計測自動制御学会東北支部 第208回研究集会 (2003.5.22)
資料番号 208-7

分散演算を用いた高性能2次元FIRフィルタのVLSIアーキテクチャ

High-Performance VLSI Architecture for 2-D FIR Filters Based on Distributed Arithmetic

野崎 剛*，佐々木友寿*，恒川佳隆*，田山典男*

Takeshi Nozaki*, Tomohisa Sasaki*, 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)，高次形2次元FIRフィルタ (high-order 2-D FIR filtes)，低消費電力(low power dissipation)，滞在時間(latency) VLSI評価 (VLSI evaluation)

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. まえがき

多次元ディジタル信号処理は，画像処理やセンサアレイ信号処理などの幅広い分野で用いられ，今後さらに重要になっていくと考えられる．多次元ディジタル信号のフィルタリングの一つとして，2次元FIRディジタルフィルタがある．これは安定性が常に保証されており，完全な直線位相特性が容易に実現できるため，画像処理などの2次元信号の処理に広く利用されている．しかし，所望の仕様を満足するための必要な次数が高いため，フィルタリングに必要なハードウェア量や計算量が大きくなるという問題がある．

2次元FIRフィルタの実現法の1つとして，乗算器を用いた直接型構成に基づく手法がある[1)]．これは，基本的に各タップごとにパイプライン処理を施して，高いサンプリングレートが得られる．しかし，(M,N)次の場合には$(M+1)\times(N+1)$個もの乗算器を必要とするため，高次フィルタの実現には膨大なハードウェア量と消費電力を必要とする．

本稿では，分散演算を用いた高性能2次元FIRフィルタのVISIアーキテクチャを提案する．まず，内積演算の処理時間が語長のみに依存する分散演算に着目して，次数の増加に対してサンプリングレートを保持しながらほぼ一定の小さな滞在時間に抑える[2)]．しかし，分散演算に基づく従来形の構成はROMを用いている部分の消費電力が大きいため，高次フィルタの実現には非常に大きな消費電力を必要とする．そこで，ROMではなく我々が提案してきた最適関数回路を用いた分散演算に基づく構成に着目する[3)]．これにより，従来形分散演算に基づく構成で問題であった消費電力を解決でき，高次向き2次元FIRフィルタを実現する．しかし，これを極めて高い次数のフィルタの実現に用いると，加算器数が増加するために消費電力が増加してしまう．また，同時に加算段数が増加するために消費電力だけでなく滞在時間が増加してしまう．この問題を解決するために，SFA(Serial Full Adder)を用いた直線位相特性に基づく手法をさらに適用した構成を提案する．これにより，滞在時間を考慮しながら消費電力を減少でき，極めて高い次数に向いた2次元FIRフィルタを実現できる．最後に，本プロセッサに対してVLSI評価を行うことによって，本提案法が極めて高い次数の2次

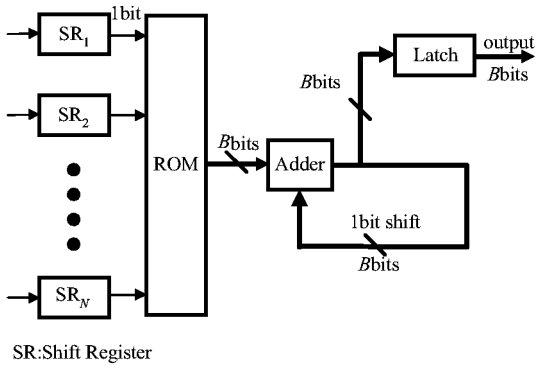


Fig. 1　Basic structure of distributed arithmetic

元FIRフィルタに有効な手法であることを明らかにする．

2.　分散演算に基づく構成

高次向き2次元FIRフィルタを実現するために，乗算器を用いずに実現でき，しかも処理時間が語長のみに依存する分散演算に着目する[2]．そこで，本章では分散演算について示す．次に，2次元FIRフィルタに分散演算を適用できることを示し，その構成を示す．

