

《计算机系统结构》习题解答(清华 2 版)

目录

第一章 (P33)

1.7-1.9 (透明性概念), 1.12-1.18 (Amdahl 定律), 1.19、1.21、1.24 (CPI/MIPS)

第二章 (P124)

2.3、2.5、2.6 (浮点数性能), 2.13、2.15 (指令编码)

第三章 (P202)

3.3 (存储层次性能), 3.5 (并行主存系统), 3.15-3.15 加 1 题 (堆栈模拟), 3.19 中 (3) (4) (6) (8) 问 (地址映象/替换算法—实存状况图)

第四章 (P250)

4.5 (中断屏蔽字表/中断过程示意图), 4.8 (通道流量计算/通道时间图)

第五章 (P343)

5.9 (流水线性能/时空图), 5.15 (2 种调度算法)

第六章 (P391)

6.6 (向量流水时间计算), 6.10 (Amdahl 定律/MFLOPS)

第七章 (P446)

7.3、7.29 (互连函数计算), 7.6-7.14 (互连网性质), 7.4、7.5、7.26 (多级网寻径算法), 7.27 (寻径/选播算法)

第八章 (P498)

8.12 (SISD/SIMD 算法)

第九章 (P562)

9.18 (SISD/多功能部件/SIMD/MIMD 算法)

(注: 每章可选 1-2 个主要知识点, 每个知识点可只选 1 题。有下划线者为推荐的主要知识点。)

第一章 (P33)

1.7

(1) 从指定角度来看，不必要了解的知识称为透明性概念。

(2) 见下表，“√”为透明性概念，“P”表示相关课文页数。

模 m 交叉，√，P5	浮点数据，×，P4	通道与 I/O 处理机，×，P4
总线宽度，√，P5	阵列运算部件，×，P455	结合型与独立型通道，√？，李
单总线，√，P4	访问保护，×，P4	中断，×，P4
指令控制方式，√，P5	堆栈指令，×，P4	最小编址单位，×，P4
Cache 存储器，√，P131		

1.8 见下表，“√”为透明性概念，“P”表示相关课文页数。

指令地址寄存器，×，P5	指令缓冲器，√，P5	时标发生器，√，P5
条件码寄存器，×，P4	乘法器，√，李	主存地址寄存器，√，P5
磁盘，×，P4	先行进位链，√，P5	移位器，√，李
通用寄存器，×，P4	中断字寄存器，×，P4	

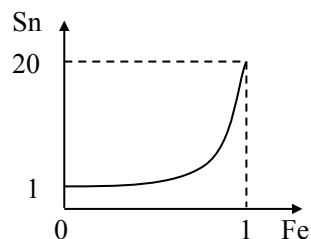
1.9 见下表，“√”表示都透明，“应”表示仅对应用程序员透明，“×”表示都不透明。

数据通路宽度，√，P5	虚拟存储器，应，P131	Cache 存储器，√，P131
程序状态字，×，P4	“启动 I/O”指令，应，P236	“执行”指令，×，P246
指令缓冲寄存器，√，P5		

1.12 已知 $Se=20$ ，求作 $Fe-Sn$ 关系曲线。

将 Se 代入 Amdahl 定律得

$$S_n = \frac{1}{1 - \frac{19}{20} F_e}$$



1.13 上式中令 $Sn=2$ ，解出 $Fe=10/19 \approx 0.526$

1.14 上式中令 $Sn=10$ ，解出 $Fe=18/19 \approx 0.947$

1.15 已知两种方法可使性能得到相同的提高，问哪一种方法更好。

(1) 用硬件组方法，已知 $Se=40$ ， $Fe=0.7$ ，解出 $Sn=40/12.7 \approx 3.1496$ （两种方法得到的相同性能）

(2) 用软件组方法，已知 $Se=20$ ， $Sn=40/12.7$ ，解出 $Fe=27.3/38 \approx 0.7184$ （第二种方法的百分比）

(3) 结论：软件组方法更好。因为硬件组需要将 Se 再提高 100% ($20 \rightarrow 40$)，而软件组只需将 Fe 再提高 1.84% ($0.7 \rightarrow 0.7184$)。

$$1.17 \quad S_n = \frac{1}{0.1 + \frac{0.9}{5}} = \frac{5}{1.4} \approx 3.57$$

1.18 记 f —— 时钟频率, $T=1/f$ —— 时钟周期, B —— 带宽 (Byte/s)。

$$\text{方案一: } B_1 = \frac{1 \times 4}{T} = 4f(\text{Byte/s})$$

$$\text{方案二: } B_2 = \frac{75\% \times 2 + 25\% \times 1}{2T} \times 4 = 3.5f(\text{Byte/s})$$

1.19 由各种指令条数可以得到总条数, 以及各百分比, 然后代公式计算。

$$IC = \sum_{i=1}^4 IC_i = 10^5$$

$$(1) \quad CPI = \sum_{i=1}^4 (CPI_i \times \frac{IC_i}{IC}) = 1 \times 0.45 + 2 \times 0.32 + 2 \times 0.15 + 2 \times 0.08 = 1.55$$

$$(2) \quad MIPS = \frac{f}{CPI \times 10^6} = \frac{40 \times 10^6}{1.55 \times 10^6} = \frac{40}{1.55} \approx 25.806$$

$$(3) \quad T = \frac{IC}{MIPS \times 10^6} = \frac{1.55}{400} \approx 0.003876 \text{ (秒)}$$

1.21

$$(1) \quad CPI = 1 \times 0.6 + 2 \times 0.18 + 4 \times 0.12 + 8 \times 0.1 = 2.24$$

$$(2) \quad MIPS = \frac{f}{CPI \times 10^6} = \frac{40 \times 10^6}{2.24 \times 10^6} \approx 17.86$$

1.24 记 T_c —— 新方案时钟周期, 已知 $CPI = CPI_i = 1$

原时间 = $CPI \times IC \times 0.95T_c = 0.95IC \times T_c$

新时间 = $(0.3 \times 2/3 + 0.7) \times IC \times T_c = 0.9IC \times T_c$

二者比较, 新时间较短。

第二章 (P124)

2.3 (忽略 P124 倒 1 行 ~ P125 第 8 行文字, 以简化题意) 已知 2 种浮点数, 求性能指标。

此题关键是分析阶码、尾数各自的最大值、最小值。

原图为数据在内存中的格式, 阶码的小数点在其右端, 尾数的小数点在其左端, 遵守规格化

要求。

由于尾数均为原码，原码的绝对值与符号位无关，所以最大正数与最小负数的绝对值相同，可用“±最大绝对值”回答；最小正数与最大负数的绝对值相同，可用“±最小绝对值”回答。

第 1 小问中，阶码全部位数为 8，作无符号数看待真值为 0~255，作移-127 码看待真值为-127~+128；尾数（不计符号位）有 23 位小数，另加 1 位整数隐藏位，所以尾数绝对值为 $1.0 \sim 2.0 - 2^{-23}$ ，有效位数 $p=24$ ；

第 2 小问中，阶码全部位数为 11，作无符号数看待真值为 0~2047，作移-1023 码看待真值为-1023~+1024；尾数（不计符号位）有 52 位小数，另加 1 位整数隐藏位，所以尾数绝对值为 $1.0 \sim 2.0 - 2^{-52}$ ，有效位数 $p=53$ 。

