高スループット形 FIR フィルタの 滞在時間最小化 VLSI アーキテクチャ

非会員 恒川佳隆 (岩手大)

非会員 野崎 剛 (岩手大)

正 員 三浦 守 (岩手大)

VLSI Architecture for FIR Filter with Very High Throughput Keeping Small Latency

Yoshitaka Tsunekawa, Non-member, Takeshi Nozaki, Non-member, Mamoru Miura, Member (Iwate University)

This paper presents VLSI architecture for FIR filter with very high throughput keeping small latency. In this architecture, a block implementation with characteristic of multirate system is applied, so that very high sampling rate becomes possible while processing speed of PE's is held constant. Moreover, by using distributed arithmetic, of which processing time depends on only word length and data of input and output are given serially, latency can be minimized and number of signal pins can be reduced to $2L$ for block length L . Next, we reveal a new method of lower power dissipation, by considering not only the properties of function Φ but also impulse response of even symmetry and odd symmetry. As a result we show that a 32-tap FIR digital filter can be realized with very high sampling rate of 95.2MHz and the small latency of 315ns, by using $0.6\mu\text{m}$ CMOS technology. In this case, the processor is implemented with relatively low power of 5.41W.

キーワード： FIR フィルタ, 高スループット, 滞在時間最小化, VLSI アーキテクチャ, 低消費電力

1. まえがき

最近, 計測・制御や通信など様々な分野でデジタルフィルタが用いられている。特に, デジタル伝送システムなど通信分野では, LSI 技術の進展を背景として, 高いスループットをもった FIR フィルタの実現法が提案されている。その1つとして, 直接形構成に基づく実現法が挙げられる⁽¹⁾。これは, 基本的に各タップの演算ごとにパイプライン処理を施すことによって高い処理能力を達成する実現法である。その結果, 高いスループットが得られる反面, 滞在時間(Latency)が大きくなってしまう。

ロボット制御やセンサ信号処理などのフィードバック制御系ではこれらを使用する場合, 滞在時間が非常に大きくなると安定性が損なわれる。また, 自律的動作を目指す知能ロボット制御においては, 外界環境の変化に瞬時に応答する必要があるため, 周波数特性ばかりでなく瞬時応答性が重要となる。このため, スループットばかりではなく滞在時間が大きな問題となる⁽²⁾⁽³⁾⁽⁴⁾。

それに対して, 滞在時間を1クロックすなわち1回の積和演算時間に抑えた高スループット形 FIR フィルタの実現法として, 転置形構成に基づく実現法が提案されている⁽⁵⁾。しかし, より高スループット化を図った場合には, 全てのタップに入力信号を同時に転送する共通バスの配線

遅延によって, パイプラインのピッチが制限される。また, 非常に多くのレジスタを用いて各タップ間のデータのタイミングを合わせる必要があるため, 滞在時間が大幅に増加する。そのため, 滞在時間を抑えた上でより高スループット化を実現することは困難である。

本論文では, 高スループット形 FIR フィルタの滞在時間最小化 VLSI アーキテクチャを提案する⁽⁶⁾。本提案法ではブロック実現のマルチレート性に着目したシステムアーキテクチャを用いている。これによって, プロセッサの動作速度を一定に抑えながら, 従来の高速化法では得られない高いスループットが得られる。しかし, この実現法では高スループット化に伴い処理要素(Processing Element, 以下 PE)数が増加するため, システム全体の消費電力と信号線数が増大してしまう。また, 高次のフィルタリングを行う場合, 内積演算の項数の増加による滞在時間の増加が考えられる。そこで PE に対して, 我々が提案してきた滞在時間を考慮した高次 FIR フィルタの高速・低消費形 VLSI アーキテクチャを適用する⁽⁷⁾。これは分散演算(Distributed Arithmetic)の特長に着目した手法であり, 滞在時間の最小化とシリアル入出力が可能となる。また, 高スループット化に伴う消費電力の増大を抑えるため, 分散演算の関数の生成に ROM を用いた一般的な手法ではなく, 論理ゲートによる最適関数回路を用いた低消費電力化を図っている。

図1 システム構成
Fig.1 Structure of system.

さらに、本提案法ではインパルス応答の偶対称性ばかりでなく奇対称性にも着目することによって、より一層の低消費電力化が実現可能となることを示す。

2. ブロック実現

N タップ FIR フィルタの入出力関係は次式のように示される。

$$y(k) = \mathbf{A}\mathbf{X}(k) \quad (1)$$

ここで、 $y(k)$ は出力を表すスカラーである。また、入力ベクトル $\mathbf{X}(k)$ および係数ベクトル $\mathbf{A}$ は

$$\mathbf{A} = [a_0, a_1, \dots, a_{N-1}] \quad (2)$$

$$\mathbf{X}(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T \quad (3)$$

と表される。

これまでいくつかの高スループット形 FIR フィルタの実現法が提案されている⁽¹⁾⁽⁵⁾。これらの実現法はクロックレートの向上に基づいた高速化法であるが、このレートの向上には限界があるため、ある一定以上の高速化を図ることは困難である。また、従来の FIR フィルタの実現法の1つである直接形構成に基づく実現法は高スループット化に伴い滞在時間が増大してしまう。一方、転置形構成に基づく実現法では、基本的に滞在時間は1回の積和演算時間であるが、さらに高スループット化を図る場合には共通パスの配線遅延という問題が生じる。また、タップ間のデータのタイミングを合わせるために必要なレジスタ数が非常に多くなり、それに伴い滞在時間が大幅に増加するという問題も生じる。そのため、滞在時間を抑えた上でこれまで以上に高いスループットを得るためには、従来の実現法とは異なる新たな実現法が必要となる。そこで、我々はブロック実現⁽⁶⁾に着目して高速化を図る。