2.1　分散演算

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いま，項数Nの係数ベクトル$\boldsymbol{a}=(a_1,\ldots,a_N)$と変数ベクトル$\boldsymbol{v}=(v_1,\ldots,v_N)$との内積

$$y=\boldsymbol{a}\boldsymbol{v}=\sum_{i=1}^{N}a_iv_i \tag{1}$$

を考える．ただし，$-1\le v_i<1$で，v_iはBビットの固定小数点形の2の補数表示である．これを

$$v_i=-v_i^0+\sum_{k=1}^{B-1}2^{-k}v_i^k \tag{2}$$

と表す．ここで，v_i^kはv_iのkビット目の値で0または1である．式(2)を式(1)に代入すると，内積演算$\boldsymbol{av}$は，

$$y=-\Phi(v_1^0,\ldots,v_N^0)+\sum_{k=1}^{B-1}2^{-k}\Phi(v_1^k,\ldots,v_N^k) \tag{3}$$

と表される．ただし，関数Φは

$$\Phi(v_1^k,\ldots,v_N^k)=\sum_{i=1}^{N}a_iv_i^k \tag{4}$$

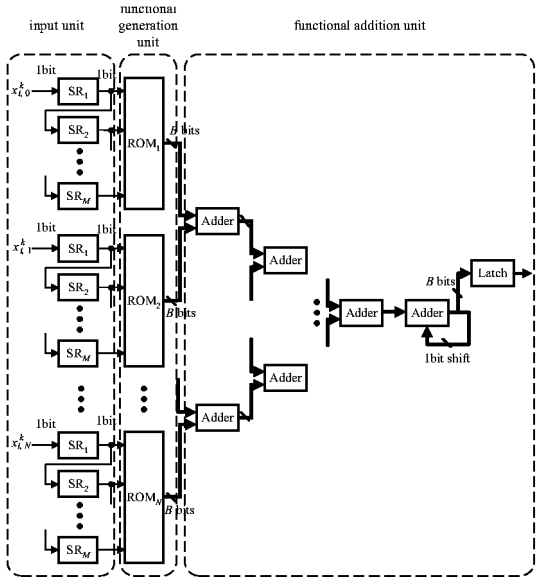


Fig. 2　2-D FIR filter based on distributed arithmetic

である． Fig. 1 に分散演算の基本構造を示す．この構成ではN個のシフトレジスタから$(v_1^k,\ldots,v_N^k)$を出力して，これをROMにアドレス値として入力する．ROMには入力の各ビットと係数との内積演算の結果すなわち式(4)の関数Φがテーブルとして書き込まれている．計算時にはそのテーブルの参照により得られた値を関数加算部で順次1ビットシフトしながら加算する処理を語長B回繰り返す．これは，項数Nが増加しても処理時間を一定の小さい値にでき，しかも乗算器を用いずに構成できるため，ハードウェア量を非常に小さくすることができる．

2.2　分散演算に基づく2次元FIRフィルタ

(M,N)次の2次元FIRフィルタの入出力関係は，

$$y(m,n)=\sum_{i=0}^{M}\sum_{j=0}^{N}h(i,j)x(m-i,n-j) \tag{5}$$

と表される．ここで，$h(i,j)$，$x(m-i,n-j)$と$y(m,n)$はそれぞれインパルス応答と入力と出力である．2次元FIRフィルタの実現法として，乗算器を用いた直接形構成に基づく構成がある[1]．これは高いサンプリングレートが得られるが，高次のフィルタ実現には$(M+1)\times(N+1)$個の乗算器を必要とするため，1次元フィルタでは考えれられない程の膨大なハードウェア量と消費電力を必要とする．そこで，分散演算の特長を利用したフィルタを実現する．

ここで，式(5)を

$$y(m,n) = \sum_{j=0}^{N} f(j) \tag{6}$$

と表す．なお，

$$f(j) = \sum_{i=0}^{M} h(i,j)x(m-i,n-j) \tag{7}$$

である．$x(m-i,n-j)$を$-1 \leq x(m-i,n-j) < 1$で，Bビットの固定小数点形の2の補数表示とすると，

$$\begin{aligned} x(m-i,n-j) &= x^0(m-i,n-j) \\ &+ \sum_{k=1}^{B-1} 2^{-k}x^k(m-i,n-j) \end{aligned} \tag{8}$$

と表される．ここで，$x^k(m-i,n-j)$は$x(m-i,n-j)$のkビット目の値で0または1である．式(8)を式(7)に代入すると，

$$\begin{aligned} f(j) &= -\Phi(x^0_{0,j},\ldots,x^0_{M,j}) \\ &+ \sum_{k=1}^{B-1} 2^{-k}\Phi(x^k_{0,j},\ldots,x^k_{M,j}) \end{aligned} \tag{9}$$

