計測自動制御学会東北支部 第228回研究集会 (2006.5.17)
資料番号 228-5

係数遅延と出力滞在時間が共に最小となるNCLMS適応ディジタルフィルタの高性能VLSIアーキテクチャ

A High-Performance VLSI Architecture for NCLMS Adaptive Digital Firters Minimizing Simultaneously Coefficient Delay and Latency

○佐藤 慎悟†, 菅野 大輔†, 高橋 強††, 恒川 佳隆†

○Shingo SATO†, Daisuke KANNO†, Kyo TAKAHASHI††, Yoshitaka TSUNEKAWA†

†岩手大学, ††岩手県工業技術センター

†Iwate University, ††Iwate Industrial Research Institute

キーワード： NCLMS(Non-canonical LMS), スケジューリング(scheduling), 係数遅延(Coefficient Delay), 出力滞在時間(latency), パイプラインアーキテクチャ(pipelined architecture)

連絡先： 〒020-8551 盛岡市上田4-3-5 岩手大学工学部
佐藤慎悟, Tel.:(019)621-6468, Fax.:(019)621-6468, E-mail: t3306014@iwate-u.ac.jp,

1. まえがき

現在, 適応ディジタルフィルタ(Adaptive Digital Filter,ADF)はエコーキャンセラ, アダプティブイコライザ, ノイズキャンセラなど広範囲に用いられている. さらに, 近年進展の目覚しいブロードバンド通信やマルチメディアなどの広帯域信号処理への応用や携帯電話, モバイルコンピューティングなどの小型・低消費電力が要求される携帯端末への応用も期待されている. これより, 適応ディジタルフィルタを実現する際には, 高速なサンプリングレート, 短い出力滞在時間(Latency), 高速な収束速度, 小規模なハードウエア, 低消費電力など実に多くの性能が要求される. しかし, これらの中には相反する性能も含まれるため, 要求される性能を同時に満たすことはきわめて困難であり, 高性能な適応アルゴリズムと効果的なアーキテクチャが望まれている.

これまで, LMSアルゴリズムのパイプライン処理向きのアルゴリズムであるDelayed LMS(DLMS)アルゴリズムが提案されている. これは, 係数更新に遅延を挿入してパイプライン処理を行うことにより, 高速なサンプリングレートを達成可能である. しかし, タップ数の増加に伴って収束速度が大きく劣化することが知られている. ここで, 係数更新に挿入される遅延を「更新遅延」と呼ぶことにする.

また, LMS適応ディジタルフィルタ(LMS Adaptive Digital Filter,LMS-ADF)の構造に対して転置型の構造を有するLMS適応ディジタルフィルタが提案されている. これはNCLMS適応ディジタルフィルタ(Non-canonical LMS Adaptive Digital Filter,NCLMS-ADF)と呼ばれており, フィルタモジュールの構成がタップ間隔のパイプライン構成となり出力滞在時間が非常に短い特長を有する. しかし, 一方でタップ数の増加に伴い収束

