

# 计算机系统结构

## ---习题讲解

吴 非

Tel: 15071339557

Email: [wufei@hust.edu.cn](mailto:wufei@hust.edu.cn)

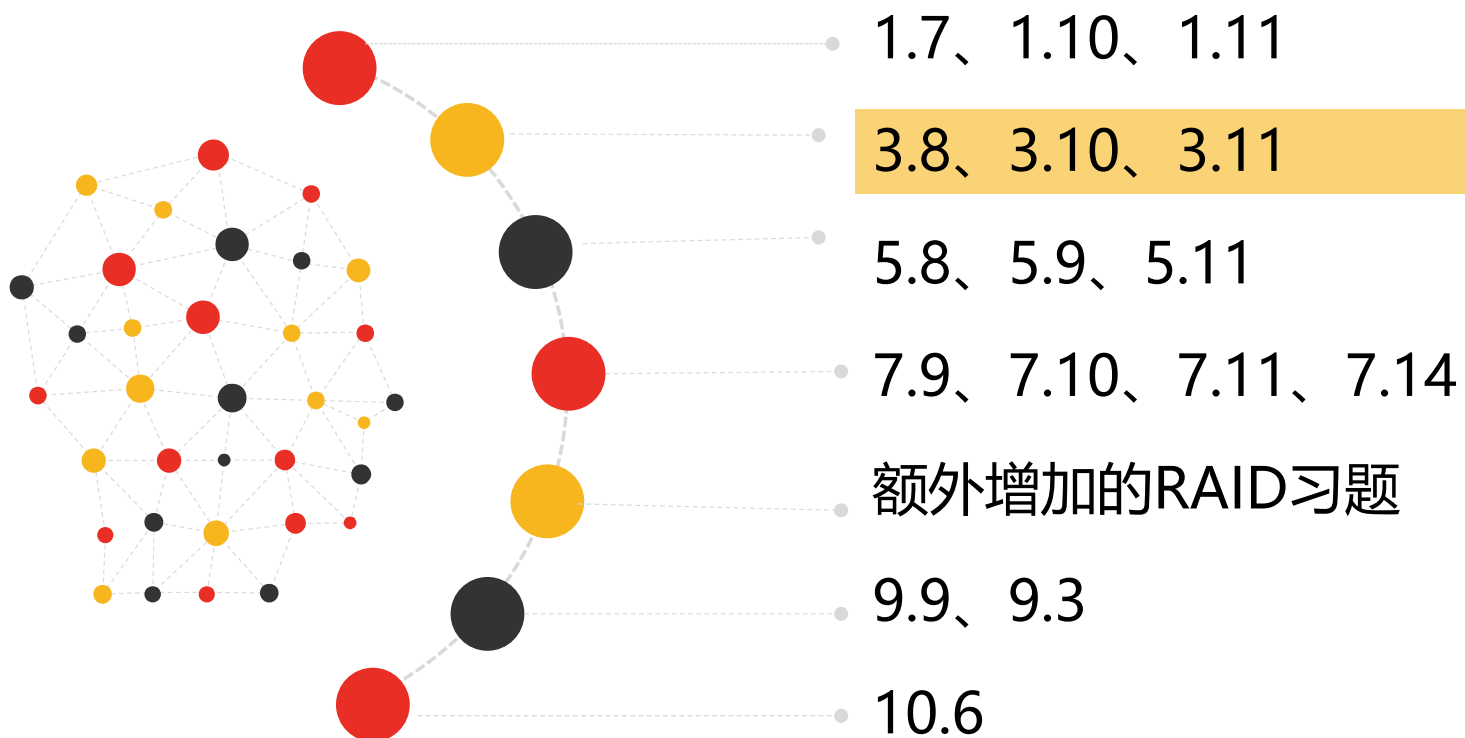
华中科技大学信息存储系统教育部重点实验室

武汉光电国家研究中心信息存储实验室





# 计算机系统结构习题内容





# 第一章

Amdahl定律:

$$S_n = \frac{T_0}{T_n} = \frac{1}{(1 - Fe) + \frac{Fe}{Se}}$$

$$\text{CPU时间} = IC * CPI * \text{时钟周期时间} = \sum_{i=1}^n (CPI_i + IC_i) * \text{时钟周期时间}$$

$$CPI = \frac{\text{时钟周期数}}{IC} = \frac{\sum_{i=1}^n (CPI_i * IC_i)}{IC} = \sum_{i=1}^n (CPI_i * \frac{IC_i}{IC})$$

$$MIPS \text{ 速率} = \frac{f}{CPI \times 10^6}$$



## 习题1.7

对于一台400MHz计算机执行标准测试程序，程序中指令类型，执行数量和平均时钟周期数如下：

指令类型	指令执行数量	平均时钟周期数
整数	45000	1
数据传送	75000	2
浮点	8000	4
分支	1500	2

求该计算机的有效CPI、MIPS和程序执行时间。

解：

$$CPI = \sum_{i=1}^4 (CPI_i \times \frac{IC_i}{IC}) = 1 \times \frac{45000}{129500} + 2 \times \frac{75000}{129500} + 4 \times \frac{8000}{129500} + 2 \times \frac{1500}{129500} = \frac{230000}{129500} \approx 1.776$$

$$MIPS = \frac{f}{CPI \times 10^6} \approx \frac{400 \times 10^6}{1.776 \times 10^6} \approx \frac{400}{1.776} \approx 225.225 MIPS$$

$$T = \frac{IC}{MIPS \times 10^6} \approx \frac{129500 \times 1.776}{400 \times 10^6} \approx 0.000575 (\text{秒}) = 0.575 (\text{毫秒})$$



## 习题1.10

计算机系统有三个部件可以改进，这三个部件的加速比如下：部件加速比1 = 30；部件加速比2 = 20；部件加速比3 = 10；

(1) 如果部件1和部件2的可改进比例为30%，那么当部件3的可改进比例为多少时，系统的加速比才可以达到10？

(2) 如果三个部件的可改进比例为30%、30%和20%，三个部件同时改进，那么系统中不可加速部分的执行时间在总执行时间中占的比例是多少？

$$T_e = T_o \left[ (1 - f_e) + \frac{f_e}{S_e} \right] \quad S = \frac{1}{(1 - f_e) + \frac{f_e}{S_e}} \quad S = \frac{1}{(1 - \sum_i f_i) + \sum_i \frac{f_i}{S_i}}$$



## 习题1.10

$$S = \left\{ [1 - (f_1 + f_2 + f_3)] + \frac{f_1}{S_1} + \frac{f_2}{S_2} + \frac{f_3}{S_3} \right\}^{-1}$$

$$10 = \left\{ [1 - (0.3 + 0.3 + f_3)] + \frac{0.3}{30} + \frac{0.3}{20} + \frac{f_3}{10} \right\}^{-1} \quad f_3 = \frac{65}{180} = 0.36$$

$$\begin{aligned} p &= \frac{[1 - (0.3 + 0.3 + 0.2)]T}{\frac{0.3T}{30} + \frac{0.3T}{20} + \frac{0.2T}{10} + 0.2T} \\ &= \frac{0.2}{\frac{0.3}{30} + \frac{0.3}{20} + \frac{0.2}{10} + 0.2} \\ &= \frac{0.2}{\frac{0.6}{60} + \frac{0.9}{60} + \frac{1.2}{60} + \frac{12}{60}} \\ &= \frac{12}{14.7} = 0.82 \end{aligned}$$



## 习题1.11

假设浮点数指令FP指令的比例为30%，其中浮点数平方根FPSQR占全部指令的比例为4%，FP操作的CPI为5，FPSQR操作的CPI为20，其他指令的平均CPI为1.25。

现有两种改进方案：（1）把FPSQR操作的CPI减至3；（2）把所有的FP操作的CPI减至3

试比较两种方案对系统性能的提高程度。

### 解法1：

利用原始CPI的唯一性，先使用已知条件求出原始CPI，再求出除去FPSQR指令外其他指令的平均CPI，最后比较改进后的CPI大小。

$$\text{原始CPI} = 5 \times 30\% + 1.25 \times (1 - 30\%) = 2.375$$

设除FPSQR外其余指令的平均CPI为X

$$\text{则 } 2.375 = 20 \times 4\% + (1 - 4\%)X, \text{ 解出 } X = 1.640625$$

$$\text{方案1: } \text{CPI}_1 = 3 \times 4\% + 1.640625 \times (1 - 4\%) = 1.695$$

$$\text{方案2: } \text{CPI}_2 = 3 \times 30\% + 1.25 \times (1 - 30\%) = 1.775$$

结论：方案1导致的新CPI更小，性能更好



## 习题1.11

解法2:

用Amdahl公式求。记指令总条数=M, 时钟周期长度=CYCLE。

$$\begin{aligned}\text{原始总时间 } T_{\text{old}} &= 0.3M \times 5 \times \text{CYCLE} + 0.7M \times 1.25 \times \text{CYCLE} \\ &= M \times 2.375 \times \text{CYCLE}\end{aligned}$$

$$T_{\text{FP}} = 0.3M \times 5 \times \text{CYCLE} = M \times 1.5 \times \text{CYCLE},$$

