



清华大学
Tsinghua University

Project 点评及课程总结

刘卫东

计算机科学与技术系

总体情况



实验分组

- 146人选课，共有46+3组

完成情况

- 几乎全部完成了基本要求，并有一定的扩展功能
 - 中断、FLASH、VGA、旁路、分支预测等
- 其中，有一些组有特色鲜明的扩展
 - 3组同学选择了与软件工程联合的实验
 - 可运行的教学计算机系统（硬件调试环境、网络）
 - 分时、双机通信
 - 性能测试
- 不足之处
 - 课程压力大
 - 时间安排欠合理

设置实验的思路



- ✿ 建立计算机系统概念
- ✿ 培养硬件设计调试能力
- ✿ 巩固基本概念和理解基本理论
- ✿ 硬软件相结合
- ✿ 层次计算机的概念
- ✿ 培养工程能力和综合能力
 - ✦ 分析问题
 - ✦ 解决问题
 - ✦ 表达
 - ✦ 领导力和沟通



清华大学
Tsinghua University

课程总结

计算机能做什么？

刘卫东

计算机科学与技术系



科学的力量取决于 大众对它的了解 —— 培根



计算机的力量取决于 大众对它的应用

——张钹



计算机的应用取决于 我们对它的掌握



学习目的

- 了解计算机的组成
 - 五大组成部件
- 掌握计算机的运行原理
 - 计算机为什么能执行高级语言程序
- 了解现代计算机中的一些核心技术
 - 流水、Cache、虚拟存储
- 提高编程能力
- 培养设计计算机的技能
- 成为计算机科学家、计算机专家



学习目标

- 掌握单CPU计算机的完整硬件组成
 - ▣ 基本工作原理
 - ▣ 内部运行机制
 - ▣ 建立完整计算机系统概念
- 了解计算机系统的新发展
- 达到能独立设计一台完整计算机的水平
 - ▣ 硬件、软件齐全
 - ▣ 功能基本完整
- 知识和能力两方面都提高

信息技术 (Information Technology)

- * 计算技术 (Computation) — 计算机
- * 通信技术 (Communication) — 通信机
- * 网络技术 ——

计算机网络 = 计算 + 通信

(Network = Computation + Communication)



