超高速パイプライン FFT プロセッサの構成[†]

亀山 充隆*・樋口 龍雄*・今野 淳一**・高須 賀馨**

Design and Implementation of a Ultra-High-Speed Pipelined FFT Processor

Michitaka KAMEYAMA*, Tatsuo HIGUCHI*, Junichi KONNO**
and Kaoru TAKASUKA**

High-speed, low-cost FFT processors have been expected in many signal processing applications. In this paper, a new architecture of a pipelined FFT processor is proposed to meet higher performance requirements. The processor is an SIMD machine which consists of processing elements (PE's), their interconnection circuit, and a control unit. A multiplier, an adder and memories are used as main components in each PE. One of the most important features in the FFT processor is to realize a pipelining operation so that the operating ratios of the functional elements in PE's become high. Especially, the operating ratio of the multiplier is a key point because it is very cost-effective in signal processing hardware components. In order to get the higher throughput in the multiplier, two RAM's for alternative storing and loading are provided in each memory unit. Thus, the processor can be operated at the highest speed in the FFT processors with the same number of the multipliers. An auto-scaling circuit is also provided to get a higher-precision arithmetic. The 1024-point, 8-bit FFT processor with 11-stage PE's including a windowing stage has been implemented by conventional TTL IC's. The sampling frequency of 2.6 MHz and FFT processing time of 0.4 msec can be obtained. This performance is superior to those of the other implemented FFT processors and enables many ultra-high-speed signal processing appli-

cations.

Key Words: fast Fourier transform, operating ratio, pipelined processor, processing element (PE)

1. ま え が き

高速フーリエ変換(FFT)¹⁾は、信号スペクトル解析の手段などとして広い応用が知られているが、その高速処理はますます必要とされている。FFTは、バタフライ演算の規則的なくり返しにより実行できるため、従来から、高速化を目的としてパイプライン処理に基づいた専用プロセッサが提案されてきた^{2)~4)}。しかしながら、速度や精度などの性能を満足させるために全体の回路構成が複雑であったり、乗算器やメモリなどの機能素子の稼働率に着目したパイプライン処理を完全に行う構成とはなっていないため、必ずしも十分な処理速度が得られないなどの問題点を有していた。

筆者らは、以上のような観点から、メモリ及び演算部からなる同一構造を有するプロセッシングエレメント(以下PEと呼ぶ)を複数個縦続接続し、SIMD(Single Instruction Stream, Multiple Data Stream)に基づくマイクロプログラム制御により、各PE内部におけるバタフライ演算はもちろん、PE間においても徹底したパイプライン処理を適用して、LSI化にも適した超高速FFTプロセッサを開発した⁵⁾。本プロセッサの最も基本となるアーキテクチャは、PE内の機能素子の稼働率を考慮してパイプライン処理を効率よく行う点にある。特に、デジタル信号処理では最も重要なハードウェアである乗算器に対して、ほぼ100%の稼働率を原理的に実現できる。すなわち、同一個数の乗算器を使用するFFTプロセッサの中では、原理的にも最高の処理速度を有するシステムを実現できることになる。本プロセッサを市販の個別ICにより試作した結果、データ語長8ビット、1,024点

† 第25回自動制御連合講演会で発表(昭57・11)

* 東北大学工学部、仙台市荒港字青葉

** 旭化成工業(株)システム機器エンジニアリング研究所
富士市鯉島 2-1

* Faculty of Engineering, Tohoku University, Sendai

** System's Engineering Laboratory, Asahi Chemical Industry Co. Ltd., Fuji

(Received October 2, 1984)

(Revised March 5, 1985)