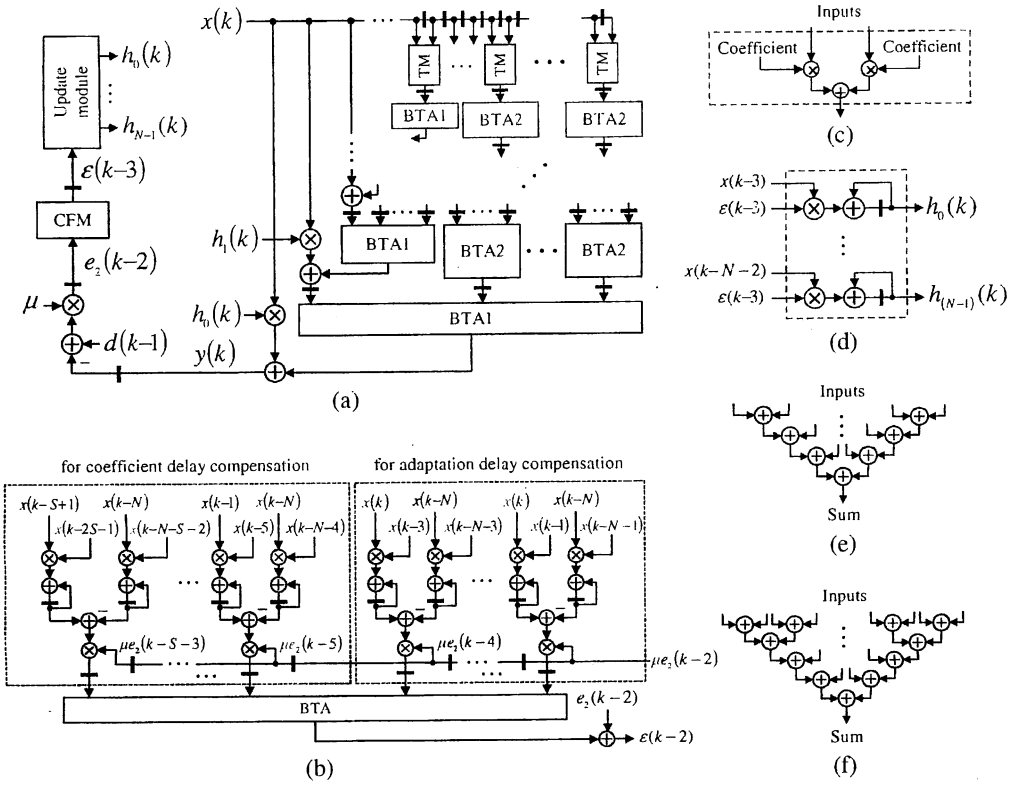

Fig. 1　従来法の適応フィルタの構成 (a)全体の構成，(b)補正値計算モジュール(CFM)，(c)タップモジュール(TM)，(d)係数更新モジュール，(e)BTA1，(f)BTA2

速度が大幅に劣化することも知られている．この原因は，出力計算に過去の係数を用いることによる推定誤差が発生するためである．なお，フィルタ出力の計算に過去の係数を用いることによる係数の遅延を「係数遅延」と呼ぶことにする．次いで，アップサンプリングとダウンサンプリングを用いたFIRフィルタの係数を分離して係数の遅延の最大値を大幅に減少させ，収束速度を改善する構成が提案されている[3]．

これに対して，我々は係数遅延と出力滞在時間が最小となる適応ディジタルフィルタの構成を提案してきた．この構成は従来のNCLMS-ADFの構成に基づいており，従来と同等の出力滞在時間を達成しつつ，同時に係数遅延を大幅に削減することで従来のNCLMSよりも高速な収束速度を達成できる高性能なアルゴリズムである．

本稿では，これまでの構成を従来法とし，この構成のさらなる高性能化を図る．まず，従来法のアルゴリズムレベルの構成に対して，加算器レベルのVLSIアーキテクチャを導出する．提案法の導出は構成に用いられている乗算器内部を含めた多入力加算器の構成に我々が提案している高速4入力2出力加算器を適用する．これにより，VLSI向きの規則性と係数遅延のさらなる減少を同時に実現する．また係数遅延の減少により，補正項生成に必要なハードウェア量を削減することが可能となる．同時に，演算器のスケジューリングを行うことでクリティカルパスを減少させ，更なる高速化を達成する．最後に，提案法のVLSIアーキテクチャを示しVLSI評価を行う．これにより提案法の有効性を明らかにする．

2. 従来法の構成

従来法の構成を 図 1 に示す．この構成はタップ出力の計算と並列に動作する多入力加算器を用いており，これにより大幅に係数遅延を減少できている．なお加算器，乗算器の滞在時間をそれぞれτ_{add}，τ_{mlt}とすると，従来法のピッチτ_jは

$$\tau_j = \tau_{add} + \tau_{mlt} \quad (1)$$

である．また，係数遅延の最大値が最小となる構成として最終ステージ以外には1タップを配置し，最終ステージに残りのタップを配置する構成を用いている．これは各ステージでタップ出力と並列に動作する多入力加算器の入力数を多くするためである．多入力加算器には加算段数の異なるバイナリツリーアダー(Binary Tree Adder,BTA)のBTA1とBTA2を用いており，これらの加算段数は以下のように決定される．まず，加算器と乗算器の滞在時間の比を次のように定義する．

$$m = \left\lfloor \frac{\tau_{mlt}}{\tau_{add}} \right\rfloor \quad (2)$$

ここで，$\lfloor a \rfloor$はa以下の最大の整数を表す．BTA1は乗算器と並列に動作するため，加算段数はτ_{mlt}以下の最大の段数となり，mとなる．また，BTA2はタップモジュールの計算の出力と並列に動作するため，加算段数は$\tau_{add}+\tau_{mlt}$以下の最大の段数となり，$m+1$となる．なお，mの値は使用する加算器と乗算器の方式や演算語長に依存する．よって，mの値により各ステージに配置される多入力加算器の段数は決定される．これより，全タップ数Nとmが決まるとステージ数は一意に決定され，係数遅延の最大値も一意に決定される．

3. 提案法の導出

本章では従来法のVLSI向きアーキテクチャを導出する．導出の際には語長は16ビットとし，遅延時間の評価値としてnand，nor，notゲートを1Δと定義した単位ゲート遅延を用いる[2]．

まず，加算器を桁上げ先見加算器(CLAA)，乗算器をBoothアルゴリズムとWallace木を用いた構成とする．Boothアルゴリズムはシフト，反転，判別を行うのみであるため，ハードウェア規模は小さくなる．そのため，乗算器は加算器の集合体として扱うことができ，構成の大部分を多入力加算器で構成することが可能となる．そこで，提案法で用いる多入力加算器について検討する．

3.1 多入力加算器の構成

多入力加算器の構成は一般的に桁上げ伝播加算器(CPA)やCLAAを用いたバイナリツリーアダー方式と桁上げ保存形式がある．前者の場合，桁上げ伝播が行われるため，語長の増加に対して速度が低下し高速化が難しくなる．これに対し桁上げ保存形式は桁上げ信号を伝播する必要がないため語長に依存しない高速な処理が可能となる．

