시스토릭 어레이프로세서상에서 離散 푸리에變換의 具現에 關한 硏究

金 斗 京*

A Study on the Implementation of the Discrete Fourier Transform on Systolic Array Processors

Kim, Doo-gyung *

Summary

In the VLSI technology, the research on special processors for high speed processing is demanded. The availability of low cost, high density, fast processing memory devices has opened a new research area in high speed computing systems.

The discrete fourier transform algorithm is written as a matrix-based algorithm and mapped on a 2-dimensional systolic array processor. The significance of this method is that the total time required to complete an N-point DFT is $3\sqrt{N}+N/2-1$ time units.

The architecture features local inter-connections, systemclocks and modulality. The output of DFT computations are pipelined out directly in the correct order on the 2-dimensional systolic array processors.

In the paper, the total time units concepts on the implementation of those processors are described.

序 論

高性能 信號處理와 科學演算에 있어서 計算處理能力의 向上을 要求하고 있다. 그러한 점에서 저가격 高集積 高速處理의 값싼 記憶裝置들의 利用은 高性能 竝列處理機들의 設計에 重要한 影響을 주고 있다(Kung, 1978).

이러한 高性能 컴퓨터는 制御裝置들의 複雜한 設計와 機械資源의 割當에 대한 最適計劃을 必要로 한다. 시스템의 設計는 竝列計算 알고리즘과 最適處理 하드웨어, 소프트웨어 構造들에 관련된 폭 넓은 知識을 要求한다(Kung, 1984).

計算處理 速度의 重要한 改善은 프로세서 要素들의 竝列處理 速度에 기인한다. 프로세서의 모듈럴리티, 시스템클럭, 相互連結은 어레이프로세서 시스템에 重要한 要素이다. 시스토릭이나 웨이브프론트어레이 같은 국부적 相互連結網은 그들의 尨大한 竝列處理와 正規的인 데이터흐름의 信號處理 알고리즘을 具現하는 데 適合하다(winograd,

* 電子計算所

1978).

따라서 單純하고 正規的인 데이터흐름을 가진 DFT의 演算을 위해서 본 硏究에서는 2차원 시스토릭 어레이프로세서를 具現하는 데 目的이 있다. 특히 DFT의 對稱性을 利用하여 演算處理 時間을 줄이고 적은 個數의 프로세서 要素들로서 프로세서를 具現하고자 한다.

硏 究 方 法

1. 1차원 시스토릭 어레이프로세서

2차원 DFT 프로세서를 具現하기 전에 Fig.1에서 보여주는 1차원 프로세서에 DFT 具現에 關하여 論議한다.

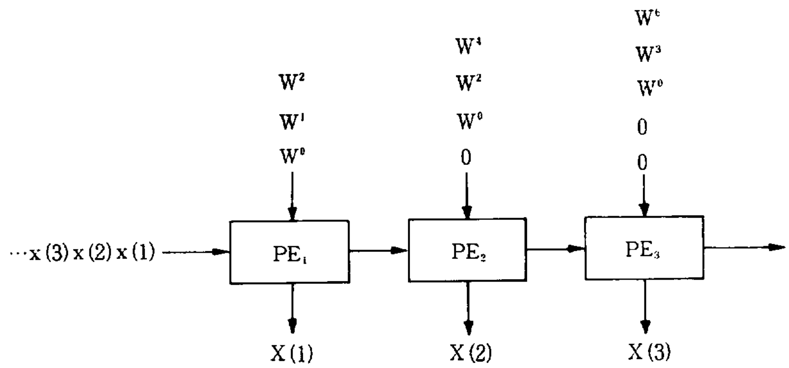

Fig. 1. 1-dimensional DFT systolic processor.

入力 데이터열은 加重因子들 W^k가 適當한 PE들에 로드되는 동안에 좌에서 우로 PE들에 파이프라인된다. 加重因子 W^k는 時變換이며 DFT의 出力 X(k)는 노드에 머물어 있으며 궁극적으로 全域連結網 위에서 나온다. PE들의 수는 DFT의 對稱性에 의해서 N/2을 必要로 한다. 단 N는 점들의 수이다.