ここで、 $(L+N-1)$ 次の入力ベクトル $\hat{\mathbf{X}}(k)$ および L 次出力ベクトル $\hat{\mathbf{Y}}(k)$ を

$$\hat{\mathbf{X}}(k) = [x(kL), x(kL-1), \dots, x(kL-L-N+2)]^T \quad (4)$$

$$\hat{\mathbf{Y}}(k) = [y(kL), y(kL-1), \dots, y(kL-L+1)]^T \quad (5)$$

と定義する。このとき、ブロック実現は、

$$\hat{\mathbf{Y}}(k) = \hat{\mathbf{A}}\hat{\mathbf{X}}(k) \quad (6)$$

と表される。ここで、 $\hat{\mathbf{A}}$ は、

図2 マルチレートシステム

Fig.2 Multirate system.

$$\hat{\mathbf{A}} = \begin{bmatrix} a_0 & a_1 & \dots & a_{N-1} & 0 & \dots & 0 \\ 0 & a_0 & \dots & a_{N-2} & a_{N-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_0 & a_1 & \dots & a_{N-1} \end{bmatrix} \quad (7)$$

と表される。式(7)は、 $(L+N-1) \times L$ の Toeplitz 行列である。なお、 L はブロック長と呼ばれる。

3. ブロック実現に基づくシステムアーキテクチャ

式(6)を各要素に分けて、

$$y(kL) = [a_0, \dots, a_{N-1}]$$

$$[x(kL), \dots, x(kL-N+1)]^T$$

$$y(kL-1) = [a_0, \dots, a_{N-1}]$$

$$[x(kL-1), \dots, x(kL-N)]^T$$

$\vdots$

$$y(kL-L+1) = [a_0, \dots, a_{N-1}]$$

$$[x(kL-L+1), \dots, x(kL-L-N+2)]^T$$

(8)

と記述する。これより、 $\hat{\mathbf{Y}}(k)$ の各要素は、式(1)の係数ベクトルと変数ベクトルとの内積であることがわかる。そこで、式(1)を L 回繰り返して出力 $\hat{\mathbf{Y}}(k)$ を求めるのではなく、式(8)の各要素を同時並列処理することによって、処理時間を減少させる。この同時並列処理を用いたシステムの構成を図1に示す。図1のシステムでは、 $\hat{\mathbf{Y}}(k)$ の各要素を各 PE に割り当てる。また、入力部は、出力 $\hat{\mathbf{Y}}(k)$ を求めるのに必要な入力 $\hat{\mathbf{X}}(k)$ を、各 PE に転送する機能をもつ。この構成法では、 L サンプル同時に出力を求めるため、入力部において L 個の入力値 $x(kL-L+1) \sim x(kL)$ が必要となる。

このときのシステム全体の構成は図2として表すことができる。図2に示すシステムの L 個の入力時系列 $x(kL-L+1) \sim x(kL)$ は、入力時系列 $x(k)$ を L 個おきに間引いたものである。この並びかえられた L 個の入力時系列をシステムの入力として、それぞれに対してフィルタリングを実行し、 L 個の出力時系列 $y(kL-L+1) \sim y(kL)$ を同時に求める。その結果、システム全体の処理は元の入力時系列の1出力当たりの処理時間すなわちサンプリング周期 T に対して LT という低いレートで実行可能となる。このことから、ブロック実現に基づくシステムはマルチレート性を有することがわかる。

図3 分散演算に基づく FIR フィルタの構成
Fig.3 Structure of FIR filter using distributed arithmetic.

このシステムにおける PE の処理時間は、ブロック長 L に依存せずに実現できる。そこで、このシステムに対して入力部がブロック長に依存しないと仮定した場合、システム全体のサンプリング周波数 F は PE のサンプリング周波数 F_p を用いて

$$F = L \cdot F_p \quad (9)$$

と表せる。式(9)からブロック長 L に比例した高いサンプリング周波数が得られることがわかる。しかし、これに対応して PE 数が増加するため、消費電力が増大する。ゆえに、消費電力を考慮した高いスループットを実現するためには、PE の低消費電力化が必要不可欠である。

さらに、ブロック長が増加するにしたがって信号線数が増大するという問題も生じる。特に、入力信号がパラレルに転送される場合、信号線数が大幅に増加してしまう。そこで、信号線数を極力抑えるためには、ビットシリアル転送を可能にさせる構成法を用いる必要がある。

また、滞在時間 S は PE の滞在時間 S_p と入力部の滞在時間 S_i を用いて、

$$S = S_p + S_i \quad (10)$$

と表せる。なお、ここでは滞在時間のあるサンプルが入力されてからそのサンプルに対応した演算結果が得られるまでの時間と定義する。次数が増加した場合、PE で実行される内積演算の項数が増加するため、PE の滞在時間 S_p の増加を考慮する必要がある。また、ブロック長が増加した場合、PE 数が増加するため、入力部の出力数が増加する。そのため、入力部の滞在時間 S_i の増加を考慮する必要がある。これより、滞在時間を最小化するためには、基本的に次数に依存しない S_p とブロック長に依存しない S_i を実現できる構成法が必要である。