桁上げ保存形式にもWallace木や冗長2進加算木，4入力2出力加算木などの構成法がある．Wallace木は，加算段数が$\log_{3/2} N$ (Nは入力数を表す)であり高速な処理が可能であるが，配線やレイアウトが複雑になりVLSIに適していない．冗長2進加算木は，規則的な構造でレイアウトが容易であり，加算段数は$\log_2 N$に比例し高速な処理が可能であるが，2進表現から冗長2進表現に変換するのに時間を要する．これに対して，4入力2出力加算木は規則的な構造でレイアウトが容易であり，加算段数は$\log_2 N$に比例し高速な処理が可能である[6]．以上より，本稿では多入力加算器に4入力2出力加算器(以下，4-2加算器とする)を用いて実現する．ここで，図 2 に我々が提案している高速4-2加算器の構成を示す．これは上位ビットへの桁上げを高速に出力できるType1 FAと，1つの入力を3Δだけ遅らせて入力できるType2 FAの 2 種類の全加算器を接続することで構成され

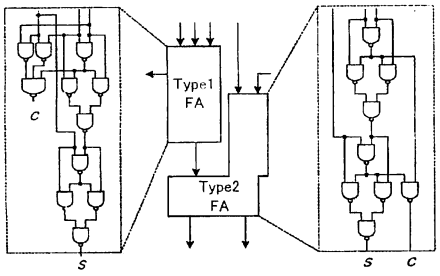

Fig. 2　高速4-2加算器の構成

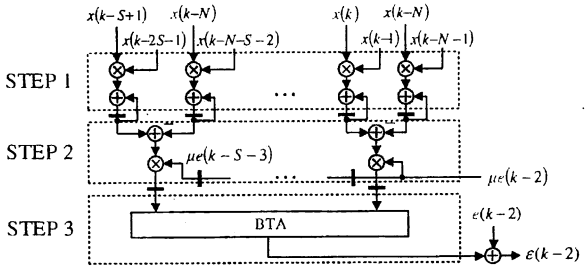

Fig. 3　補正値計算モジュール

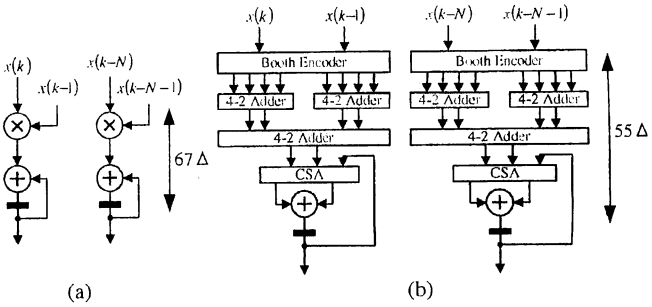

Fig. 4　STEP1の構成，(a)従来の構成，(b)提案法の構成

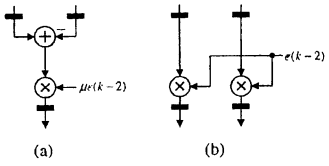

Fig. 5　STEP2の構成，(a)従来の構成，(b)提案法の構成

る．従来の4-2加算器が12Δなのに対し，この高速4-2加算器は9Δとなり3Δ高速である[1]．

次に，この高速4-2加算器を用いた提案法の構成について示す．従来法の構成はフィルタモジュール，誤差信号生成モジュール，補正値計算モジュール，係数更新モジュールの4つから構成されている．この構成に高速4-2加算器を用いた場合，VLSI向きの規則性と係数遅延や演算時間の削減を同時に実現することができる．本稿では，さらなる高速化を達成するために，より効果的な演算器のスケジューリングを行いクリティカルパスを短縮する．スケジューリングを行う際は比較的に自由度の低い補正値生成モジュールと係数更新モジュール，誤差信号生成モジュールから先に導出し，それらの構成に応じてフィルタモジュールの構成を導出する．

3.2　補正値生成モジュール

補正値計算モジュールの構成を 図 3 に示す．このように，補正値計算モジュールは3つの段階からなる構成と見なすことができる．ここでは，各段階ごとに構成を導出していく．

[STEP1の構成]

図 4 にSTEP1の構成を示す．ここでは入力信号同士の乗算とその値の累積加算を行っている．本構成では，乗算後に行う加算処理を乗算器内部で行う．これにより累積加算を行う加算器を削減できる．また演算時間は55Δとなり，従来法と比較して12Δの演算時間を短縮できる．

[STEP2の構成]

図 5 にSTEP2の構成を示す．ここでは，STEP1で求められた2個の累積加算値の減算を行い，さらに減算結果と誤差信号との乗算を行う構成となっている．本構成では，この部分でクリティカルパスとなるのを回避するため演算器のスケジューリングを行う．図 5 (b)に示すように，それぞれの累積加算値と誤差信号との乗算のみを行い，減算処理を次の段階に移行する．これにより，STEP2での演算時間を減少させることができる．