计算机的诞生

1943 电子器件的应用（电子管）

1946 ENIAC 电子计算机

1949 EDSAC 存储程序计算机

1953 IBM 701（电子管）

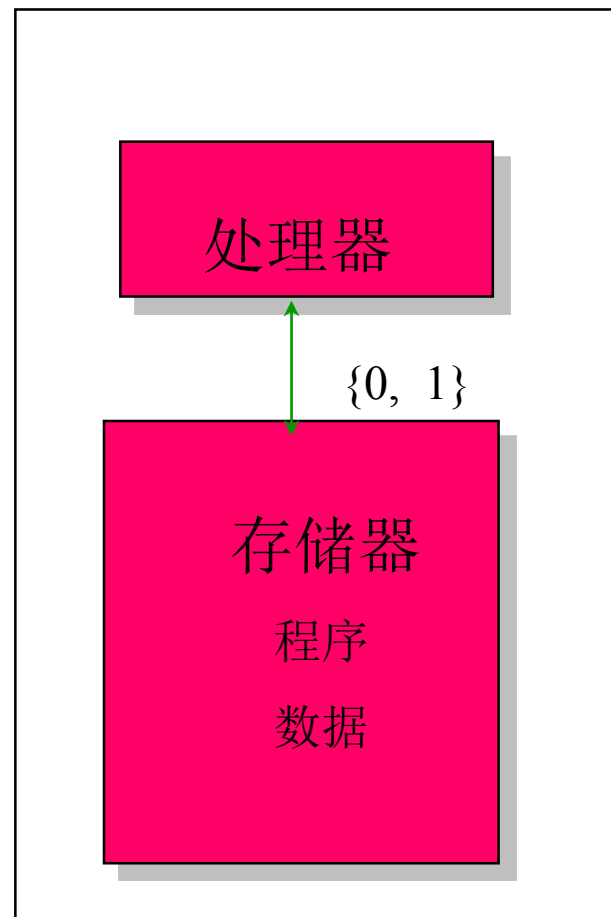
1960 晶体管计算机

1965 IBM 360（集成电路）

* 冯诺曼(von Neumann)结构

按地址存储 数据和程序

的串行计算结构



计算机（理论上）能干什么？



对 $\{0, 1\}$ 进行以下基本运算：

$+$ ： $1+1=0^*$, $1+0=1$ ($0+1=1$) , $0+0=0$

$-$ ： $1-1=0$, $1-0=1$, $0-1=1^*$, $0-0=0$

\wedge ： $1 \wedge 1=1$, $1 \wedge 0=0$ ($0 \wedge 1=0$) , $0 \wedge 0=0$

\vee ： $1 \vee 1=1$, $1 \vee 0=1$ ($0 \vee 1=1$) , $0 \vee 0=0$

$-$ ： $-1=0$, $-0=1$

如此简单的运算 能 解决复杂的问题？



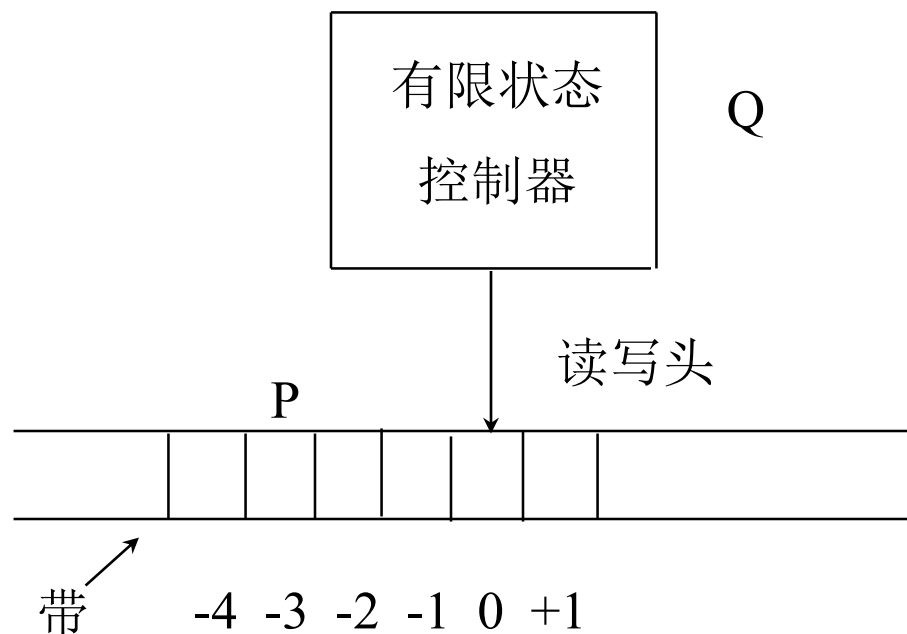
图灵机 (Turing Machine)

确定型图灵机 (1932)

- * 有穷符号集 P
- * 有穷状态集 Q
- * 转移函数

$F: Q \times P \rightarrow$

$Q \times P \times \{-1, +1\}$



通用机 (Universal Machine) 概念

- “computable numbers”

图灵可计算——

在有限机械步中可完成的计算

- “computational complexity”

计算复杂性：指数（与问题规模的关系）

多项式

计算机能做什么？



- ❖ 文字处理、科学计算、通讯，社会生活的方方面面
- ❖ 只能执行指令系统中的指令，完成规定的功能
- ❖ 只能完成二进制算术加法运算和逻辑运算
- ❖ 只能完成逻辑运算