또한 DFT의 對稱性을 利用하면 N점 DFT 演算을 遂行하는 데 要求되는 時間은 N+N/2 單位 時間이다. 單位時間은 프로세서요소(PE)에서 데이터 操作하는 데 必要한 時間이라고 假定한다. 1차원 DFT 어레이프로세서는 너무 느리다. 상당히 큰 N점 DFT 演算은 固定된 크기의 2차원 시스토릭 어레이프로세서를 使用하는 것이 바람직하다.

2. 매트릭스 演算에서 離散 푸리에 變換 알고리즘

DFT 알고리즘은 다음과 같이 定義된다.

$$X(k)=\sum_{n=1}^{N} W_N^{(k-1)(n-1)}\, x(n) \quad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

X(k) : DFT의 k번째 점 k=1,2,……, N

x(n) : 신호열 x(n)의 n번째 점

W_N : $e^{-j(2\pi/N)}$

信號列 x(n)은 正方 매트릭스로 在配列될 수 있다. 예를들면 N=16일 때 信號列 x(n)은 매트릭스 X에 의해서 表現된다.

$$X=[x_{ij}]=\begin{bmatrix} x_{11}\cdots\cdots\cdots\cdots x_{14} \\ x_{21}\cdots\cdots\cdots\cdots x_{24} \\ x_{31}\cdots\cdots\cdots\cdots x_{34} \\ x_{41}\cdots\cdots\cdots\cdots x_{44} \end{bmatrix}$$

$$=\begin{bmatrix} x(1) & x(2) & x(3) & x(4) \\ x(5) & \cdots & \cdots & x(8) \\ x(9) & \cdots & \cdots & x(12) \\ x(13) & \cdots & \cdots & x(16) \end{bmatrix} \cdots\cdots (2)$$

새로운 지수 $n=4(i-1)+j$ ························ (3)

(1), (2), (3) 식을 結合하면

$$X(k)=\sum_{i=1}^{4}\sum_{j=1}^{4}W_N^{(k-1)((4(i-1)+j)}\ x_{ij}$$

$$=\sum_{j=1}^{4}a_{ki}x_{ij}b_{jk} \quad \cdots\cdots\cdots\cdots (4)$$

$$a_{ki}=W_N^{4(k-1)(i-1)}$$

$$b_{jk}=W_N^{4(j-1)(k-1)}$$

(4) 식을 매트릭스-매트릭스-매트릭스 곱셈 方程式과 比較하면

$D=AXB$ ································ (5)

$A=[a_{ki}]_{16x4}$

$X=[x_{ij}]_{4x4}$

$B=[b_{jk}]_{4x16}$

$X(k)=d_{kk}$ d_{kk} : 매트릭스 D의 對角線 項

信號들의 DFT는 매트릭스-매트릭스-매트릭스 곱셈 結果들의 對角線 項들로서 表現할 수 있다. 따라서 매트릭스 D의 對角線에 없는 項들은 計算할 必要가 없다.

3. 시스토릭 어레이들의 프로세서 要素

프로세서 要素의 내적 계산을 위한 方法은 Fig. 2에 表示된다. Fig.2에서 $c_o \leftarrow c_i+a_ib_i$로 表示되며 데이터 b_i經路 위에 있는 작은 分割은 데이터 b_i가 프로세서에 維持되거나 우측으로 通過하는 것을 意味한다.

이 프로세서 요소들의 重要性은 데이터 b_i의 흐름 制御가 融通性 있게 하기 위하여 같은 位置에서 反復 使用에 대해 維持될 수 있다는 것이다. 2차원 DFT 시스토릭 어레이프로세서를 設計하는데 이러한 利點을 利用한다.

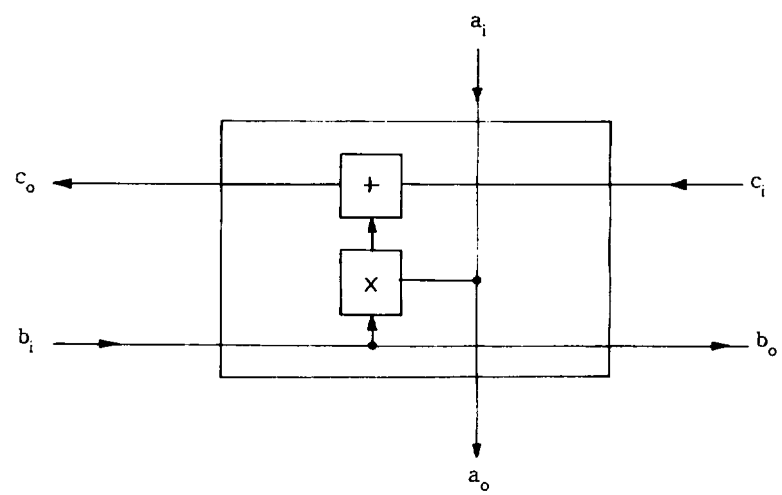

Fig. 2. Processor element for inner product.