[1]nand, nor, notゲートを1Δと定義した単位ゲート遅延を用いたときの評価値である[2]．

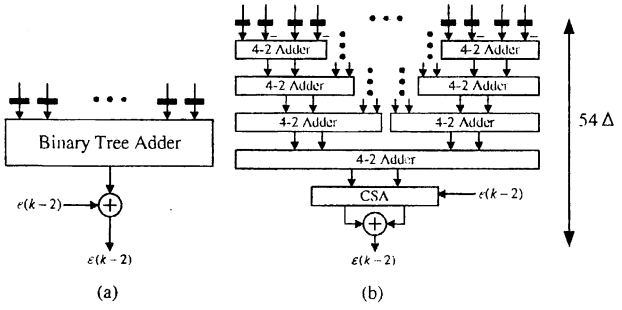

Fig. 6　STEP3の構成，(a)従来の構成，(b)提案法の構成

[STEP3の構成]

図6にSTEP3の構成を示す．ここでは，STEP2までに求められた値を多入力加算器で加算して補正された誤差信号$\epsilon(k-2)$を求める構成となっている．本構成では，まず1段目の4-2加算器でSTEP2からスケジューリングにより移行された減算処理を行う．2,3,4段目でこれらの加算処理を行い，5段目で補正対象の誤差信号との加算を行う．最後に6段目の加算器で桁上げ保存形式から通常の2進形式へと変換する．このSTEP3におけるゲート遅延は54Δであり，これまでの最大ゲート遅延55Δよりも小さい．

STEP3への入力数は語長16ビットの場合で32入力となる．本構成では1時刻の更新遅延もしくは係数遅延を補正するために2入力必要とするため，補正値計算モジュールで実現可能な遅延量は16となる．このうちパイプライン処理による更新遅延に3時刻を使用するため，実現可能なタップ数は係数遅延の最大値が13になるタップ数である．そのタップ数は極めて膨大な数になる．

3.3　係数更新モジュール

係数更新モジュールの構成を 図 7 に示す．ここでは係数更新値を求めるために入力信号と誤差信号の累積加算を行っている．本構成では，乗算後に行う加算処理を乗算器内部で行う．これにより累積加算を行う加算器を削減することができる．さらに，従来の構成と比較して12Δの演算時間を短縮することが可能である．

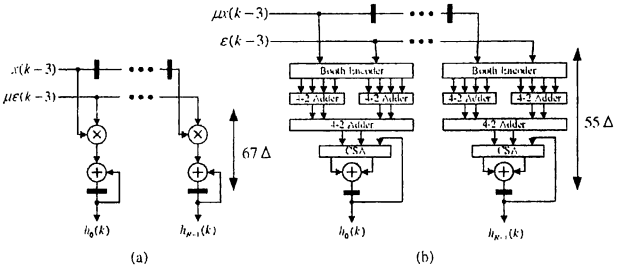

Fig. 7　係数更新モジュールの構成，(a)従来の構成，(b)提案法の構成

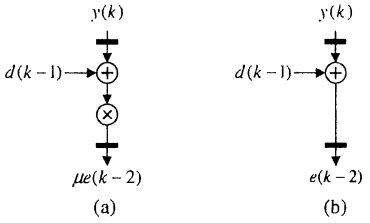

Fig. 8　誤差生成モジュールの構成，(a)従来法の構成，(b)提案法の構成

3.4　誤差信号生成モジュール

誤差信号生成モジュールの構成を 図 8 に示す．従来法ではこのモジュールで誤差信号を生成し，さらにステップサイズとの乗算を行っていた．しかし，この構成では誤差信号生成モジュールがクリティカルパスとなり，高速化の妨げとなる．そこで，本構成ではステップサイズと誤差信号との乗算に対してスケジューリングを行い，図 8 に示すように純粋な誤差信号のみを生成する構成とする．しかし，ステップサイズが乗算された誤差信号$\mu e(k)$が補正値生成モジュール及び係数更新モジュールにおいて必要とされる．そこで，以下のようにそれぞれ計算式に着目して乗算器のスケジューリングを行う．

[補正値生成式]

タップ数Nにおける適応ディジタルフィルタの誤差信号$\tilde{e}(k)$に付加される補正値$\Lambda(k)$は次式で与

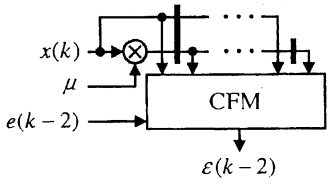

Fig. 9　スケジューリング後の補正値計算モジュールの構成

えられる．

$$\Lambda(k) = -2\mu\,[\hat{e}(k-4), \hat{e}(k-5), \ldots, \hat{e}(k-S-5)] \times \begin{bmatrix} \sum_{i=0}^{N-1} x(k-3-i)x(k-4-i) \\ \sum_{i=0}^{N-1} x(k-3-i)x(k-5-i) \\ \sum_{i=0}^{N-1} x(k-3-i)x(k-6-i) \\ \sum_{i=1}^{N-1} x(k-3-i)x(k-7-i) \\ \sum_{i=2}^{N-1} x(k-3-i)x(k-8-i) \\ \vdots \\ \sum_{i=S-1}^{N-1} x(k-3-i)x(k-S-5-i) \end{bmatrix} \tag{3}$$

