中国科学院大学
University of Chinese Academy of Sciences

计算机科学导论

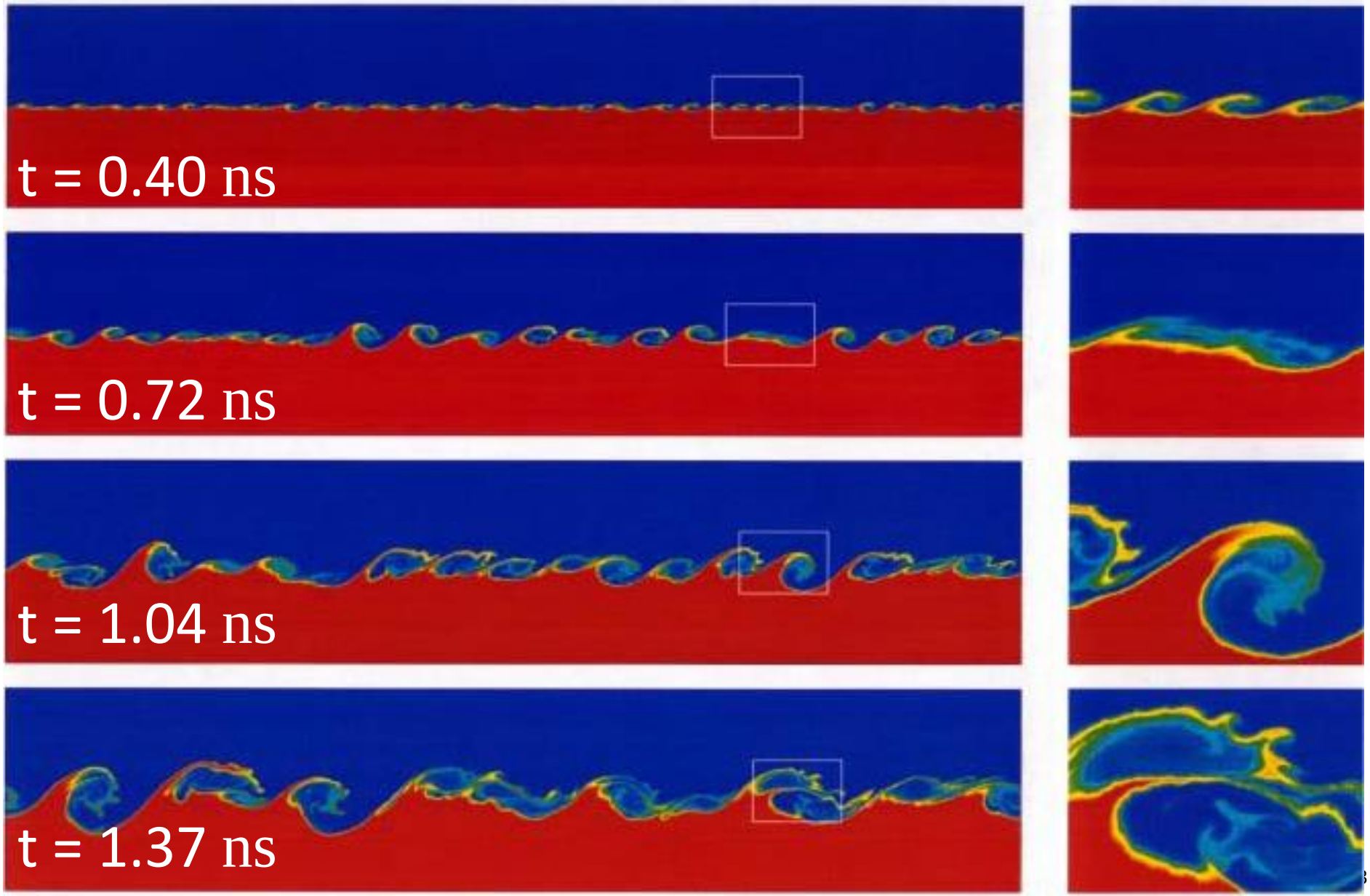
期末复习

徐志伟
中科院计算所 中国科学院大学
zxu@ict.ac.cn

提纲

- 计算机科学导论（与计算思维）的定位
  - 一种认识世界、解决问题的新的科学思维方式
    - 研究计算过程（信息变换过程），而不是物质和能量运动
    - 比特精准的、巧妙构造的、自动执行的数字符号操作序列
    - 不限于IT工作者
  - 知道计算机的工作原理，能够与同行交流
    - 可以动手开发运行算法和程序，验证正确性，分析复杂度
    - **可以动脑构造计算机系统，与同行合作解决具体问题**
    - 知道理论计算机和真实计算机的局限
  - 是计算机科学的入门概述
    - 理论：数据结构，自动机与形式语言，算法
    - 硬件：数字电路，计算机组成，计算机体系结构，计算机网络
    - 软件：编程，编译，操作系统
    - 应用：数据库，多媒体，人工智能；职业操守

Atoms in the Surf



计算机科学的特色

美国科学院与工程

- “计算机科学是研究计算机以及它们能干什么的一门学科。它研究**抽象计算机**的能力与局限，**真实计算机**的构造与特征，以及用于求解问题的无数**计算机应用**。”
  - 计算机科学涉及**符号**及其操作；
  - 关注多种**抽象**概念的创造和操作；
  - 创造并研究**算法**；
  - 创造各种**人工结构**，尤其是不受物理定律限制的结构；
  - 利用并应对**指数增长**；
  - 探索计算能力的**基本极限**，
  - 关注与人类**智能**相关的复杂的、分析的、理性的活动。

