计算机组织与结构期末复习

高深

第一章

1. 计算机体系结构 (architecture) 是那些对程序员可见的系统属性
   (也就是直接影响程序的逻辑执行)

   计算机组织 (organization) 是指实现结构规范的操作单元及其(相互)连接.

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

   organization 的属性
   - 对程序员是透明的硬件细节
     - 控制信号
     - 计算机和外设的接口
     - 存储器使用技术

3. 是否有乘法指令是 architecture 问题; 而这条指令是通过特定的乘法单元, 还是重复使用系统的加法单元来实现, 是组织问题.

4. 系列机是有相同的结构, 不同的组织.

5. ☆
   - 结构: 部件相互关联的方式
   - 功能: 作为有关结构组成部分的各个部件的操作

6. 本质上说, 计算机能够完成的基本功能 (也就是 P275 所讲的指令的分类)
   - 数据处理
   - 数据存储
   - 数据传送
   - 控制

   当由连接与计算机相连的设备接收数据或向其发送数据时, 这个过程称为输入输出 (I/O), 这个设备称为 I/O 设备.

7. 与外部环境的所有连接
   - 外围设备 (本机) (连接与计算机相连的设备)
   - 通信线路

   ☆简称处理器.

8. 计算机
   - CPU
     - CU 控制器
       - 时序逻辑
       - 控制器的寄存器和译码器 } 统称程序定义语言 (microprogram)
       - 控制存储器
     - ALU
     - 寄存器 (为 CPU 提供内部存储)
     - CPU 内部互连
   - 主存储器 (P217, 又称实存储器)
   - 辅存 (磁盘)
   - I/O
   - 系统互连 (BUS)

注: CU 的 flash memory 里存着 instruction set, 于是成为可编程器件.

第2章

1. 冯诺依曼机的思想：程序存储
   结构上：取指 — 执行（控制器从主存储器中取指）
   （具体的要在后面的章节了解到每一步骤的详细！）

2. 电子管（真空管）— 晶体管 — 集成电路
   Vacuum tube　transistor　Integrated circuits
   ↓
   IBM system/360 — 提出了各型机的规范

3. 系列机 {
   - 相同或相似的指令集（结构相同）⟹ 程序可以向上兼容而不能向下移植
   - 相同或相似的操作系统
   - 更高的速度
   - 更多的I/O端口数
   - 更大的内存容量
   - 成本增加
   }

4. {
   - CPI — average cycle per instruction
   - MIPS — million instruction per second
   - 加速比 —
   }

（该去休息是否！）

第2章

1. 冯诺依曼机的思想：程序存储
   核心点：取指 — 执行（控制器从主存储器中取指）
   （具体的要在后面的章节了解到每一步有几个节拍！）

2. 电子管（真空管）— 晶体管 — 集成电路
   Vacuum tube　transistor　Integrated circuits
   ↓
   IBM system/360 — 提出了系列机的概念

3. 系列机：
   - 相同或相似的指令集（结构相同）⟹ 程序可以相上兼容而不能相下移植
   - 相同或相似的操作系统
   - 更高的速度
   - 更多的I/O端口数
   - 更大的内存容量
   - 或事增加

4. CPI — average cycle per instruction
   MIPS — million instruction per second
   加速比 —

（痕去估反是这！）

第3章

1. CPU的内部 registers
- MAR：存储地址寄存器，对下一次读或写指定内存地址
- MBR：存储缓冲寄存器，写向写到内存或从内存读取的数据
- I/O BR：用于I/O模块和CPU的数据交换
- I/O AR：用于I/O AR 指定一个特定的I/O设备

2. 指令周期
- 取指周期
- 执行周期

3. CPU中用一个称为PC的寄存器来保存待取指令的地址。

⇓ 读取存的指令

装入IR

4. 中断，详细见P_{174}和后面的章节

取了指必然执行完再响应中断，响应时将CPU状态save并压栈

5. 连接各种模块的通路的集合称为互连结构。(interconnection structures)