4. 処理要素の低消費電力化

本システムの消費電力、信号線数および滞在時間を抑えるためには、前章で述べた問題を考慮する必要がある。そこで PE に対して、我々がこれまで提案してきた滞在時間を考慮した高次 FIR フィルタの高速・低消費形 VLSI アーキテクチャを適用する⁽⁷⁾。まず、このアーキテクチャで適用した分散演算によって滞在時間の最小化が図れ、その上

信号線数を大幅に抑えることができることを示す。また、最適関数回路の性質および関数の分割化によって、低消費電力化が実現可能となることを示す。さらに、本章では、インパルス応答の偶対称性だけではなく奇対称性も利用したペ어링テーブル法を提案し、より一層の低消費電力化が可能となることを示す。

<4.1> 分散演算に基づく FIR フィルタ実現

一般的な分散演算は、定係数の内積演算を ROM のテーブルルックアップによって実現する計算手法である⁽⁹⁾⁽¹⁰⁾⁽¹¹⁾。ここでは FIR フィルタの入出力関係が式(1)のように変数と定係数との内積演算であることに着目して、FIR フィルタに分散演算の適用を考える。

N タップ FIR フィルタの入出力関係は式(1)を基にして、次式のように表す。

$$y(k) = \sum_{i=0}^{N-1} a_i x_{k-i} \quad (11)$$

ただし、 $-1 \leq x_{k-i} < 1$ で、 x_{k-i} は B ビットの固定小数点形の 2 の補数表示である。つまり、

$$x_{k-i} = -x_{k-i}^0 + \sum_{j=1}^{B-1} 2^{-j} x_{k-i}^j \quad (12)$$

と表される。ここで、 x_{k-i}^j は x_{k-i} の j ビット目の値で 0 または 1 である。式(12)を式(11)に代入すれば、内積演算 $y(k)$ は次式で示される。

$$y(k) = -\Phi(x_k^0, \dots, x_{k-(N-1)}^0) + \sum_{j=1}^{B-1} 2^{-j} \Phi(x_k^j, \dots, x_{k-(N-1)}^j) \quad (13)$$

ただし、 Φ は

$$\Phi(x_k^j, \dots, x_{k-(N-1)}^j) = \sum_{i=0}^{N-1} a_i x_{k-i}^j \quad (14)$$

である。式(13)は、 $(x_k^j, \dots, x_{k-(N-1)}^j)$ をアドレスとして、入力変数の各ビットと係数との内積演算の結果 Φ をテーブルとして書き込んでおく。そして、計算時にはテーブルを参照して得られた値を順次 1 ビット右シフトしながら、加え合わせる操作を行うことによって構成できる。図3に分散演算に基づく FIR フィルタの構成を示す。なお、図3は関数 Φ の生成に ROM を用いた一般的な構成である。この計算手法では、内積演算が乗算器を用いずに、語長 B 回分のシフトと加算で計算できる。

図3に示す構成の滞在時間 S_p は、語長 B とパイプラインの段数 D とピッチ τ を用いて

$$S_p = (B + D)\tau \quad (15)$$

と表せる。なお、パイプラインの段数 D は、次数の増減に対してほぼ一定である。そのため、 S_p も次数の増減に対してほぼ一定の値となる⁽⁷⁾。

また、図3に示す入力部の構成は、ビットシリアルにデータを連続で入力させることができる。ゆえに、データが各ステージにおいてピッチ τ で連続に処理することが可能となり、入力値を処理時間 $B\tau$ という次数に依存しない一

図4 関数 Φ の最適化
Fig.4 Optimization of function Φ .

定の時間で処理できる。つまり、PE のサンプリング周期は $B\tau$ であり、次数に依存しない値となる⁽⁷⁾。

さらに、入力信号をビットシリアルに転送できるため、パラレル転送の構成に対して信号線を大幅に抑えることができる。

<4.2> 最適関数回路

分散演算に基づく FIR フィルタは、関数 Φ の生成に ROM を用いた場合、非常に大きな消費電力を必要とする。これは、ROM の消費電力が非常に大きくなるためである。そこで、関数 Φ の生成を 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回路を構成することによって低消費電力化を図る。この手法を関数 Φ の最適化と呼ぶことにする。また、このとき生成される回路を最適関数回路、ROM のアドレス線に相当するものを最適関数選択線と呼ぶことにする。関数 Φ の最適化とは、図 4 のように関数 Φ の冗長性を削除する手法である。以下にその手法を示す⁽⁷⁾。

ROM に書き込まれている関数テーブル Φ は

$$W_1 = \Phi(x_k^j, \dots, x_{k-(N-1)}^j) = \begin{bmatrix} w_0^{B-1} & \dots & w_0^0 \\ \vdots & \dots & \vdots \\ w_{2^N-1}^{B-1} & \dots & w_{2^N-1}^0 \end{bmatrix} \quad (16)$$

として表せる。ここで、 $x_k^j \dots x_{k-(N-1)}^j$ は N ビット長の入力

図5 最適関数回路
Fig.5 Optimal functional circuit.