图灵机 自动机

冯诺依曼计算机
笔记本电脑

信息隐藏GO程序

数字符号

从电路到算法的
各种抽象

布尔逻辑
命题逻辑
谓词逻辑

斐波那契数列递归程序
P与NP

图灵机不可计算问题
哥德尔不完备定理

计算思维：通过信息变换解决问题

其特征是比特精准、构造解题、自动执行

1比特（bit）是一个二进制数位，取值0或1

具体有十种理解

理解1：自动执行。计算机能够自动执行由离散步骤组成的计算过程。

理解2：正确性。计算机求解问题的正确性往往可以精确地定义并分析。

理解3：通用性。计算机能够求解任意可计算问题。

理解4：构造性。人们能够构造出聪明的方法让计算机有效地解决问题。

理解5：复杂度。这些聪明的方法（算法）具备时间/空间复杂度。

理解6：连接性。很多问题涉及用户/数据/算法的连接体，而非单体。

理解7：协议栈。连接体的节点之间通过协议栈通信交互。

理解8：抽象化。少数精心构造的计算抽象可产生万千应用系统。

理解9：模块化。多个模块有规律地组合成为计算系统。

理解10：无缝衔接。计算过程在计算系统中流畅地执行。

自动

通用

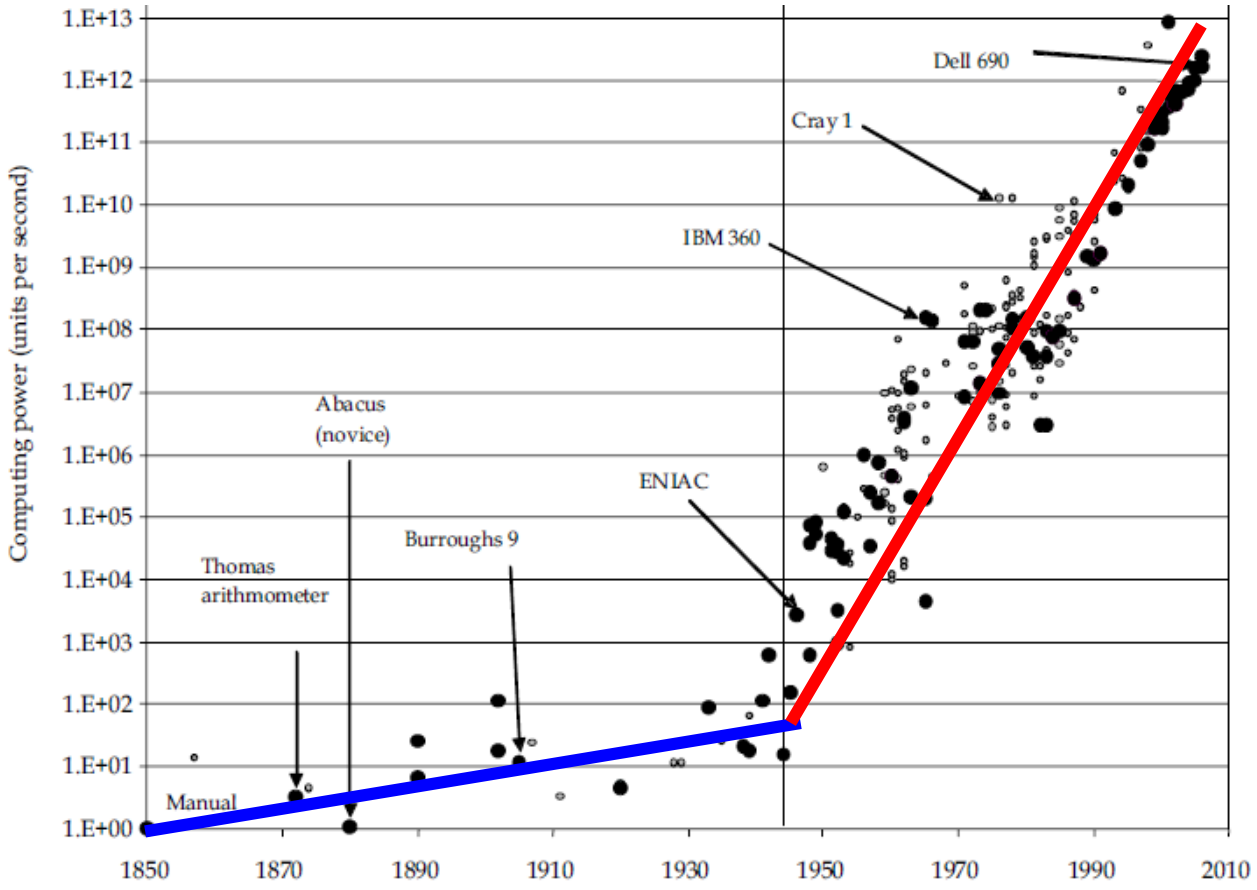
算法

联网

抽象

Nordhaus's Law

- 为什么在1945年左右出现拐点？
- 自动执行！
- 自动执行的数字（符号）电子计算机
- 存储程序计算机



一些计算机科学基本抽象

- 抽象使得计算机科学成为一门优美的领域
  - 本质：采用一种方法解决该层次的所有问题，应对系统的复杂性，**以不变的抽象应系统的万变**

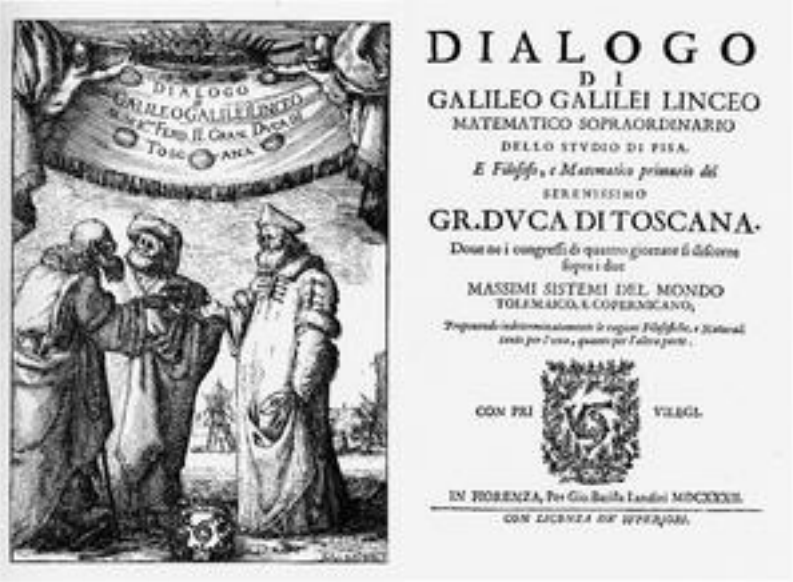
数字符号	**比特、字节、四进制数、十六进制数、整数、数组、BMP 图像。**	
软件	**算法**	巧妙的信息变换方法。例如信息隐藏算法。
	程序	算法的代码实现。例如实现信息隐藏算法的 hide.go 程序。
	进程	运行时的程序。例如在 Linux 环境中的 hide 进程。
	指令	程序的最小单位，计算机能够直接执行。
硬件	**指令流水线**	每条指令都通过“取指-译码-执行-写回”四个操作组成的指令流水线得以自动执行。所有指令都由这一种机制执行，指令流水线由若干时钟周期组成。
	时序电路	等同于**自动机**，说明每一个时钟周期的操作。时序电路由组合电路与存储单元组成，理论上等同于**图灵机**。
	组合电路	实现二值逻辑表达式（**布尔逻辑表达式**）。

什么是“有效解决问题”？

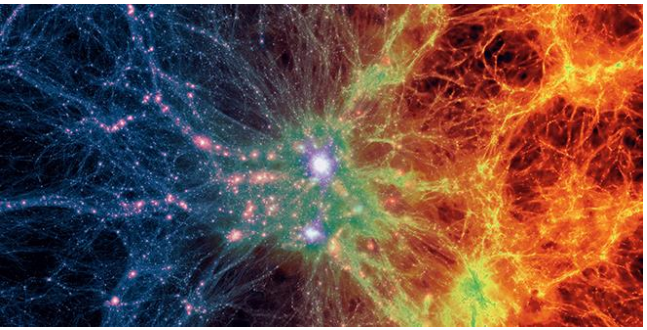
- 可以是自然科学、工程技术、社会科学的问题
  - 也可以是经济社会的实用问题
- **有效建模**（encoding, modeling）并**有效计算**
  - 将这些问题转换成为计算过程问题
  - 求解计算问题
  - 将解转换回原问题
- 讲究算得准、算得快、算的爽（应对复杂性）
  - 多准才是准？→ 领域各自的准确性 + 比特精准
  - 多快才是快？→ 用户体验 + 越快越好
    - 科学计算flops，电子商务tps，股票交易s→ms→μs
  - **通过“比特精准、巧妙构造、自动执行的计算过程”解题**

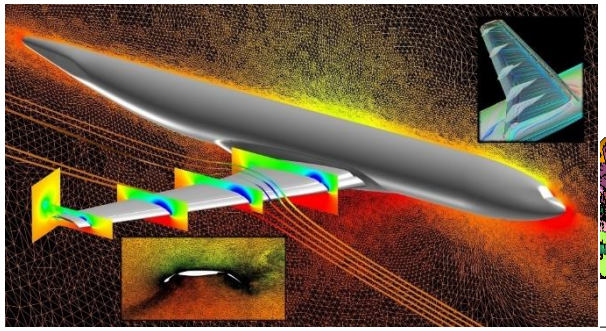
为什么计算机科学可以解决多类问题？

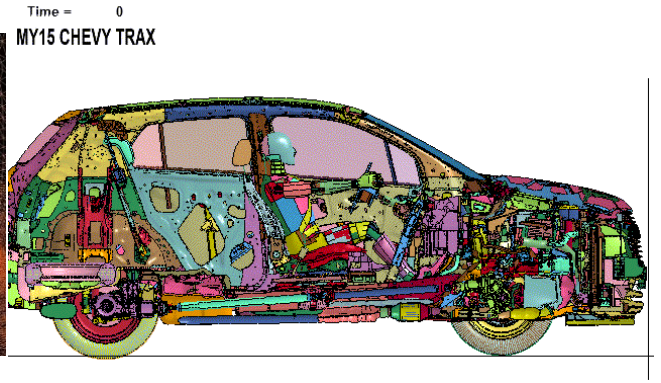
- 乔姆斯基数字无穷性假说（也称为离散无穷性）
- 维基百科： **Digital infinity** is a technical term in theoretical linguistics. Alternative formulations are "discrete infinity" and "the infinite use of finite means". The idea is that all human languages follow a simple logical principle, according to which a limited set of digits—irreducible atomic sound elements—are combined to produce an infinite range of potentially meaningful expressions.
- 据说来源于伽利略： In his *Dialogo*, Galileo describes with wonder the discovery of a means to communicate one‘s “most secret thoughts to any other person ... with no greater difficulty than the various collocations of twenty-four little characters upon a paper.” This is the greatest of all human inventions, …



为什么计算机科学渗透广而深？







科学、技术、经济、社会的各种问题和过程

各领域的专业表达

都可以用该领域的语言表示

数字无穷性假说
+
计算透镜假说

有穷数字符号的组合可表达无穷语言
+
Nature computes.
Society computes.

计算问题和计算过程

给地球做CT

二进制表示

- $(110.101)_2=1\times2^2+1\times2^1+1\times2^{-1}+1\times2^{-3}=(6.625)_{10}$
- 6.625的二进制表示是什么？

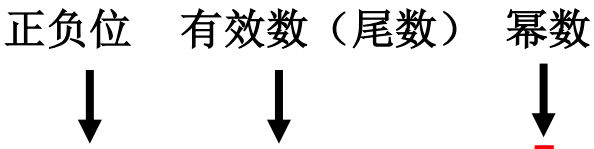
- 课程讨论了如下数据类型的二进制表示，同学们通过GO编程练习动手掌握
  - ASCII字符
  - 自然数
  - 整数
  - ……
  - 数组、切片
  - bmp图像文件

符号有窍门
二进制补码例子：-127到127的整数加法

- 直截了当的表示
  - 需要8比特，2^7=128，以及1比特表示正负
  - 63 = 00111111，64 = 01000000
  - (-63) = 10111111，(-64) = 11000000
  - 63 + 64 = 00111111 + 01000000 = 01111111 = 127
  - (-63) + (-64) = 10111111 + 11000000 = 11111111 = (-127)
  - 63 + (-63) = 00111111 + 10111111 = 11111110 = (-126) 错！
- 补码表示
  - 负数：绝对值的逐位取反，然后加00000001，即从最低一位加1
  - (-63) = 00111111逐位取反 + 00000001 = 11000000 + 00000001 = 11000001
  - 63 + (-63) = 00111111 + 11000001 = 00000000 = 0 正确！
- 溢出：127+1=？ (-127)+(-1)=?