6. 总线(bus)是连接两个或多个设备的通信通路

系统总线：连接计算机主要部件（处理器 (CPU)、I/O、存储器）的总线
- 数据Bus
- 地址Bus
- 控制Bus

Bus
- 系统总线：连接计算机主要部件
- 其他

- 数据总线：数据总线宽度（线的数目）是决定系统总体性能的关键因素。
- 地址总线：地址总线宽度决定了系统能够使用的最大的存储器容量，
- 控制总线

- 高位选择模块
- 低位选择具体的存储单元
- 示是I/O端口
  - 统一编址
  - 独立编址

例子："32位机"
- 地址寄存器32位
- 数据寄存器32位
- 字长32位
- 总线宽度32位，寻址范围2^{32}
- 最大内存 $2^{32} \times 32$

（2^{32} ⇐ 地址总线；32 ⇐ 数据总线）

- 数据总线宽度 d bit
- 地址总线宽度 a bit，寻址范围2^a。

一个字
- 指令
- 数据

⇐ 计算机中存储、传送和操作的量本单位

地址的位数

⇑

最大存储器位数

⇑

地址总线宽度

"数据"
- 数据字
- 指令字

7. 多总线层次结构.(速度相近的在一起!)

① 采用多总线结构的目的?

② 局部总线 连接 处理器/CPU 和 高速缓存

PSB 连接 高速缓存控制器 连接 系统总线 连接 主存储器

③ 新时代采用了 高速总线 连接 高速缓存控制器 连接 系统总线

高速缓存控制器 连接 局部总线

8. 总线 { 专用总线 dedicated (separate data & address lines) —— 结构简单,控制电路复杂
复用总线 multiplexed (shared lines, address valid or data valid control line) —— 控制电路复杂,有限的性能

9. 总线仲裁方式 (与P173中断方式)

集中式 { 链式 —— 最vulnerable. (菊链arbiter的高优先级) 有限的性能
计数器定时查询
独立请求 —— 最快

分布式 { 特点: 1. 每个module都可以claim the bus. 2. Control logic on all modules.

集中式: 一个总线控制器(仲裁器)的设备之间的总线

分布式: 没有中央控制器.

{ BR = bus request
BS = bus busy (我总觉得是bus stop啊!)
BG = bus grant

10. The operation in general.

{ Bus request
Bus arbitration
device addressing
data transfer
Bus release

11. 内存的最大范围 = 2^地址总线 × 数据总线 不一定对!

地址总线宽度对系统容量有影响,越宽可以访问的单元越多.

数据总线越宽传送的位数越多.

12. 同步时序没有异步时序灵活,

因为同步总线上的所有设备都是按固定的时钟步走,系统不能发挥高性能设备的优势

对异步时序来说,无论设备快慢新旧,都可以共享总线.

第四章 Cache

1. 存取速度↓
   - 处理器内部的寄存器
   - Cache（高速缓存）—SRAM
   - 主存（DRAM组成）
   - 固定硬盘
   - 光盘、磁带

   处理器内部存储器（由处理器直接存取）

   外部存储器（处理器通过I/O模块存取）

2. 存储器的特性
   - 位置
   - 容量

3. 外部存储器
   - 寄存器：位为单元
   - Cache：block为单元
   - 主存（main memory）：memory units为单元

4. 数据存取方式
   - 非连续存取：磁盘
   - 连续存取：磁盘
   - 随机存取：主存、Cache
   - 关联存取：Cache（高速缓存）

   非连续、时间可变

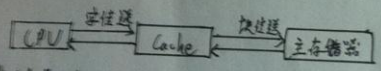
   时间固定

5. 主机内存储器
   - 寄存器
   - Cache
   - 主存储器

   主辅存储器
   - 磁盘
   - CD-ROM
   - CD-RW
   - DVD

   离线存储器
   - 磁带
   - MO
   - WORM

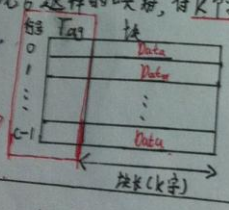
6.



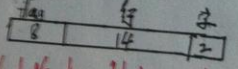
7. 主存储器：有2^n个可寻址的字组成，每个字有一个惟一的n位地址

   为了实现b这样的映射，将K个字作为一个块，有$M=\frac{2^n}{K}$个块。

Cache：由C行组成，



直接映射：主存地址：



$(i=j \bmod m)$

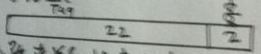
i：Cache行号　j：主存块　m：Cache行数

C.f. No two blocks in the same line (in cache) have the same tag field.