変数、 $w_i^{B-1} \dots w_i^0$ は i 番地に格納されている B ビットのデータである。また、 W_1 は $2^N \times B$ の行列である。本手法では、まず図 4(a)(b)のように式(16)の行ベクトルに対して同一な行ベクトルすなわち共有可能となる行ベクトルを括る。その結果、生成される行列 W_2 は、

$$W_2 = \begin{bmatrix} w_{hc1}^{B-1} & \dots & w_{hc1}^0 \\ \vdots & \dots & \vdots \\ w_{hc1}^{B-1} & \dots & w_{hc1}^0 \end{bmatrix} \quad (17)$$

となる。さらに、図 4(b)(c)のように式(17)の列ベクトルに対しても同様な手法を施す。その結果、生成される行列 W_3 は、

$$W_3 = \begin{bmatrix} w_{hc1}^{vc1} & \dots & w_{hc1}^{vcj} \\ \vdots & \dots & \vdots \\ w_{hc1}^{vc1} & \dots & w_{hc1}^{vcj} \end{bmatrix} \quad (18)$$

となる。ここで、式(18)で表される W_3 の相異なる各行ベクトルを横のカテゴリと呼ぶことにする。同様に、 W_3 の相異なる各列ベクトルを縦のカテゴリと呼ぶことにする。

この冗長性の削除を基にして、最適関数回路は図 5 に示す横のカテゴリに対する部分機能回路と縦のカテゴリに対する部分機能回路を縦続接続した回路構成になる。図 5 の各部分機能回路には、二段論理簡単化後に多段論理最適化を施す⁽¹²⁾⁽¹³⁾。これらの冗長性の削除によって、最適関数回路のゲート数およびファンイン数が減少する。

最適関数回路は、横および縦のカテゴリ数が減少するにしたがって、消費電力が減少する性質をもつ⁽⁷⁾。また、全てのビットパターンが 0 となるカテゴリの数が多いほどゲート数が減少し、その結果低消費電力となる性質もある。さらに、FIR フィルタの係数値が全体としては小さな値をもつため、内積演算の結果に対する上位ビットは 1 つの縦のカテゴリとして括ることができる。そのため、FIR フィルタの係数を用いた関数 Φ のテーブルは、最適関数回路の消費電力を抑える性質をもつ⁽⁷⁾。

<4.3> 関数の分割化

最適関数回路を用いた構成は、次数が増加するほど、縦および横のカテゴリ数が増加する。そのため、膨大なゲート数を必要とする。そこで、ゲート数を大幅に減少する手法として、関数 Φ の分割化法が有効となる。これは、関数 Φ の分割数を Q とした場合、以下の処理を施す。

図6 ペアリングテーブル法に基づく構成($N=8$, $Q=2$ の場合)

Fig.6 Structure using paring-table method in case of $N=8$ and $Q=2$.

$$\begin{aligned}
 W_1 &= \Phi(x_k^j, \dots, x_{k-(N-1)}^j) \\
 &= \sum_{i=0}^{N-Q} a_i x_{k-i}^j + \dots + \sum_{i=Q}^{N-1} a_i x_{k-i}^j \\
 &= \Phi(x_k^j, \dots, x_{k-\frac{N-Q}{Q}}^j) + \dots + \Phi(x_{k-Q}^j, \dots, x_{k-N+1}^j) \\
 &= W_1^1 + W_1^2 + \dots + W_1^Q
 \end{aligned} \quad (19)$$

ここで

$$Q' = \{(Q-1)N\} / Q \quad (20)$$

$$W_3^1 = \Phi(x_{k-((l-1)N)/Q}^j, \dots, x_{k-(N-Q)/Q}^j) \quad (21)$$

である。これによって、ゲート数の大きな最適関数回路をいくつかの小さなゲート数をもつ最適関数回路の加算として実現できる。この結果、消費電力が大幅に減少できる。

関数 Φ の分割数は本実現法を用いたシステムの性能を大きく左右するため、最適な分割数を算出する必要がある。そこで、最適な分割数について考察する。まず、関数 Φ の分割数が増加した場合、最適関数選択線が減少するため、この回路における横のカテゴリ数が減少する。これによって、この回路のゲート数は減少する。しかし、このとき加算器数が増加するために、関数 Φ の加算部のゲート数は増加する。一方、分割数が減少した場合、最適関数回路のゲート数は増加し、関数 Φ の加算部では減少する。これより、消費電力を最小にする最適な分割数が存在し、それは加算器と最適関数回路のゲート数の総和が最小となる値といえる。

<4.4> ペアリングテーブル法

FIR フィルタは完全な直線位相特性を実現できる。この直線位相特性によって形成されるインパルス応答すなわちフィルタ係数は、偶対称と奇対称の2通りある。我々はこれまで偶対称性に着目したペアリングテーブル法を提

表1 ペアリングテーブル法を用いた関数 Φ テーブル($N=8$, $N/Q=4$ の場合)

Table 1. Table of function Φ using paring-table method in case of $N=8$ and $N/Q=4$.