如何表示实数：浮点数概念

- 本课程编程练习中不涉及
- 特定的有穷数字符号组合不能精确表达无穷数
- 办法：近似表达
  - 例如，圆周率π=3.14159265…，假设6位十进制精度



  - $3.14159 = + 314159 \times 10^{-5}$

- 一个浮点数可由两个整数（定点数）表示

逻辑思维

- 比特精准的最基本体现
- 命题逻辑、布尔函数、组合电路
  - 公理系统（从代数角度理解）
    - 元素：常数（0,1）、变量；布尔表达式
    - 算子：基本操作（与、或、非）；其他（$\oplus$，$\rightarrow$）
    - 公理（性质）
    - 推理规则
  - 用真值表辅助
- 谓词逻辑
  - 多了断言和量词（**全称量词∀**和**存在量词∃**）
    - 断言是会传回“真”或“假”的函数
      - 任何自然数，要么它是偶数，要么它加1后为偶数。
      - $\forall n\ [\mathrm{Even}(n) \vee \mathrm{Even}(n+1)]$

性质

- $x \vee 0 = x, x \vee 1 = 1, x \wedge 0 = 0, x \wedge 1 = x$
- $x \vee \neg x = 1, x \wedge \neg x = 0$
- $x \wedge y = y \wedge x, x \vee y = y \vee x$ （交换律）
- $(x \wedge y) \wedge z = x \wedge (y \wedge z)$
- $(x \vee y) \vee z = x \vee (y \vee z)$ （结合律）
- $(x \wedge y) \vee z = (x \vee z) \wedge (y \vee z)$
- $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$ （分配律）
- $\neg(x \vee y) = \neg x \wedge \neg y$
- $\neg(x \wedge y) = \neg x \vee \neg y$ （De Morgan律）

- $x \rightarrow y = \neg x \vee y$
- $x \oplus y = (\neg x \wedge y) \vee (x \wedge \neg y)$

任意布尔函数有范式（唯一性）

- 合取范式(conjunctive normal form, CNF)
  - $f(x_1,\ldots,x_n)= Q_1 \wedge Q_2 \wedge Q_3 \ldots \wedge Q_m$
  - 其中: $Q_i = l_1 \vee l_2 \vee \ldots \vee l_n$，　$l_j = x_j$ 或 $\neg x_j$
- 析取范式(disjunctive normal form, DNF)
  - $f(x_1,\ldots,x_n)= Q_1 \vee Q_2 \vee Q_3 \ldots \vee Q_m$
  - 其中: $Q_i = l_1 \wedge l_2 \wedge \ldots \wedge l_n$，　$l_j = x_j$ 或 $\neg x_j$
  - 等价于真值表
- 写出2个变量的全部函数的析取范式（一共有2^{2^n}=8个）
  - $y = \overline{x_1} \cdot x_1 = 0$ 【注】　　$y = \overline{x_1} \cdot \overline{x_2} + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2} + x_1 \cdot x_2 = 1$
  - $y = x_1 \cdot \overline{x_2} + x_1 \cdot x_2 = x_1$　　$y = \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2} + x_1 \cdot x_2 = x_1 + x_2$　　……

x_1	x_2	y
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**0**

x_1	x_2	y
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**1**

x_1	x_2	y
0	0	**0**
0	1	**0**
1	0	**1**
1	1	**1**

x_1	x_2	y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

逻辑思维

- 存在无穷多个素数。
- 假如只有有限个素数，……，得出矛盾

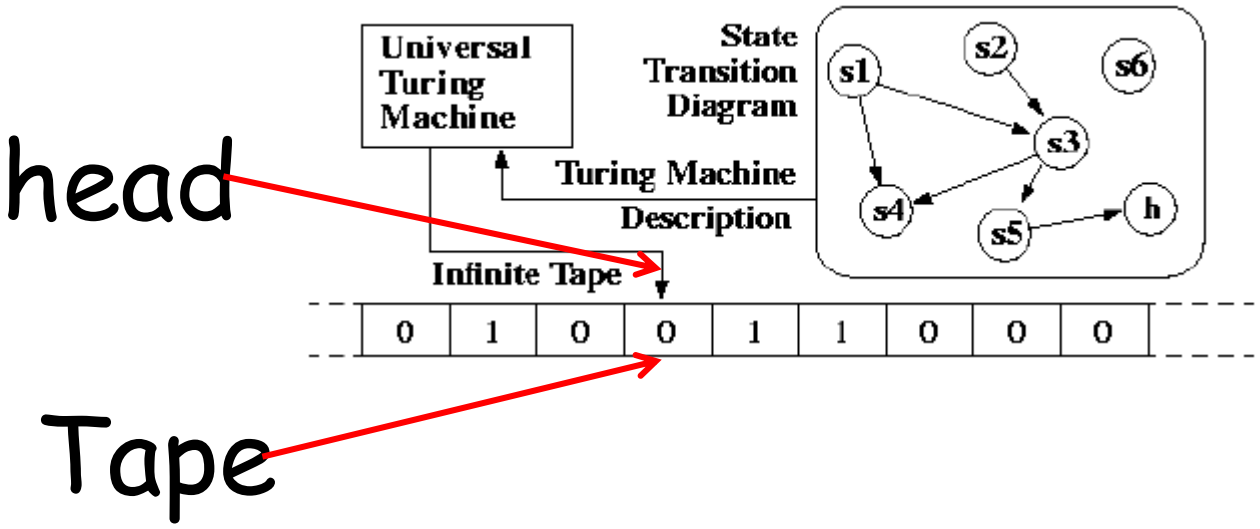
$$\forall n, \exists m, [(m > n) \wedge (\text{Prime}(m))]$$

$$\neg(\exists n, \forall m, [(m > n) \rightarrow \neg(\text{Prime}(m))])$$

- **能够比特精准地推导和证明**
  - 能自己想明白
  - 能和别人说明白

图灵机(Turing Machine)

教科书234-235页有一个例子



...an unlimited memory capacity obtained in the form of an infinite tape marked out into squares, on each of which a symbol could be printed. At any moment there is one symbol in the machine; it is called the scanned symbol. The machine can alter the scanned symbol and its behavior is in part determined by that symbol, but the symbols on the tape elsewhere do not affect the behavior of the machine. However, the tape can be moved back and forth through the machine, this being one of the elementary operations of the machine. Any symbol on the tape may therefore eventually have an innings. (by Turing 1948)

图灵机的通用性和局限性

- 计算机能够求解任意可计算问题
  - 有限性：真实计算机只有有限精度和有限存储
- 丘奇-图灵论题（**Church-Turing Hypothesis**）
  - （任何人类用纸和笔所能做的计算）与（图灵机所能做的计算）等价。
  - *Any reasonable attempt to model mathematically computer algorithms and their performance is bound to end up with a model of computation and associated time cost that is equivalent to Turing machines within a polynomial.*
- 存在不可计算问题
  - 例如：停机问题

Godel不完备性定理

- 定理一：任意一个包含一阶谓词逻辑与初等数论的形式系统，都不可能同时拥有完备性和一致性。即存在一个真命题，它在这个系统中不能被证明。
- 定理二：任意一个包含初等数论的系统S，当S无矛盾时，它的无矛盾性不可能在S内证明。

Kurt Godel

1906-1978

真和可以被证明是两件事情!!

算法思维

- **高德纳的算法定义：**一个算法是一组有穷的规则，给出求解特定类型问题的运算序列，并具备下列五个特征：

（1）有穷性：一个算法在有限步骤之后必然要终止。

（2）确定性：一个算法的每个步骤都必须精确地（严格地和无歧义地）定义。

（3）输入：一个算法有零个或多个输入。

（4）输出：一个算法有一个或多个输出。

（5）能行性：一个算法的所有运算必须是充分基本的，原则上人们用笔和纸可以在有限时间内精确地完成它们。

算法思维

- 分治思想
  - 快速排序
  - 单因素优选法
  - 大整数乘法
  - 矩阵乘法
- 贪心、穷举、回溯、动态规划……
- 随机算法、近似算法、在线算法、流算法……
- 数据结构、理论计算机基础、算法设计与分析、高等算法……

算法例子：
求解斐波那契数列的数学问题

- 数学表达：F(0)=F(1)=1；F(n)= F(n-1)+ F(n-2)
- 一个递归调用程序，直观上与数学表达很接近

```
1.  package main
2.  import "fmt"
3.  func main() {
4.          for i := 0; i <= 50; i++ {
5.                  fmt.Println(i, ":", fibonacci(i))
6.          }
7.  }
8.  func fibonacci(n int) int {
9.          if n == 0 || n == 1 {
10.                 return n
11.         }
12.         return fibonacci(n-1) + fibonacci(n-2)
13. }
```

GO代码

算法的复杂度分析

- 求解斐波那契数列的递归算法GO代码耗时多少？

```
func fibonacci(n int) int {
        if n == 0 || n == 1 {
                return n
        }
        return fibonacci(n-1) + fibonacci(n-2)
}
```

- 复杂度分析
  - T(n) = T(n-1) + T(n-2)
    - 2*T(n-2) < T(n) < 2*T(n-1), 当n > 2时
  - T(n) < 2*T(n-1) < 4*T(n-2) < 8*T(n-3) … < 2^n
  - T(n) > 2*T(n-2) > 4*T(n-4) > 8*T(n-6) … > $2^{n/2}$
  - **T(n) = O(2^n)， T(n) = Ω($2^{n/2}$)**　　**指数复杂度！**
- 动态规划算法：复杂度O(n)

小o，大O， Ω，Θ记号

共同假设：当n足够大

- $f(n) = o(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
  - $n^{1.58} = \mathbf{o}(n^2)$， $n^{1.58} \neq \mathbf{o}(n^{1.58})$ ， $n^2 \neq \mathbf{o}(n^{1.58})$
- $f(n) = O(g(n))$ ∃常数$c > 0, f(n) \leq cg(n)$
  - $n^{1.58} = \mathbf{O}(n^2)$， $n^{1.58} = \mathbf{O}(n^{1.58})$， $n^2 \neq \mathbf{O}(n^{1.58})$
- $f(n) = \Omega(g(n))$ ∃常数$c > 0, f(n) \geq cg(n)$
  - $n^{1.58} \neq \mathbf{\Omega}(n^2)$， $n^{1.58} = \mathbf{\Omega}(n^{1.58})$， $n^2 = \mathbf{\Omega}(n^{1.58})$
- $f(n) = \Theta(g(n))$ $f(n) = O(g(n))$ 并且 $f(n) = \Omega(g(n))$
  - $n^{1.58} \neq \mathbf{\Theta}(n^2)$， $n^{1.58} = \mathbf{\Theta}(n^{1.58})$， $n^2 \neq \mathbf{\Theta}(n^{1.58})$

NP vs P

- 输入2*n*个数，判断是否可以把这些数等分成两组（每组n个数），使得两组的和相同。
- 是否是NP的？
  - 是！
  - 为什么？
    - 给定结果，即满足题意的分组方式
    - 验证算法：验证每组都是n个数，且两组数的和相等
    - **验证算法是多项式的**（事实上验证算法的复杂度是O(n)）！
- 是否是P的？
  - 目前不知道。如果找到多项式时间算法，则NP=P。