计算机为什么能完成这些工作



✚ 将任务分解成算逻运算的组合

- ▣ 程序设计和算法

- ▣ 好处：固化操作，能快速复制

- ▣ 缺点：并不是所有任务都能分解成算逻运算的组合

✚ 自动快速执行算逻运算

- ▣ 比人脑运算要快

- ▣ 不足：受时间、空间限制

✚ 接收指令，并输出结果

- ▣ 存储程序

- ▣ 输入输出系统和设备



怎么完成

❖ ALU

- ❖ 完成算术逻辑运算

❖ 存储器

- ❖ 存储程序和数据

❖ 输入设备

- ❖ 人机交互

❖ 输出设备

- ❖ 人机交互

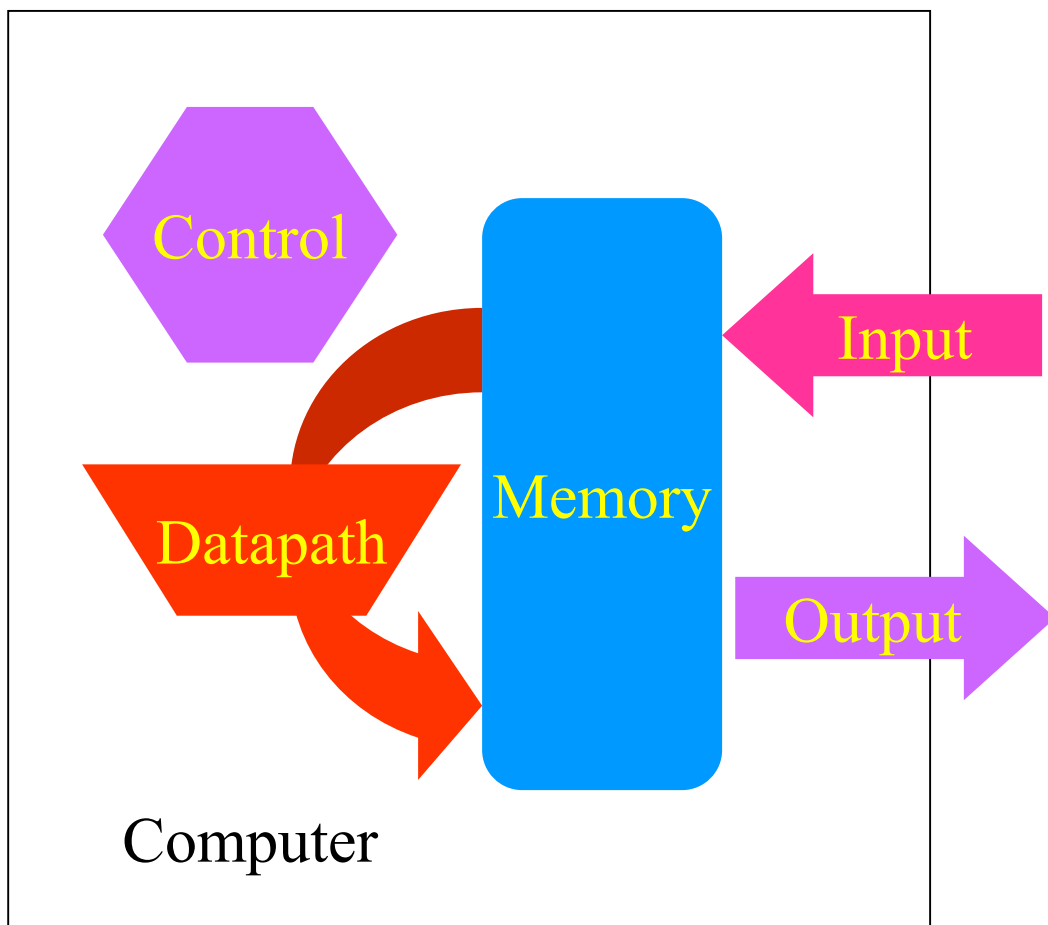
❖ 总线

- ❖ 各部件之间连接和数据交换

❖ 控制器

- ❖ 自动、连续完成

什么是计算机？



- ✚ Datapath: 根据指令要求和功能完成算术和逻辑运算。
- ✚ Control: CPU的组成部分，它根据程序指令来指挥datapath, memory以及I/O运行，共同完成程序功能。
- ✚ Memory: 存放运行时程序及其所需要的数据的场所。
