



# 第五章 重叠、流水和向量处理机

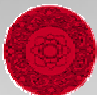
## 本章重点

- | 一次重叠方式中，各种相关的处理；
- | 流水线的时空图和性能分析；
- | 流水的局部性相关的处理，全局相关的处理和中断的处理；
- | 单功能非线性流水线的调度；
- | 向量流水方式；
- | 向量流水机上，向量指令之间并行和链接执行的特点。



## 本章的难点

- | 给出指令间微操作的时间重叠关系要求，算出全部指令完成所需要的时间；
- | 给出数学计算公式，在二功能静态流水线上，为获得尽可能高的性能，如何调整其操作的流入顺序，画出时空图，计算吞吐率，效率及加速比；
- | 消除流水线速度性能瓶颈采取的措施及相应的时空图画法，吞吐率和效率的计算；



## 本章的难点

- | 优化单功能非线性流水线的调度方案，相应的时空图画法，吞吐率和效率的计算；
- | **CRAY-1**向量指令序列中，向量指令之间并行、链接、串行的判断，计算完成全部指令所需要的拍数；
- | 在超长指令字、超标量处理机、超流水线处理机，给出指令数和并行的度数，画出相应的时空图，计算相对于度为1的常规标量流水处理机处理的加速比。



## 本章内容要点

- | 重叠方式
- | 流水方式
- | 向量的流水处理和向量流水处理机
- | 指令级高度并行的超级处理机



# 本章的内容要点(一)

## | 重叠方式

- z 指令的顺序方式和重叠方式解释
- z 重叠方法对计算机组成的要求
- z “一次重叠”方式的相关控制

## | 流水方式

- z 流水是重叠的引申
- z 流水线的分类
- z 流水线处理机的性能
- z 流水机器的相关
- z 流水机器局部性相关的处理
- z 流水机器全局性相关的处理
- z 流水机器的中断处理
- z 流水线的任务调度



## 本章的内容要点(2)

- | 向量的流水处理与向量流水处理机
  - z 向量的流水处理
  - z 向量流水处理机
- | 指令级高度并行的超级处理机
  - z 超标量处理机： $m$ 条指令流水线同时并行；
  - z 超长指令字处理机是将水平型微码和超标量处理相结合；
  - z 超流水线处理机是采用多相的高频时钟。



假设一条指令的解释分为取指、分析与执行3步，每步的相应时间为 $t_{\text{取指}}$ ， $t_{\text{分析}}$ ， $t_{\text{执行}}$ 。

(1) 分别计算下列几种情况下，执行100条指令所需时间的一般关系式

(i) 顺序方式；

(ii) 仅“执行<sub>k</sub>”与“取指<sub>k+1</sub>”重叠

(iii) 仅“执行<sub>k</sub>”、“分析<sub>k+1</sub>”、“取指<sub>k+2</sub>”重叠

(2) 分别在 $t_{\text{取指}}=t_{\text{分析}}=2$ ， $t_{\text{执行}}=1$ ；及 $t_{\text{取指}}=t_{\text{执行}}=5$ ， $t_{\text{分析}}=2$ 两种情况下，计算出上述各结果





## 三种情况下的时间关系

取指 <sub>k</sub>	分析 <sub>k</sub>	执行 <sub>k</sub>	取指 <sub>k+1</sub>	分析 <sub>k+1</sub>	执行 <sub>k+1</sub>
-----------------	-----------------	-----------------	-------------------	-------------------	-------------------

顺序方式工作的时间关系图

取指 <sub>k</sub>	分析 <sub>k</sub>	执行 <sub>k</sub>
-----------------	-----------------	-----------------

取指 <sub>k+1</sub>	分析 <sub>k+1</sub>	执行 <sub>k+1</sub>
-------------------	-------------------	-------------------

取指 <sub>k+2</sub>	分析 <sub>k+2</sub>	执行 <sub>k+2</sub>
-------------------	-------------------	-------------------

仅“执行<sub>k</sub>”与“取指<sub>k+1</sub>”重叠

取指 <sub>k</sub>	分析 <sub>k</sub>	执行 <sub>k</sub>
-----------------	-----------------	-----------------

取指 <sub>k+1</sub>	分析 <sub>k+1</sub>	执行 <sub>k+1</sub>
-------------------	-------------------	-------------------

取指 <sub>k+2</sub>	分析 <sub>k+2</sub>	执行 <sub>k+2</sub>
-------------------	-------------------	-------------------

仅“执行<sub>k</sub>”、“分析<sub>k+1</sub>”、“取指<sub>k+2</sub>”重叠



## 第二题解答

### | (1) 顺序方式工作时

$$100 * (t_{\text{取指}} + t_{\text{分析}} + t_{\text{执行}})$$

### | (2) 仅“执行<sub>k</sub>”与“取指<sub>k+1</sub>”重叠

$$t_{\text{取指}} + 100t_{\text{分析}} + t_{\text{执行}} + 99 * \max\{t_{\text{取指}}, t_{\text{执行}}\}$$

