

计算机系统结构

第四讲单功能非线性流水线的调度

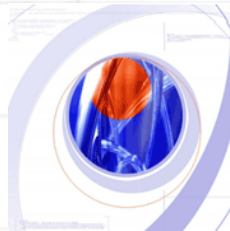
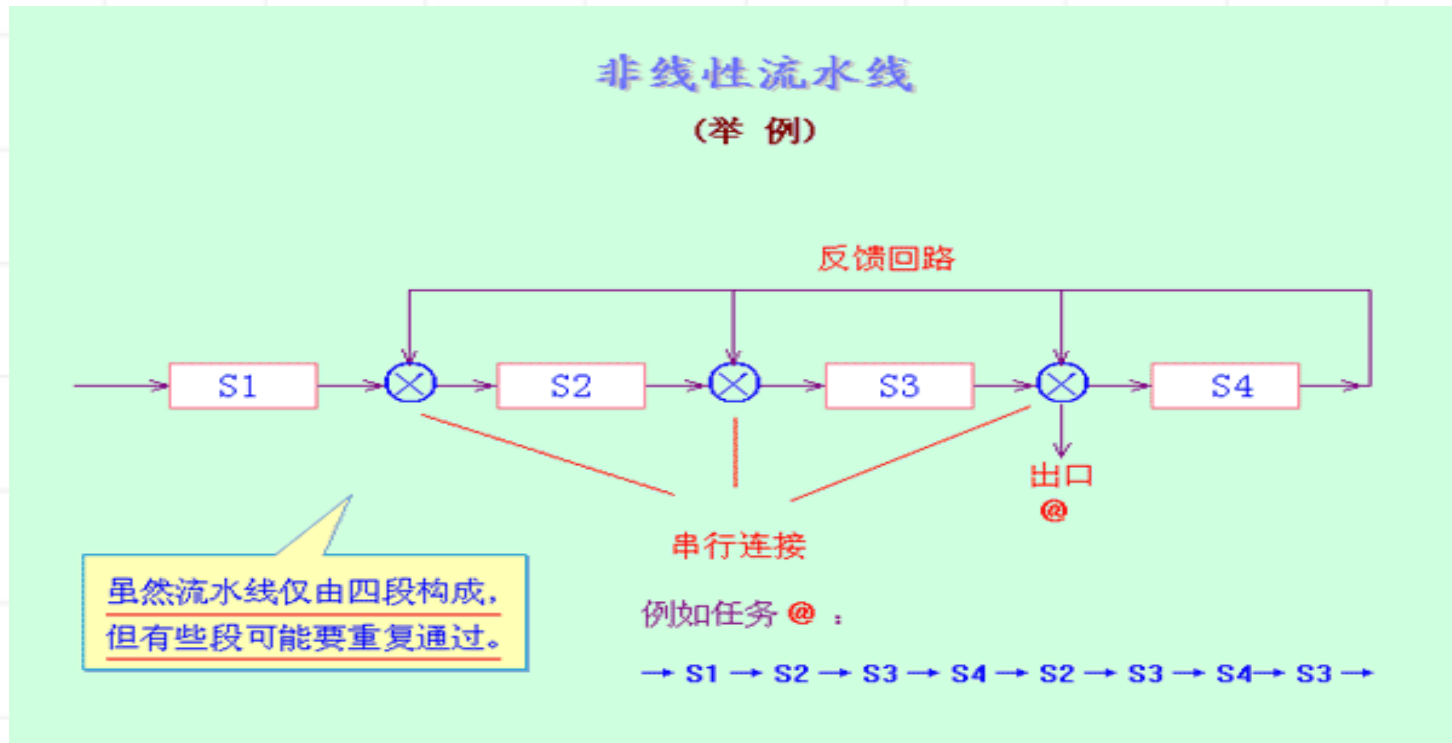
冯丹

武汉光电国家研究中心



Computer Architecture

1. 非线性流水线及其调度问题



2. 基础概念

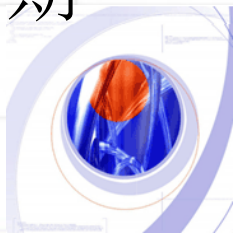
— 启动距离

- 向一条非线性流水线的输入端连续输入两个任务之间的时间间隔称为非线性流水线的启动距离。

— 禁用启动距离

- 会引起非线性流水线功能段使用冲突的启动距离则称为禁用启动距离。

— 启动距离和禁用启动距离一般都用时钟周期数来表示，是一个正整数。



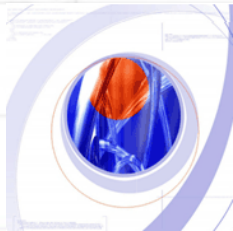
3. 预约表

例：一个5功能段非线性流水线预约表

时间 功能段	1	2	3	4	5	6	7	8	9
S1	√								√
S2		√	√					√	
S3				√					
S4					√	√			
S5							√	√	

