Part 1

第一章 导论

1. 计算机体系结构（architecture）：对程序员可见的系统属性，直接影响程序逻辑执行。

   计算机组织（organization）：实现结构规范的操作单元及其相互连接。

2. architecture 属性
   - 指令集
   - 表示各种数据类型的比特数
   - 输入输出机制
   - 内存寻址技术

   organization 属性 —— 对程序员透明的硬件细节
   - 控制信号
   - 计算机与外设的接口
   - 存储器使用技术

   e.g.: 是否具有某一指令：architecture 问题
   该指令如何实现：organization 问题

3. 系列机（A family of compute models）：Architecture 相同，organization 不同。

4. 结构：部件相互关联的方法
   功能：作为结构组成部分的单个独立部件的操作。

5. 本质上说，计算机执行的基本功能
   - 数据处理
   - 数据存储
   - 数据传送
   - 控制

6. 计算机的主要结构组件
   - 中央处理单元（CPU）
     - 控制单元（CU）
     - 算术逻辑单元（ALU）
     - 寄存器（register）
     - CPU 内部互连。
   - 主存储器
   - I/O
   - 系统互连（BUS）

第二章　计算机的演变和性能.

1. 冯诺伊曼机 { 思想：程序存储
核心：取指-执行.

3个功能单元 { 算术逻辑运算单元(ALU)
主存储器(memory)
控制单元(~~ALU~~ CU)
I/O设备

2、发展史：

Microelectronics微电子　Gates 门
memory cells存储位元

Vacuum Tube — Transistor — Integrated circuits — 系列机(IBM提出)
电子管　晶体管　集成电路

3. 寄存器：存储器缓冲寄存器(MBR)：将出/入存储器的字.
存储器地址寄存器(MAR)：指定从MBR写入或读出到MBR的存储器地址
指令寄存器(IR)：正在执行的8位操作码指令
程序计数器(PC)：将从存储器中获取的下一条(对)指令的地址。

累加器(AC)、乘商寄存器(MQ)、指令缓冲寄存器(IBR).

4. 取指周期(fetch cycle)：下一条指令操作码 → IR
地址 → MAR
指令 ← IBR或存储器
存储器取字 → MBR

5. 执行周期(execute cycle)：译码、执行.

6. 系列机 { 相同/相似 指令集
相同/相似 操作系统
更高速度
更多I/O端口
更大内存
更高成本

7. 性能设计/评价

(1) $$CPI = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$$

(2) $$MIPS = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

(3) $$Speedup = \frac{后性能}{前性能} = \frac{前时间}{后时间} = \frac{1}{(1-f)+\frac{f}{N}}$$

Part 2

第三章 计算机功能和互连的顶层视图.

1. 指令周期 { 取指周期 / 执行周期

2. 中断（7章 & 12章）：提高处理效率
在执行周期之后，下一个取指周期之前.

3. 互连结构（interconnection structure）
指连接各模块的通路的集合.

4. 总线（bus）：连接两个或多个设备的通信通路.

Bus { 系统总线：连接计算机主要部件（CPU, I/O, 存储器） { 数据 dbus / 地址 abus / 控制 sbus
其它

数据：宽度决定系统总体性能（关键因素）

地址：宽度决定系统能够使用的最大存储器容量.

e.g: dbus 宽 d bit; abus 宽 abus, 寻址范围 2^a

5. 仲裁：{ 集中式 { 链式：越近越先响应，快，不公平 / 计数：时间取模，公平，慢 / 单独：独立请求机制
分布式

6. 总线类型 { 专用（dedicated）
复用（multiplexed）

7. 时序 { 同步：时钟周期控制一切
异步：个体事件先后关系，一件事激发另一件事.

第四章 Cache存储器

1、存储体系结构：

CPU内部寄存器 → Cache（SRAM） → 主存（DRAM） → 固态硬盘 → 光盘、磁带

内部存储器（internal memory） 外部存储器

2、存储器的特性：{ 位置 / 容量

3、寄存器以字（word）或块（block）为单位

内存以字（word）为单位

磁盘以扇区（block）为单位

4、访问方式：{ 顺序存取（磁带） } 共享读写结构，时间不等
直接存取（磁盘）
随机存取 RAM（随机存取存储器/主存） } 时间固定
关联存取，cache（部分）

5、CPU ⇄（字）Cache ⇄（块）主存储器

6、主存储器：由2^n个可寻址的字组成，每个字有唯一的n位地址

7、映射：{ 直接映射 / 全相联映射 / 组相联映射

(1) 直接映射：主存地址 | TAG | LINE | WORD |

WORD：标识某个块中唯一的一个字/字节（w位）

LINE：标识cache行中的一个（$m=2^r$行）r位

TAG：用作标记，某一块中的第几个数据。

★在cache中，不会在同一行有两个不同的块　　寻址：寻找行→检查TAG.

