計測自動制御学会東北支部 第196回研究集会 (2001.7.30)
資料番号 196-4

分散演算を用いた複素FIRフィルタの高性能VLSIアーキテクチャ

High-Performance VLSI Architecture of Complex FIR filters Using Distributed Arithmetic

○野崎剛*，恒川佳隆*，田山典男*

○Takeshi Nozaki*, Yoshitaka Tsunekawa*, and Norio Tayama*

*岩手大学 工学部

*Faculty of Engineering, Iwate University

キーワード： 分散演算 (distributed arithmetic)，複素FIRフィルタ (complex FIR filters)，最適関数回路 (optimum function circuit), SFA(Serial full adder)，SFS(Serial full subtractor)

連絡先： 〒020-8551 盛岡市上田4–3–5 岩手大学 工学部
野崎剛，Tel.: (019)621-6468，Fax.: (019)621-6468，E-mail: t5300005@iwate-u.ac.jp

1. はじめに

近年，画像を送信する携帯電話や携帯情報端末に代表されるようなマルチメディア携帯端末が注目されている[1)]．マルチメディア携帯端末では，画像処理，音声処理，通信処理などの多岐にわたりディジタル信号処理を用いている．ゆえに，信号処理プロセッサを用いてディジタル信号を実現することが一般的である．また，複素信号処理の有効性が明らかにされるにつれ，複素FIRフィルタが注目されている[2)]．携帯情報端末では主に複素信号処理を主体とするデータ通信用変複調処理には，複素数を直接取り扱える複素信号処理プロセッサが適している．

複素FIRフィルタの一般的な実現法の1つとして，乗算器を用いた直接型構成に基づく実現法がある．これを用いて高いサンプリングレートを実現する手法として，タップ毎にパイプライン処理を施す方法が考えられる．しかし，高次で用いると滞在時間が非常に大きくなる．また，タップ数を1増加すると4個の乗算器と4個の加減算器を用いるため，比較的低次の複素FIRフィルタでも，実FIRフィルタとは異なり膨大な消費電力を必要とする．

本稿では，分散演算を用いた複素FIRフィルタの高性能VLSIアーキテクチャを提案する．高次でも滞在時間を考慮した高いサンプリングレートを実現するために，分散演算の処理時間が語長のみに依存することに着目する．また，分散演算を用いた従来の実現法では関数の生成にROMを用いるが，本実現法ではわれわれが提案してきた論理ゲートによって構成される最適関数回路を適用して，大幅な低消費電力化を実現する[3)8)9)]．さらに，最適関数回路を用いた構成にSFA(Serial Full Adder)とSFS(Serial Full Subtractor)を適用した新たな手法を提案し，さらに低消費電力化を図る．最後に，4入力2出力加算を用いた構成法を適用し，消費電力と滞在時間の低減を実現する[9)]．

以上の手法を用いた本実現法は，高次でも滞在時間を考慮した高いサンプリングレートと低消費電力化を実現できることを明らかにする．

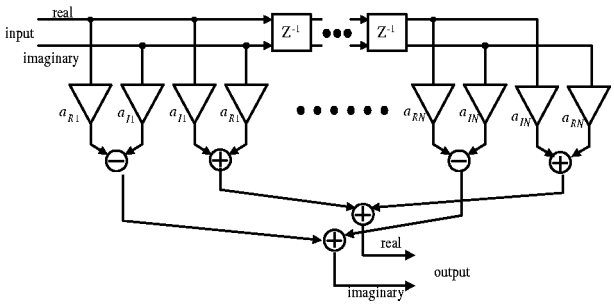


Fig. 1　Structure based on direct form using multipliers with complex coefficients.

2. 分散演算を用いた複素FIRフィルタ

図1に示す直接型構成に基づく複素FIRフィルタの構成では，1タップ増加するごとに4個の乗算器と4個の加減算器を必要とする．そのため，実フィルタとは異なり比較的低次でも膨大な消費電力を必要とする．

そこで，われわれは分散演算からアプローチする．分散演算とは定係数の内積演算をテーブル・ルックアップにより実現する計算手法である[4)~6)]．項数Nの複素係数ベクトル$\boldsymbol{a}=(a_{R1}+ja_{I1}, a_{R2}+ja_{I2}, \ldots, a_{RN}+ja_{IN})$と複素変数ベクトル$\boldsymbol{v}=(v_{R1}+jv_{I1}, v_{R2}+jv_{I2}, \ldots, v_{RN}+jv_{IN})$との内積

$$\begin{aligned} y=\boldsymbol{av} &= \sum_{i=1}^{N}\{(a_{Ri}\cdot v_{Ri}-a_{Ii}v_{Ii}) \\ &\quad +j(a_{Ii}\cdot v_{Ri}+a_{Ri}v_{Ii})\} \quad (1) \end{aligned}$$

