

Lecture 01 计算机概述 *OK!*

主要内容:

- 1、计算机的基本构成
- 2、计算机的工作原理
- 3、设计 CPU 要做的事
- 4、性能评价

难点内容: 性能评价

一、计算机的基本构成

计算机=硬件+软件

(软件控制硬件)

=硬件+程序+数据

(程序控制硬件加工处理数据)

=硬件+指令序列+数据

(指令序列控制硬件加工处理数据)

硬件=主机+输入输出设备

=CPU+存储器+输入设备+输出设备

=数据通路+控制器+存储器+输入设备+

输出设备（五大部件）

=运算器+寄存器+缓存+控制器+存储器+

输入设备+输出设备

冯·诺依曼体系结构！

二、计算机的工作原理

计算机的工作原理：编写程序、**存储程序**、执行程序

程序=指令序列+数据

指令：控制计算机完成一些基本操作的命令

比如：**加法指令**负责做加法，**减法指令**负责做减法，等等

指令系统：所有指令的集合

指令按**功能**分类：数据传送类、数据运算类、
流程控制类、其它

数据传送类指令：将什么样的数或什么地方
的数送到什么地方。

数据运算类指令：将什么样的数或什么地方
的数和什么样的数或什么地方数做什么样的运

算，结果放到什么地方。

流程控制类指令：让程序的执行流程从**当前位置**无条件或有条件地转移到另外一个位置，即**目标位置**，接下来从目标位置开始继续执行。

指令有责任提供为了完成其功能所需要的一切信息！CPU 执行指令就是根据这些信息去做该做的事。

如何提供这些信息既有**形式**问题，还有**格式**问题！

机器指令：2 进制形式的指令，所有信息全部以 2 进制形式提供

机器语言：所有机器指令构成的集合，即机器指令系统

汇编指令：文字符号形式的指令

汇编语言：所有汇编指令构成的集合，即汇编指令系统