$v_0^k, v_7^k, v_1^k, v_6^k$	$\Phi(v_0^k, v_7^k, v_1^k, v_6^k)$	
	Even symmetry	Odd symmetry
0000	$(0)_2$	$(0)_2$
0001	$(a_1)_2$	$(-a_1)_2$
0010	$(a_1)_2$	$(a_1)_2$
0011	$(2a_1)_2$	$(0)_2$
0100	$(a_0)_2$	$(-a_0)_2$
0101	$(a_0 + a_1)_2$	$(-a_0 - a_1)_2$
0110	$(a_0 + a_1)_2$	$(-a_0 + a_1)_2$
0111	$(a_0 + 2a_1)_2$	$(-a_0)_2$
1000	$(a_0)_2$	$(a_0)_2$
1001	$(a_0 + a_1)_2$	$(a_0 - a_1)_2$
1010	$(a_0 + a_1)_2$	$(a_0 + a_1)_2$
1011	$(a_0 + 2a_1)_2$	$(a_0)_2$
1100	$(2a_0)_2$	$(0)_2$
1101	$(2a_0 + a_1)_2$	$(-a_1)_2$
1110	$(2a_0 + a_1)_2$	$(a_1)_2$
1111	$(2a_0 + 2a_1)_2$	$(0)_2$

$(\cdot)_2$ denotes binary representation.

案してきた⁽⁷⁾。これは、同じ係数を組み合わせることによって、横のカテゴリ数を減少させる手法である。ここでは、さらに奇対称性に着目した新たなペアリングテーブル法も提案する。この手法に基づく構成を図6に示し、インパルス応答の偶対称性および奇対称性に着目した低消費電力化法を述べる。

フィルタ係数が偶対称の場合、図6の最適関数回路 OFC_1 に書き込まれている関数 Φ は、

$$\begin{aligned}
 \Phi(v_0^k, v_7^k, v_1^k, v_6^k) &= a_0 v_0^k + a_7 v_7^k + a_1 v_1^k + a_6 v_6^k \\
 a_0 &= a_7 \\
 a_1 &= a_6
 \end{aligned} \quad (22)$$

と表せる。また、フィルタ係数が奇対称の場合は、

$$\begin{aligned}
 \Phi(v_0^k, v_7^k, v_1^k, v_6^k) &= a_0 v_0^k + a_7 v_7^k + a_1 v_1^k + a_6 v_6^k \\
 a_0 &= -a_7 \\
 a_1 &= -a_6
 \end{aligned} \quad (23)$$

と表せる。式(22), (23)より、 OFC_1 に書き込まれている関数 Φ は表1となる。なお、表1に示す関数 Φ のパターン数は横のカテゴリ数に一致する。本手法を用いない場合、横のカテゴリ数は基本的に16である。それに対して、本手法を用いた場合、横のカテゴリ数が基本的に9まで減少できる。しかも、係数が奇対称の場合、0という横のカテゴリ値がいくつか生成される。ゆえに、横のカテゴリ数に着目した低消費電力化を考えた場合、奇対称性に着目したペアリングテーブル法がより有効である。

ここで、最適関数選択線数が m の場合について示す。このとき、本構成法を用いない場合、横のカテゴリ数は基本

図7 入力部の構成 ($L=3$, $N=3$ の場合)
Fig.7 Structure of input units in case of $L=3$ and $N=3$.

的に 2^m となる。一方、本構成法を用いた場合、基本的に $3^{m/2}$ となる。その上、0 という値の横のカテゴリの数が

$$\sum_{k=0}^{m/2} m/2 C_k \quad (24)$$

だけ生成される。ここで、

$$C_j = \frac{i!}{j!(i-j)!} \quad (25)$$

である。

5. 入力部の構成

PE に分散演算を用いたことによって、入力部の $(L+N-1)$ 個の出力値を各 PE に N 個ずつ分けて、ビットシリアル転送する必要がある。本章では、入力部に必要な動作を示し、その動作からブロック長および次数の増加に対して時間遅れが生じない構成を提案する。

ここで、 L 個の各 PE が必要とする $(L+N-1)$ 個の入力を、 $x(kL-L-N+2) \sim x(kL)$ とする。このとき、 B クロック後すなわち L サンプル周期後の全ての PE の動作に必要な入力は、 $x((k+1)L-L-N+2) \sim x((k+1)L)$ となる。この入力値は、 $(N-1)$ 個の値 $x(kL-N+2) \sim x(kL)$ と、新たな L 個の値 $x((k+1)L-L+1) \sim x((k+1)L)$ が、PE 全体の動作に必要であることを示す。つまり、入力部では、 L サンプル周期前の PE の動作時に用いた $(N-1)$ 個の値と、新たな L 個の値を出力する必要がある。

そこで、ブロック長および次数の変化に対して、時間遅れが生じない入力部の構成を図7に示す。なお、図7の構成は、ブロック長3、タップ数3の場合を示す。この構成では、3 個のデータを PE へ転送すると同時に、 L サンプル周期後に PE の動作に必要なデータを B ビット長のシフトレジスタに格納する。このデータは、 L サンプル周期後に PE へ転送される。このように、データの流れが連続的に行われるため、時間遅れの生じない構成が実現可能となる。

図8 プロセッサの構成
Fig.8 Structure of processor.