ここで$x(k)$は入力信号であり，Sはフィルタのステージ数である．この式を変形すると，

$$\Lambda(k) = -\,[\hat{e}(k-4), \hat{e}(k-5), \ldots, \hat{e}(k-S-5)] \times \begin{bmatrix} \sum_{i=0}^{N-1} 2\mu x(k-3-i)x(k-4-i) \\ \sum_{i=0}^{N-1} 2\mu x(k-3-i)x(k-5-i) \\ \sum_{i=0}^{N-1} 2\mu x(k-3-i)x(k-6-i) \\ \sum_{i=1}^{N-1} 2\mu x(k-3-i)x(k-7-i) \\ \sum_{i=2}^{N-1} 2\mu x(k-3-i)x(k-8-i) \\ \vdots \\ \sum_{i=S-1}^{N-1} 2\mu x(k-3-i)x(k-S-5-i) \end{bmatrix} \tag{4}$$

となり，式 (4) は，ステップサイズがあらかじめ入力信号に乗算できることを示している．そこで提案法では，入力信号とステップサイズを乗算するようにスケジューリングを行う．図 9 にスケジューリングを行った場合の補正値計算モジュールの構成を示す．このモジュールへの入力は通常のFIRディジタルフィルタへの入力と同様に信号線に遅延器を配置した入力遅延線を介して行う．そして，入力信号$x(k)$が入力されたと同時にステップサイズとの乗算を行い，信号$\mu x(k)$を入力

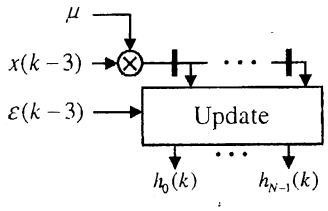

Fig. 10　スケジューリング後の係数更新モジュールの構成

遅延線で遅延させていく．これにより，ステップサイズが乗算された遅延信号を実現する．

[係数更新式]

係数更新式は以下のように表される．

$$\boldsymbol{H}(k+1) = \boldsymbol{H}(k) + 2\mu\epsilon(k-3)\,\boldsymbol{X}(k-3) \tag{5}$$

ここで，$\boldsymbol{H}(k)$，$\boldsymbol{X}(k)$はそれぞれタップ係数ベクトル，入力信号ベクトルを表し，

$$\boldsymbol{H}(k) = [h_0(k), h_1(k), \ldots, h_{N-1}(k)]^T$$
$$\boldsymbol{X}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T$$

である．ここで，式 (5) を次のように変形する．

$$\boldsymbol{H}(k+1) = \boldsymbol{H}(k) + \epsilon(k-3)\,\{2\mu\,\boldsymbol{X}(k-3)\} \tag{6}$$

式 (6) は，ステップサイズをあらかじめ入力信号に乗算できることを示している．そこで提案法では補正値計算モジュールと同様に入力遅延線を介して$\mu x(k)$の遅延信号を実現する．図 10 にその構成を示す．これより，係数更新モジュールに対して新たな入力遅延線が必要となるが，これは補正値計算モジュールの入力遅延線と共有可能である．

3.5　フィルタモジュール

図 11 に従来法のフィルタモジュールの構成を示す．各タップ出力の計算に着目すると，図 12 (a)に示す乗算器を用いており，入力信号とタップ係数を乗算して1つの積を得ている．し