NP vs P

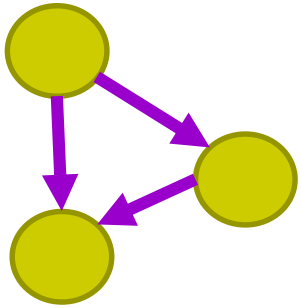
- 输入2*n*个数，判断是否可以把这些数等分成两组（每组n个数），使得两组的和不相同。
- 是否是NP的？
  - 是！方法类似之前。
- 是否是P的？
  - 是！
  - 只要这2n个数不全相同，则答案为是。为什么？
  - O(n)

NP vs. P

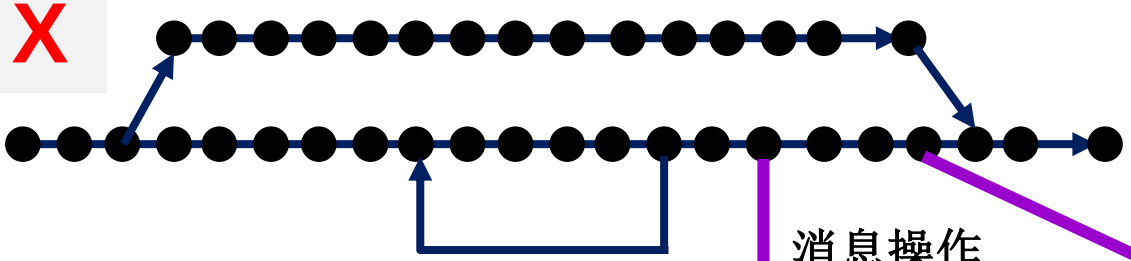
- 输入2*n*个已经排好序的数，判断是否可以把这些数等分成两组（每组n个数），使得两组的和不相同。
- 是否是P的？
  - 是！类似之前
  - 如何判断这2n个数是否都相同？
  - 只需要读第一个数和最后一个数，判断他们是否相同。
  - O(1)

网络思维

- 计算过程涉及（由多个**节点**连接而成的）**网络**
  - 网络成为计算过程的对象、执行系统
- 核心概念：连通性、消息传递、协议
  - **名字空间、拓扑、协议栈**



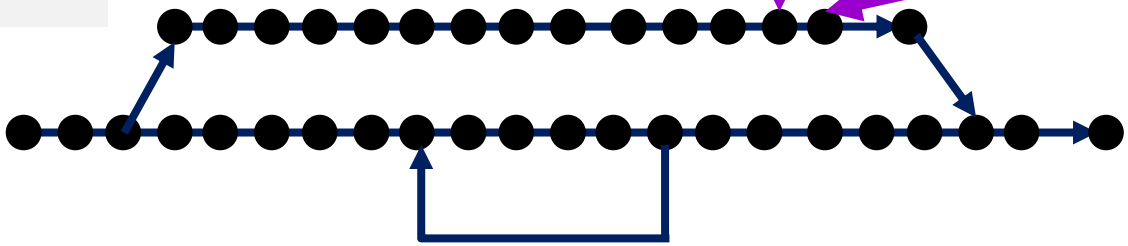
X



消息操作

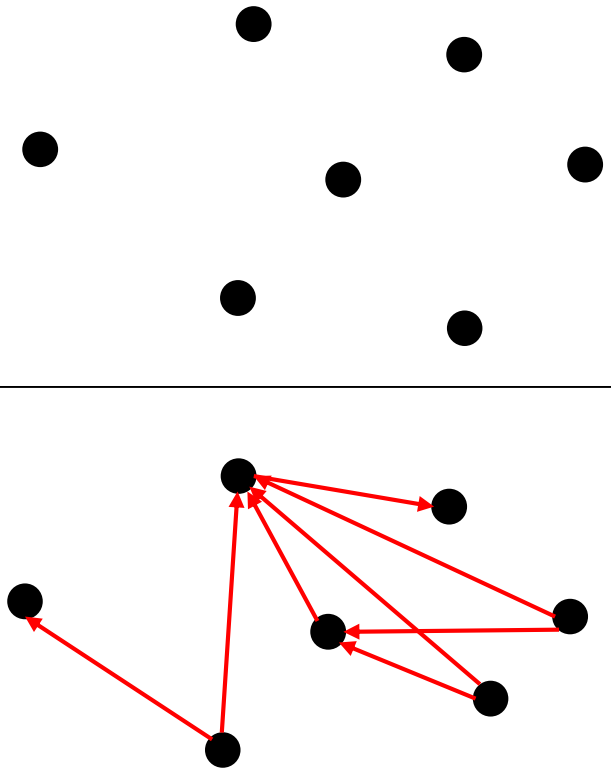
Z

Y



连通性与消息传递是松耦合关系

- 网络思维并不必涉及消息传递（或通信协议）
- 此时，重点是**连通性**（connectivity）
  - 即：有什么节点？节点之间如何连接？
  - 拓扑本身就有价值
- 搜索引擎实例
  - 第一代：无网络思维
    - 只关心节点的内容
  - 第二代：有网络思维
    - Page、 Kleinberg、李彦宏
    - 关心节点内容
    - **还关心网络拓扑（pagerank）**



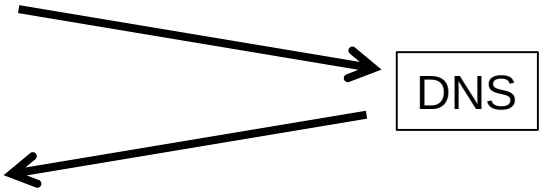
名字空间

- Name space；naming
- 主要用于指称网络中的节点

名字空间实例	节点的名字举例	名字空间解释
微信名字	中关村民	腾讯公司规定的任意“合法的”字符串
电子邮箱地址	zxu@ict.ac.cn	用户名@因特网域名
手机号码	189-8888-9999	通信公司规定的11位10进制数字串
本机文件路径（本地路径）	/我的文件/教材.pdf	本机操作系统规定的文件名
本机网卡地址（MAC地址）	00-1E-C9-43-24-42	全球统一规定的12位16进制数字串
网站域名	www.ict.ac.cn	互联网协议栈规定的域名
网站IP地址	159.226.97.84	IP协议规定的合法地址

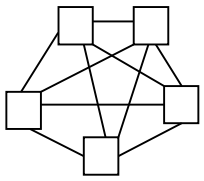
名字空间设计的两个问题

- 设计与理解名字空间的基本考虑
  - **唯一性**：zxu@ict.ac.cn　vs.　中关村民
    - 重用性：手机号码 vs. 万维网资源的URI
  - **动态性**：一个图形加速卡可以插到另一台设备总线上吗？
    - 固定IP地址 vs. 动态生成IP地址
  - **友好性**：中关村民 vs. 以太网MAC地址
- 不同层次间的名字如何解析
  - 本地解析 vs. 全网解析（远程解析）
    - http://www.ict.ac.cn/本地路径...
    - http://159.226.97.84/本地路径...

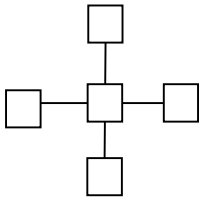


按动态性划分的三类网络拓扑

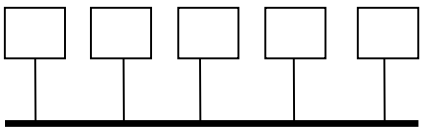
- 静态网络：节点完全确定、连接完全确定
- 动态网络：节点完全确定、连接部分确定
- 演化网络：节点部分确定、连接部分确定
- **你的微信朋友圈是什么网络？**