6. VLSI 設計

<6.1> プロセッサの仕様

本実現法の性能を明らかにするために、PARTHENON を用いて VLSI 設計し、その評価を行う⁽¹⁴⁾。なお、実部品として用いたセルライブラリの設計ルールは、 $0.6\mu\text{m}$ CMOS スタンダードセル(VLSI テクノロジ社)であり、電源電圧は 5.0V である。ここでは、Remez アルゴリズムを用いて直線位相特性を実現する。この場合のフィルタ仕様は、規格化通過域端周波数 0.10 、規格化阻止域端周波数 0.20 となる低域通過形フィルタとする。また、入出力のデータ形式は 14 ビットの固定小数点とする。ただし、オーバフローを考慮して演算語長は、整数部を 2 ビット設けた 16 ビットにする。

<6.2> プロセッサの設計

図6に示した構成を基にして PE を設計する。この PE を高速化するためにパイプライン処理が施される。その結果、PE の入力部、関数 Φ の生成部、関数 Φ の加算部、シフト加算部のステージから構成される。これによって、パイプライン処理のピッチはほぼ加算器の演算時間に抑えられる。なお、加算器はモジュール化した 4 ビットの桁上げ先見加算器を 4 個用いている。

関数 Φ の分割数はプロセッサの性能に影響するため、消費電力が最小となる最適分割数を算出することは重要となる。我々は多くの設計例から、一般的に最適関数選択線数を 4 として分割化を行った場合にゲート数および消費電力が最小となることを見出している。また、その選択線数が 3 あるいは 5 の場合においてもゲート数および消費電力が最小に近い値となることも確認している。したがって、全ての場合において基本的にこの指針に基づいて PE を設計すればよい。また、分割数を Q とした場合、関数の加算部は 2 進木であるため、この部分は $\lceil \log_2 Q \rceil$ 段のステージ構成となる。ここで、 $\lceil z \rceil$ は z 以上となる最小の整数

表2 ブロック長に対する性能
Table 2. Performance for block length.

Block length	10	20
Number of taps	16	16
Machine cycle[ns]	15	15
Number of cycles	14	14
Sampling rate of PE (F_p) [MHz]	4.76	4.76
Sampling rate[MHz]	47.6	95.2
Number of gates	31,940	60,918
Area[mm ²]	3.65	6.96
Power dissipation [W]	1.68	3.22
Division number	4	4
Latency [ns]	300	300
Number of signal pins	20	40

を示す。

PE を基本的なモジュールとしたプロセッサ全体の構成を図8に示す。本プロセッサの制御は、PE が規則的に動作する構成となっているため、1個のカウンタを用いた制御モジュールによって容易に行える。つまり、制御のために複雑な回路を必要としない。

7. VLSI 評価

<7.1> ブロック長の変化に対する VLSI 評価

ここでは、ブロック長の変化に対する VLSI プロセッサの性能を表2に示し、本提案法の高速化について述べる。

従来の実現法ではクロックレートの向上に基づいて高速化を図ってきたが⁽¹⁾⁽⁶⁾、このレートの向上には限界があるため、ある一定以上の高速化を図ることは容易ではない。また、従来の FIR フィルタ実現の1つである直接形構成に基づく実現法は、高スループット化に伴い滞在時間が増大してしまう。一方、転置形構成に基づく実現法では基本的に滞在時間は1回の積和演算時間であるが、さらに高スループット化を図った場合には滞在時間は増大する。なぜなら、共通バスで伝搬される信号の各タップ間でのタイミングを合わせるために、非常に多くのレジスタを必要とするからである。そのため、滞在時間の増加を抑えながら極めて高いスループットを得ることは困難である。

一方、本実現法はブロック長10の場合には47.6MHz、ブロック長20の場合には95.2MHzというブロック長に比例した高いサンプリング周波数を実現できる。しかも、プロセッサの動作速度を4.76MHzという一定の値で実現できる。これは、本提案法がブロック実現におけるマルチレート性を有効に利用して得られた結果である。また、高スループット化を図っても滞在時間が300nsという一定の値で実現できるという特長もある。さらに、本実現法は同一のPEモジュールを拡張することによって、従来の高速化法では得ることのできなかった極めて高いスループットが実現可能となる。

また、分散演算を適用したことによって入出力信号がシ

表3 タップ数に対する性能
Table 3. Performance for number of taps.

Number of taps	16	32
Block length	20	20
Machine cycle[ns]	15	15
Number of cycles	14	14
Sampling rate of PE (F_p) [MHz]	4.76	4.76
Sampling rate[MHz]	95.2	95.2
Number of gates	60,918	115,377
Area[mm ²]	6.96	12.7
Power dissipation [W]	3.22	5.88
Division number	4	8
Latency [ns]	300	315
Number of signal pins	40	40

表4 ROM 実現の性能

Table 4. Performance of implementation using ROM's.

Number of taps	32
Block length	20
Machine cycle[ns]	15
Number of cycles	14
Sampling rate[MHz]	95.2
Power dissipation [W]	19.9
Number of gates	182,143
Area[mm ²]	20.8
Division number	4
Latency [ns]	315

リアル転送で十分となる。その結果パラレル転送の場合500本以上の信号線数を必要とするのに対して、本提案法では40本までに抑えることができる。

<7.2> タップ数の変化に対する VLSI 評価

タップ数の変化に対する VLSI プロセッサの性能を表3に示す。16タップのプロセッサの滞在時間は300nsであり、これに対して32タップのプロセッサの滞在時間は315nsである。これから、次数の増加に対して、滞在時間を最小化できることがわかる。これは、分散演算の処理時間が語長のみに依存することに着目して得られた結果である。ただし、16タップと32タップの滞在時間の差は、関数の分割数すなわち加算部のステージ数が1段違うためである。また、タップ数が増加しても95.2MHzというサンプリング周波数を低減させずに実現できる。これは分散演算におけるパイプライン処理と時間遅れが生じない入力部の構成法によって得られた結果である。これより、高次の場合でも滞在時間を抑えた上で高いスループットが実現できることがわかる。

さらに、本提案による低消費電力化法の有効性を示すために、関数 Φ の生成にROMを用いた32タップVLSIプロセッサを評価し、その性能を表4に示す。なお、ROMを用いたプロセッサの分割数は、最適関数回路を用いた場

表5 インパルス応答に対する性能
Table 5. Performance for impulse response.