所占比例为  $1.5/2.375 \approx 63\%$

$$T_{\text{FPSQR}} = 0.04M \times 20 \times \text{CYCLE} = M \times 0.8 \times \text{CYCLE},$$

所占比例为  $0.8/2.375 \approx 34\%$

$$\text{方案1: } Se = 20/3, Fe \approx 34\%, Sn_1 = 1 / [(1 - Fe) + Fe / Se] \approx 1.4$$

$$\text{方案2: } Se = 5/3, Fe \approx 63\%, Sn_2 = 1 / [(1 - Fe) + Fe / Se] \approx 1.3$$

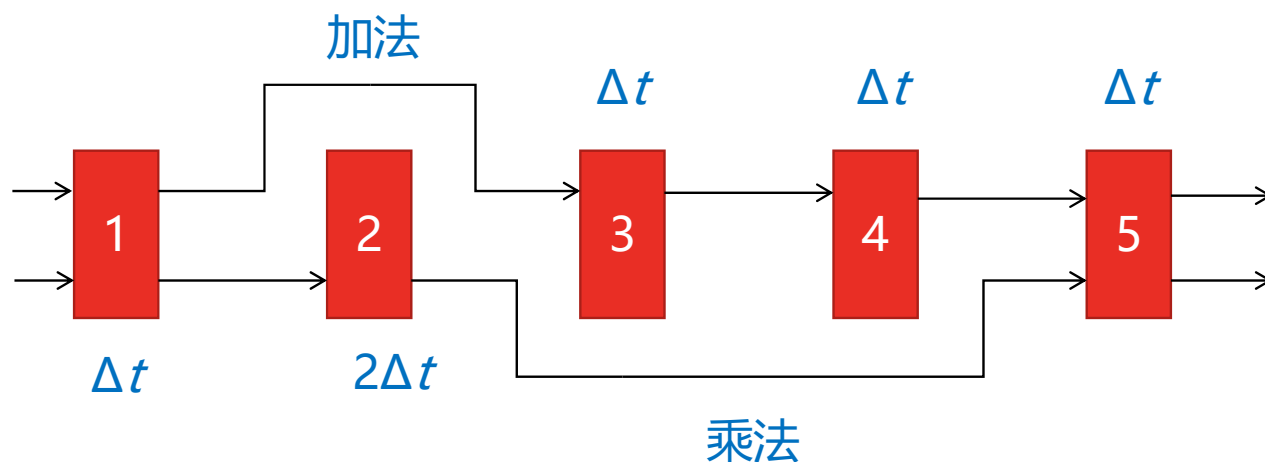
结论: 方案1导致加速比更大, 性能更好





## 习题3.8 (时空图、性能指标)

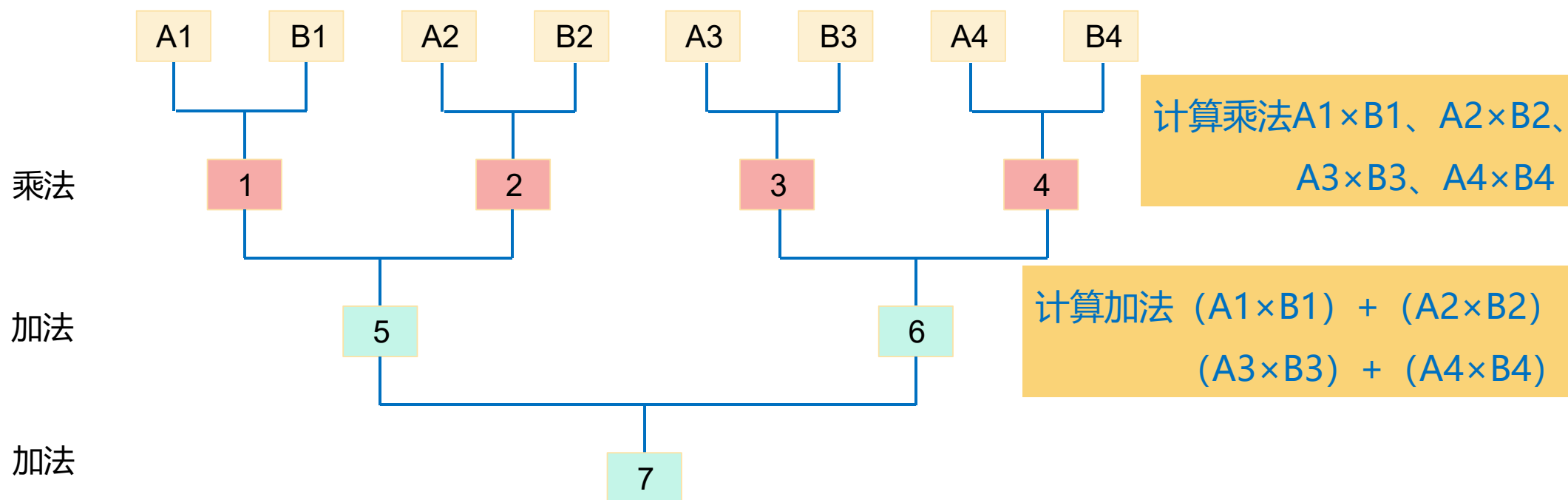
3.8 有一条**动态**多功能流水线由5段组成，加法用1、3、4、5段，乘法用1、2、5段，第2段的时间为 $2\Delta t$ ，其余各段的时间均为 $\Delta t$ ，而且流水线的输出可以直接返回输入端或暂存于相应的流水寄存器中。现要在该流水线上计算  $\sum_{i=1}^4 (A_i \times B_i)$ ，画出其时空图，并计算其吞吐率、加速比和效率。





## 习题3.8 (时空图、性能指标)

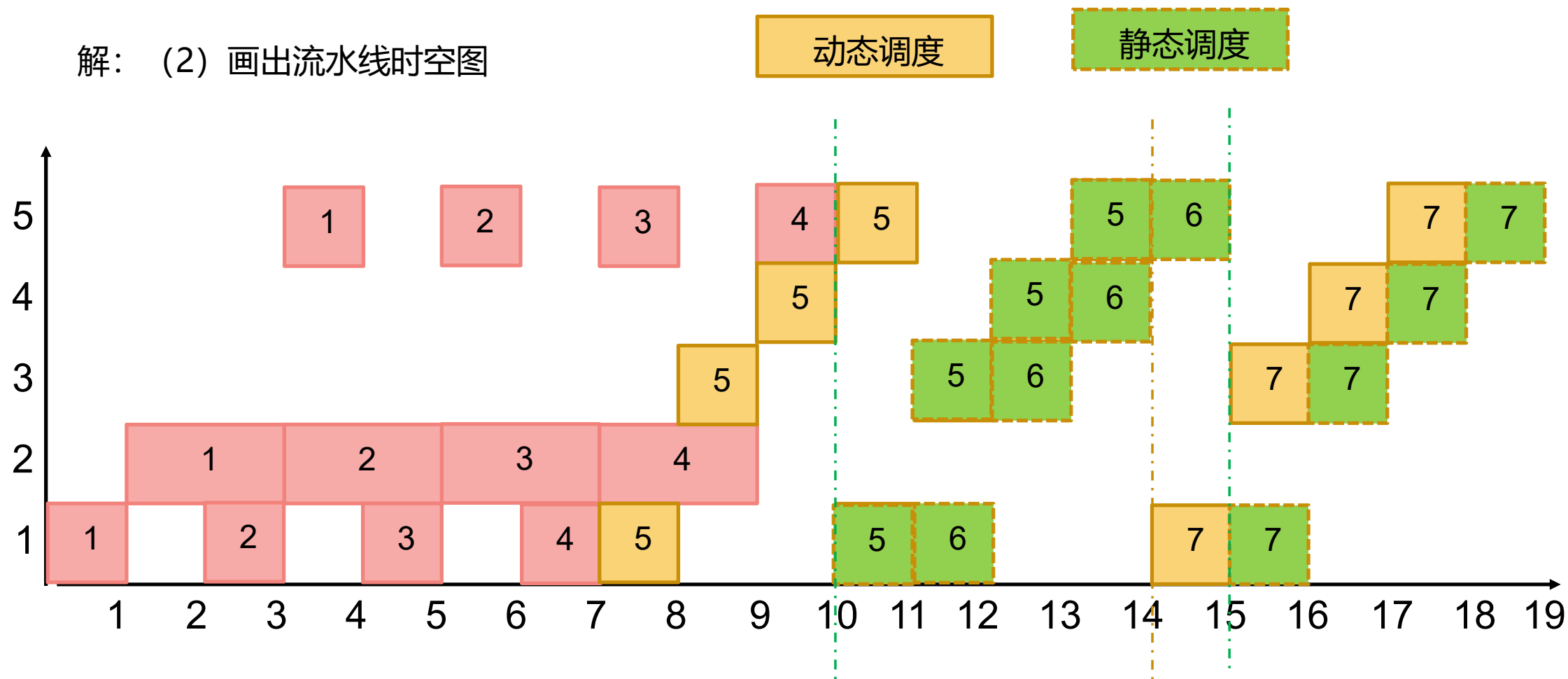
解：(1) 提高流水线效率，减少切换次数，根据最小二叉树，选择适合于流水线调度的计算策略；