### | (3) 仅“执行<sub>k</sub>”、“分析<sub>k+1</sub>”、“取指<sub>k+2</sub>”重叠

$$t_{\text{取指}} + \max\{t_{\text{分析}}, t_{\text{取指}}\} + 98 * \max\{t_{\text{取指}}, t_{\text{分析}}, t_{\text{执行}}\} + \max\{t_{\text{执行}}, t_{\text{分析}}\} + t_{\text{执行}}$$

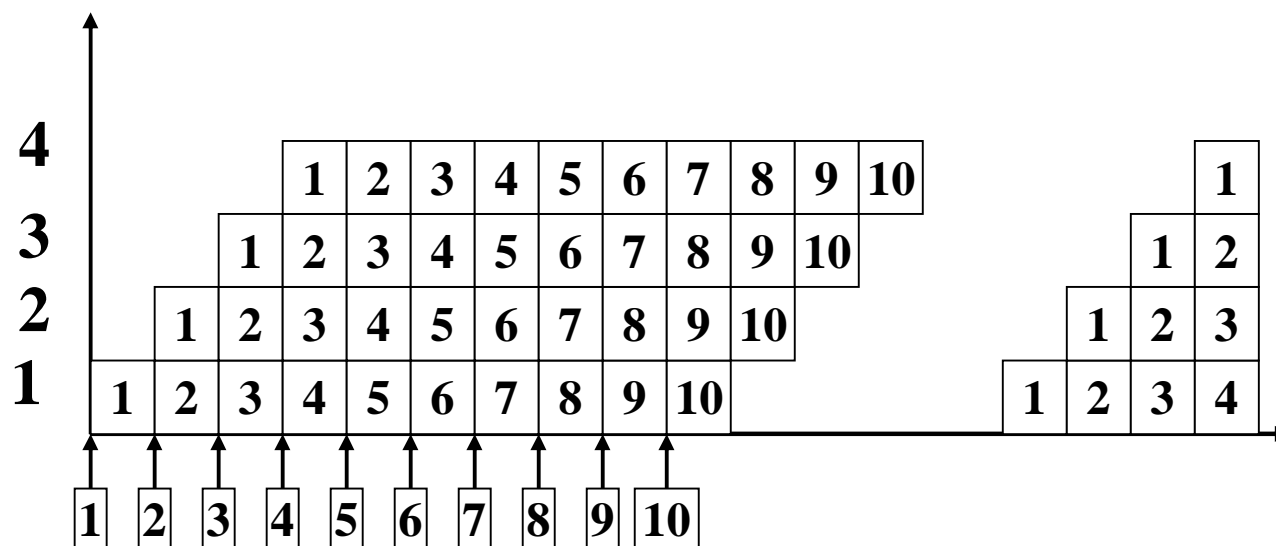


## 第五题

| 流水线 4 个功能部件，每个部件的延迟时间是  $\Delta t$ ，输入 10 个数据，间歇  $5 \Delta t$ ，又输入 10 个数据，周期工作，求吞吐率，并画出时-空图