Impulse response	Even symmetry			Odd symmetry		
Division number	8	6	4	8	6	4
Number of taps	32					
Block length	20					
Machine cycle [ns]	15					
Number of cycles	14					
Sampling rate [MHz]	95.2					
Number of gates	115,377	113,768	122,933	110,076	102,657	112,062
Area [mm ²]	12.7	11.78	13.5	12.54	11.65	12.55
Power dissipation [W]	5.88	5.59	6.73	5.81	5.41	6.12

合のサンプリング周波数と滞在時間を同じにした上で消費電力が最小となる値を選択した。本実現法を用いた 32 タップのプロセッサは、ROM を用いたプロセッサに対して、70.5%の消費電力の低減が可能となった。これは、関数 Φ の冗長性に着目した最適関数回路と直線位相特性に着目したペアリングテーブル法によって得られた結果である。

<7.3> インパルス応答に対する VLSI 評価

インパルス応答に対する VLSI 評価を表 5 に示す。ここでは、奇対称のインパルス応答を、偶対称のインパルス応答 h_j を用いて、

$$h_j = -h_{N-j} \quad (26)$$

とする。まず初めに、各分割数ごとに消費電力の比較を行う。タップ数 32 に対して分割数を 8 とした場合、奇対称のインパルス応答を利用したプロセッサでは、偶対称のインパルス応答を利用したプロセッサに対して、消費電力が 1.2% 減少した。また、分割数が 6 および 4 の場合には、それぞれ 3.2% および 9.1% の消費電力の減少となった。これは、奇対称性を利用したペアリングテーブル法は、偶対称性を利用した手法よりも 0 に属する横のカテゴリ値が多いため、より一層の低消費電力化が可能となることを示している。また、先に示した消費電力の比較結果から、分割数が減少するにしたがって奇対称性を利用したペアリングテーブル法による低消費電力化の効果が大きくなることがわかる。

次に消費電力が最小となる分割数を評価する。32 タップのプロセッサにおいて最適な分割数は 6 となった。すなわち、この場合には消費電力を最小にする最適分割数は最適関数選択線数 4 ではなく 6 の近傍で分割化した値となっている。これは、ペアリングテーブル法を適用することによって最適関数回路のゲート数が大幅に削減できたためである。したがって、最適関数回路と関数 Φ の加算部のゲート数の総和が最小となる選択線数がペアリングテーブル法を用いなかった場合に比べて大きくなり、その結果より分割数が少ない 6 の値で最小となった。

<7.4> 他の実現法との比較

ここでは、以下に示す低滞在時間、高スループット型の他の 3 つの実現法との比較を示し、本実現法の有効性を示す。

- (1) 我々が提案した実現法
- (2) Christo らのシストリックアレーを用いた実現法⁽¹⁵⁾
- (3) Lee らのハードウェア効率を考慮した実現法⁽¹⁶⁾
- (4) Yoshino らの転置型構成に基づいた実現法⁽⁵⁾

これらの実現法を比較するため、32 タップの FIR フィルタに対して PARTHENON を用いてそれぞれの性能評価を行った。その結果を表 6 に示す。なお、セルライブラリの設計ルールは、我々の場合と同様 0.6 μ m CMOS スタンダードセルを用いた。

Christos らのシストリックアレーを用いた実現法では、上限で約 143MHz という極めて高いサンプリングレートの実現が可能となる。しかし、この実現ではすべてのデータが 'bit-skewed' すなわち入力データが LSB から順次 1 clock cycle で 1 bit ずつずれて入力され、同様に出力も LSB から順次 1 bit ずつずれて出力されるため、タップ数を N 、語長を B とすると滞在時間は $(2B+N+\log_2 N-1)$ となる。従って、滞在時間がタップ数に依存して増加するため、高次に対して問題となる。また、すべてのタップのフィルタ係数に対する bit-products をゲート化する必要があるため、表 6 に示すようにゲート数および消費電力が極めて大きくなる。その場合のハードウェア量を評価すると、アレー数で $((B/2) \times (B+1) + (B/2)^2)$ 、また 1 アレー当たり Full Adder : $2N$, 2INPUT-AND : $2N$, Latch : $3N$ を必要とする。従って、タップ数の増加に対してハードウェア量が膨大となり実現が困難となる。それに対して、我々が提案した実現法ではこれと同じサンプリングレートを実現した場合においても、消費電力が 9W と約 1/5 程度となる。従って、この実現法と比較した場合極めて効率的な構成となっていることが分かる。

Lee らの実現法では、プログラマブルフィルタを効率的に実現できる点に特長がある。ただし、表 6 のゲート数には、VMA(Vector Merge Adder), Pre-processing, Coefficient Registers および部分積の生成回路は含まれていない。また、

表 6 他の実現法との性能比較 ($N = 32$, $B = 14$ の場合)
Table 6. Comparison of performance for the various implementation methods in case of $N = 32$ and $B = 14$.