ただし，関数Φは

$$\Phi(x^k_{1,j},\ldots,v^k_{M,j}) = \sum_{i=0}^{M} h(i,j)x^k(m-i,n-j) \tag{10}$$

である．これより，式(10)より式(5)は

$$\begin{aligned} y(m,n) &= \sum_{j=0}^{N}\{-\Phi(x^0_{0,j},\ldots,x^0_{M,j}) \\ &+ \sum_{k=1}^{B-1} 2^{-k}\Phi(x^k_{0,j},\ldots,x^0_{M,j})\} \end{aligned} \tag{11}$$

と表される．式(10)(11)で示された分散演算を適用した(M,N)次のFIRフィルタの構成は Fig. 2 のように表される．なお，ここでは入力部と関数生成部と関数加算部に大別する．各ROMには$(M+1)$個の入力データ$\{x^k_{i,0},x^k_{i,1},\ldots,x^k_{i,N}\}$がビットシリアルに入力される．ROMには入力の各ビットと係数との内積演算の結果すなわち式(10)の関数Φがテーブルとして書き込まれている．計算時にはそのテーブルの参照により得られた値を関数加算部で順次1ビットシフトしながら加算する処理を語長B回繰り返す．これは，次数が増加してもサンプリングレートを保持しながら，ほぼ一定の小さな滞在時間に抑えることができる．さらに，乗算器を用いた直接形構成に基づく構成と比較して，ハードウェア量を大幅に抑えることができる．しかし，2次元FIRフィルタでは$N \times 2^M$ワードもの膨大なROM容量を必要とするため，1次元フィルタとは異なり比較的高い次数でも非常に大きな消費電力を必要とする．

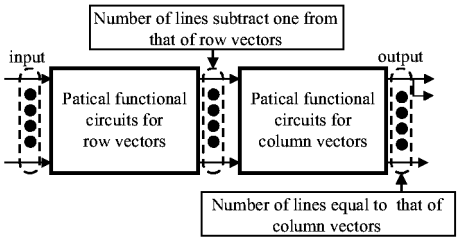


Fig. 3　Structure of optimum functional circuit

この問題を解決する方法として関数の分割化法があり，高次向き2次元フィルタを実現するためには必要不可欠である[3]．これは，大きなROM容量をいくつかの小さなROM容量の加算として実現する手法である．ここで分割数をQとすると，それを以下のように表せる．

$$\begin{aligned} &\Phi(x^k_{0,j},\ldots,x^k_{M,j}) \\ &= \sum_{i=1}^{M/Q} h(i,j)x^k(m-i,n-j) + \\ &\quad \ldots + \sum_{i=Q'}^{M} h(i,j)x^k(m-i,n-j) \\ &= \Phi(x^k_{0,j},\ldots,x^k_{M/Q,j}) + \\ &\quad \ldots + \Phi(x^k_{Q',j},\ldots,v^k_{M,j}) \end{aligned} \tag{12}$$

