国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第15卷第4期 1993年12月 Vol. 15 No. 4

离散 Hartley 变换的 MIMD 并行算法*

曾泳泓

（系统工程与数学系）

摘　要　本文把长为 $N=N_1N_2$（N_1 为奇数）的离散 Hartley 变换（DHT）分解成 N_1 个长 N_2 的 DHT 及一些附加运算，附加运算也可以变成 N_2 个长 N_1 的 DHT. 由此得到计算它的一种 MIMD 并行算法，若用 N_1 台处理机并行计算，只需 $M(N_2)+\frac{N}{2}+\frac{3}{2}N_2-2$ 个乘法步和 $A(N_2)+\frac{N}{2}+\frac{5}{2}N_2-1$ 个加法步，这里 $M(N_2)$ 和 $A(N_2)$ 分别表示计算一个长 N_2 的 DHT 所需的乘法数及加法数。并行机的有效利用率接近于 1.

关键词　离散 Hartley 变换（DHT），并行算法，快速算法

分类号　O158

1　引　言

Hartley 变换是实数域中定义的一种离散变换，它和 DFT 等离散变换一样，也可用于计算离散卷积，且计算速度比用 DFT 更快。自从 Hartley 变换提出以来，对其快速算法的研究已取得了很大进展[2~7]，其运算量比利用 DFT 计算已有很大减少。随着大规模集成电路的发展，并行处理是计算机发展的主流，因而研究 DHT 的并行算法很有必要。由于已有的快速算法一般是针对串行计算机设计的，对并行机不一定有效，应该针对并行机的特点设计新算法。到目前为止，对 DHT 的并行算法的研究还不多，而且大部分研究局限于在已有快速算法的基础上做修改或在程序级上考虑并行。本文利用“分而治之”的思想提出了长度为 $N=N_1N_2$（N_1 为奇数）的 DHT 的一种 MIMD 并行算法。

2　算法描述

设 $N=N_1N_2$，N_1 为奇数，N_2 为任意自然数，长 N 的实数序列 $x(n)$（$n=0, 1, \cdots, N-1$）的 DHT 的定义为

$$X(k)=\sum_{n=0}^{N-1}x(n)\operatorname{cas}\frac{2\pi nk}{N},\quad k=0, 1, \cdots, N-1$$

其中，$\operatorname{cas}\alpha=\cos\alpha+\sin\alpha$. 下面为了说明方便起见，允许上述定义中的 k 取任何整数（包

* 1992年4月16日收稿

括负整数)。

令 $C_u(k)=X(N_1k+u)$, $u=0, 1, \cdots, N_1-1$, k 为整数。显然,只要计算出所有$C_u(k)$ $(k=0, 1, \cdots, N_2-1)$ 便 得到了 $X(k)$, 计算 $C_u(k)$的办法是把它们转化成长 N_2 的 DHT 的计算。

$$
\begin{aligned}
C_0(k) = X(N_1k) &= \sum_{n=0}^{N-1} x(n)\operatorname{cas}\frac{2\pi nk}{N_2} \\
&= \sum_{n=0}^{N_2-1}\left(\sum_{i=0}^{N_1-1} x(iN_2+n)\operatorname{cas}\frac{2\pi(iN_2+n)k}{N_2}\right) \\
&= \sum_{n=0}^{N_2-1}\left(\sum_{i=0}^{N_1-1} x(iN_2+n)\right)\operatorname{cas}\frac{2\pi nk}{N_2}, \quad k=0, 1, \cdots, N_2-1
\end{aligned}
$$

令

$$
x_0(n) = \sum_{i=0}^{N_1-1} x(iN_2+n), \quad n=0, 1, \cdots, N_2-1
$$

则

$$
C_0(k) = \sum_{n=0}^{N_2-1} x_0(n)\operatorname{cas}\frac{2\pi nk}{N_2}, \quad k=0, 1, \cdots, N_2-1 \tag{1}
$$

这是长 N_2 的 DHT.