を考える．ただし，入力変数v_{Ri}とv_{Ii}は，$-1 \leq v_{Ri}, v_{Ii} < 1$で，$B$ビットの固定小数点形の2の補数表示である．つまり，

$$v_{Ri} = -v_{Ri}^0+\sum_{k=1}^{B-1}2^{-k}v_{Ri}^k \quad (2)$$

$$v_{Ii} = -v_{Ii}^0+\sum_{k=1}^{B-1}2^{-k}v_{Ii}^k \quad (3)$$

と表される．ここで，v_{Ri}^kとv_{Ii}^kはそれぞれv_{Ri}とv_{Ii}のkビット目の値で0または1である．式(2)と(3)を式(1)に代入すると，

$$\begin{aligned} y &= y_{Re}+jy_{Im} \quad (4) \\ y_{Re} &= \{-\Phi_{Re1}(v_{R1}^0,\ldots,v_{RN}^0) \\ &\quad +\sum_{k=1}^{B-1}2^{-k}\Phi_{Re1}(v_{R1}^k,\ldots,v_{RN}^k)\} \\ &\quad +\{\Phi_{Re2}(v_{I1}^0,\ldots,v_{IN}^0) \\ &\quad +\sum_{k=1}^{B-1}2^{-k}\Phi_{Re2}(v_{I1}^k,\ldots,v_{IN}^k)\} \quad (5) \\ y_{Im} &= \{-\Phi_{Im1}(v_{R1}^0,\ldots,v_{RN}^0) \\ &\quad +\sum_{k=1}^{B-1}2^{-k}\Phi_{Im1}(v_{R1}^k,\ldots,v_{RN}^k)\} \\ &\quad +\{-\Phi_{Im2}(v_{I1}^0,\ldots,v_{IN}^0) \\ &\quad +\sum_{k=1}^{B-1}2^{-k}\Phi_{Im2}(v_{I1}^k,\ldots,v_{IN}^k)\} \quad (6) \end{aligned}$$

となる．ただし，Φ_{Re1}，Φ_{Re2}，Φ_{Im1}とΦ_{Im2}はそれぞれ

$$\Phi_{Re1}(v_{R1}^k,\ldots,v_{RN}^k) = \sum_{i=1}^{N}a_{Ri}v_{Ri}^k \quad (7)$$

$$\Phi_{Re2}(v_{I1}^k,\ldots,v_{IN}^k) = -\sum_{i=1}^{N}a_{Ii}v_{Ii}^k \quad (8)$$

$$\Phi_{Im1}(v_{R1}^k,\ldots,v_{RN}^k) = \sum_{i=1}^{N}a_{Ii}v_{Ri}^k \quad (9)$$

$$\Phi_{Im2}(v_{I1}^k,\ldots,v_{IN}^k) = \sum_{i=1}^{N}a_{Ri}v_{Ii}^k \quad (10)$$

である．分散演算を用いた複素FIRフィルタの構成は図2のように示すことができる．これは，入力部，関数生成部と関数加算部で構成される．図1

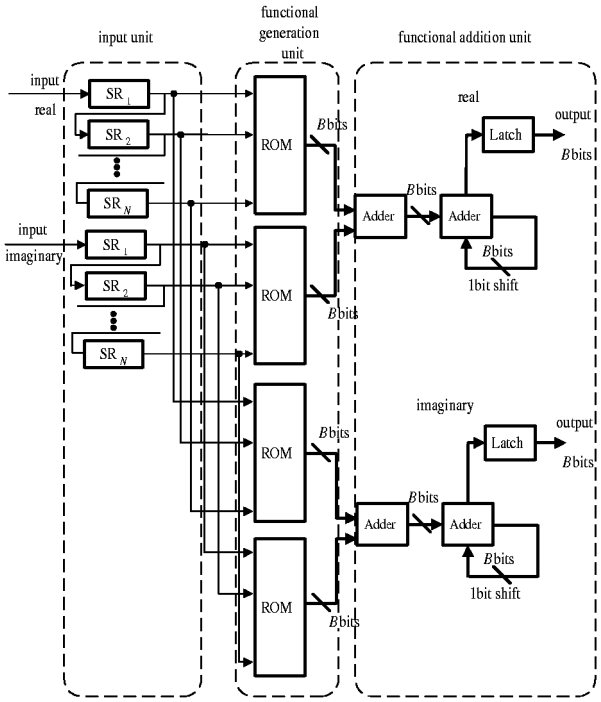


Fig. 2　Basic structure based on the distributed arithmetic with complex coefficients.