最大绝对值为最大阶码与最大尾数绝对值的组合，最小绝对值为最小阶码与最小尾数绝对值的组合。代入相关公式后得最终结果如下表。

	32 位	64 位
±最大绝对值	$\pm (1-2^{-24}) \cdot 2^{129}$	$\pm (1-2^{-53}) \cdot 2^{1025}$
±最小绝对值	$\pm 2^{-127}$	$\pm 2^{-1023}$
表数精度 δ	2^{-24}	2^{-53}
表数效率 η	100%	100%

注：如果修改题目，将 1、2 小问的尾数规定为纯小数（即 1 位隐藏位是小数点后第 1 位），则尾数真值降为原值的 1/2，全部结果改为下表。

	32 位	64 位
±最大绝对值	$\pm (1-2^{-24}) \cdot 2^{128}$	$\pm (1-2^{-53}) \cdot 2^{1024}$
±最小绝对值	$\pm 2^{-128}$	$\pm 2^{-1024}$
表数精度 δ	2^{-24}	2^{-53}
表数效率 η	100%	100%

2.5

- (1) $r_m = 2, r_e = 2, p = 24$ （隐藏最高位）, $q = 7$ 。
(2) $N_{\max} = 1.7 \times 10^{38}, -|N|_{\min} = -1.47 \times 10^{-39}$
 $\delta \leq 5.96 \times 10^{-8} \approx 10^{-7.22}, \eta = 100\%$

2.6

- (1) $0.2 = 0.333333H \times 16^0$

设阶码为移-63 码（即 -2^6+1 ，原题未指明）

1 位	7 位	6 位
0	0111111	333333

$0.2 = 0.110011001100110011001101B \times 2^{-2}$

（其中最高有效位需隐藏）

阶码为移-127 码（即 -2^7+1 ）

1 位	8 位	23 位
0	0111110 1	10011001100110011001101

- (2) 符号位不变，（阶码 - 63）× 4 + 127；尾数左规，除去最高位；
(3) 符号位不变，（阶码 - 127）/ 4 + 63；尾数补最高位，按除法余数右移若干位，左补 0。

2.13 已知 10 条指令使用频度，求 3 种编码方法的平均码长与信息冗余量。

(1)此问中的“最优 Huffman 编码法”实际是指码长下限，即信源的平均信息量——熵，代公式得 $H=2.9566$ 。

(2)Huffman 编码性能如下表；

(3)2/8 扩展编码是 8/64/512 法的变种，第一组 2 条指令，码长为 2（1 位扩展标志，1 位编码），第二组 8 条指令，码长为 4（1 位扩展标志，与第一组区别，加 3 位编码），编码性能如下表；

(4)3/7 扩展编码是 15/15/15 法的变种，第一组 3 条指令，码长为 2（共有 4 种组合，其中 3 种组合分别代表 3 条指令，留 1 种组合作为扩展前缀标志），第二组 7 条指令，码长为 5（2 位固定的前缀扩展标志，与第一组区别，加 3 位编码，只用其中 7 种组合），编码性能如下表。

	Huffman 编码	2/8 扩展编码	3/7 扩展编码
平均码长 \bar{L}	2.99	3.1	3.2
信息冗余量 R	1.10%	4.61%	7.59%

2.15

(1) 15 条/63 条/64 条

(2) 14 条/126 条/128 条

第三章 (P202)

3.3 直接代公式计算存储层次性能指标。

(1)74ns, 38ns, 23.6ns

(2)0.258, 0.315, 0.424（单位：美元/K 字节，换算成美元/字节还要除以 1024）

(3) $T_{64K} > T_{128K} > T_{256K}$

$C_{64K} < C_{128K} < C_{256K}$

(4)19.092, 11.97, 10.0064。答：256K 方案最优。

3.5 已知 $\overline{K_n} = \frac{1-(1-g)^n}{g}$ ，其中 $g=0.1$

依题意有 $\overline{K_{n+1}} = \frac{1-(1-g)^{n+1}}{g} \geq \overline{K_n} + 0.2 = \frac{1-(1-g)^n}{g} + 0.2$

整理得 $0.9^n \geq 0.2$ ，解出 $n \leq \frac{\lg 0.2}{\lg 0.9} \approx 15.28$ ，向下取整，得 15；

按另一种题意理解是向上取整，得 16，也对。

3.15 欲知可能的最高命中率及所需的最少主存页数，较好的办法是通过“堆栈模拟法”，求得命中次数随主存页数变化的函数关系。下图就是“堆栈模拟图”，其中“√”表示命中。

P=	4	5	3	2	5	1	3	2	3	5	1	3	命中次数
	4	5	3	2	5	1	3	2	3	5	1	3	
		4	5	3	2	5	1	3	2	3	5	1	
			4	5	3	2	5	1	1	2	3	5	
				4	4	3	2	5	5	1	2	2	
						4	4	4	4	4	4	4	

n=1													0
n=2									√				1
n=3					√				√			√	3
n=4					√		√	√	√	√	√	√	7
n=5					√		√	√	√	√	√	√	7

(1) $H_{\max} = 7/12 \approx 58.3\%$

(2) $n=4$

(3) 当 1 次页面访问代表连续 1024 次该页内存存储单元访问时，后 1023 次单元访问肯定是命中的，而第 1 次单元访问的命中情况与这 1 次页面访问的命中情况相同。根据上图中最高命中情况，共有 7 次页命中（折算为 7×1024 次单元命中），5 次页不命中（折算为 5×1023 次单元命中，也可写为 $5 \times 1024 - 5$ ），单元访问总次数为 12×1024 ，故有：

$$H_{\text{cell}} = (12 \times 1024 - 5) / (12 \times 1024) = 12283 / 12288 \approx 99.96\%$$

3.15 加 1 题 一个二级存储层次，采用全相联映象和最久没有使用算法，实存共 5 页，为 2 道程序分享，页地址流分别如下

$$P_1 = 1 \quad 2 \quad 3 \quad 4 \quad 1 \quad 3 \quad 2 \quad 1$$

$$P_2 = 1 \quad 2 \quad 3 \quad 4 \quad 2 \quad 2 \quad 3 \quad 3$$

试作 2 个实存分配方案，分别使 2 道程序满足

(1) 命中率相同；

(2) 命中次数之和最大。

解：分别为 2 道程序作“堆栈模拟图”，其中“√”表示命中。

$$P_1 = \quad 1 \quad 2 \quad 3 \quad 4 \quad 1 \quad 3 \quad 2 \quad 1 \quad \text{命中次数} \quad N_{(1)}$$

1	2	3	4	1	3	2	1
	1	2	3	4	1	3	2
		1	2	3	4	1	3
			1	2	2	4	4

$n_1 = 1$								0
$n_1 = 2$								0
$n_1 = 3$					√		√	2
$n_1 = 4$				√	√	√	√	4

$P_2 =$	1	2	3	4	2	2	3	3	命中次数
	$N_{(2)}$								
	1	2	3	4	2	2	3	3	
		1	2	3	4	4	2	2	
			1	2	3	3	4	4	
				1	1	1	1	1	

$n_2 = 1$						✓		✓	2
$n_2 = 2$						✓		✓	2
$n_2 = 3$					✓	✓	✓	✓	4
$n_2 = 4$					✓	✓	✓	✓	4