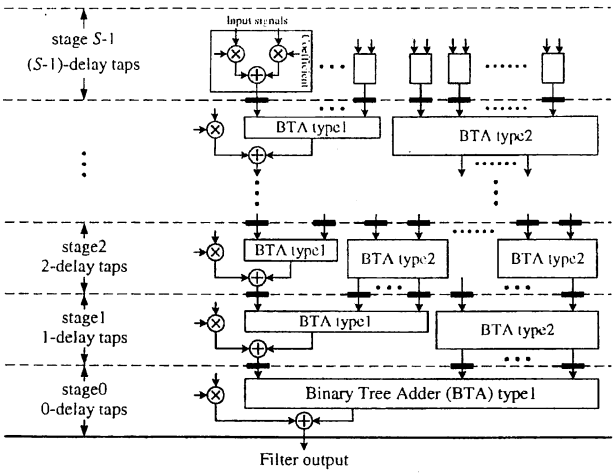

Fig. 11　従来法のフィルタモジュールの構成

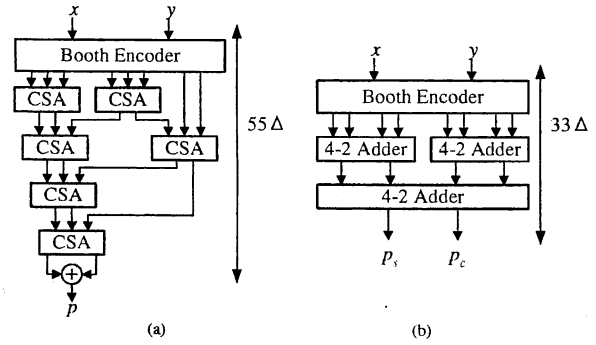

Fig. 12　乗算器の構成(16ビット)，(a)従来の乗算器の構成，(b)桁上げ保存形式で出力する乗算回路の構成

かし，最終的に要求される信号はフィルタ自体の出力であり，各タップ出力は必ずしも1出力である必要はない．そこで，各タップ出力に用いる乗算器に 図 12 (b)を用いる．これにより，乗算器内部に用いられる最終段の加算器を取り除くことができ，各ステージのクリティカルパスを大幅に短縮できる．乗算の演算時間の短縮は，回路の高速化や多入力加算器の段数増加による係数遅延の削減などを実現できる．さらに，フィルタ内部の加算器を高速4-2加算器のみで実現できるため，非常に規則的な構造を達成できる．

以下，提案法のフィルタモジュールを1ステージごとに導出していく．

[ステージ0の構成]

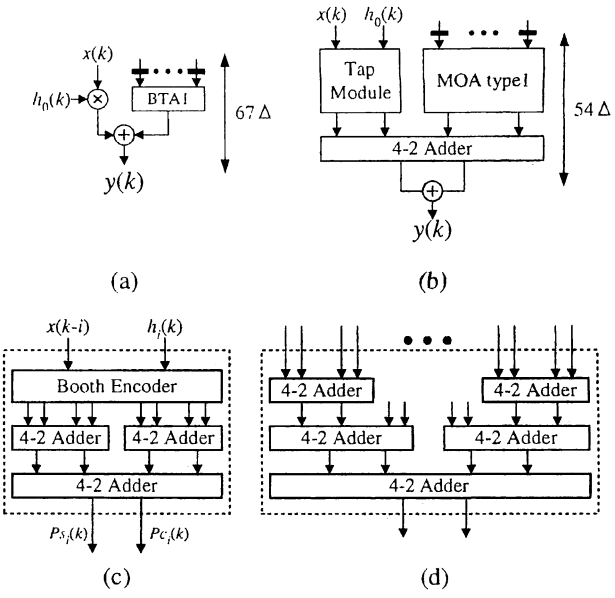

Fig. 13　ステージ0の構成，(a)従来法の構成，(b)提案法の構成，(c)タップ出力回路，(d)多入力加算器(Multi Operand Adder,MOA)タイプ1の構成

図 13 にステージ0の構成を示す．提案法では乗算器を桁上げ保存形で構成しており，多入力加算器タイプ1がこの乗算器と並列に動作する．ここで，多入力加算器タイプ1は3段の4-2加算木であり，その入力数は16である．そして，タップ出力と4-2加算木の出力を4-2加算器で加算し，CLAAで桁上げ保存形式から通常の2進数へと変換し，最終的なフィルタ出力を得る．

ステージ0の演算時間は54Δであり，これまで導出してきたモジュールのクリティカルパスである55Δより短い値である．

[ステージ1の構成]

図 14 にステージ1の構成を示す．ここでは，タップ出力回路を桁上げ保存形式の乗算器で構成し，ディレーアダーラインの多入力加算器タイプ2の出力と加算を行う．そして，残りの入力線に対してはタップ出力と並列に動作する多入力加算器タイプ3を挿入する．ここで，多入力加算器タイプ2は4段の4-2加算木，タイプ3は6段の4-2加算木であり，入力数はそれぞれ32，128である．

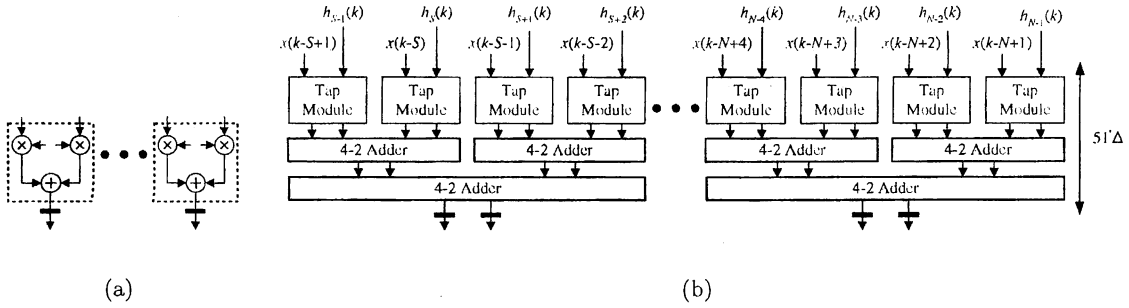

Fig. 15　最終ステージの構成，(a)従来法の構成，(b)提案法の構成

Table 1　ステージ数とタップ数の関係

ステージ数	文献3の構成	従来法の構成	提案法の構成
0	6	2	2
1	12	32	32
2	24	992	1,984
3	48	30,752	121,912
4	96	983,072	8,122,304