に示す直接型構成に基づく複素FIRフィルタでは，1サンプリング周期ごとに入力変数を各遅延器に伝播させて，内積演算を実行している．それに対して，図2では入力部のシフトレジスタを用いて入力変数を1クロックごとに1ビットずつ伝搬させている．関数生成部では，入力部から出力される変数と係数との内積演算の結果Φが格納されているROMを参照しデータを出力する．関数加算部では，関数生成部から出力される値を順次1ビットシフトしながら加算を行う．このように，分散演算を用いた複素FIRフィルタは，乗算器を用いずに実現できるためハードウェア量を抑えられ，しかも語長Bのみに依存した処理時間が可能である．

しかし，複素係数の場合は$2^{(N+2)}$のメモリ容量を必要とするため，比較的高い次数での実現を図ると膨大な消費電力を必要とする．そこで，関数Φの分割化法を用いることが有効である．これは，関数Φに対して，分割数をQとして以下の処理を施す．

$$\begin{aligned}\Phi(v_1^k,\ldots,v_N^k) &= \sum_{i=1}^{N} a_i v_i^k \\ &= \sum_{i=1}^{N/Q} a_i v_i^k + \ldots + \sum_{i=Q'}^{N} a_i v_i^k \\ &= \Phi(v_1^k,\ldots,v_{N/Q}^k) + \ldots + \Phi(v_{Q'}^k,\ldots,v_N^k)\end{aligned} \tag{11}$$

ここで，

$$Q' = \frac{N(Q-1)}{Q} + 1 \tag{12}$$

である．これによって，大きな関数テーブルをいくつかの小さな関数テーブルの加算として実現できる．この分割数が増加すると，関数テーブルの大きさが減少するため関数生成部の消費電力が低減する．一方，このとき関数加算部の加算器数が増加するため，その部分の消費電力が大きくなる．このように，分割数の変化に対して，関数生成部と関数加算部の消費電力はトレードオフの関係が成り立つ．ゆえに，消費電力を最小にする分割数があり，これを最適分割数と呼ぶ．

この分割化により関数生成にROMを用いた従来法では，関数テーブルの大きさを低減できるが，それでも膨大な消費電力を必要とする．

3. 最適関数回路

関数Φの生成にROMを用いた従来法は，膨大な消費電力を必要とする．そこで，われわれは関数ΦをROMで実現する代わりに，論理ゲートを用いてそれと同機能となる最適関数回路を提案してきた[3]．ここでは，ROMに格納されている関数テーブルを，

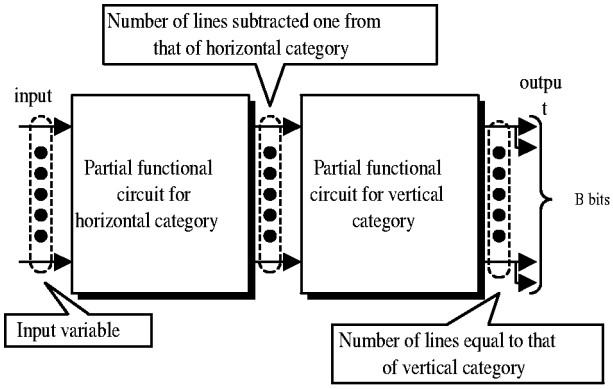


Fig. 3 Structure of optimal functional circuit.

$$\begin{aligned}\boldsymbol{W}_{ROM} &= \begin{bmatrix} \Phi(0,0,\ldots,0) \\ \vdots \\ \Phi(1,1,\ldots,1) \end{bmatrix} \\ &= \begin{bmatrix} w_0^{B-1} & \cdots & w_0^0 \\ \vdots & \cdots & \vdots \\ w_{2^Q-1}^{B-1} & \cdots & w_{2^Q-1}^0 \end{bmatrix}\end{aligned} \tag{13}$$

という$2^Q \times B$の行列で表す．式(13)の$w_i^{B-1}\cdots w_i^0$は，i番地に格納されているBビットの出力データを表す．この式の行または列には同じベクトルが存在する．そこで，これらを共有化することにより，冗長性を削除し関数テーブルの大きさを低減させる．この冗長性の削除により生成される行列は，

$$\boldsymbol{W}_{OFC} = \begin{bmatrix} w_{row_0}^{col_j} & \cdots & w_{row_0}^{col_0} \\ \vdots & \cdots & \vdots \\ w_{row_i}^{col_j} & \cdots & w_{row_i}^{col_0} \end{bmatrix} \tag{14}$$

という$row_i \times col_j$の行列で表せる．ただし，

$$row_i \leqq 2^N \quad col_j \leqq B \tag{15}$$

である．この操作によって生成される式(14)の行または列ベクトルは互いに異なる．式(14)の行と列ベクトルをそれぞれ横のカテゴリと縦のカテゴリと呼ぶ．この式を基にした最適関数回路は，図3に示すように行ベクトルに対応する部分機能回路と列ベクトルに対応する部分機能回路を縦続接続した構成で実現される．これらは，論理ゲートにより構成され，消費電力を大幅に低減できる．この回路は，式(14)の行または列ベクトルが減少するほど，消費電力を低減できる特長をもつ．

この手法を用いると，関数生成部の消費電力を大幅に低減できる．ただし，ROMを用いた構成と

比較して最適分割数が大きくなるため関数加算部において加算器数自体は増加し，その数は実フィルタの場合よりも非常に多く必要とする．われわれは，さらに複素FIRフィルタの直線位相特性に着目して，関数加算部の低消費電力を図る．