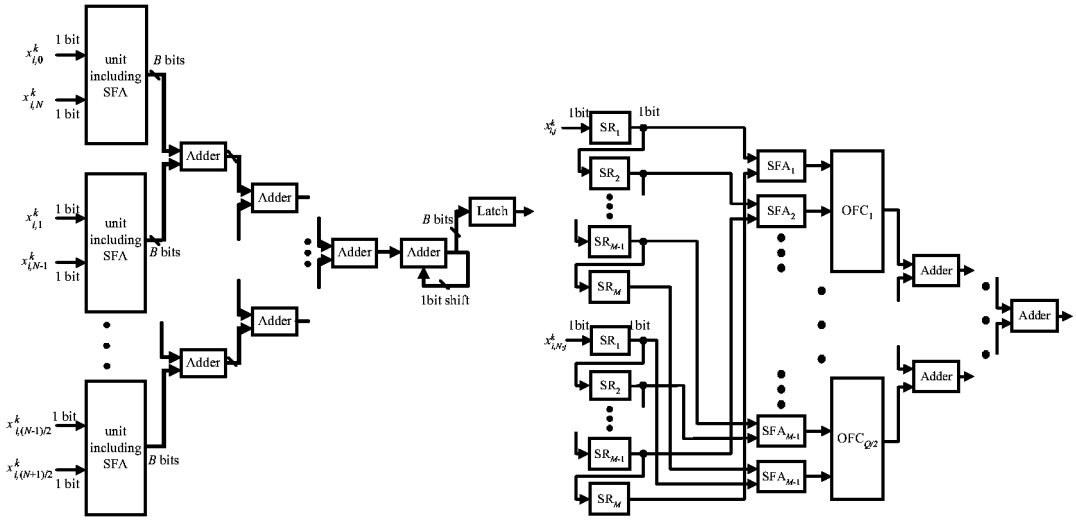
ただし，

$$Q' = \frac{(M+1)(Q-1)}{Q} \tag{13}$$

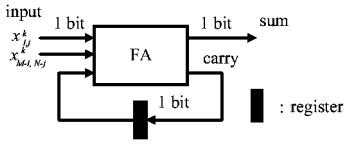
である．これにより，ROM容量を大幅に低減することができる．しかし，この手法を用いても，高次では比較的消費電力の大きいROMを多く用いるために，関数生成部の消費電力が非常に大きくなり，フィルタ全体の消費電力が増大する．

3.　最適関数回路

サンプリングレートを低減させずに関数生成部の消費電力を大幅に減少するために，ROMを用いた従来形の分散演算に基づく構成ではなく，我々が提案してきた最適関数回路を用いた構成に着目する[3]．ここで，アドレス線数lのROMに格納され



(a)Block diagram of FIR filter　　(b)Unit including SFA



(c)Structure of SFA

Fig. 4　Structure utilized linear-phase characteristic

ている関数Φのテーブルを

$$\boldsymbol{W}_{ROM} = \begin{bmatrix} w_0^{B-1} & \cdots & w_0^0 \\ \vdots & \cdots & \vdots \\ w_{2^l-1}^{B-1} & \cdots & w_{2^l-1}^0 \end{bmatrix} \tag{14}$$

という$2^l \times B$の行列で表す．w_i^jは0または1であり，式(14)の行ベクトル$[w_i^{B-1} \cdots w_i^0]$はi番地に格納されているBビットの出力データである．最適関数回路では，式(14)の行または列それぞれの同じベクトルを共有化した関数Φをテーブルとして用いる．この共有化を適用したテーブルを，

$$\boldsymbol{W}_{OPT} = \begin{bmatrix} w_0^j & \cdots & w_0^0 \\ \vdots & \cdots & \vdots \\ w_i^j & \cdots & w_i^0 \end{bmatrix} \tag{15}$$

という$i \times j$の行列で表す．ただし，

$$i \leqq 2^l - 1, \quad j \leqq B - 1 \tag{16}$$

である．最適関数回路は， Fig. 3 に示すように式(15)の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ある．これは論理ゲートで構成され，高次形2次元FIRフィルタにこれを適用すると消費電力を大幅に減少できる．

しかし，極めて高い次数のフィルタ実現に用いると，関数加算部において加算器数の増加による消費電力の増加が生じる．最適関数回路を用いた構成ではその部分の占める消費電力が大きいため，次数の増加に伴う関数加算部の消費電力の増加はフィルタ全体の消費電力に大きく影響を及ぼす．また，同時に加算段数が増加するために滞在時間も増加する．そこで，関数加算部に効果のある手法を新たに適用した構成を提案する．

4. 直線位相特性を利用したSFAに基づく構成

滞在時間を考慮した上で最適関数回路によって得られた非常に小さい消費電力をさらに減少するために，直線位相特性を有する2次元FIRフィルタのインパルス応答が点対称となる場合に着目する．ここで，MとNを奇数とおくと，この特性をもつフィルタのインパルス応答には，

