

简答题

1. 计算机系统结构、计算机组成、计算机实现

- (1) 计算机系统结构主要研究软硬件功能分配, 和对软硬件界面的确定
- (2) 计算机组成是对计算机系统结构的逻辑实现
- (3) 计算机实现是对计算机组成的物理实现
- (4) 一种系统结构可以有多种组成, 一种组成可以有多种实现

2. 佛林 Flynn 分类法

- (1) 按照指令流和数据流的多倍性特征对计算机系统进行分类
- (2) 多倍性: 同时处于同一阶段的指令或数据的个数。
- (3) 单指令流单数据流 SISD: 典型单处理机
- (4) 单指令流多数据流 SIMD: 向量处理机、超标量/超流水线
- (5) 多指令流单数据流 MISD: 不存在
- (6) 多指令流多数据流 MIMD: 多处理机系统

3. 模拟、仿真

- (1) 解决软件可移植性的方法: 系列机方法、模拟与仿真、统一高级语言
- (2) 在一台现有的计算机上实现另一台计算机的指令系统
- (3) 模拟: 全部用软件实现, 用宿主机的机器语言程序解释执行虚

拟机的指令，解释程序存储在**内存**中。速度慢。

- (4) 仿真：用硬件、固件或软硬固混合实现，用宿主机的**微程序**解释执行目标机的指令。速度快，但硬件需求高。

4. Cache 存储系统、虚拟存储系统

存储系统	Cache 存储系统	虚拟存储系统
组成	由 Cache 和主存储器构成	由主存储器和磁盘存储器构成
主要目的	提高存储器速度	扩大存储器容量
特点	从系统程序员的角度看： 速度接近 Cache 的速度； 存储容量是主存的容量； 每位价格接近主存储器	从应用程序员的角度看： 速度接近主存储器的速度； 存储容量是虚拟地址空间； 每位价格接近磁盘存储器

5. 并行存储器

- (1) 并行访问存储器（同时性并行）

① 缺点：访问冲突大

- (2) 低位交叉访问存储器（并发性并行）

① 主要目的：提高存储器访问速度

② 实现方法：用地址码的低位部分区分存储体号

- (3) 高位交叉访问存储器

① 主要目的：扩大存储器容量

- ② 实现方法：用地址码的高位部分区分存储体号
- ③ 缺点：指令或数据跨越两个体时，才可能并行工作；根据局部性原理，大多数情况下只有一个存储体忙碌。

6. 并行处理机（阵列处理机）、多处理机系统

	并行处理机	多处理机系统
定义	多个 PU 按照一定方式互连，在同一个 CU 控制下，对各自的数据完成同一条指令规定的操作。	由多个独立的处理机构成，每个处理机都能独立执行自己的程序。
佛林分类法	SIMD 计算机 从 CU 看：指令串行 从 PU 看：数据并行	MIMD 计算机
并行性	同时性并行 指令级、操作级并行 并行性存在于指令内部，识别容易。	并发性并行 任务级、作业级并行 并行性存在于指令外部，在多个任务之间，识别难。
结构灵活性	专用，PE 数千个，采取固定有限的通信	通用，PE 数十个，采取高速灵活的通信
并行任务派生	把同种操作集中在一起，由指令直接启动各 PU 同时工作。	采用专门的指令来表示并发关系，一个任务开始执行时能够派生出与它并行执行的另一些任务。
进程同步	仅一个 CU，自然同步	工作进度不相同，要采取特殊的同步措施来保持程序所要求的正确顺序。存在资源分配和进程调度问题。

7. 数据相关、控制相关、结构相关

(1) 数据相关（局部相关，影响数据引用的正确性）

- ① 定义：在执行本条指令时需要用到前面指令的执行结果。
- ② 原因：指令相关、主存操作数相关、通用寄存器相关、变址相关。写后读、读后写、写后写。
- ③ 解决方法：推后分析法、设置专用路径法（数据重定向技术、变量换名技术）。

(2) 控制相关（全局相关，影响指令流动的正确性）

- ① 定义：因为程序的执行方向可能被改变而引起的相关。
- ② 原因：条件分支指令（无条件/一般条件/复合条件转移）、子程序调用、中断。
- ③ 解决方法：延迟转移技术、指令取消技术、静态/动态分支预测技术、提前形成条件码。

(3) 结构相关（结构冒险）

- ① 定义：硬件资源冲突，同一个执行部件被多条指令使用。

8. 超标量处理机、超流水线处理机

	超标量处理机	超流水线处理机
定义	每次能同时发射多条指令	一个时钟周期内能够分时发射多次指令
并行性	空间并行性 同时性并行	时间并行性 并发性并行
提高性能方法	通过增加硬件资源为代价， 增加指令条数	通过各部分硬件的充分重叠工作， 缩短流水线周期
记法	$T(m,1)$ ，期望 $ILP=m$	$T(1,n)$ ，期望 $ILP=n$