4. SFSとSFAを用いた構成

直線位相特性を利用して低消費電力化を図る場合，実係数のFIRフィルタでは係数の偶対称性のみに着目するだけであるが，複素FIRフィルタでは実部の偶対称性だけでなく，虚部の奇対称性も考慮する．

タップ数Nが偶数の場合，直線位相特性を有する複素FIRフィルタの実部と虚部の係数は，それぞれ

$$\begin{aligned} a_{R1} &= a_{RN}, a_{R2} = a_{RN-R1}, \\ &\quad \cdots, a_{RN/2} = a_{RN/2+1} \end{aligned} \tag{16}$$

$$\begin{aligned} a_{I1} &= -a_{IN}, a_{I2} = -a_{IN-I1}, \\ &\quad \cdots, a_{IN/2} = -a_{IN/2+1} \end{aligned} \tag{17}$$

という関係が成り立つ．式(16)と(17)より，実部の内積演算は，

$$y_{Re} = y_{Re1} + y_{Re2} \tag{18}$$

$$\begin{aligned} y_{Re1} &= -\Phi_{Re1}((v^0_{R1} + v^0_{RN}), \ldots, \\ &\qquad (v^0_{RN/2} + v^0_{(RN/2+1)})) \\ &\quad + \sum_{k=1}^{B-1} 2^{-k} \Phi_{Re1}((v^k_{R1} + v^k_{RN}), \ldots, \\ &\qquad (v^k_{RN/2} + v^k_{(RN/2+1)})) \end{aligned} \tag{19}$$

$$\begin{aligned} y_{Re2} &= -\Phi_{Re2}((v^0_{I1} - v^0_{IN}), \ldots, \\ &\qquad (v^0_{IN/2} - v^0_{(IN/2+1)})) \\ &\quad + \sum_{k=1}^{B-1} 2^{-k} \Phi_{Re2}((v^k_{I1} - v^k_{IN}), \ldots, \\ &\qquad (v^k_{IN/2} - v^k_{(IN/2+1)})) \end{aligned} \tag{20}$$

と表される．ただし，Φ_{Re1}とΦ_{Re2}は，

$$\begin{aligned} &\Phi_{Re1}((v^k_{R1} + v^k_{RN}), \ldots, (v^k_{RN/2} + v^k_{(RN/2+1)})) \\ &\qquad = \sum_{i=1}^{N/2} a_{Ri}(v^k_{Ri} + v^k_{RN+1-Ri}) \end{aligned} \tag{21}$$

$$\begin{aligned} &\Phi_{Re2}((v^k_{I1} - v^k_{IN}), \ldots, (v^k_{IN/2} - v^k_{(IN/2+1)})) \\ &\qquad = -\sum_{i=1}^{N/2} a_{Ii}(v^k_{Ii} - v^k_{IN+1-Ii}) \end{aligned} \tag{22}$$

である．また，式(16)と(17)より虚部の内積演算は，

$$y_{Im} = y_{Im1} + y_{Im2} \tag{23}$$

$$\begin{aligned} y_{Im1} &= -\Phi((v^0_{R1} - v^0_{RN}), \ldots, \\ &\qquad (v^0_{RN/2} - v^0_{(RN/2+1)}))\} \\ &\quad + \sum_{k=1}^{B-1} 2^{-k} \Phi((v^k_{R1} - v^k_{RN}), \ldots, \\ &\qquad (v^k_{RN/2} - v^k_{(RN/2+1)})) \end{aligned} \tag{24}$$

$$\begin{aligned} y_{Im2} &= -\Phi((v^0_{I1} + v^0_{IN}), \ldots, \\ &\qquad (v^0_{IN/2} + v^0_{(IN/2+1)})) \\ &\quad + \sum_{k=1}^{B-1} 2^{-k} \Phi((v^k_{I1} + v^k_{IN}), \ldots, \\ &\qquad (v^k_{IN/2} + v^k_{(IN/2+1)})) \end{aligned} \tag{25}$$

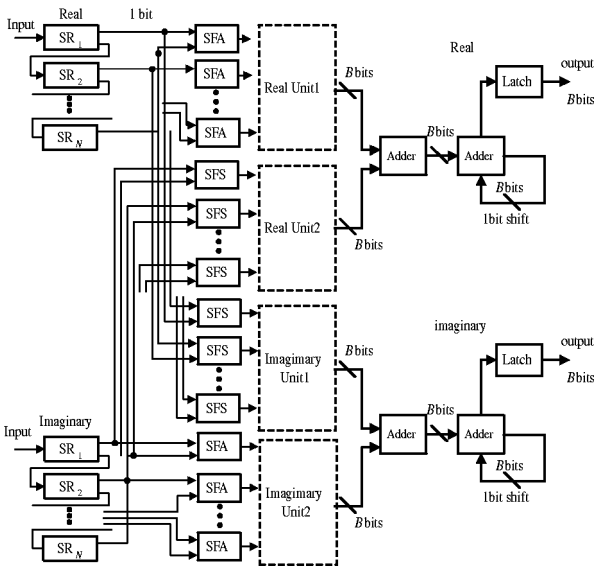
と表される．ただし，Φ_{Im1}とΦ_{Im2}は，

$$\begin{aligned} &\Phi_{Im1}((v^k_{R1} + v^k_{RN}), \ldots, (v^k_{RN/2} + v^k_{(RN/2+1)})) \\ &\qquad = \sum_{i=1}^{N/2} a_{Ri}(v^k_{Ri} + v^k_{RN+1-Ri}) \end{aligned} \tag{26}$$