# 第五题解答



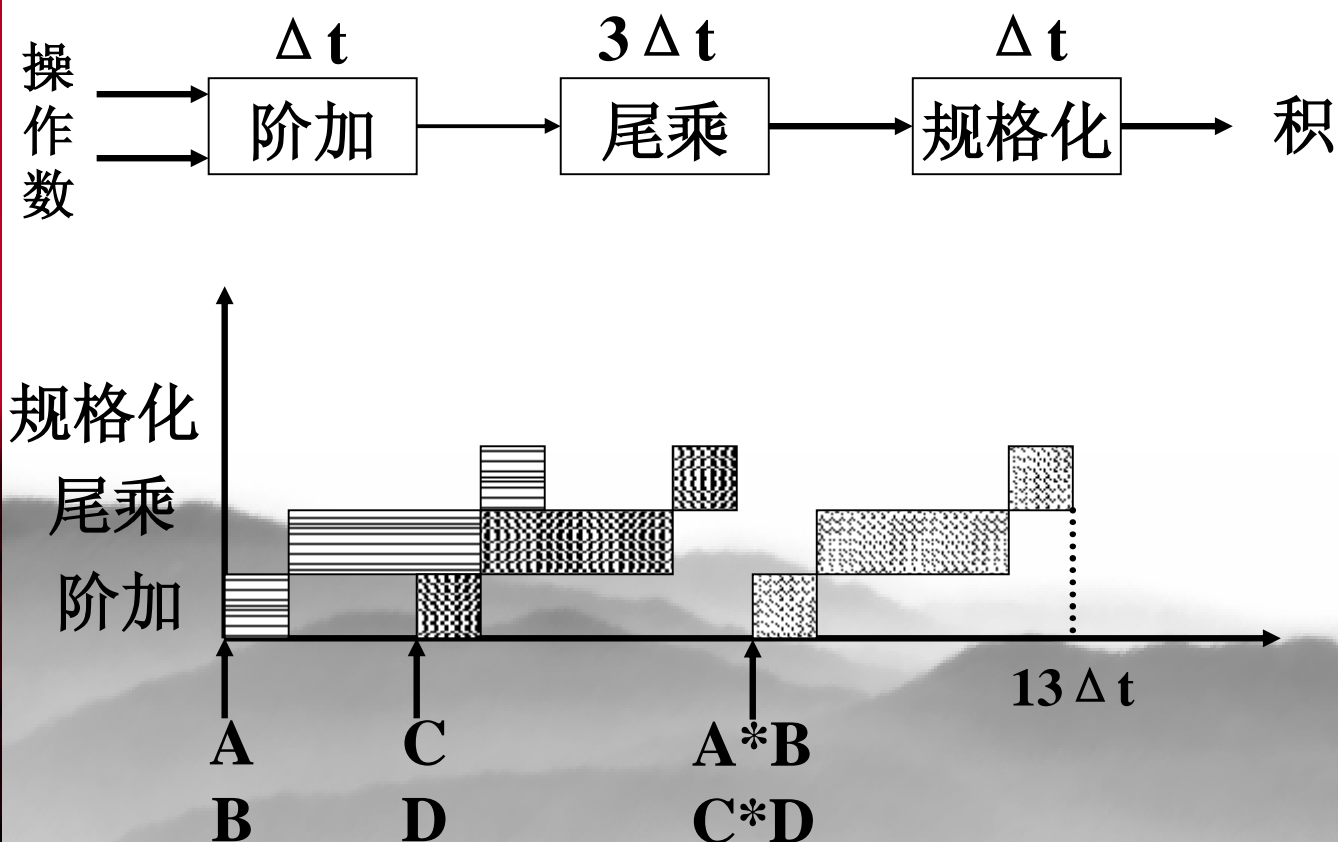
$$T_p = \frac{10}{10\Delta t + 5\Delta t} = \frac{10}{15\Delta t} = \frac{2}{3\Delta t}$$