★ cache行 = 主存块号 mod cache行数。

优点：简单，低开销；缺点：无法灵活，命中率低

(2) 全相联映射：主存地址 | TAG | WORD |

TAG：标识主存块

WORD：标识块中的字/字节

优点：允许每一个主存块装入cache中任意行，命中率高

缺点：需要复杂电路并行地检查所有cache标记

(3) 组相联映射：主存地址 | TAG | SET | WORD |

SET：标识cache中的某个组

TAG+SET：标识主存块

WORD：标识块中的字/字节

优点：允许每个字映射到某一组的任意行中。

8. 写策略.

(1) 写直达：所有写操作同时对主存与cache进行

缺点：产生大量通信量，引起瓶颈问题.

(2) 写回：只更新cache中数据，当且仅当该数据被置日时写回主存

缺点：部分主存无效，I/O存取只允许cache进行

第五章 内部存储器

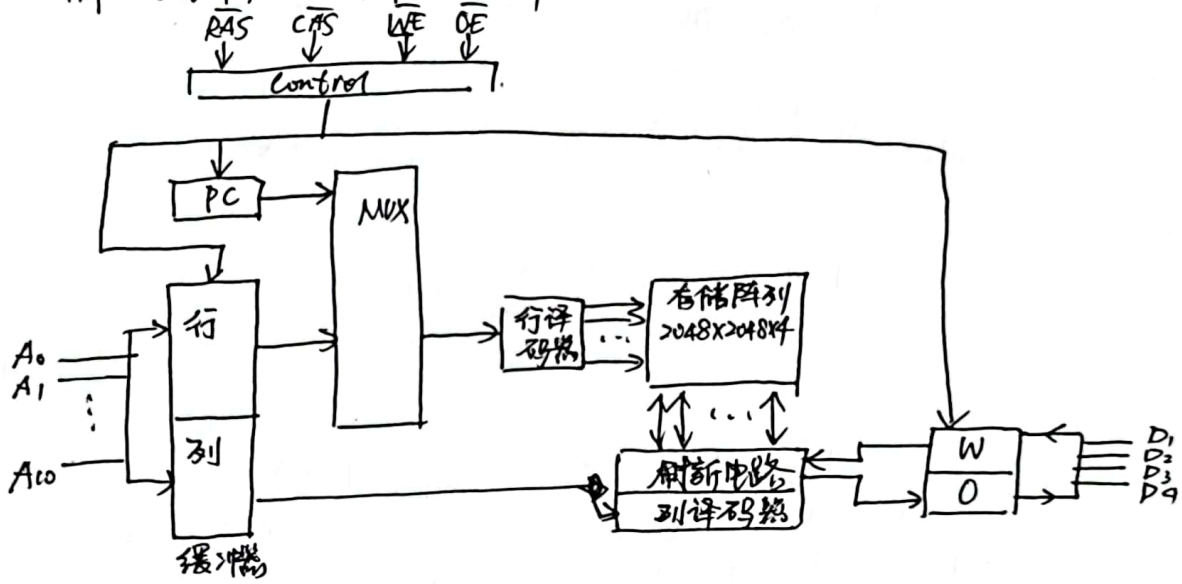
1. RAM（易失性）
   - DRAM：便宜，用于主存，电容充电存储数据，需要刷新（周期性充电）
   - SRAM：贵，用于 cache
   - 必须有直流电源供电，一旦断电，数据流失。

ROM（非易失）

ROM	不可擦除	掩膜写机制
PROM	不可擦除	电写入机制
EPROM	紫外线可擦除，芯片级	电写入机制
EEPROM	电可擦除，字节级	电写入机制
快闪	电可擦除，块级	电写入机制

2. DRAM 有 $\overline{RAS}$ 与 $\overline{CAS}$，即行选中与列选中。

$\overline{WE}$ 输入写允许，$\overline{OE}$ 输入读允许



3. 纠错：汉明码（Hamming Code）

$2^K - 1 \geq M + K$

M 为数据位数，K 为校验位位数。

每个校验位对在相应二进制序列位置编号为 1 的数据位位置操作（再异或）

第六章 外部存储器

1、 磁道：数据在盘面上被组织成一组组同心的圈 (track)

扇区：数据以扇区形式传入/出磁盘 (sector)

柱面：对于有多个磁片的磁盘，处于相同相对位置的轨道集合 (cylinder)

2、 寻道时间：使磁头对准磁道所花费的时间；

旋转延迟：等待相关扇区对齐磁头的时间

传送时间：数据传输所需时间 $T=\frac{b}{rN}$ (b：字节数；N每磁道字节数；r：转速（转/秒）)

平均访问时间 $T_a=T_s+\frac{1}{2r}+\frac{b}{rN}$ (T_s为平均寻道时间)

3、 RAID： 一组物理磁盘驱动器，在操作系统下被视为一个单一的逻辑驱动器；

数据以条带化方式分布于物理磁盘上；

冗余磁盘容量用于存储奇偶校验信息，保证磁盘可以恢复.