$$h(m,n) = h(-m,-n) \tag{17}$$

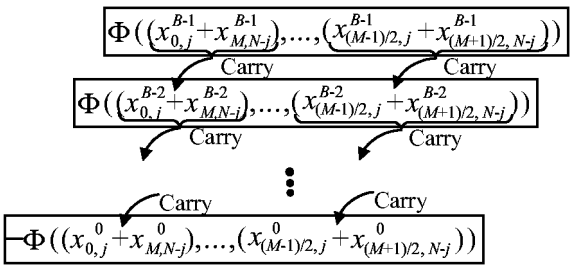
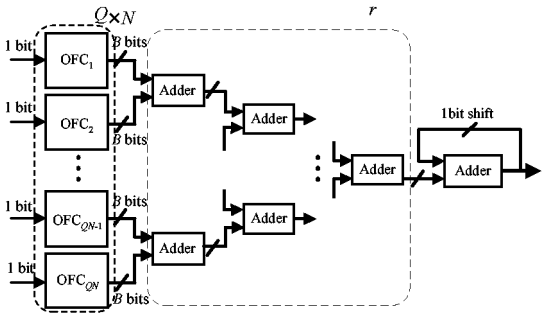
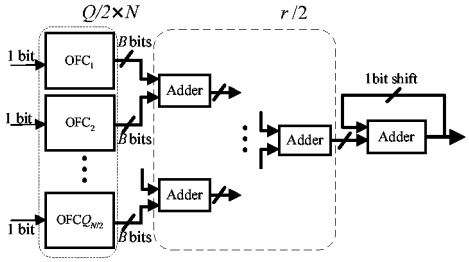


Fig. 5　Data flow of SFA



(a) Conventional structure



(b) Proposed structure using SFA.

Fig. 6　Structure of functional addition units.

という関係が成り立つ．式(17)を式(11)に代入すると，

$$\begin{aligned} y(m,n) &= \sum_{i=0}^{N}\{-\Phi((x^0_{0,j}+x^0_{M,N-j}), \\ &\quad \ldots(x^0_{(M-1)/2,j}+x^0_{(M+1)/2,N-j})) \\ &+ \sum_{k=1}^{B-1}\Phi((x^k_{0,j}+x^k_{M,N-j}), \\ &\quad \ldots,(x^k_{(M-1)/2,j}+x^k_{(M+1)/2+1,N-j}))\} \end{aligned} \tag{18}$$

と表すことができる．ただし，関数Φは

$$\begin{aligned} &\Phi((x^k_{0,j}+x^k_{M,N-j}), \\ &\quad \ldots,(x^k_{(M-1)/2,j}+x^k_{(M+1)/2,N-j})) \\ &= \sum_{i=0}^{(M-1)/2} h(i,j)\{x^k(m-i,j) \\ &\quad +x^k(m-M+i,n-M+j)\} \end{aligned} \tag{19}$$

である．ここで，式(18)と(19)で括られた二つの入力データの和を一つの入力データとして処理すると，項数を1/2に減少した演算とみなすことができる．これにより，関数Φのテーブルの大きさを1/2に低減することができる．しかも，関数の分割化を適用した本構成では加算器数を1/2に減少することも可能となる．しかし，括られた二つの入力データが2ビット長になる場合を考慮する必要がある．

この問題を解決できる手法として， Fig. 4 に示すような最適関数回路に基づく構成にSFAを適用した新たな構成を提案する．まず，式(18)の括られた$(M-1)/2$個のデータの最下位ビットの組$\{(x^{B-1}_{0,j},x^{B-1}_{M,N-j}),(x^{B-1}_{1,j},x^{B-1}_{M-1,N-j}),\cdots,(x^{B-1}_{(M-1)/2,j},x^{B-1}_{(M+1)/2,N-j}))\}$の加算を Fig. 4 (c)で示されるSFAのFA(Full Adder)で行う．この処理で生じる和とキャリーをそれぞれ最適関数回路とSFAのレジスタへ出力する．このレジスタに格納されたデータは， Fig. 5 に示すように一つ上位のビットの組$\{(x^{B-2}_{0,j},x^{B-2}_{M,N-j}),(x^{B-2}_{1,j},x^{B-2}_{M-1,N-j}),\cdots,(x^{B-2}_{(M-1)/2,j},x^{B-2}_{(M+1)/2,N-j}))\}$の加算時に入力データとして用いる．このような処理を最上位ビットすなわち語長B回繰り返す．ただし，オーバフローを防ぐために入力データを$-0.5 \leqq x_{i,j} < 0.5$に制限(Scaling)する．SFAの適用により，分割化を施した本実現法では， Fig. 6 に示すように最適関数回路の回路規模を1/2に低減できるだけでなく，関数加算部の加算器数rを$r/2$に低減することができる．本構成において，関数加算部が占める消費電力の割合が大きいため，SFAの適用による低消費電力化への効果が大きいといえる．また，同時に加算段数が1段減少できるために滞在時間も減少できる．このように，本手法は関数の分割化と最適関数回路とSFAの特徴を相互に利用して，滞在時間を考慮した低消費電力化を可能にした手法である．