注意：不要漏了  $\Delta t$

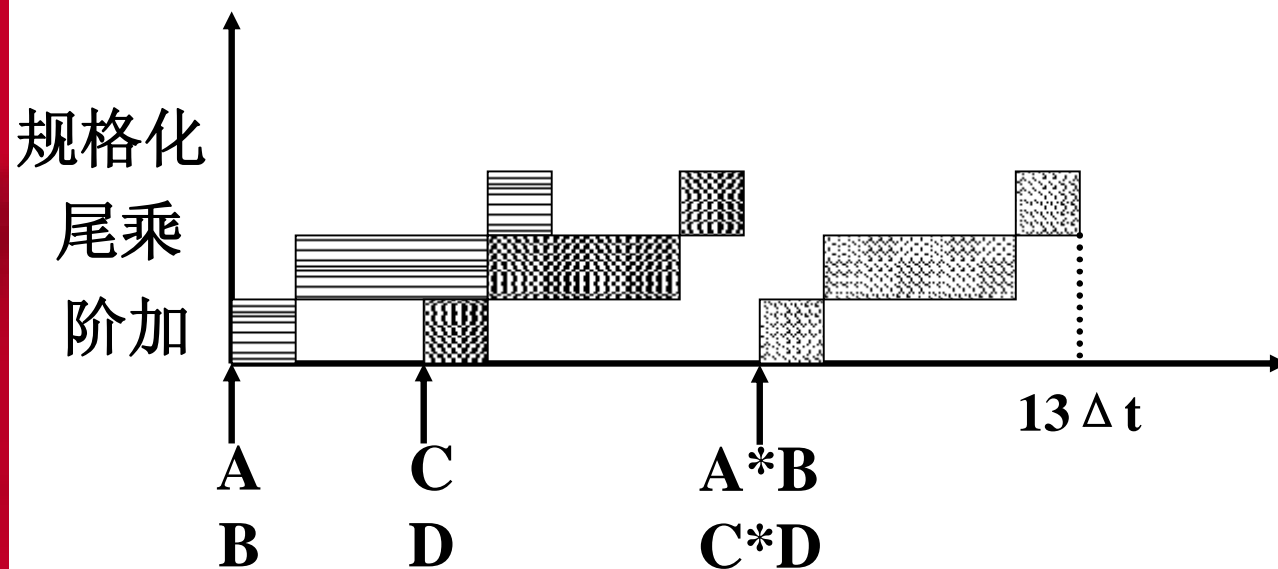


## 第六题

浮点乘法流水线，乘积可以直接返回到输入端或暂存于相应的缓冲寄存器中。  
画出 $A*B*C*D$



# 第六题解答

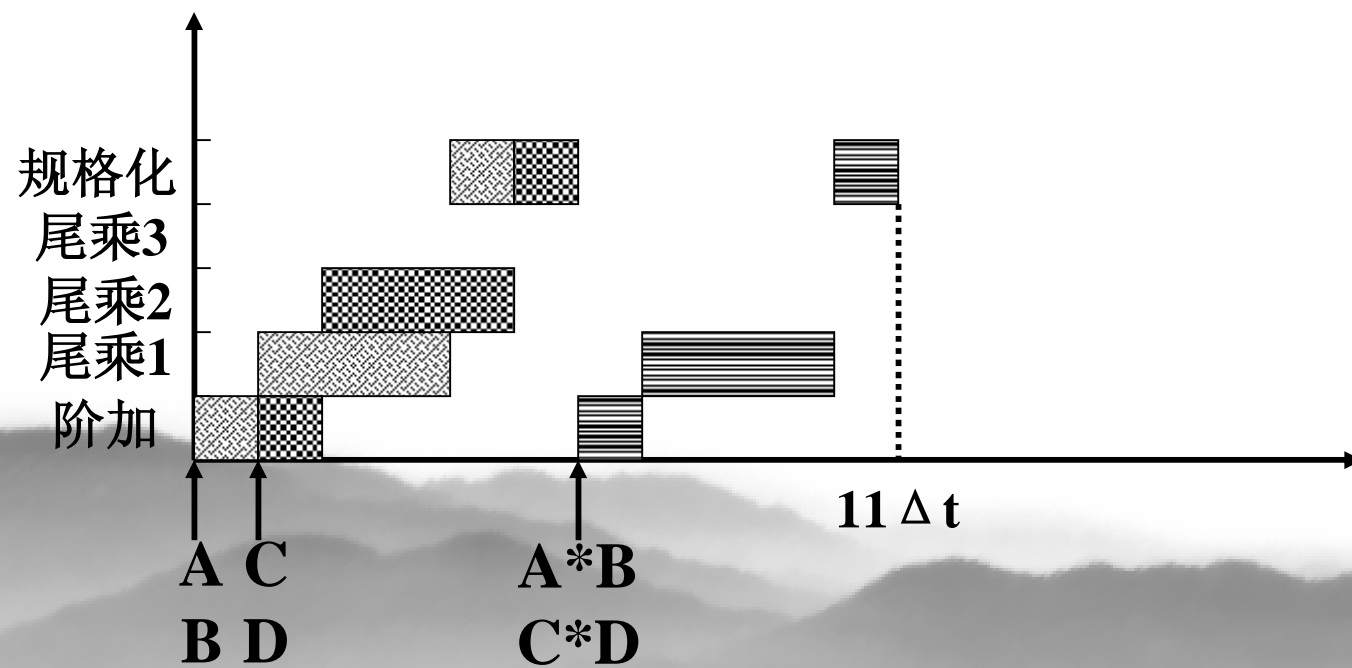
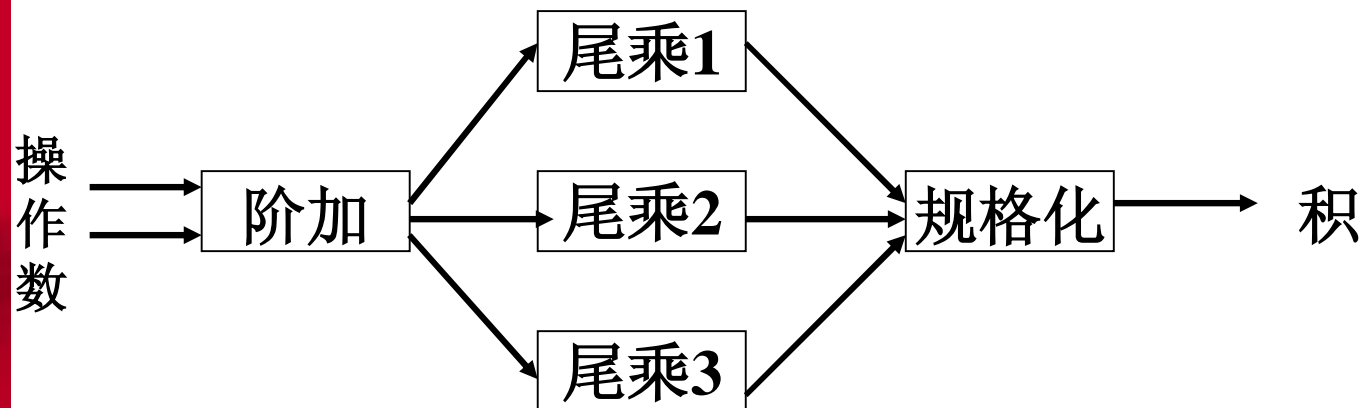


$$T_p = \frac{3}{13\Delta t}$$

$$h = \frac{(3 + 3 + 9) * \Delta t}{13 * 3 \Delta t} = \frac{15}{39} = \frac{5}{13}$$