将两图结果综合，得到 4 个分配方案的命中率情况表如下

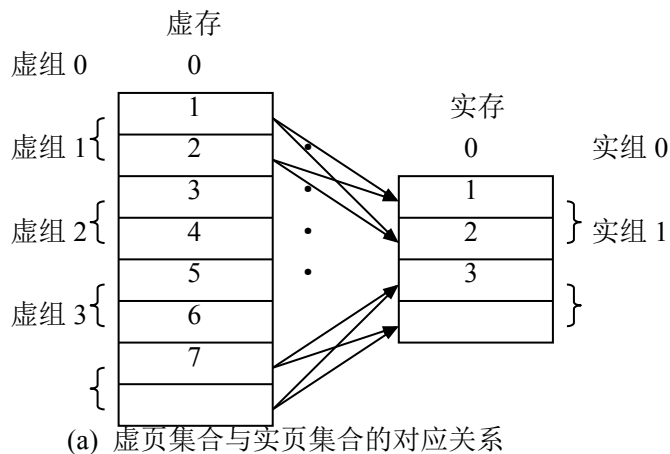
n_1	1	2	3	4
$N_{(1)}$	0	0	2	4
n_2	4	3	2	1
$N_{(2)}$	4	4	2	2
$N_{(1)}+N_{(2)}$	4	4	4	6

结论如下

- (1) 命中率相同的方案是 $n_1 = 3$ 而 $n_2 = 2$;
- (2) 命中次数之和最大的方案是 $n_1 = 4$ 而 $n_2 = 1$ 。

3. 19 中 (3) (4) (6) (8) 问

(3)



		实页 0	1	2	3
虚页	0	✓	✓		
	1	✓	✓		
	2			✓	✓
	3			✓	✓
	4	✓	✓		
	5	✓	✓		
	6			✓	✓
	7			✓	✓

(b) 对应关系表 (✓为有关系)

(4) 通过作“实存状况图”模拟各虚块的调度情况，可获得 Cache 的块地址流序列。

P=	6	2	4	1	4	6	3	0	4	5	7	3
C0			4	4*	4	4	4	4*	4	4*	4*	4*
C1				1	1*	1*	1*	0	0*	5	5	5
C2	6	6*	6*	6*	6*	6	6*	6*	6*	6*	7	7*
C3		2	2	2	2	2*	3	3	3	3	3*	3
	入	入	入	入	中	中	替	替	中	替	替	中
C=	2	3	0	1	0	2	3	1	0	1	2	3

此问最容易出错的地方是忽略“组相联”地址约束，将虚页装错实组。另外没有及时标注“*”号也容易导致淘汰对象错误。

(6) $H=4/12 \approx 33\%$

(8) 做法同 3.15 题 (3) 问， $H_{\text{cell}} = (12 \times 16 - 8) / (12 \times 16) \approx 95.8\%$

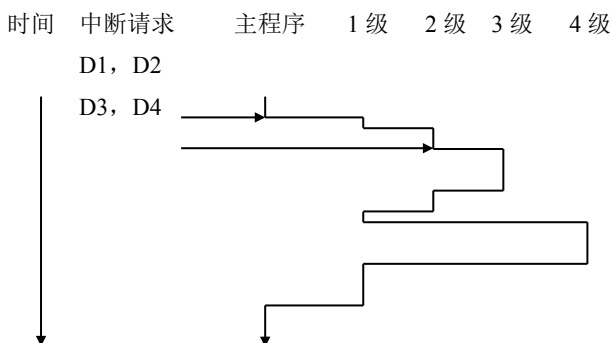
第四章 (P250)

4.5 已知中断服务次序为 3-2-4-1

(1) 中断屏蔽字表如下图；

(2) 中断过程示意图如右图。

	D1	D2	D3	D4
D1	0	1	1	1
D2	0	0	1	0
D3	0	0	0	0
D4	0	1	1	0

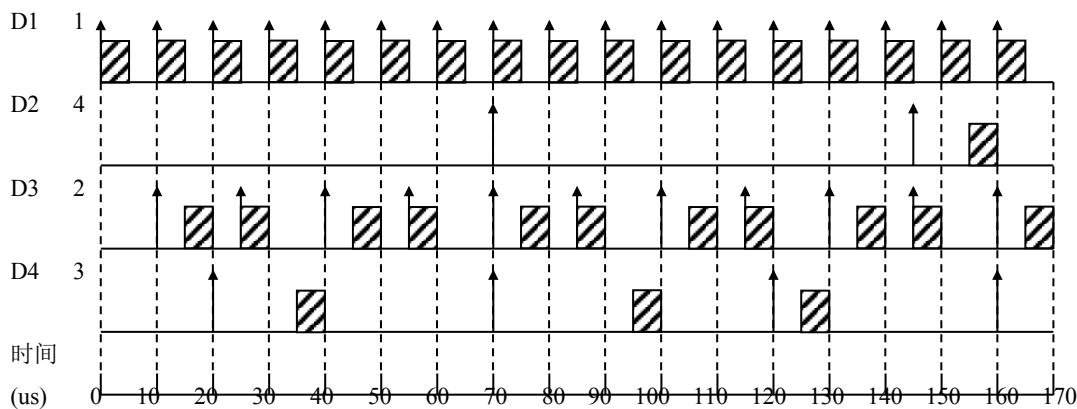


4.8

(1) $f=2 \times 10^5$ 字节/秒， $T=5\mu\text{s}$

(2) $T_s+T_d=5\mu\text{s}$ ，通道时间图如下。作图时注意：至少要画到最慢设备的第二次请求出现，才能确定是否丢失数据（因为响应优先级低的设备较易丢失数据）。

设 优
备 先
号 级



(3) 5, 160, 20, 40;

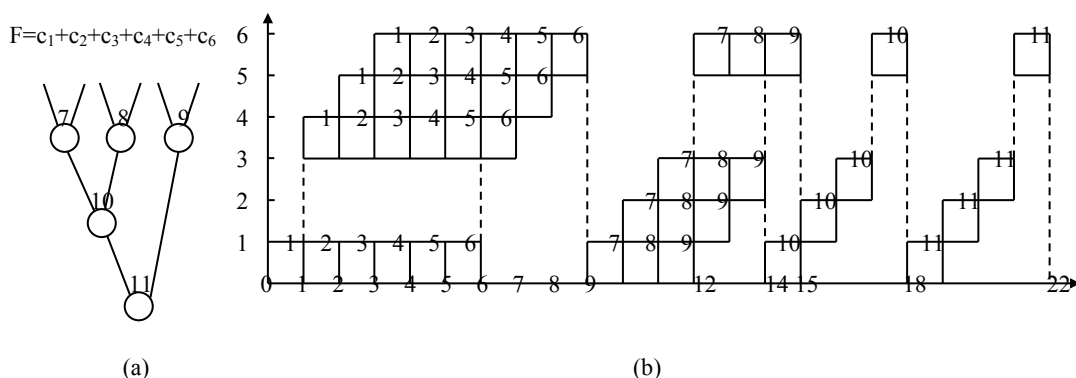
(4) D2 丢失第一次请求的数据;

(5) 参见 P245。

第五章 (P343)

5.9 为了缩短运算时间, 首先应考虑“最少切换算法”, 即先执行完所有乘法(任务编号 1-6)再执行加法(任务编号 7-11), 其次在加法中采用“最少相关算法”(即二叉树算法)。