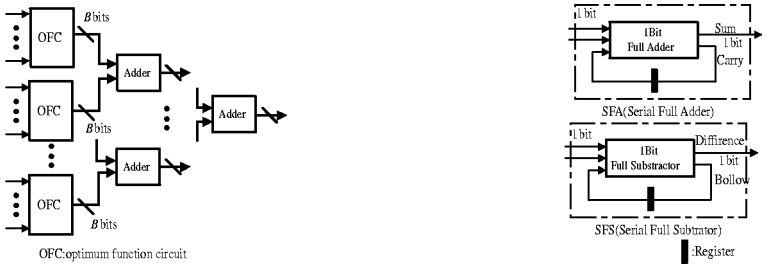
$$\begin{aligned} &\Phi_{Im2}((v^k_{I1} - v^k_{IN}), \ldots, (v^k_{IN/2} - v^k_{(IN/2+1)})) \\ &\qquad = \sum_{i=1}^{N/2} a_{Ii}(v^k_{Ii} - v^k_{IN+1-Ii}) \end{aligned} \tag{27}$$

である．係数が対称であるために，式(18) ~ (27)に示すように2つの入力変数を括弧で括ることができる．この括られた入力変数を1つの項として処理できると，項数の大きさを1/2まで減少した内積演算とみなせる．

われわれはそのことに着目して，図4に示すような最適関数回路を用いた構成にSFA(Serial Full Adder)とSFS(Serial Full Subtractor)を用いた新たな構成法を提案する．SFAを用いた構成部の処理は図5のように表すことができる．ここでは，まず式(19)と(25)の入力変数の最下位ビットv^0_iとv^0_{N-i}の加算を実行する．これによって生成される和をアドレス値として，最適関数回路から関数Φを出力する．また，このとき発生するキャリーを図4(c)に示すSFAのレジスタに格納し，入力変数の上位ビットv^1_iとv^1_{N-i}の加算時に入力として用いる．このような処理を最上位ビットまで繰り返す．一方，SFSを用いた構成部の処理は図6のように表すことができる．ここでは，式(20)と(24)の入力変数の最下位ビットv^0_iとv^0_{N-i}の減算を行う．このとき生成される差をアドレス値として，最適関数回路から関数Φを出力する．また，この減算処理で発生したボロー(桁借り)を図4(c)に示すSFSのレジ



(a) Whole structure.



(b) Structure of real unit and imaginary unit. shown in Fig4.(a).

(c) Structure of SFA and SFS.

Fig. 4 Structure using SFA and SFS.

スタに格納し，上位ビットv_i^1とv_{N-i}^1の減算時に入力として用いる．SFSでも同様に最上位ビットまで繰り返す．なお，オーバーフローを防ぐために，$-0.5 \leqq v_i^k < 0.5$に制限(Scaling)する．

本構成では，SFAとSFSの両方を適用して関数テーブルの大きさを1/2まで減少させたことにより，関数加算部の加算器数を1/2まで低減することが可能となる．その結果，最適関数回路の適用による加算器数自体の増加を抑えられるため，大幅な低消費電力化ができる．

われわれは，さらに高次に適した複素FIRフィルタを実現するため，関数加算部の性能の向上を図る．

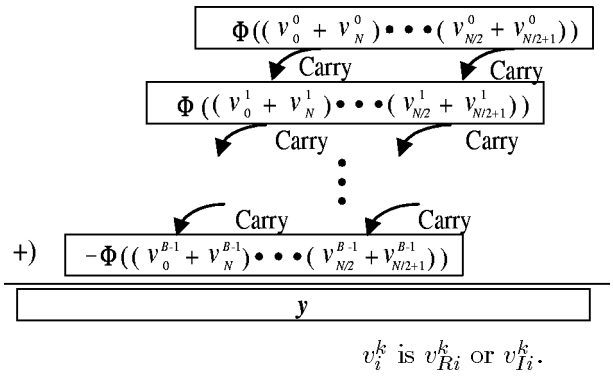


Fig. 5 Fundamental diagram of structure using SFA.