# 第六题



$$T_p = \frac{3}{11 \Delta t}$$

$$h = \frac{5 * 3}{11 * 5} = \frac{3}{11}$$



## 第九题

为提高流水线的吞吐率，可以采取哪两种方法克服速度瓶颈？现有3段流水线，经过各段的时间为  $\Delta t$ ,  $3\Delta t$ ,  $\Delta t$ 。

- (1) 计算连续输入3条和连续输入30条指令时的吞吐率和效率
- (2) 按两种途径进行改进，画出流水线结构示意图，计算连续输入3条和连续输入30条指令时的吞吐率和效率。
- (3) 通过对(1)、(2)两小题的计算结果，你得出什么有用的结论？





## 第九题解答

$$T = \frac{n}{\sum_{i=1}^m \Delta t_i + (n-1)\Delta t_j}, \quad \text{其中, } t_j = \max\{\Delta t_1, \Delta t_2, \dots, \Delta t_m\}$$

$$h = \frac{n \sum_{i=1}^m \Delta t_i}{m \left( \sum_{i=1}^m \Delta t_i + (n-1)\Delta t_j \right)}$$



## 第九题解答

提高吞吐率的两条途径：

功能段细分、重复设置多个功能部件

(1) a. 当连续流入3条指令时

$$T_p = \frac{3}{11 \Delta t} \quad h = \frac{5}{11}$$