指令的一般**格式**：由若干个**字段**构成

操作码 【操作数 1 【操作数 2 【操作数 3】】】

操作码：说明执行指令时到底要做什么，即指令的功能

操作数：执行指令时要用到的数据；可以是常

量，也可以是变量。

常量：即什么样的数据或地址，比如 34, -56,
1234

变量：即什么地方数据，可以位于寄存器，
也可以位于存储器。

地址码：变量所在的位置信息，即**地址信息**

注意：操作数如果是变量，则提供地址码。由

此，又产生一个很重要的问题：寻址方式！

寻址方式：指令提供地址信息的方式以及据此计算物理地址的方式

比如：**ADD** AX, BX

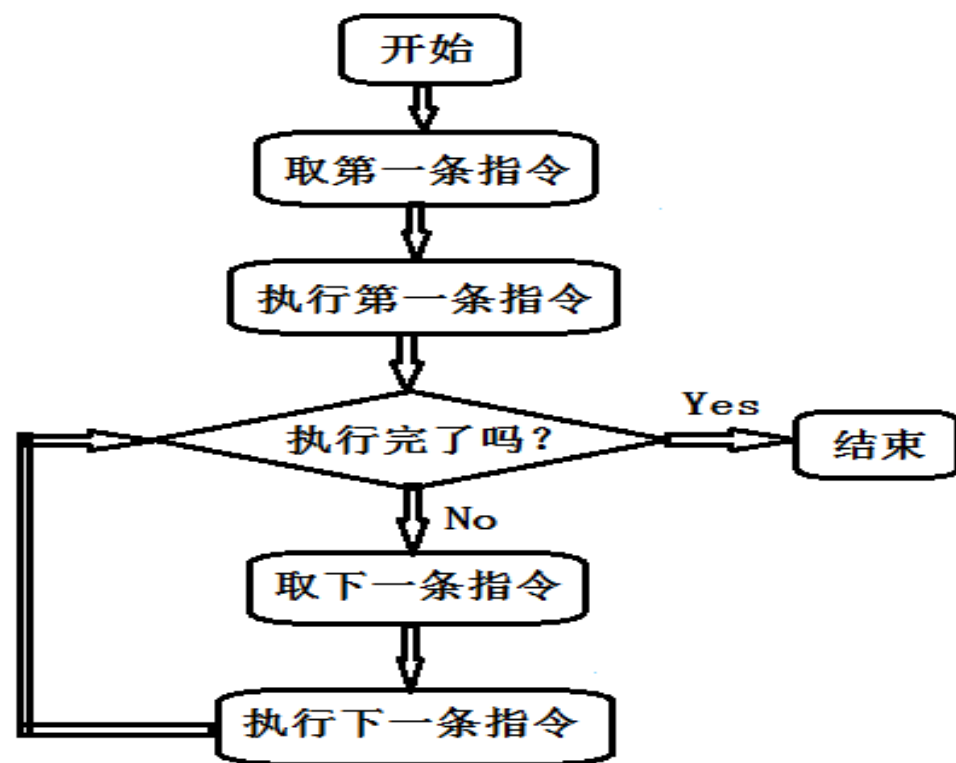
00001 001 010 XXXXX（无效位）

注意：机器指令和汇编指令并非一一对应的！
有些汇编指令没有对应的机器指令，称为**伪指令**。

编程：从指令系统中挑选合适的指令并做适当的排列以解决某个具体的问题。

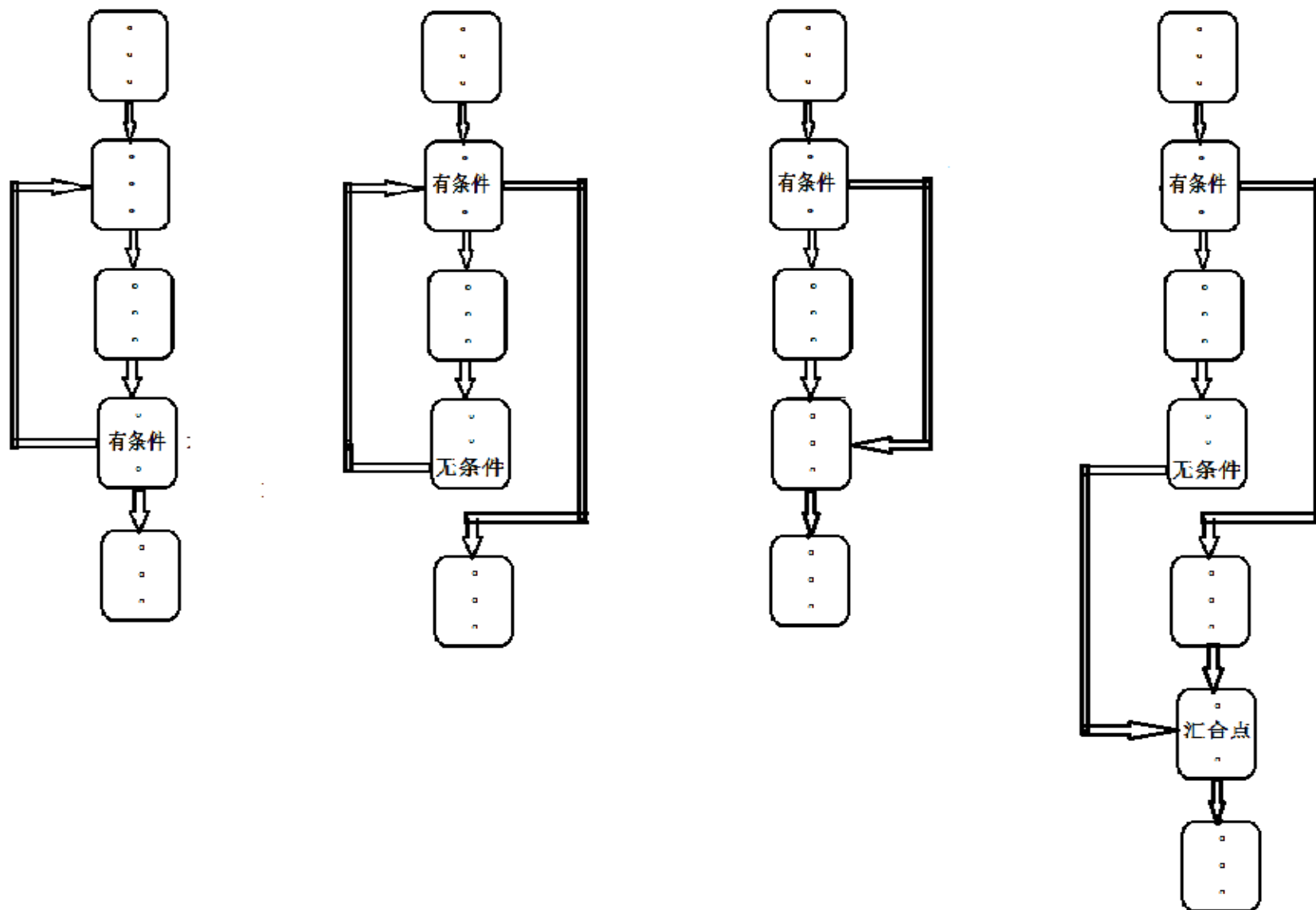
编程过程：分析问题、设计算法、根据算法写程序

程序的执行过程：首先存储程序，然后



程序总是试图顺序执行，这可以理解成一种惯性！直到碰到流程控制指令打破这种惯性，才会从当前位置转移到目标位置！

通过流程控制可以很容易构造循环和分支：



下面是 Intel X86 汇编语言程序的例子：

MOV AX, 1	MOV AX, 0	MOV AX, 1234
ADD, AX, 2	MOV CX, 5	TEST AX, 1
ADD AX, 3	ON:ADD AX, CX	JZ EVEN
ADD AX, 4	DEC CX	MOV CX, 0
ADD AX, 5	JNZ ON	JMP CON
		EVEN:MOV CX, 1
		CON:。 。 。