## 习题3.8 (时空图、性能指标)

解： (2) 画出流水线时空图



## 习题3.8 (时空图、性能指标)

解：(3) 计算性能指标

由图可见，7个任务，花费了18个 $\Delta t$ 时间，所以吞吐率为：

$$TP = \frac{7}{18\Delta t}$$

如果不用流水线，由于一次求积需 $4\Delta t$ ，一次求和需 $4\Delta t$ ，则执行上述7个任务共需  $(4 \times 4 + 3 \times 4) \Delta t = 28\Delta t$ 。所以加速比为：

$$S = \frac{28\Delta t}{18\Delta t} \approx 1.56$$

该流水线的效率可由阴影区的面积和5个段总时空区的面积的比值求得：

$$E = \frac{4 \times 4 + 3 \times 4}{5 \times 18} \approx 0.31$$

如果是静态流水线调度，执行任务花费的时间是19个 $\Delta t$ 。

思考：为什么现在的机器很少用动态流水线调度策略？



## 习题3.10 (单功能非线性流水线调度)

3.10 有一个5段流水线, 各段执行时间均为 $\Delta t$ , 其预约表如下:

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

1. 画出流水线任务调度的状态转移图。
2. 分别求出允许不等时间间隔调度和等时间间隔调度的两种最优调度策略, 以及这两种调度策略的流水线最大吞吐率。
3. 若连续输入10个任务, 求这两种调度策略的流水线实际吞吐率和加速比。



## 习题3.10 (单功能非线性流水线调度)

解:

(1) 由预约表可得

功能段S1, 禁止启动距离7-1=6;

功能段S2, 禁止启动距离5-2=3;

功能段S3, 禁止启动距离4-3=1;

功能段S4, 禁止启动距离7-4=3;

功能段S5, 禁止启动距离6-5=1;

去掉重复的, 所以禁止表为 $F=\{6,3,1\}$

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

(2) 根据初始冲突向量定义, 初始冲突向量 $C_0 = (100101)$

$$C_i = \begin{cases} 1, & i \in F \\ 0, & i \notin F \end{cases}$$

6    3    1

↑    ↑    ↑

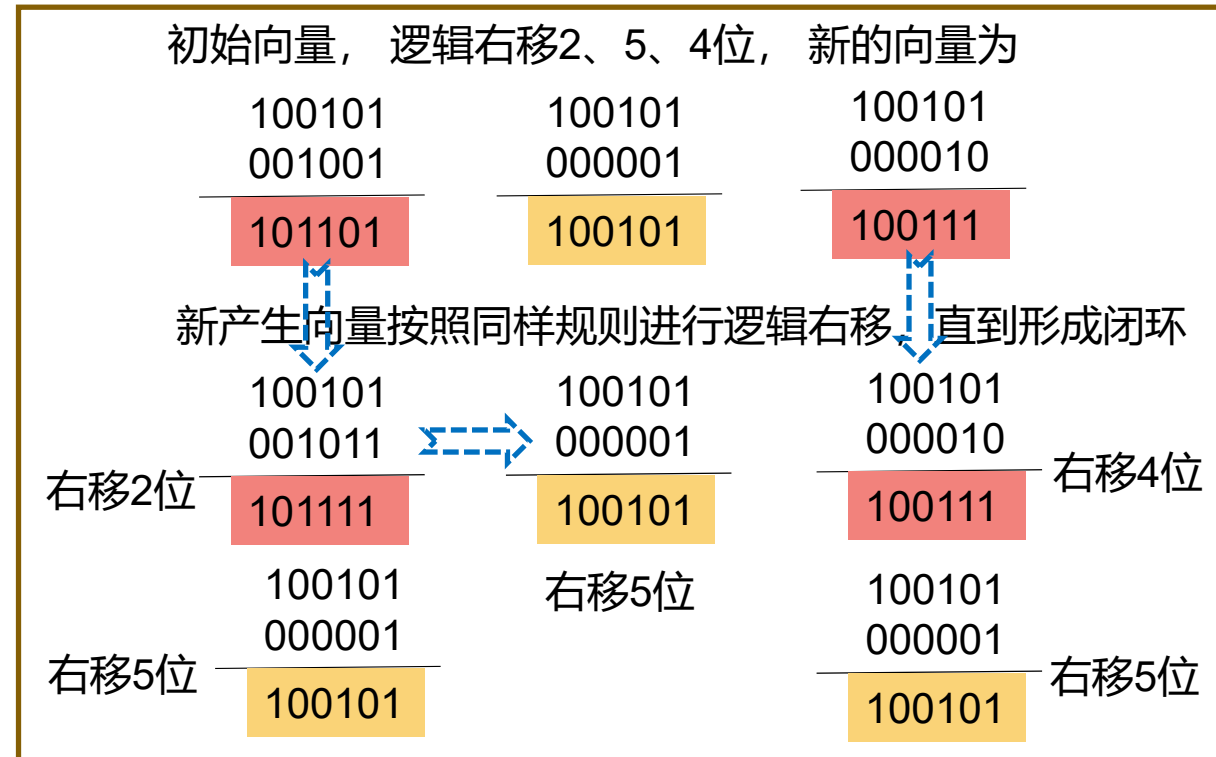
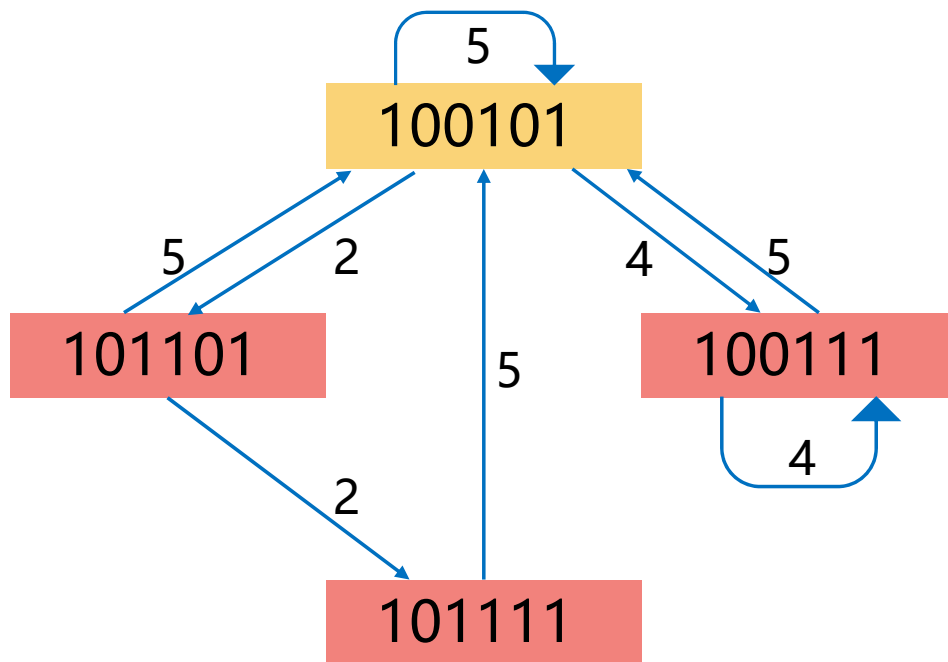


## 习题3.10 (单功能非线性流水线调度)

解:

(3) 根据初始冲突向量 $C_0 = (100101)$ , 画状态转移图

每次都是和初始冲突向量执行按位或运算!!!





## 习题3.10 (单功能非线性流水线调度)

解:

(4) 由状态转移图可得不发生段争用冲突的调度策略以及平均延迟时间

调度策略	平均延迟时间	调度策略	平均延迟时间
(2,2,5)	$3\Delta t$	(4,5)	$4.5\Delta t$
(2,5)	$3.5\Delta t$	(5)	$5\Delta t$
(4)	$4\Delta t$		

由上可知, 允许不等时间间隔调度的最优调度策略是(2,2,5), 流水线最大吞吐率为:  $1/3\Delta t$ 。

等时间间隔的调度的最优调度策略是(4), 流水线最大吞吐率为:  $1/4\Delta t$ 。





## 习题3.10 (单功能非线性流水线调度)

解:

(5) 按调度策略(2,2,5), 连续输入10 个任务的流水线实际吞吐率与加速比分别为:

$$TP_1 = \frac{10}{(2 + 2 + 5 + 2 + 2 + 5 + 2 + 2 + 5 + 7)\Delta t} = \frac{10}{34\Delta t}$$

$$S_1 = \frac{10 \times 7\Delta t}{34\Delta t} \approx 2.06$$

按调度策略(4), 连续输入10 个任务的流水线实际吞吐率与加速比分别为:

$$TP_2 = \frac{10}{(4 \times 9 + 7)\Delta t} = \frac{10}{43\Delta t}$$

$$S_2 = \frac{10 \times 7\Delta t}{43\Delta t} \approx 1.63$$



## 习题3.11 (相关, 定向, 指令调度)

3.11 在MIPS流水线(按照图3.33)上运行右边代码序列:

