

作业要认真看

## 第一章：仔细阅读 ppt，要掌握基本概念

重点：计算题目

- 例3：在一个程序中，如果浮点操作FP的比例为25%，FP的CPI为4，其他指令的CPI为1.33。浮点开平方根操作FPSQR比例为2%，FPSQR的CPI为20。如有两种方案，分别把FPSQR操作的CPI和所有FP操作的CPI均减至2，试利用CPU性能公式分析这两种方案的优劣

## 第二章：仔细阅读 ppt，要掌握基本概念

重点：指令格式设计- Huffman 编码

### Huffman编码操作码

指令序号	出现的概率	Huffman 编码法	操作码长度
I <sub>1</sub>	0.45	0	1 位
I <sub>2</sub>	0.30	1 0	2 位
I <sub>3</sub>	0.15	1 1 0	3 位
I <sub>4</sub>	0.05	1 1 1 0	4 位
I <sub>5</sub>	0.03	1 1 1 1 0	5 位
I <sub>6</sub>	0.01	1 1 1 1 1 0	6 位
I <sub>7</sub>	0.01	1 1 1 1 1 1	6 位

OpenMP 多线程并行编程，实验内容，例题

（上机题目）编写完整 OpenMP 程序，要求在程序中指定用 4 个线程执行；初始化数组 A[] 和数组 B[]，其中数组 A 和 B 各有 10240 个元素，实现：A=A+B 当数组 A 和 B 的下标是偶数时；最后输出数组 A。指出至少一种针对本程序性能优化的方法。

//参考答案：

```
#include<iostream>
#include<stdio.h>
#include<Windows.h>
#include<omp.h>
#include<time.h>
#include <stdlib.h>
using namespace std;
```

```

void OMPfun(int *A, int *B, int size) {
    int i;
    omp_set_num_threads(4);
#pragma omp parallel for
    for(i=0; i<size; i=i+2) {
        int temp=A[i]+B[i];
        A[i]=temp;
    }
}

int main() {
    int size=10240;
    int *A=new int [size];
    int *B=new int [size];
    srand((unsigned int )(time(0)));
    for(int i=0; i<size; i++) {
        A[i]=rand()%10000;
        B[i]=rand()%10000;
    }
    OMPfun(A, B, size);
    for(int i=0; i<size; ++i) {
        cout<<A[i]<<endl;
    }
    delete A;
    delete B;
    return 0;
}

```

采用 sse 进行向量化，使用动态调度，或者静态分组，减少调度次数等

### 第三章 存储系统 仔细阅读 ppt，要掌握基本概念

重点：指令格式设计- Huffman 编码

替换算法及其实现

#### 3.2 替换算法及其实现

替换算法：

主存（实存）所有页面已经全部被占用而又出现页面失效时，按照哪种算法（规则）来替换主存中某页。

注意：考虑替换算法时首先要明确所采用的相联映像方式。

采用什么替换算法：

有利于提高存储体系的性能？  
提高命中率？  
易于实现？

## Cache 的原理与性能评估

### 3.4.6 Cache-主存层次性能分析

- 块的大小、组的大小及Cache容量增大都会提高命中率
- 但Cache在调块时，CPU空等，这时希望块 较小

Cache系统的加速比与命中率的关系：

假设Cache的访问周期是 $T_c$

主存的访问周期是 $T_m$

Cache系统的等效访问周期是 $T$

则Cache系统的加速比 $S_p$ 为

$$S_p = \frac{T_m}{T} = \frac{T_m}{H T_c + (1-H) T_m} = f\left(H, \frac{T_m}{T_c}\right)$$