把超标量和超流水线结合在一起，成为超标量超流水线处理机 $T(m,n)$ 。

计算题

1. Amdahl 定律

$$(1) \text{ 加速比 } S_n = \frac{T_0}{T_n} = \frac{1}{(1 - F_e) + \frac{F_e}{S_e}}$$

$$(2) \text{ 改进占比 } F_e = \frac{\text{可改进部分占时}}{\text{改进前总用时}}, \text{ 改进程度 } S_e = \frac{\text{改进前用时}}{\text{改进后用时}}$$

2. CPU 性能指标公式

$$(1) CPI、IC、f、t、MIPS、T_e$$

$$(2) tf = 1, \text{ 频率 } f \text{ 的含义是 CPS, 时钟周期长 } t \text{ 的含义是 SPC}$$

$$(3) T_e = \frac{CPI \times IC}{f} = \frac{IC}{MIPS \times 10^6}$$

$$(4) MIPS = \frac{f(Hz)}{CPI} \div 10^6$$

$$(5) \text{ 等效 CPI} = \text{各指令 CPI} \times \text{频度}$$

3. CACHE 命中率和效率

$$(1) \text{ 访问效率 } e = \frac{T_1}{T} = \frac{T_1}{HT_1 + (1-H)T_2} = \frac{1}{H + (1-H)\frac{T_2}{T_1}}$$

$$(2) \text{ 平均访问周期 } T = H_1T_1 + H_2T_2 + (1-H_1-H_2)T_3$$

$$(3) \text{ 预取技术提高命中率 } H' = \frac{H + n - 1}{n}$$

$$n = \text{数据块大小 (字)} \times \text{数据复用率 (次)}$$

4. 缺失率和缺失代价

(1) 平均访存时间 $AMAT = \text{命中时间} + \text{缺失率} \times \text{缺失代价}$

(2) 存储器阻塞时钟周期 = 访存率 \times 缺失率 \times 缺失代价

(因存储访问带来的等效 CPI)

(3) 程序执行时间 = CPU 时间 + 存储器阻塞时间

5. 二维数组无冲突访问

(1) 并行存储体个数 $m = 2^{2^p} + 1 \geq n$ ，为质数，解出 p

(2) 同列相邻元素错开 $d_1 = 2^p$ 个体，同行相邻元素错开 $d_2 = 1$ 个

(3) a_{ij} 地址计算公式

① 体号地址 = $(2^p \times i + j + k) \bmod m$

② 体内地址 = i

③ k 是第一个元素所在存储体的体号，一般取 0

(4) 多体交叉访问存储器最大频宽 = 分体数 \times 单体频宽

单体频宽 = 存储字长 \div 存储周期

设计分析题

1. 循环展开实现流水线调度 书 P117

- (1) 流水线停顿→流水线调度→循环展开→循环展开再调度
- (2) 挑战数据相关
- (3) 另解：使用 SIMD（向量处理器或 GPU）、数据重命名技术
- (4) 动态调度算法：CDC 记分牌算法、Tomasulo 算法

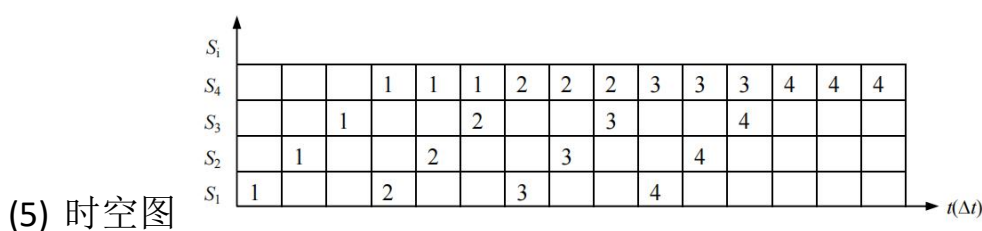
2. 流水线性能指标公式

- (1) n 为任务数， k 为流水线段数， ΔT 为时钟周期

(2) 吞吐率（任务数） $TP = \frac{n}{T_k} = \frac{n}{(n+k-1)\Delta T}$

(3) 加速比（时间比） $S = \frac{T_0}{T_k} = \frac{n \cdot k \cdot \Delta T}{(n+k-1)\Delta T}$

(4) 效率（面积比） $E = \frac{T_0}{k \cdot T_k} = \frac{n}{n+k-1}$ ， T_0 数格子



(6) $T_k = (n+k-1)\Delta T = \sum_{i=1}^k t_i + (n-1)t_{\max}$ 瓶颈段

- (7) $n \rightarrow \infty$ 时，TP、S、E 均取最大值

- (8) 瓶颈段细分、瓶颈段并联

3. 非线性流水线调度 书 P66

- (1) 禁止向量 **F**: 每一行任意两个×之间的距离, 去掉重复的
- (2) 冲突向量 **C**: 从大到小写
- (3) 状态图: 逻辑右移 (高位补 0), 移出 0 后, 和初始向量进行按位或, 得到新的状态 (向量)。注意: 任意状态移动 $m+1^*$ 位后, 都会回到初始状态。
- (4) 简单循环: 各状态最多只经过一次, 最后陷入循环
- (5) 恒定循环: 只有单个数字的循环
- (6) 最大吞吐率: $1 \div (\text{平均启动距离} \times \Delta T)$
- (7) 实际吞吐率、加速比、效率: 画时空图
- (8) 优化: 采用预留算法, 通过插入非计算延迟段实现最优调度。