- ✚ Input: 信息进入计算机的设备，如键盘、鼠标等。
- ✚ Output: 将计算结果展示给用户的设备，如显示器、磁盘、打印机、喇叭等。

❊ 数据表示

- ❑ 数值数据表示
- ❑ 逻辑数据表示
- ❑ 字符数据表示
- ❑ 检错纠错码

❊ 算术运算

- ❑ 加法运算
- ❑ 减法运算
- ❑ 乘法运算
- ❑ 除法运算
- ❑ 浮点数运算

❊ 逻辑运算

- ❑ 逻辑与、或、非

❊ 电路实现

❖ 二进制数据表示

- ❖ 数值数据和逻辑数据统一
- ❖ 字符和数值统一
- ❖ 指令和数据统一
- ❖ 最容易实现

❖ 补码数据表示

- ❖ 减法和加法统一
- ❖ 乘法和加法统一（加法、移位）
- ❖ 除法和加法统一（加、减和移位）

❖ 检错纠错码

- ❖ 奇偶校验
- ❖ 汉明校验

算术运算和逻辑运算



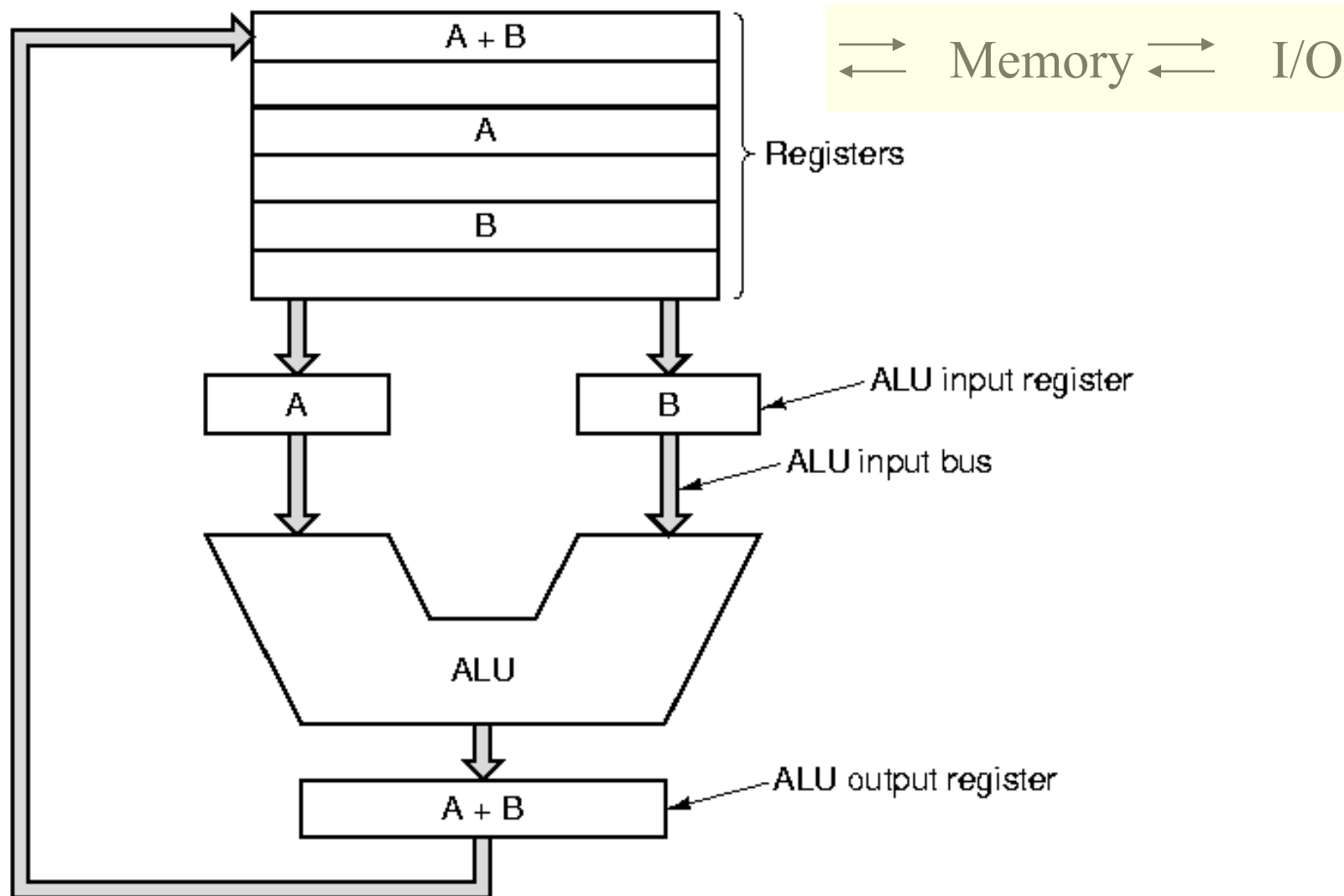
- ⊕ 完成算术运算
 - ⊞ 加、减、乘、除
- ⊕ 给出运算结果
- ⊕ 给出结果状态
 - ⊞ C、Z、V、S
- ⊕ 浮点数据的算术运算
- ⊕ 根据标志位进行逻辑判断
- ⊕ 指令中的逻辑判断