複素入力 FFT を 0.4 msec で実行できることが確認され、従来困難であった高速処理が可能となったので、その構成法の詳細について述べることにする。

2. FFT プロセッサのアーキテクチャ

基数 2 の $N=2^m$ 点 FFT は、 m ステージからなるバタフライ演算を、各ステージごとに $N/2$ 回行う必要がある。バタフライ演算は、複素入力 A, B 及び回転因子 W^k に対し $A'=A+W^k B$ と $B'=A-W^k B$ の出力を得るもので、(1) 式で示されるように 4 回の乗算が必要となる。但し、 Re は実部、 Im は虚部を示す。

$$\left. \begin{aligned} \text{Re}(A') &= \text{Re}(A) + \{\text{Re}(B)\text{Re}(W^k) \\ &\quad - \text{Im}(B)\text{Im}(W^k)\} \\ \text{Im}(A') &= \text{Im}(A) + \{\text{Im}(B)\text{Re}(W^k) \\ &\quad + \text{Re}(B)\text{Im}(W^k)\} \\ \text{Re}(B') &= \text{Re}(A) - \{\text{Re}(B)\text{Re}(W^k) \\ &\quad - \text{Im}(B)\text{Im}(W^k)\} \\ \text{Im}(B') &= \text{Im}(A) - \{\text{Im}(B)\text{Re}(W^k) \\ &\quad + \text{Re}(B)\text{Im}(W^k)\} \end{aligned} \right\} \quad (1)$$

Fig. 1 Eight-point FFT by decimation-in-time

Fig. 2 Block diagram of the pipelined FFT processor

Fig. 1 は $N=8$ の場合の時間まびき法による FFT の信号流れ図であり、本プロセッサはこのアルゴリズムに基づいて構成される。

プロセッサ全体の構成を Fig. 2 に示す。初段では、実時間で AD 変換された入力信号が逐次到来する。各ステージは、メモリ及び演算部からなる同一の PE で構成され、初段において入力信号に窓関数を施すステージに引き続いて、 m 段の PE が連続接続されている。各 PE では、 $N/2$ 回のバタフライ演算を FFT アルゴリズムの各ステージと対応させてパイプライン的にを行い、その出力を逐次、次段のメモリへ書き込んでおく。以下、この 1 ステージが処理されている区間を 1 サイクルと呼ぶことにする。各 PE は、共通の制御部内のマイクロプログラムによりその動作が制御される。

本プロセッサの最大の特徴は、バタフライ演算用の入力データの読み出しと、前段からのバタフライ演算結果の書き込みを同時に行えるように異なるアドレス信号を有する 2 個のメモリ (RAM) を備えていること

Fig. 3 Data flow in the eight-point pipelined FFT

である。N/2回のバタフライ演算が終了すると、それまで書き込み用であったメモリが読み出し用のメモリ、すなわち、つぎのサイクルの入力データを読み出すためのメモリとして使用される。一方、今まで読み出し用であったメモリが書き込み用として使用される。以上により、PE 間においてもパイプライン的なデータの流れが可能となり、N点の入力信号を1フレームとするデータが逐次到来する場合、プロセッサ全体としてバタフライ演算を行う乗算器の稼働効率を原理的にほぼ100%に向上できる。Fig. 3は簡単のためN=8点のFFTに対して、このデータの流れを図示したものである。前段からの1フレーム分の出力が各サイクルごとに次段に書き込まれ、同時に、前回のサイクルで書き込まれた1フレーム分のデータが読み出されることにより、バタフライ演算が常に行われることがわかる。

3. ハードウェア構成

3.1 メモリ演算部

Fig. 4はPE内部のメモリ部の構成を示している。制御信号Qの値に従い、二つのRAMのうちのM₁あるいはM₂のいずれか一方に、前段からのデータが書き込まれると同時に、他方のRAMは既に前回のサイクルで書き込まれたデータが読み出される。すなわち、制御信号Qは、1サイクルの最初に1回EX-

Fig. 4 Memory unit in a PE

CHGの信号が1となるごとにDフリップフロップの状態を反転させ、Q=1のときM₁に、Q=0のときM₂に演算結果を記憶する。時間まびき法によるFFTでは、各ステージごとの出力データを記憶するときに、ストアアドレスがビット逆順序により並べ替えられるため、各ステージ固有のストアアドレスの接続が行われている。また、回転因子W^Nも、各ステージ固有のデータであらかじめROMに記憶されている。