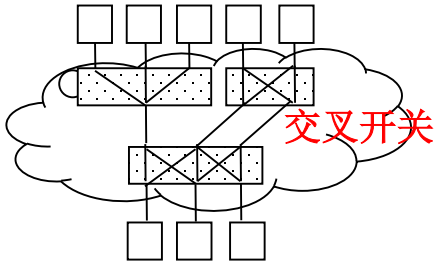
(a) 全连通图



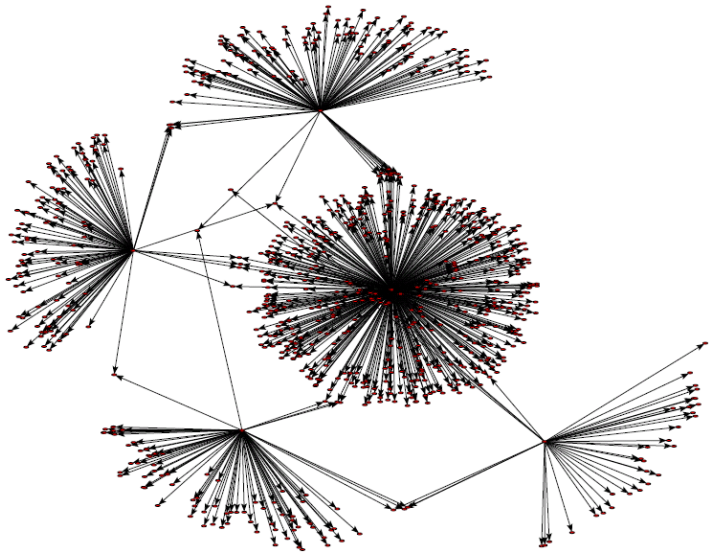
(b) 星型网络



(c) 总线



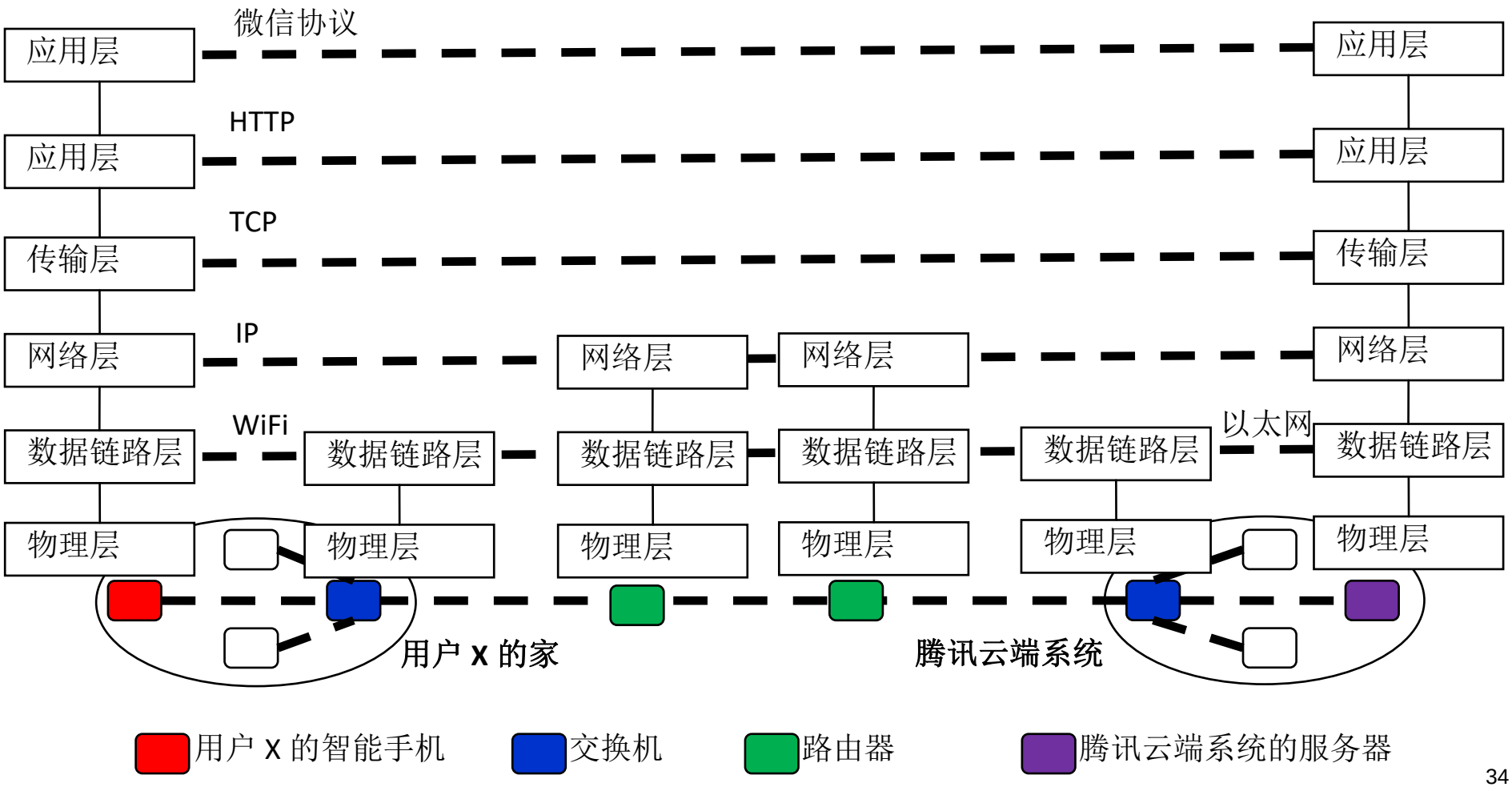
(d) 交换网络



(e) 演化网络

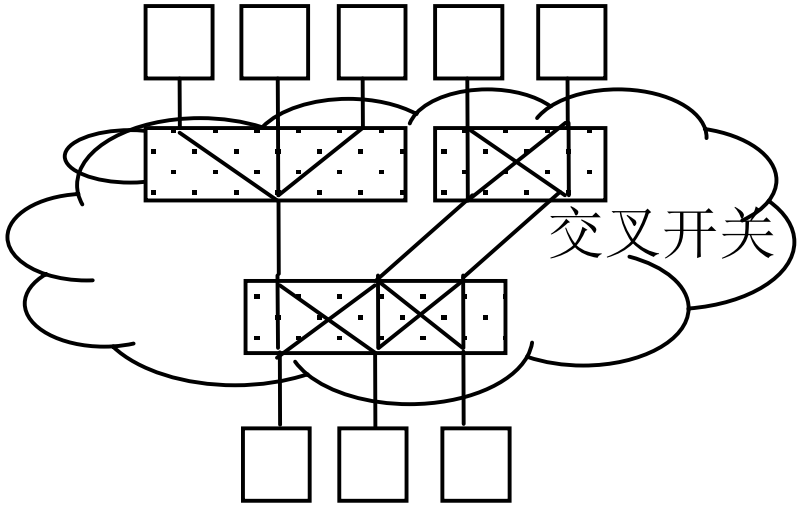
通信过程涉及互联网协议栈的哪些接口？

- 对等接口、层间接口



“到家了！”如何解析成底层的消息包？

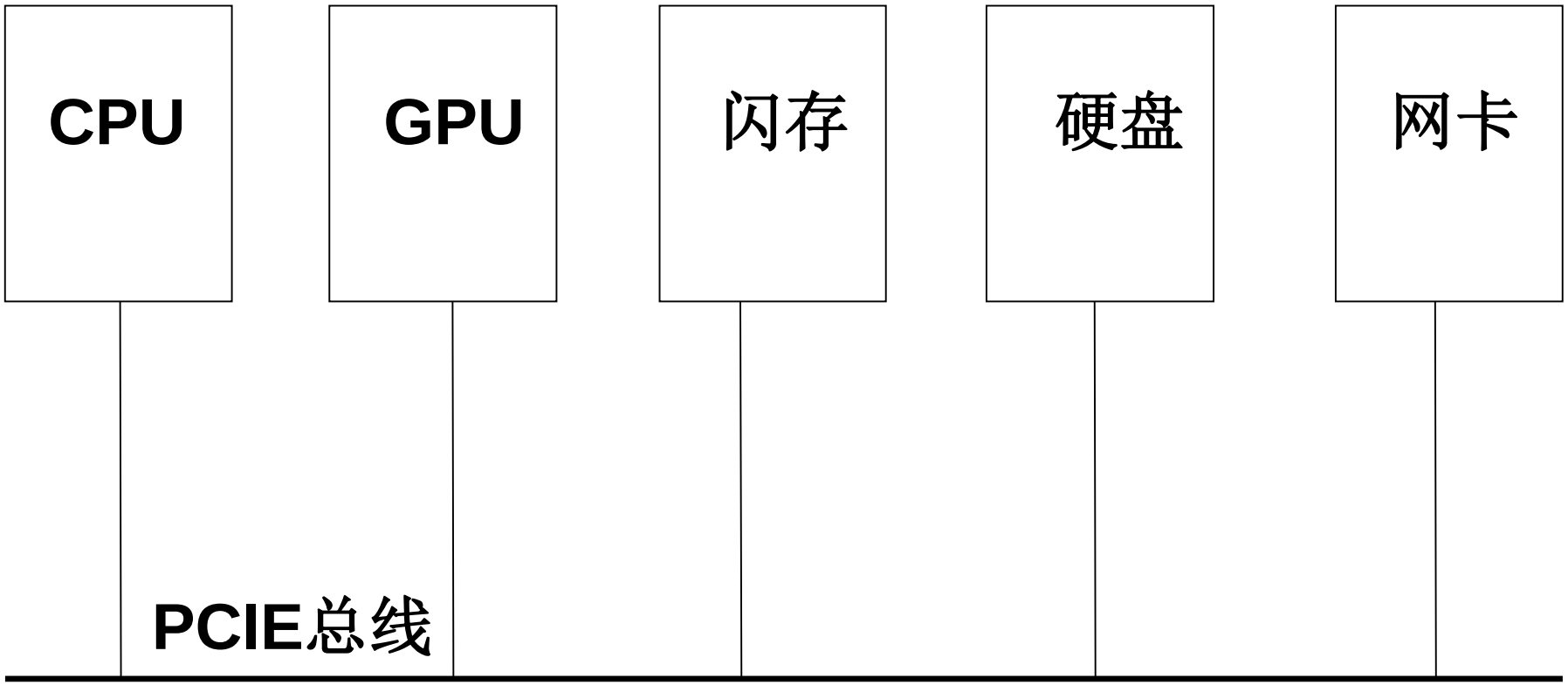
- 关键技术点
  - 线路虚拟化：分组交换
  - 包：首部+数据
  - 逐层解析并传输



微信协议：中关村民送消息“回家了！”给好友 Y

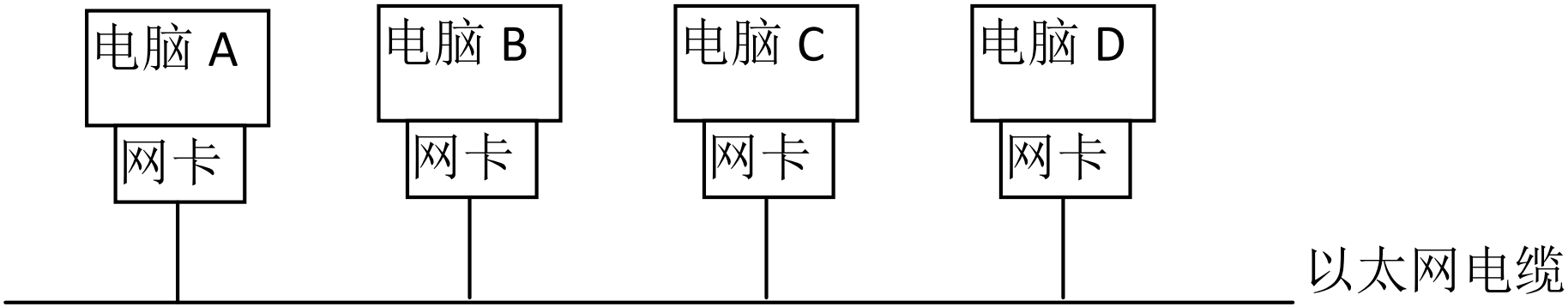
应用层		
应用层	HTTP	GET http://mp.weixin.qq.com/...回家了！...
传输层	TCP	TCP 首部 [80 端口] | GET :// /...回家...
网络层	IP	IP 首部 [140.206.160.199] | TCP 首部 | GET :// /...回家...
WiFi 层	802.11	WiFi 首部 | IP 首部 | TCP 首部 | GET :// /...回家...