其中, R3的初始值是 $R2 + 396$ 。假设: 在整个代码序列的运行过程中, 所有的存储器访问都是命中的, 并且在一个时钟周期中对同一个寄存器的读操作和写操作可以通过寄存器“定向”。问:

1. 在没有任何其它定向硬件的支持下, 请画出该指令序列执行的流水线时空图。假设采用排空流水线的策略处理分支指令, 且所有的存储器访问都可以命中Cache, 那么执行上述循环需要多少个时钟周期?
2. 假设该流水线有正常的定向路径, 请画出该指令序列执行的流水线时空图。假设采用预测分支失败的策略处理分支指令, 且所有的存储器访问都可以命中Cache, 那么执行上述循环需要多少个时钟周期?
3. 假设该流水线有正常的定向路径, 请对该循环中的指令进行调度。注意可以重新组织指令的顺序, 也可以修改指令的操作数, 但是不能增加指令的条数。请画出该指令序列执行的流水线时空图, 并计算执行上述循环需要的时钟周期数?

```
LOOP:    LW R1, 0(R2)
          ADDI R1, R1, #1
          SW 0(R2), R1
          ADDI R2, R2, #4
          SUB  R4, R3, R2
          BNZ  R4, LOOP
```

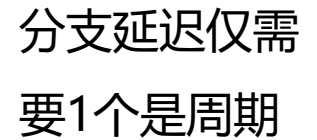
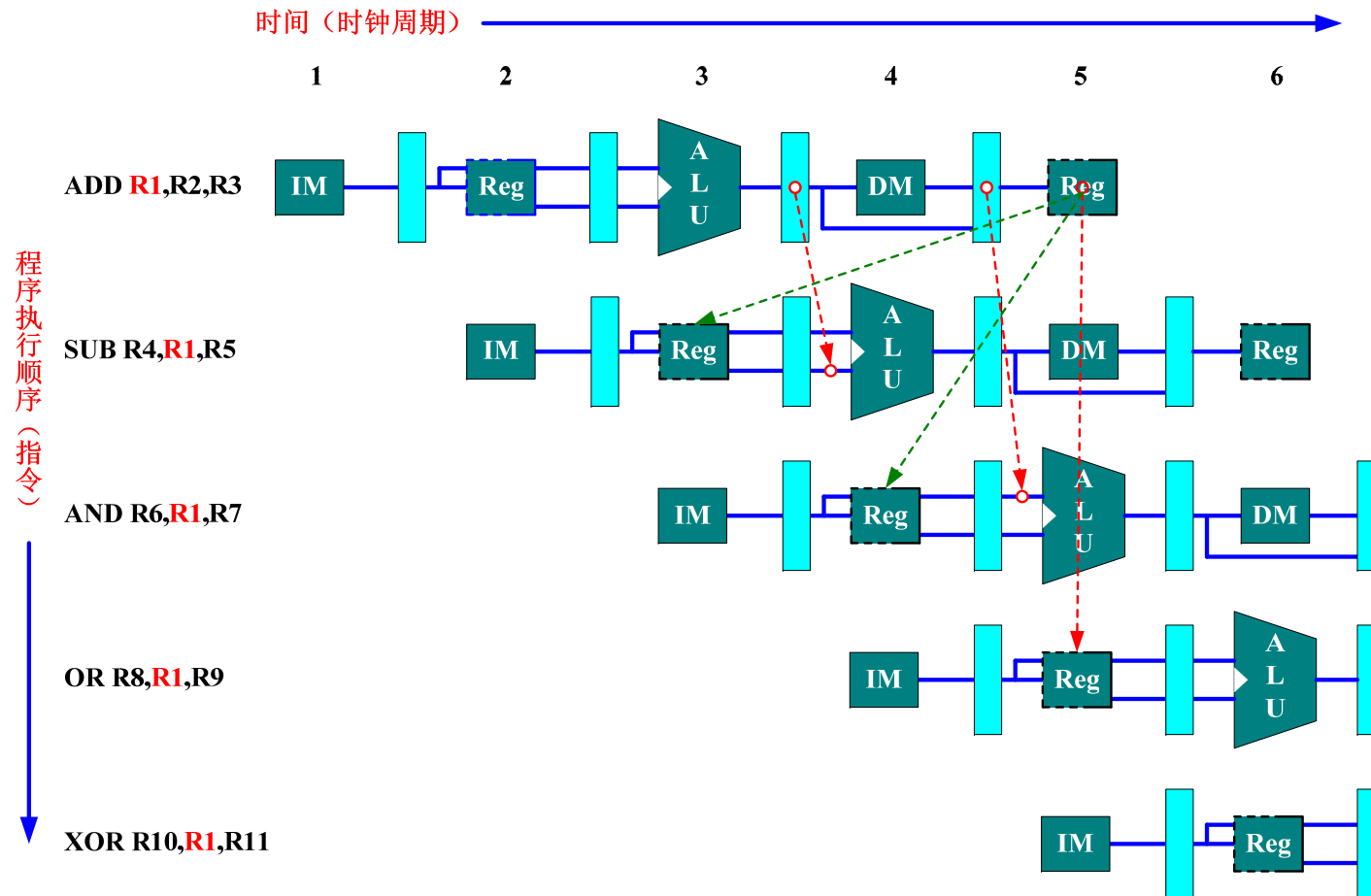


图 3.33 为减少分支延迟,对图 3.32 进行改进后的流水线数据通路



## 习题3.11 (相关, 定向, 指令调度)

采用定向技术消除数据相关



数据重定向, 解决数据相关问题



## 习题3.11 (相关, 定向, 指令调度)

解:

1. **无定向**: 每轮循环从第1条指令开始到下轮循环第1条指令开始 (此时最后一条指令bnez r4,Loop才执行完ID 周期) 为15拍。

末轮循环的最后一条指令 (bnez r4,Loop) 在ID周期后还执行3 拍才结束.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
lw r1,0(r2)	IF	ID	EX	MEM	WB														
daddi r1,r1,1		IF	S	S	ID	EX	MEM	WB											
sw r1,0(r2)					IF	S	S	ID	EX	MEM	WB								
daddi r2,r2,4								IF	ID	EX	MEM	WB							
dsub r4,r3,r2									IF	S	S	ID	EX	MEM	WB				
bnez r4,Loop												IF	S	S	ID	EX	MEM	WB	★
halt															IF				
lw r1,0(r2)																IF	ID	EX	
$15 \times 99 + 3 = 1488$																			

需要进行 $396/4=99$ 次循环, 由于每次分支都清空流水线。从上图可以看出每次循环需要15个时钟周期, 因此总共需要的时钟周期数=  $15 \times 99 + 3 = 1488$