4. DFT의 演算處理 方法

데이터 信號들의 수 $N=16=n^2$, $n=4$이다. 單純하게 하기 위하여 하나의 PE에서 데이터를 操作하는 데 單位時間이 걸리고 인접 PE들에 데이터를 通過시킨다고 假定하면 Fig.3에서 매트릭스 C는 매트릭스 A와 X의 적을 나타낸다. 매트릭스 A는 상측에 시스토릭적으로 PE들에 들어가며 반면에 매트릭스 X는 좌측에서 시토릭적으로 PE들에 들어간다.

x_{1j}, $j=1,2,3,4$는 최우측 PE에 이른다. 行의 데이터 x의 시스토릭 動作은 일시적으로 停止한다. 매트릭스 B는 더 以上 移動없이 좌측에서 최우측

열로 파이프라인된다. 매트릭스 D(D=CB, C=AX)의 對角線項들은 최좌측 PE에서 計算된다. k번째 점 DFT X(k)는 최좌측 PE들의 끝에서 파이프라인되어 나온다. 점선들은 데이터 a_{ij}의 순환을 意味한다.

매트릭스 A_i, i=1,2,3,4는 同一하므로 순환은 x_{ij}가 PE들에 있는 동안은 外部 데이터 버스들을 使用하는 것이 可能하다. 總 演算時間은 N점 DFT를 完成하기 위하여 3n+N-1 單位時間을 要求한다. $n=\sqrt{N}$이다.

DFT의 對稱 性質을 適用하면 3n+N/2-1 單位時間으로 줄일 수 있다.

3n 單位時間은 DFT 演算에서 초기화의 부담요소이다. N가 클 때 $\sqrt{N}$은 N과 비교하면 작다. 그러므로 N점 DFT를 完成하는 데 必要한 總 演算時間은 O(N/2)이다. Fig.3.에서 보여주는 DFT 配列은 4×5 DFT 어레이프로세서이다. N가 클 때 2차원 시스토릭 어레이프로세서를 使用하는 DFT 演算은 두가지 方法이 있다. 하나는 큰 2차원 시스토릭 어레이를 PE들의 총 수가 $N=\sqrt{N}$되는(N는 정수의 제곱) 프로세서를 使用하는 것이고 다른 하나는 PE들의 수를 增加함이 없이 DFT 演算의 많은 점들에 대해서 固定된 크기의 構造를 反復 使用하는 것이다.

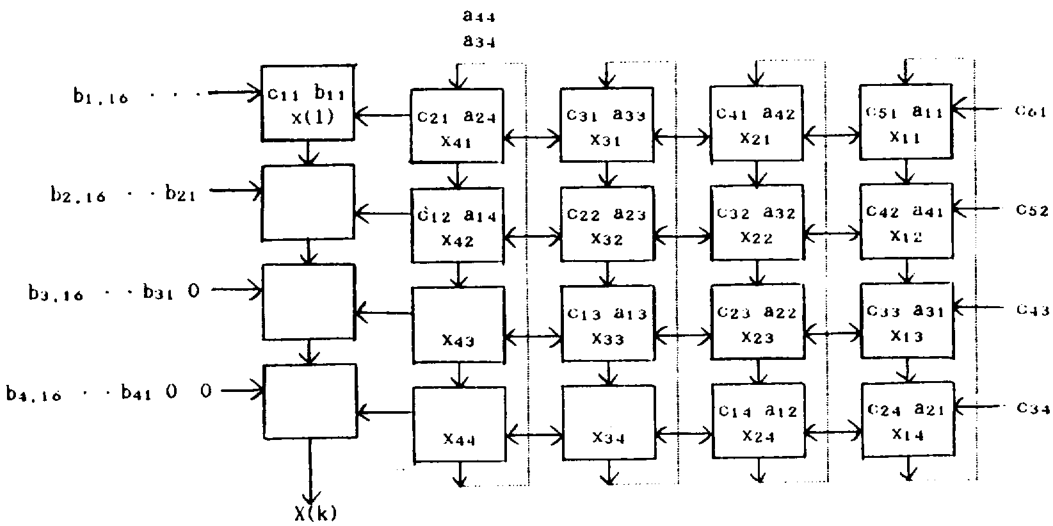

Fig. 3. 4×5 2-dimensional systolic array pro- cessor.

