump If Not Carry (无进位转移)
- Call (记录了“从何处跳转”)
- Return (返回到跳转点)
- 数据与代码分别写入RAM
 - “数据”存储器
 - “代码”存储器

PRO: 代码与数据存储器同步、顺序

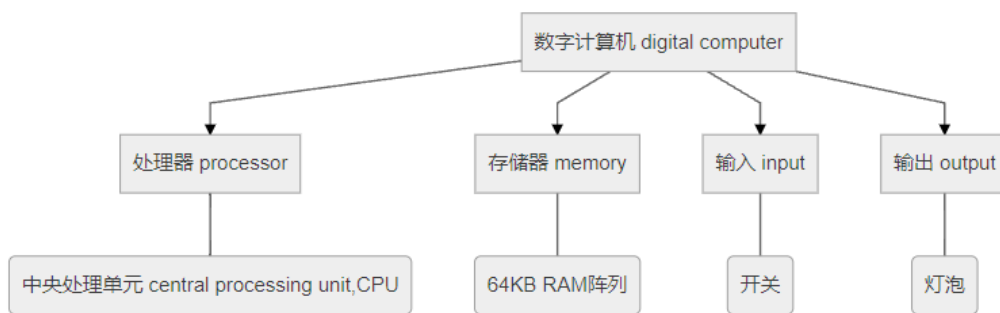
- 保存这些数的存储单元的地址不连续
- 不允许使用前面的计算结果

Solution: 根本性的修改

- 每一个指令在存储器中仅占据3个字节的空间(除了Halt)，其中1个字节为代码本身，另外两个字节用来存放1个16位地址
- 设计关键: 把代码RAM阵列的数据输出到3个8位锁存器
 - 指令代码
 - 地址高字节
 - 地址低字节
- 牺牲了运算速度 (TANSTAFEL工程准则)
- 放弃两个独立的RAM阵列: 操作码和操作数存放在同一个RAM阵列

DIF: 计算机(computer)与计算器(calculator)

- 能否控制重复操作或循环(looping)



What? 处理器的若干组件

- 累加器: 简单锁存器, 保存内部数据
- 算术逻辑单元(ALU): 算术运算与逻辑运算
- 程序计数器(Program Counter, PC)

What? 硬件与软件

- 硬件(hardware)
 - 继电器
 - 电线
 - 开关
 - 灯泡
- 软件(software)
 - 输入到存储器中的指令
 - 输入到存储器中的数值

“软件” = “计算机程序 (computer program)” “软件” = “计算机程序 (computer program)” “软件” = “计算机程序 (computer program)”

Who? 程序设计人员=编码员(coders)=软件工程师(software engineers)

What? 汇编语言(assembly language)

- 全数字的机器语言和指令的文字描述的一种结合体
- 标号表示存储器地址
- 手工汇编, 纸上操作
- 错误(bug)

机器所需要的仅仅是能够做加减法的硬件以及利用条件跳转指令执行代码的方法。

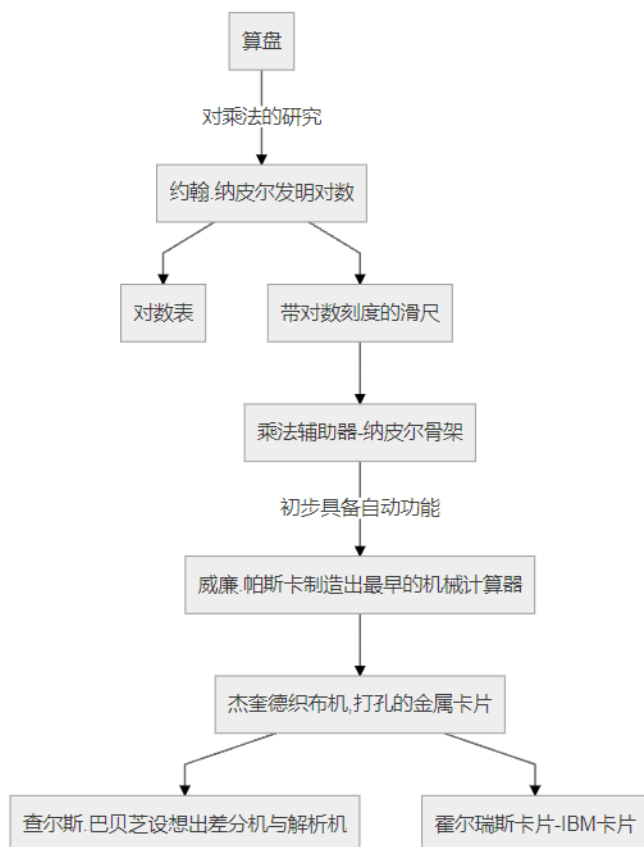
本书中不讲有关**算法(algorithm)**的内容。



18. 从算盘到芯片

计算工具的发展史

How? 计算工具的发展史



How? 比较微处理器性能的三个衡量标准

- 数据通路宽度
- 最大时钟频率
- 寻址能力

以上三个数字指标并不影响一台计算机的计算能力，关键点在于**处理器的速度**。

19. 两种典型的微处理器

将中央处理器的所有构件封装到一块硅芯片上，即微处理器。
两种典型微处理器：Intel 8080和Motorola 6800

What? 计算机的指令集 (Intel 8080)

助记符指令含义

LOD	加载
STO	保存
STA	保存到累加器
LDA	加载到累加器
STAX	保存到BC或DE寄存器
LDAX	加载到BC或DE寄存器
MOV	把一个寄存器中的内容转移到另一个寄存器（可以是原来的寄存器、累加器）
MVI	传送立即数
HLT	停止
ADD	加法（累加器）
ADC	进位加法（累加器）
SUB	减法（累加器）
SBB	借位减法（累加器）
AND	与
OR	或
XOR	异或
CMP	比较，可设置标志位
ADI	立即数加法
ACI	立即数进位加法
SUI	立即数减法
SBI	立即数借位减法
ANI	立即数与
XRI	立即数异或
ORI	立即数或
CPI	立即数比较
STC	令CF置1
CMC	令CF取反
CMA	累加器中的数按位取反
DAA	十进制调整累加器，使用BCD码进行十进制的算术运算
INR	寄存器或存储器中的数加1
DCR	寄存器或存储器中的数减1
RLC	使累加器循环左移
RRC	使累加器循环右移
RAL	带进位的累加器循环左移
RAR	带进位的累加器循环右移
PUSH	压入堆栈
POP	弹出堆栈
LXI	对寄存器对和堆栈指针加载扩展的立即数（两个字节）
INX	对寄存器对和堆栈指针加1
DCX	对寄存器对和堆栈指针减1
DAD	把由任意2个寄存器组成的16位寄存器对的内容加到寄存器对HL中
SHLD	直接保存HL中的数据
LHLD	直接加载数据到HL
PCHL	将HL保存的数据加载到程序计数器
SPHL	将HL保存的数据加载到堆栈指针
XTHL	把HL中的内容和堆栈顶部2个字节进行交换

助记符指令含义

XCHG	把DE中的内容和HL中的内容进行交换
JMP	转移
JZ	零转移
JNZ	非零转移
JC	进位转移
JNC	无进位转移
CALL	调用
RET	返回跳转点
OUT	把累加器中的内容写入到紧跟该指令后的字节所寻址的端口
IN	把一个字节从端口读入到累加器
DI	禁止中断
EI	允许中断
RST	重新启动，立即跳转到特定的地址
NOP	无操作

What? 标志位(flag)

- CF-进位标志位
- ZF-零标志位
- SF-符号标志位
- PF-奇偶标志位
- AF-辅助进位标志位

实现人机交互：输入/输出设备(I/O)

CMP：Motorola 6800相比于Intel 8080

1. 时钟信号较简单
2. 没有独立I/O端口
3. B也是累加器，无其他8位寄存器
4. 16位索引寄存器（类似HL寄存器）
5. 对应操作码和助记符完全不同
6. 无奇偶标志位，有溢出标志位
7. 汇编语言不兼容
8. 6800高前低后，8080低前高后

Why? 根据摩尔定律增加的大量晶体管

1. 增加数据宽度
2. 应对新的指令（浮点数、重复计算、图片与电影）
3. 使用多种现代技术提高运行速度

20. ASCII码和字符转换

如何用比特来存储文本呢？
ASCII码。

- 字符编码集
- 文本字符串(string)
- 1874年波多电传码（默里编码）应用于电传打字机
 - DAD:
 1. 转义码易导致出错
 2. 不能唯一表示
 3. 不能区分大小写
- 回车与换行

- 美国信息交换标准码-ASCII码
- Unicode（统一化字符编码标准）
 - PRO：

改变了字符与存储空间之间“单字符单字节”的等价对应关系（在存储空间上付出代价）

21. 总线

总线(bus)就是数字信号的集合，而这些新好呗提供给计算机上的每块电路板。

计算机中各部件按照功能被分别安装在两个或更多的电路板上

电路板之间通过总线(bus)通信

- 四类信号：
 - 地址信号
 - 数据输出信号
 - 数据输入信号
 - 控制信号
- 开放体系结构(Open Architecture)原则
- IBM PC
- 三态输出
- 电子射线管(CRT)-视频显示器/监视器
- 字符生成器，只读存储器(read-only memory,ROM)

颜色数量 = 2 每个像素赋予的比特数 颜色数量=2{每个像素赋予的比特数} 颜色数量 = 2每个像素赋予的比特数
- **磁介质存储器**逐渐发展成目前最为流行的长期存储器。
 - 录音电话
 - 卷轴式磁带（顺序访问）
 - 磁盘（快速访问）
 - 软盘
 - 硬盘

DIF：随机访问存储器与磁介质存储器

- memory与storage
- 主要区别：存储的信息是否易失
- **显著区别**：微处理器发出地址信号通常寻址随机访问存储器
- 形象比喻：办公桌上的桌面与文件柜

22. 操作系统

Microprocessor + RAM + Keyboard + V D + DiskDriver

MISSING:软件

键盘处理程序及命令处理程序简化了工作，淘汰了控制面板。同时命令处理程序使得计算机变得可交互(interactive)。

- 只读存储器(ROM、PROM、EPROM)

文件系统(file system)可以记录存储位置。

What? 内存空间

0000h 系统参数

0000h 系统参数

0100h 临时程序区域(TPA)

控制台命令处理程序(CCP)

基本磁盘操作系统(BDOS)

最高地址 基本输入/输出系统(BIOS)

- CP/M OS三组分：
 - CCP
 - BDOS
 - BIOS

What? 控制台可识别命令

命令 内容

DIR 显示磁盘的目录信息

ERA 删除磁盘中的文件

REN 改变文件名

TYPE显示文本文件的内容

SAVE 保存文件

- 不能被CP/M识别的命令被默认认为是保存在磁盘上的一个**程序名**。

OS提供应用程序接口(API)，让程序能够方便地访问计算机的硬件。API是一个与设备无关(device-independent)的接口，它屏蔽了硬件之间的差异。

- 1981 IBM PC —— MS-DOS
 - 文件系统采用FAT形式
- 20世纪70年代初 UNIX系统
 - 可移植性(portable)
 - UNIX思想
 - 时分复用技术
 - Linux系统

23. 定点数和浮点数

2.75用二进制怎样表示呢？

- BCD码

What? 定点数与浮点数

- 定点格式数
 - 钱款、账户
 - 浮点格式数
 - 极大数或极小数
 - 科学计数法
 - 符号位s
 - 指数位e
 - 有效数位f
- $$(-1)^s * 1.f * 2^{e-127} \quad (-1)^s * 1.f * 2^{e-127} \quad (-1)^s * 1.f * 2^{e-127}$$
- 单精度(float)与双精度(double)