Fig. 5は、演算部の構成を示している。乗算器はメモリからのデータを入力とするレジスタKRとJRで示される乗算入力に対してレジスタMRに出力を格納する。また、加算器はマルチプレクサによって選択されたデータを2入力として、結果をAccAあるいはAccBに格納する。演算部内のレジスタ全ては、乗算及び加算においてパイプライン的なデータの流れが可能となるためのパイプラインレジスタの役割を果たしており、それぞれの制御信号の立上がりで新しいデータ値が取り込まれ、これが立下がりでレジスタの出力に表われてくるものとする。

以下の議論では、最大の場合でも、加算は1単位クロック、乗算は2単位クロックそしてメモリアクセスは1単位クロック分の遅延を有しているものとする。

Fig. 5 Arithmetic unit in a PE

Fig. 8 Microprogram definition

(注1) 文献3)によると理論的には妥当な仮定であると考えられるが、実際は、個別 IC の選び方により多少異なることがある。

のくり返し回数を計数するカウンタである。また、アドレスカウンタとこれに接続されている並列加算器は、メモリ部のロード及びストアアドレスをカウンタ値から減算により算出するための回路である。パイプライン的な制御信号の発生を演算部と同じ1クロック単位のピッチで行えるように、パイプラインレジスタ2段を通して出力されている。Fig. 8は、このようにして発生されるマイクロプログラムのフィールドフォーマットを示している。これは、水平形20ビットから構成され、算術演

ST1 フィールドは、メモリ部の RAM の書き込み制御を行う。EXCHG フィールドは、FFT の各サイクルの切換え、さらにマイクロプログラムカウンタと

アドレスカウンタのクリアを行い、新たな1サイクルのための初期設定を行う。

INC フィールドは、アドレスカウンタのインクリメントを行うもので、このカウンタの内容を以下 A と表わすことにする。メモリ部の RAM のロードアドレス LA_d は(2)式のように表現される。

$$LA_d = 2[A - \{3 - (2X_{1-1} + X_{1-2})\}] + LCPX \quad (2)$$

[] 内は、Fig. 7 の上部の加算器からの出力を示しているが、 $(X_{1-1}X_{1-2})_2$ で表わされる2進数に対して、1の補数(各ビットを反転)をとった値だけ減算される。また、回転因子用 ROM のロードアドレス LA'_d は、[] の下位1ビットを切り捨てることにより(3)式で与えられる。

$$LA'_d = F(A - \{3 - (2X_{1-1} + X_{1-2})\}) + SCPX \quad (3)$$

但し、 $F(x)$ は整数 x が奇数のとき $x-1$ 、 x が偶数のとき x の値をとる関数である。さらに、RAM のストアアドレス SA_d は(2)式と同様にして(4)式のように表現できる。

$$SA_d = 2[A - \{3 - (2X_{2-1} + X_{2-2})\}] + SCPX \quad (4)$$

但し、この SA_d は、各ステージごとに異なるビット逆順序による並べ替えが行われている。(2)、(3)、(4)式においては、最下位ビットは、LCPX 及び SCPX により決定されるが、これは、複素データにおいて実部のとき0、虚部のとき1を示している。

(3) 制御グループ

ループカウンタは、 2^{m-1} 進カウンタであり、1ステージ内のバタフライ演算の回数に対応して、LOOP=1 でクリアされ、PLR₀ の出力の REPEAT=1 をクロックとしてインクリメントすることにより、ループのくり返し回数を計数する。ループカウンタの出力が全て1以外すなわち最大値 $(2^{m-1}-1)$ 以外は、PLR₁ からの REPEAT=1 の信号と同期して、マイクロプログラムカウンタにあらかじめ設定されたループの戻り番地であるロードアドレスが設定される。したがって、ループを脱出するときは、PLR₁ の出力の REPEAT=1 を通過した時点でループカウンタの出力が全て1のときである。