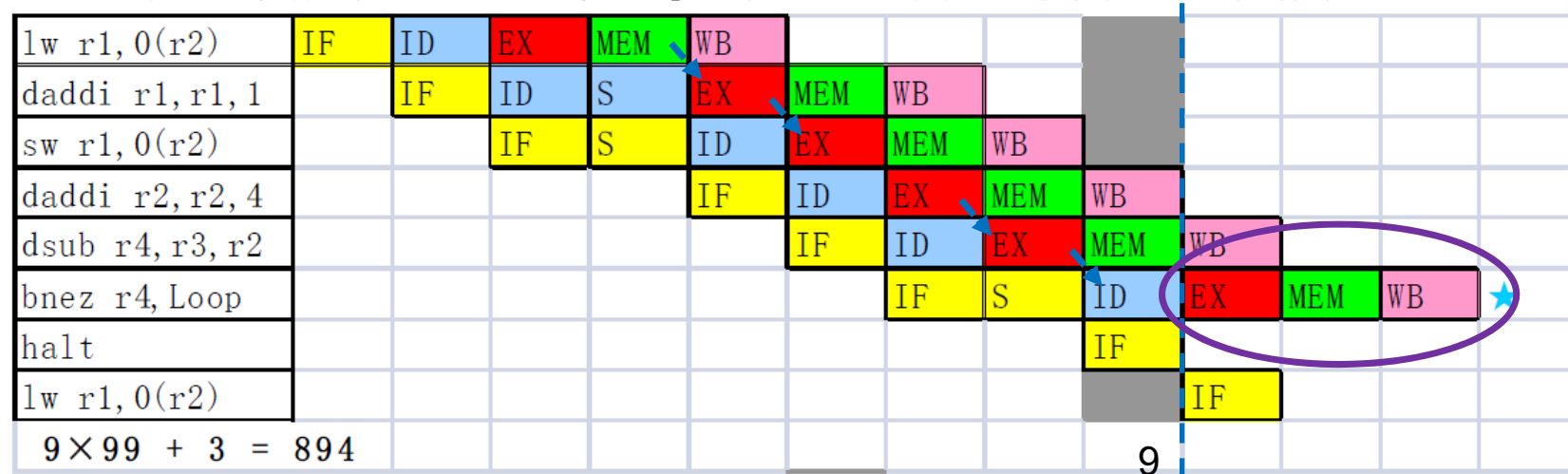


## 习题3.11 (相关, 定向, 指令调度)

2. 采用定向技术, 按“预测分支失败”的策略处理分支指令的

每轮循环第1 条指令开始到下轮循环第1 条指令开始为9拍

末轮循环的最后一条指令 (bnez r4,Loop) 在ID周期后还执行3拍才结束。

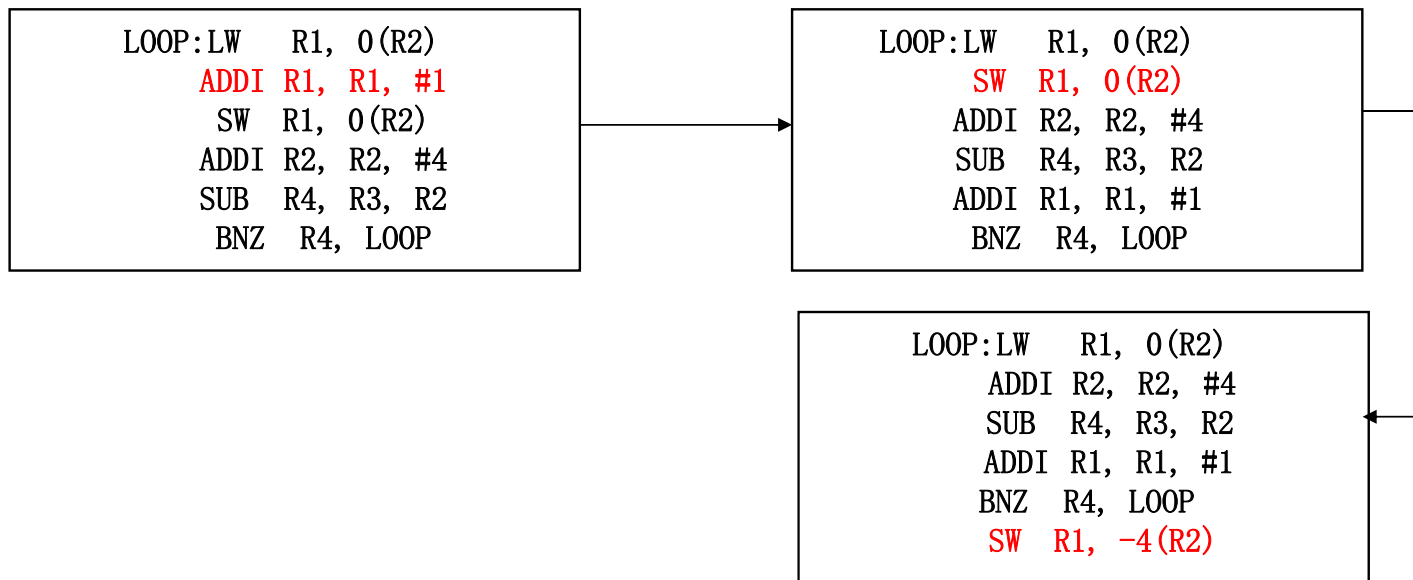


需要进行 $396/4=99$ 次循环, 由于每次分支都清空流水线。从上图可以看出每次循环需要9个时钟周期, 因此总共需要的时钟周期数为 $9 \times 99 + 3 = 894$



## 习题3.11 (相关, 定向, 指令调度)

3. 指令执行重新排序如下:
- |      |            |                     |
|------|------------|---------------------|
| lw   | r1,0(r2)   | ;加法寄存器R1←取数(R2)     |
| addi | r2,r2,#4   | ;指针R2←指针R2+4        |
| sub  | r4,r3,r2   | ;R4←R3-R2           |
| addi | r1,r1,#1   | ;R1←R1+1            |
| bnez | r4,Loop    | ;若R4≠0, 循环          |
| sw   | r1, -4(r2) | ;分支延迟槽, 存数(R2-4)←R1 |





## 习题3.11 (相关, 定向, 指令调度)

3. 采用定向技术、单周期延迟分支、指令调度技术, 重新安排指令顺序如下:

loop: lw r1, 0(r2)

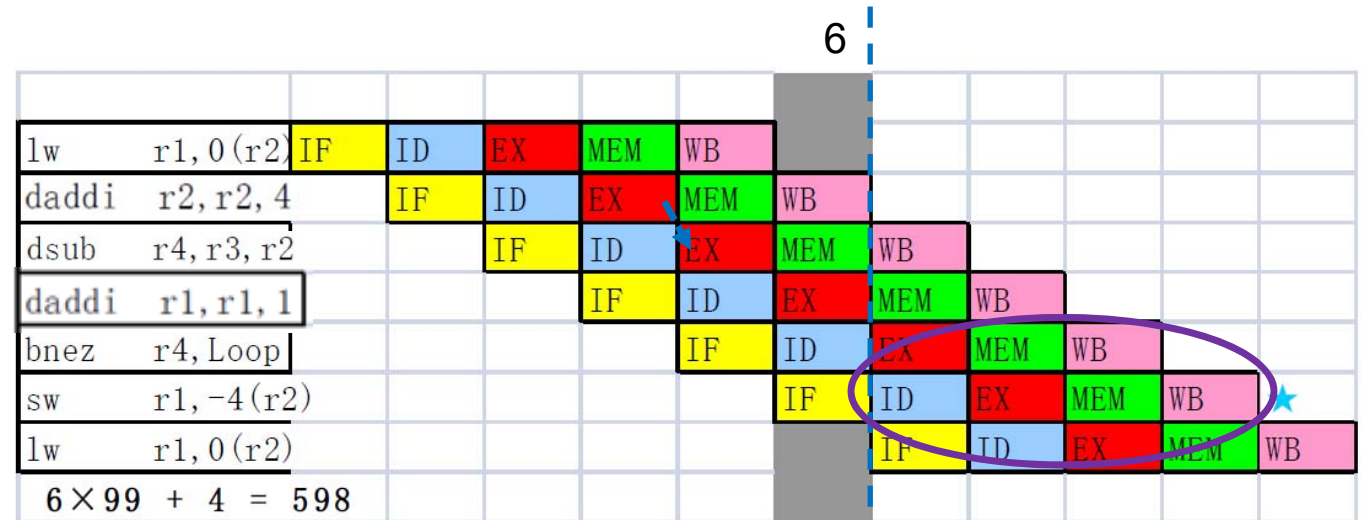
addi r2,r2,#4

sub r4,r3,r2

addi r1,r1,#1

bnez r4,loop

sw r1,-4(r2)



每轮循环从第1 条指令开始到下轮循环第1 条指令开始为6 拍;

末轮循环的最后一条指令 (sw r1,-4(r2)) 在ID周期后还执行4 拍才结束;

总拍数=  $6 \times 99 + 4 = 598$





## 习题5.8 (分支预测技术)

假设有一条长流水线，仅仅对条件转移指令使用分支目标缓冲。假设分支预测错误的开销为4个时钟周期，缓冲不命中的开销为3个时钟周期。假设命中率为90%，预测精度为90%，分支频率为15%，没有分支的基本CPI为1。

(1) 求程序执行的CPI。

(2) 相对固定的2个时钟周期延迟的分支处理，哪种更快？

解：

(1) 程序执行的CPI=没有分支的基本CPI+分支带来的额外开销

额外开销 =  $15\% \times (90\% \text{命中} \times 10\% \text{预测错误} \times 4 + 10\% \text{没命中} \times 3) = 0.099$

所以程序执行的CPI =  $1 + 0.099 = 1.099$ 。

(2) 采用固定的2个时钟周期延迟的分支处理

$$\text{CPI} = 1 + 15\% \times 2 = 1.3$$

由 (1) (2) 知分支目标缓冲方法执行速度快。



## 习题5.9 (分支预测技术)

假定分支目标缓冲的命中率为90%，程序中无条件转移指令为5%，其它指令的CPI为1。假设分支目标缓冲包含分支目标指令,允许无条件转移指令进入分支目标缓冲,则CPI是多少。假定原来的CPI为1.1。

(1) 原来不采用分支目标缓冲器BTB情况下

$$\text{实际CPI} = \text{理想CPI} + \text{各种停顿拍数} = 1 + 5\% \times L = 1.1$$

$$\text{解出 } L = 2$$

(2) 现在采用分支目标缓冲器BTB情况下

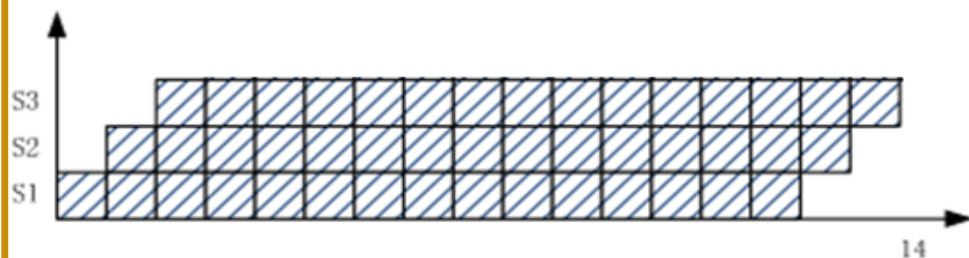
$$\text{实际CPI} = \text{理想CPI} + \text{各种停顿拍数} = 1 + 5\% \times 10\% \times 2 = 1.01$$



## 习题5.11 (超标量/超长指令字/超流水)

设指令流水线由取指令,分析指令和执行指令3个部件构成,每个部件 $\Delta t$ ,连续12条指令,分别画出ILP为4的超标量,超长指令字处理机和超流水线的时空图,并分别计算相对标量流水处理机的加速比.