在一个主板上的总线仲裁实例



局域网与以太网

- 不能使用集中式的总线仲裁方式
- 1973年，麦特考夫发明以太网
  - 解决冲突的指数退避方法
    - 第一次传输试图失败后，等候[0, T]中间的一个随机值
    - 第二次重试失败后，等候[0, 2T]中间的一个随机值
    - 第三次重试失败后，等候[0, 4T]中间的一个随机值



以太网和四台电脑构成的局域网

计算系统思维

- 通过**抽象**，将**模块**组合成为系统，**无缝**执行计算过程
  - **抽象化**：一个通用抽象代表多个具体需求
  - **模块化**：系统由多个模块组合而成
    - 计算机 = 硬件 + 系统软件 + 应用软件
    - 全系统一致性；信息隐藏原理，接口概念
  - **无缝衔接**
    - 避免缝隙：
      - 扬雄周期原理、波斯特尔鲁棒性原理、冯诺依曼穷举原理
    - 重视瓶颈：阿姆达尔定律
      - 系统性能改进受限于系统瓶颈（针对某任务）
        - 加速比 = $1 / ((1-f)/p + f) \rightarrow 1/f$ 当 $p \rightarrow \infty$

一些计算机科学基本抽象

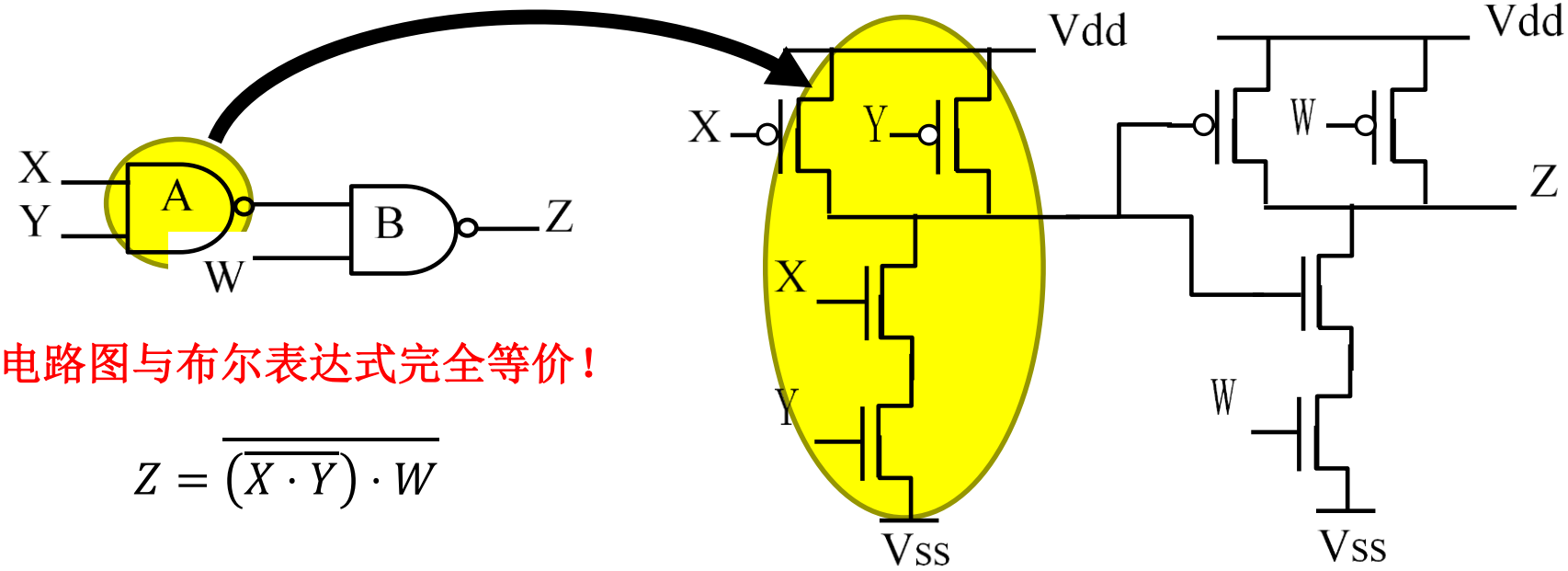
- 抽象使得计算机科学成为一门优美的领域
  - 本质：采用一种方法解决该层次的所有问题，应对系统的复杂性，**以不变的抽象应系统的万变**

数字符号	**比特、字节、四进制数、十六进制数、整数、数组、BMP 图像。**	
软件	**算法**	巧妙的信息变换方法。例如信息隐藏算法。
	程序	算法的代码实现。例如实现信息隐藏算法的 hide.go 程序。
	进程	运行时的程序。例如在 Linux 环境中的 hide 进程。
	指令	程序的最小单位，计算机能够直接执行。
硬件	**指令流水线**	每条指令都通过“取指-译码-执行-写回”四个操作组成的指令流水线得以自动执行。所有指令都由这一种机制执行，指令流水线由若干时钟周期组成。
	时序电路	等同于**自动机**，说明每一个时钟周期的操作。时序电路由组合电路与存储单元组成，理论上等同于**图灵机**。
	组合电路	实现二值逻辑表达式（**布尔逻辑表达式**）。