3.3 オートスケーリング回路

FFT においては、有限語長による演算誤差が生じるため、点数 N が多い場合は、語長を十分長くとる必要がある。しかしながら、語長の増加は処理時間及びハードウェア量を増大させるため、より少ない語長で高精度演算を行うハードウェアを検討する必要がある。

今、Fig. 5 に示される演算部のデータ語長に関しては、乗算器が $n \times n$ ビット入力で、 n ビット出力、

Fig. 9 Auto-scaling circuit

加算器が n ビット2入力で $n+1$ ビット出力を有するものとする。但し、加算出力が再び加算入力として使用されるループは、単にデータの転送が目的であり、 n ビットのままである。このような固定小数点演算においては、FFT の1ステージごとのバタフライ演算結果を $1/2$ だけ乗じて切捨てあるいは丸めを施し、次段のメモリに転送すれば、オーバーフローは完全に避けることができる。しかし、入力データ値によっては、与えられたダイナミックレンジを十分活用していない場合がある。特に、微小な入力信号に対しては、演算結果が0に近くなるため、各ステージを通るたびに $1/2$ を乗じることによる切捨てあるいは丸め誤差の影響が大きくなる。そこで、オーバーフローしない範囲でこれらの誤差をできるだけ小さくするために、Fig. 9 のオートスケーリング回路を考案した。

加算器の出力は、 x_n を符号ビットとする2の補数で $(x_n x_{n-1} \dots x_0)_2$ のように表わされるとする。これに対して、Fig. 9 のように各 PE 間で次段のメモリに、この値を書き込むタイミングで、(5)式の条件を破線内の EXOR (排他的論理和) 回路により判定する。

$$\left. \begin{array}{l} x_{n-1}=1, x_n=0 \text{ のとき} \\ x_{n-1}=0, x_n=1 \text{ のとき} \end{array} \right\} \quad (5)$$

この条件に一致したら、次段において下位1ビットを切捨てることにより、上位 n ビットを転送させる。このため、スケールファクタ用のカウンタをインクリメ

ントし、この状態を 1 サイクルの間保持し続ける。

破線で示されている部分は、1 種のコンパレータであり、1 サイクルの最初すなわち EXCHG のタイミングで $\bar{Q}=0$ に DFF がセットされる。同時に、各ステージのカウンタはロードモードとなり、前ステージからのカウント値がシフトレジスタのようにして移動する。さらに、コンパレータ出力値 $\bar{Q}$ は、接続されている DFF にロードされ、次ステージにおける転送制御のための信号 SF_i をラッチする。この SF_i により、 $SF_i=1$ のとき下位 1 ビットを切捨て上位 n ビットを転送する。一方、 $SF_i=0$ のときは、下位 n ビット全てをオーバーフローなしに転送できる。したがって、カウンタには、1 つのフレームに対応する各ステージのスケールファクタの累積値が (6) 式のように求められる。

$$SF(i) = \sum_{k=1}^i SF_k \quad (6)$$

但し、入力信号は $(n+1)$ ビットの精度で A/D 変換され、 SF_i は、窓関数を施された入力自体のスケールファクタである。最終的には、 $i=m$ での $SF(i)$ の値により、 $2^{-SF(m)}$ のスケーリングが行われたことを意味している。

4. ファームウェア

Fig. 8 のマイクロプログラムの定義に従い、実際にマイクロプログラムを構成したものが Fig. 10 である。マイクロプログラムカウンタの各アドレス (以下 MPC と記す) は、Fig. 6 のタイムチャートのクロック番号に対応するタイミングを示していることとなる。