➤ 预约表

- 横向（向右）：时间（一般用时钟周期表示）
- 纵向（向下）：流水线的段
- 如果在第 n 个时钟周期使用第 k 段，则第 k 行和第 n 列的交叉处的格子里有一个√。
- 如果在第 k 行和第 n 列的交叉处的格子里有一个√，则表示在第 n 个时钟周期要使用第 k 段。



4. 禁止表

— 禁止表F

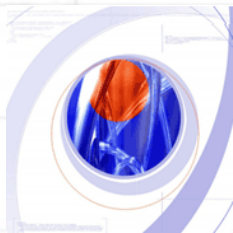
- 一个由禁用启动距离构成的集合

— 根据预约表写出禁止表F

- 对于预约表的每一行的任何一对 $\sqrt{}$ ，用它们所在的列号相减（大的减小的），列出各种可能的差值，然后删除相同的，剩下的就是禁止表的元素。

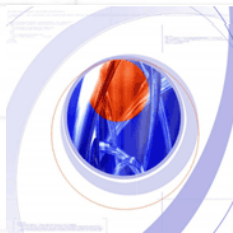
— 在上例中

- 第1行的差值只有一个：8；
- 第2行的差值有3个：1，5，6；
- 第3行只有一个 $\sqrt{}$ ，没有差值；
- 第4和第5行的差值都只有一个：1；
- 其禁止表是：F = { 1, 5, 6, 8 }



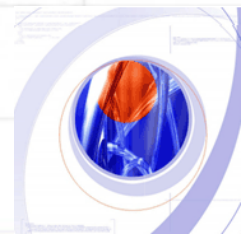
5-1. 冲突向量之初始冲突向量

- 冲突向量C
 - 一个N位的二进制位串
- 根据禁止表F写出初始冲突向量C₀
- 设C₀ = (c_Nc_{N-1}...c_i...c₂c₁)，则：
$$c_i = \begin{cases} 1 & i \in F \\ 0 & i \notin F \end{cases}$$
- 当第一个任务流入流水线后，初始冲突向量C₀决定了下一个任务需间隔多少个时钟周期才可以流入
 - c_i=0：允许间隔i个时钟周期后送入后续任务
 - c_i=1：不允许间隔i个时钟周期后送入后续任务
- 对于上面的例子
 - F = { 1, 5, 6, 8 }
 - C₀ = (10110001)



5-2. 第二个任务流入后的新冲突向量

- 假设第二个任务是在与第一个任务间隔 j 个时钟周期流入，这时，由于第一个任务已经在流水线中前进了 j 个时钟周期，其相应的禁止表中各元素的值都应该减去 j ，并丢弃小于等于0的值。
- 对冲突向量来说，就是逻辑右移 j 位（左边补0）。
- 在冲突向量上，就是对它们的冲突向量进行“或”运算。
 - $\text{SHR}^{(j)}(C_0) \vee C_0$ 注： $\text{SHR}^{(j)}$ 表示逻辑右移 j 位



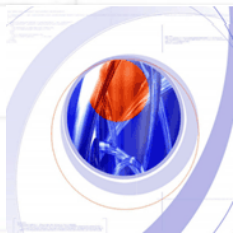
5-3. 任意一个任务进入后的新冲突向量

– 推广到更一般的情况

- 假设: C_k 为当前的冲突向量, j : 允许的时间间隔
- 则新的冲突向量为: $SHR^{(j)}(C_k) \vee C_0$

– 获得所有可能的冲突向量, 构成流水线状态集合

- 从初始冲突向量 C_0 出发, 对于所有允许的时间间隔都按上述步骤求出其新的冲突向量, 并且把新的冲突向量作为当前冲突向量, 反复使用上述步骤, 直到不再产生新的冲突向量为止。