2. 在Cache中寻找主存的过程可分为

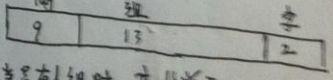
   寻找行 → 检查tag

关联映射：主存地址：



（需要复杂的电路来并行检查所有Cache行的标记）

组关联映射：主存地址：



（当只有1组时，变成关联映射）

（当1行为1组时，变为直接映射）

第5章 内部存储器

1. RAM { DRAM —便宜，用于主存。用电容充电存储数据，需要周期性地充电（刷新）
   SRAM —贵，用于Cache.

易失性，
必须有直流电源供电。
一但断电，数据将会丢失。

ROM
不需要提供电源，
不能写入数据。

2. DRAM有 $\overline{RAS}$ 和 $\overline{CAS}$。
   $\overline{RAS}$ 行选中　$\overline{CAS}$ 列选中

WE 输入（写）允许，OE 写（输出）允许

3. $2^k - 1 \geq M + K$

（M是数据的位数，K是所需校验位的位数）

第6章 外部存储器

1. 磁盘容量（判题去吧！）

P127 图6.6好好理解 磁道、扇区、柱面 (track、sector、cylinder)

2. 总的平均存取时间
   - 寻道时间(s)：使磁头对准磁道所花费的时间。
   - 旋转延迟：一旦磁道选定，磁盘控制器将处于等待状态，直到相关扇区转到磁头可读写位置。
   - 传送时间：$T=\frac{b}{rN}$

$T_{总}=T_s+\frac{1}{2r}+\frac{b}{rN}$ (b为所要传送的字节数、N为磁道所含字节数、r为旋转速率，单位为转/秒)

3. RAID
   - 工业界：Redundant（冗余的，冗余的） Array of Independent Disks 独立磁盘冗余阵列
   - 学术界：Redundant Array of Inexpensive Disks

特点：
1. RAID是一组物理磁盘驱动器，在操作系统下被视为单一的逻辑驱动器
2. 数据分布在一组物理磁盘上
3. 冗余磁盘容量用于存储奇偶校验信息，保证磁盘万一损坏时能恢复数据。

PSW寄存器 (program state word 程序状态字)　　第七章

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)

输入输出

计算机系统三大模块 (自学) { 处理器模块 / 存储器模块 / I/O模块

连接到I/O模块的外部设备常被称为外围设备 (peripheral device) 或者称外设

对 简答题:

① 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。当处理器发送一个命令到I/O模块时，它必须等待，直到I/O操作完成；如果处理器快于I/O模块时，则时间 wasted.

(处理器周期地检查I/O模块的状态，直到发现操作完成为止)

② 中断驱动式I/O:

当处理器发送一个I/O命令后，它继续执行其他指令；而当I/O模块完成工作后，I/O模块就去中断处理器工作.

③ DMA: (CPU只在数据送始开始和结束时参与) (DMA模块成为了CPU处理)

存储器直接存取

I/O模块和主存直接交换数据，处理器不参与.

2. (中断)识别设备技术 { 1. 多条中断线 / 2. 软件轮询 — 费时 (轮询每个I/O模块来确定哪个模块产生中断) / 3. 菊花链式 (硬件轮询，向量) / 4. 总线仲裁 (向量) (采用优先级方案)

优先级决定模块次序

先总第10章结合着来看！　第11章

I/O指令寻址方式 { 存储映射式 (memory-mapped) / 分离式 (isolated)

☆ 进程只在主存中执行　第8章　操作系统支持（回去做题）

1. 最重要的系统程序是操作系统

2. 主存分配 { 给操作系统（常驻监控程序）
　　　　　　当前正在执行的程序 { 分区
　　　　　　　　　　　　　　　　　分页（对程序是不可见）
　　　　　　　　　　　　　　　　　分段（对程序是可见）

3. { 逻辑地址：相对于程序开始处的单元的地址 ——→ 由页号和 offset 组成.
　　物理地址：主存中一个实际单元的地址 ——→ 由页帧号和 offset 组成

程序中的指令只包含这个

后者是虚拟地址.

4. { 页：一个程序的程序块
　　页表：表示进程每一页的页帧地址.（操作系统保存）
　　页帧：主存中可用的程序块.

段：就是 Memory 中切出来的 code, data, heap 什么的.

☆ NaN（Not a Number）：不是一个数　第9章　计算机算术.

