# 一、概论思想

**永恒理念：**

1. 抽象

抽象<--解析-->具体

抽象的层次越高越好，而且它与工作效率成正比。忽略抽象之下的细节，提高效率，当问题出现时再深入细节。

2. 在脑子里不要对硬件和软件做任何区分

不管未来职业取向是计算机软件还是硬件，两者都懂必然会使你更强，更有效率。从两方面思考必然会使其更容易。

**重要思想：**

1. 所有计算机只要给予足够时间和内存，它们所能完成的计算任务是相同的。小的慢的计算机不能做的事，大的快的计算机也一样做不了。

图灵机（Turing machine）

2. 问题提出-->算法--编程-->程序--编译（compile）/汇编-->指令集结构（Instruction Set architecture, ISA）-->微结构microarchitecture-->逻辑电路-->器件

自然语言的二义性

算法：确定性definiteness；可计算性effective computability；有限性finiteness

程序：高级语言，低级语言（汇编语言，通常一种低级语言只对应一种计算机）

编译器：例子，x86的C编译器 | 汇编器：assembly，翻译汇编语言程序至ISA

指令集结构：程序和计算机硬件之间接口的一个完整定义

# 二、数据类型与运算

bit（位，比特）：称这样一个要么是1，要么是0的符号单位为一个bit，即所谓的二进制表示方法。

编码：0和1的序列，每个编码对应一个特定的值或状态。

数据类型：一种既定义了数值的表达方式（或编码方式），又定义了相关的操作方法的表达方式。

反码（1’s Complement）：将一个正数的所有bit全部取反，即得到该正数所对应的负数编码，例如“+5”表示“00101”，“-5”则为“11010”。

符号位表示法（Signed Magnitude）：以最高bit代表符号，0为正数，1为负数。

补码（2’s Complement）：取反加1，保证正负数之和为0（忽略进位），保证相邻码字之差为00001。

符号扩展（Sign-Extension，SEXT）：二进制正数前面添加任意位数的0不改变其值，负数前添加任意位数的1不改变其值。主要应用在两个不同长度的二进制数相加的场合。

溢出：同符号数相加结果大于表达范围称为溢出，符号相反不会发生溢出。

ALU：算数逻辑运算单元，Arithmetic and Logic Unit，计算机完成加法运算的机制。

位矢量（bit vector）：假设存在n个单元，可以用一个n-bit的二进制数代表这n个单元。当某个单元空闲时，我们将相应的bit清零，反之设1。这样的二进制数即为位矢量。

ASCII码：American Standard Code for Information Interchange，8-bit的编码方式，一种用于在计算机处理单元和输入/输出设备之间传递“字符”（character）的编码标准。

# 三、数字逻辑、逻辑电路、内存

MOS晶体管：metal-oxide semiconductor，金属氧化物半导体

n-MOS：栅极gate，源极source，漏极drain；栅极接入2.9V电压，则导通，反之断开open circuit。

p-MOS：工作机制和n-MOS正好相反，或称互补

CMOS电路：Complementary Metal-Oxide Semiconductor，同时存在p-MOS和n-MOS的电路。

Decision Element：决策单元，或被成为组合逻辑结构combinational logic structure，不具备记忆功能的逻辑结构（或结构单元）。

译码器：在所有输出中有且仅有一个为1，其余皆为0。主要用于解释一个二进制数，或者是传递选择信号。

操作码：opcode，每条指令的具体操作由一个4-bit二进制数决定，这个数即为操作码。

多路复用器multiplexer, MUX: 从多个输入中选择一个，并将其与输出相连。选择信号决定选择哪一个。

全加器：一种计算加法的逻辑电路，有三个输入，两个被加值，一个进位值；两个输出，一个和，一个进位值。

可编程逻辑阵列（Programmable Logic Array, PLA）：可以实现任何逻辑函数的可构建模块；如果逻辑函数的输入数目是n，需要2n个AND门，输出数目=或门数目。

逻辑完备性（Logical Completeness）：AND, OR, NOT逻辑门集合是逻辑完备的，只要有足够的三种门，就可以实现任何逻辑函数。

R-S锁存器（R-S Latch）：锁存器，触发器trigger，寄存器register有相同的含义和特性，只是强调的特点不同。可以设置和存储数据状态。

门控D锁存器：包含R-S锁存器和控制电路（两个门电路）。可以控制读写，可以控制写入信号。

可写：WE，write enable

寄存器：若干个门控D锁存器组合成的一个独立单元，每个门控D锁存器负责存储一个bit。

字节：即8-bit信息量

寻址空间（address space）：内存中可独立识别的位置总数为内存的寻址空间。16MB内存指的是包含大约224个存储位置，需要24-bit宽度的地址。

寻址能力（addressability）：每个内存位置中包含的bit数目。一般的内存都是字节寻址（byte-addressable），也就是每个内存位置存储一个字节信息。

内存：即由地址线（输入即为地址），译码器（解释地址线），字线（word line，译码器输出线），寄存器，多路复用器（字线决定哪个地址启动的逻辑单元）组成的。

组合逻辑电路（Combinational Logic Circuit）：电路的输出完全取决于电路当前的输入信号，它不关心以前电路的状态。

时序逻辑电路（Sequential Logic Circuit）：输出既与当前电路输入相关，又与之前的电路状态相关。

有限状态机（Finite State Machine, FSM）<--> 状态图

主从锁存器（master-slave flip-flop）：由两个门控D锁存器和控制读写的时钟信号构成。

# 四、计算机模型和指令集结构

冯诺依曼模型（von Neumann）：包含五个组成部分，内存（memory）、处理单元（processing unit）、输入（input）、输出（output）、控制单元（control unit）。

MAR: Memory Address Register， 内存地址寄存器

MDR: Memory Data Register，内存数据寄存器

字长（word length），字：ALU所能处理的量化大小通常被成为该计算机的字长，量化的基本单位是一个字。

外围设备（peripheral）：附属的输入输出设备

控制单元：包含指令寄存器（instruction register，IR），保存正在被执行的指令；PC寄存器（Program Counter）用来指示下一条待处理的指令，或被成为指令指针（instruction pointer）。

指令：指令是计算机执行的最小单位。指令由操作码和操作数两部分组成。操作码表示指令做什么的，操作数表示操作的对象有哪些。

指令周期（instruction cycle）：一般指令周期包含6个节拍（phase），取指令fetch，译码decode，地址计算evaluate address，取操作数fetch operand，执行execute，存放结果store result。

机器周期/时钟周期（machine cycle）：表示为主频，例，3.3GHz意味着每秒完成3.3billion机器周期。

指令种类：运算指令（operate instruction），数据搬移指令（data movement instruction），控制指令（control instruction）改变程序的执行顺序。

指令集结构包含信息：内存组织方式，寄存器组，指令集（包括操作码，数据类型，寻址模式）等信息。

通用寄存器（General Purpose Register，GPR）：和内存特性一致的临时存储空间，访问速度仅一个周期

指令集（instruction set）：定义包括操作码的集合，数据类型和寻址模式

寻址模式：包括寄存器模式（register mode）。。。

# 五、编程与调试

结构化编程（structured programming）/系统分解（system atic decomposition）：将一个大任务分解成多个独立子任务（模块），每个单元具备独立实现和运行的特性。常见的构建方法有顺序结构（sequential construct）、条件结构（conditional construct）、循环结构（iterative construct）。

EOT: End of Text，是个ASCII字符。

调试：1.赋值；2.控制指令序列的执行（单步：single-stepping，设置断点：set breakpoint）；3.内容查看

# 六、汇编语言