(1) $MPC=0$: Fig. 11 (a) のように、前段から

Address	Arithmetic Memory & Control Address
0	1000000000 0011111111 00
1	0000000000 0011111110 00
2	1010000000 0011011100 00
3	0000000100 0011011100 00
4	1010000000 0011011110 00
5	0100110100 0001011111 10
6	1011101001 0010110000 00
7	0000001001 1010101000 00
8	1011110100 0011111111 00
9	0100101010 1000101000 00
10	1011100110 0011011100 00
11	0000000110 101100010 01
12	1011110101 0011011110 00
13	0100110101 1001000111 00
14	1011101001 0010110000 00
15	0000001001 1000101000 00
16	0011110100 0000100000 00
17	0100101010 1001110000 00
18	0001100110 0000000000 00
19	0000100010 100000110 00
20	0001110101 000000010 00
21	0000000001 100001010 00
22	0000000000 0100000000 00
23	0000000000 0000000000 00
(No Operation)	
31	0000000000 0000000000 00

Fig. 10 Microprogram coding

の出力すなわち次段の入力データは逐次格納されていくものとする。また、回転因子はあらかじめ Fig. 11 (b) のように ROM に記憶されている。INC=1 により、 $Re(B)$ と $Im(W^*)$ が JR と KR それぞれロードされる。これらのデータは、クロックの立上がりで乗算器に入力される。

(2) $MCP=1$: 乗算出力待ちのためのノーオペレーション。

(3) $MCP=2$: 乗算出力が MR に転送されると共に、JR と KR に $Re(B)$ と $Re(W^*)$ がロードされる。

(4) $MPC=3$: AccB に MR の内容が転送される。

(5) $MPC=4$: 乗算出力が MR に転送されると共に、JR と KR に $Re(B)$ と $Im(W^*)$ がロードされる。

(6) $MPC=5$: MR から AccB の内容が減算され、AccB に格納される。また、INC=1、 $(X1-1, X1-2)=(0, 0)$ のアドレス制御により、 $Re(A)$ が LR にロードされる。ここで、LOOP=1 とすることにより、 $MPC=6$ 以降のくり返し演算のためのループカウンタをクリアしている。

(7) $MPC=6$: 乗算出力が MR に転送されると共に、JR と KR に $Im(B)$ と $Re(W^*)$ がロードされる。さらに、AccB と LR の内容とが加算され、AccA に転送される。

(8) $MPC=7$: MR の内容が AccA に転送されると共に、AccA の内容が次段のメモリのストアアドレスに書き込まれる。

(9) $MPC=8 \sim 13$: $MPC=0 \sim 5$ と同様な動作に加えて、(1) 式のバタフライ演算結果を次段のメモリに書き込む動作が追加される。但し、 $MPC=11$ においては、REPEAT=1 とすることにより、制御部のパイプライン動作に起因する 2 マイクロステップの遅れを考慮して、マイクロプログラムカウンタに戻り番

Fig. 11 (a) Data structure in RAM
(b) Data structure in ROM

地6をロードするための制御信号を発生している。 2^{m-1} 進カウンタであるループカウンタの出力が全て1かつ REPEAT=1 の条件になると、マイクロプログラムカウンタはインクリメントされ、ループを脱出することになる。

(10) $MPC=14\sim 21$: 1 サイクルの最後のバタフライ演算を行い、次段のメモリにストアする。すなわち、Fig. 6 のクロック 6~13 に示される動作そのものに相等する。

(11) $MPC=22$: EXCHG=1 とすることにより、マイクロプログラムカウンタ及びアドレスカウンタをクリアし、つぎのサイクルのための初期設定を行う。

(12) $MPC=23\sim 31$: 2^5 進カウンタをマイクロプログラムカウンタとして用いたため、便宜上、そう入されたノーオペレーションである。

5. 試作結果と性能

本プロセッサの最小単位であるクロック周期を t_c とすると、 N 点 FFT の動作性能は以下のように与えられる。

$$\left. \begin{array}{l} \text{サンプリング周波数: } f_s = 1/4t_c \\ \text{FFT 処理時間: } T = 4Nt_c \\ \text{周波数分解能: } f = f_s/N \\ \text{滞在時間 (Latency): } T_l = (m+1)T \end{array} \right\} \quad (7)$$

(a)

(b)

