

课程教学大纲及目的：

- 计算机系统的基本概念、分析设计方法和关键技术。
- 流水线技术、指令级并行技术、向量处理机技术，以及互联网络、阵列计算机、多处理机等

第一章 计算机系统结构的基础知识

系统结构的相关概念

以语言为特征,划分为6个等级

- ◆ 计算机系统的层次结构 **概念**
- ◆ 广义机器、虚拟机器、透明性、编译、解释
- ◆ 计算机系统结构、组织和实现
- ◆ 计算机系统分类方法\ Flynn 分类法

系统分析技术

- ◆ 大概率事件优先 **原理**
- ◆ **Amdahl**定律、加速比 **定义**
- ◆ 程序访存的局部性 **原理**
- ◆ **CPU**性能 **公式**

$$S_e = \frac{T_0}{T_e} = \frac{1}{1 - F_e + F_e / S_e}$$

性能评价标准

- ◆ 性能指标 (**CPU**时间, **CPI**, **MIPS**, **MFLOPS**)
- ◆ 性能比较

$$CPI = \sum_{i=1}^n (CPI_i \times I_i) / I_N = \sum_{i=1}^n (CPI_i \times I_i / I_N)$$

2021/6/23

- ◆ 指令平均周期数 **CPI**

例1：计算机系统有三个部件可改进，这三个部件的加速比如下：部件1加速比 $S_1=10$ ；部件2加速比 $S_2=20$ ；部件3加速比 $S_3=30$ ；问：

（1）如果部件1和部件2的所占比例为25%，那么当部件3所占比例为多少时，系统的加速比才可以达到5？

（2）如果三个部件所占比例为10%、20%和30%，当三个部件改进后，原系统中不可加速部分的执行时间在总执行时间中所占的比例变为多少？

解：多部件改进情况下Amdahl定理为，

$$S = \frac{1}{(1 - \sum_i F_i) + \sum_i \frac{F_i}{S_i}}$$

2021/6/23 设部件3原所占为 F_3 （ F_i 为可加速部件 i 在未优化系统中的比例）³

$$5(1-0.5-F3+ 0.25/10+0.25/20+F3/30)=1$$

$$F3\sim=34.9\%$$

2 系统中不可加速部分的执行时间

$$(1-0.1-0.2-0.3)T=0.4 T$$

$$\text{改进后系统总时间} = (1-0.6+0.1/10+0.2/20+0.3/30)T$$

$$=0.43 T$$

不可加速部分的执行时间在总执行时间中所占的比例

$$p = \frac{[1 - (0.1 + 0.2 + 0.3)]T}{\frac{0.1}{10}T + \frac{0.2}{20}T + \frac{0.3}{30}T + 0.4T} = \frac{0.4}{0.43} = 0.93 = 93\%$$

例2 某台主频为400MHz的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：求该计算机的有效CPI、MIPS和程序执行时间。

指令类型	指令执行数量/条	平均时钟周期数
整数	45 000	1
数据传送	75 000	2
浮点	8 000	4
分支	1 500	2

解：（1） $CPI = (45000 \times 1 + 75000 \times 2 + 8000 \times 4 + 1500 \times 2) / 129500 = 1.776$

（2） $MIPS速率 = f / CPI = 400 / 1.776 = 225.225 MIPS$

（3）程序执行时间 = $(45000 \times 1 + 75000 \times 2 + 8000 \times 4 + 1500 \times 2) / 400 \times 1000000 = 0.000575s$

例3假设某应用程序中有4类操作，通过改进，各操作获得不同的性能提高。具体数据如下表所示：

- (1) 各类操作单独改进后，程序获得的加速比分别是多少
- (3) 4类操作均改进后，整个程序的加速比是多少？

操作类型	程序中的数量 (百万条指令)	改进前的执行时间 (周期)	改进后的执行时间 (周期)
操作1	10	2	1
操作2	30	20	15
操作3	35	10	3
操作4	15	4	1

解：1) 根据Amdahl定律

操作类型	各类操作的指令在程序中所占的比例 F_i	各类操作的加速比 S_i	各类操作单独改进后，获得的加速比
操作1	11.1%	2	1.06
操作2	33.3%	1.33	1.09
操作3	38.9%	3.33	1.37
操作4	16.7%	4	1.14

2) 4类操作均改进后，整个程序的加速比：

$$S_n = \frac{1}{(1 - \sum F_i) + \sum \frac{F_i}{S_i}} \approx 2.16$$

第三章 流水线技术

流水线技术

流水线的概念： ^{同一时间一个功能} 静（动）态流水线、^{同一时间多个功能} 单（多）功能流水线
^{串行无反馈} 线性流水线、^{串行有反馈} 非线性流水线

流水线表示方法--时空图、连接图

流水线特点：五段流水线（访存部件ME, 转移EX）

性能指标：吞吐率、加速比、效率、最佳段数？

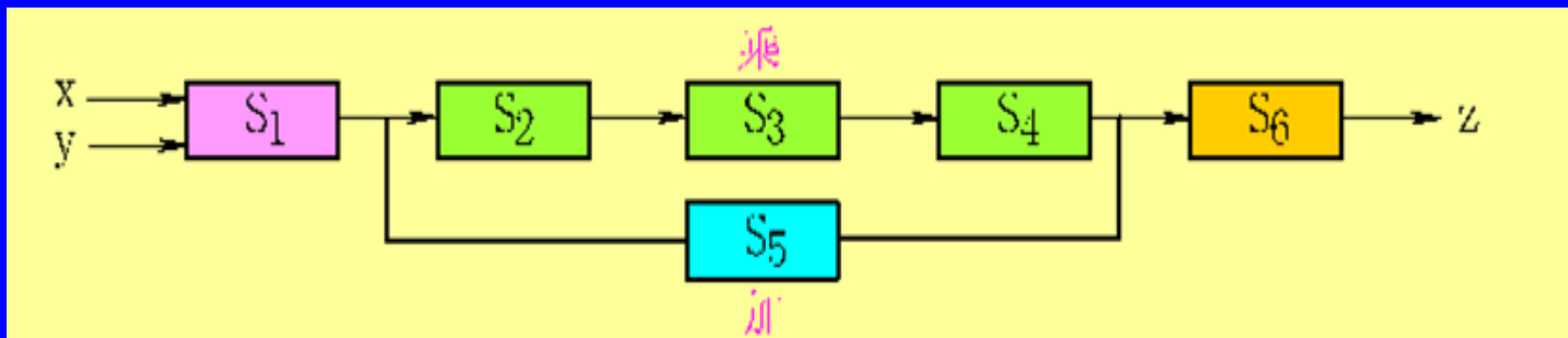
指令 \ 流水段	ALU	LOAD/STORE	BRANCH
IF	取指	取指	取指
ID	译码，读寄存器堆	译码，读寄存器堆	译码，读寄存器堆
EX	执行	计算访存有效地址	计算转移目标地址， 设置条件码
ME	(空操作)	访问存储器(读或写)	若条件成立，将转移 目标地址送 PC
WB	结果写回寄存器堆	将读出的数据写 入寄存器堆	(空操作)

单功能与多功能流水线

(按照流水线所完成的功能来分类)

- 单功能流水线：只能完成一种固定功能的流水线。
- 多功能流水线：流水线的各段可以进行不同的连接，以实现不同的功能。

例：多功能流水线：加、乘流水线



流水线相关与冲突及解决办法

- 数据相关（真数据相关）/ 名相关 / 控制相关 分支指令

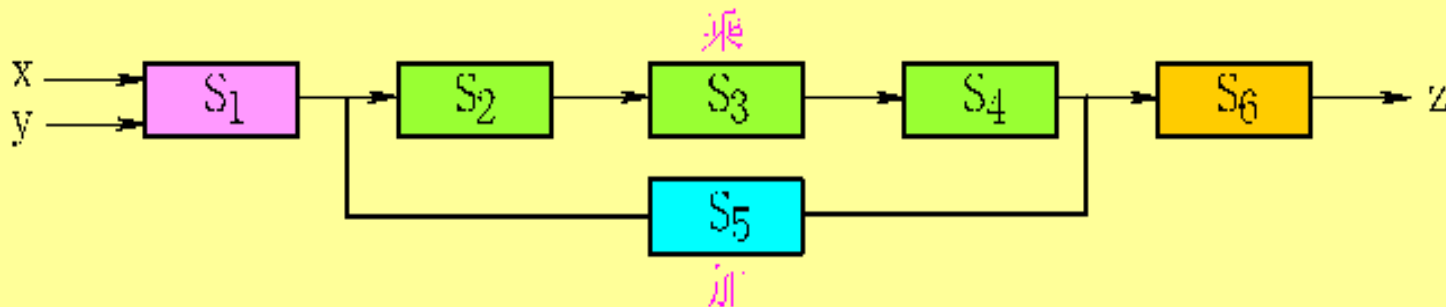
- 结构冲突、数据冲突、控制冲突：

数据冲突解决办法 控制冲突解决办法

多功能流水线时空图

例1 下图所示为静态加、乘双功能流水线，段S1、S2、S3、S4、S6组成乘法流水线，S1、S5、S6组成加法流水线。设向量 $a=(a_1, a_2, a_3, a_4)$ ，向量 $b=(b_1, b_2, b_3, b_4)$

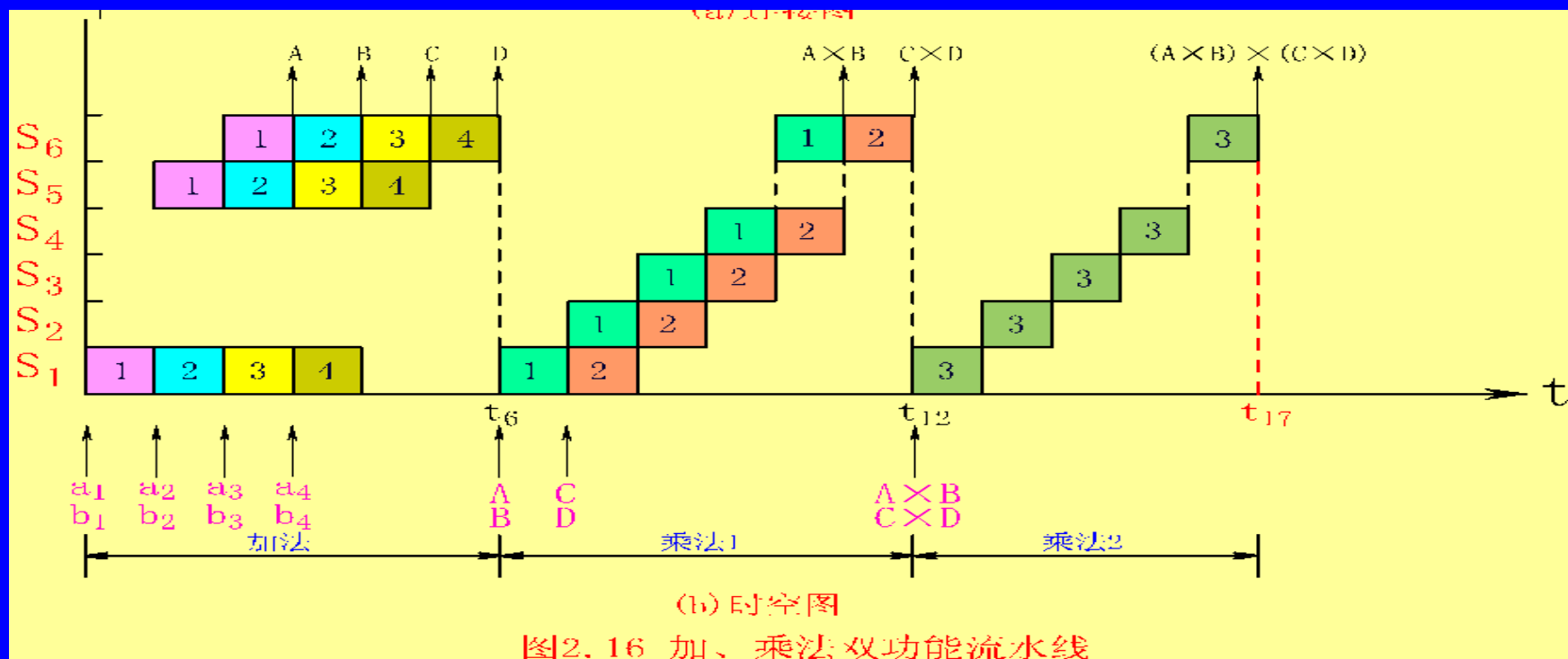
计算 $\prod_{i=1}^4 (a_i + b_i)$ 。画出流水线时空图，求P、S、E



(a) 连接图

解： 1 分析: 先进行4次加法,再进行乘法.

2 画出流水线时空图



静态: 加法完成后再进行乘法

动态: 不要求加法完成

1) 由时空图, 17个 Δt 时间内输出7个结果, 因此

$$P = \frac{n}{T_k} = \frac{7}{17\Delta t}$$

2) 串行方法完成, 需4次加法和3次乘法, 总时间为

$$T_0 = 4 \times 3\Delta t + 3 \times 5\Delta t = 27\Delta t$$

加速比

3) 效率

$$\therefore S = \frac{T_0}{T_k} = \frac{27\Delta t}{17\Delta t} = 1.88$$

$$E = \frac{\text{有效时空区面积}}{\text{全部时空区总面积}} = \frac{3 \times 4\Delta t + 5 \times 3\Delta t}{6 \times 17\Delta t} = \frac{27}{102} = 0.264$$

例2 一条段数为4的流水线，无条件分支在第二个时钟周期结束时就被解析出来，条件分支要到第三个时钟周期结束时才能够被解析出来。所有类型的指令都必须经过第一个流水段的处理。问在没有任何控制相关的情况下，该流水线相对于存在上述控制相关情况下的加速比是多少？假设各种分支指令数占有所有指令数的百分比如下：

条件分支	20% (其中的60%是分支成功的)
跳转和调用	5%

解：没有控制相关时流水线的平均CPI=1

存在控制相关时：由于无条件分支在第二个时钟周期结束时被解析出来，而条件分支要到第3个时钟周期结束时才能被解析出来。所以：

（1）若使用排空流水线的策略，则对于条件分支，有两个额外的stall，对无条件分支，有一个额外的stall：

$$\text{CPI} = 1 + 20\% * 2 + 5\% * 1 = 1.45$$

$$\text{加速比} S = \text{CPI} / 1 = 1.45$$

（2）若使用预测分支成功策略，则对于不成功的条件分支，有两个额外的stall，对无条件分支和成功的条件分支，有一个额外的stall 1（PP81, 图 3.27）

$$\text{CPI} = 1 + 20\% * (60\% * 1 + 40\% * 2) + 5\% * 1 = 1.33$$

$$\text{加速比} S = \text{CPI} / 1 = 1.33$$

(3) 若使用预测分支失败策略，则对于成功的条件分支，有两个额外的stall；对无条件分支，有一个额外的stall；对不成功的条件分支，其目标地址已经由PC 值给出，不必等待，所以无延迟：

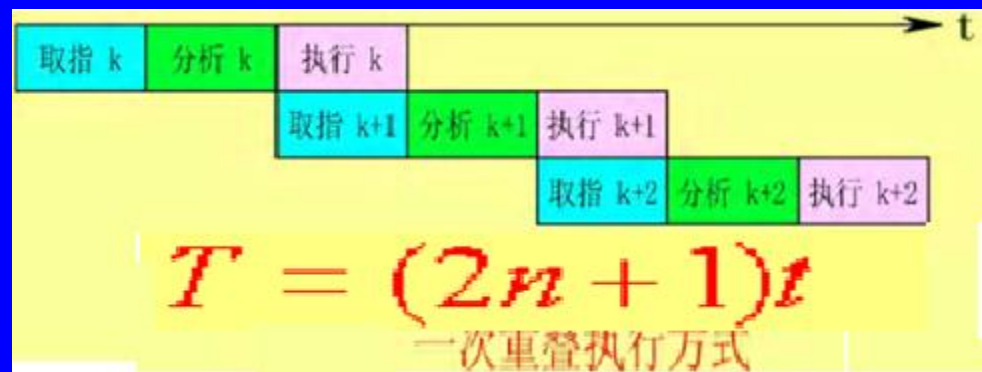
$$\text{CPI} = 1 + 20\% * (60\% * 2 + 40\% * 0) + 5\% * 1 = 1.29$$

$$\text{加速比} S = \text{CPI} / 1 = 1.29$$

例3 设一条指令的执行过程分为取指令、指令分析、指令执行三个子过程，且这三个子过程延迟时间相等并 t 。

假设某过程共有 $n=1000$ 条指令，写出如下三种情况下机器执行程序所需时间和加速比。

- (1) 指令顺序执行方式；
- (2) 一次重叠执行方式；
- (3) 二次重叠执行方式。



解:

(1) 顺序执行方式

$$T1 = 3 \times 1000 \times t = 3000t$$

(2) 一次重叠执行方式

$$T2 = (2n+1) t = (2 \times 1000 + 1) t = 2001t$$

(3) 二次重叠执行方式

$$T3 = (n+2) t = (1000 + 2) t = 1002t$$

(4) $T1 > T2 > T3$

加速比 :

$$s1=1$$

$$s2=3000/2001=1.5$$

$$S3=3000/1002=3.0$$

第五章 指令级并行及其开发

本章研究：如何利用各种技术来开发更多的指令级并行（硬件的方法）

- ◆ 指令级并行度ILP

- ◆ 开发ILP的方法可以分为两大类

 - ★ 主要基于硬件的动态开发方法（动态调度）

 - 记分牌动态调度算法

 - Tomasulo算法

 - ★ 基于软件的静态开发方法（静态调度）

- ◆ 理想流水线的CPI加上各类停顿的时钟周期数：

$$CPI_{\text{流水线}} = CPI_{\text{理想}} + \text{停顿}_{\text{结构冲突}} + \text{停顿}_{\text{数据冲突}} + \text{停顿}_{\text{控制冲突}}$$

◆ 动态分支预测技术

分支历史表BHT

分支目标缓冲器BTB

◆ 超标量处理机

◆ 超流水线处理机

◆ 超长指令字 (VLIW)处理机

数据相关及其处理技术

流水线时空图

超标量流水线调度策略及时空图

1.按序流出按序完成； 2.按序流出无序完成； 3.无序流出

记分牌动态调度算法

基本思想:记分牌硬件实现了对指令的动态调度。支持乱序执行,在没有**结构冲突**时尽早地执行没有数据冲突的指令,使多条指令同时处于执行阶段。

◆记分牌维护**3**张表:

指令的执行状态

功能部件状态

寄存器状态。

◆为了乱序执行,译码段**ID**分解成**流出**和**读操作数**
流出:指令译码,检查是否存在结构冲突。
读操作数:等待数据冲突消失,然后读操作数。

每条指令的执行过程分为4段（主要考虑浮点操作）

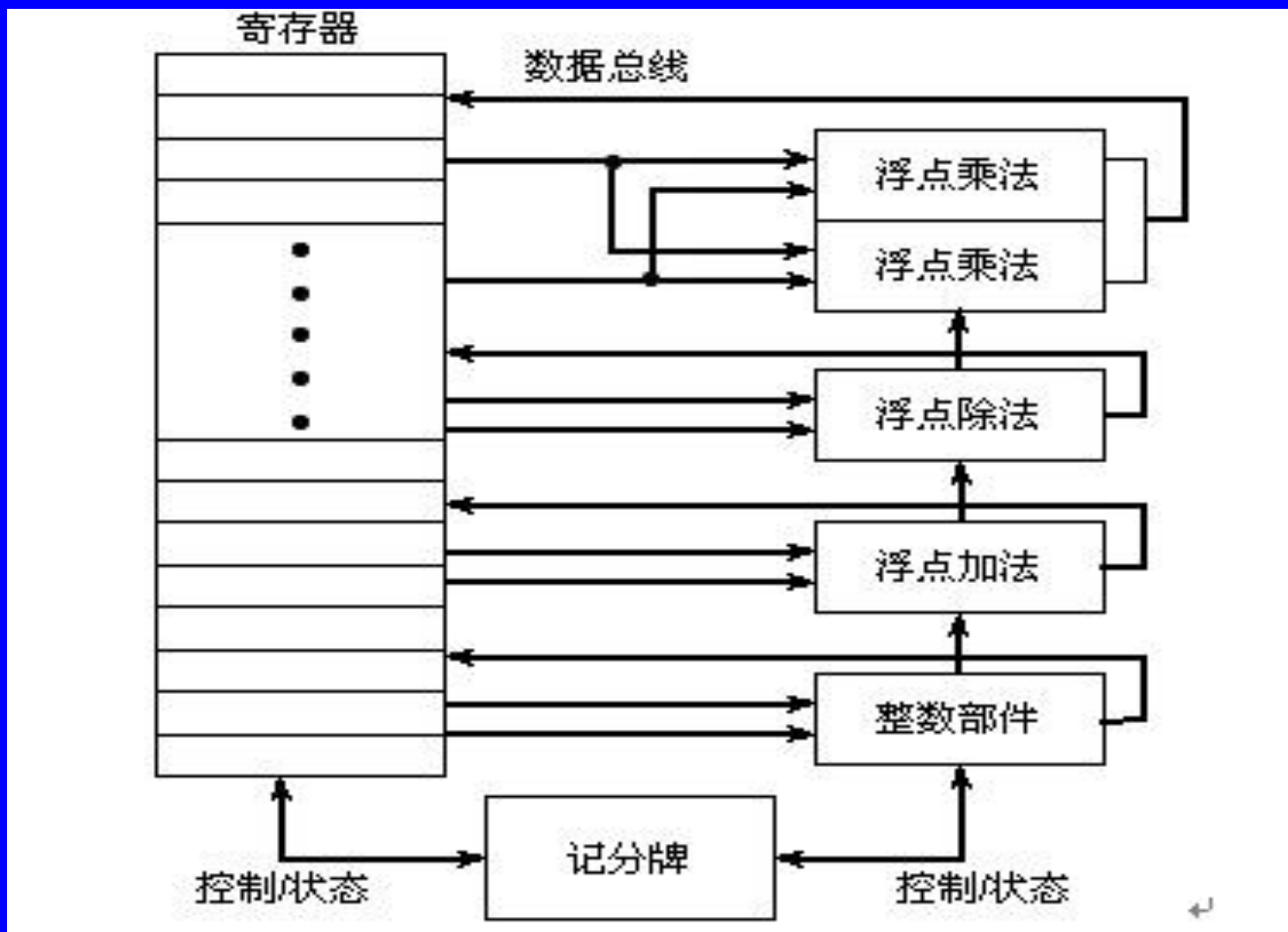
流出：如果当前流出指令所需的功能部件空闲，并且所有其他正在执行的指令的目的寄存器与该指令的不同，就向功能部件流出该指令，并修改记分牌内部记录表。解决了WAW冲突

读操作数：监测源操作数的可用性，如果数据可用，就从寄存器中读出源操作数并开始执行。
解决了RAW冲突，导致乱序执行。

执行：取到操作数后，功能部件开始执行。当产生出结果后，就通知记分牌它已经完成执行。

在浮点流水线中，这一段要占用多个时钟周期。

写结果



记分牌的MIPS处理器的基本结构

指令的记分牌调度

★ 记分牌中记录的信息由3部分构成

- ◆ 指令状态表：记录正在执行的各条指令已进入到哪一段。
- ◆ 结果寄存器状态表：指出哪个功能部件（编号）将把结果写入该寄存器。
- ◆ 功能部件状态表：记录各个功能部件的状态。每个功能部件有一项，每一项由以下9个字段组成：
 - ★ **Busy**：忙标志，指出功能部件是否忙。初值为“no”；
 - ★ **Op**：该功能部件正在执行或将要执行的操作；
 - ★ **Fi**：目的寄存器编号；
 - ★ **Fj, Fk**：源寄存器编号；
 - ★ **Qj, Qk**：指出向源寄存器Fj、Fk写数据的功能部件；
 - ★ **Rj, Rk**：标志位，“yes”表示Fj, Fk中的操作数就绪且还未被取走。否则就被置为“no”。

下述指令，第一条指令已经执行并写入了结果，第二条指令已完成正等待写结果，给出记分牌（或Tomasulo）法所用的指令、功能部件（或保留站）和结果寄存器状态表。（假定各功能部件 均为1个）

L.D F6, 34(R2)

L.D F2, 45(R3)

MULT.D F0, F2, F4

SUB.D F8, F6, F2

DIV.D F10, F0, F6

ADD.D F6, F8, F2

解： (1) 指令状态表 （假定各功能部件 均为1个）

指 令	指令状态表			
	流出	读操作数	执行	写结果
L.D F6,34(R2)	√	√	√	√
L.D F2, 45(R3)	√	√	√	
MULT.D F0, F2, F4	√			
SUB.D F8, F6, F2	√			
DIV.D F10, F0, F6	√			
ADD.D F6, F8, F2				

L. D F6, 34 (R2)
 L. D F2, 45 (R3)
 MULT. D F0, F2, F4
 SUB. D F8, F6, F2
 DIV. D F10, F0, F6

部件名称	功能部件状态表								
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	yes	L.D	F2	R3				no	
Mult	yes	MULT.D	F0	F2	F4	Integer		no	yes
Add	yes	SUB.D	F8	F6	F2		Integer	yes	no
Divide	yes	DIV.D	F10	F0	F6	Mult1		no	yes

- ★ **Busy**: 指出功能部件是否忙。初值为“no”;
- ★ **Op**: 该功能部件正在执行或将要执行的操作; (no) (yes)
- ★ **Fi**: 目的寄存器编号;
- ★ **Fj, Fk**: 源寄存器编号;
- ★ **Qj, Qk**: 向源寄存器Fj、Fk写数据的功能部件;
- ★ **Rj, Rk**: 标志位, Fj, Fk中的操作数就绪且未被取走?

L. D	F6, 34 (R2)
L. D	F2, 45 (R3)
MULT. D	F0, F2, F4
SUB. D	F8, F6, F2
DIV. D	F10, F0, F6
ADD. D	F6, F8, F2

	结果寄存器状态表							
	F0	F2	F4	F6	F8	F10	...	F30
部件名称	Mult1	Integer			Add	Divide		

假设MULT. D准备写结果，则

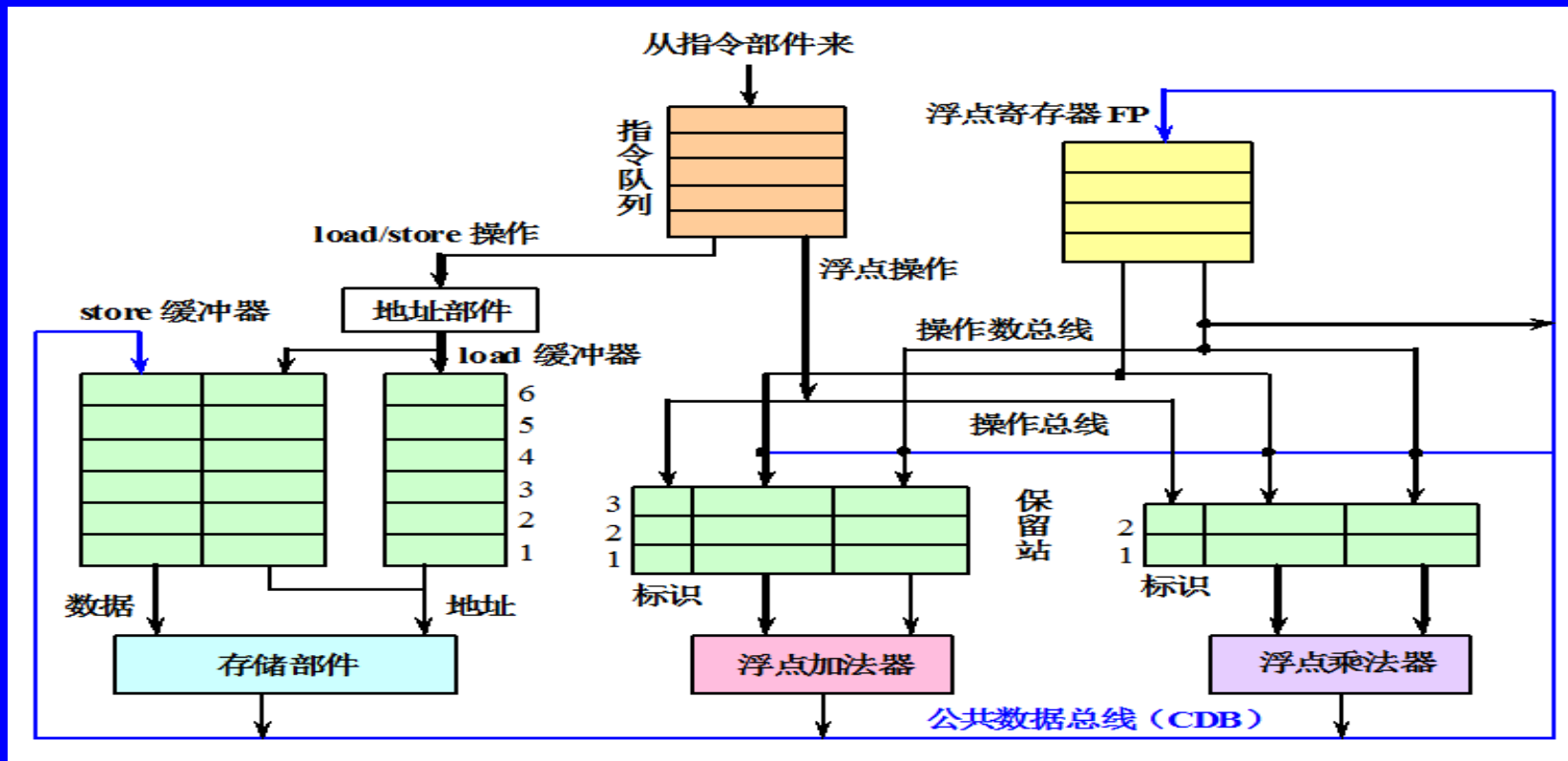
指 令		指令状态表			
		流出	读操作数	执行	写结果
L.D	F6, 34(R2)	√	√	√	√
L.D	F2, 45(R3)	√	√	√	√
MULT.D	F0, F2, F4	√	√	√	
SUB.D	F8, F6, F2	√	√	√	√
DIV.D	F10, F0, F6	√			
ADD.D	F6, F8, F2	√	√	√	

L. D	F6, 34 (R2)
L. D	F2, 45 (R3)
MULT. D	F0, F2, F4
SUB. D	F8, F6, F2
DIV. D	F10, F0, F6

部件名称	功能部件状态表								
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	no								
Mult1	yes	MULT.D	F0	F2	F4			no	no
Mult2	no								
Add	yes	ADD.D	F6	F8	F2			no	no
Divide	yes	DIV.D	F10	F0	F6	Mult1		no	yes

	结果寄存器状态表							
	F0	F2	F4	F6	F8	F10	...	F30
部件名称	Mult1			Add		Divide		

Tomasulo算法



保留站：保存已经流出并等待到本功能部件执行的指令，在保留站通过流出逻辑来完成的寄存器换名（顺序流出，乱序执行）

公共数据总线CDB：所有功能部件计算结果都送到CDB，由它把这些结果直接送到各个需要该结果的地方（乱序完成）

指令状态表,保留站及寄存器状态表与记分牌类似

指 令		指令状态表		
		流出	执行	写结果
L.D	F6 , 34(R2)	√	√	√
L.D	F2 , 45(R3)	√	√	
MUL.D	F0 , F2 , F4	√		
SUB.D	F8 , F6 , F2	√		
DIV.D	F10 , F0 , F6	√		
ADD.D	F6 , F8 , F2	√		

名称	保留站						
	Busy	Op	Vj	Vk	Qj	Qk	A
Load1	no						
Load2	yes	LD					45+Regs[R3]
Add1	yes	SUB		Mem[34+Regs[R2]]	<u>Load2</u>		
Add2	yes	ADD			<u>Add1</u>	<u>Load2</u>	
Add3	no						
Mult1	yes	MUL		Reg[F4]	<u>Load2</u>		
Mult2	yes	DIV		Mem[34+Regs[R2]]	<u>Mult1</u>		

	寄存器状态表							
	F0	F2	F4	F6	F8	F10	...	F30
Qi	Mult1	Load2		Add2	Add1	Mult2	...	

例题： 在一台单流水线处理机上执行下面程序。指令经过**取指、译码、执行、写结果**四个流水段，每个流水段延迟时间**5ns**。但**LS**和**ALU**部件的执行段只能一个工作，**LS**部件完成**LOAD**和**STORE**操作，**ALU**部件完成其它操作。两个操作部件的输出端和输入端有直接输出通路相互切换连接，**ALU**部件产生的条件码能直接送入控制器。假定采用**静态分支预测技术**，每次都**预测转移不成功**。画出指令流水线的时空图。

I1 SUB R0, R0 ; $R0 \leftarrow 0$

I2 LOAD R1, #8 ; $R1 \leftarrow$ 向量长度 8

I3 LOOP: LOAD R2, A (R1) ; A: 向量的一个元素

I4 MUL R2, R1 ; $R2 \leftarrow (R2) \times (R1)$

I5 ADD R0, R2 ; $R0 \leftarrow (R0) + (R2)$

I6 DNE R1, LOOP ; $R1 \leftarrow R1 - 1$, 若 $(R1) \neq 0$ 转向 LOOP

I7 STORE R0, S ; 保存结果



每次预测不成功 最后一次成功 $8 + 7 * 6$ (错) + 2 (对) = 52

- 乘法时加法部件空闲，加法时乘法部件空闲
- 静态流水线：乘法完成时进入功能转换

首次条件码在时间 $T=8$ ，预测 (不) 成功

第四章 向量流水处理机

向量处理方法

横向处理/ 纵向处理/ 纵横处理:

向量流水处理机结构

1 存储器-存储器结构: 纵向处理

2 寄存器寄存器结构: 纵横处理

提高向量处理机性能的方法

多功能部件的并行操作

链接技术 WD相关

分段开采

多处理机系统结构

向量处理机性能的主要参数

- ◆ 一条向量指令的处理时间
- ◆ 每秒多少个浮点运算结果 (MFLOP或一个浮点运算的时间)
- ◆ 一组向量指令的处理时间
- ◆ 向量流水线的最大性能 R_{∞}

2021/6/23

链接技术：具有先写后读相关的两条指令，在不出现功能部件冲突和 V_i 冲突的情况下，可以把功能部件链接起来进行流水处理，以达到加快执行的目的。

例1：某向量机上进行向量运算 $D=A \times (B+C)$,假设向量长度 $N < 64$,且 B 和 C 已经存至 $V0$ 和 $V1$,则下面三条指令就可以完成上述的运算。

$V3 \leftarrow A$ (6拍)

$V2 \leftarrow V0 + V1$ (6拍)

$V4 \leftarrow V2 \times V3$ (7拍)

分别求三条指令全部用**串行、并行和链接**执行的时间(读写各需1拍),画出链接操作图

(假定数据入、出各功能部件,包括主存,假定需1个时钟周期)

向量长度为N，则不同方法的执行时间为：

1) 3条指令全部用串行方法，执行时间为

$$[(1+6+1)+N-1] + [(1+6+1)+N-1] + [(1+7+1)+N-1] \\ = 3N+22(\text{时钟周期})$$

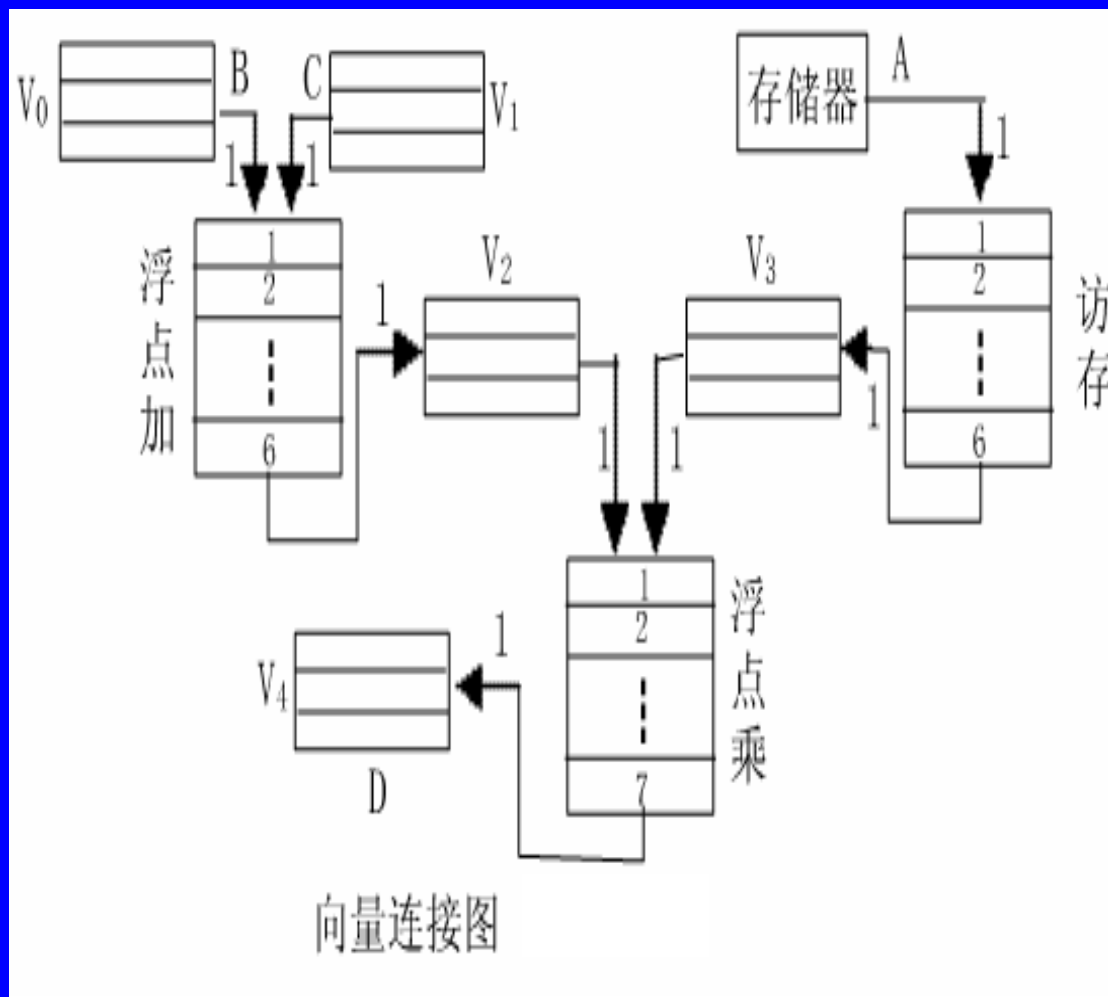
2) 前两条指令并行执行，第3条指令顺序执行，时间为

$$[(1+6+1)+N-1] + [(1+7+1)+N-1] = 2N+15(\text{时钟周期})$$

3) 前两条指令并行执行，并采用链接，时间为

$$(1+6+1)+(1+7+1)+N-1=17+N-1=N+16(\text{时钟周期})$$

链接操作图



- ◆ **例2** 某向量流水机有三个向量访存部件，其中的两个用于向量Load，一个用于向量Store。三个向量访存部件可同时使用。该流水机的向量寄存器长度为64。若要进行向量运算 $Z = X + s * Y$ （s为标量，已保存于标量寄存器S1中），且向量长度N小于64，则下列指令段可完成上述运算：

LOADV	V1, M(X)	;6拍
LOADV	V2, M(Y)	;6拍
MULSV	V3, V2, S1	;7拍
ADDV	V4, <u>V1</u> , V3	;6拍
STOREV	<u>V4</u> , M(Z)	;6拍

分别求出以上指令段采用串行方法（向量访存部件可并行工作）以及链接方法的执行时间（读写寄存器各需1拍），并画出链接操作示意图。

- ◆ 串行：共四个编队，时间为

$$(1+6+1+N-1) + (1+7+1+N-1) + (1+6+1+N-1) + (1+6+1+N-1) =$$

- ◆ 链接：共两个链接编队，时间为

$$(1+6+1+N-1) + (1+7+1) + (1+6+1+N-1) + (1+6+1) =$$

LOADV V1, M(X) ;6拍

LOADV V2, M(Y) ;6拍

MULSV V3, V2, S1 ;7拍

ADDV V4, V1, V3 ;6拍

STOREV V4, M(Z) ;6拍

- ◆ I1,I2,I3为一链接编队，I4,I5为一链接编队

第9章 互连网络

- 9.1 互连函数
- 9.2 互连网络的结构参数与性能指标
- 9.3 静态互连网络
- 9.4 动态互连网络
- 9.5 消息传递机制

第9章 互连网络

互连网络:静态 \ 动态互连网络

动态网络: 总线网络、多级互连网络和交叉开关网络

互连网络三要素: 互联结构、开关和控制方式。

基本互联函数

PM2I移数函数

$$PM2+i(x) = \langle x+2^i \rangle \bmod N$$

$$PM2-i(x) = \langle x-2^i \rangle \bmod N$$

◆ 交换函数 均匀洗牌函数

网络参数: 网络规模、结点度、距离、直径, 等分宽度

路由选择和消息传递方法: 线路交换和包交换 (存储转发, 虚拟直通、虫孔方式):

流量控制策略和通信模式

通道流量用传输消息所使用的通道数来表示。

通信时延用包的最长传输时间来表示。

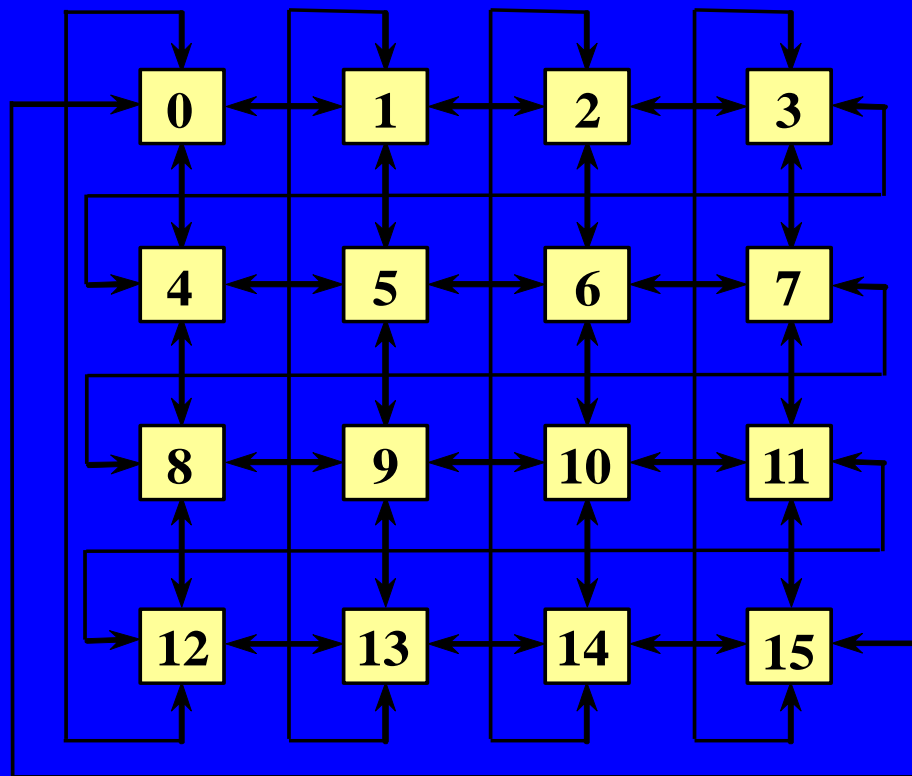
优化寻径网络以最小**通道流量**或**通信时延**为目标。

最小流量 / 最小时延

典型互联网络: 立方体型 和 Illiac网, Omega网络 (混洗函数)

网格的变形--- ILLIAC IV 网

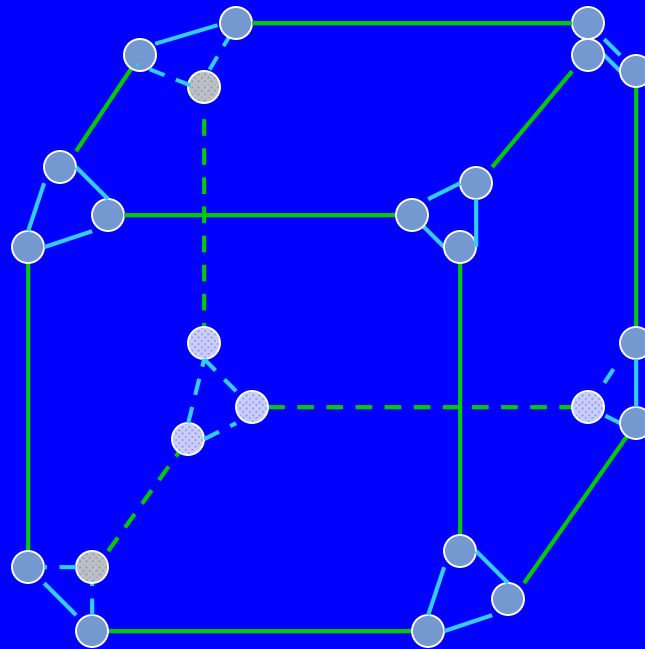
采用 $PM2_{\pm 0}$ 和 $PM2_{\pm n/2}$ 构成其互连网络，实现各处理单元之间的上下左右互连。



用移数函数构成ILLIAC IV 阵列机的互连网络

2021/6/26 每行尾与下一行的头，每列尾与下一列的头相接-网格卷绕。

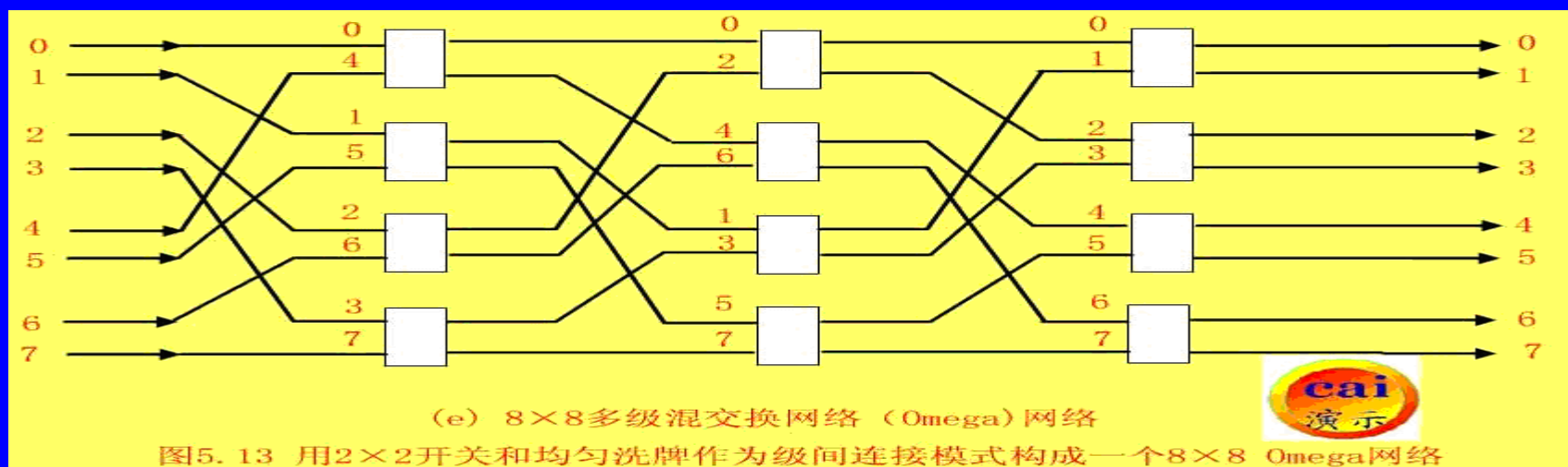
带环立方体（CCC）



带环3-立方体

- ★ 一个带环 n -立方体由 $N = 2^n$ 个结点环构成，每个结点环是一个有 n 个结点的环，结点总数为 $n \cdot 2^n$ 个。直径通常为 $2n$ ，结点度为 3 ，对称。

- ♦ Omega网络:
- ♦ 8×8 Omega网络, 有3级 2×2 开关。网络左有8个输入, 右侧有8个输出。级间连接(ISC)是对8个对象的均匀洗牌模式。



- ♦ n 输入的Omega网络需要 $\log_2 n$ 级 2×2 开关, 每级要用 $n/2$ 个开关模块, 网络共需 $n/2 \log_2 n$ 个开关。每个开关模块采用单元控制方式。

第13章 阵列处理机

- ◆ 阵列处理机
- ◆ 阵列处理机的结构
 - 分布式存储器的阵列机:
 - 共享存储器的阵列机:
- ◆ 阵列处理机的特点(与流水线向量对比)
- SIMD机与并行算法的关系
 - 解有限差分方程
 - 递归折叠求和算法

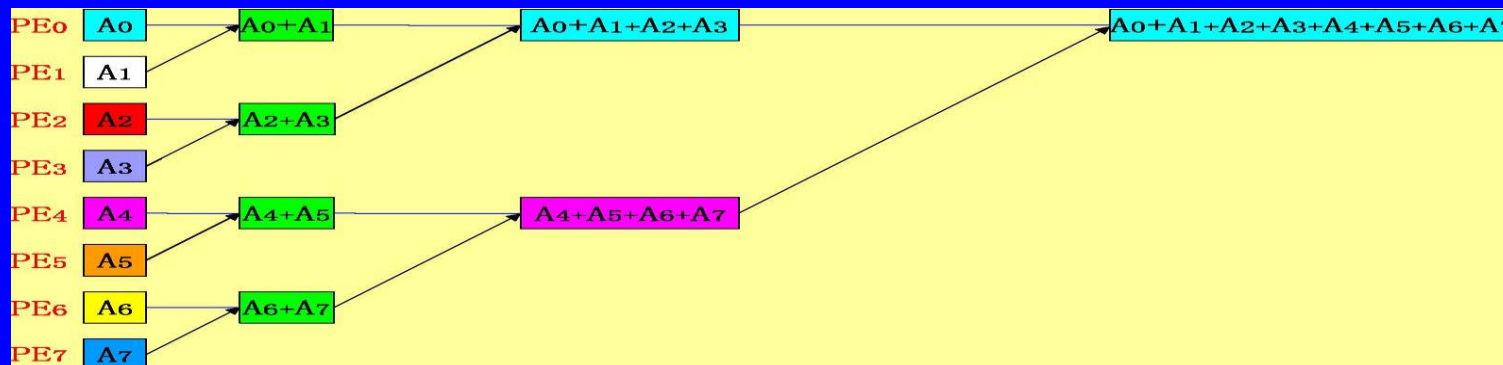


图6.12 8个PE的递归折叠求和算法

例1. 试分别在下面两种计算机系统中用最短的时间来计算表达式 $s=A1*B1+A2*B2+...A32*B32$ 。假设加法和乘法分别需要两个和四个单位时间，从存储器取指令、取数据、译码的时间忽略不计，所有的指令和数据已装入有关的PE。试确定下列每种情况的最小计算时间：

- 1 一台SISD串行计算机。
- 2 一台有8个PE的SIMD计算机，8个PE用移数函数PM2I连接。每个PE用一个单位时间可以把数据直接送给它的相邻PE。操作数 A_i 和 B_i 最初存放在 $PE_{i \bmod 8}$ 中，其中 $i=1, 2, \dots, 32$ 。每个PE可在不同时刻执行加法或乘法

解 1 在SISD计算机中,需要32次乘法和31次加法。
共需要时间: $T=4*32+2*31=190$ 单位时间

2 在SIMD计算机上计算的算法:

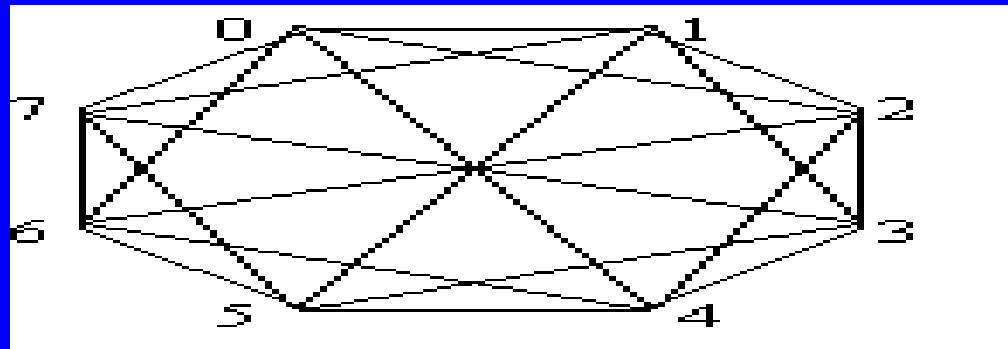
- ◆ 假定向量中的32对元素平均地分配到8个处理器中,每个处理器分配4对, 则每个处理器计算时间为
 $4*4+3*2$

总时间 =每个处理器计算时间+递归折叠求和算法

$$T=4*4+3*2+1+2+1+2+1+2=31$$

PM2I计算机

$$\begin{aligned} \text{PM2} + i(x) &= (x + 2^i) \bmod N \\ \text{PM2} - i(x) &= (x - 2^i) \bmod N \end{aligned}$$



第10章 多处理机

多处理机概念:

- ◆ **MIMD**计算机的特点、分类
- ◆ **Cache**的一致性问题 and 原因
- ◆ 监听协议法 、 目录表协议

多处理机分类

- ◆ 紧耦合系统 \ 松耦合系统
- ◆ 同构型 \ 异构型 多处理机系统。
- ◆ 按系统组成**结构**
 - 并行向量处理机(PVP)
 - 对称多处理机(SMP)
 - 大规模并行处理机(MPP)
 - 分布共享存储器多处理机(DSM)
 - 工作站机群(COW)

注意事项

- 1 认真复习，多看书，适当作题；重点与难点
- 2 理解重要概念，原理、方法，分类, 各章摘要和小节
- 3 答疑时间地点： 考前一日 6月28日 教三 1017
上午 11: 00—12:00
下午 1: 00 ---3:30
- 4 平时与期末成绩

预祝大家考试顺利！

谢 谢 ！