电路实现



- ⊕ ALU
- ⊕ 移位器
- ⊕ 寄存器组
- ⊕ Q寄存器
- ⊕ 多路选通电路
- ⊕ 译码器

Datapath



运算器



✿ 算术逻辑运算

✿ 数据表示

- ❑ 原、反、补码

- ❑ 检错纠错码

- ❑ 浮点数据表示IEEE754

✿ 数据运算

- ❑ 补码加、减运算

- ❑ 原码一位乘除运算

- ❑ 浮点数算术运算

✿ 电路实现

❖ 自动执行指令

- ❑ 将指令系统的指令转换为完成指令功能对应的控制信号
- ❑ 分步骤执行指令
- ❑ 得到下一条指令的地址

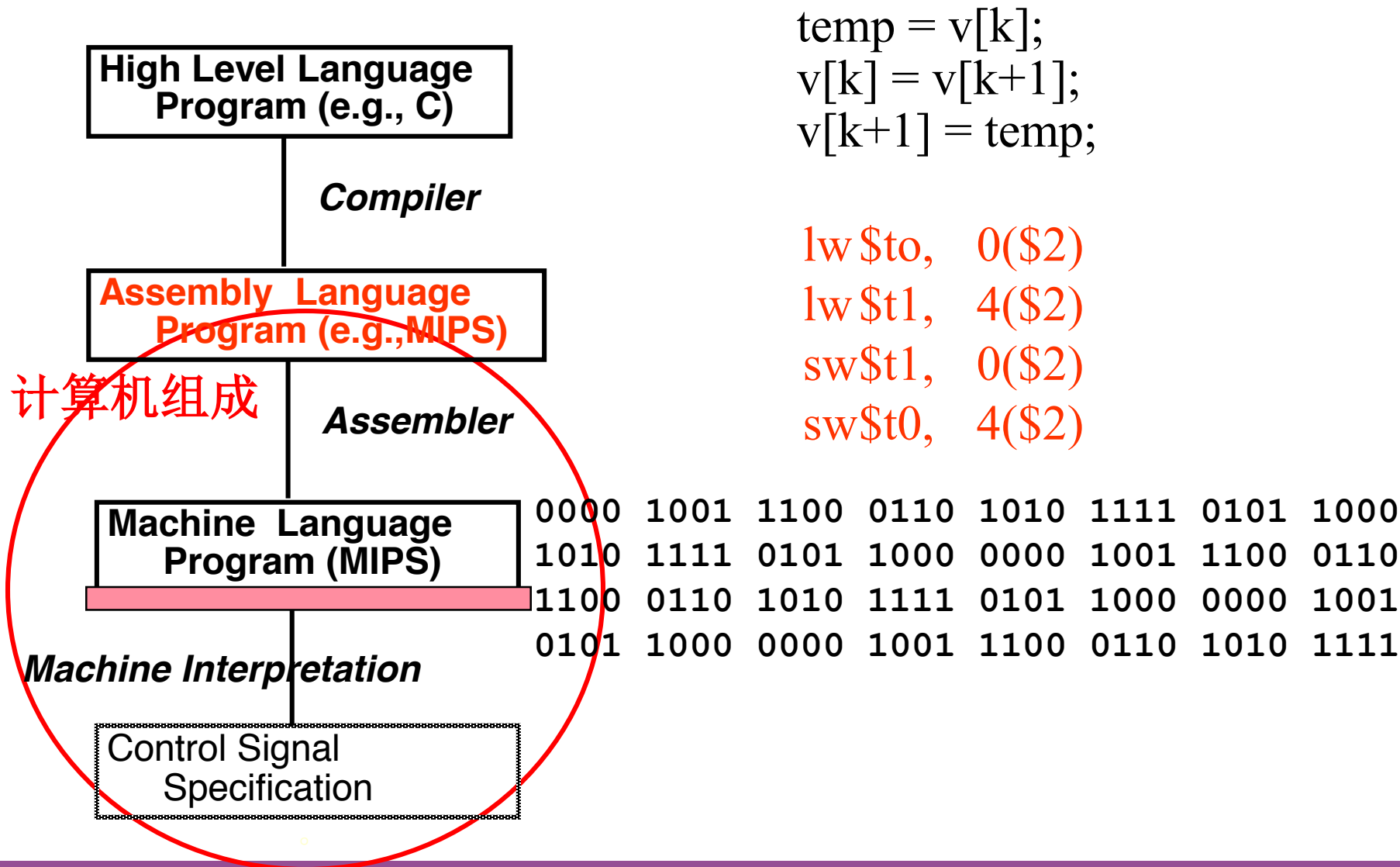
❖ 连续执行指令

- ❑ 下地址
- ❑ 节拍
- ❑ 段间寄存器

❖ 提高指令执行速度

- ❑ 指令流水
- ❑ 多流水线设置
- ❑ 多核
- ❑ 并行计算机

不同层次的程序



指令系统和指令格式



指令和指令系统

- 指令是指指挥计算机各部件完成规定功能的命令