令 $D_u(k)=C_u(k)+C_{N_1-u}(k-1)$, 则

$$
\begin{aligned}
D_u(k) &= X(N_1k+u)+X(N_1k-u) \\
&= \sum_{n=0}^{N-1} x(n)\left[\operatorname{cas}\frac{2\pi n(N_1k+u)}{N}+\operatorname{cas}\frac{2\pi n(N_1k-u)}{N}\right] \\
&= \sum_{n=0}^{N-1}\left(2x(n)\cos\frac{2\pi nu}{N}\right)\operatorname{cas}\frac{2\pi nk}{N_2}
\end{aligned}
$$

类似于 $C_0(k)$的计算,若令

$$
X_u(n) = \sum_{i=0}^{N_1-1} 2X(iN_2+n)\cos\frac{2\pi(iN_2+n)u}{N}, \quad n=0, 1, \cdots, N_2-1,
$$

则

$$
D_u(k) = \sum_{n=0}^{N_2-1} x_u(n)\operatorname{cas}\frac{2\pi nk}{N_2}, \quad k=0, 1, \cdots, N_2-1 \tag{2}
$$

这也是长为 N_2 的 DHT.

$$
\begin{aligned}
C_u(k)-C_{N_1-u}(k-1) &= \sum_{n=0}^{N-1} x(n)\left[\operatorname{cas}\frac{2\pi n(N_1k+u)}{N}-\operatorname{cas}\frac{2\pi n(N_1k-u)}{N}\right] \\
&= \sum_{n=0}^{N-1} x(n)\left[-2\sin\frac{2\pi nu}{N}\sin\frac{2\pi nk}{N_2}+2\sin\frac{2\pi nu}{N}\cos\frac{2\pi nk}{N_2}\right] \\
&= \sum_{n=0}^{N-1}\left(2x(n)\sin\frac{2\pi nu}{N}\right)\left[\cos\frac{2\pi nk}{N_2}-\sin\frac{2\pi nk}{N_2}\right] \\
&= \sum_{n=0}^{N_2-1}\left(2\sum_{i=0}^{N_1-1} x(iN_2+n)\sin\frac{2\pi(iN_2+n)u}{N}\right)\left[\cos\frac{2\pi nk}{N_2}-\sin\frac{2\pi nk}{N_2}\right],
\end{aligned}
$$

若令

$$x_u'(n) = 2\sum_{i=0}^{N_1-1} x(iN_2+n)\sin\frac{2\pi(iN_2+n)u}{N}, \quad n=0,1,\cdots,N_2-1$$

并设

$$D_u'(k) = \sum_{n=0}^{N_2-1} x_u'(n)\operatorname{cas}\frac{2\pi nk}{N_2}, \quad k=0,1,\cdots,N_2-1 \tag{3}$$

由于

$$\operatorname{cas}\frac{2\pi n(N_2-k)}{N_2} = \cos\frac{2\pi nk}{N_2} - \sin\frac{2\pi nk}{N_2}$$

所以

$C_u(k)-C_{N_1-u}(k-1)=D_u'(N_2-k)$， $k=0, 1, \cdots, N_2$. 这里 $D_u'(N_2)=D_u'(0)$. (3) 式为长 N_2 的 DHT.

根据前面的结果有

$$\begin{aligned} C_u(k) &= (D_u(k) + D_u'(N_2-k))/2, \quad k=0,1,\cdots,N_2-1 \\ C_{N_1-u}(k-1) &= (D_u(k) - D_u'(N_2-k))/2, \quad k=1,2,\cdots,N_2 \end{aligned} \tag{4}$$

其中 $D_u(N_2)=D_u(0)$. 因此，只要计算出 $D_u(k)$和 $D_u'(k)$ ($k=0, 1, \cdots, N_2-1; u=1, 2, \cdots, \frac{N_1-1}{2}$)便可求出所有 $C_u(k)(u=1, 2, \cdots, N_1-1; k=0, 1, \cdots, N_2-1)$. $D_u(k)$和 $D_u'(k)$的计算分别采用长 N_2 的 DHT 计算，如 (2) 式和 (3) 式所示。