### 1. 标量流水处理机

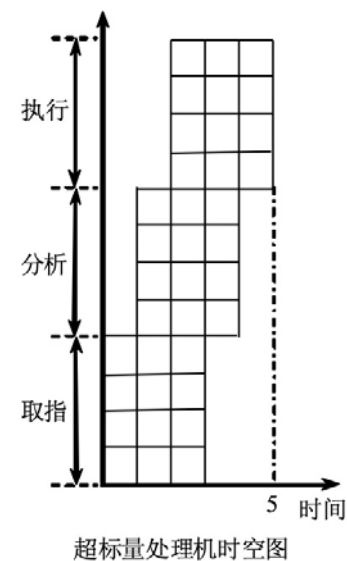


$$T_k = (k+n-1) \Delta t = (3+12-1) \Delta t = 14 \Delta t$$

### 2. 超标量处理机

$$\begin{aligned} \text{ILP}=4, \text{ 执行12条指令 } T_k &= (k+n-1) \Delta t \\ &= (3+3-1) \Delta t = 5 \Delta t \end{aligned}$$

$$\text{加速比 } S = 14 \Delta t / 5 \Delta t = 2.8$$





## 习题5.11 (超标量/超长指令字/超流水)

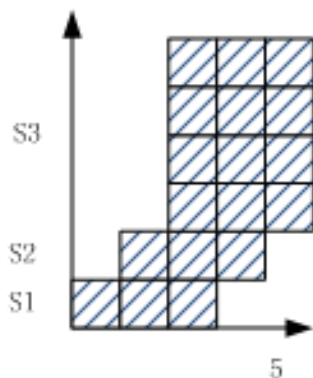
### 3. 超长指令字处理机

采用指令级并行技术, ILP=4,

12个任务组装成3条长指令, 每条含4条小指令,  $n=3$ 。

$$T_k = (k+n-1) \Delta t = (3+3-1) \Delta t = 5 \Delta t,$$

$$\text{加速比 } S = 14 \Delta t / 5 \Delta t = 2.8$$



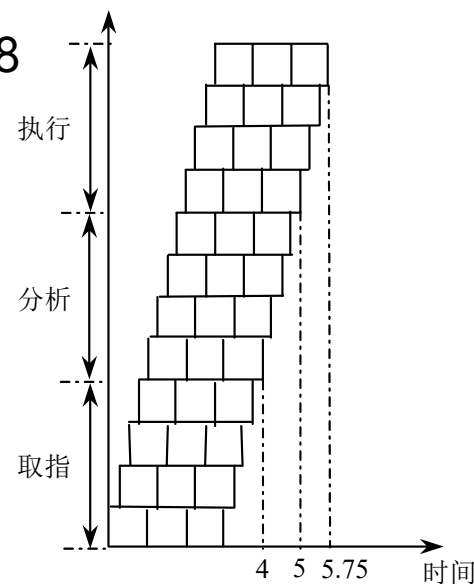
### 4. 超流水线处理机

每1/4个时钟周期启动一条指令。

执行完12条指令需  $T_4 = 5.75 \Delta t$

$$T_k = (k+n-1) \Delta t = (3+3-1) \Delta t = 5 \Delta t,$$

$$\text{加速比 } S = 14 \Delta t / 5 \Delta t = 2.8$$





## 习题7.9 (两级Cache)

假设在3000次访存中，第一级cache不命中110次，第二级cache不命中55次。

试问：在这种情况下，该cache系统的局部不命中率和全局不命中率各是多少？

解：

第一级cache不命中率（全局和局部）是 $110/3000$ ，即3.67%；

第二级cache的局部不命中率是 $55/110$ ，即50%；

第二级cache的全局不命中率是 $55/3000$ ，即1.83%。



## 习题7.10 (存储系统性能指标)

给定以下的假设，试计算直接映象Cache和两路组相联Cache的平均访问时间以及CPU的性能。  
由计算结果能得出什么结论？

- (1)理想Cache情况下的CPI为2.0，时钟周期为2ns，平均每条指令访存1.2次；
- (2)两者Cache容量均为64KB，块大小都是32字节；
- (3)组相联Cache中的多路选择器使CPU的时钟周期增加了10%；
- (4)这两种Cache的失效开销都是80ns；
- (5)命中时间为1个时钟周期；
- (6)64KB直接映象Cache的失效率为1.4%，64KB两路组相联Cache的失效率为1.0%。



## 习题7.10（存储系统性能指标）

平均访存时间 = 命中时间 + 失效率 × 失效开销

平均访问时间1-路 =  $1 \times 2.0 + 1.4\% \times 80 = 3.12\text{ns}$

平均访问时间2-路 =  $1 \times 2.0 \times (1 + 10\%) + 1.0\% \times 80 = 3.0\text{ns}$

两路组相联的平均访存时间比较低

$\text{CPU}_{\text{time}} = (\text{CPU执行周期数} + \text{存储停顿周期数}) \times \text{时钟周期}$

$\text{CPU}_{\text{time}} = (\text{IC} \times \text{CPI执行周期数} + \text{总访存失效次数} \times \text{失效开销}) \times \text{时钟周期}$

$= \text{IC} \times (\text{CPI执行} \times \text{时钟周期} + \text{每条指令的访存次数} \times \text{失效率} \times \text{失效开销} \times \text{时钟周期})$

$\text{CPU}_{\text{time 1-way}} = \text{IC}(2.0 \times 2 + 1.2 \times 0.014 \times 80) = 5.344\text{IC}$

$\text{CPU}_{\text{time 2-way}} = \text{IC}(2.2 \times 2 + 1.2 \times 0.01 \times 80) = 5.36\text{IC}$

相对性能比:  $5.36/5.344 = 1.003$

直接映象的访存时间是两路组相联的1.04倍，两路组相联的平均CPU时间是直接映象的1.003倍。

因此这里选择直接映象。



## 习题7.11 (伪相联)

伪相联中，假设在直接映象位置没有发现匹配，而在另一个位置才找到数据（伪命中）时，需要1个额外的周期，而且不交换两个Cache中的数据，失效开销为50个时钟周期。

假设 2KB直接映象Cache的总失效率为0.098，2路相联的总失效率为0.076；

128KB直接映象Cache的总失效率为0.010，2路相联的总失效率为0.007。

试求：

- (1) 推导出平均访存的时间公式。
- (2) 利用 (1) 中得到的公式，对于2KBCache和128KBCache，重新计算伪相联的平均访存时间。请问哪一种伪相联更快？





## 习题7.11 (伪相联)

$$\text{命中时间}_{\text{伪相联}} = \text{命中时间1路} + \text{伪命中率}_{\text{伪相联}} \times 1$$

$$\begin{aligned}\text{伪命中率}_{\text{伪相联}} &= \text{命中率2路} - \text{命中率1路} = (1 - \text{失效率2路}) - (1 - \text{失效率1路}) \\ &= \text{失效率1路} - \text{失效率2路}.\end{aligned}$$

$$\text{平均访存时间}_{\text{伪相联}}$$

$$= \text{命中时间1路} + (\text{失效率1路} - \text{失效率2路}) \times 1 + \text{失效率2路} \times \text{失效开销2路}$$

$$\text{平均访存时间2KB} = 1 + (0.098 - 0.076) \times 1 + (0.076 \times 50) = 4.822$$

$$\text{平均访存时间128KB} = 1 + (0.010 - 0.007) \times 1 + (0.007 \times 50) = 1.353$$

128KB的伪相联Cache要快一些。



## 习题7.12 ( TLB )

假设采用理想存储器系统时的基本CPI是1.5，主存延迟是40个时钟周期；传输速率为4B/时钟周期，且Cache中50%的块修改过的。每个块中有32B，20%的指令是数据传送指令。并假设没有写缓存，在TLB不命中的情况下需要20时钟周期，TLB不会降低Cache命中率。CPU产生指令地址或Cache不命中时产生的地址有0.2%没有在TLB中找到。

(1) 在理想TLB情况下，计算均采用写回法16KB直接映像混合Cache、16KB两路组相联混合Cache和32KB直接映像混合Cache机器的实际CPI。

(2) 在实际TLB情况下，用(1)的结果，计算均采用写回法16KB直接映像混合Cache、16KB两路组相联混合Cache和32KB直接映像混合Cache机器的实际CPI。

其中假设16KB直接映像混合Cache、16KB两路组相联混合Cache和32KB直接映像混合Cache的不命中率分别为2.9%、2.2%和2.0%；25%的访存为写访存。