- 提高加速比的最好途径是提高命中率

51

## 第四章 流水线 仔细阅读 ppt，要掌握基本概念 流水线的性能分析

### 1.3 流水线的性能分析

主要指标：吞吐率、加速比和效率

#### 1.吞吐率 (Through Put)

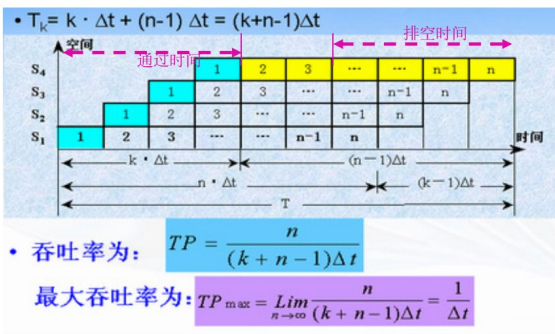
- 流水线吞吐率的最基本公式： $TP = \frac{n}{T_k}$

其中：n为任务数， $T_k$ 为完成n个任务所用的时间。

- 各段执行时间相等，输入连续任务情况下，完成n个任务需要的总时间为：

$$T_k = (k + n - 1) \Delta t$$

其中：k为流水线的段数， $\Delta t$ 为时钟周期。

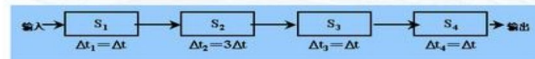


- 各段时间不等，完成n个连续任务：

吞吐率： $TP = \frac{n}{\sum_{i=1}^k t_i + (n-1) \max(\Delta t_1, \Delta t_2, \dots, \Delta t_k)}$

最大吞吐率： $TP = \frac{1}{\max(\Delta t_1, \Delta t_2, \dots, \Delta t_k)}$

- 流水线各段执行时间不相等的解决办法



设每个功能段延迟时间相等，都是 $\Delta t_0$ ，则 $T = 20 \times \Delta t_0$ ， $n = 7$ 。

吞吐率为：

$$TP = \frac{n}{T} = \frac{7}{20 \times \Delta t_0} = 0.35 \times \frac{1}{\Delta t_0}$$

$$T_{\text{非流水}} = 4 \times \Delta t_0 + 3 \times 6 \times \Delta t_0 = 34 \times \Delta t_0$$

$$S = \frac{T_{\text{非流水}}}{T_{\text{流水}}} = \frac{34 \times \Delta t_0}{20 \times \Delta t_0} = 1.70$$



$$E = \frac{7 \text{ 个任务占用的时空区}}{8 \text{ 个段总的时空区}} = \frac{34 \times \Delta t_0}{8 \times 20 \times \Delta t_0} = 0.21$$

- 冲突问题
  - 前面产生了冲突的例子
  - 流水线设计中要解决的**重要问题之一**
- 实际流水中常见的三种相关使得流水线出现冲突（第3节继续讨论）
  - **资源相关**是指当指令在重叠执行过程中，不同指令争用同一功能部件产生资源冲突时产生的
  - **数据相关**是指指令在流水线中重叠执行时，当后继指令需要用到前面的指令产生的结果时发生的
  - **控制相关**是当流水线遇到转移指令引起的。据统计，转移指令约占总指令的四分之一左右，比起数据相关，它会使流水线丧失更多的功能

## 数据检测、控制相关问题与解决方案

### 2.2 无冲突调度

- 由E.S.Davidson及其学生于1971年提出
- 例：**一条4功能段的非线性流水线，每个功能段的延迟时间都相等，它的预约表如下：
  - (1) 写出流水线的禁止向量和初始冲突向量。
  - (2) 画出调度流水线的状态图。
  - (3) 求最小启动循环和最小平均启动距离。
  - (4) 求平均启动距离最小的恒定循环。

功能段 \ 时间	1	2	3	4	5	6	7
S <sub>1</sub>	X						X
S <sub>2</sub>		X				X	
S <sub>3</sub>			X		X		
S <sub>4</sub>				X			