信息隐藏原理实例

- 图中三者表达同样的逻辑电路
- 但抽象不同，暴露的信息不同

W	X	Y	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



门电路图与布尔表达式完全等价！

$$Z = \overline{(\overline{X \cdot Y}) \cdot W}$$

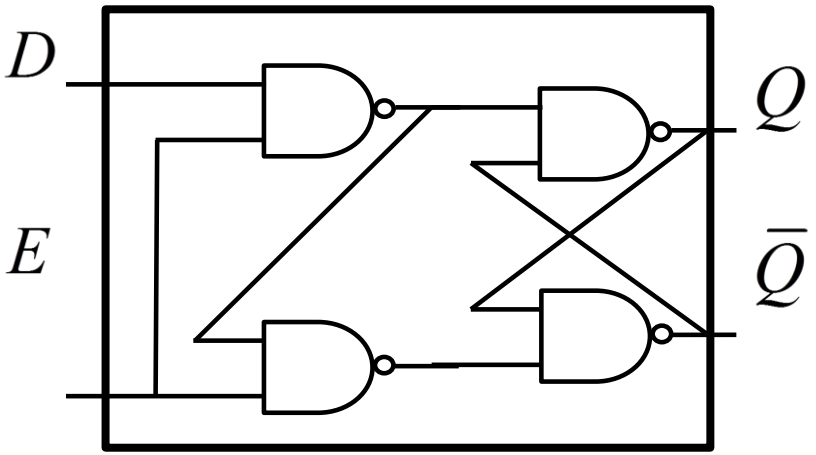
接口：逻辑值+逻辑操作　　电压值+晶体管操作

D-触发器

- Delay Flip-Flop
  - Enable往往用时钟信号CLK
  - 一拍时钟后，Q=D

E	D	Q	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

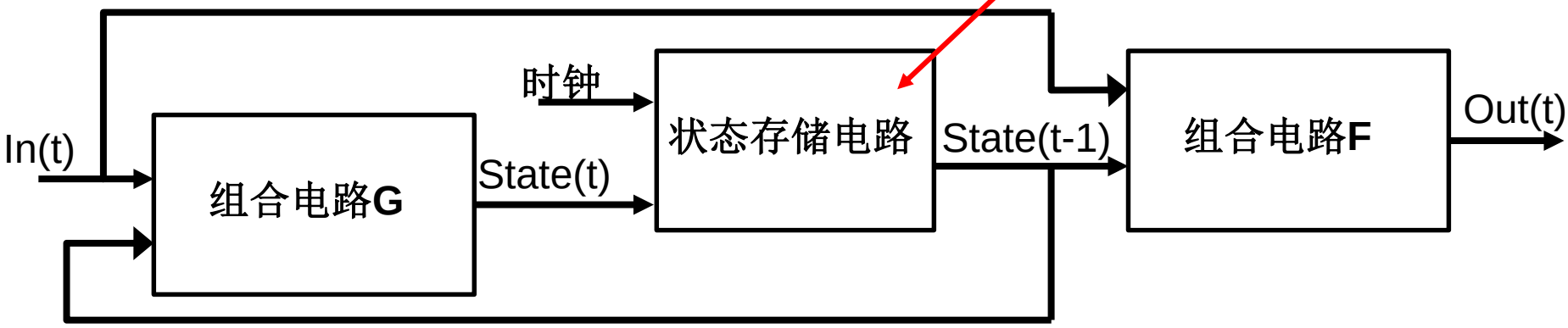
抽象化
隐藏内部细节



组合电路与状态电路产生自动机
即时钟同步的时序电路

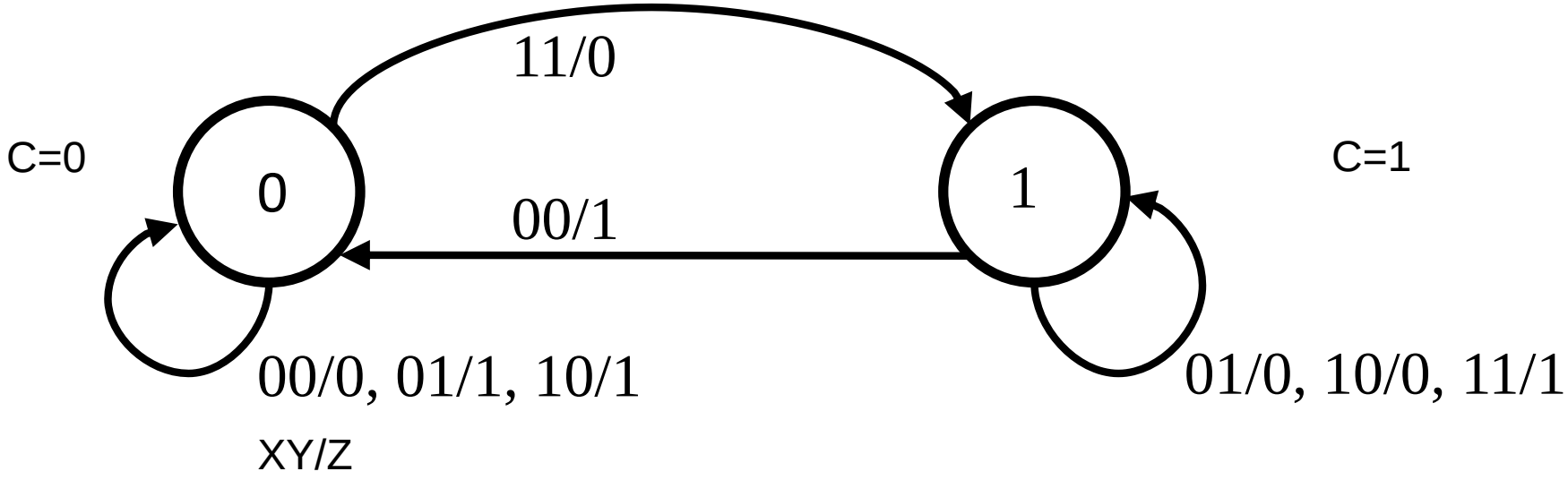
- 第t时刻的输出是t时刻输入与t-1时刻状态的函数
- 第t时刻的状态是t时刻输入与t-1时刻状态的函数
  - Out(t) = F(In(t), State(t-1))
  - State(t) = G(In(t), State(t-1))

往往用**D-触发器实现**



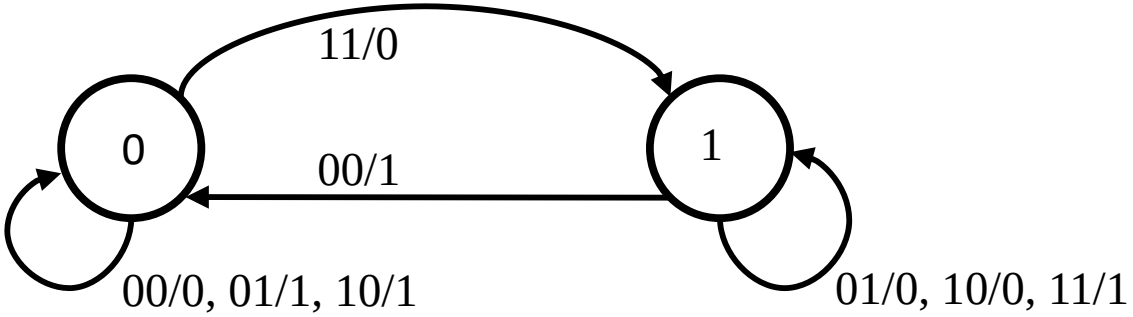
自动机实现4位串行加法过程

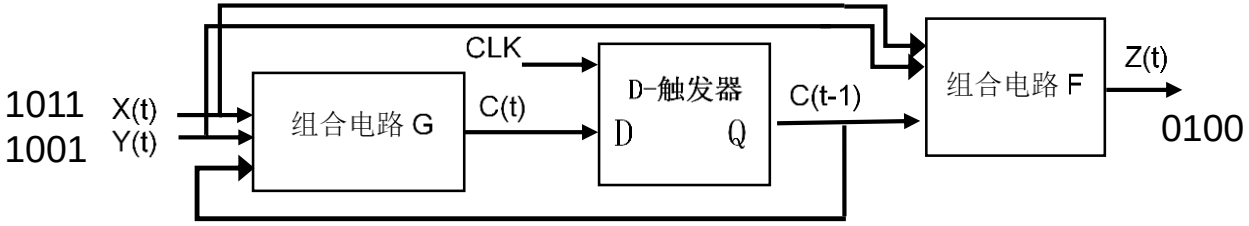
- 1011+1001 = 10100
  - t=0的初始状态：C=0。自动机处于左边状态。
  - t=1：X=1，Y=1；有向边11/0适用，自动机转移到右边状态，输出Z=0。
  - t=2：X=1，Y=0；有向边10/0适用，自动机保持在右边状态，输出Z=0。
  - t=3：X=0，Y=0；有向边00/1适用，自动机转移到左边状态，输出Z=1。
  - t=4：X=1，Y=1；有向边11/0适用，自动机转移到右边状态，输出Z=0。



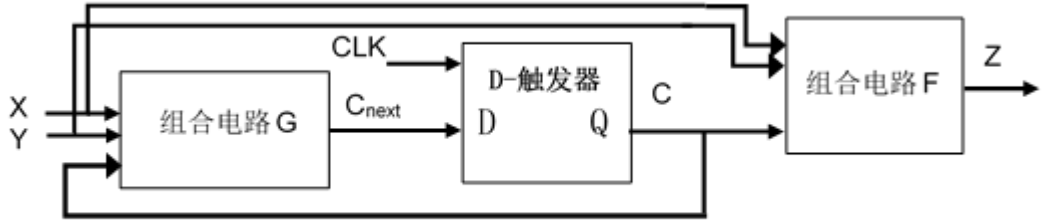
时序电路实现串行加法器

Z=X+Y=1011+1001=10100



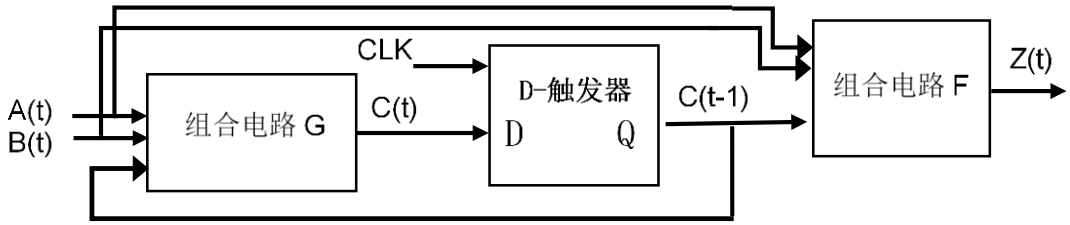


C	X	Y	C_{next}	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



完成实现并验证！

C	X	Y	C_{next}	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$Z = X \oplus Y \oplus C$$
$$C_{next} = (X \cdot Y) + (X \oplus Y) \cdot C$$

1011+1001 = 10100的4位加法过程如下：

- t=0的初始状态：C=0。
- t=1：X=1，Y=1；

$Z = 1 \oplus 1 \oplus 0 = \mathbf{0}, C = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = 1$

- t=2：X=1，Y=0；

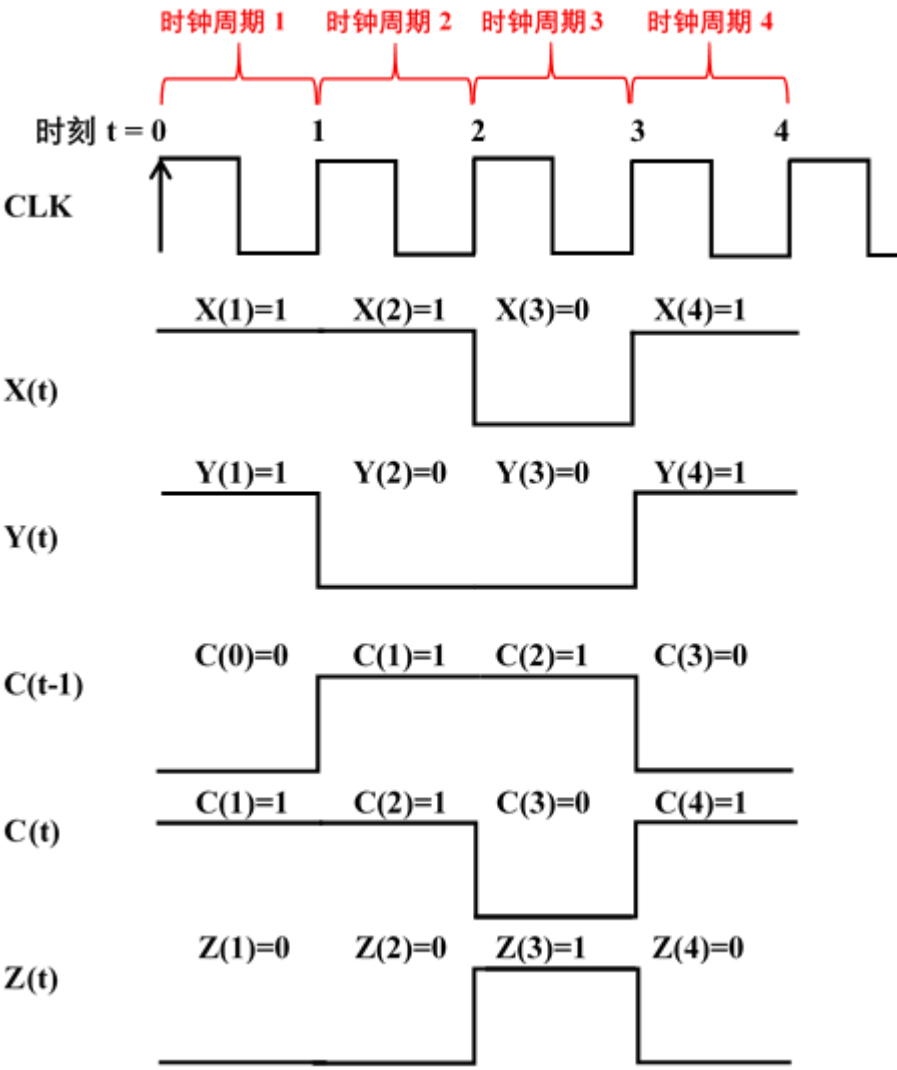
$Z = 1 \oplus 0 \oplus 1 = \mathbf{0}, C = (1 \cdot 0) + (1 \oplus 0) \cdot 1 = 1$