浮点数运算可以处理一些较为繁杂的函数运算，如平方根、指数、对数和三角函数。

24. 高级语言与低级语言

使用机器码（低级语言）编写程序如同用牙签吃东西，费力又费时；人们想出了一种效率更高的编程方法——高级语言。

- 低级语言：汇编语言
 - 与硬件关系紧密
 - 比机器语言要简单，但微处理器不能解释汇编语言
 - 希望最好由计算机独自完成语言转换的工作——汇编器(assembly)
 - DAD:
 - 非常乏味，微处理器芯片级的编程，微小的细节
 - 不可“移植”
- 高级语言
 - 需要编写一个编译器（复杂）
 - AD:
 - 易用性
 - 可移植性
 - DAD:
 - 没有让处理器变强大
 - 不能使用某些处理器的特有功能
- 第一个编译器——A-0
- 最古老的高级语言——FORTRAN
- ALGOL
程序设计：科学or艺术？ 程序设计：科学or艺术？ 程序设计：科学or艺术？
- 第一个商务系统所使用的程序设计语言——COBOL
 - 支持读取记录(record)和生成报表(report)
- IBM PL/I
 - 融合性
 - ALGOL的块结构
 - FORTRAN的数学函数功能
 - COBOL处理记录和报表的能力
 - 使用不广泛
- BASIC
- Pascal
 - 继承性
 - ALGOL的大部分结构
 - COBOL的记录处理功能
- Ada
- C
 - 支持移位操作与按位布尔运算操作
 - 指针(pointer)
 - “高级汇编语言”

所有的类ALGOL语言设计模式都是基于冯·诺依曼计算机体系的。此类语言进行改进后产生了主要应用在图形化操作系统中的面向对象程序设计语言。

25. 图形化革命

要充分利用计算机日益增长的运算和处理能力，就必须不断改进计算机系统中的用户接口(User Interface)，它是人机交互的轴心。图形化革命

- 《思维之际》与麦克斯存储器(Memex)
- 用户界面(User Interface)
- 从字符显示到图形显示
- 软件之所以被设计出来，其最终目的是——《为了扩展人类的智慧》。
- Alto——**图形用户界面(GUI)**
- 麦金托什机(Macintosh)——Mac OS
- Windows
- 面向对象程序设计思想(OOP)
 - 对象(object)实际上是代码和数据的组合
 - 应用于图形操作系统
 - C++
- 集成开发环境(IDE)
- 可视化教程(Visual Programming)

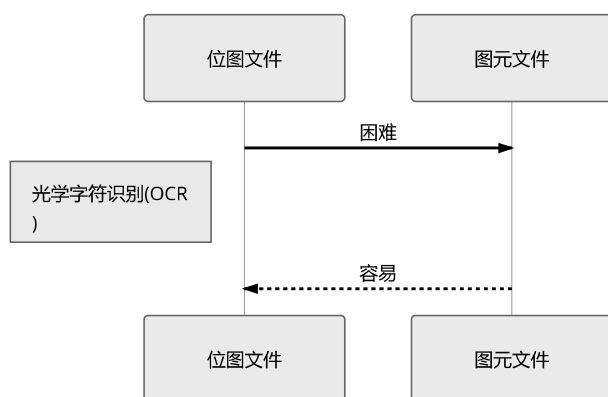
What? 计算机图形的两个分支

- 矢量（图元文件）
- 光栅（位图）

位图和图元文件都是数字化的可视信息。

What? 数据压缩(Data Compression)

- 无损压缩技术
 - 游程长度编码(RLE)
 - 图形交换格式(GIF)-LZW
- 有损压缩技术
 - 联合图像专家组(JPEG)-也包含无损压缩技术



音频信息也能转换成比特和字节。

- 脉冲编码调制技术(PCM)-CD
- 振动是声音之源
- 声波可以被模拟
- 模拟数字转换器(ADC)与数字模拟转换器(DAC)
 - 模拟电压与二进制数的转换