## 习题7.12 ( TLB )

方法一：

(1) 假设TLB不命中率=0

Cache中50%的块修改过，所以不命中时，替换Cache需要1次从内存取一块，50%次写回一块，共1.5次。

均摊不命中开销=不命中率 $\times$ 1.5  $\times$  [40+32B/4B+0 $\times$ 20]=不命中率 $\times$ 72

实际CPI1=1.5+1.2 $\times$ 不命中率 $\times$ 72=1.5+不命中率 $\times$ 86.4

3种Cache结构的不命中率得：

Cache结构	不命中率	实际CPI
16KB直接混合映像	0.029	4.0056
16KB两路混合映像	0.022	3.4008
32KB直接混合映像	0.020	3.2280



## 习题7.12 ( TLB )

(2) 假设TLB不命中率=0.2%

均摊不命中开销= 不命中率 $\times$ 1.5 $\times$ [40+32B/4B+0.2% $\times$ 20]

= 不命中率 $\times$ 1.5 $\times$ 48.04=不命中率 $\times$ 72.06

实际CPI2 = 1.5+1.2 $\times$ 不命中率 $\times$ 72.06 = 1.5+不命中率 $\times$ 86.472

3种Cache结构的不命中率后得

Cache结构	不命中率	实际CPI
16KB直接混合映像	0.029	4.0077
16KB两路混合映像	0.022	3.4024
32KB直接混合映像	0.020	3.2294



## 习题7.12 ( TLB )

方法二：

存储停顿主要由主存中取指令所引起，而load 和store指令访问数据由TLB 引起。

$CPI = CPI_{\text{执行}} + \text{存储停顿周期数} / \text{指令数}$

$$\frac{\text{存储停顿周期数}}{\text{指令数}} = \frac{\text{取指令停顿}}{\text{指令数}} + \frac{\text{数据访问停顿} + \text{TLB停顿}}{\text{指令数}}$$
$$\frac{\text{停顿周期数}}{\text{指令数}} = \frac{\text{存储访问}}{\text{指令数}} \times \text{失效率} \times \text{失效开销}$$

(1) 对于理想TLB，TLB 失效开销为0。

对于统一Cache，若读/写失效且块是干净的，失效开销为

$$P_{\text{干净数据}} = \text{主存延迟} + \text{传输一个块需要使用的的时间} = 40 + 32/4 = 48 \text{ (拍)}$$

若为读/写失效且块是脏的，失效开销为

$$P_{\text{脏数据}} = (\text{主存延迟} + \text{传输一个块需要使用的的时间}) \times 2 = 96 \text{ (拍)}$$

$$CPI = 1.5 + (1 + 20\%) \times \text{失效率} \times (50\% \times 48 + 50\% \times 96) = 1.5 + 1.2 \times R \times 72$$



## 习题7.12 ( TLB )

$$\frac{TLB\text{停顿}}{\text{指令数}} = \left( \frac{TLB\text{访问次数}}{\text{指令数}} \right) \times TLB\text{失效率} \times TLB\text{失效开销}$$

设 $f_{\text{数据}}$ 为数据访问指令频率),  $R_t$ 和 $P_t$ 分别是TLB 的失效率和失效开销,  $R_c$ 是Cache的失效率, 在Cache命中时, TLB 不会增加命中时间, 因此, 对CPI产生影响的TLB停顿/指令数为:

$$\frac{TLB\text{停顿}}{\text{指令数}} = (1 + f_{\text{数据}}) \times R_c \times R_t \times P_t \times (1 + 50\%)$$

其中,  $(1 + f_{\text{数据}})$ : 每条指令的访问内存次数。

由条件得:  $TLB\text{停顿}/\text{指令数} = (1 + 20\%) \times R_c \times 0.2\% \times 20 \times 1.5 = 0.072R_c$



## 习题7.14 (写策略)

假设一台计算机具有以下特性：

95%的访存在Cache中命中；块大小为两个字，且失效时整个块被调入；CPU发出访存请求的速率为 $10^9$ 字/秒；25%的访存为写访问；存储器的最大流量为 $10^9$ 字/秒（包括读和写）；

主存每次只能读或写一个字；在任何时候，Cache中有30%的块被修改过；

写失效时，Cache采用写分配法。

试对于以下两种情况计算主存频带的平均使用比例。

(1) 写直达Cache；

(2) 写回法Cache。



## 习题7.14 (写策略)

采用按写分配

(1) 写直达cache:

读命中, 不访问主存;

写命中, 更新cache和主存, 访问主存一次。

读失效, 将主存中的块调入cache中, 访问主存两次;

写失效, 将要写的块调入cache, 访问主存两次, 再将修改的数据写入cache和主存, 访问主存一次, 共三次。上述分析如下表所示:

访问命中	访问类型	频率	访存次数
Y	读	$95\% \times 75\% = 71.3\%$	0
Y	写	$95\% \times 25\% = 23.8\%$	1
N	读	$5\% \times 75\% = 3.8\%$	2
N	写	$5\% \times 25\% = 1.3\%$	3

一次访存请求最后真正的平均访存次数  
 $= (71.3\% \times 0) + (23.8\% \times 1) + (3.8\% \times 2) + (1.3\% \times 3)$   
 $= 0.35$   
已用带宽  $= 0.35 \times 10^9 / 10^9 = 35.0\%$





## 习题7.14 (写策略)

(2) 写回法cache访问命中,有两种情况:

读命中, 不访问主存;

写命中, 采用写回法, 不访问主存。

访问失效,有一个块将被换出, 这也有两种情况:

如果被替换的块没有修改过, 将主存中的块调入cache块中, 访存两次;

如果被替换的块修改过, 则首先将修改的块写入主存, 需要访存两次; 然后将主存中的块调入cache块中, 需要访问主存两次, 共四次访存。

$$\begin{aligned} &\text{一次访存请求最后真正的平均访存次数} \\ &= 66.5\% * 0 + 28.5\% * 0 + 3.5\% * 2 + 1.5\% * 4 \\ &= 0.13 \\ &\text{已用带宽} = 0.13 \times 10^9 / 10^9 = 13\% \end{aligned}$$

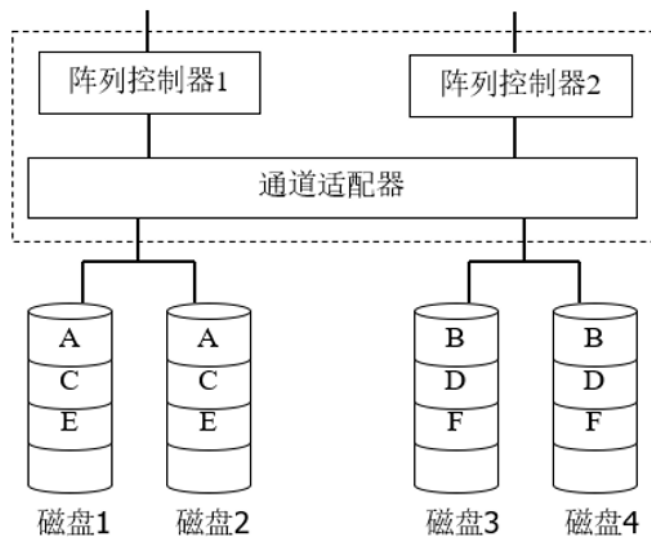
访问命中	块为脏	频率	访存次数
Y	N	$95\% * 70\% = 66.5\%$	0
Y	Y	$95\% * 30\% = 28.5\%$	0
N	N	$5\% * 70\% = 3.5\%$	2
N	Y	$5\% * 30\% = 1.5\%$	4



## 第八章 补充习题

一个廉价磁盘冗余阵列由4个磁盘配置为RAID 10级，其结构如图，采用双控制器（RC）结构，任何一个阵列控制器失效不影响系统工作。已知各部分可靠度为：阵列控制器 $R_1=0.9$ ，通道适配器 $R_2=0.95$ ，磁盘 $R_3=0.95$ 。画出系统可靠性框图；

- (1) 写出系统可靠性 $R$ 的表达式；
- (2) 计算 $R$ 的数值（保留小数点后两位）；

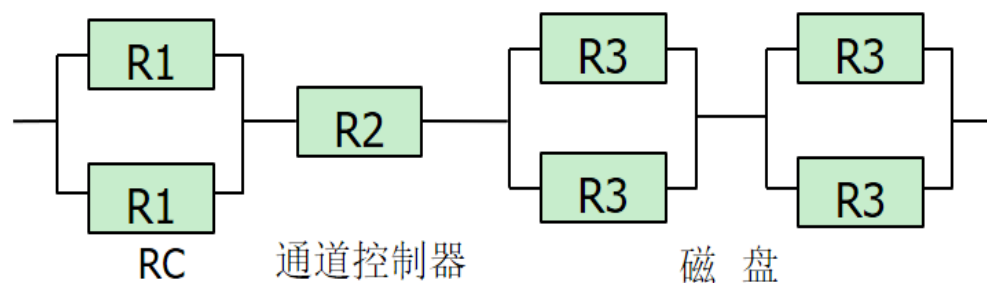




## 第八章 补充习题

解：

(1) 可靠性模型如下：



(2) 系统可靠度  $R = (1 - (1 - R_1)^2) \times R_2 \times [1 - (1 - R_3)^2]^2$

(3)  $R = [1 - (1 - R_1)^2] \times R_2 \times [1 - (1 - R_3)^2]^2 = 0.99 \times 0.95 \times [1 - (1 - 0.95)^2]^2 = 0.94$