1. sign-magnitude representation（原码）符号—幅值表示法. $A=\begin{cases}\sum_{i=0}^{n-2}2^i a_i, & a_{n-1}=0\\ -\sum_{i=0}^{n-2}2^i a_i, & a_{n-1}=1\end{cases}$ （范围 $-(2^{n-1}-1)\sim 2^{n-1}-1$）

2. ones complement 反码.

3. twos complement 补码 $A=-2^{n-1}a_{n-1}+\sum_{i=0}^{n-2}2^i a_i$ （范围 $-2^{n-1}\sim 2^{n-1}-1$）

4. 补码求负叫做"2的求补运算"(twos complement operation)

5. 上溢：两个符号相同的数相加相减，当且仅当结果的符号位出现相反才出现上溢.

（无论是否有进位，都可能有上溢！）

第10章 指令集，特征和功能

1. CPU的操作被它所执行的指令所确定，这些指令被称为机器指令或机器指令集。
   这些指令的集合称为指令集。
2. 为源和结果操作数位于如下四个范围
   - 主存或盘存
   - CPU寄存器
   - I/O设备
   - 立即数

大多数CPU使用的是以下一条指令地址为隐含的（由程序计数器得到）单地址、双地址或三地址的变体。

5. 隐含地址是被称为累加器(AC)的CPU寄存器。（单地址指令的第二个地址必须是隐含的）

第11章

1. CPU有多种指令格式。
   计算机结构都提供不只一种寻址方式。

2. 寻找操作数的途径（此函数指令集）
   - 立即寻址：1. 用于定义和使用常数或设置变量的初始值 2. 数用之前不存储
   - 直接寻址
   - 间接寻址
   - 寄存器寻址
   - 寄存器间接寻址
   - 偏移寻址 ⟹
     - 相对寻址（隐含用的是PC寄存器，局部性原理）
     - 基址寄存器寻址 ⟹ 指令：EA=A+(R)
     - 变址 ⟹ 指令：EA=(A)+(R)
   - 栈寻址（寄存器间接寻址、隐含寻址方式）

操作码	操作数

操作数也是地址字段

⇓

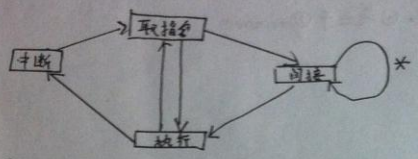
确定哪一种寻址方式
1. 不同的操作码用不同的寻址方式
2. 指令格式中的一位或几位用作方式字段

第12章 CPU结构和功能

1. 寄存器 { 用户可见寄存器
控制和状态寄存器 { PC
IR
MAR —— 与地址总线相连
MBR —— 与数据总线相连

2. 纯存CPU设计
都包括寄存器称为PSW的一个或一组Register.
一般会有条件代码加上其他状态信息.

3.



指令周期 { 取指
执行
中断
间址

P350、351、351中的图李蔚再好好看一下.

4. 指令流水 (MS下考? P352)

第13章 RISC. (大量寄存器使用可暴力美学)

1. RISC设计关键特点 (共性) { 1. 一个有限简单的指令集.
2. 大量的通用寄存器或使用编译器技术来优化寄存器的使用.
3. 强调指令流水的优化.

2. 窗口分成三个固定长度域 { 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和结果)

3. RISC结构的特征 { 每周期一条指令.
寄存器到寄存器之间的操作. (大多数指令是R ←→ R的)
简单的寻址方式.
简单的指令格式.

4. 提高流水线效率的方式 { 延迟转移
延迟装入
循环展开.