AD：数字化声音相比于模拟声音

- 可以实现无失真的转录或复制

CD可以存储声音，也可以存储数据。(CD-ROM)

- 多媒体(multimedia)
 - 声音
 - 音乐
 - 视频

How? 实现计算机与人对话

- 预先录制，文件形式存储
- 把ASCII码转换成波形数据

把波形数据转换成ASCII码称为**语音识别**，是一个极其复杂的过程。从本质上来讲这是一个利用编程技术使得计算机“理解”人类语言的过程，这正是人工智能(Artificial Intelligence)所研究的领域。

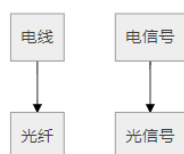
- 数字化电影
 - 压缩技术：移动图像专家组(MPEG)
 - HDTV与DVD
 - DVD将替代CD-ROM成为新的软件存储媒介

与万·布什提出的概念不同，所有信息并不一定要触手可及，真正使它们达到信息共享的做法是**计算机互联**，这样做还可以更有效地利用存储空间。

- 调制与解调
- Internet 非中心化的系统
 - 协议的集合
 - TCP/IP协议
 - 万维网(World Wide Web)
 - HTML格式 **不是编程语言**
 - 超文本(hypertext)-相关链接信息
 - 服务器端(Server)与客户端(Client)

Why? Web站点不为计算机提供一个可执行程序

- 兼容性
- 安全性
- Java
 - 介于编译语言与解释语言之间，编译的结果不是机器码，而是Java字节码
 - Java字节码可以在Java虚拟机上被解释
 - **平台无关性**：Java程序的运行不受限于机器与图形操作系统