Photo. 1 (a) Processing element
(b) FFT processor

本プロセッサの動作を確認するため、市販の TTL などの個別 IC を用いて試作した諸元と性能を以下に示す。

点数: $N=1024$ 点, データ語長: $n=8$ ビットであり、Photo. 1 は、本プロセッサの PE と、PE をウィンドーステージも含めて 11 個用いたプロセッサ全体の概観を示す写真である。なお、並列乗算器として MPY-12 HJ (最大遅延 115 nsec), RAM として MB 8149-45 (45 nsec), また、ROM として MB 7138 H (45 nsec) などを用いている。

この結果、クロック周期 $t_c=97$ nsec, サンプリング周波数 $f_s=2.6$ MHz, FFT 処理時間 $T=0.4$ msec, 周波数分解能 $f=2.54$ kHz, 滞在時間 $T_l=4.4$ msec の性能で良好に動作できることが確認された。試作プロセッサの実際の速度限界は、乗算器やメモリではなく、むしろ、くり返しループの制御のためのカウンタのインクリメントや、アドレス発生のための加算などで決定されている。これらは、カスタム LSI 化により、カウンタや加算器が乗算器やメモリなどと同等のプロセス技術で実現されれば、容易に改善できるものであると考えられる。

Fig. 12(a) は、AE 信号入力をオートスケール回路を含む試作プロセッサにより、約 $f_s=1.34$ MHz で処理しこのパワースペクトラムを表示したものである⁵⁾。比較のため、同一入力に対して、オートスケールなしの 14 ビットの演算精度でシミュレーションを行ったものを Fig. 12(b) に示す。これにより両者はほとんど同じ結果が得られることがわ

(a)

(b)

Fig. 12 (a) Power spectrum of AE signal processed by the implemented FFT processor
(b) Simulated result

かる。

6. む す び

本論文で提案したパイプライン FFT プロセッサは、乗算器を原理的に 100% の効率で動作させることが可能であり、同一個数の乗算器を用いるアーキテクチャの中では最適であると考えられる。

このため、試作プロセッサの性能は極めて優れており、従来不可能であった多くの応用が期待される。現在、本プロセッサは、このような応用の一つである高速自動 AE 解析装置として実用化されている⁶⁾。この場合、オートスケーリングの効果により、これを行わない固定小数点 12~14 ビットの精度に相等する処理が行えることが明らかにされている。本プロセッサのより高速化及び高信頼化を目的とした LSI 化も推進中であり、さらに、その特長が最大限発揮されると考えられる。

最後に、東北大学大学院で本プロセッサの基礎的研究を行った藤原昭広氏(現在、キャノン(株)勤務)、シミュレーションで御協力頂いた本学川又政征助手、並

びに応用面で御協力頂いた本学新妻弘明助教授に感謝する。

参 考 文 献

- 1) J.W. Cooley and J.W. Tukey: An Algorithm for the Machine Calculation of Complex Fourier Series, *Mathematics of Computation*, **19**, 297/301 (1965)
- 2) 所, 森, 金子, 島田, 内田, 相磯: ハードウェアによる高速フーリエ変換, *電子通信学会論文誌 (D)*, **58-D**, 9, (1975)
- 3) K. Hwang: *Computer Arithmetic*, John Wiley & Sons. Inc. (1979)
- 4) L.R. Rabiner and B. Gold: *Theory and Application of Digital Signal Processing*, Prentice-Hall, 573/626 (1975)
- 5) 亀山, 樋口, 今野, 高須賀: パイプライン方式に基づく FFT 専用プロセッサの構成, 第 25 回自動制御連合講演会前刷, 2028 (1982)
- 6) H. Niitsuma, N. Chubachi, M. Kameyama, T. Higuchi and H. Sekine: Automatic AE Spectrum Analyzed Using a New Pipelined FFT Processor, *First Int. Symp. on Acoustic Emission from Reinforced Composites*, The Society of the Plastic Industry, Inc., Session 5, 1/7 (1983)