5. 4入力2出力加算器

従来の分散演算を用いた実現法では，関数加算部にCLA加算器によるBinary Treeを用いている．これを高い次数で用いると，関数加算部では加算器数を多く必要とする．

そこで，われわれが提案してきた，4入力2出力加算器を用いた新しい構成法に着目する[9]．一般的

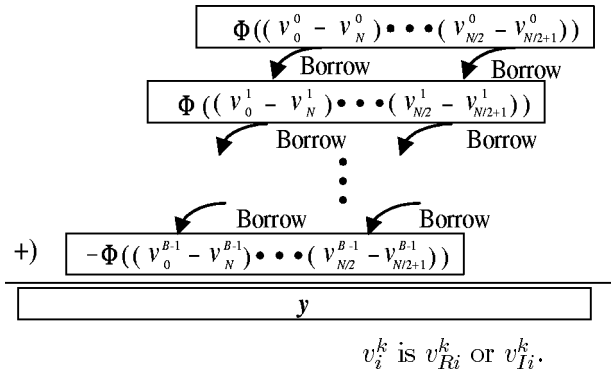


Fig. 6 Fundamental diagram of structure using SFS.

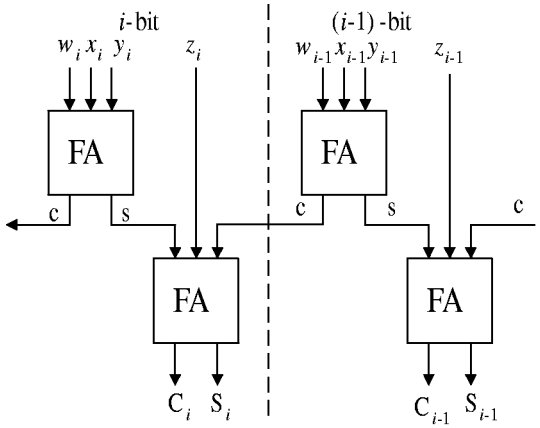
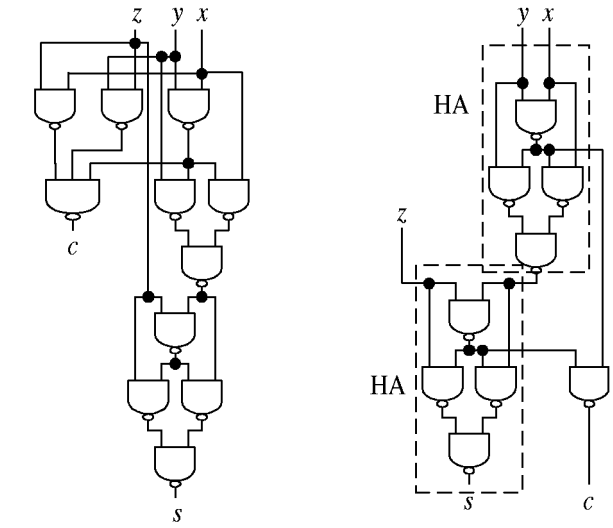


Fig. 7 General structure of 4to2adder.

な4入力2出力加算器をFA(Full Adder)で構成する場合，一般的に図7に示すようにFAを縦続接続した構成が考えられる．この図の1段目のFAでは，3つの入力変数w,x,yを同時に入力し，和とキャリーを出力する．2段目のFAでは，1段目のFAから出力された和s，入力変数zと1ビット下位から伝搬されたキャリーcとの加算を行っている．この処理を高速に行うためには，図7の1段目から出力されるキャリーの伝搬を高速に行う必要がある．そして，1段目と2段目のFAをほぼ同時に並列処理させることによって，より高速な処理を実現できる．

FAには，図8に示すように単体構成と2個のHA(Half Adder)を縦続接続した構成が考えられる．図8(a)は，3個の入力変数を同時に入力する必要があり，キャリーを高速に求めることができる．図8(b)は，入力変数zをHAの処理時間分遅れて入力でき，FAの単体構成より少ないゲート数で構成できる特長をもつ．これらのFAの特徴に着目して，高速でしかもハードウェア量を抑えた4入力2出力加算器を構成する．

図7の1段目のFAで，上位へのキャリー伝搬を高速に行うために，図8(a)の単体構成を用いる．一方，2段目のFAでは，高速な処理を実現するため，1段目のFAに対しほぼ同時に並列処理させる必要がある．2段目のFAにおいて，1段目のFAから出力される和sは，入力変数zとキャリーcより遅れて伝搬されるため，図8(b)の縦続接続した構成を用いる．これを用いることにより，1段目と2段目のFAをほぼ同時に並列処理でき，処理時間を低減することができる．さらに，これは少ないゲート数で実現できるため，ハードウェア量を抑えることもできる．以上のFAで構成された4入力2出力加算器は，図9のようになる．



(a) Structure composed of one unit

(b) Structure composed of two HA's

Fig. 8 Structure of FA.