光子的速度无与伦比 光子的速度无与伦比 光子的速度无与伦比

2019/9/24 9:24完成编辑

Additions

计算机技术发展史上的杰出人物简列

- 塞缪尔·莫尔斯
(<https://baike.baidu.com/item/%E5%A1%9E%E7%BC%AA%E5%B0%94%C2%B7%E8%8E%AB%E5%B0%94%E6%96%AF/8696998?fromtitle=%E8%8E%AB%E5%B0%94%E6%96%AF&fromid=389249&fr=aladdin>): 发明了莫尔斯码, 促进了第一条电报线路架设完成 (**现代通信开始的标志**)
- 查尔斯·巴贝奇
(<https://baike.baidu.com/item/%E6%9F%A5%E5%B0%94%E6%96%AF%C2%B7%E5%B7%B4%E8%B4%9D%E5%A5%87/5466849>): **计算机事业的伟大先驱**, 分别提出了差分机与解析机的设想
- 路易斯·布莱叶 (<https://baike.baidu.com/item/%E5%B8%83%E8%8E%B1%E5%8F%B6>): 发明了布莱叶盲文 (“点字”)
- 奥古斯都·德·摩根
(<https://baike.baidu.com/item/%E5%A5%A5%E5%8F%A4%E6%96%AF%E9%83%BD%C2%B7%E5%BE%B7%C2%B7%E6%91%A9%E6%A0%B9/2094657>): 发表《形式逻辑》, 提出德·摩根定律
- 乔治·布尔 (<https://baike.baidu.com/item/%E4%B9%94%E6%B2%BB%C2%B7%E5%B8%83%E5%B0%94>): 发表《思维规律》, 提出布尔代数 (**数学描述逻辑的方法出现**)
- 赫尔曼·霍尔瑞斯
(<https://baike.baidu.com/item/%E8%B5%AB%E5%B0%94%E6%9B%BC%C2%B7%E4%BD%95%E4%B9%90%E7%A4%BC/3751583?fromtitle=Herman%20Hollerith&fromid=9819984>): 发明了霍尔瑞斯穿孔卡片 (在1890年美国人口普查中取得成功), 创立了制表机公司 (**IBM公司的前身**)
- 康拉德·楚泽
(<https://baike.baidu.com/item/%E5%BA%B7%E6%8B%89%E5%BE%B7%C2%B7%E6%A5%9A%E6%B3%BD/8179197>): 建造完成第一台继电器计算机
- 艾伦·M·图灵
(<https://baike.baidu.com/item/%E8%89%BE%E4%BC%A6%C2%B7%E9%BA%A6%E5%B8%AD%E6%A3%AE%C2%B7%E5%9B%BE%E7%81%B5?fromtitle=%E5%9B%BE%E7%81%B5&fromid=121208>): **计算机科学之父、人工智能之父**, 提出著名的图灵机模型与图灵测试法, 曾在二战期间协助军方破译德国的"Enigma"代码生成器
- 约翰·安布罗斯·弗莱明、德·福雷斯特
(<https://baike.baidu.com/item/%E5%BE%B7%E7%A6%8F%E9%9B%B7%E6%96%AF%E7%89%B9>): 发明了**真空管** (**成为取代继电器的关键部件**)
- 霍华德·艾肯 (<https://baike.baidu.com/item/%E8%89%BE%E8%82%AF/2286664>): 在IBM公司的资助下, 研制成功世界上第一台自动连续可控计算机(ASCC)Harvard Mark I
- J·普利斯普·埃克特 (<https://baike.baidu.com/item/%E5%9F%83%E5%85%8B%E7%89%B9/8148590>)、约翰·莫克利 (<https://baike.baidu.com/item/%E8%8E%AB%E5%A5%87%E5%88%A9>): 设计出使用了18,000个真空管, 大约30吨重的电子数字积分计算机(ENIAC), 开发了第一台商用计算机——通用自动计算机(UNIVAC)
- 万·布什 (<https://baike.baidu.com/item/%E8%8C%83%E5%86%85%E7%93%A6%C2%B7%E5%B8%83%E4%BB%80/1085792?fr=aladdin>): **“信息时代的教父”**, 信息论之父香农的老师, 发表论文**《思维之际(As We May Think)》**, 其中提出了微缩摄影技术和麦克斯存储器(Memex)的概念
- 约翰·冯·诺依曼
(<https://baike.baidu.com/item/%E7%BA%A6%E7%BF%B0%C2%B7%E5%86%AF%C2%B7%E8%AF%BA%E4%BE%9D%E6%9B%BC/986797?fromtitle=%E8%AF%BA%E4%BE%9D%E6%9B%BC&fromid=30397>): **计算机之父**, 提出了计算机的“冯·诺依曼结构”
- 克劳德·香农
(<https://baike.baidu.com/item/%E5%85%8B%E5%8A%B3%E5%BE%B7%C2%B7%E8%89%BE%E5%B0%94%E4%BC%8D%E5%BE%B7%C2%B7%E9%A6%99%E5%86%9C/10588593?fromtitle=%E9%A6%99%E5%86%9C&fromid=1146248>): **信息论之父**, 发表《通过程中代数理论》, 开创了信息论研究领域
- 约翰·巴丁 (<https://baike.baidu.com/item/%E7%BA%A6%E7%BF%B0%C2%B7%E5%B7%B4%E4%B8%81/4722240?fr=aladdin>)、沃尔特·布拉坦: 发明了**晶体管** (**取代真空管, 开创了固态电子器件时代**)
- 威廉·肖克利 (<https://baike.baidu.com/item/%E5%A8%81%E5%BB%89%C2%B7%E8%82%96%E5%85%8B%E5%88%A9>): 晶体管的发明者之一, 在加利福尼亚帕罗奥图市成立肖克利半导体实验室, **硅谷**走向电子产业新时代的引导者
- 戈登·E·摩尔 (<https://baike.baidu.com/item/%E6%88%88%E7%99%BB%C2%B7%E6%91%A9%E5%B0%94>): Intel公司的创始人之一, 提出著名的**摩尔定律**
- 肯·汤普森: **UNIX操作系统之父**