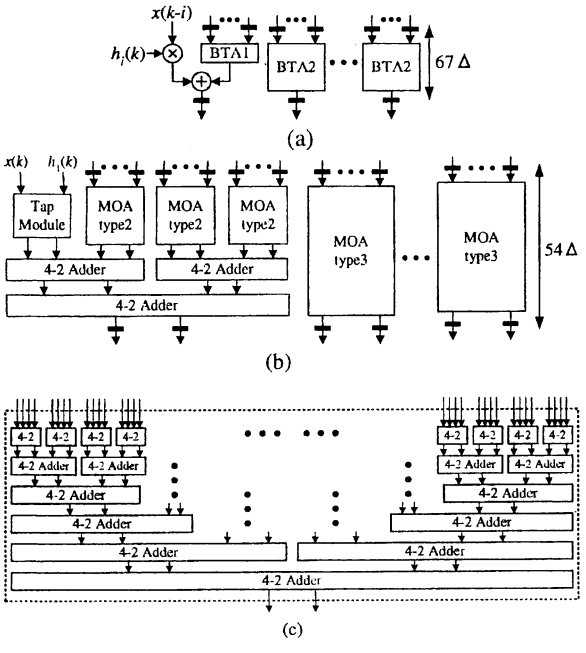

Fig. 14　ステージ1の構成，(a)従来法の構成，(b)提案法の構成，(c)多入力加算器の構成

以降，最終ステージまで同様の構成である．

[最終ステージ]

図 15 に最終ステージの構成を示す．このステージは4タップの出力の加算処理を行うモジュールであり，桁上げ保存形式で出力する．

以上の操作により，フィルタモジュールの構成が導かれる．表 1 に文献3に示されている係数遅延が極めて小さい構成，従来法の構成[1]と，今回提案したVLSI向き構成のステージとタップ数の関係を示す．これより，提案法の構成では従来法よりも係数遅延を抑えることが可能となる．

3.6　全体の構成

構成したすべてのモジュールを組み合わせることにより，図 16 の構成が得られる．この構成では，多入力加算器として4-2加算木を多数用いており，特にフィルタモジュールで非常に規則的な構造を達成している．そして，これまでの構成と比較して12Δのクリティカルパス削減を達成しているため，より高速な処理が可能となる．