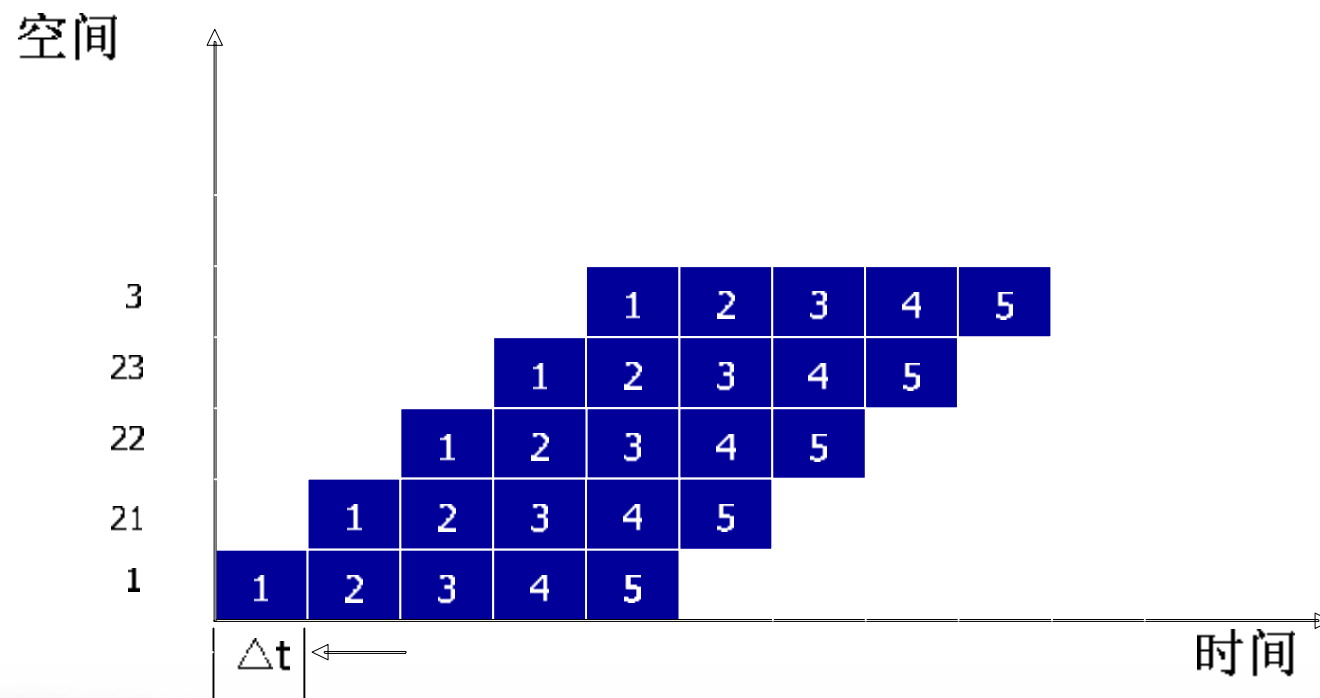
(1) b. 当连续流入30条指令时

$$T_p = \frac{15}{46 \Delta t}, \quad h = \frac{25}{46}$$



# 第九题解答

## (2) 将第二段划分为三段



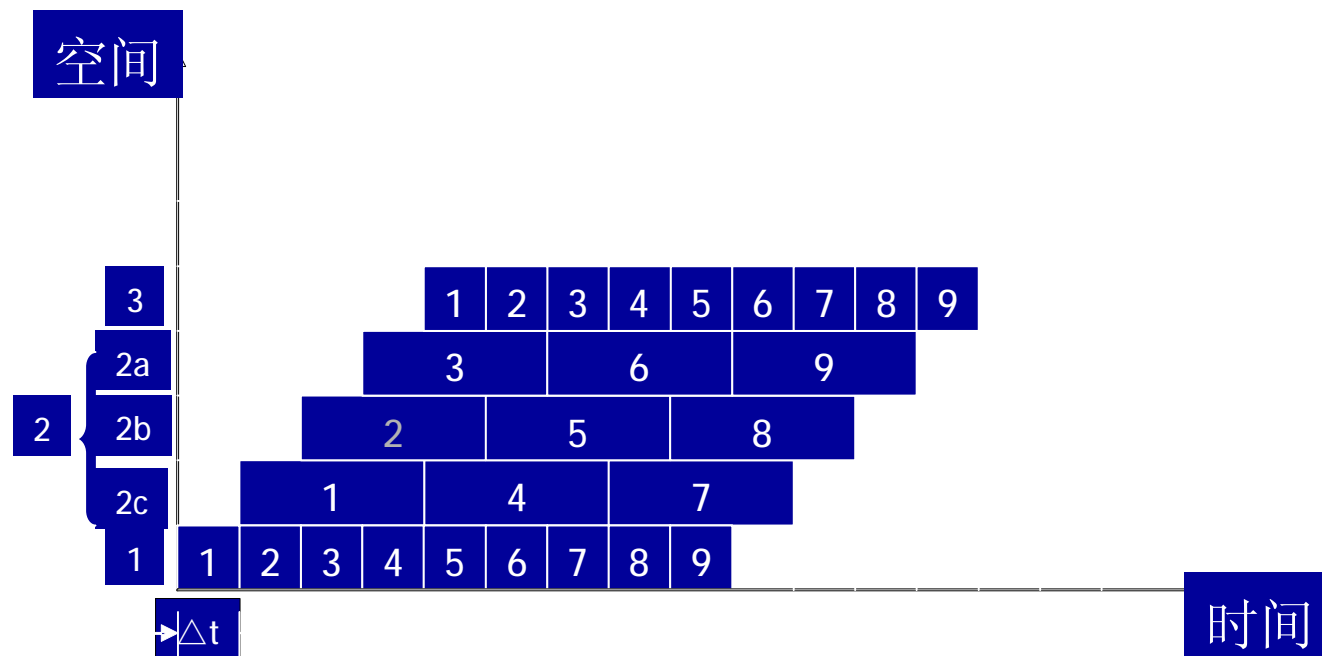
$$n = 3 \text{ 时: } T_p = \frac{3}{7\Delta t}, \quad h = \frac{3}{7}$$

$$n = 30 \text{ 时: } T_p = \frac{15}{17\Delta t}, \quad h = \frac{15}{17}$$



# 第九题解答

## (2)重复设置功能部件时空图



$$n = 3 \text{ 时: } T_p = \frac{3}{7 \Delta t}, \quad h = \frac{3}{7}$$

$$n = 30 \text{ 时: } T_p = \frac{15}{17 \Delta t}, \quad h = \frac{15}{17}$$



## 第九题解答

| (3)得出结论：  
吞吐率取决于瓶颈段



## 第十题

| 双功能静态流水线，4段，时间分别为  $\Delta t$ ,  $2\Delta t$ ,  $2\Delta t$ ,  $\Delta t$ , 其中，1, 2, 4为加法，1, 3, 4为乘法，输出可以马上回授到输入端。试计算