この4入力2出力加算器を関数加算部に適用した構成を図10に示す．この構成を用いることにより，図10(a)に示す従来法のCLA加算器数をaとすると，本提案法では本加算器数を$a/2$まで低減することができ,しかもハードウェア量を抑えた構成であるため，消費電力も低減できる．さらに，本加算器では処理時間の低減を図ったことにより，CLA加算器の処理時間よりも大幅に短くできる．加算段数は同じであるが，これらの構成のパイプラインのピッチは，図10に示す関数加算部の最終段のCLA加算器の処理時間に依存するため，パイプライン段数を低減でき，滞在時間を小さくできる．これは，次数の増加に伴い効果が大きくなるため，高次には適した手法であるといえる

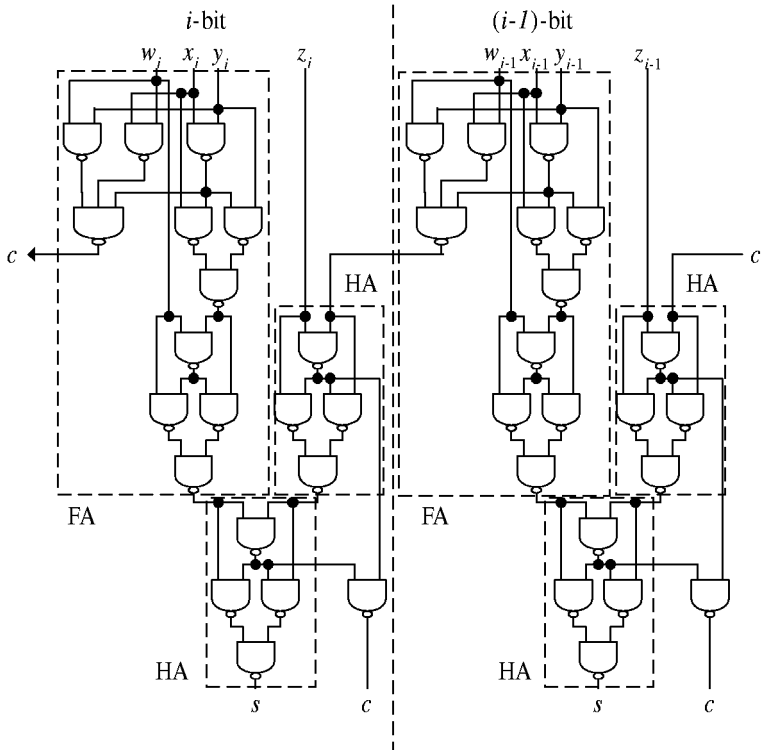
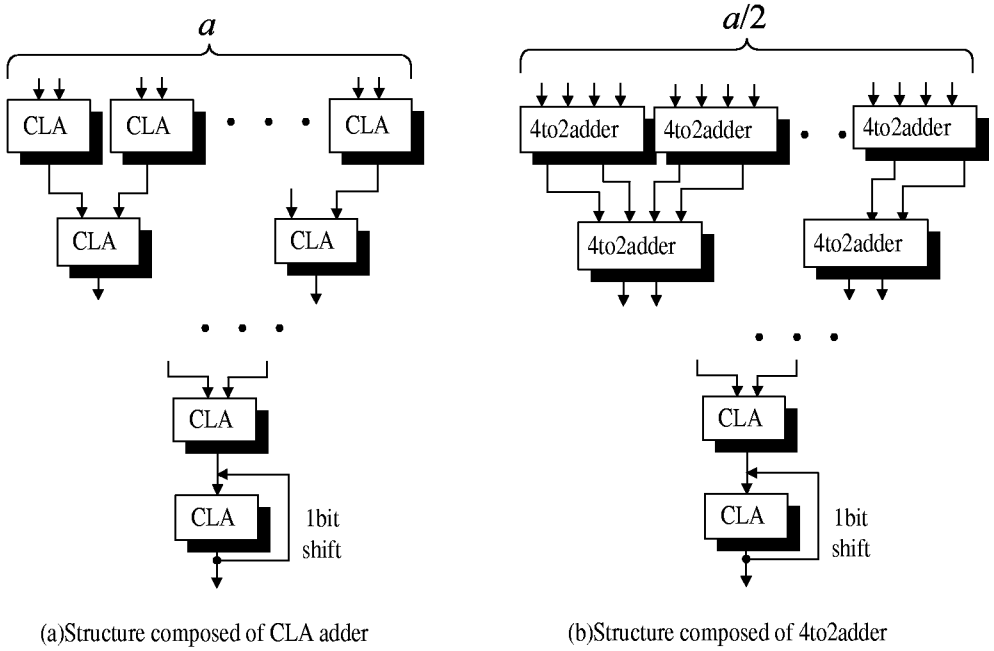


Fig. 9 Proposed 4to2 adder.



(a)Structure composed of CLA adder (b)Structure composed of 4to2adder

Fig. 10 Structure of functional addtion unit.

Table 1　Evaluation of 60-tap complex FIR filters.