5.　VLSI評価

本プロセッサをVLSI設計システムPARTHENONで設計する[5]．ここでは，実部品として用いるセルライブラリーの設計ルールを0.6μmCMOSスタンダードセルとし，電源電圧を5.0Vとする．また，ここでは標準偏差$\sigma = 2$のガウシアンフィルタを設計する．

まず，ROMを用いた分散演算に基づく従来形の

Table 1　VLSI evaluation of 2-D FIR filters (M=N=31)

	Proposed method using OFC and SFA	Proposed method using OFC	Conventional method using ROM	Method using multipliers for direct form
Power dissipation [W]	6.05	9.74	35.1	87.5
Area [mm^2]	13.9	22.1	45.8	450.0
Number of gates	124594	198871	271852	3897211
Word length [bit]	14	14	14	14
Machine cycle [ns]	16	16	16	48
Sampling rate [MHz]	4.46	4.46	4.46	20.8
Latency [ns]	368	384	384	336

OFC : Optimum Function Circuit

構成との比較を行う．これは，関数生成部で比較的消費電力の大きいROMを多く用いるため，35.1Wという大きな消費電力を必要とする．それに対して，本プロセッサでは関数の分割化と最適関数回路を用いて，関数生成部の大幅な低消費電力化を実現させた．これにより，ROMを用いた分散演算に基づく従来形のプロセッサの消費電力に対して約82.8%もの大幅な減少を可能にした．

さらに，消費電力と同時に滞在時間を減少するために，SFAを適用して関数加算部の加算段数を一段減少させ，その上加算器数を1/2に減少させた．SFAの適用により，最適関数回路のみを適用したプロセッサに対して，消費電力と滞在時間をそれぞれ約37.1%と約4.2%減少させた．

さらに，乗算器を用いた直接型構成に基づくプロセッサとの比較・評価を行う．Table 1より，乗算器を用いたプロセッサに対して，本プロセッサのゲート数と消費電力はそれぞれ約96.8%と約93.1%もの大幅な減少が可能となることがわかる．この比較対象のプロセッサは本プロセッサよりも高いサンプリングレートを得られるが，1024個もの乗算器を必要とするためにTable 1に示すように非常に膨大なハードウェア量と消費電力を必要とする．このことから，乗算器を用いた実現法ではサンプリングレートと滞在時間と消費電力を同時に考慮した高次の2次元FIRフィルタを実現するのは非常に困難であるといえる．

さらに次数が増加しても，処理時間が語長に依存する分散演算を利用した本プロセッサは，一定のサンプリングレートにした上で滞在時間をほぼ一定の小さい値に抑えることができる．以上のことから，本提案法が極めて高い次数の2次元FIRフィルタ実現において非常に有効な手法の一つであるといえる．

6.　むすび

本稿では，高性能2次元FIRフィルタのVLSIアーキテクチャを提案した．まず，処理時間が語長のみに依存する分散演算を用いて，次数の増加に対してサンプリングレートを保持しながらほぼ一定の小さな滞在時間に抑えた．しかし，ROMを用いた従来形の分散演算を適用した構成で非常に大きな消費電力を必要とする．そこで，我々が提案してきた最適関数回路を用いて，サンプリングレートを低減させずに消費電力を大幅に減少させた．しかし，これを極めて高い次数のフィルタ実現に用いると，加算器数と加算段数が増加するために消費電力と滞在時間が同時に増加してしまう．そこで，これらに効果のある手法として，最適関数回路を用いた構成にSFA(Serial Full Adder)を用いた直線位相特性に基づく手法を提案した．これにより，滞在時間を考慮しながら最適関数回路によって得られた非常に小さい消費電力をさらに減少させた．最後に，本プロセッサを(31,31)次の高い次数でVLSI評価した結果，本提案法が極めて高い次数のFIRフィルタ実現に有効な手法の一つであることを明らかにした．

参考文献

1) 西川　清：多次元FIRフィルタの設計，コンピュートロール，No.30，pp.26-37，コロナ社，1990.

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

3) 恒川, 野崎, 三浦: 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**-7/8, 1098/1107(1998)

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

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