记 $c_1=A_1 \times B_1, \dots, c_6=A_6 \times B_6$, 下图(a)是加法的计算顺序二叉树, 注意任务 10 应该用前一级最早完成的任务 7 和 8 的结果, 如果用任务 9 的结果则要推迟 1 拍启动, 使总时间增加 1 拍。



根据时空图(b)得

$$TP = 11 / (22 \Delta t) = 1 / (2 \Delta t)$$

$$S = (6 \times 4 \Delta t + 5 \times 4 \Delta t) / (22 \Delta t) = 2$$

$$E = (6 \times 4 \Delta t + 5 \times 4 \Delta t) / (6 \times 22 \Delta t) = 1/3$$

5.15 $\Delta t = 10\text{ns} = 10^{-8}$ 秒

(1) $F = \{1, 2, 5\}$, $C = (10011)$

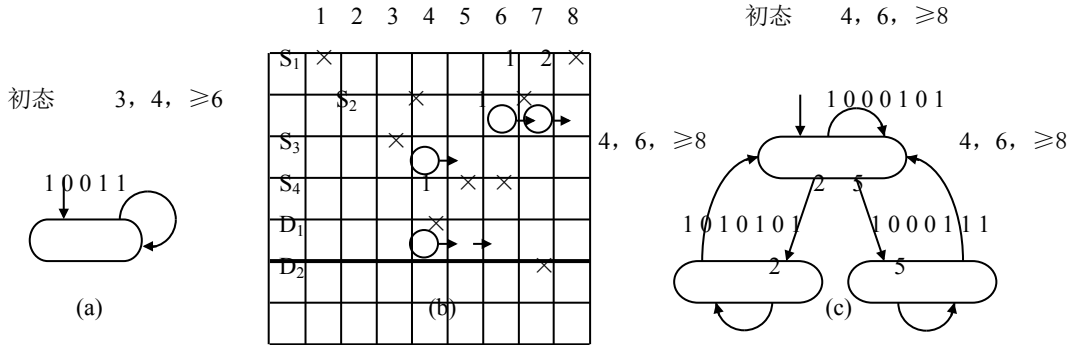
(2) 状态转移图如下图(a)所示。

(3) 最小启动循环= (3), 最小平均启动距离= $3\Delta t$ 。

(4) 插入 2 个延迟, 最小启动循环= (2), 最小平均启动距离= $2\Delta t$ 。

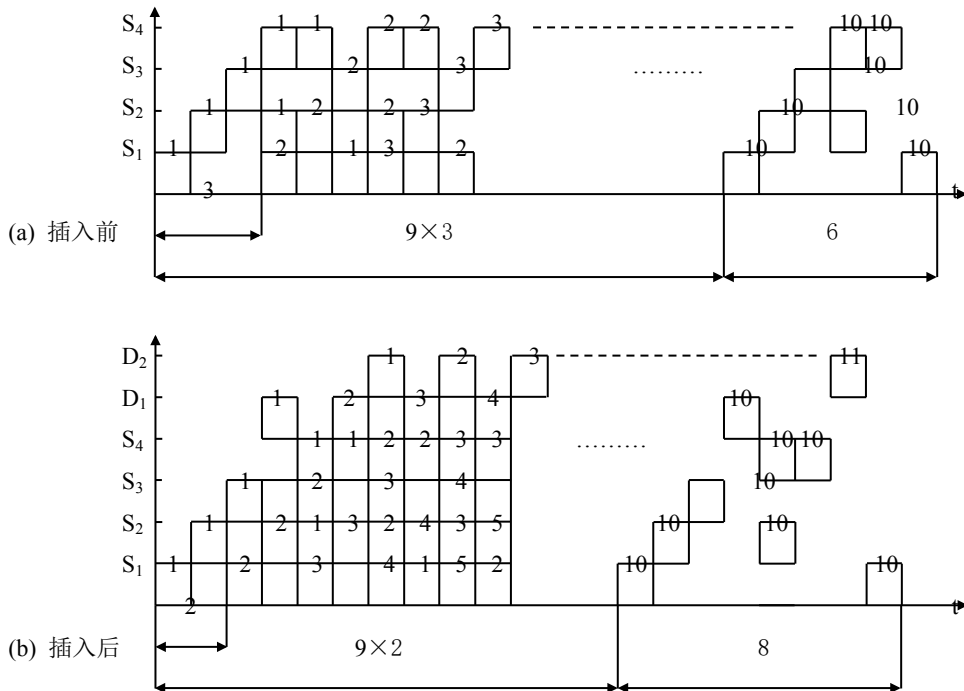
(5) 新预约表如下图(b)所示。

(6) $F = \{1, 3, 7\}$, $C = (1000101)$, 状态转移图如下图(c)所示。



(7) 插入前 $TP_{\max} = 1/3 \Delta t = 1/30\text{ns}$, 插入后 $TP_{\max} = 1/2 \Delta t = 1/20\text{ns}$ 。

(8) 插入前 $TP = 10/33 \Delta t = 1/33\text{ns}$, 插入后 $TP = 10/26 \Delta t = 1/26\text{ns}$, 如下图所示。

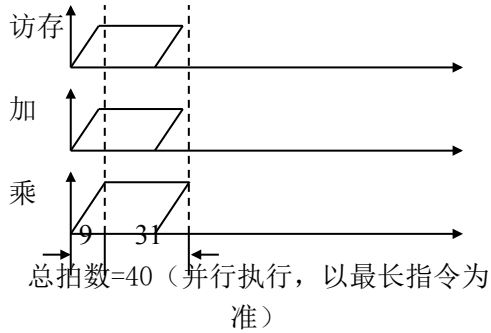


第六章 (P391)

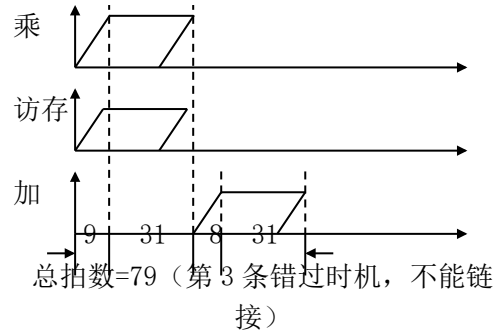
6.6 (注意阅读 P372 倒数第 9 行—倒数第 6 行)

已知 $n=32$, $k_{\text{加}}=6$, $k_{\text{乘}}=7$, $k_{\text{访存}}=6$, $k_{\text{倒数}}=14$, 启动、输出延迟各 1。求各小题总拍数。

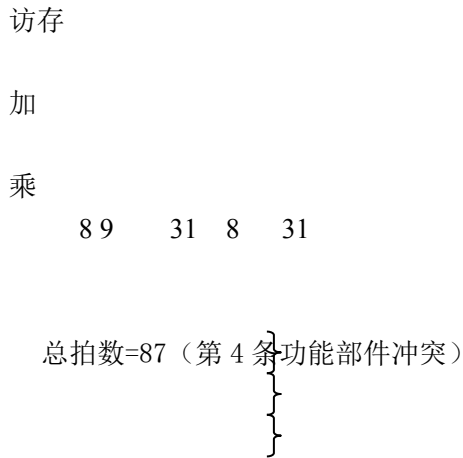
- (1) $V0 \leftarrow \text{存储器}$
 $V1 \leftarrow V2 + V3$
 $V4 \leftarrow V5 * V6$ } 并行