- 计算机系统的全部指令的集合称为指令系统

操作码

- 指明指令需要完成的功能
- 对指令进行译码的输入

操作数地址

- 指明指令处理的对象

指令格式

- 如何在指令字中安排操作码和操作数地址

寻址方式



立即数寻址

▣ 常量

寄存器寻址

直接寻址

间接寻址

变址寻址

堆栈寻址

基地址寻址

▣ 段表

多周期控制器组成



①**程序计数器PC**：存放指令地址，有自增或接收新值功能。

②**指令寄存器IR**：存放指令内容：操作码与操作数地址。

主脉冲源与启停控制线路，根据需要给出主脉冲信号。

③**指令执行步骤标记线路**：指明每条指令的执行步骤。

④**控制信号记忆或产生线路**：给出计算机各功能部件协同运行所需要的控制信号。

各部件包括

运算器部件

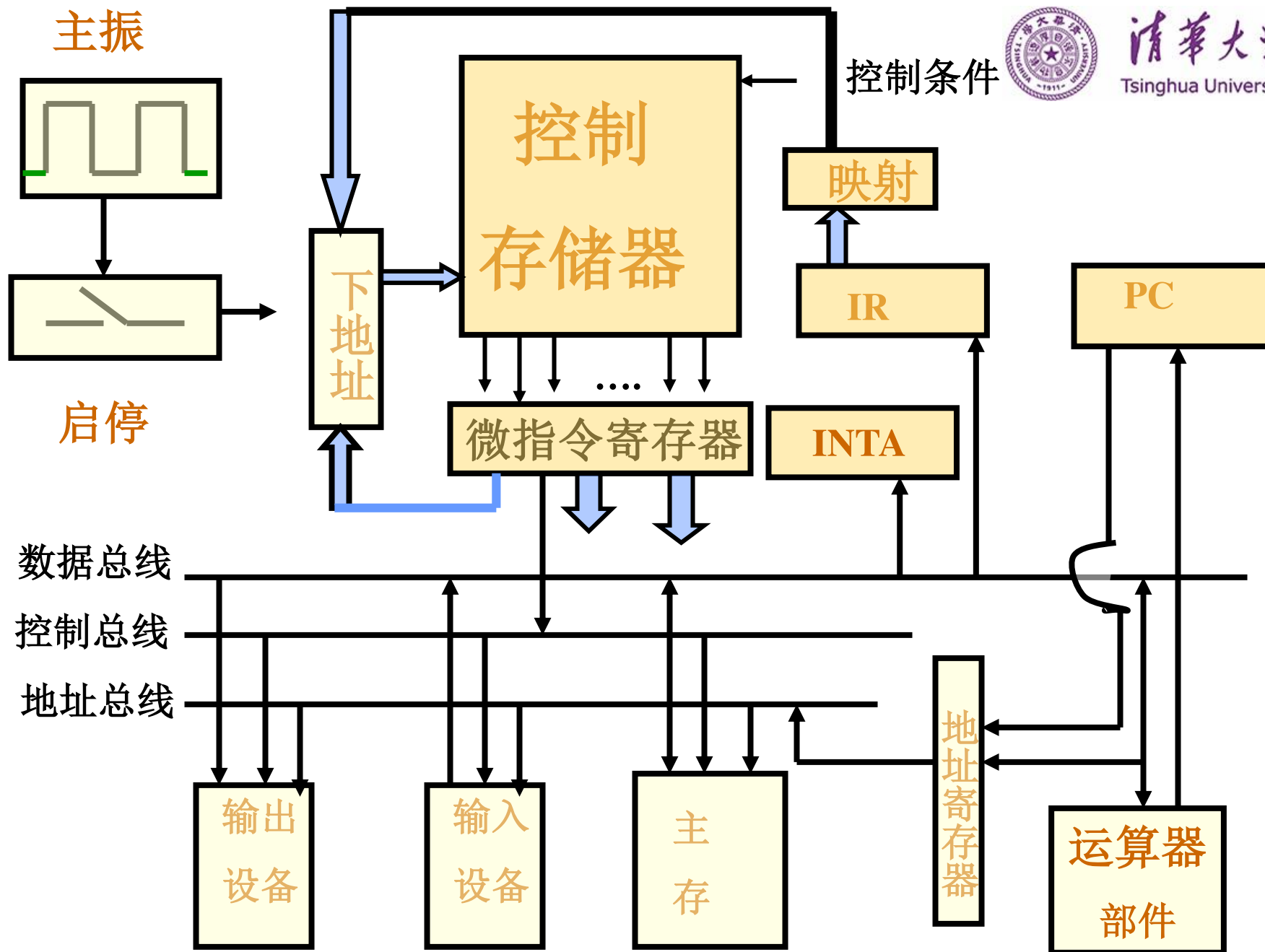
主存储器部件

总线及输入/输出接口(输入/输出设备)