下面考虑 $x_0(n)$，$x_u(n)$及 $x_u'(n)$的计算

利用和差化积公式将 $\cos\frac{2\pi(iN_2+n)u}{N}$和 $\sin\frac{2\pi(iN_2+n)u}{N}$展开并代入 $x_u(n)$ 和 $x_u'(n)$ 的表达式中，代简后得到

$$\begin{aligned} x_u(n) = {} & \left[\sum_{i=0}^{N_1-1} x(iN_2+n)\cos\frac{2\pi iu}{N_1}\right] 2\cos\frac{2\pi nu}{N} - \\ & \left[\sum_{i=0}^{N_1-1} x(iN_2+n)\sin\frac{2\pi iu}{N_1}\right] 2\sin\frac{2\pi nu}{N} \\ x_u'(n) = {} & \left[\sum_{i=0}^{N_1-1} x(iN_2+n)\cos\frac{2\pi iu}{N_1}\right] 2\sin\frac{2\pi nu}{N} + \\ & \left[\sum_{i=0}^{N_1-1} x(iN_2+n)\sin\frac{2\pi iu}{N_1}\right] 2\cos\frac{2\pi nu}{N} \end{aligned}$$

令

$$\begin{aligned} \bar{C}_u(n) &= \sum_{i=0}^{N_1-1} x(iN_2+n)\cos\frac{2\pi iu}{N_1}, \quad u=0,1,\cdots,\frac{N_1-1}{2} \\ \bar{S}_u(n) &= \sum_{i=0}^{N_1-1} x(iN_2+n)\sin\frac{2\pi iu}{N_1}, \quad u=1,2,\cdots,\frac{N_1-1}{2} \\ & n=0,1,\cdots,N_2-1 \end{aligned} \tag{5}$$

则

$$
\begin{aligned}
x_0(n) &= \bar{C}_0(n) \\
x_u(n) &= \bar{C}_u(n)2\cos\frac{2\pi un}{N} - \bar{S}_u(n)2\sin\frac{2\pi un}{N} \\
x'_u(n) &= \bar{C}_u(n)2\sin\frac{2\pi un}{N} + \bar{S}_u(n)2\cos\frac{2\pi un}{N}
\end{aligned} \tag{6}
$$

$$
u = 1, 2, \cdots, \frac{N_1-1}{2}; \quad n = 0, 1, \cdots, N_2 - 1
$$

归纳前面的结果，长 N_1N_2 的 DHT 可用如下 办法计算。

step1　根据（5）式计算出所有 $\bar{C}_u(n)$ 及 $\bar{S}_u(n)$.

step2　根据(6)式计算出 $x_0(n)$，$x_u(n)$ 及 $x'_u(n)$.

step3　计算 $x_0(n)$，$x_u(n)$ 及 $x'_u(n)$ 的长 N_2 的 DHT $C_0(k)$，$D_u(k)$ 及 $D'_u(k)$.

step4　由(4)式计算出 $C_u(k)$.

实际计算时，(4) 式中的乘法因子 1/2 可合并到（6）式的计算中去，(5) 式的计算可以进一步简化。

$$
\begin{aligned}
\bar{C}_u(n) &= x(n) + \sum_{i=1}^{(N_1-1)/2} x(iN_2+n)\cos\frac{2\pi iu}{N_1} \\
&\quad + \sum_{i=1}^{(N_1-1)/2} x((N_1-i)N_2+n)\cos\frac{2\pi(N_1-i)u}{N_1} \\
&= x(n) + \sum_{i=1}^{(N_1-1)/2} [x(iN_2+n) + x(N-iN_2+n)]\cos\frac{2\pi iu}{N_1}
\end{aligned}
$$