## 第9章---各种互连函数总结

交换置换函数定义:  $E(X_{n-1}X_{n-2}\dots X_1X_0) = X_{n-1}X_{n-2}\dots X_1X_0$ , 其中  $0 \leq i \leq n-1$

立方体函数定义: Cube<sub>i</sub>的功能是对入端结点编号二进制形式的第*i*位取反

Cube<sub>i</sub>( $X_{n-1}\dots X_{i+1}X_iX_{i-1}\dots X_0$ ) =  $X_{n-1}\dots X_{i+1}X_iX_{i-1}\dots X_0$ , 其中  $0 \leq i \leq n-1$

均匀洗牌  $\sigma$ : shuffle ( $X_{n-1}X_{n-2}\dots X_0$ ) =  $X_{n-2}\dots X_0X_{n-1}$  (循环左移)

蝶式互连函数butterfly  $\beta$ : 最高位与最低位互换;

反位序函数  $\rho$ : 就是二进制各位次序颠倒过来;

PM2I函数定义: PM2<sub>±i</sub>的功能是对入端结点编号加或减 $2^i$ , 然后再作模N运算

$$\text{PM2}_{+i}(X) = X + 2^i \bmod N$$

$$\text{PM2}_{-i}(X) = X - 2^i \bmod N$$

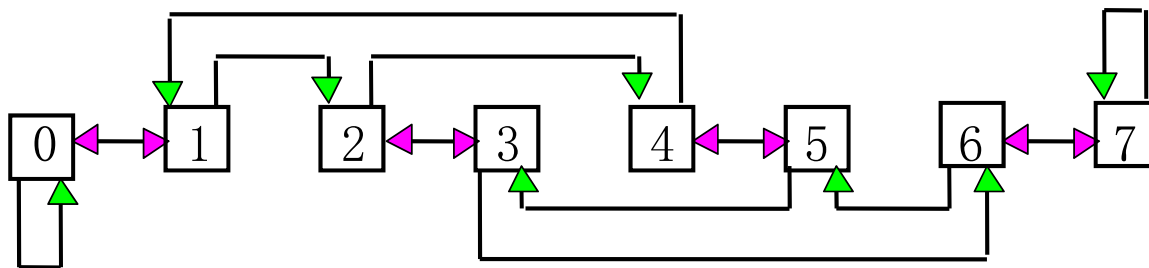
其中  $X = 0 \sim N-1$ ,  $i = 0 \sim n-1$



## 第9章---几个单级静态网络参数

### ✓ 单级混洗—交换网络

- 网络的直径是 $2n-1$ 。
- 0和 $N-1$ 的入度和出度各为1，其它的各为2；



### ✓ 单级PM2I网络

- $2n-1$ 种不同的置换，度为 $2n-1$ ；
- 单级PM2I网络的直径是

✓  $n$ -立方体中结点的度都是 $n$ ，直径也是 $n$



## 习题9.9

9.9 设32个处理器编号为0、1、...、31,

(1) 分别计算下列互连函数:

(2) 用 $Cube_0$ 和 $\sigma$ 构成混洗交换网 (每步只能使用 $Cube_0$ 或 $\sigma$ 一次), 网络直径是多少? 从5号处理机发送数据到7号处理机, 最短路径要经过几步? 请列出经过的处理机编号。

(1)

$$Cube_2(12) = Cube_2(01100B) = 01000B = 8$$

$$\sigma(8) = \sigma(01000B) = 10000B = 16$$

$$\beta(9) = \beta(01001B) = 11000B = 24$$

$$PM\ 2I_{+3}(28) = PM\ 2I_{+3}(11100B) = 11100B + 01000B \bmod 2^5 = 00100B = 4$$

$$Cube_0(\sigma(4)) = Cube_0(\sigma(00100B)) = 01001B = 9$$

$$\sigma(Cube_0(18)) = \sigma(Cube_0(10010B)) = 00111B = 7$$



## 习题9.9

(2)

$2^5$ 个结点的混洗交换网的直径是 $2n-1 = 2 \times 5 - 1 = 9$ ;

从5号处理机 (00101B) 发送数据到7号处理机 (00111B), 最短路径要经过6步, 包含5步左移和1步求反 (因为 $00101B \oplus 00111B = 00010B$ ), 经过的处理机编号为:  $00101B \rightarrow 01010B \rightarrow 10100B \rightarrow 01001B \rightarrow 10010B \rightarrow 10011B \rightarrow 00111B$

(3)

网络直径是 $\lceil 5/2 \rceil = 3$ ;

结点度是 $2n-1 = 2 \times 5 - 1 = 9$ ;

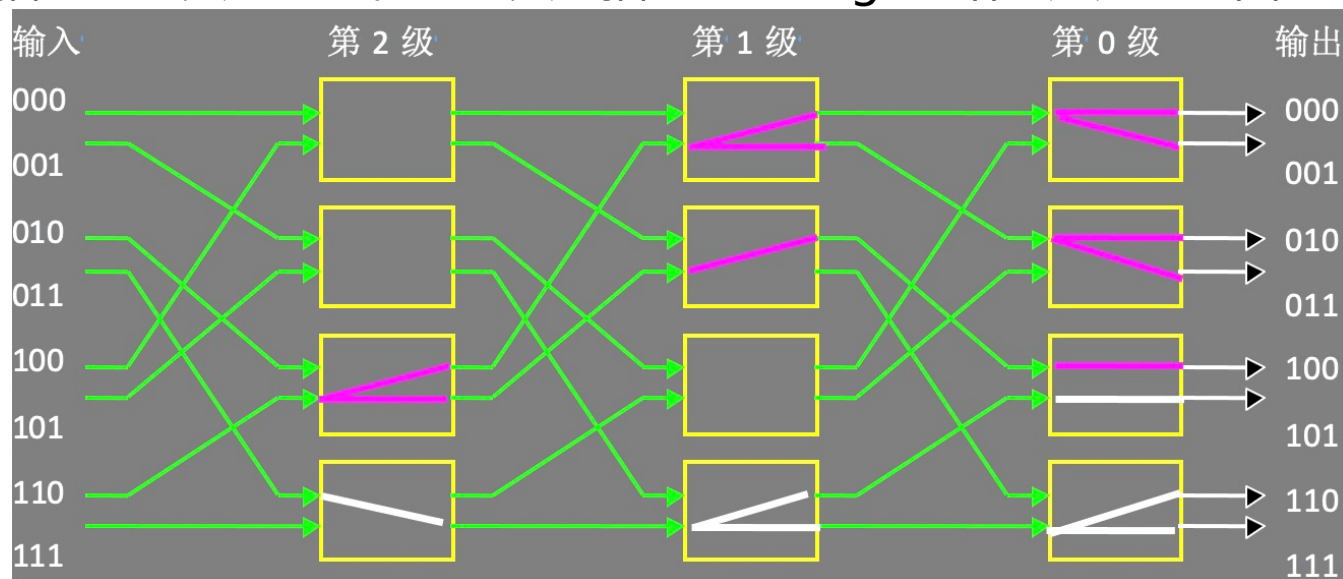
与2号处理机距离最远的是13、15、21、23号处理机。



## 习题9.13 (多级互连网络)

有 $\log_2 N$ 级，每级用 $N/2$ 个 $2 \times 2$ 开关，共需要 $N/2 \cdot \log_2 N$ 个开关。

用一个 $N=8$ 的三级Omega网络连接8个处理机（ $P_0 \sim P_7$ ），8个处理机的输出端分别依次连接Omega的8个输入端 $0 \sim 7$ ，8个处理机的输入端分别依次连接Omega的8个输出端 $0 \sim 7$ ，如果处理机 $P_6$ 要把数据播送到处理机 $P_0 \sim P_4$ ，处理机 $P_3$ 要把数据播送到处理机 $P_5 \sim P_7$ ，那么，Omega网络能否同时为它们的播送要求实现连接，画出实现播送的Omega网络的开关状态图。







## 习题10.6 (并行处理对性能的提高)

一个具有32台处理机的系统，对远程存储器访问时间是2000ns。除了通信以外，假设计算中的访问均命中局部存储器。当发出一个远程请求时，本地处理机挂起。处理机的时钟周期是10ns，假设指令基本的CPI为1.0（假设所有访存均命中cache）。对于下述两种情况：

- (1) 没有远程访问；
- (2) 0.5%的指令需要远程访问。

试问前者比后者快多少？

解：已知远程访问率  $p = 0.5\%$ ，远程访问时间  $t = 2000\text{ns}$ ，时钟周期  $T = 10\text{ns}$

远程访问开销  $C = t/T = 2000\text{ns}/10\text{ns} = 200$ （时钟周期数）

有0.5%远程访问的机器的实际CPI2为： $\text{CPI2} = \text{CPI1} + p \times C = 1.0 + 0.5\% \times 200 = 2.0$

只有局部访问的机器的基本  $\text{CPI1} = 1.0$ ， $\text{CPI2}/\text{CPI1} = 2.0/1.0 = 2$ （倍）

因此，没有远程访问状态下的机器速度是有0.5% 远程访问的机器速度的2倍。

谢谢!