- t=3：X=0，Y=0；

$Z = 0 \oplus 0 \oplus 1 = \mathbf{1}, C = (0 \cdot 0) + (0 \oplus 0) \cdot 1 = 0$

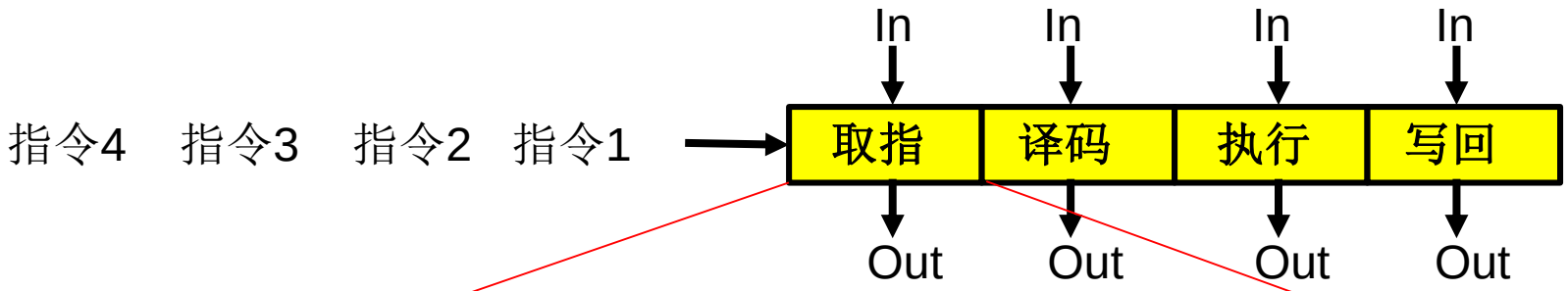
- t=4：X=1，Y=1；

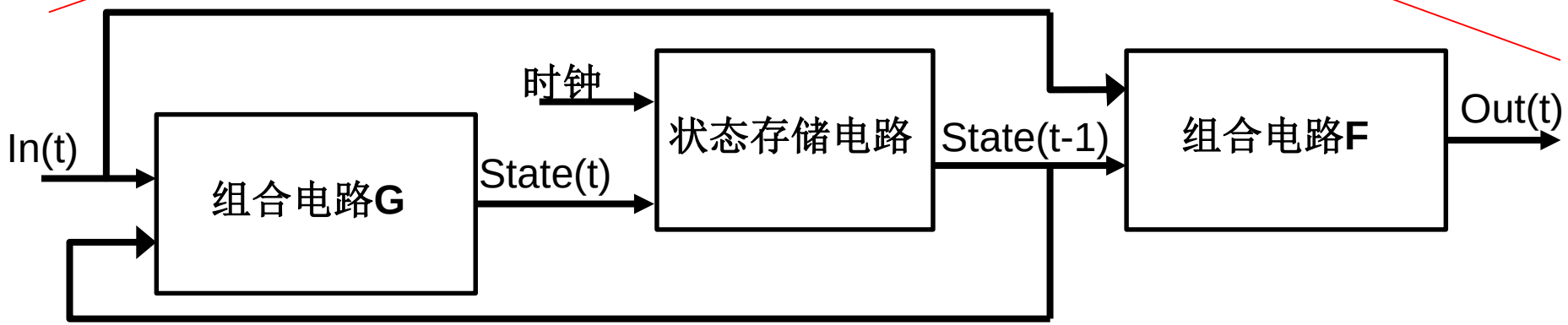
$Z = 1 \oplus 1 \oplus 0 = \mathbf{0}, C = (1 \cdot 1) + (1 \oplus 1) \cdot 0 = \mathbf{1}$



自动机、时序电路是系统基本概念
几乎无处不在

- 例如，任何程序、所有指令都是由指令流水线执行
  - 指令流水线是时钟同步的时序电路

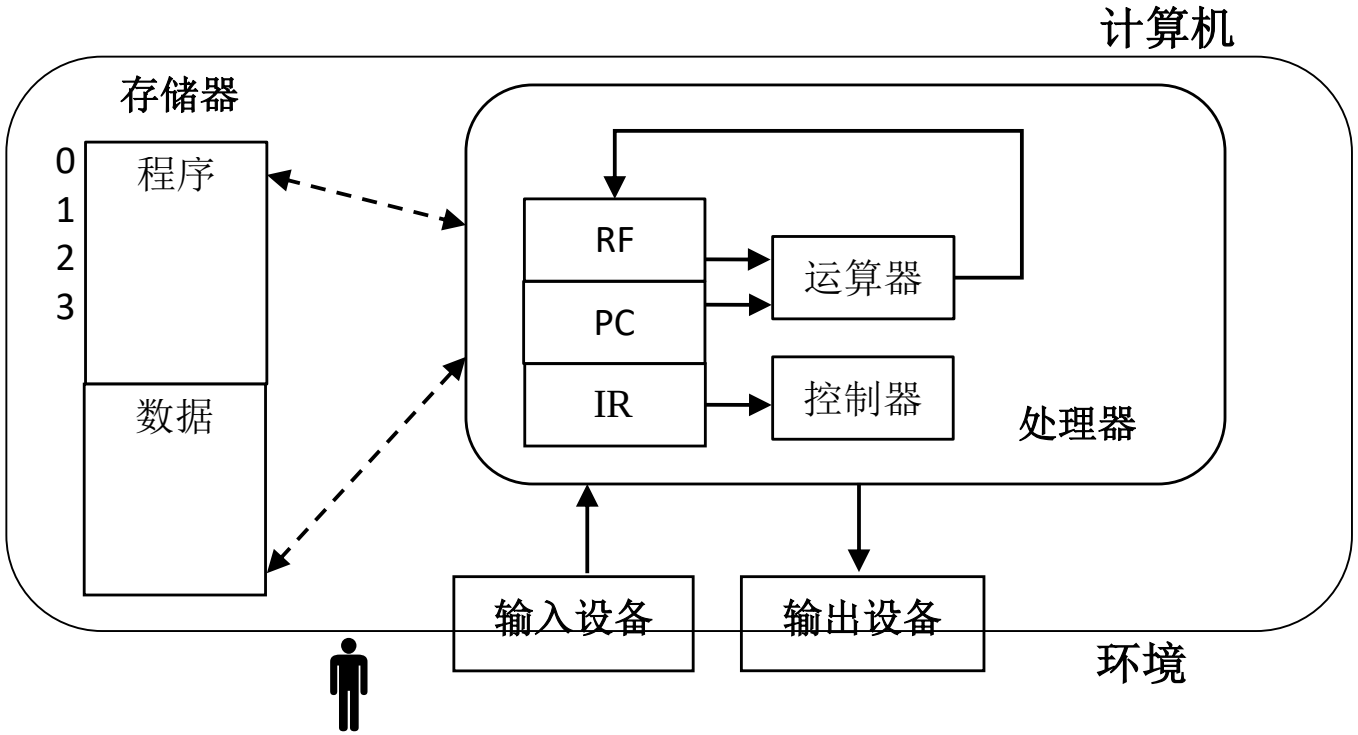




存储程序计算机（冯诺依曼模型）

- 二进制数据及其算术逻辑操作
- 计算机 = 处理器 + 存储器 + 输入输出设备
  - 处理器 = 运算器+控制器+寄存器
  - 【当代处理器的核心是指令流水线】
- 存储程序计算机（stored program computer）
- 串行执行

RF: register file
PC: program counter
IR: instruction register



软件

- 基础软件
  - 固件（firmware）：实现最基本的功能，通常“固化”在只读存储器芯片中
  - 系统软件：操作系统、编译器
  - 中间件：数据库软件、万维网服务器
- 应用软件
  - 办公软件、通信软件、行业软件、游戏软件、电子商务软件、上网软件等
- 二进制码：机器语言程序
- 源码：汇编语言或高级语言程序

自动执行与无缝衔接

- 自动执行高阶：无缝衔接
  - 扬雄周期原理
  - 波斯特尔鲁棒性原理
  - 冯诺依曼穷举原理
  - 阿姆达尔定律
- 自动执行难题尚未完全解决
  - 基本解决了单机自动执行难题
  - 在多台计算机上执行的计算过程如何自动执行？
  - 在人机物三元计算的万物互联网时代如何自动执行？
    - 网络思维（名字空间、拓扑、协议栈）
    - IEEE CS 2022报告：无缝智能

正常执行与异常处理

- 正常执行
  - 第一条指令在哪里？
  - 当前指令如何执行？
  - 下一条指令在哪里？
- 异常处理
  - 中断
    - 执行完毕当前指令，然后执行异常处理程序
  - 硬件出错
    - 立即执行一个事先设计好的异常处理程序
  - 保底异常
    - 为了做到穷举，保底异常（通常被称为machine check）覆盖其他异常没有覆盖的情况

重视瓶颈：阿姆达尔定律

- 系统性能改进受限于系统瓶颈
  - 加速比 = 1 / ((1-f)/p + f) →1/f 当 p→∞
    - “假如一个系统可以分成两部分X和Y, X+Y=1, 0≤X≤1, 0≤Y≤1。Y能够改善（即缩小它的数值），X不能被改善（即X是瓶颈）。那么，系统最多能被改善到1/X。”
  - 一台电脑使用了500 MHz主频的处理器芯片，假设X＝Y＝0.5，即程序代码只有一半可以随处理器速度的增加而改善
    - 那么，即使我们将处理器的速度提高1000倍（提到500 GHz）、1万倍、甚至无穷大，整个电脑的速度也最多只能变到1/0.5=2，即提高一倍

谢谢 Thank You

Q&A

zxu@ict.ac.cn

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY