

第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构

4.3 提高向量处理机性能的方法

4.4 向量处理机性能评价

4.5 向量处理机实例



2022年睿思芯科的
高性能RISC-V向量处理器

4.1 向量处理方式

- **向量处理机：**

- 数据级并行DLP：向量数据表示（SIMD）

- 向量指令

- （指令级并行ILP、线程级并行TLP）

- **标量处理机：**不具有向量数据表示和相应的向量指令的流水线处理机

- 在同一时间内只处理一条数据（整数或浮点数），标量处理器是一种单指令流单数据流结构（SISD）

4.1 向量处理方式

3.5.1 向量处理方式

- 以计算表达式 $D = A \times (B + C)$ 为例

A 、 B 、 C 、 D ：长度为 N 的向量

(1) 横向(水平)处理方式

(2) 纵向 (垂直)处理方式

(3) 纵横 (分组)处理方式

4.1 向量处理方式

(1) 横向(水平)处理方式

- 向量计算是按行的方式从左到右横向地进行

- 先计算: $d_1 \leftarrow a_1 \times (b_1 + c_1)$
- 再计算: $d_2 \leftarrow a_2 \times (b_2 + c_2)$
.....
- 最后计算: $d_N \leftarrow a_N \times (b_N + c_N)$

- 组成循环程序进行处理

$$k_i \leftarrow b_i + c_i$$

$$d_i \leftarrow k_i \times a_i$$

➤ 数据相关: N 次 功能切换: $2N$ 次

- 不适合于向量处理机的并行处理

4.1 向量处理方式

(2) 纵向 (垂直)处理方式

- 向量计算是按列的方式从上到下纵向地进行。

$$\begin{array}{l} \text{先计算} \left\{ \begin{array}{l} k_1 \leftarrow b_1 + c_1 \\ \dots\dots \\ k_N \leftarrow b_N + c_N \end{array} \right. \quad \text{再计算} \left\{ \begin{array}{l} d_1 \leftarrow k_1 \times a_1 \\ \dots\dots \\ d_N \leftarrow k_N \times a_N \end{array} \right. \end{array}$$

- 表示成向量指令：

$$K = B + C$$

$$D = K \times A$$

- 两条向量指令之间数据相关：1次 功能切换：1次

4.1 向量处理方式

(3) 纵横 (分组)处理方式

- 把向量分成若干组，组内按纵向方式处理，依次处理各组
- 对于上述的例子，设：

$$N = S \times n + r$$

- 其中 N 为向量长度, S 为组数, n 为每组的长度, r 为余数
- 若余下的 r 个数也作为一组处理，则共有 $S+1$ 组
- 运算过程为：

4.1 向量处理方式

➤ 先算第1组:

$$k_{1 \sim n} \leftarrow b_{1 \sim n} + c_{1 \sim n}$$

$$d_{1 \sim n} \leftarrow k_{1 \sim n} \times a_{1 \sim n}$$

➤ 再算第2组:

$$k_{(n+1) \sim 2n} \leftarrow b_{(n+1) \sim 2n} + c_{(n+1) \sim 2n}$$

$$d_{(n+1) \sim 2n} \leftarrow k_{(n+1) \sim 2n} \times a_{(n+1) \sim 2n}$$

➤ 依次进行下去，直到最后一组：第S+1组

➤ 每组内各用两条向量指令

数据相关：1次 功能切换：2次

第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构 →

1. 存储器-存储器型

2. 寄存器-寄存器型

4.3 提高向量处理机性能的方法

4.4 向量处理机性能评价

4.5 向量处理机实例

4.2 向量处理机的结构

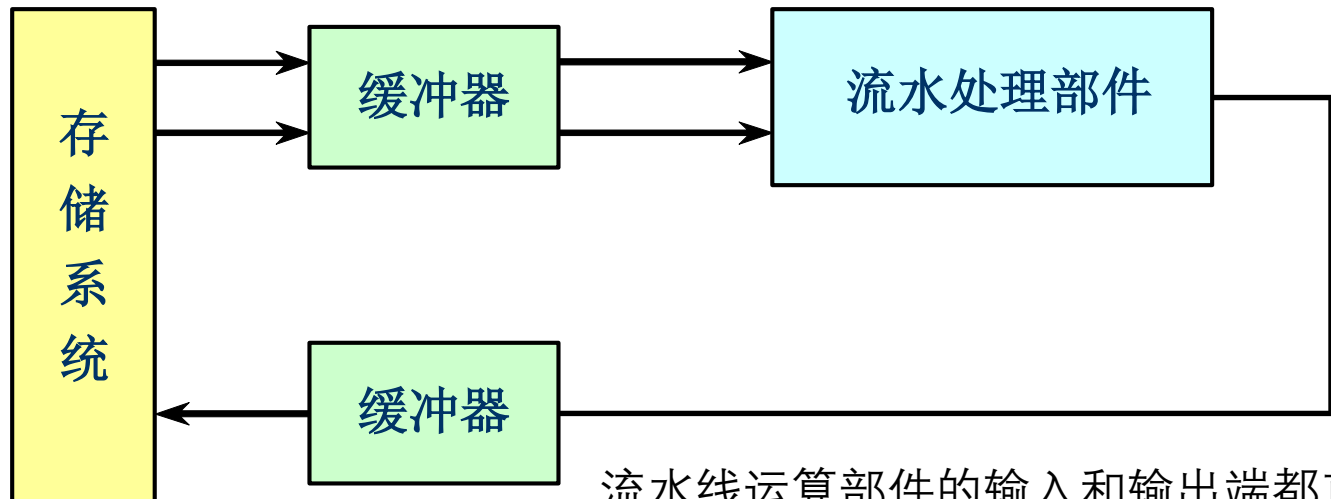
- 向量处理机的结构因具体机器不同而不同
由所采用的向量处理方式决定。
- 有两种典型的结构
 - 存储器-存储器型结构：纵向处理方式采用
 - 寄存器-寄存器型结构：分组处理方式采用

4.2 向量处理机的结构

- 存储器－存储器结构

- 向量指令的源向量和目的向量都存放在存储器中，运算的中间结果需要送回存储器。
- 存储器－存储器型操作的运算流水线

例如：STAR-100、CYBER-205



流水线运算部件的输入和输出端都直接（或经过缓冲器）与存储器相联

4.2 向量处理机的结构

- 要充分发挥存储器-存储器结构流水线效率，存储器要不断地提供源操作数，并不断地从运算部件接收结果
(每拍从存储器读取两个数据，并向存储器写回一个结果)
- ✓ 对存储器的带宽以及存储器与处理部件的通信带宽提出了非常高的要求。
- ✓ **解决方法：**采用多体交叉并行存储器和缓冲器技术
例如，70年代初问世的**Star 100**
 - 存储器：**32个**体交叉
 - 每个体的数据宽度：**8个字**（字长**64位**）
 - 最大数据流量：**每秒2亿字**

4.2 向量处理机的结构

- 寄存器－寄存器结构

在向量的分组处理方式中，对向量长度 N 没有限制，但组的长度 n 却是固定不变的。

- 设置能快速访问的向量寄存器，用于存放源向量、目的向量及中间结果，让运算部件的输入、输出端都与向量寄存器相联，构成寄存器－寄存器型操作的运算流水线。
- 典型的寄存器－寄存器结构的向量处理机

美国的CRAY-1、我国的YH-1巨型机

4.2 向量处理机的结构

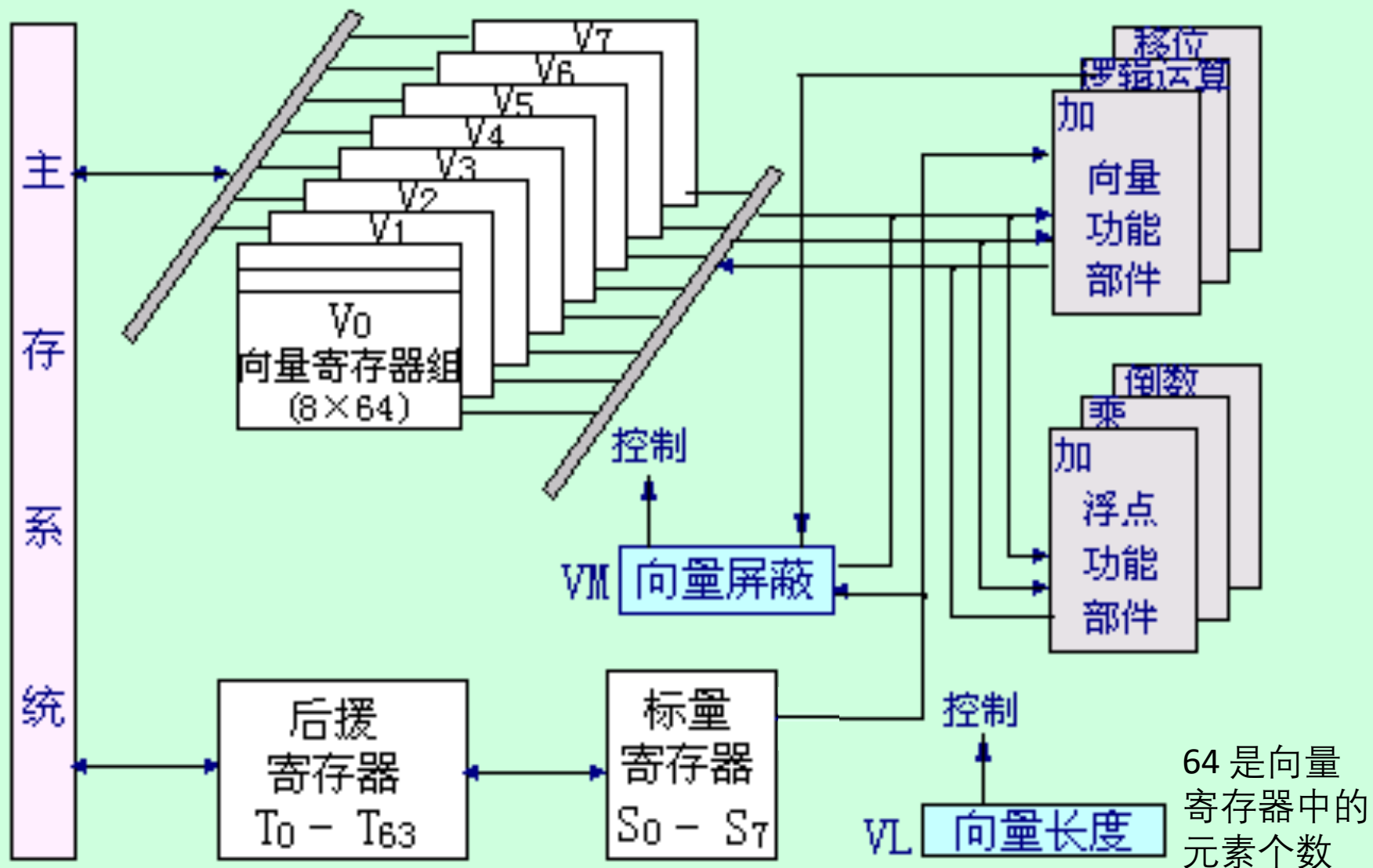
- 以CRAY-1机为例

- 美国CRAY公司，1976年
- 每秒1亿次浮点运算
- 时钟周期：12.5 ns

- CRAY-1的基本结构

- 功能部件：共有12条可并行工作的单功能流水线，可分别流水地进行地址、向量、标量的各种运算。

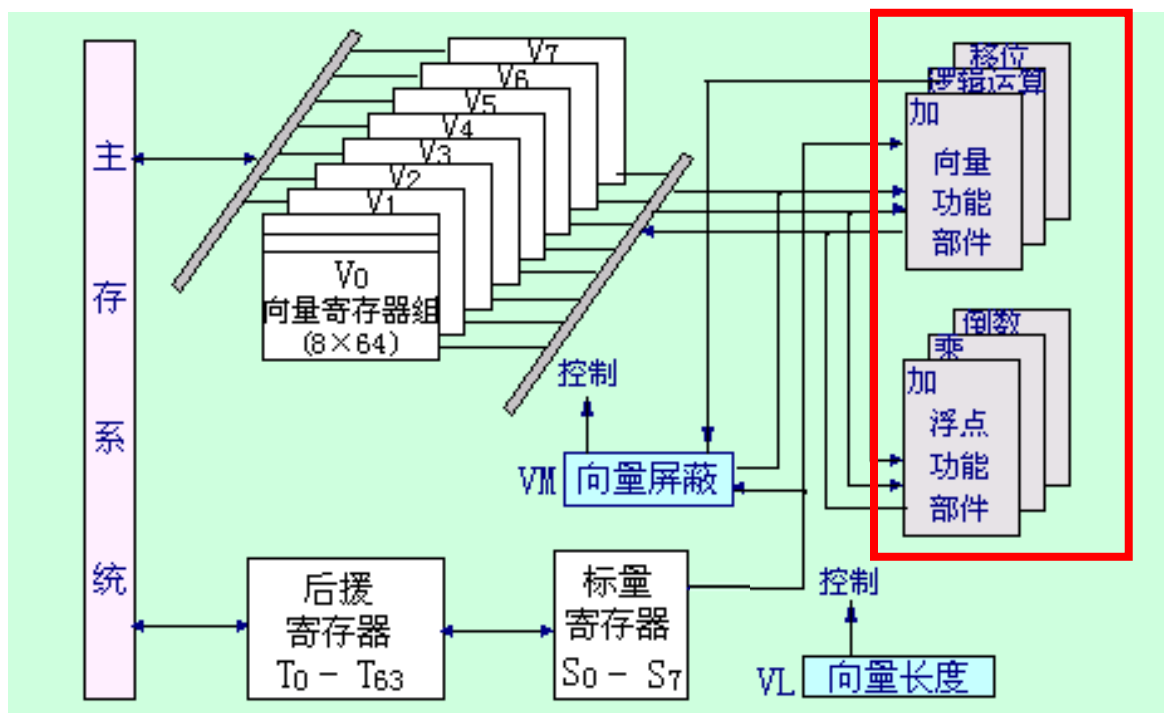
4.2 向量处理机的结构：CRAY-1



4.2 向量处理机的结构：CRAY-1

➤ 6个单功能流水部件：进行向量运算

- ✓ 整数加（3拍）
- ✓ 浮点加（6拍）
- ✓ 逻辑运算（2拍）
- ✓ 浮点乘（7拍）
- ✓ 移位（4拍）
- ✓ 浮点迭代求倒数（14拍）

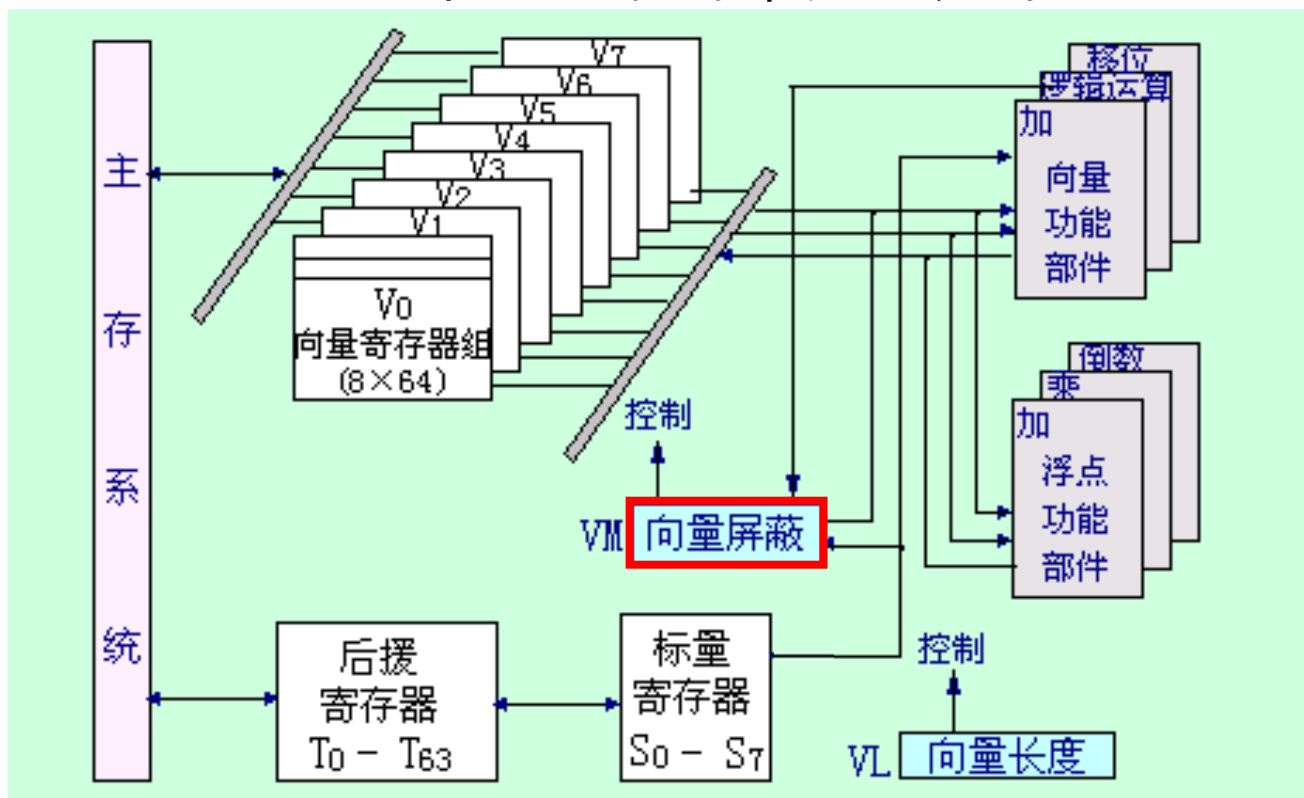


括号中的数字为其流水经过的时间，每拍为一个时钟周期，即12.5 ns。

4.2 向量处理机的结构：CRAY-1

➤ 向量屏蔽寄存器VM

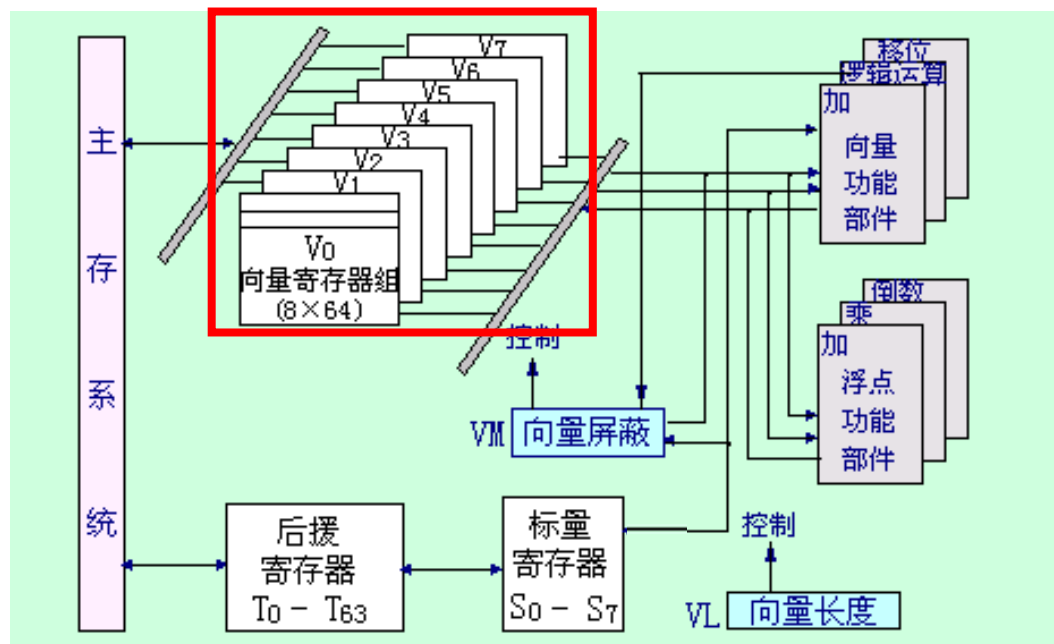
- ✓ 64位：每一位对应于向量寄存器的一个单元
- ✓ 作用：用于向量的归并、压缩、还原和测试操作、对向量某些元素的单独运算等



4.2 向量处理机的结构：CRAY-1

➤ 向量寄存器组V

- 由**512**个**64**位的寄存器组成，分成8块
- 编号： $V_0 \sim V_7$
- 每一个块称为一个向量寄存器，可存放一个长度（即元素个数）不超过**64**的向量

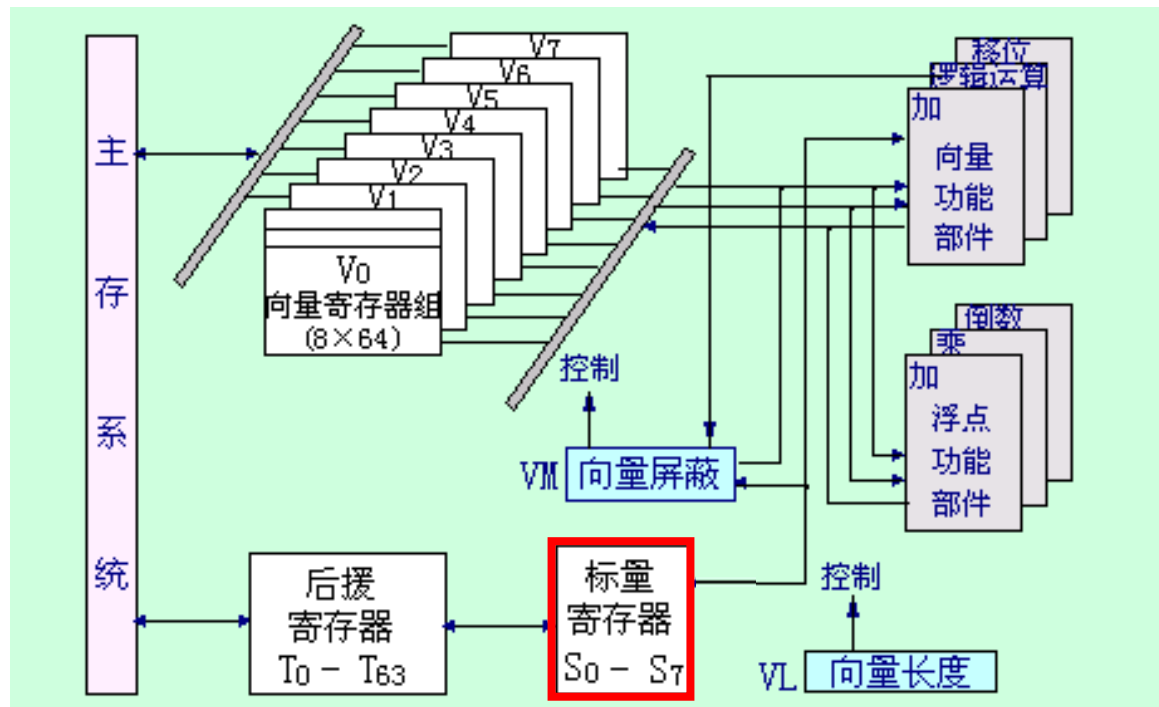


每个向量寄存器可以每拍向功能部件提供一个数据元素，或者每拍接收一个从功能部件来的结果元素

4.2 向量处理机的结构：CRAY-1

➤ 标量寄存器S和快速暂存器T

- 标量寄存器有8个： $S_0 \sim S_7$ 64位
- 快速暂存器T用于在标量寄存器和存储器之间提供缓冲。



4.2 向量处理机的结构：CRAY-1

● CRAY-1向量处理的显著特点

- 每个向量寄存器 V_i 都有连到6个向量功能部件的单独总线
- 每个向量功能部件也都有把运算结果送回向量寄存器组的总线
- 只要不出现 V_i 冲突和功能部件冲突，各 V_i 之间和各功能部件之间都能并行工作，大大加快了向量指令的处理

4.2 向量处理机的结构：CRAY-1

- **V_i 冲突**：并行工作的各向量指令的源向量或结果向量使用了相同的 V_i

例如：源向量相同

$$V_3 \leftarrow V_1 + V_2$$
$$V_5 \leftarrow V_4 \wedge V_1$$

- **功能部件冲突**：并行工作的各向量指令要使用同一个功能部件

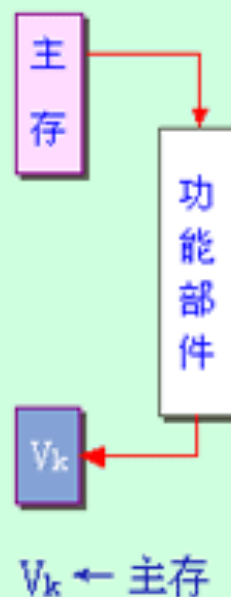
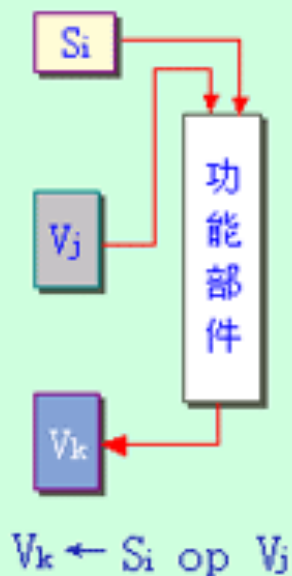
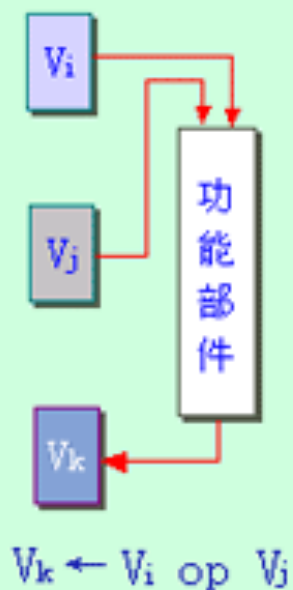
例如：都需使用乘法功能部件

$$V_3 \leftarrow V_1 \times V_2$$
$$V_5 \leftarrow V_4 \times V_6$$

4.2 向量处理机的结构：CRAY-1

● CRAY-1向量指令类型

- $V_k \leftarrow V_i \text{ op } V_j$
- $V_k \leftarrow S_i \text{ op } V_j$
- $V_k \leftarrow \text{主存}$
- $\text{主存} \leftarrow V_i$



第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构

4.3 提高向量处理机性能的方法 ➡

4.4 向量处理机性能评价

4.5 向量处理机实例

1. 多功能部件
2. 链接
3. 循环开采
4. 多处理机

4.3 提高向量处理机性能的方法

- (1) 设置多个功能部件，使它们并行工作；
- (2) 采用**链接技术**，加快一串向量指令的执行；
- (3) 采用**循环开采技术**，加快循环的处理；
- (4) 采用多处理机系统，进一步提高性能。

4.3 提高向量处理机性能的方法

(1) 设置多个功能部件

- 设置多个独立的功能部件。这些部件能并行工作，并各自按流水方式工作，从而形成了多条并行工作的运算操作流水线。

例如：CRAY-1向量处理机有4组12个单功能流水部件：

- ✓ 向量部件：向量加，移位，逻辑运算
- ✓ 浮点部件：浮点加，浮点乘，浮点求倒数
- ✓ 标量部件：标量加，移位，逻辑运算，数“1”/计数
- ✓ 地址运算部件：整数加，整数乘

4.3 提高向量处理机性能的方法

(2) 链接技术

- **链接特征：**具有先写后读相关的两条指令，在不出现功能部件冲突和源向量冲突的情况下，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。
- **链接特性的实质**
把流水线**定向/转发**的思想引入到向量执行过程的结果

4.3 提高向量处理机性能的方法

- **假设：**把向量数据元素送往向量功能部件以及把结果存入向量寄存器需要一拍时间，从存储器中把数据送入访存功能部件需要一拍时间。

例3 考虑在Cray-1上利用链接技术执行以下4条指令：

$V_0 \leftarrow \text{存储器}$	// 访存取向量：7拍
$V_2 \leftarrow V_0 + V_1$	// 向量加：3拍
$V_3 \leftarrow V_2 \ll A_3$	// 按 (A_3) 左移：4拍
$V_5 \leftarrow V_3 \wedge V_4$	// 与操作：2拍

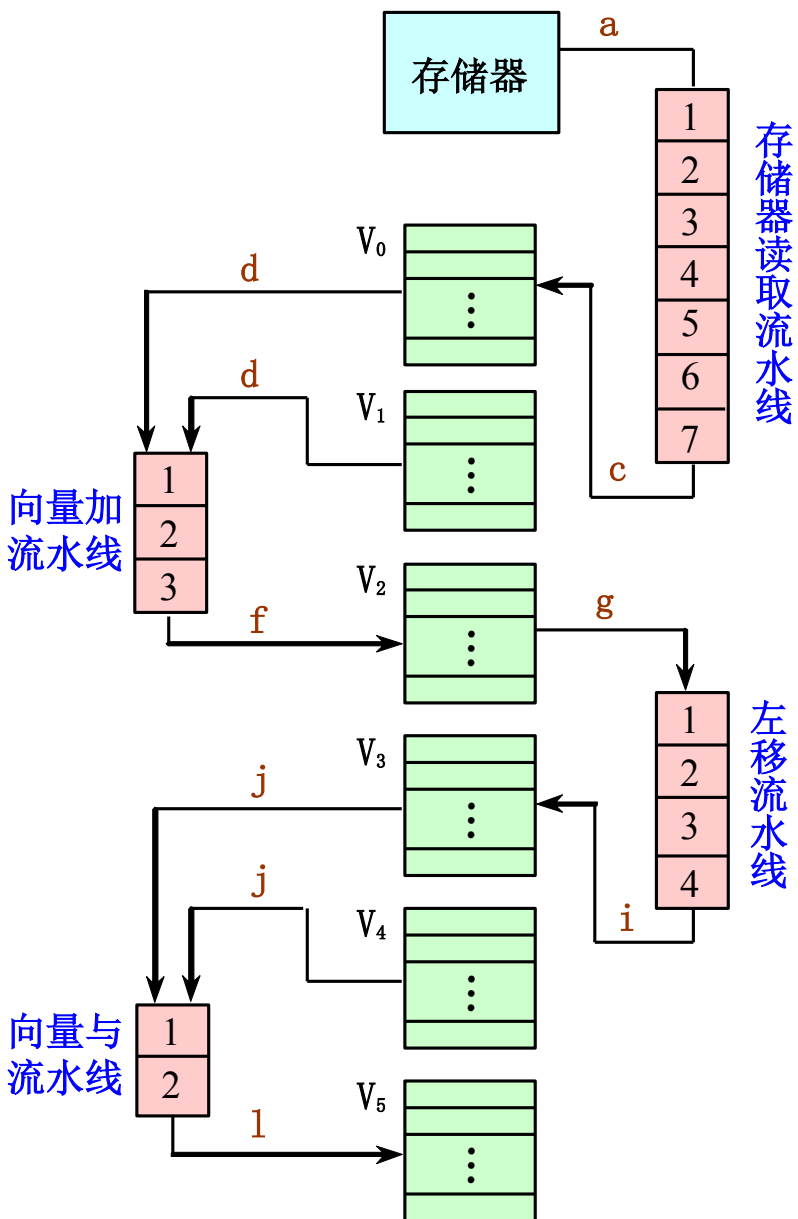
画出链接示意图，并求该链接流水线的通过时间。如果向量长度为64，则需要多少拍才能得到全部结果。

4.3 提高向量处理机性能的方法

$V_0 \leftarrow \text{存储器}$	// 访存取向量：7拍
$V_2 \leftarrow V_0 + V_1$	// 向量加：3拍
$V_3 \leftarrow V_2 \ll A_3$	// 按 (A_3) 左移：4拍
$V_5 \leftarrow V_3 \wedge V_4$	// 与操作：2拍

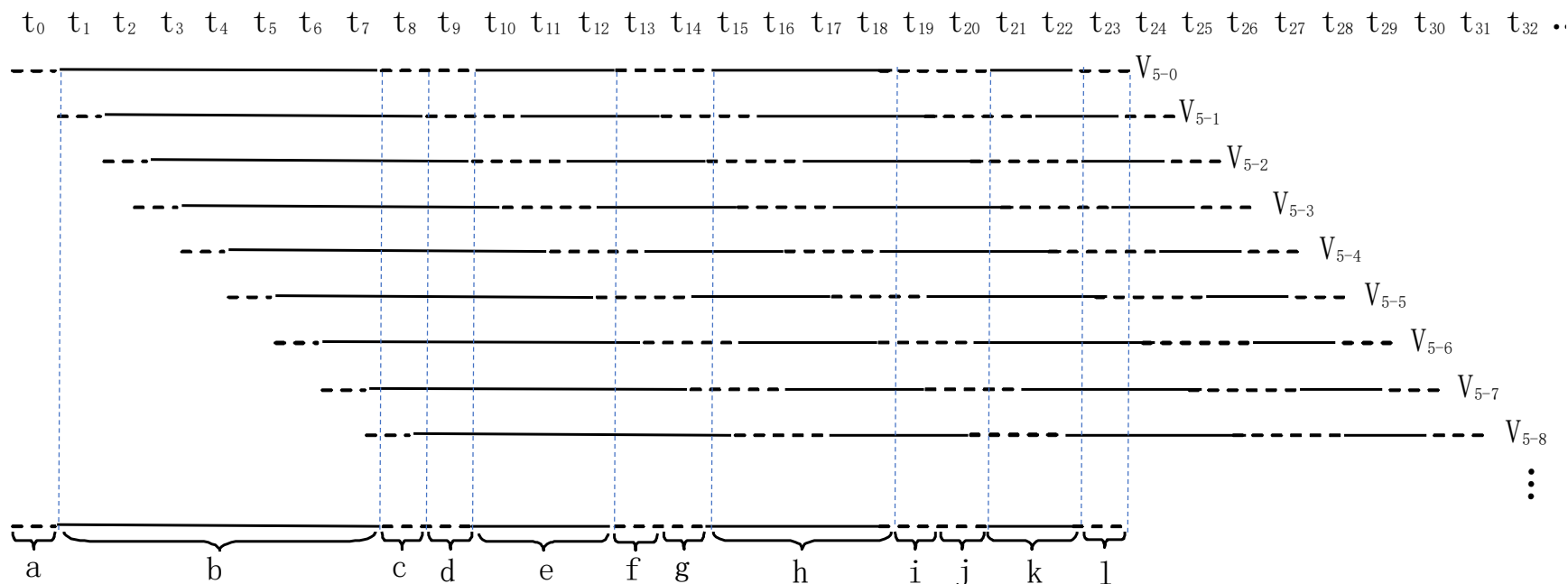
解：对这4条指令进行分析可知：它们既没有部件冲突，也没有寄存器冲突，相邻两条指令之间都存在先写后读相关，因而可以把访存流水线、向量加流水线、向量移位流水线以及向量逻辑运算流水线链接成一个较长的流水线。

Cray-1 的流水线链接举例



4.3 提高向量处理机性能的方法

链接操作的时间图：



- a:** 存储字到“读功能部件”的传送时间
- b:** 存储字经过“读功能部件”的通过时间
- c:** 存储字从“读功能部件”到 V_0 分量的传送时间
- d:** V_0 和 V_1 中操作数到整数加功能部件的传送时间
- e:** 整数加功能部件的通过时间
- f:** 和从整数加功能部件到 V_2 分量的传送时间

- g:** V_2 中的操作数分量到移位功能部件的传送时间
- h:** 移位功能部件的通过时间
- i:** 结果从移位功能部件到 V_3 分量的传送时间
- j:** V_3 和 V_4 中的操作数分量到逻辑部件的传送时间
- k:** 逻辑功能部件的通过时间
- l:** 最后结果到 V_5 分量的传送时间

4.3 提高向量处理机性能的方法

例4 在CRAY-1上用链接技术进行向量运算 $D=A \times (B+C)$

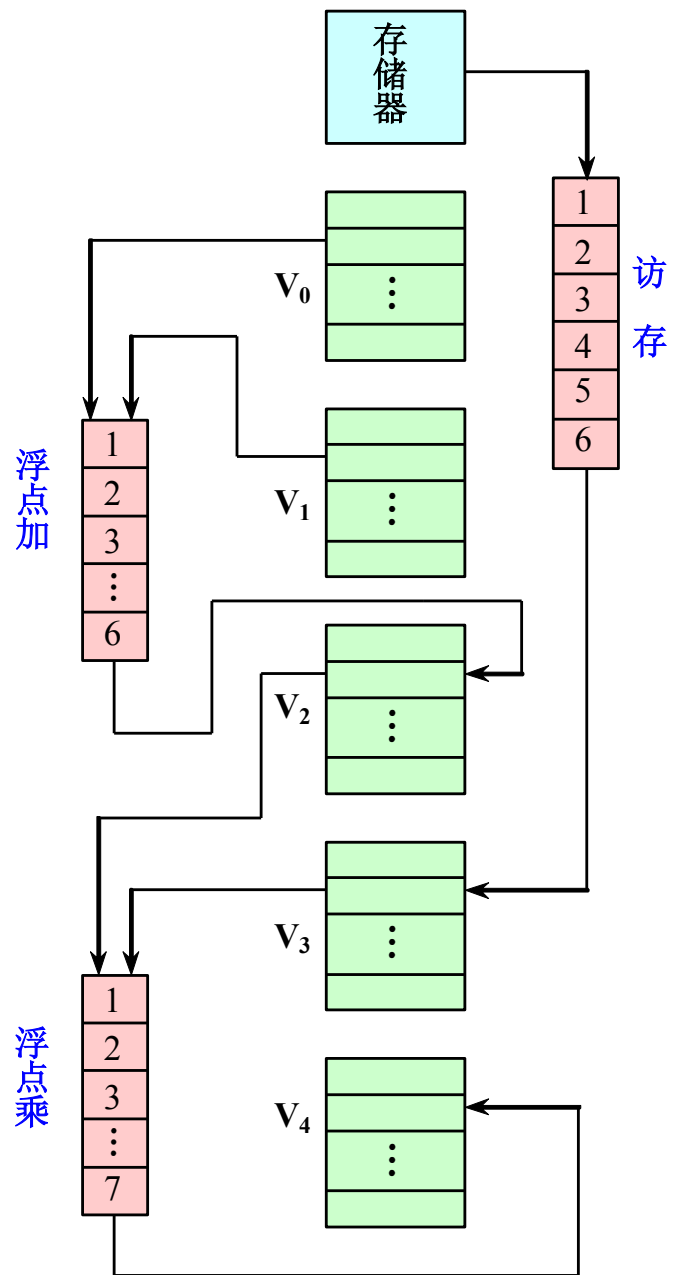
假设向量长度 $N \leq 64$ ，向量元素为浮点数，且向量 B 、 C 已存放在 V_0 和 V_1 中。画出链接示意图，并分析非链接执行和链接执行两种情况下的执行时间。

解： 用以下三条向量完成上述运算：

$V_3 \leftarrow \text{存储器}$ // 访存取向量 A

$V_2 \leftarrow V_0 + V_1$ // 向量 B 和向量 C 进行浮点加

$V_4 \leftarrow V_2 \times V_3$ // 浮点乘，结果存入 V_4



链接示意图

4.3 提高向量处理机性能的方法

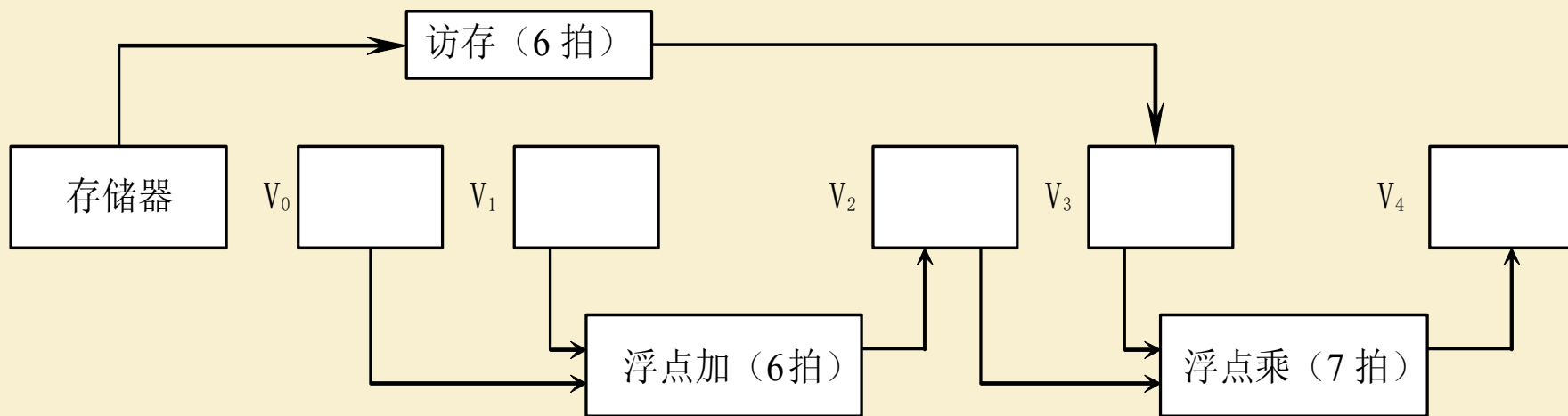


图 3.44 通过链接技术实现指令重叠执行

- 假设：**把向量数据元素送往向量功能部件以及把结果存入向量寄存器需要一拍时间，从存储器中把数据送入访存功能部件需要一拍时间。

4.3 提高向量处理机性能的方法

- 3条指令全部用串行方法执行，则执行时间为：

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

- 前两条指令并行执行，然后再串行执行第3条指令，则执行时间为：

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

- 第1、2条向量指令并行执行，并与第3条指令链接执行

4.3 提高向量处理机性能的方法

- ✓ 从访存开始到把第一个结果元素存入 V_4 所需的拍数
(亦称为**链接流水线的建立时间**) 为:

$$[(1 + 6 + 1) + (1 + 7 + 1)] = 17 \text{ (拍)}$$

- ✓ 3条指令的执行时间为:

$$[(1 + 6 + 1) + (1 + 7 + 1)] + (N - 1) = N + 16 \text{ (拍)}$$

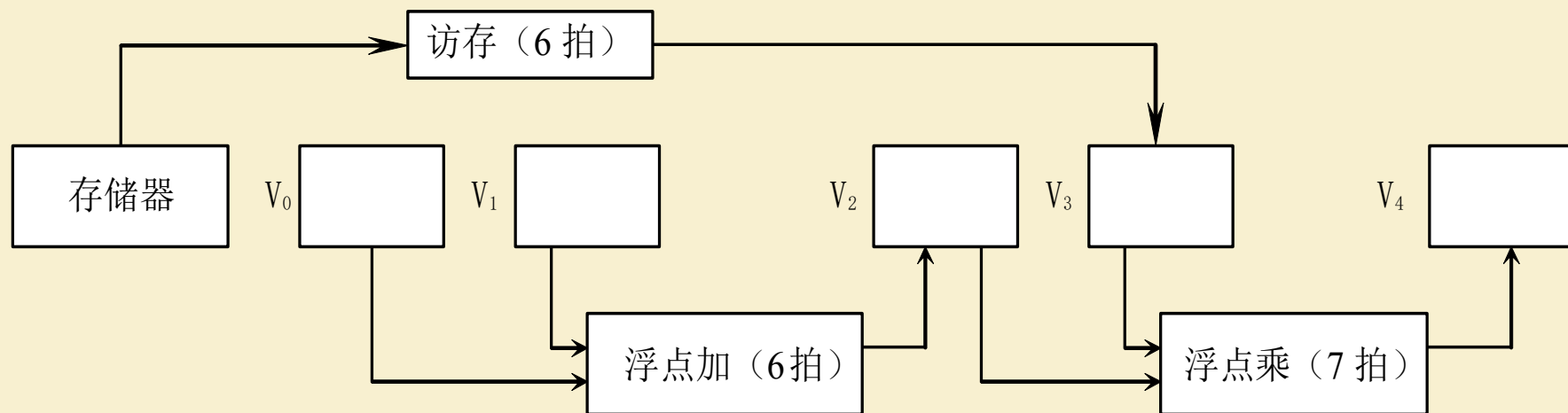


图 3.44 通过链接技术实现指令重叠执行

4.3 提高向量处理机性能的方法

● 进行向量链接的要求

保证：无向量寄存器使用冲突和无功能部件使用冲突

- 只有在前一条指令的第一个结果元素送入结果向量寄存器的那一个时钟周期才可以进行链接。
- 当一条向量指令的两个源操作数分别是两条先行指令的结果寄存器时，要求先行的两条指令产生运算结果的时间必须相等，即要求有关功能部件的通过时间相等。
- 要进行链接执行的向量指令的向量长度必须相等，否则无法进行链接。

4.3 提高向量处理机性能的方法

(3) 分段开采技术

如果向量的长度大于向量寄存器的长度，该如何处理呢？

- 当向量的长度大于向量寄存器的长度时，必须把长向量分成长度固定的段，然后循环分段处理，每一次循环只处理一个向量段。
- 这种技术称为分段开采技术。
 - 由系统硬件和软件控制完成，对程序员是透明的

4.3 提高向量处理机性能的方法

例5 设 A 和 B 是长度为 N 的向量，考虑在Cray-1向量处理器上实现以下的循环操作：

DO 10 I=1, N

10 $A(I) = 5.0 * B(I) + C$

4.3 提高向量处理机性能的方法

$$A(I) = 5.0 * B(I) + C$$

➤ 当 $N \leq 64$ 时，可以用以下指令序列：

$S_1 \leftarrow 5.0$; 将常数5.0送入标量寄存器 S_1

$S_2 \leftarrow C$; 将常数C送入标量寄存器 S_2

$VL \leftarrow N$; 在向量长度寄存器VL中设置向量长度N

$V_0 \leftarrow B$; 从存储器中将向量B读入向量寄存器 V_0

$V_1 \leftarrow S_1 \times V_0$; 向量B中的每个元素分别和常数 S_1 相乘

$V_2 \leftarrow S_2 + V_1$; 向量 V_1 中的每个元素分别和常数 S_2 相加

$A \leftarrow V_2$; 将计算结果从向量寄存器 V_2 存入存储器
; 的向量A

4.3 提高向量处理机性能的方法

➤ 当 $N > 64$ 时，就需要进行分段开采

- 循环次数 K :

$$K = \left\lfloor \frac{N}{64} \right\rfloor$$

- 余数 L :

$$L = N - 64 \times \left\lfloor \frac{N}{64} \right\rfloor$$

4.3 提高向量处理机性能的方法

$$\mathbf{A(I)} = 5.0 * \mathbf{B(I)} + \mathbf{C}$$

$S_1 \leftarrow 5.0$; 将常数5.0送入标量寄存器 S_1

$S_2 \leftarrow C$; 将常数C送入标量寄存器 S_2

$VL \leftarrow L$; 在向量长度寄存器VL中设置向量长度L

$V_0 \leftarrow B$; 从存储器中将向量 $B[0..L-1]$ 读入向量
寄存器 V_0

$V_1 \leftarrow S_1 * V_0$; 向量B中的每个元素分别和常数 S_1 相乘

$V_2 \leftarrow S_2 + V_1$; 向量 V_1 中的每个元素分别和常数 S_2 相加

$A \leftarrow V_2$; 将计算结果从向量寄存器 V_2 存入存储器
的向量 $A[0..L-1]$

处理余数部分,
计算L个元素

4.3 提高向量处理机性能的方法

For (I=0 to K-1) {

$V_0 \leftarrow B$; 从存储器中将向量 $B[L+I*64 \cdots L+I*64+63]$

; 读入向量寄存器 V_0

$V_1 \leftarrow S_1 * V_0$; 向量 B 中的每个元素分别和常数 S_1 相乘

$V_2 \leftarrow S_2 + V_1$; 向量 V_1 中的每个元素分别和常数 S_2 相加

$A \leftarrow V_2$; 将计算结果 V_2 存入存储器的向量

; $A[L+I*64 \cdots L+I*64+63]$

}

循环
K次
分段
处理

4.3 提高向量处理机性能的方法

(4) 采用多处理机系统

许多新型向量处理机系统采用了多处理机系统结构。

例如：

- CRAY-2
 - ✓ 包含4个向量处理机
 - ✓ 浮点运算速度最高可达1800MFLOPS
- CRAY Y-MP、C90
 - 最多可包含16个向量处理机

第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构

4.3 提高向量处理机性能的方法

4.4 向量处理机性能评价 →

4.5 向量处理机实例

- 1 指令处理时间
- 2 峰值性能 R_{∞}
- 3 半性能向量长度 $n_{1/2}$
- 4 向量长度临界值 n_v

4.4 向量处理机的性能评价

- 衡量向量处理机性能的主要参数:

- (1) 向量指令的处理时间
- (2) 向量处理机的峰值性能 R_{∞}
- (3) 半性能向量长度 $n_{1/2}$
- (4) 向量长度临界值 n_v

4.4 向量处理机的性能评价

(1) 向量指令的处理时间

➤ 执行一条向量长度为 n 的向量指令所需的时间为：

$$T_{vp} = T_s + T_{vf} + (n - 1)T_c$$

- **T_s ：向量流水线的建立时间：**为了使处理部件流水线能开始工作（即开始流入数据）所需要的准备时间
- **T_{vf} ：向量流水线的流过时间：**从向量指令开始译码，到第一对向量元素流过流水线直到产生第一个结果元素所需的时间
- **T_c ：流水线瓶颈段的执行时间**

4.4 向量处理机的性能评价

- 如果流水线不存在“瓶颈”，每段的执行时间等于一个时钟周期，则上式可以写为：

$$T_{vp} = [s + e + (n - 1)]T_{clk}$$

- s : 向量流水线的建立时间所对应的时钟周期数
- e : 向量流水线的流过时间所对应的时钟周期数
- T_{clk} : 时钟周期时间

- 不考虑 T_s ，并令 $T_{start}=e-1$ ： $T_{vp} = [T_{start}+n]T_{clk}$

- T_{start} : 向量功能部件启动所需的时钟周期数

从一条向量指令开始执行到还差一个时钟周期就产生第一个结果所需的时钟周期数。可称之为该向量指令的启动时间。此后，便是每个时钟周期流出一个结果，共有 n 个结果

4.4 向量处理机的性能评价

- 对于一组向量指令，其执行时间主要取决于**三个因素**：
 - 向量的长度
 - 向量操作之间是否链接
 - 向量功能部件的冲突和数据的冲突性
- 把几条能在同一个时钟周期内一起开始执行的**向量指令集合称为一个编队**。
 - 同一个编队中的向量指令之间一定不存在流水向量功能部件的冲突和数据的冲突。

4.4 向量处理机的性能评价

同一个编队中的向量指令之间一定不存在流水向量功能部件的冲突和数据的冲突

- 编队后，这个向量指令序列的总的执行时间为各编队的执行时间的和。

$$T_{all} = \sum_{i=1}^m T_{vp}^{(i)}$$

- $T_{vp}^{(i)}$ ：第*i*个编队的执行时间
- m ：编队的个数

4.4 向量处理机的性能评价

- 当一个编队是由若干条指令组成时，其执行时间就应该由该编队中各指令的执行时间的最大值来确定。

$T_{start}^{(i)}$ ：第*i*编队中各指令的启动时间的最大值

$$T_{all} = \sum_{i=1}^m T_{vp}^{(i)} = \sum_{i=1}^m (T_{start}^{(i)} + n)T_c = (\sum_{i=1}^m T_{start}^{(i)} + mn)T_c = (T_{start} + mn)T_c$$

$$T_{start} = \sum_{i=1}^m T_{start}^{(i)}$$

该组指令总的启动时间（时钟周期个数）

- 表示成时钟周期个数

$$T_{all} = T_{start} + mn \quad (\text{拍})$$

4.4 向量处理机的性能评价

例6 假设每种向量功能部件只有一个，那么下面的一组向量指令能分成几个编队？

LV V1, Rx

MULTSV V2, R0, V1

LV V3, Ry

ADDV V4, V2, V3

SV Ry, V4

解：分为4个编队

- ✓ 第一编队：LV
- ✓ 第二编队：MULTSV; LV
- ✓ 第三编队：ADDV
- ✓ 第四编队：SV

4.4 向量处理机的性能评价

- 一个编队内所有向量指令执行完毕所要的时间为：
(假设第*i*个编队中所有向量指令处理的向量元素个数均为*n*)

$$\begin{aligned} T_c^i &= \max_j [T_{start}^{ij} + n] T_{clk} \\ &= (T_{start}^i + n) T_{clk} \end{aligned}$$

- T_c^i : 第*i*个编队的执行时间
- T_{start}^{ij} : 第*i*个编队中第*j*条指令所使用向量功能部件的启动时钟周期数

4.4 向量处理机的性能评价

➤ 编队后的向量指令序列总的执行时间为：

$$\begin{aligned} T_v &= \sum_{i=1}^m T_c^i \\ &= \sum_{i=1}^m (T_{start}^i + n) T_{clk} \\ &= (\sum_{i=1}^m T_{start}^i + mn) T_{clk} \\ &= (T_{start} + mn) T_{clk} \end{aligned}$$

- m ：向量指令序列编队的个数
- T_{start} ：向量指令序列编队总的启动时钟周期数

4.4 向量处理机的性能评价

- 编队采用分段开采技术后，向量指令序列执行所需的总的时钟周期数为：

$$\begin{aligned} T_n &= \left\lfloor \frac{n}{MVL} \right\rfloor \times (T_{loop} + T_{start} + m \times MVL) \\ &\quad + [T_{loop} + T_{start} + m \times (n - \left\lfloor \frac{n}{MVL} \right\rfloor \times MVL)] \\ &= \left\lfloor \frac{n}{MVL} \right\rfloor \times (T_{loop} + T_{start}) + mn \end{aligned}$$

- T_{loop} ：分段开采所需的额外的时间开销
- MVL ：向量处理机的向量寄存器长度

4.4 向量处理机的性能评价

例7 在某向量处理机上执行DAXPY的向量指令序列，计算向量表达式：

$$Y = \alpha \times X + Y$$

其中 X 和 Y 是双精度浮点向量，最初保存在外部存储器中， α 是一个双精度浮点常数，已存放在浮点寄存器F0中。

计算该表达式的向量指令序列如下：

```
LV      V1, Rx
MULTFV  V2, F0, V1
LV      V3, Ry
ADDV    V4, V2, V3
SV      Ry, V4
```

4.4 向量处理机的性能评价

解：可以把上述5条向量指令按如下方式进行编队：

- 第一编队： **LV** V1, Rx;
- 第二编队： **MULTFV** V2, F0, V1; **LV** V3, Ry;
- 第三编队： **ADDV** V4, V2, V3;
- 第四编队： **SV** Ry, V4。

假设： $T_{loop} = 15$

向量存储部件的启动： 12个时钟周期

向量乘法部件的启动： 7个时钟周期

向量加法部件的启动： 6个时钟周期

向量寄存器长度： MVL

分别对于不采用向量链接技术和采用链接技术的两种情况，求完成上述向量操作的总执行时间。

4.4 向量处理机的性能评价

- 对n个向量元素进行计算所需的时钟周期数为

$$\begin{aligned} T_n &= \left\lceil \frac{n}{MVL} \right\rceil \times (T_{loop} + T_{start}) + mn \\ &= \left\lceil \frac{n}{64} \right\rceil \times (15 + 12 + 12 + 6 + 12) + 4n = \left\lceil \frac{n}{64} \right\rceil \times 57 + 4n \end{aligned}$$

- 采用向量链接技术，那么指令序列可以编队为

- 第一编队：LV V1,Rx; MULTFV V2,F0,V1;
- 第二编队：LV V3,Ry; ADDV V4,V2,V3;
- 第三编队：SV Ry,V4

4.4 向量处理机的性能评价

- 第一编队启动需要**12+7=19**个时钟周期
- 第二个编队启动需要**12+6=18**个时钟周期
- 第三个编队启动仍然需要**12**个时钟周期

➤ 对n个向量元素进行计算所需的时钟周期数为：

$$\begin{aligned} T_n &= \left\lceil \frac{n}{MVL} \right\rceil \times (T_{loop} + T_{start}) + mn \\ &= \left\lceil \frac{n}{64} \right\rceil \times (15 + 19 + 18 + 12) + 3n = \left\lceil \frac{n}{64} \right\rceil \times 64 + 3n \end{aligned}$$

4.4 向量处理机的性能评价

(2) 向量处理机的峰值性能 R_{∞}

- R_{∞} 表示当向量长度为无穷大时，向量处理机的最高性能，也称为峰值性能。

$$R_{\infty} = \lim_{n \rightarrow \infty} \frac{\text{向量指令序列中浮点运算次数} \times \text{时钟频率}}{\text{向量指令序列执行所需的时钟周期数}}$$

- 对于上述例题向量指令序列中的操作，只有两条浮点操作向量指令：**MULTFV V2,F0,V1** 和 **ADDV V4,V2,V3**
 - 假设该向量处理机的时钟频率为200 MHz，那么：

4.4 向量处理机的性能评价

$$R = \lim_{n \rightarrow \infty} \frac{\text{向量指令序列中浮点运算次数} \times \text{时钟频率}}{\text{向量指令序列执行所需的时钟周期数}} \quad \text{MFLOPS}$$

$$= \lim_{n \rightarrow \infty} \frac{2 \times n \times 200}{\left\lceil \frac{n}{64} \right\rceil \times 64 + 3n} \quad \text{MFLOPS}$$

$$= \lim_{n \rightarrow \infty} \frac{2 \times n \times 200}{4n} \quad \text{MFLOPS}$$

$$= 100 \quad \text{MFLOPS}$$

4.4 向量处理机的性能评价

(3) 半性能向量长度 $n_{1/2}$

- 向量处理机的运行性能达到其峰值性能的一半时所必须满足的向量长度。
- 对于上面的例子

由于该向量处理机的峰值性能 $R_{\infty} = 100 \text{ MFLOPS}$

所以根据半性能向量长度的定义有：

$$\frac{2 \times n_{1/2} \times 200}{\left\lceil \frac{n_{1/2}}{64} \right\rceil \times 64 + 3n_{1/2}} = 50$$

4.4 向量处理机的性能评价

假设 $N_{1/2} \leq 64$ ，那么有：

$$64 + 3n_{1/2} = \frac{2 \times n_{1/2} \times 200}{50} = 8n_{1/2}$$

$$5n_{1/2} = 64, \quad n_{1/2} = 12.8$$

$$n_{1/2} = 13$$

4.4 向量处理机的性能评价

(4) 向量长度临界值 n_v

- 对于某一计算任务而言，向量方式的处理速度优于标量串行方式处理速度时所需的**最小向量长度**。
- 对于上述**DAXPY**的例子
 - 假设，在标量串行工作方式下实现DAXPY循环的开销为10个时钟周期。那么在标量串行方式下，计算DAXPY循环所需要的时钟周期数为：

```
LV    V1, Rx
MULTFV V2, F0, V1
LV    V3, Ry
ADDV  V4, V2, V3
SV    Ry, V4
```

$$T_s = (10 + 12 + 7 + 12 + 6 + 12) \times n_v = 59n_v$$

4.4 向量处理机的性能评价

- 在向量方式下，计算DAXPY循环所需要的时钟周期数为：

$$T_v = 64 + 3n_v$$

- 根据向量长度临界值的定义，有：

$$T_v = T_s$$

$$64 + 3n_v = 59n_v$$

$$n_v = \left\lceil \frac{64}{56} \right\rceil = 2$$

第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构

4.3 提高向量处理机性能的方法

4.4 向量处理机性能评价

4.5 向量处理机实例

4.5 向量处理机实例

系统型号	推出时间	最大配置，时钟周期， 操作系统/编译系统	特色和要点
Cray 1S	1976年	有10条流水线的单处理机，12.5ns， COS/CF7 2.1	第一台基于ECL的超级计算机
Cray 2S/4-256	1985年	256M字存储器的4台处理机，4.1ns， COS或UNIX/CF77 3.0	16K字的本地存储器， 移植了UNIX V
Cray X-MP 416	1983年	16M字存储器的4台处理机，128M字 SSD，8.5ns，COS/CF77 5.0	使用共享寄存器组用于 IPC
Cray Y-MP 832	1988年	128M字存储器的8台处理机，6ns， CF77 5.0	X-MP的改进型
Cray Y-MP C-90	1991年	每台处理机2条向量流水线，16台处理 机，4.2ns，UNICOS/CF77 5.0	最大的Cray机器
CDC Cyber 205	1982年	有4条流水线的单处理机，20ns， 虚拟OS/FTN200	存储器-存储器系统结 构

4.5 向量处理机实例

系统型号	推出时间	最大配置, 时钟周期, 操作系统/编译系统	特色和要点
ETA 10E	1985年	单处理机, 10.5ns, ETAV/FTN 200	Cyber 205的后继型号
NEC SX-X/44	1991年	每台处理机4组流水线, 4台处理机, 2.9ns, F77SX	
Fujitsu VP2600/10	1991年	5条流水线的单处理机和双标量处理 机, 3.2ns, MSP.EX/F77 EX/VP	使用可重构微向量寄 存器和屏蔽
Hitachi 820/80	1988年	512MB存储器, 18个流水线功能部 件的单处理机, 4ns, FORT 77/HAP V23-OC	64个I/O通道, 最大传 输率为288MB/秒

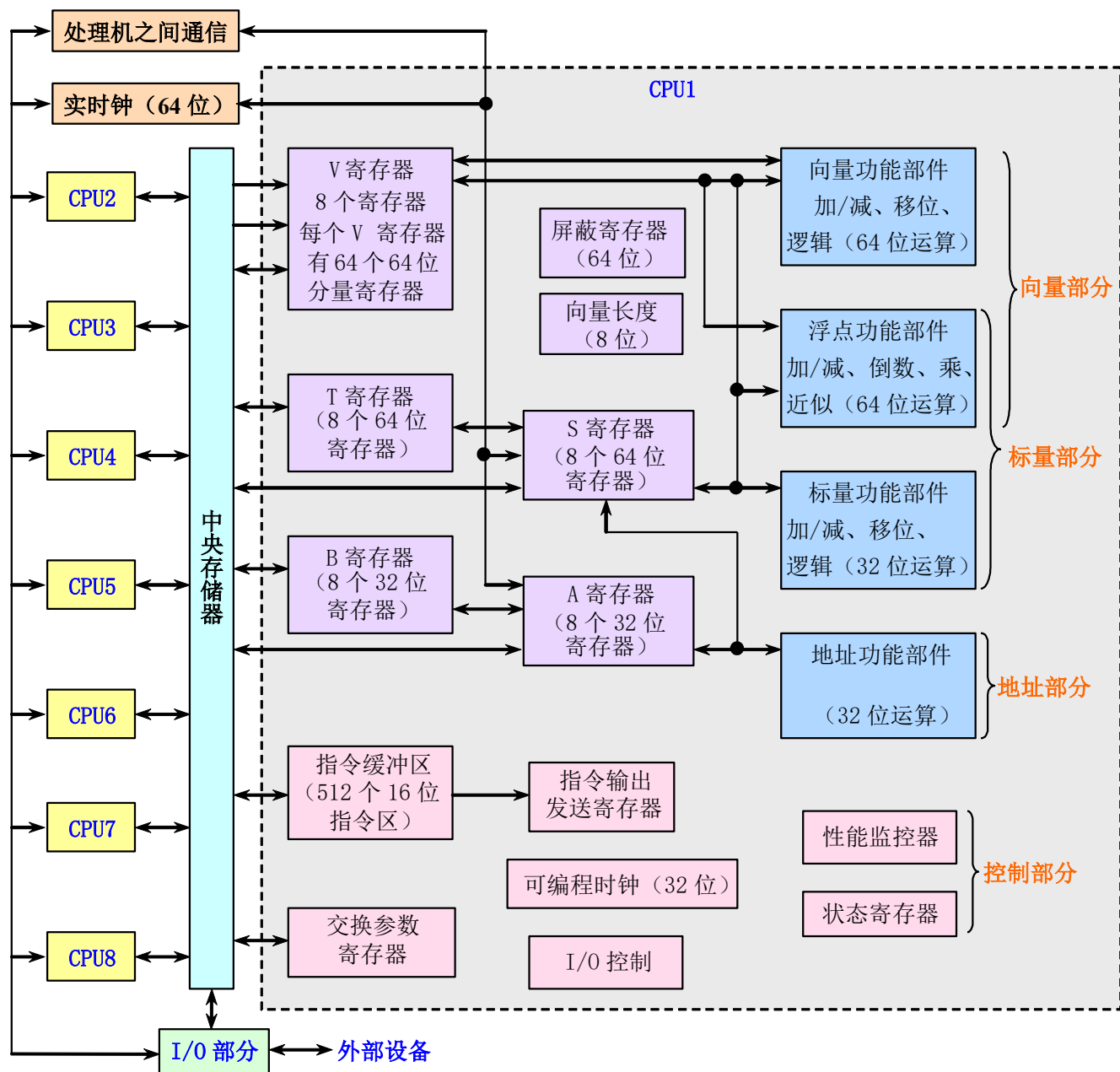
美国和日本生产的代表性的向量处理机的简要信息

4.5 向量处理机实例

Cray Y-MP和C-90

➤ Cray Y-MP 816

- ✓ 1991年问世
- ✓ 系统结构图
 - 可以配置1台、2台、4台或8台处理机
 - 8个CPU共享中央存储器、I/O 子系统、处理机通信子系统和实时时钟
 - CPU的时钟周期：6ns
- ✓ CPU的计算系统由14个功能部件组成，分为向量、标量、地址和控制4个子系统。
 - 向量和标量指令可以并行地执行
 - 所有算术运算都是“寄存器-寄存器”类型
 - 向量指令可以使用14个功能部件中的8个



4.5 向量处理机实例

✓ 中央存储器

- 分成256个交叉访问的存储体
通过每个CPU对4个存储器端口的交叉访问可以实现对存储器的重叠存取。
- 容量最大可达1GB。固态存储器的容量最大可达4GB。
- 4个存储器访问端口允许每个CPU同时执行两个标量和向量取操作、一个存储操作和一个独立的I/O操作。

这些并行的存储器访问也采用流水线方式，使得向量读和向量写可以同时进行。

4.5 向量处理机实例

- ✓ 系统使用了大量地址寄存器、标量寄存器、向量寄存器、中间寄存器和临时寄存器。
 - 通过对寄存器及多体存储器和算术/逻辑流水线的使用，可以实现功能流水线灵活的链接。
- ✓ 浮点和整数算术运算都是64位。
- ✓ 大型指令高速缓存可同时存放512条16位的指令。
- ✓ 主机中的处理机之间的通信系统包括用于快速同步目的的共享寄存器群。
 - 每个群由共享地址寄存器、共享标量寄存器和信号灯寄存器组成。
 - CPU之间向量数据通信是通过共享存储器实现的。
- ✓ I/O子系统支持3类通道，传输速率分别：6MB/s，100MB/s和1GB/s。

4.5 向量处理机实例

➤ C-90

- ✓ 由16个类似于Y-MP的CPU组成
- ✓ 16台处理机共享主存储器的容量高达256M字(2GB)
- ✓ SSD存储器的容量最多达16GB
 - 可选作第二级主存储器
- ✓ 两条向量流水线和两个功能部件可以并行操作，每个时钟周期能产生4个向量计算结果。

意味着每台处理机有4路并行性，因此16台处理机每个时钟周期最多可以产生64个向量计算结果。

4.5 向量处理机实例

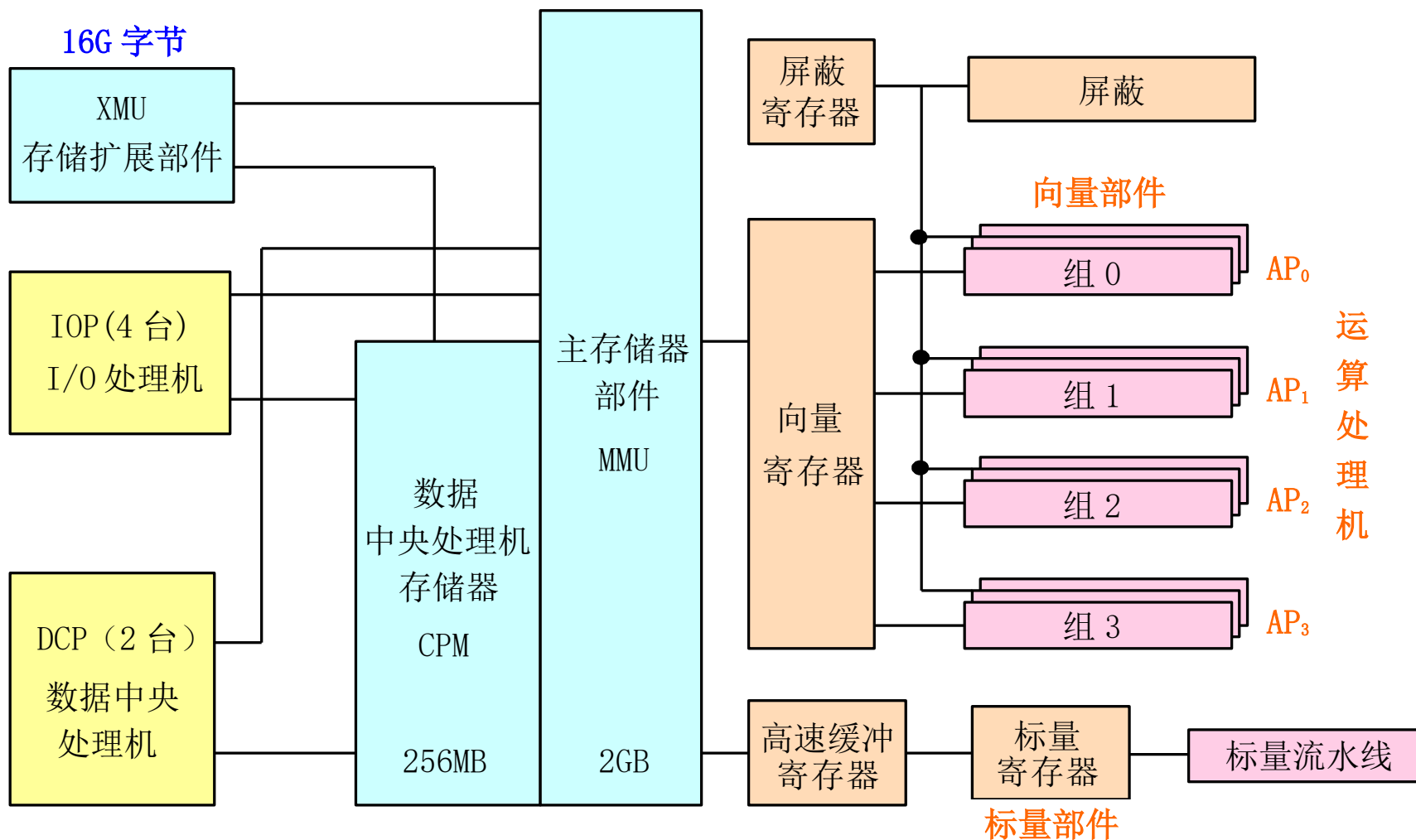
- ✓ 运行UNICON操作系统，提供向量化的FORTRAN 77和C编译器。
- ✓ 64路并行性和4.2ns时钟周期相配合，可使系统的峰值性能达到16GFLOPS，系统最大I/O吞吐率为13.6MB/s。

4.5 向量处理机实例

➤ NEC SX-X44

- ✓ NEC 1991年推出 峰值速度可达到22GFLOPS
 - 主要措施之一：使用了基于VLSI和高密度封装技术的2.9ns的时钟
- ✓ 系统结构图
- ✓ 4台运算处理机通过共享寄存器或通过2GB的共享存储器进行通信。
 - 每台处理机有4组向量流水线
每组包括2条加法/移位流水线和2条乘法/逻辑流水线
 - 类似于C-90，4台处理机可达到64路并行

4.5 向量处理机实例



4.5 向量处理机实例

✓ 高速标量部件

- 采用了具有128个标量寄存器的RISC系统结构，通过把指令重新排序来开发较高的并行性。

✓ 主存储器为1024路的交叉访问存储器

- 其扩展存储器的最大容量高达16GB
- 最大传输率：2.75GB/s

✓ 系统最多可以配置4台I/O处理机

- 每台I/O处理机的数据传输率：1GB/s
- 最多可以提供256个通道，用于高速网络、图形和外围操作，支持100MB/s的通道传输。

第4章 向量处理机

4.1 向量处理方式

4.2 向量处理机的结构

4.3 提高向量处理机性能的方法

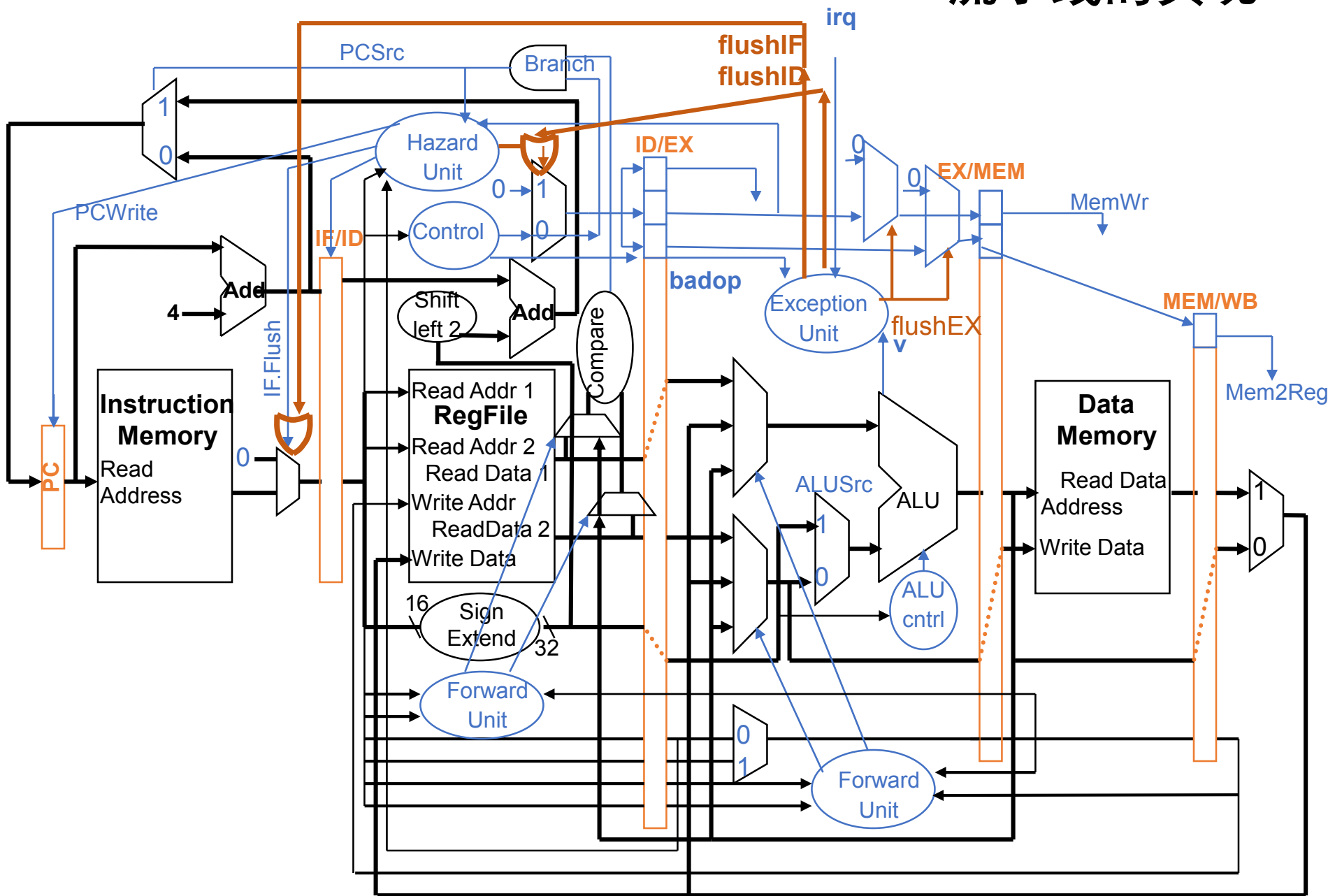
4.4 向量处理机性能评价

4.5 向量处理机实例

● 课本作业：

4.5 , 4.7

流水线的实现



计算机体系结构教学日历

	内容		作业
第一周	1 基本概念	1 课程体系及要求	1.6 CPU性能CPI
		2 定量分析	1.9 Amdahl 定律
第二周	2 指令集	3 指令格式及CISC	熟悉MIPS指令与程序
		4 RICS及MIPS	
第三周	3 流水线	5 基本概念	3.9 吞吐率和效率
		6 性能指标	3.11 静态流水时空图
第四周		7 相关与冲突	3.12 动态流水时空图
		8 MIPS流水线实现	3.16 向量处理机
第五周	4 指令级并行	9 动态调度	4.2 Tomasulo算法
		10 分支预测	4.4 分支目标缓冲CPI
第六周		11 多指令流出	4.7 指令调度
	5 存储系统	12 Cache层次体系	5.8 Cache命中率
第七周		13 Cache设计	5.10、 5.12 直接相联、组映像，
	6 输入输出	14 I/O与总线	数据传输速率 6.6， 6.8， 6.10
第八周	7 实验课	15 计算机性能评价	计算机上机测试、分析，
		16 流水线实现测试	提交实验结果和报告