**解：(1)禁止向量：**预约表中每一行任意两个“X”之间距离的集合。本例中为 **(2, 4, 6)**

**(2)冲突向量：** $C = (C_m \dots C_i \dots C_2 C_1)$

其中： $m$ 是禁止向量中的最大值。

如果 $i$ 在禁止向量中，则 $C_i = 1$ ，否则 $C_i = 0$ 。

本例中  **$C = (101010)$**

**(3)构造状态图** 初始时 $S=C$

**S逻辑右移2、4、6位时，不作任何处理，**

**逻辑右移1、3、5和大于等于7时：**

S右移1位之后： $010101 \vee 101010 = \mathbf{111111}$ ，

S右移3位之后： $000101 \vee 101010 = \mathbf{101111}$ ，

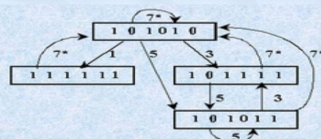
S右移5位之后： $000001 \vee 101010 = \mathbf{101011}$ ，

S右移7位或大于7位后：还原到它本身。

101111右移5位之后： $000001 \vee 101010 = \mathbf{101011}$ ，

101011右移3位之后： $000101 \vee 101010 = \mathbf{101111}$ ，

101011右移5位之后： $000001 \vee 101010 = \mathbf{101011}$ 。



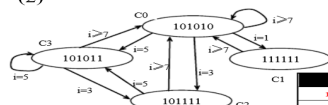
➤当移出的位为1，表示用这些启动距离向流水线输入任务会发生功能段的冲突，因此在状态图中**不做任何处理**

➤当移出的位为0，表示用这个启动距离向流水线输入任务不发生功能段的冲突，这时做“按位或”产生新的冲突向量

➤初始向量经过7次或大于7次逻辑右移后再与原来的初始冲突向量进行“按位或”运算，结果必然还是原来的初始冲突向量本身

(1) 禁止向量 $F = (6, 4, 2)$ ；冲突向量 $C = (101010)$

(2)



(3)

MAL为4；

流水线的最小启动循环为：

(1, 7) 或 (3, 5) 或 (5, 3)；

MAL最小的恒定循环为(5)。

简单循环	平均启动距离
1, 7 (C0-C1-C0)	4
3, 5 (C0-C2-C0)	5
5, 3 (C0-C3-C0)	5
3, 5, 7 (C0-C2-C3-C0)	5
5, 3, 7 (C0-C3-C2-C0)	4
3, 5, 7 (C0-C3-C2-C0)	5
5, 3, 7 (C0-C3-C2-C0)	5
3, 5 (C0-C3-C3-C0)	4
5 (C0-C0)	5
7 (C0-C0)	7

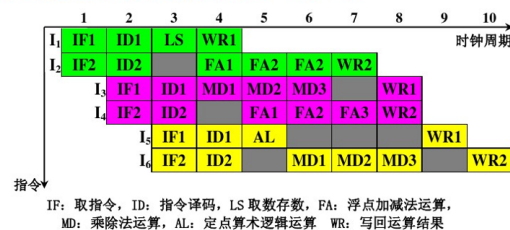
## 超标量处理机



### 1. 顺序发射顺序完成

共用10个时钟周期完成  
还有8个空闲的时钟周期

顺序发射顺序完成的指令流水线时空图



## 性能函数

- 在每个周期发射  $m$  条指令的超标量处理机上执行的时间为

$$T(m,1) = (k + \frac{N-m}{m})\Delta t$$

- 超标量处理机相对于单流水线标量处理机的加速比为:

$$S(m,1) = \frac{T(1,1)}{T(m,1)} = \frac{m(k+N-1)}{N+m(k-1)}$$

- 超标量处理机的加速比的最大值为:  $S(m,1)_{\max} = m$

### 5.3 超流水线处理机性能

- 指令级并行度为  $(1,n)$  的超流水线处理机, 执行  $N$  条指令所需的时间为:

$$T(1,n) = (k + \frac{N-1}{n})\Delta t$$

- 超流水线处理机相对于单流水线普通标量处理机的加速比为:

$$S(1,n) = \frac{T(1,1)}{T(1,n)} = \frac{(k+N-1)\Delta t}{(k + \frac{N-1}{n})\Delta t} = \frac{n(k+N-1)}{nk+N-1}$$