同理

$$
\bar{S}_u(n) = \sum_{i=1}^{(N_1-1)/2} [x(iN_2+n) - x(N-iN_2+n)]\sin\frac{2\pi iu}{N_1}
$$

令

$$
\begin{aligned}
&c_o(n) = x(n),\ c_i(n) = x(iN_2+n) + x(N-iN_2+n) \\
&s_i(n) = x(iN_2+n) - x(N-iN_2+n)
\end{aligned}
$$

则

$$
\begin{aligned}
\bar{C}_n(n) &= \sum_{i=0}^{(N_1-1)/2} c_i(n)\cos\frac{2\pi iu}{N_1}, \quad u = 0, 1, \cdots, \frac{N_1-1}{2} \\
\bar{S}_n(n) &= \sum_{i=1}^{(N_1-1)/2} s_i(n)\sin\frac{2\pi iu}{N_1}, \quad u = 1, 2, \cdots, \frac{N_1-1}{2}
\end{aligned} \tag{7}
$$

所以，算法中 step1 可用两步完成：

step1′　计算 $c_o(n)=x_o(n)$ ，$c_i(n)=x(iN_2+n)+x(N-iN_2+n)$，
　　$s_i(n)=x(iN_2+n)-x(N-iN_2+n)$，$n=0, 1, \cdots, N_2-1$.

step1″　根据(7)式计算 $\bar{C}_u(n)$ 及 $\bar{S}_u(n)$.

若有 N_1 台处理机，step1′的计算可分配每台处理机计算一个 $c_i(n)$ 或 $s_i(n)$ $(n=0, 1, \cdots, N_2-1)$，故只需 N_2 个加法步；step1″让每台处理机负责一个 $\bar{C}_n(n)$ 或 $\bar{S}_n(n)$ $(n=0, 1, \cdots, N_2-1)$ 的计算，需要 $\frac{N_1-1}{2}N_2$ 个乘法步及 $\frac{N_1-1}{2}N_2$ 个加法步；step2 中让每台处理机

负责计算一个 $x_n(n)$ 或 $x'_n(n)$ （$n=0, 1, \cdots, N_2-1$），共需 $2(N_2-1)$ 个乘法步及 N_2-1 个加法步；step3 中分配每台处理机计算一个长 N_2 的 DHT，需 $M(N_2)$ 个乘法步及 $A(N_2)$ 个加法步，这里 $M(N_2)$ 和 $A(N_2)$ 分别表示计算长 N_2 的 DHT 所需的乘法数和加法数；step4 可让每台处理机负责一个 $C_u(k)$ 的计算，共需 N_2 个加法步。所以，用 N_1 台处理机并行计算的总的并行步数为

乘法步 $M(N_2)+\frac{1}{2}N+\frac{3}{2}N_2-2$

加步步 $A(N_2)+\frac{1}{2}N+\frac{5}{2}N_2-1$

在算法中，每台处理机分配的任务量基本相等且没有重复计算，故并行机的有效利用率[8]接近于 1.

对长 N_2 的 DHT 的计算，可采用已有的快速算法[2~8]（串行算法）。若 N_2 比较小，也可以直接计算。

3 算法的进一步分析

上述方法把长 N_1N_2 的 DHT 转化成 N_1 个长 N_2 的 DHT 及一些附加运算，附加运算最复杂的部分是计算（5）式，若 N_1 比较小，用前面修正的办法直接计算即可；若 N_1 比较大，直接计算运算量较大。下面我们可看出，它也可以转化成长 N_1 的 DHT.