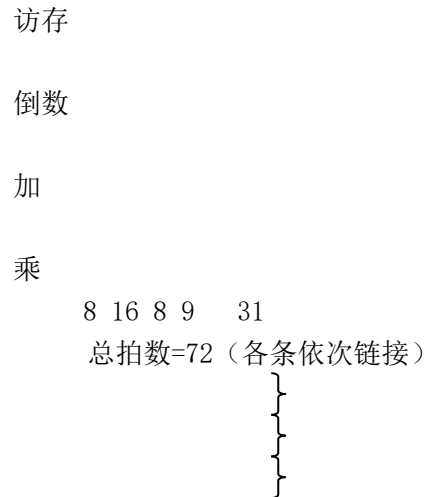
- (2) $V2 \leftarrow V0 * V1$ 并行
 $V3 \leftarrow \text{存储器}$
 $V4 \leftarrow V2 + V3$ } 串行 (P372)



- (3) $V0 \leftarrow \text{存储器}$ 并行
 $V3 \leftarrow V1 + V2$ 链接
 $V4 \leftarrow V0 * V3$
 $V6 \leftarrow V4 + V5$ 串行

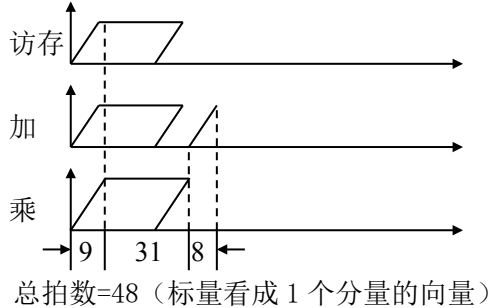


- (4) $V0 \leftarrow \text{存储器}$ 链接
 $V1 \leftarrow 1 / V0$ 链接
 $V3 \leftarrow V1 + V2$ 链接
 $V5 \leftarrow V3 * V4$



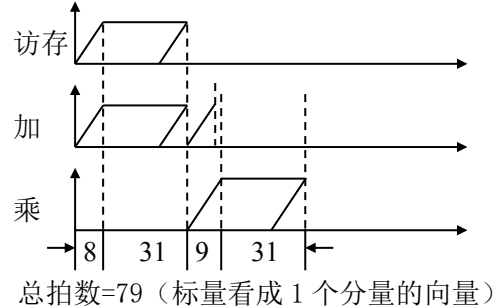
(5) $V0 \leftarrow \text{存储器}$
 $V1 \leftarrow V2 + V3$
 $V4 \leftarrow V5 * V6$
 $s0 \leftarrow s1 + s2$

} 并行
 } 串行



(6) $V3 \leftarrow \text{存储器}$
 $V2 \leftarrow V0 + V1$
 $s0 \leftarrow s2 + s3$
 $V3 \leftarrow V1 * V4$

} 并行
 } 串行
 } 并行



(7) $V3 \leftarrow \text{存储器}$ 并行
 $V2 \leftarrow V0 + V1$ 链接
 $V4 \leftarrow V2 * V3$
 $\text{存储器} \leftarrow V4$ 串行

访存

加

乘

8 9 31 8 31
 总拍数=87 (第 4 条功能部件冲突)

(8) $V0 \leftarrow \text{存储器}$ 链接
 $V2 \leftarrow V0 + V1$
 $V3 \leftarrow V2 * V1$ 串行
 $V5 \leftarrow V3 * V4$ 串行

访存

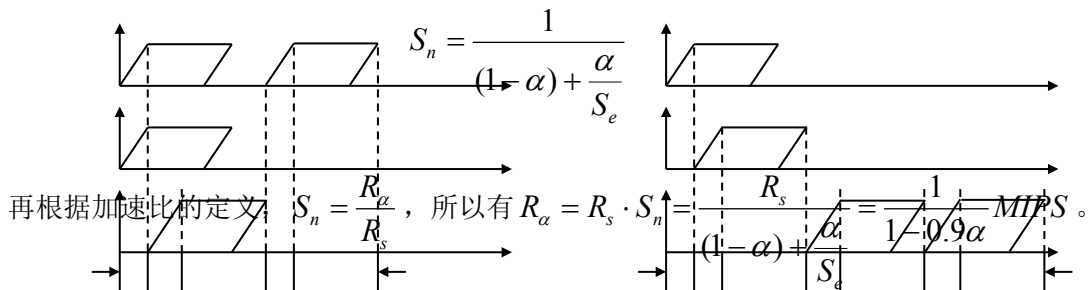
加

乘

8 8 31 9 31 9 31
 总拍数=127 (V_i 冲突, 功能部件冲突)

6.10 已知向量速率 $R_v = 10\text{MFLOPS}$, 标量速率 $R_s = 1\text{MFLOPS}$, 并记 α 为可量化百分比。

(1) 推导法 1: 使用 Amdahl 定律, 在这里可将标量速率 R_s 作为原速率, 局部加速后的速率为向量速率 R_v , 于是局部加速比 $S_e=10$, 全局加速比为



(若将向量速率 R_v 作为原速率, 局部减速后的速率为标量速率 R_s , 则局部加速比 $S_e=0.1$, 推出的全局加速比 S_n 同上式。)

推导法 2: 为了推导, 定义 T 为总时间, N 为总任务数。于是有平均速率 $R_a = \text{吞吐率 TP} = N/T$ 。记 $N = N_v + N_s$, 且 $\alpha = \frac{N_v}{N} = \frac{N_v}{N_v + N_s}$, 则 $1 - \alpha = \frac{N_s}{N} = \frac{N_s}{N_v + N_s}$, 于是有 $N_v = \alpha \cdot N$ 和 $N_s = (1 - \alpha) \cdot N$

$$\text{显然: 总时间 } T = T_v + T_s = \frac{N_v}{R_v} + \frac{N_s}{R_s} = \frac{\alpha \cdot N}{R_v} + \frac{(1 - \alpha) \cdot N}{R_s}$$

$$\text{所以: } R_a = \frac{N}{T} = \frac{N}{\frac{\alpha \cdot N}{R_v} + \frac{(1 - \alpha) \cdot N}{R_s}} = \frac{1}{\alpha \cdot \frac{1}{R_v} + (1 - \alpha) \cdot \frac{1}{R_s}}$$

$$\text{或者: } \frac{1}{R_a} = \alpha \cdot \frac{1}{R_v} + (1 - \alpha) \cdot \frac{1}{R_s}$$

(2) 已知 $R_v = 10\text{MFLOPS}$, $R_s = 1\text{MFLOPS}$,

$$R_a = \frac{1}{0.1\alpha + (1 - \alpha)} \text{MFLOPS} = \frac{10}{10 - 9\alpha} \text{MFLOPS}$$

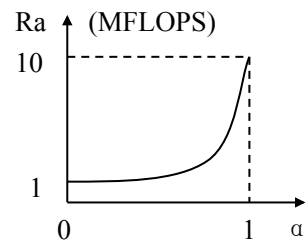
R_a 与 α 的关系图如右图所示。

(3) 已知 $R_a = 7.5\text{MFLOPS}$, 解出

$$\alpha = \frac{10}{9} \left(1 - \frac{1}{7.5}\right) = \frac{10}{9} \times \frac{13}{15} \approx 0.96 = 96\%$$