- 加速比的最大值为:  $S(1,n)_{\max} = n$

### 6.2 超标量超流水线处理机性能

- 指令级并行度为  $(m,n)$  的超标量超流水线处理机, 连续执行  $N$  条指令所需要的时间为:

$$T(m,n) = (k + \frac{N-m}{m \cdot n})\Delta t$$

- 超标量超流水线处理机相对于单流水线标量处理机的加速比为:

$$S(m,n) = \frac{S(1,1)}{S(m,n)} = \frac{(k+N-1)\Delta t}{(k + \frac{N-m}{m \cdot n})\Delta t} = \frac{mn(k+N-1)}{mnk+N-m}$$

- 在理想情况下, 超标量超流水线处理机加速比的最大值为:  $S(m,n)_{\max} = m \cdot n$

## 第五章 并行处理, 仔细阅读 ppt, 要掌握基本概念

### 流水线的性能分析

### 互连网络

#### 互连网络的表示方法

为了在输入结点与输出结点之间建立对应关系,

互连网络有三种表示方法:

#### (1) 互连函数表示法:

如:  $f(x_{n-1} \dots x_1 x_0) = x_0 x_{n-2} \dots x_1 x_{n-1}$

#### (2) 图形表示法



#### (3) 输入输出对应表示法

输入: 0 1 2 3 4 5 6 7  
输出: 1 0 3 2 5 4 7 6



### 5.3.2 单级互连函数

- 交换互连函数(立方体单级网络)
- 全混选单级函数 (Shuffle)
- 蝶式函数 (Butterfly)
- 反位序函数 (Bit Reversal)
- 移数函数(PM2I单级网络)

解:  $(12)_{10} = (1100)_2$

(1) Cube3, (2) PM2+3, (3) PM2-0,

(4) Shuffle, (5) Butterfly, (6) Reversal

1100最高位取反得0100, **4号处理机**

$(12 + 8) \text{ MOD } 16 = 4$ , **4号处理机**

$12 - 1 = 11$ , **11号处理机**

1100循环左移1位得到1001, **9号处理机**

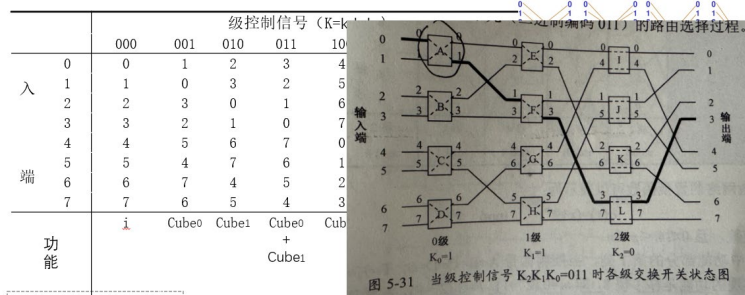
1100的最高最低位交换0101, **5号处理机**

1100的位序反过来为0011, **3号处理机**

## 多级互联网络

### 5.3.6 多级互连网络

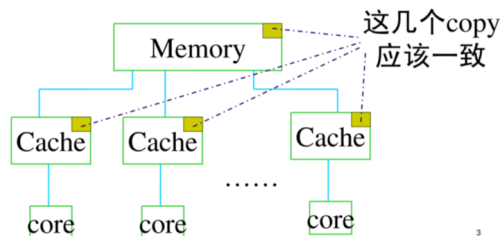
**控制:** 级控制(开关为1时交换功能, 否则为直通)



## 多个 Cache 一致性问题

多Cache一致性 详细版

### 多个Cache一致性问题 (多处理机系统、多核系统、DSM)



第六章, 第七章: 仔细阅读 ppt, 要掌握基本概念

预祝大家生活和学习一切顺利!!!