4. 超标量、超流水线处理机

- (1) 超标量:
$$T(m,1) = (k + \left\lceil \frac{N-m}{m} \right\rceil) \Delta t$$
- (2) 超流水线:
$$T(1,n) = (k + \frac{N-1}{n}) \Delta t$$

解 标量流水处理机的 $ILP=1$, 连续执行 12 条指令的时空图如图 3.34 所示。

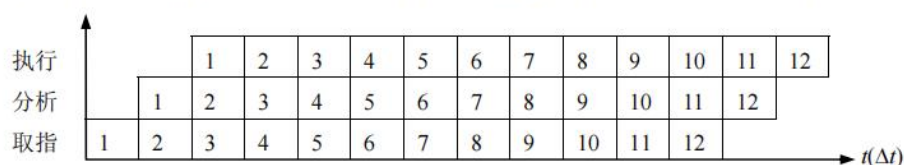
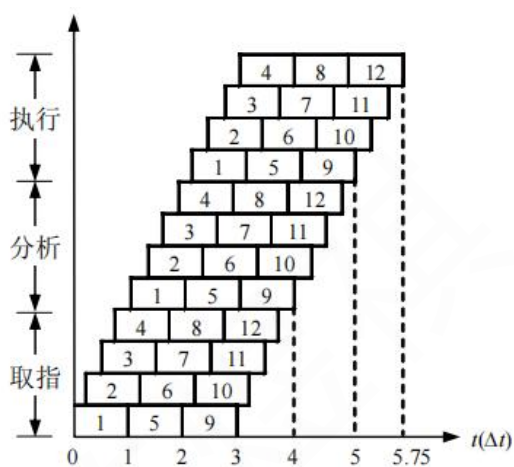
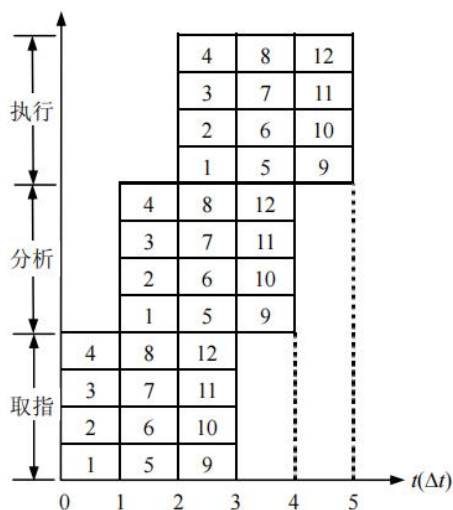


图 3.34 标量流水处理机的时空图

执行完 12 条指令所需时间为 $T_1=14\Delta t$ 。



5. 互连网络基础

(1) 互连函数（左高右低）

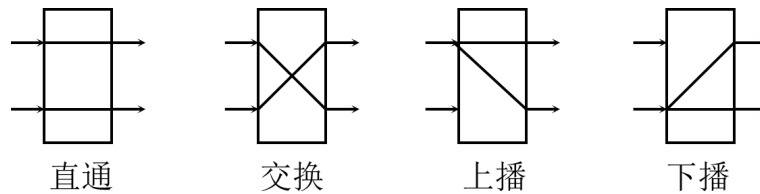
- ① 恒等置换 I : 不变
- ② 交换函数 E_k 、方体置换 $Cube_k$: 第 k 位取反
- ③ 全混洗函数 S : 循环左移一位
- ④ 逆混洗函数 S^{-1} : 循环右移一位
- ⑤ 蝶式函数 B : 最高位和最低位交换
- ⑥ 反位序函数 R : 全部顺序相反
- ⑦ 加减 2^i 置换 $PM2I$: 加减 2^i 然后取模
- ⑧ 子/超 k : 仅最低/最高 k 位变换，其余不变

(2) 交换开关

① 二功能开关（交换开关）：0 直通，1 交换

② 四功能开关：直通、交换、上播、下播

③ 级控制、单元控制、部分级控制



6. 三种互连网络（ $\log_2 N$ 级开关，每级 $N/2$ 个）

网络名称	开关类型	控制方式	级间连接	级号	寻径算法
间接二进制 N 方体网络	二功能	级控制	恒等-子蝶 2 起步-逆混洗	从低到高	$S \oplus D$: 0 直通, 1 交叉
Omega 网络	四功能	单元控制	全混洗-恒等	从高到低	终端标记寻径: 0 向上, 1 向下
STARAN 网络	二功能	级控制	恒等-Cube0 起步	从低到高	无