5. 큰 N에 대한 固定된 크기의 DFT 프로세서의 反復 使用

2차원 DFT 시스토릭 어레이프로세서의 使用을 극대화 시키기 위하여 큰 N점 演算에 대해서 固定된 크기의 DFT 프로세서의 使用을 反復하는 것이다. 이 方法은 매트릭스 分割에 基礎를 두고 있다. N=64점 DFT演算을 하기 위하여 4×5 DFT 프로세서를 使用한다. 2-2절에서 使用한 方法을 利用하면 信號列 x(n)는 매트릭스 X로서 재정돈되다.

$$X=[x_{ij}]=\begin{bmatrix} x(1) & \cdots\cdots & x(8) \\ x(9) & \cdots\cdots & x(16) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ x(57) & \cdots\cdots & x(64) \end{bmatrix}$$

새로운 지수 n=8(i-1)+j이다.

매트릭스에 基礎를 둔 DFT 알고리즘을 方程式

(5)식의 形態로 다시 쓰면

$A = [a_{ki}]_{64\times 4} \qquad a_{ki} = W_N^{8(k-1)(i-1)}$

$X = [x_{ij}]_{4\times 16}$

$B = [b_{jk}]_{16\times 64} \qquad b_{jk} = W_N^{(j-1)(k-1)}$

$W_N = e^{-j(2\pi/64)}$

$X(k) = d_{kk}, \quad k = 1,\ 2,\ \cdots\cdots\cdots,\ 64$

알고리즘을 4×5 DFT 시스토릭 프로세서에 맞추기 위해서 매트릭스 A, X, B를 4×4 매트릭스로 分割할 必要가 있다.

$$A = \begin{bmatrix} A_1 \\ \cdot \\ \cdot \\ A_{16} \end{bmatrix}$$

$A_1 = A_2 = \cdots\cdots\cdots = A_{16}$이며 4×4 매트릭스들이다.

$X = [x_{11};\ x_{12};\ x_{13};\ x_{14}]$

$x_{11},\ x_{12},\ x_{13},\ x_{14}$는 4×4 매트릭스들이다.

$$B = \begin{bmatrix} B_{11}; & \cdots\cdots\cdots; & B_{1,16} \\ B_{21}; & \cdots\cdots\cdots; & B_{2,16} \\ B_{31}; & \cdots\cdots\cdots; & B_{3,16} \\ B_{41}; & \cdots\cdots\cdots; & B_{4,16} \end{bmatrix}$$

B_{ij}는 4×4 매트릭스들이며 i=1, 2, 3, 4이고 j=1, 2, ⋯⋯⋯, 16이다. 信號列 x(n)의 DFT는 다음과 같이 計算될 수 있다.

$$\begin{bmatrix} x(1) \\ \cdot \\ \cdot \\ \cdot \\ x(64) \end{bmatrix} = \begin{bmatrix} \mathrm{diag}[A_1X_{11}B_{11} & + \cdots\cdots + A_1X_{14}B_{41}] \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \mathrm{diag}[A_1X_{11}B_{1,16} & + \cdots\cdots + A_1X_{14}B_{4,16}] \end{bmatrix}$$

여기서 x_{11}는 信號列의 첫번 째 集合이며 4×5 DFT 매트릭스 어레이프로세서에 파이프라인된다. 그리고 $A_1X_{11}B_{1j}$(j=1,2,⋯⋯,16)의 對角線 項들의 演算이 끝날때가지 PE들에 維持된다. 信號列 두번째 集合 X_{12}는 4×5DFT 프로세서에 파이프라인되며 $A_1\times_{12}B_{2j}$(j=1,2,⋯,16)의 對角線 項들을 計算하기 위하여 PE들에 維持된다. 마찬가지로 X_{13}은 $A_1X_{13}B_{3j}$와 X_{14}는 $A_1X_{14}B_{4j}$에 聯關되어 計算된다.