4.　VLSI評価

提案する適応ディジタルフィルタのVLSIアーキテクチャをVLSI設計システムPARTHENON

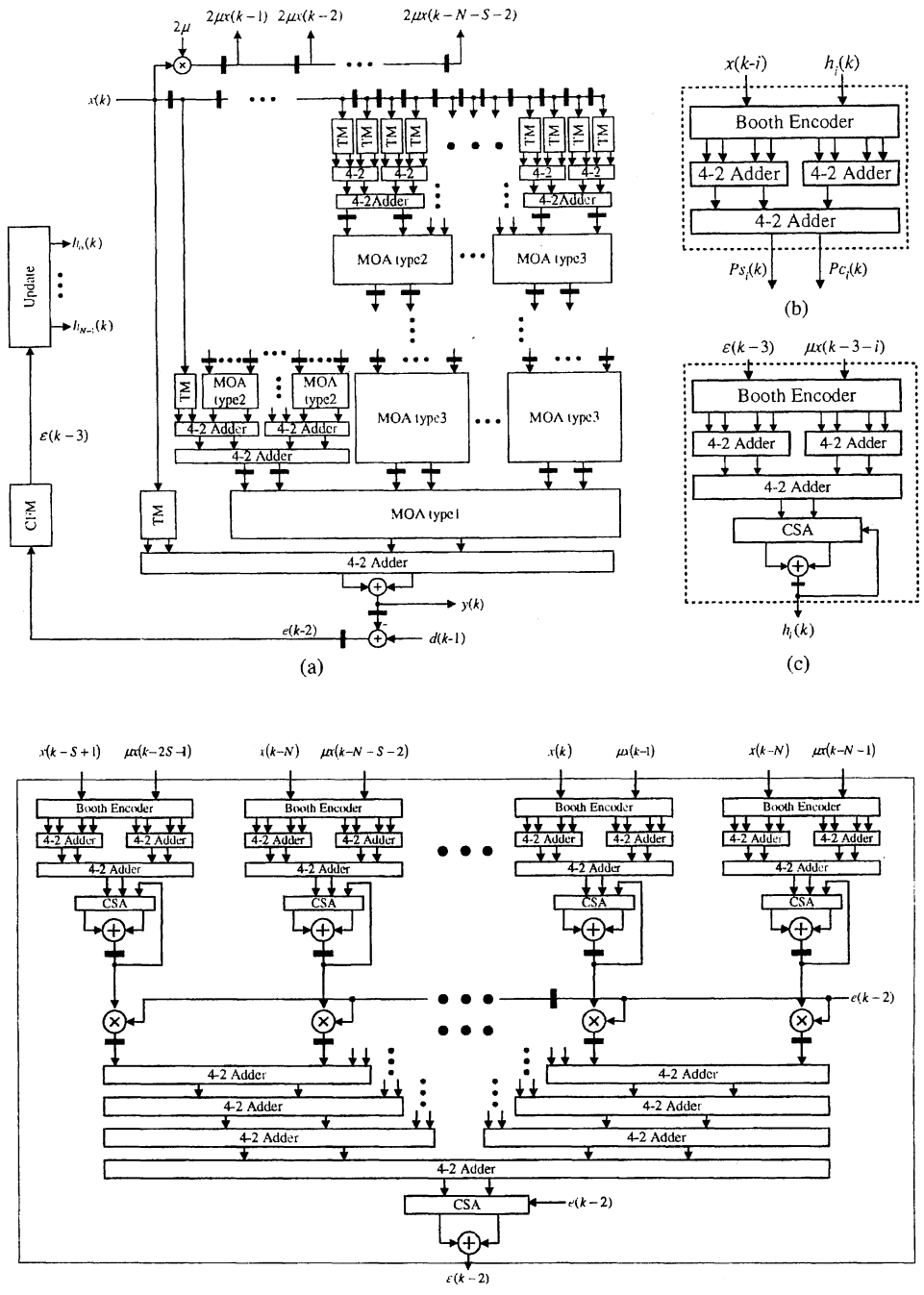

Fig. 16　提案法の構成，(a)全体の構成，(b)タップモジュール(TM)，(c)係数更新モジュール，(d)補正値計算モジュール

により設計および評価を行う．設計ルールは0.6μmCMOSスタンダードセルとし，電源電圧は5.0[V]とした．語長は16ビットとし，フィルタのタップ数は128とした．そして，比較対象として従来のNCLMSの構成，文献3の構成，従来法の構成[1]を用いた．その結果を表2に示す．

表より，提案法の構成はクリティカルパスを削減したことにより，文献3や従来法と比較して16.39[MHz]と高速なサンプリングレートを達成し，従来のNCLMSよりも短い出力滞在時間であ

Table 2 VLSI evaluations for 128-taps.

	従来のNCLMSの構成	文献3の構成	従来法の構成[1)]	提案法の構成
Machine cycle[ns]	195	95	72	61
Sampling rate[MHz]	5.13	9.62	13.89	16.39
Latency[ns]	65	112	72	61
Power dissipation[W]	7.02	6.63	9.01	9.10
Area[mm^2]	119.18	111.79	117.03	98.48
Number of gates	1,071,491	953,982	1,014,037	833,539

る61[ns]を達成できた．また，多入力加算器に4-2加算器を規則的に用いたことにより同時にハードウェア量も削減することができた．これにより，他の構成と比較してサンプリングレート1[MHz]当たりの消費電力を低くすることが可能となった．

以上から，提案法は高速性や短い出力滞在時間，小規模ハードウェア，低消費電力など要求される多くの性能を同時に満たすことができる．なお，本稿では語長が16ビットにおけるアーキテクチャの導出およびVLSI評価を行ったが，語長の増加に対しても本提案の手法で同様の性能を有するVLSIアーキテクチャの導出が可能である．

5. むすび

本稿では，係数遅延と出力滞在時間が最小となる適応ディジタルフィルタのVLSIアーキテクチャを提案した．提案法は，従来法のアルゴリズムレベルの構成をさらに高性能化するために加算器レベルでVLSI向きの構成を導出した．さらに，構成をスケジューリングすることでクリティカルパスを減少させ，高速なサンプリングレートを達成した．導出した構成は，特にフィルタモジュールにおいて極めて規則的な構造を実現している．さらに収束速度の劣化原因である係数遅延も削減することができた．係数遅延の減少は，補正項生成に必要なハードウェアも削減可能である．以上から提案法はタップ数に依存しない短い出力滞在時間，高速なサンプリングレート，良好な収束特性，低消費電力，小規模ハードウェアなどを同時に実現可能である．

参考文献

1) 高橋，菅野，恒川，“係数遅延と出力滞在時間が最小なFIRフィルタを用いた適応ディジタルフィルタの高性能パイプラインアーキテクチャ”，電子情報通信学会投稿中

2) Hwang K, “Computer Arithmetic/Principles, Architecture, and Design, ”John Wiley & Sons(1979)

3) J. Okello, H. Ochi, Y. Itoh, Y. Fukui and M. Kobayashi, “A New Architecture for Implementing Pipelined FIR ADF Based on Classification of Coefficients,” IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol.49, No.6, pp.418-426, June 2002.

4) H. T. Kung, “Why Systolic Architecture?, ”IEEE Computer, Vol.15,No.1,pp.37-46, 1982.

5) C. L. Seitz, “Concurrent VLSI Architecture,” IEEE Trans. On Computers, Vol.C-33, No.12, pp.1247-1265, 1984.

6) 高木直史，算術演算のVLSIアルゴリズム，コロナ社(2005)