$$A * (B + C * (D + E * F)) + G * H$$

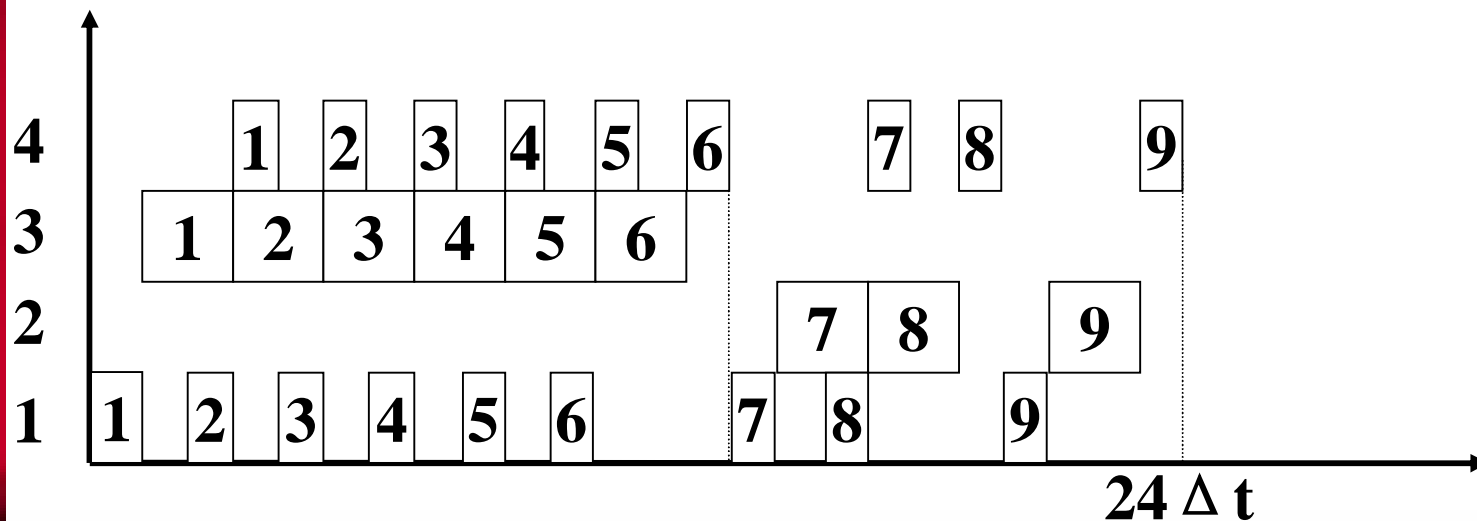
画出能获得尽可能高的吞吐率的流水时空图。如果对瓶颈子部件进行进一步的细分，最少须多长时间可完成运算，若3无法再细分，只能采用并联方法改进，问流水线的效率为多少？



# 第十题解答

解题原则：后面越早需要的运算结果，就越尽可能早的开始运算！

$$A * C(D + E * F) + A * B + G * H$$



- 1:计算A\*C    2:计算E\*F    3:计算A\*B  
 4:计算G\*H    5:计算A\*C\*D    6:计算A\*C\*E\*F  
 7:计算A\*C\*E\*F+A\*C\*D    8:计算A\*B+G\*H  
 9:计算结束



## 第十题解答

$$T_p = \frac{9}{24\Delta t} = \frac{3}{8\Delta t}$$

$$h = \frac{6 * 4 + 4 * 3}{24 * 4} = \frac{9}{24} = \frac{3}{8}$$





# 第十题解答

↑ 对瓶颈部件细分的时空图如下：

4				1	2	3	4	5	6				7	8				9				
32			1	2	3	4	5	6														
31		1	2	3	4	5	6															
22												7	8				9					
21											7	8				9						
1	1	2	3	4	5	6				7	8				9							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19			