최종 덧셈 動作은 4×5 DFT 프로세서로부터 첫번째 集合 [diag[$A_1X_{11}B_{1j}$], j=1, 2, ⋯⋯⋯, 16]의 計算結果를 저장하고 X(k) (k=1, 2, ⋯⋯⋯, 64)을 구하기 위하여 그것들을 두번 째 集合 [diag[$A_1X_{12}B_{2j}$]], 세번 째 集合 [diag[$A_1X_{13}B_{3j}$]], 네번 째 集合 [diag[$A_1X_{14}B_{4j}$]]들과 시스토릭적으로 計算하는 코프로세서를 使用하므로서 遂行된다.

結果 및 考察

m(m+1) DFT 2차원 프로세서를 反復 使用하여 DFT점들의 첫번 째 반을 計算하는 데 必要한 總 演算時間에 대한 一般 公式은 다음과 같다.

總 演算時間 = (3m−1) + (2m−1) (c−1) + N/2−1

N : 信號列에서 全體 점들의 수

c : N/m^2

N=64에서 4×5 DFT 프로세서를 使用하므로 m=4이다.

總 演算時間 = (3×4−1) + (2×4−1) (64/16−1) + 64/2−1 = 65

DFT의 對稱性을 利用하지 않으면 97單位時間이므로 對稱性을 利用하면 32單位時間을 줄일 수 있다. 만일 $\sqrt{N}(\sqrt{N}+1)$ DFT 프로세서를 使用하면 總 演算時間 = $3\sqrt{N}+N/2-1$이다. N=64일 때 PE의 갯수는 $\sqrt{64}\times(\sqrt{64}+1)=72$개이며 總 演算時間은 $3\sqrt{64}+64/2-1=55$이다.

그런데 4×5 DFT 프로세서에서는 PE의 個數가 20개이며 總 演算時間은 65單位時間이다. 處理速度는 좀 느리지만 集積度를 考慮한 費用면을 생각하면 4×5 DFT 프로세서를 使用하는 것이 經濟的이다. 만일 N=1024이면 $\sqrt{N}(\sqrt{N}+1)$ 프로세서에서는 PE의 個數는 1056이며 總 演算時間은 607 單位時間이다. 4×5 DFT 프로세서에서는 PE의

個數는 20개이며 總 演算時間은 964 單位時間인 것을 알 수 있다.

結 論

컴퓨터 處理에서 費用과 處理 速度는 重要한 意味를 갖는다. 處理 速度만을 強調한다면 $\sqrt{N}(\sqrt{N}+1)$ 프로세서를 使用하는 것이 좋지만 費用과 處理速度를 考慮할 때 集積할 素子들의 수를 最小로 하고 適切한 處理 速度를 갖는 4×5 DFT 시스토릭 프로세서를 使用하는 것이 效率的이다.

따라서 상당히 큰 N점 DFT演算에서 固定된 크기의 2차원 시스토릭 프로세서를 使用하여 反復演算하는 方法이 重要하다. 그리고 2차원 DFT 시스토릭 프로세서는 매트릭스－매트릭스 곱셈이나 매트릭스－매트릭스－매트릭스 곱셈의 對角線 項들을 計算하는 데 有用하게 使用할 수 있다.

參 考 文 獻

Agarwal R.C., and Cooley J.W. 1977, New Algorithm for Digital Convolution, ASSP-25 IEEE, 392-410.

Cooley J. W., and Turkey J.W. 1965, An Algorithm for the Machine Calculation of Complex fourier Transform, Math. Com-uter, 297-301.

Kung H.T., and Leiserson C.E. 1978, Systolic Array for VLSI, SIAM., 256-282.

Kung S.Y., July 1984, On Supercomputering with Systolic/ Wavefront Array Processors, Proc., IEEE, 867-884.

Kung S.Y., July 1985, VLSI Array Processors, ASSP Mag., IEEE, 4-22.

임제탁, 이두수, 1981, 디지탈 信號處理의 基礎, 탑출판사.

Oppenheim A. V., and Schafer R.W. 1980, Digital Signal Processing, Prentice-Hall Co.

Winograd S. 1978, On Computing the Discrete Fourier Trasforms, Math., Computer, 175-199.

Yeh H.G. 1985, Processor Elements and Systolic Arrays, Conference Record of the 19th Asiloar Con. on Circuit, Systems and Computer.