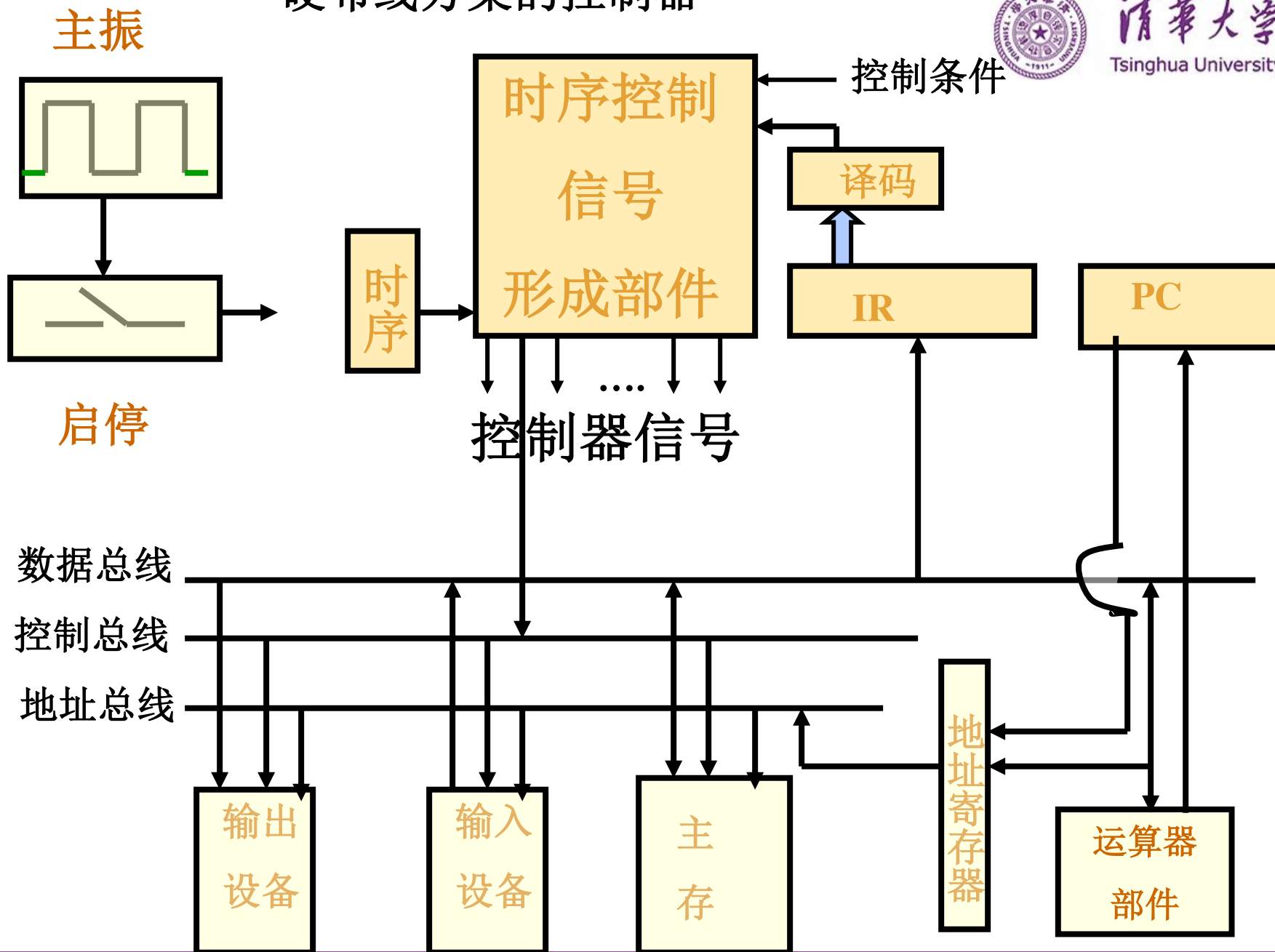
也包括

控制器部件

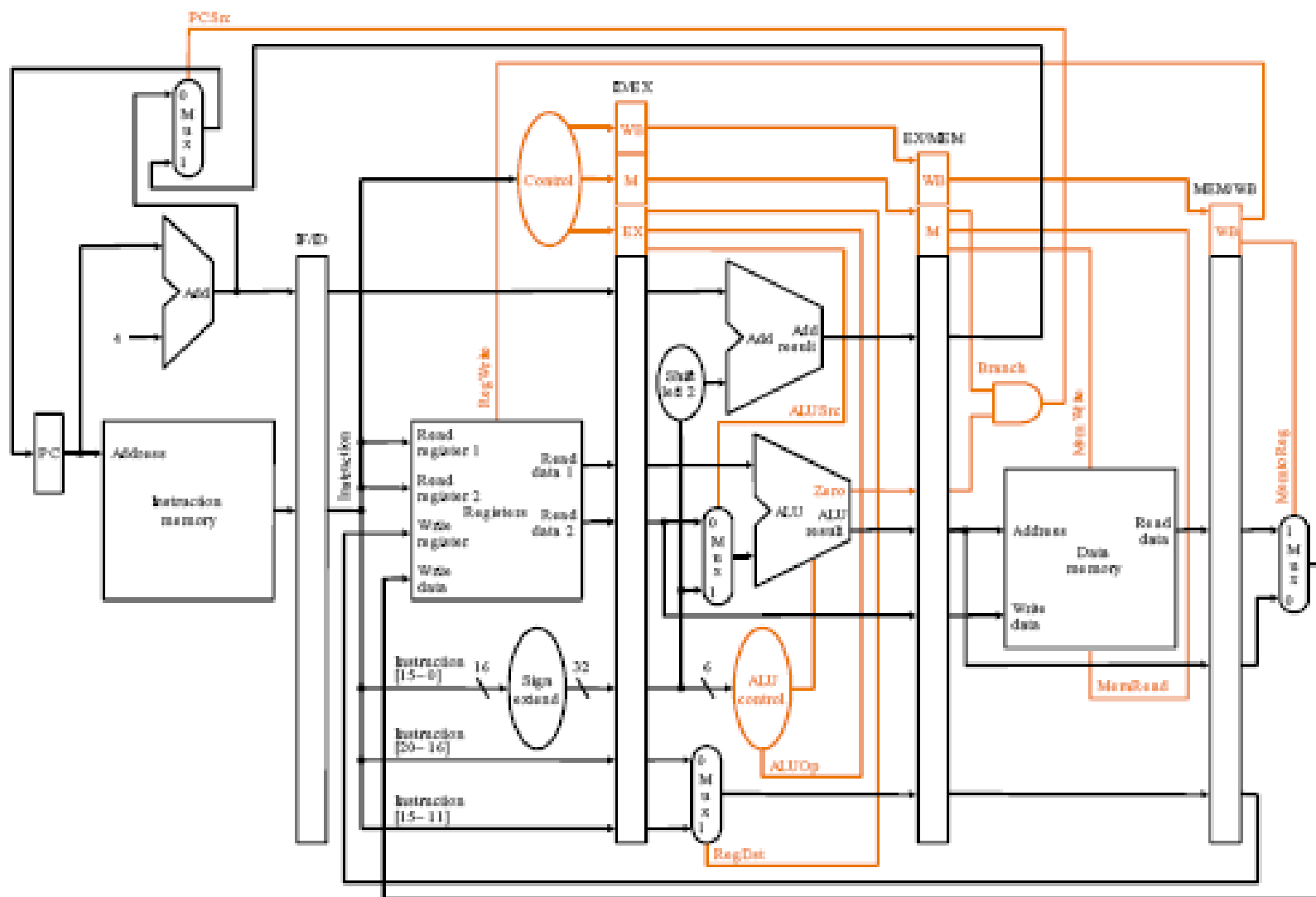
设计中的难点，在于解决对运算器、控制器的控制



硬布线方案的控制器



支持流水的CPU



- ⊕ 下地址字段
 - ⊞ 指出下一个微操作
- ⊕ 微控存
 - ⊞ 给出全部控制信号
- ⊕ 为什么能完成?
 - ⊞ 指令系统是有限的，且指令的微操作也是有限的
 - ⊞ 存储器技术
- ⊕ 有什么好处?
 - ⊞ 扩展容易、实现简单、兼容性好
 - ⊞ 复杂的指令系统CISC