本课程第 2 章、第 5-7 讲重点介绍指令系统和汇编语言。

三、设计 CPU 要做的事

设计 CPU 要做的两件事：

1) 设计指令系统； 2) 实现指令系统

设计指令系统要做的事： 需要支持哪些指令？

每条指令的功能、格式、寻址、编码等！

别忘了，指令有责任提供为了完成其功能所需要的一切信息！CPU 执行指令就是根据这些信息

去做该做的事。

实现指令系统要做的事：用硬件电路实现指令系统中的每一条指令，即：指令系统中的每一条指令要做的事都要用硬件做出来，包括**取指令**。

比如：ADD AX, 1234

本课程第 4 章、第 8-10 讲告诉大家如何设计 CPU（第 2 件事）。

四、CPU 的性能计算

时钟频率： 周期性时钟信号的频率

时钟周期： 频率的倒数

程序的指令数： 静态指令数、动态指令数

静态指令数： 程序不执行时，程序中所有指令的数量

动态指令数： 程序执行时，实际执行的指令数

别忘了，程序中有分支、循环、调用等！

执行速度（吞吐率）：单位时间内执行的指令数，通常以 MIPS 为单位。比如，3 MIPS，即每秒钟执行 3 百万条指令

程序的执行时间（CPU 时间）：用户 CPU 时间、系统 CPU 时间

注意：不含等待 I/O 的时间

用户 CPU 时间: CPU 执行用户自己所写程序代码所花费的时间

系统 CPU 时间: CPU 执行系统调用程序代码所花费的时间

注意，用户程序通常会调用系统程序！

系统功能调用: 系统提供的实现某些特定功能的程序代码，普通用户很难编写（很麻烦，或者

根本写不出), 直接调用。

比如:

MOV AH, 2

MOV DL, 39H

INT 21H; 此处有一个系统功能调用

.

利用 Intel X86 系统的中断调用指令调用系

统提供的 21H 号中断服务程序的 2 号功能：在显示器上显示数字 9。

CPI: 每条指令所花费的时钟周期数，通常取平均值

IPC: CPI 的倒数，即每个时钟周期所执行的指令数，通常取平均值

基本公式:

CPU 时间 = CPU 时钟周期数 * 时钟周期

CPU 时钟周期数 = 程序的指令数 * CPI

CPU 时间 = 指令数 * CPI * 时钟周期

例题: 用一台 4GHz 处理器执行标准测试程序,
其指令分配与所需时钟周期数如下:

指令类型	指令条数	时钟周期数
------	------	-------

整数运算	4.5×10^6	1
数据传送	3.2×10^6	2
浮点运算	1.5×10^6	2
控制转移	0.8×10^6	2

(1) 有效 CPI 是多少？

(2) MIPS 速率是多少？

(3) 程序的执行时间是多少？

参考解答：

(1) 利用各类指令的条数对各类指令的 CPI 做加权平均

$$\frac{4.5 \times 10^6 \times 1 + 3.2 \times 10^6 \times 2 + 1.5 \times 10^6 \times 2 + 0.8 \times 10^6 \times 2}{4.5 \times 10^6 + 3.2 \times 10^6 + 1.5 \times 10^6 + 0.8 \times 10^6} = \frac{4.5 \times 1 + 3.2 \times 2 + 1.5 \times 2 + 0.8 \times 2}{4.5 + 3.2 + 1.5 + 0.8} = 1.55$$

(2) 执行速度 = IPC * 频率 = 频率 / CPI

$$\frac{4 \times 10^9}{1.55} \approx 2.58065 \times 10^9 \text{ IPS} = 2580.65 \text{ MIPS}$$

(3) 执行时间=CPI*指令数*周期

$$\frac{1.55 \times (4.5 \times 10^6 + 3.2 \times 10^6 + 1.5 \times 10^6 + 0.8 \times 10^6)}{4 \times 10^9}$$

$$= \frac{1.55 \times 10^7}{4 \times 10^9} = 0.003875s = 3.875ms$$