5-4. 冲突向量集合

(1) $C_0 = (10110001)$

引入后续任务可用的时间间隔为：2、3、4、7个时钟周期

如果采用2，则新的冲突向量为：

$$(00101100) \vee (10110001) = (10111101)$$

如果采用3，则新的冲突向量为：

$$(00010110) \vee (10110001) = (10110111)$$

如果采用4，则新的冲突向量为：

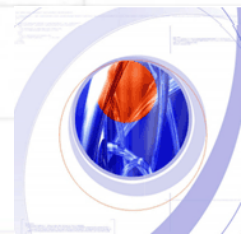
$$(00001011) \vee (10110001) = (10111011)$$

如果采用7，则新的冲突向量为：

$$(00000001) \vee (10110001) = (10110001)$$

(2) 对于新向量 (10111101) ，其可用的时间间隔为2个和7个时钟周期。用类似上面的方法，可以求出其后续的冲突向量分别为 (10111101) 和 (10110001) 。

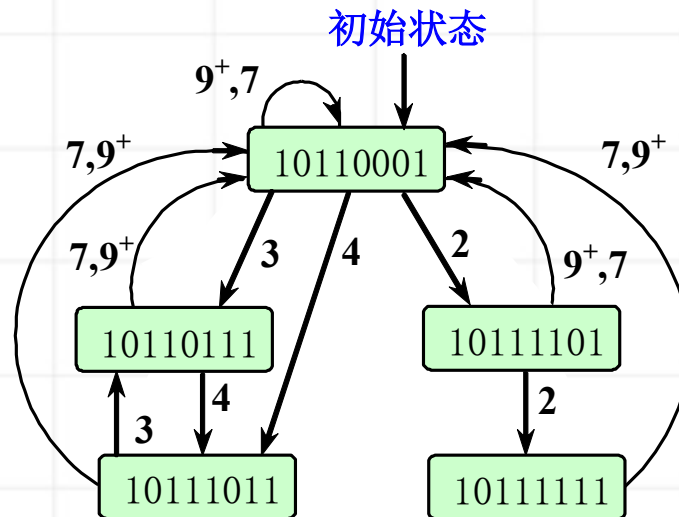
(3) 对于其他新向量，也照此处理。



6.流水线状态转移图

– 流水线状态转移图的三要素

- 流水线状态集合（所有可能的冲突向量）
- 有向弧：表示状态转移的方向
- 弧上的数字：表示引入后续任务（从而产生新的冲突向量）所用的时间间隔（时钟周期数）



7-1.调度方案与最优调度方案

— 调度方案

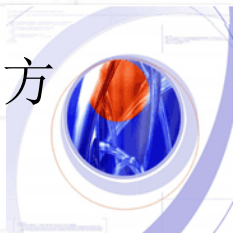
- 根据流水线状态图，由初始状态出发，任何一个闭合回路即为一种调度方案。

— 最优调度方案

- （任务进入流水线的）平均时间间隔最小的调度方案
- 列出所有可能的调度方案，计算出每种方案的平均时间间隔，从中找出其最小者即为最优调度方案。

— 上例中，各种调度方案及其平均间隔时间。

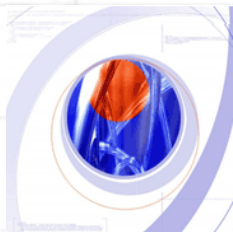
- **最佳方案：**（3，4），平均间隔时间：**3.5**个时钟周期（吞吐率最高）
- 方案（4，3）的平均间隔时间也是3.5，但它不是最佳方案，为什么？



7-2.调度方案与最优调度方案

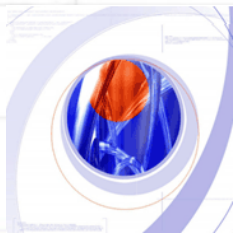
各种调度策略及平均延迟拍数

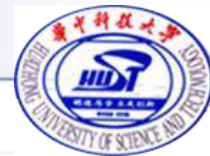
调度策略	平均延迟拍数
(2,7)	4.5
(2,2,7)	3.67
(3,7)	5
(3,4)	3.5
(3,4,3,7)	4.25
(3,4,7)	4.67
(4,3,7)	4.67
(4,7)	5.5
(7)	7



8.等时间间隔调度方案

- 方案 (3, 4) 是一种不等时间间隔的调度方案，与等间隔的调度方案相比，在控制上要复杂得多
- 为了简化控制，也可以采用等间隔时间的调度方案，但吞吐率和效率往往会下降不少
 - 在上述例子中，等时间间隔的方案只有一个：
(7)，其吞吐率下降了一半





谢谢大家