(4) 已知 $R_a = 2\text{MFLOPS}$, $\alpha = 0.7$, 解出

$$R_v = \frac{\alpha}{\frac{1}{R_a} - (1 - \alpha) \cdot \frac{1}{R_s}} = \frac{0.7}{\frac{1}{2} - 0.3 \times 1} = 3.5(\text{MFLOPS})$$



第七章 (P446)

7.3 已知输入端编号 $13 = 1101\text{B}$ 。

$$(1) \text{Cube}_3(1101\text{B}) = 0101\text{B} = 5$$

$$(2) \text{PM}_{2+3}(13) = (13 + 2^3) \bmod 16 = 21 \bmod 16 = 5$$

$$(3) \text{PM}_{2+0}(13) = (13 - 2^0) \bmod 16 = 12$$

$$(4) \text{Shuffle}(1101\text{B}) = 1011\text{B} = 11$$

$$(5) \text{Shuffle}(\text{Shuffle}(1101\text{B})) = \text{Shuffle}(1011\text{B}) = 0111\text{B} = 7$$

7.4 用多级混洗—交换网络, $n = 4$, 拓扑结构同教材 P410 图 7.21(e), 控制信号=1010B, 自左向右各级交换开关状态依次为交换—直连—交换—直连。

7.5 输入结点编号 $j = 9$, $f(j) = j \oplus \text{控制信号} = 1001\text{B} \oplus 1100\text{B} = 0101\text{B} = 5$, 答为 5 号处理机。

7.6 直连状态时: 编号在第 i 位不同的结点之间不能通信;
交换状态时: 编号在第 i 位相同的结点之间不能通信。

7.7 用单级混洗—交换网可实现, 总共混洗 3 步。

证: 设矩阵 $A = (a_{ij})_{8 \times 8}$ 按行展开依次存放在 64 个单元中, 则任意元素 a_{ij} 的地址为 $8i + j$, 而 a_{ji} 的地址为 $8j + i$ 。按混洗函数的定义, 3 次混洗后, $\text{shuffle}^3(8i + j) = 8 \times (8i + j) \bmod 63 = i + 8j$, 也就说将元素 a_{ij} 地址变换成 a_{ji} 的地址。由于 a_{ij} 是矩阵中的任意元素, 所以 3 次混洗可实现矩阵转置 $(a_{ij})^T_{8 \times 8} = (a_{ji})_{8 \times 8}$ 。

7.8 最多 5 级, 因为对于任给的输入结点编号 $j = X_6X_5X_4X_3X_2X_1X_0$, PM2I 多级网络中 $i=2$ 级的功能是 $\text{PM2}_{\pm 2}(j) = j \pm 2^2 \bmod 128$, $\pm 2^2$ 运算只有可能改变 j 中的 $X_6 \sim X_2$, 所以最多使用 $\text{Cube}_6 \sim \text{Cube}_2$ 就能实现代换了。

7.9 由于 $N = 16$, 即 $n = 4$, 每个结点编号用 4 位二进制数表示。PM2 $_{\pm 0}$ 函数功能是对结点编号加 1 或减 1, 其结果最多可将编号的 4 位都取反 (如 $1111\text{B} + 1 = 0000\text{B}$), 所以用每步只能对 1 位取反的单级立方体网络来模仿, 最差情况下要 4 步。

7.10 用混洗—交换网络模拟 Cube 网。

当模拟 Cube_0 功能时, 只需一次交换即可完成; 而模拟 Cube_i 且 $i \neq 0$ 时, 需先作 $n - i$ 步混洗, 再作 1 步交换, 最后作 i 步混洗才能完成, 共计 $n + 1$ 步。

综上所述, 下限为 1 步, 上限为 $n + 1$ 步。

7.11 求单级立方体网络和单级混洗—交换网络的最大广播步数, 这两种网络的最大广播步数与最大距离 (即直径) 相同。

(1) 单级立方体网络直径 = n ($\text{Cube}_{n-1} \sim \text{Cube}_0$ 各 1 次);

(2) 单级混洗—交换网络直径 = $2n-1$ ($n-1$ 次混洗, n 次交换)。

7.12 已知 $N = 16$, 用多级立方体网络或者多级混洗—交换网络均能实现, 两者可以互相模拟, 对同一置换的寻径算法相同, 控制信号也相同, 下面以多级立方体网络为例分析。

4 组 4 元交换: $f_1 = \text{Cube}_1\text{Cube}_0$;

2 组 8 元交换: $f_2 = \text{Cube}_2\text{Cube}_1\text{Cube}_0$;

1 组 16 元交换: $f_3 = \text{Cube}_3\text{Cube}_2\text{Cube}_1\text{Cube}_0$;

利用 Cube 函数的结合律、交换律以及同一律 (又称自反律) 可以推得

$$f = f_1 f_2 f_3 = \text{Cube}_3 \text{Cube}_1 \text{Cube}_0$$

拓扑结构图略 (可参考 7.26 题的多级混洗—交换网络拓扑结构图)。

网络开关使用级控方式, 控制信号为 1011B (其中 bit_i 控制级 i , “0” 表示直连, “1” 表示交换)。

7.13 $N = 8$ 的蝶式置换。

(1) $f(X_2X_1X_0) = X_0X_1X_2$;

(2) 至少需 2 次通过，每次都是 N 个数据同时发送，同时接收，中途不储存；

(3) 控制信号的设置有 4 种方案，如下所示。其中“0”表示直连，“1”表示交换。

$$\begin{pmatrix} 101 \\ 000 \\ 000 \\ 101 \end{pmatrix} \begin{pmatrix} 100 \\ 000 \\ 000 \\ 100 \end{pmatrix} \quad \begin{pmatrix} 001 \\ 000 \\ 000 \\ 001 \end{pmatrix} \begin{pmatrix} 101 \\ 000 \\ 000 \\ 101 \end{pmatrix} \quad \begin{pmatrix} 000 \\ 101 \\ 101 \\ 000 \end{pmatrix} \begin{pmatrix} 000 \\ 100 \\ 100 \\ 000 \end{pmatrix} \quad \begin{pmatrix} 000 \\ 001 \\ 001 \\ 000 \end{pmatrix} \begin{pmatrix} 000 \\ 101 \\ 101 \\ 000 \end{pmatrix}$$

7.14

(1) 共 $N!$ 种；

(2) 一次通过有 $N^{\frac{N}{2}}$ 种不同；

(3) $N = 8$ 时，百分比 = $\frac{N^{\frac{N}{2}}}{N!} \times 100\% = \frac{8^4}{8!} \times 100\% \approx 10.16\%$