第14章 指令级并行性和超标量处理器

1、超标量：为改善标量(scalar)指令执行性能而设计的机器。

↓

本质：在不同流水线中不相关地执行指令的能力。（允许指令以不同于原程序程序的次序来执行）

2、超流水：利用了多数流水阶段所完成的任务只需比时钟周期一半还少的时间。

于是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。

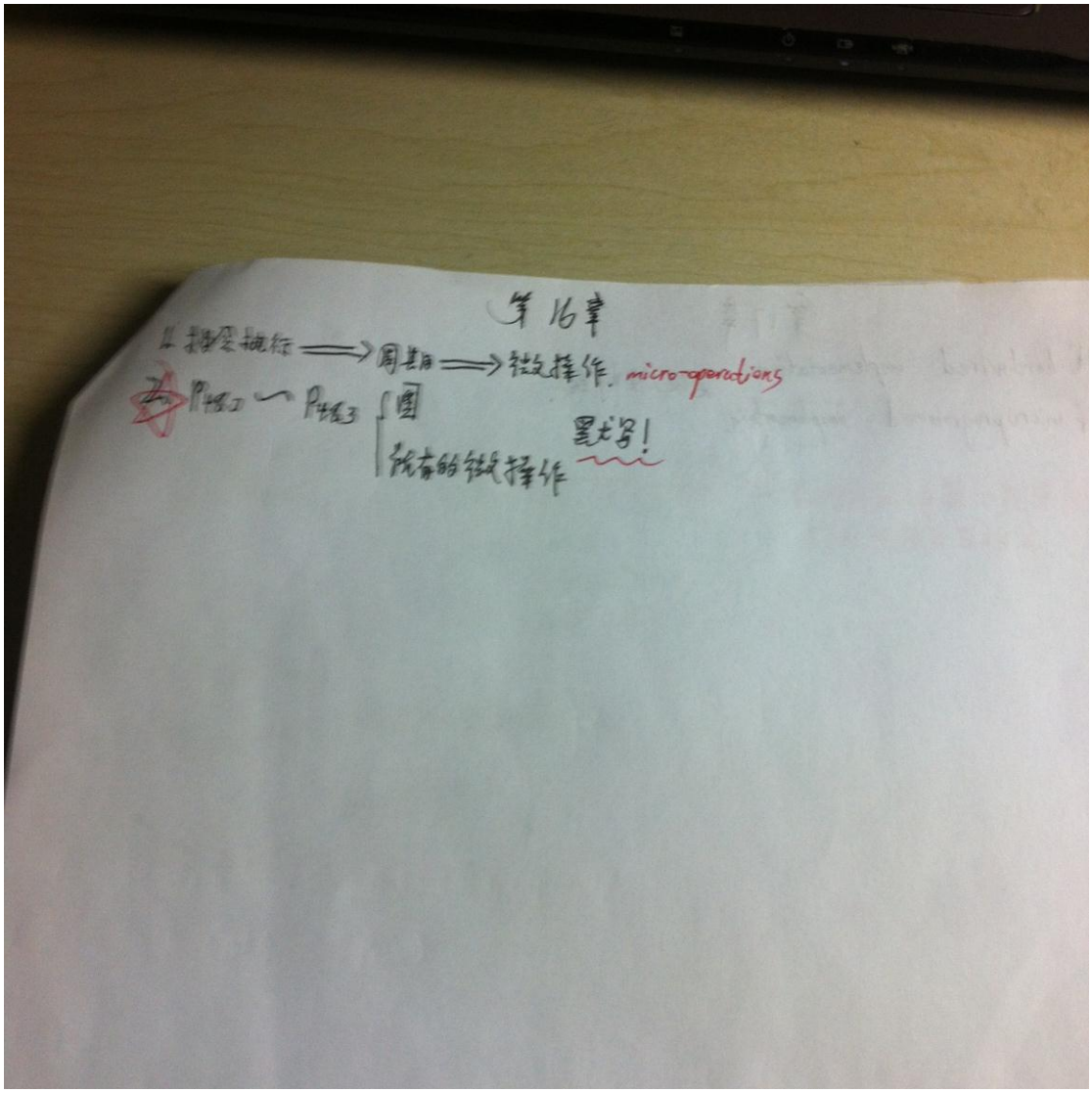
3、二者的相同 { 1、在稳定状态下具有相同的指令数在执行
2、在程序开始和每次转移引目标时，①落后于①processor

4、限制 { 1、真实数据相关性
2、过程相关性
3、资源冲突
4、输出相关性
5、反相关性

5、指令级并行性(parallelism)：当顺序中的指令是独立的，并因而能通过重叠来并行执行时，则存在...