第七章　输入/输出.

1. I/O 技术
   - 编程式I/O
   - 中断驱动式 I/O
   - 存储器直接存取(DMA)

2. 编程式I/O：数据在处理器和I/O模块间交换，处理器执行一个使它直接控制I/O操作的程序，包括检测设备状态，发送读、写命令以及传送数据。当处理器发送命令到I/O模块，它必须等待，直到 I/O 操作完成。

   *若处理器快于 I/O 模块，则有时间浪费。

3. 中断驱动式I/O：当处理器发送 I/O 指令后，继续执行其他指令，当 I/O 模块完成 I/O 操作，中断处理器工作.

4. DMA：（CPU只在数据传送的开始与结束时参与）（DMA模块代替 CPU）

   DMA模块将处理器强制挂起，有效地窃取一个周期.
   I/O 模块与主存直接交换数据，避免处理器的影响.

5. 中断识别设备技术
   - 多条中断线（全部处理）
   - 软件轮询（轮询每个I/O模块确定中断位置）
   - 菊花链（硬件轮询，向量）
   - 总线仲裁（向量）（采用优先级方案）

6. I/O 指令寻址格式（第10、11章）
   - 分离式（isolated）
   - 存储映射式（memory-mapped）

第8章 操作系统支持.（OS）

1、主存分配
- 操作系统（常驻内存程序）
- 当前执行程序
  - 分区
  - 分页（程序员不可见）
  - 分段（程序员可见）

2、地址
- 逻辑（虚拟地址）：相对于程序起始单元的地址；（页号 + offset）
- 物理地址：主存中实际单元地址。（页帧号 + offset）

3、
- 页：一个程序的程序块
- 页表：表示进程每一页的页帧地址
- 页帧：存储器中可用的程序块。

Part 3

第九章 计算机技术

1. 原码：符号-幅值表示法：$A=\sum_{i=0}^{n-2}2^i\cdot a_i\cdot(-1)^{a_{n-1}}$ $(-(2^{n-1}-1)\sim 2^{n-1}-1)$

2. 反码：按位取反

3. 2的补码表示法：$A=-2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^i a_i$ $(-2^{n-1}\sim 2^{n-1}-1$

4. 移码：将固定偏移值加到数中存储

5. 取负：2的补码表示：① 整数按位取反 ② 当作无符号数+1

加法：直接相加

*上溢：当且仅当符号相同的数相加时可能出现（符号位反）

减法：取负相加.

6. 浮点数 IEEE格式（1位符号位，8位阶码（移码），23位有效值）

底为2.

$(-1)^S\times 1.S\times 2^{E-128}$

64位：11位阶，52位有效值

$(-1)^S\times 1.S\times 2^{E-2^{10}}$

第十章 指令集：特征与功能

1. CPU所执行的操作被执行的指令确定，称为机器指令或计算机指令。
   指令的集合为指令集。

2. 机器指令要素：
   - 操作码(operation code)
   - 源操作数引用
   - 结果操作数引用
   - 下一指令引用

3. 源和结果操作数位于：
   - 主存/虚存
   - CPU寄存器
   - 立即数
   - I/O设备

4. 指令类型：
   - 数据处理
   - 数据存储
   - 数据移动
   - 控制

5. 地址数目：大多数CPU使用的是以下一条指令地址为隐含(由PC得到)
   的单、双或三操作数指令要体。

6. 单地址指令中，第二个地址隐含，为累加器(AC)。

第十一章 指令集：寻址方式和指令格式

1、
- CPU有多种指令格式
- 计算机结构大都提供多种寻址方式

2、

寻址方式	算法	优点	缺点
立即寻址	操作数=A	无存储器访问	操作数幅值有限
直接寻址	EA=A	简单	地址范围有限
间接寻址	EA=(A)	地址范围大	多重存储器访问
寄存器寻址	EA=R	无存储器访问	地址范围有限
寄存器间接寻址	EA=(R)	地址范围大	额外存储器访问
偏移寻址	EA=A+(R)	灵活	复杂
栈寻址	EA=栈顶.	无存储器访问	应用有限.

* A = 指令中地址字段内容

EA = 被访问位置的物理有效地址

R = 指向寄存器的指令地址字段内容.

(X) = 存储器位置X或寄存器X的内容.