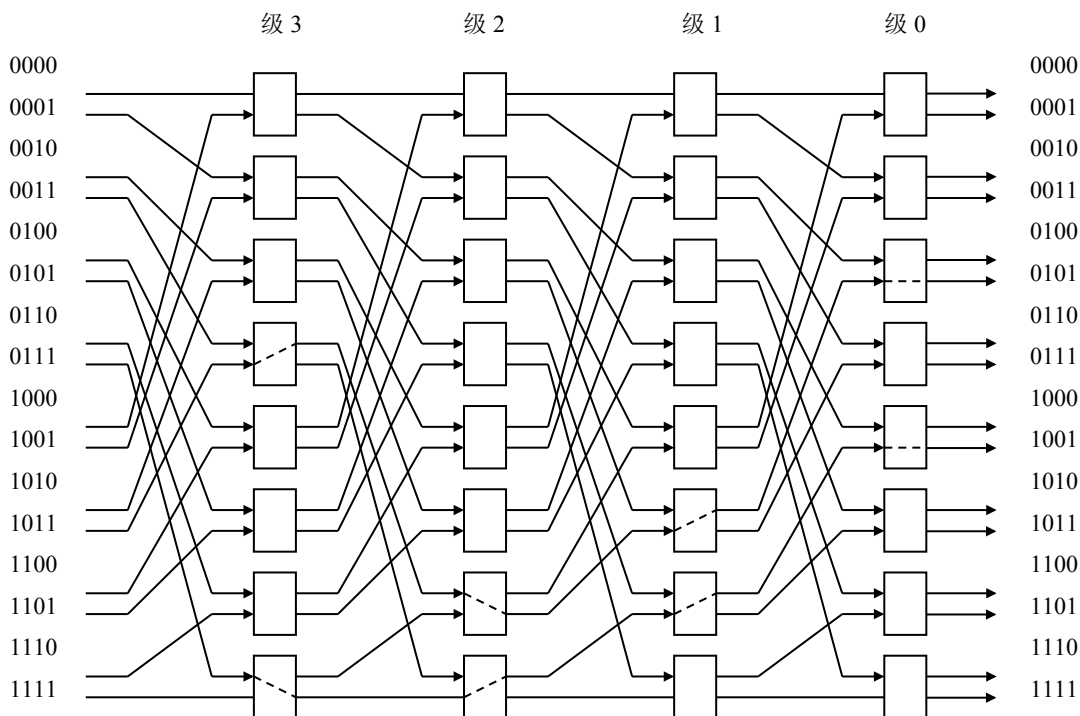
7.23 同 7.11 题

7.26(1) ~ (3)；

(1) 见下图实线。

(2) 见下图虚线；不会阻塞，因为两条路径的控制信号都是 1110，形成级控模式，所以不会阻塞。

(3) 一次通过实现的置换数为 $16^8 = 4294967296$ ，全部置换数为 $N! = 20922789888000$ ，前者约占后者的 0.02%。



7.27

(1) 已知 $N = 64$, $n = 6$, 源结点 $s = 101101B$, 目的结点 $d = 011010B$, 方向矢量 $r = s \oplus d = 110111B$, 以低维度优先顺序寻径, 路径为

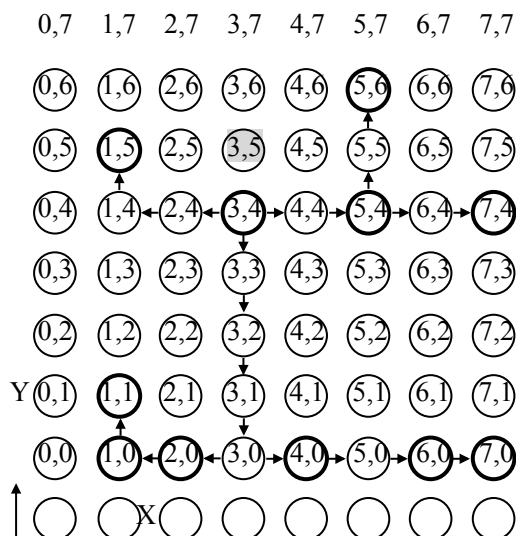
$s = 101101B \rightarrow 101100B \rightarrow 101110B \rightarrow 101010B \rightarrow 111010B \rightarrow 011010B = d$ (下划线为当前寻径维)

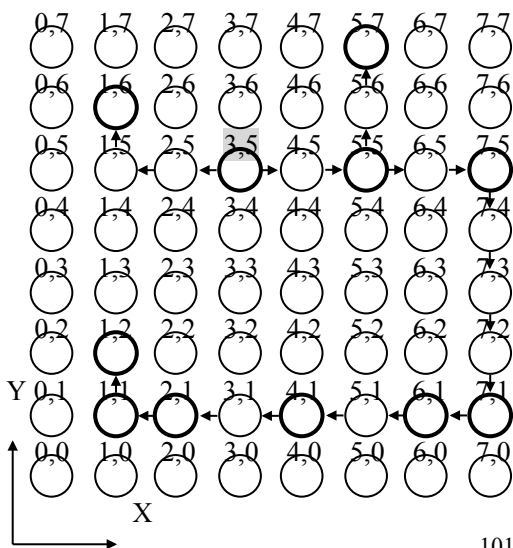
(2) 求给定无向图中 2 棵选播树 (即生成树)。

(i) 求最小成本生成树 (通道数最少), 可考虑 Prim 算法、Kruskal 算法或标记法。一个参考操作方法是: 先对临近结点群分别构造最短子树, 然后在子树之间作最短互连。

(ii) 求由结点 (3, 5) 出发的单源最短路径生成树 (各距离最短), 可考虑贪心算法。对 X-Y 网格图来说, 从树根到某一树叶的任何路径只要在各维均无反向移动即为最短路径 (满足此条件的最短路径有多条)。要得到单一树根对于多片树叶的综合最短路径, 可以先分别作出各条单播最短路径, 然后在不增加各路径长度的前提下, 尽可能地进行路段合并。

这两小问结果如下图所示 (其中 b 图第一步必须选择向下, 而不能向右)。



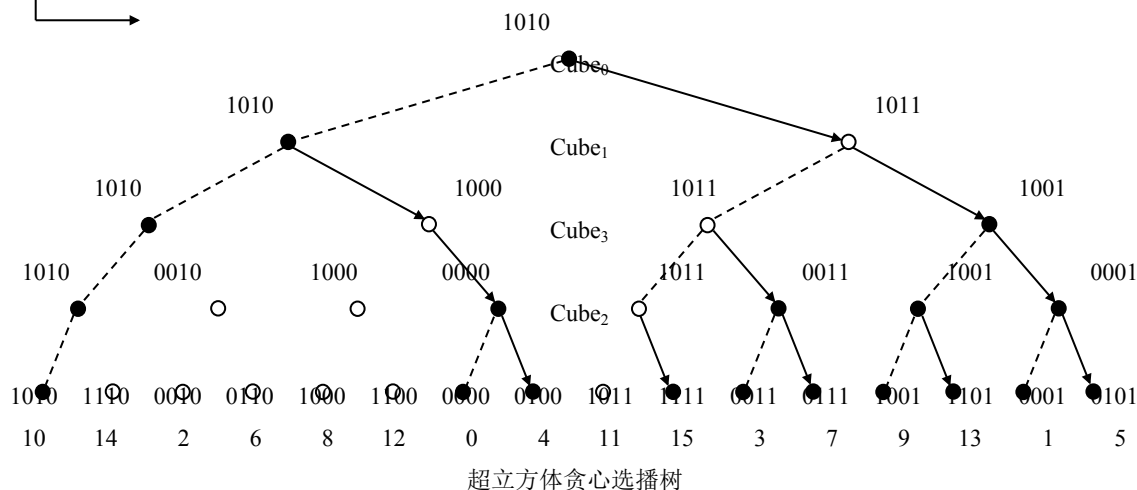


(a)

(b)