1:计算 $A * C$     2:计算 $E * F$     3:计算 $A * B$

4:计算 $G * H$     5:计算 $A * C * D$     6:计算 $A * C * E * F$

7:计算 $A * C * E * F + A * C * D$

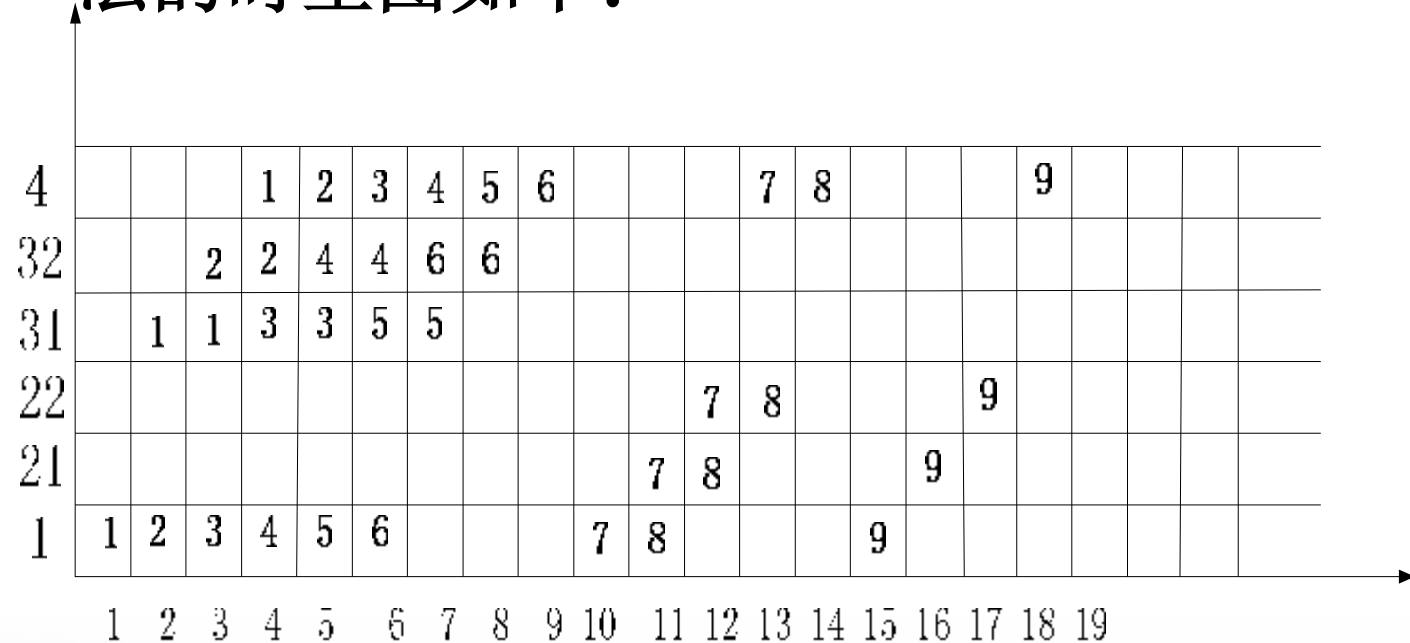
9:计算结束

则对流水线瓶颈子过程细分后，最少需要18  $\Delta t$ 可完成全部运算。



# 第十题解答

(3)若子过程3已无法再细分，采用并联方法的时空图如下：



$$h = \frac{6 * 4 + 3 * 4}{18 * 6} = \frac{36}{18 * 6} = \frac{1}{3}$$



## 第十题解答

### 重要问题

乘法顺序不同，计算结果不同

自己试着画出首先计算 $A*B$ 的时空图



# 第十四题

## 预约表

时间 段 \ 时间	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$
$S_1$	✓								✓
$S_2$		✓	✓						
$S_3$				✓			✓	✓	
$S_4$				✓	✓				
$S_5$						✓	✓		

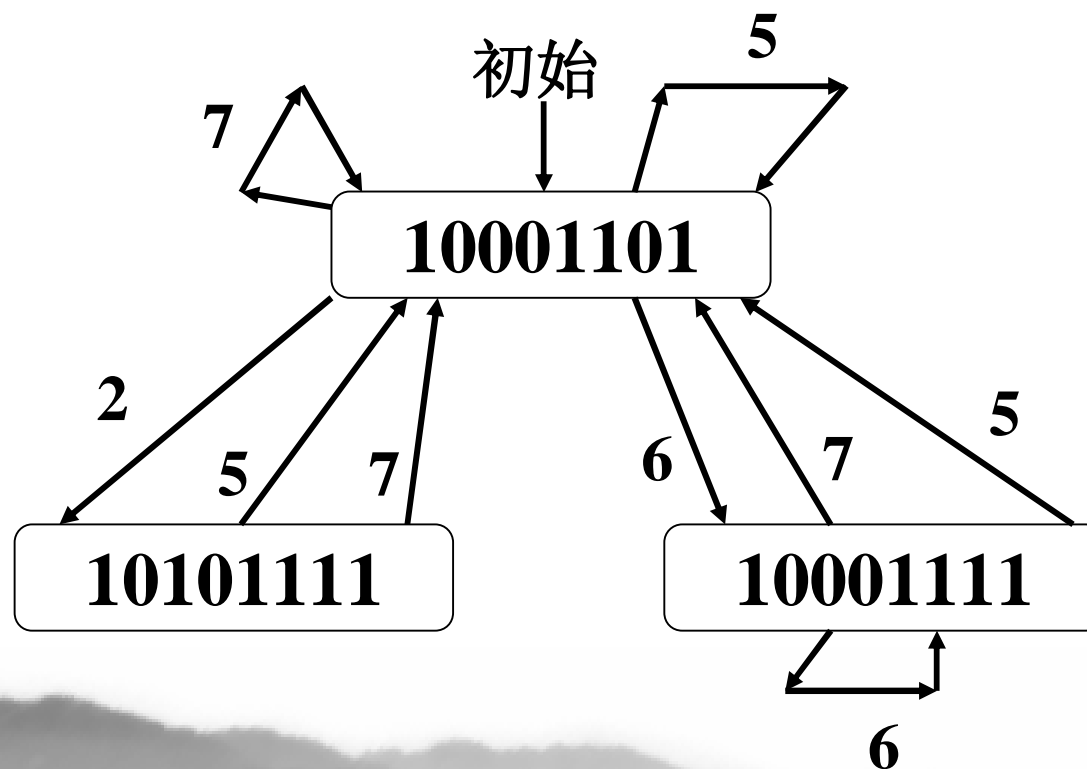
1.延迟禁止表{1, 3, 4, 8}

2.冲突向量 (10001101)



# 第十四题

## 状态转意图



## | 调度方案

调度方案	平均延迟
(2, 5)	3.5
(2, 7)	4.5
5	5
(6, 5)	5.5
(6)	6
(6, 7)	6.5
(7)	7
(5, 2)	3.5