令

$$W_u(n) = \sum_{i=0}^{N_1-1} x(iN_2+n)\operatorname{cas}\frac{2\pi iu}{N_1}, \quad u=0, 1, \cdots, N_1-1 \tag{8}$$

对固定的 n，（8）式为长 N_1 的 DHT，根据 $\overline{C}_u(n)$ 及 $\overline{S}_u(n)$ 的定义不难得到

$$\begin{cases} \overline{C}_u(n) = (W_u(n) + W_{N_1-u}(n))/2, & u=0, 1, \cdots, \dfrac{N_1-1}{2}, \\ \overline{S}_u(n) = (W_u(n) - W_{N_1-u}(n))/2, & u=1, 2, \cdots, \dfrac{N_1-1}{2} \end{cases} \tag{9}$$

$$n = 0, 1, \cdots, N_1-1$$

故由 $W_u(n)$ 很容易得到 $\overline{C}_u(n)$ 和 $\overline{S}_u(n)$.

因此，长 N_1N_2 的 DHT 可转化为 N_1 个长 N_2 的 DHT 及 N_2 个长 N_1 的 DHT 以及一些简单的附加运算（计算（4），（6），（9）式）。实际计算时，（9）中和（4）中的乘法因子 1/2 都可合并到（6）式的计算中去，即使是串行计算，（4），（6）及（9）式也只需 $2N-2N_2$ 个乘法及 $3N-N_1-N_2$ 个加法。所以，对串行计算，运算量为

$$M(N) = N_1M(N_2) + N_2M(N_1) + 2N - 2N_2,$$
$$A(N) = N_1A(N_2) + N_2A(N_1) + 3N - N_1 - N_2$$

对 N_1 点及 N_2 点的 DHT 若采用前面的技巧继续分解，可得到计算长 N 为奇复合数的 DHT 的递归快速算法，由于分解后的 N_1 个长 N_2 的 DHT 及 N_2 个长 N_1 的 DHT 都是彼此独立的，所以适合于多处理机并行处理。

在实际计算中，很多情况下长度 N 取为 p^l，这时 N 分解为 $N=p^{l_1}p^{l_2}$，$l_1+l_2=l$. 故长 p^l 的 DHT 可用 p^{l_2} 个长 p^{l_1} 的 DHT 及 p^{l_1} 个长 p^{l_2} 的 DHT 及一些简单的附加运算代替，

在多机情况下，p^{l_2}个长 p^{l_1}的 DHT 可同时计算，p^{l_1}个长 p^{l_2}的 DHT 也可同时计算，而长 p^{l_1}或 p^{l_2}的 DHT 可采用已知的快速算法计算。

参 考 文 献

1 Bracewell R N. Assessing the Hartley transform. IEEE Trans. ASSP. 1990, 38(12)

2 曾泳泓．任意长度离散 Hartley 变换的快速算法．电子科学学刊．1993，15(2)：121～127

3 茅一民．长度为 p^l 的离散哈脱莱变换的快速算法．电子科学学刊．1990，(6)

4 王中德．快速 W 变换——算法与程序．中国科学（A 辑）．1988；(5)：549—560

5 Pei S C, Wu J L. Electronics Letters. 1986, 22(1)

6 Bracewell R N. Electronics Letters. 1987, 23(10)

7 Sorensen H V et al. IEEE Trans. ASSP. 1985, 33(10)

8 奪贤福等．同步并行算法．国防科技大学出版社，1986

A MIMD Parallel Algorithm for Discrete Hartley Transform

Zeng Yonghong

(Department of System Engineering and Mathamatics)

Abstract

This paper turns a discrete Hartley transform (DHT) of length $N=N_1N_2$ (N_1 is odd) into N_1 DHT's of length N_2 and some additional operations. The additional operations can also be turned to N_2 DHT's of length N_1. Therefore, a MIMD parallel algorithm is obtained. If N_1 processors are available, $M(N_2)+\frac{1}{2}N+\frac{3}{2}N_2-2$ multiplication steps and $A(N_2)+\frac{1}{2}N+\frac{5}{2}N_2-1$ addition steps are enough for computing a DHT of length $N=N_1N_2$, where $M(N_2)$ and $A(N_2)$ represent the number of multiplications and additions for a DHT of length N_2 respectively. The efficiency of the parallel algorithm with respect to itself is approximately 1.

Key words discrete Hartley transform (DHT), parallel algorithm, fast algorithm