(3) 求作超立方体贪心选播树

源结点编号的二进制形式 1010 在 $Cube_0 \sim Cube_3$ 位分别与 7、6、5、6 个目的结点的二进制形式不同，所以总时间=4（如某一位没有与源结点不同的目的结点，则该维不发送，总时间减少 1），并且 $Cube_0$ 方向应该最先发送， $Cube_2$ 方向最后发送，其它 2 个方向的发送先后顺序没有限制。我们可以采用 $Cube_0$ 、 $Cube_1$ 、 $Cube_3$ 、 $Cube_2$ 的发送顺序，生成的选播树如下图所示，总流量=11。



7.29 已知 $N = 256$, $n = 8$, 起始结点编号 $j = 123 = 01111011B$ 。根据混洗函数的循环移位性质, $Shuffle^{10}(j) = Shuffle^2(j) = 11101101B = 237$

第八章 (P498)

8.12 问题为 $S=A1 \times B1 + \dots + A32 \times B32$, 其中 $T_{乘}=4 \Delta t$, $T_{加}=2 \Delta t$, $T_{传}=1 \Delta t$ 。

(1) 在串行计算机上, 各操作不论是否相关均不能重叠, 总时间恒等于各操作单独时间之和, 所以不必考虑运算顺序。 $T=32 \cdot T_{乘}+31 \cdot T_{加}=(32 \times 4+31 \times 2) \Delta t=190 \Delta t$

(2) 设此双向环可以并行传送 (即为“移数环”, 因为 SIMD 系统各种数据操作都能并行)。

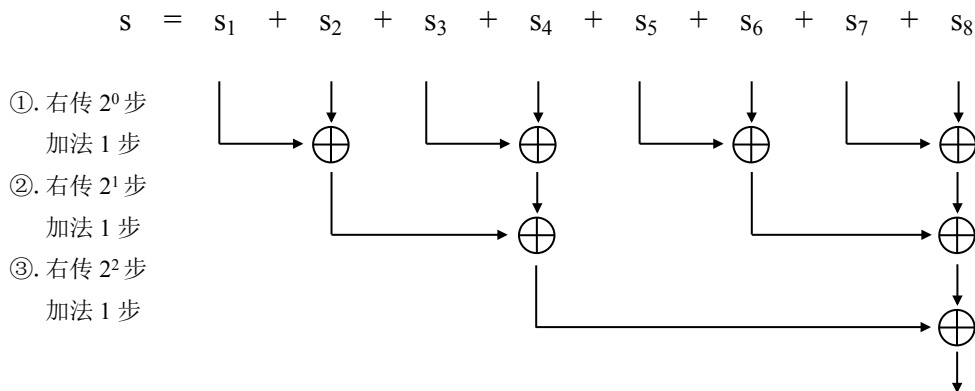
按平均分配原则, 每个结点内有 4 对数据。

首先在各结点用串行算法它们的相乘与求和, 需时 $T_1=4 \cdot T_{乘}+3 \cdot T_{加}=(4 \times 4+3 \times 2) \Delta t=22 \Delta t$;

然后用二叉树并行算法将 8 个结点中的部分和相加 (见下图), 其中并行加法需 3 次, 每次时间相同, 而并行传送 3 次的每次时间却随距离倍增, 依次为 1、2、4 步, 所以有

$T_2=(1+2+4) \cdot T_{传}+3 \cdot T_{加}=(7 \times 1+3 \times 2) \Delta t=13 \Delta t$;

总时间 $T=T_1+T_2=35 \Delta t$



第九章 (P562)

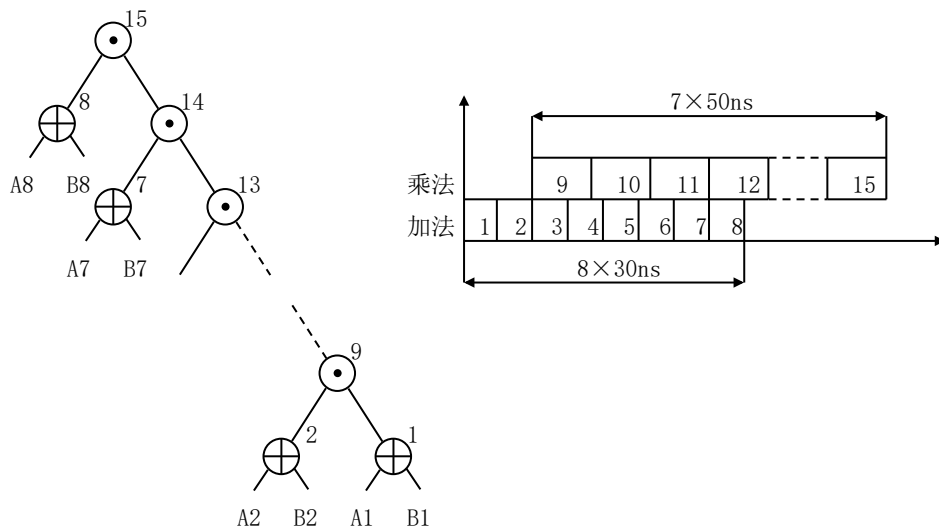
9.18 问题为 $S=(A1+B1) \times \dots \times (A8+B8)$ ，其中 $T_{\text{加}}=30\text{ns}$ ， $T_{\text{乘}}=50\text{ns}$ ， $T_{\text{传}}=10\text{ns}$ 。

将加法记为任务 1-8，乘法记为任务 9-15。

(1) 在串行计算机上，同 8.12 题 1 问分析，共计 15 步运算， $T=8 \cdot T_{\text{加}}+7 \cdot T_{\text{乘}}=(8 \times 30+7 \times 50)\text{ns}=590\text{ns}$ 。

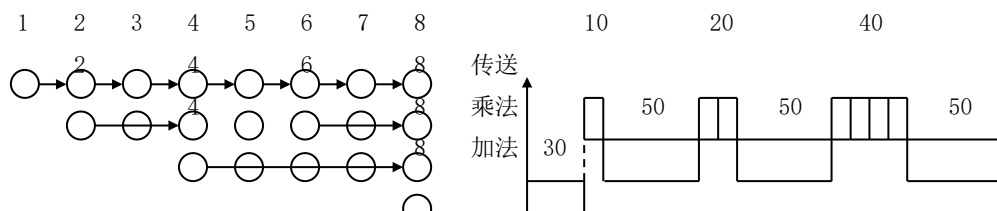
(2) 多功能部件 SISD 计算机的工作方式可参考 P346 题 18(3)。

为了充分利用加法器与乘法器的可并行性，尽量让加法与乘法交替进行，可自左向右顺序运算（见下图）。 $T=2 \cdot T_{\text{加}}+7 \cdot T_{\text{乘}}=(2 \times 30+7 \times 50)\text{ns}=410\text{ns}$



(3) 同 8.12 题 2 问，设单向环可以并行传送（即为“移数环”，理由同 8.12 题 2 问）。

$T=T_{\text{加}}+3 \cdot T_{\text{乘}}+(1+2+4) \cdot T_{\text{传}}=(30+3 \times 50+7 \times 10)\text{ns}=250\text{ns}$



(4)在全互连网络上，任意两个结点之间的距离均为 1 步，所以任何置换都能在 1 步完成，故
 $T = T_{\text{加}} + 3 \cdot T_{\text{乘}} + (1+1+1) \cdot T_{\text{传}} = (30+3 \times 50+3 \times 10) \text{ns} = 210 \text{ns}$