	Proposed method	Conventional Method using ROM's	General method using multipliers for direct form
Power dissipation[W]	1.62	11.29	30.5
Area [mm^2]	7.06	16.9	181.3
Number of gates	34591	51861	852362
Sampling rate [MHz]	2.65	2.65	2.36
Latency [ns]	486	567	1154

6. VLSI評価

これまで述べてきた構成法に対して，VLSI設計システムPARTHENONで設計および評価を行う[10]．なお，実部品として用いたセル・ライブラリーの設計ルールは，0.8μmCMOSスタンダードセル(VLSIテクノロジ社)とし，電源電圧は5.0Vとする．ここでは，直線位相特性をRemezアルゴリズムを用いて実現する．フィルタ仕様は，規格化通過域端周波数0.12，規格化阻止域端周波数0.15となる低域通過形フィルタとする．

分散演算を用いた従来法では関数の生成にROMを用いている．これを適用した複素FIRフィルタプロセッサと比較すると，表1より本プロセッサの消費電力は従来型プロセッサに対し約85%減少できる．複素FIRフィルタは実フィルタより膨大な消費電力を必要とするため，1%あたりの消費電力の減少量は非常に大きい．そのため，本実現法により達成された低消費電力化は，実フィルタでは得ることが困難なくらい大幅な低消費電力化であるといえる．また，表1より本プロセッサではそれと同時に滞在時間を約14%低減できる．このように，本提案法は最適関数回路を用いた構成とSFA・SFSを用いた構成の特徴を利用し，しかも関数加算部に4入力2出力加算器を用いることによりはじめて，このような大幅な低消費電力化と同時に滞在時間を減少できる．

複素FIRフィルタの一般的な手法の1つとして，乗算器を用いた直接型構成に基づく実現法がある．これは，(タップ数$\times$4)個の乗算器と(タップ数$\times$4)個の加減算器を必要とするため膨大な消費電力を必要とする．そのため，本プロセッサの約18.8倍に相当する消費電力を必要とする．

以上のことから，本プロセッサは比較的高い次数でも大幅に消費電力を低減できると同時に滞在時間を抑えることができるため，複素FIRフィルタの実現に極めて有効な手法であるといえる．

7. まとめ

本稿では，分散演算を用いた複素FIRフィルタのVLSIアーキテクチャを提案した．滞在時間を抑えながら高いサンプリングレートを実現するために分散演算を用いた．また，消費電力を大幅に低減するために最適関数回路を適用した．さらに，最適関数回路の構成による加算器数自体の増加を抑えるためにSFAとSFSを適用し，その上関数加算部に4入力2出力加算器を用いて消費電力と滞在時間をさらに減少させた．以上の手法を用いた本実現法をVLSI評価した結果，60タップという高次の複素FIRフィルタを486ns(0.8 μ mCMOSスタンダードセル)という小さい滞在時間に抑えながら，2.65MHzという比較的高いサンプリングレートを実現させた．また，それと同時に消費電力を1.62Wまで大幅に低減させた．

本提案法は，さらに高い次数でも分散演算の処理時間が語長のみに依存するために，滞在時間とサンプリングレートをほぼ一定に抑えながら，大幅な低消費電力化が可能となる．

参考文献

1) 根岸良征，渡部英二，西原明法，柳澤健：実積和演算にも適した複素乗算ユニットの一構成法，電情報通信学会論文誌A, Vol.J84-A, No.3, pp.278-286 (2001)

2) 小杉宏：複素ディジタルフィルタの周波数サンプリング法による設計と構成，電子通信学会論文誌A, Vol.J60-A, No.11, pp.1007-1014 (1977)

3) 恒川佳隆，野崎剛，三浦守: 滞在時間を考慮した高次FIRフィルタの高速・低消費電力形VLSIアーキテクチャ,電気学会論文誌C, vol.118-C, No. 7/8, pp.1098-1107 (1998)

4) C. F. Chen:Implementing FIR Filters with Distributed Arithmetic, IEEE Trans., Acoust. Speech & Signal Process., **ASSP-33**-4, 1318/1321 (1985)

5) C. S. Burrus:Digital Filter Structures Described by Distributed Arithmetic, IEEE Trans., Circuit & Syst., **CAS-24**, 674/680 (1977)

6) A. Peled and B. Liu:A New Hardware Realization of Digital Filters, IEEE Trans., Acoust. Speech & Signal Process., **ASSP-22**-12, 456/462 (1974)

7) L. Mintzer: FIR Filter with Field-programmable Gate Arrays, Journal of VLSI Signal Processing, **6**, 119/127 (1993)

8) 田村惣一，恒川佳隆，吉田等明，三浦 守:直線位相特性を用いた高次FIRフィルタの 高性能VLSIアーキテクチャ，第181回計測制御東北支部研究集会

9) 野崎　剛，恒川佳隆，田山典男:分散演算を用いた高次FIRフィルタの高性能VLSIアーキテクチャ，第194回計測制御東北支部研究集会

10) NTT データ通信株式会社:PARTHNON User's Manual (1990)