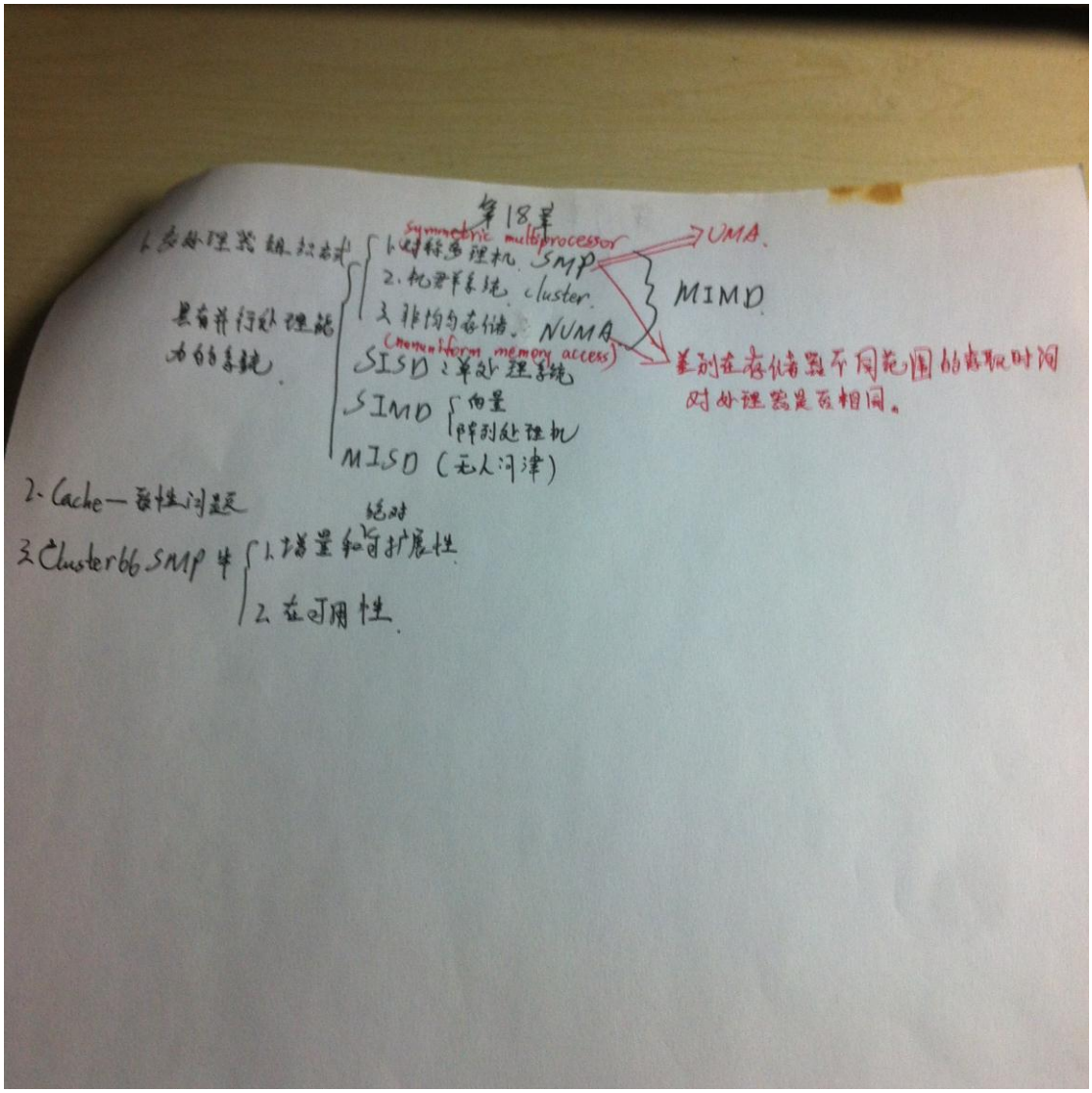
机器并行性：处理器获取指令级并行性好处的能力大小

6、超标量指令发射策略 { 1、按序发射，按序执行
2、 " ，无序 "
3、无序 " ，无序 "


第16章
1. 指令地址 ⟹ 周期 ⟹ 微操作 micro-operations
$P_{指令1}$ ~ $P_{指令3}$
图
微命令的微操作
黑体字!

第17章

人 { 硬布线 hardwired implementation
　　微程序 microprogrammed implementation } 实现控制器.


第18章
1.多处理器技术
具有并行处理能力的系统
对称多处理机
2.机群系统 cluster
3.非均匀存储 NUMA
(Non-uniform memory access)
SISD 单处理系统
SIMD 向量
阵列处理机
MISD (无人问津)
Symmetric multiprocessor
SMP
UMA
MIMD
差别在存储器不同范围的访问时间对处理器是否相同。
2. Cache一致性问题
3. Cluster较SMP
1. 绝对增量和可扩展性
2. 在可用性