3. 偏移寻址
- 相对寻址（隐含引用PC寄存器） EA=A+(PC)
- 基址寄存器寻址 EA=A+(R)
- 变址 → EA=A+(R)，(R)←(R)+1. *"1"为1个偏移量.
  - 前变址 EA=(A+R)
  - 后变址 EA=(A)+(R) (R)←(R)+1

4. 指令格式
- 操作码
- (隐式/显式的)操作数
- 寻址方式

5、确定寻址位：
- 寻址方式数目
- 操作数数目
- 寄存器与存储器比较
- 寄存器组数目
- 地址范围
- 地址粒度

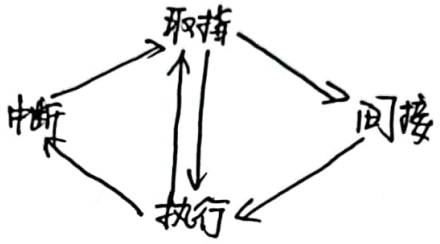
6、变长指令：寻址灵活，但复杂.

第十二章 CPU结构与功能

1、寄存器
- 用户可见寄存器
- 控制和状态寄存器
  - PC
  - IR
  - MAR — ABus
  - MBR — DBus

2、很多CPU设计包括称为程序状态字(PSW)的一个或一组寄存器.
一般含有条件码加上其他状态信息
符号、零、进位、相等、溢出、中断允许/禁止、监管.

3、指令周期



4、指令流水线技术：将指令分阶段，预处理下一条指令
*可能被中断，转移影响，造成转移损失.

5、分支指令.
- 多个指令流：复制流水线开始部分，允许其同时预处理 *竞争延迟，额外成本
- 预取分支目标：识别出条件分支时预取分支目标指令.
- 循环缓冲器：含有数条最近顺序取来指令，检索转移目标.
- 分支预测：通过分支历史表预测分支发生/不发生.

第十三章 精简指令集计算机（RISC）

1. RISC 关键特征
   1. 有限简单的指令集
   2. 大量寄存器或利用编译器优化寄存器使用
   3. 强调流水线优化

2. 窗口分为三个固定长度域
   - 参数寄存器域（继承父过程向下传递的参数）
   - 局部寄存器域（用于局部变量）
   - 临时寄存器域（用于当前过程与下一级过程交换参数/结果）

3. RISC 结构特征
   - 每个周期一条机器指令
   - 大多数操作寄存器—寄存器
   - 简单寻址方式
   - 简单指令格式

4. 提高流水线效率方式
   - 延迟转移
   - 延迟装入
   - 循环展开

第14章 指令集并行性与超标量处理器

1. 超标量 本质：在不同流水线中独立执行指令的能力
   * 允许指令以不同于原程序的顺序的次序执行.

2. 超流水：多数流水所段完成的任务只需要比时钟周期一半还少的时间.
   * 双倍的内部时钟速率允许在一个外部时钟周期中完成两个任务.

3. 对比：
   - 二者在稳定状态下只有相同的指令将同时执行
   - 程序开始和每次转移到目标时，超流水滞后于超标量.

4. 限制
   - 真实数据相关性：后指令需前指令的数据
   - 过程相关性：分支后指令相关于分支前指令.
   - 资源冲突：多个指令竞争同一资源
   - 输出相关性：更新同一寄存器
   - 反相关性：后指令破坏了前指令的值.

5. 指令集并行性：当顺序指令独立，可能通过重叠并行执行.

6. 机器并行性：处理器获取指令级并行性好处的能力程度

7. 超标量发射策略
   - 按序发射按序完成
   - 按序 乱序
   - 乱序 按序

Vector instruction 向量指令

– The basic operating object is a vector, that is, a group of numbers arranged in order 基本操作对象是向量，即有序排列的一组数

PART 4

第十五章 控制器操作

1、编写顺序：程序编写为指令的顺序
时间顺序：程序执行顺序

2、微操作：指令的最小单位，CPU操作

3、输入
- 时钟信号
- 指令寄存器
- 标志
- 控制信号

输出：
- 处理器控制信号
  - R-R
  - 激活ALU功能
- 到控制总线的控制信号
  - To 主存
  - To I/O

第十六章 微程序控制

1. 硬布线：组合电路，输入－输出－控制

2. 微程序：由微程序指定 控制逻辑，由微指令序列组成

3. 微指令 { 水平 (horizontal)：控制线的每个位都在 控制线上
垂直 (vertical)：每个执行的动作都有代码，译码器译码为控制信号.

Part 5

第十章 并行处理

1. 指令流
   - SISD：单一处理器 — 单一指令流 — 单一存储器
   - SIMD：单一指令 — 多个部件（锁步方式） — 多个存储器
   - MIMD：多处理器 — 多指令流 — 多存储器

2. SMP（对称多处理器）
   - 2+个相似功能处理器
   - 共享主存与I/O设备
   - 共享对I/O设备访问
   - 处理器功能一致
   - 由一个集成操作系统提供各处理器交互。

3. 集群化好处（cluster）
   - 绝对可扩展性
   - 增量可扩展性
   - 高可用性
   - 优异的性价比

第十八章 多核计算机