组合逻辑控制器



节拍发生器

- ▣ 标明当前微操作
- ▣ 完成微操作间的转换

控制信号生成逻辑

- ▣ 输入：节拍状态和指令操作码
- ▣ 输出：该微操作的全部控制信号

为什么能完成？

- ▣ 精简的指令系统、更强大的逻辑实现能力

有什么好处？

- ▣ 速度快、适合流水线操作

指令流水的控制



❊ 段间寄存器

- ❑ 标明当前的流水段
- ❑ 保存上一流水步骤的结果及后续控制信号

❊ 控制信号生成逻辑

- ❑ 输入：指令操作码
- ❑ 输出：该指令的全部控制信号

❊ 为什么能完成？

- ❑ 规整的指令系统、更强大的逻辑实现能力
- ❑ 精致的指令执行步骤划分

❊ 有什么好处？

- ❑ 多条指令并行执行，性能高

多周期CPU的控制器设计



- ⊗ 确定指令系统
 - ⊗ 操作码、操作数地址、寻址方式
- ⊗ 划分指令流程
- ⊗ 设计每个微操作的控制信号
- ⊗ 设计节拍或下地址
- ⊗ 设计时序、启停等其他电路

指令流水



- ➊ 指令流水的基本概念
- ➋ 指令流水中的冲突
 - ❏ 结构冲突
 - ❏ 数据冲突
 - ❏ 控制冲突
- ➌ 指令流水冲突的解决方案
 - ❏ 插入等待周期（气泡）
 - ❏ 增加资源
 - ❏ 旁路技术
 - ❏ 分支预测
 - ◆ 动态/静态
- ➍ 指令流水实现

存储器



- ❊ 处于计算机中心
- ❊ 层次存储器结构
 - ❑ 高速缓冲存储器 (Cache)
 - ❑ 主存储器 (DRAM)
 - ❑ 虚拟存储器
- ❊ 半导体存储器
 - ❑ SRAM
 - ❑ DRAM
 - ❑ FLASH
- ❊ 磁表面存储器
- ❊ 每种存储介质的存储原理、特点

层次存储器系统



程序的局部性原理

- 时间局部性

- 空间局部性

层次间应满足的原则

- 包含性

- 一致性

- ☉ 硬件实现

- ☉ 参数

- ☉ 地址映射

 - ☐ 直接映射

 - ☐ 全相联

 - ☐ 多路组相联

- ☉ 提高命中率

 - ☐ 块大小、Cache容量、替换算法

 - ☐ 写策略

虚拟存储器



- 逻辑空间到物理空间

- 虚存的管理

 - 段式管理、页式管理、段页式管理

- 段表

 - 段起始地址、段长、控制位

- 页表

 - 实页号、控制位

 - 每一个虚页均在页表中有一个表项进行说明

- 快表 (TLB)

MIPS协处理器CP0



✿ 用于处理难以用常规指令解决的问题

- ✿ 配置

- ✿ Cache控制

- ✿ 异常/中断控制

- ✿ 存储管理控制

- ✿ 其他事项

✿ 使用寄存器实现

输入输出系统和设备



❊ 控制方式

- ❑ CPU如何控制输入/输出？（输入/输出方式）

❊ 传输方式

- ❑ 使用传输通道、方式、速率等（总线、接口）

❊ 数据识别和转换

- ❑ 数/模转换、语音识别等，转换为字符、数据等计算机能识别的格式（设备）

❖ 程序直接控制（轮询）

- ❑ CPU直接使用输入/输出指令来控制外部设备

❖ 程序中斷

- ❑ 外部设备请求，CPU响应，CPU与外设并行工作

❖ 直接存储访问（DMA）

- ❑ 专用输入/输出控制器
- ❑ 独占总线/总线周期窃取

❖ 通道

- ❑ 字节多路通道
- ❑ 选择通道
- ❑ 数组多路通道

❖ 外围处理机

❁ 多设备共享的信息通道

- ❁ 地址

- ❁ 数据

- ❁ 控制

❁ 多总线系统

❁ 总线仲裁

- ❁ 主设备和从设备

- ❁ 集中仲裁、分布仲裁

❁ 总线传输

- ❁ 同步

- ❁ 异步

成组传送



对主存储器的要求

- Cache、DMA等
- SDRAM
- PCI总线

经过等待时间后，按总线时钟传送数据

- ❊ 提供主机识别（指定、找到）使用的I/O设备的支持
（为每个设备规定几个地址码或编号）
- ❊ 建立主机和设备之间的控制与通信机制
- ❊ 提供主机和设备之间信息交换过程中的数据缓冲机构
- ❊ 提供主机和设备之间信息交换过程中的其他特别需求支持

外部设备功能



❊ 完成数据的输入（和/或）输出

- ❑ 信号转换

- ❑ 数据采集

❊ 与接口进行连接

- ❑ 接口信号，电平标准等

❊ 与主机进行通信

- ❑ 通过总线进行

- ❑ 速度和方式

关于考试的说明



❊ 闭卷考试：

- ❑ 1月19日 上午8：00~10:00，
- ❑ 二教401（计21、计22、计25）
- ❑ 二教403（计23、计24、其他）

❊ 时间：2小时

❊ 考试范围：

- ❑ 教学大纲规定的范围
- ❑ 通读教材

❊ 考题形式：填空、选择、判断、简答、综合

❊ 注意事项：

- ❑ 严禁任何形式的作弊
- ❑ 不使用计算器
- ❑ 仔细审题
- ❑ 回答简洁

❊ 占总评比例：40% 或 100%