	(1)	(2)	(3)	(4)
Number of taps	32			
Machine cycle [ns]	15	7	20	18
Sampling rate [MHz]	95.2	142.8	7.14	55.6
Power dissipation [W]	5.59	42.6	0.93	2.44
Number of gates [k]	114	376	25.6	29.2
Area [mm ²]	11.8	44.8	2.96	3.17
Latency [ns]	315	448	180	18

タップ数に依存せず、低滞在時間が得られる点も大きな特長である。しかし、サンプリングレートが語長を B とした場合($B/2$)に比例した値となるため、表 6 に示すように上限で約 7MHz 程度と処理速度に問題がある。また、フィルタ係数を可変としているため、プロセッサの信号線数が($N \times B$)以上必要となり、高いタップ数に対して問題となる。それに対して、我々の実現法ではブロック長を L とした場合、タップ数に依存することなく $2L$ だけでよい。

Yoshino らの転置型構成に基づいた実現法では、CSD を用いて極めて効率的に、低滞在時間で、高スループットが実現可能となる。従って、転置型構成本来の特徴である入力から出力まで 1 clock cycle すなわち 1 回の積和演算時間である場合には、我々が提案した実現法よりも有効となってくる。しかし、この実現法でさらに高速化を図るためにパイプラインピッチを減少させた場合、分割化のためのレジスタを各タップ毎に入れる必要があるため、滞在時間がタップ数に比例して増加する。従って、この場合には転置型構成の本来もっている低滞在時間の特長が損なわれることになる。また、ピッチを減少するほど非常に多くのレジスタ数を必要とするため、ハードウェア量が増大する。さらに、パイプラインピッチも極端には減少できないため、サンプリングレートの向上には限界がある。

従来の実現法ではいずれもサンプリングレートに上限を有していたが、我々の提案した実現法では動作速度が一定で遅い PE をブロック長分用いることによって、理論的にはブロック長に比例した高スループット化が実現可能となる。従って、本実現法を用いることによって従来得ることのできなかった高速化が可能となり、この点が本提案法の最も大きな特長と言える。また、分散演算の特長を用いることによって、極めて高いサンプリングレートを効率的に、しかもタップ数に依存することなく低滞在時間および小さな信号線数で実現可能となる。これらの結果から、本実現法の有効性が明らかになった。

8. むすび

本論文では、高スループット形 FIR フィルタの滞在時間最小化 VLSI アーキテクチャを提案した。まず、本提案法ではクロックレート向上による従来の高速化法とは異なる新たな手法として、ブロック実現のマルチレート性に着目したシステムアーキテクチャを示した。次に、このシス

テムアーキテクチャに分散演算を適用することによって、信号線数と滞在時間を大幅に抑えることができた。これは、分散演算の特長であるシリアル入出力と内積演算の処理時間が語長のみに依存する性質に着目して得られた結果である。また、高スループット化に伴う消費電力の増大を抑えるために、本実現法では関数の生成に ROM を用いた従来の方法ではなく、論理ゲートによる最適関数回路を用いた低消費電力化を図った。さらに、より一層の低消費電力化を目的として、FIR フィルタが完全な直線位相特性を実現できることに着目し、インパルス応答の偶対称性だけではなく新たに奇対称性を利用したペアリングテーブル法を提案した。

最後に、本提案法を用いて VLSI プロセッサを設計し、その評価を行った。その結果、タップ数 32 の FIR フィルタがプロセッサの動作速度を 4.76MHz($0.6\mu\text{mCMOS}$ スタンダードセル)の一定値に抑えながら、95.2MHz という高いサンプリング周波数と 315ns の滞在時間で実現できた。さらに、本実現法では同一の PE モジュールを拡張することによって、従来の高速化法では得ることのできなかった極めて高いスループットが実現可能となる。また、次数が増加した場合においても滞在時間をほぼ一定に抑えた上で、一定の高いスループットを保つことができる。その結果、高次のフィルタを実現する場合においても有効な手法となる。

本提案法はアーキテクチャからのアプローチによる実現法であるが、さらに素子の微細化や電源電圧の低減化などデバイスからのアプローチを用いることによって、より有効な手法となる。

謝辞 本研究にあたり、PARTHENON を御提供いただいた NTT コミュニケーション科学研究所の皆様様に深く感謝いたします。

(平成 10 年 1 月 26 日受付, 平成 10 年 5 月 22 日再受付)

文献

- (1) C. A. Mead and L. A. Conway : "Introduction to VLSI Systems", Addison-Wesley, (1980)
- (2) 亀山充隆, 樋口龍雄 : 「ロボットと VLSI コンピュー

- タ」, 日本ロボット学会誌, Vol.6, No.4, pp.332-p.338, (1988)
- (3) 恒川佳隆, 三浦守: 「分散演算を用いた制御向き状態空間デジタルフィルタの高性能 VLSI アーキテクチャ」, 信学論(A), Vol.J78-A, No.3, pp.357-364, (1995)
- (4) 今井正治編: 「ASIC 技術の基礎と応用」, 電子情報通信学会, (1994)
- (5) T. Yoshino, R. Jain, P. T. Yang, H. Davis, W. Gass, and A. H. Shah: "A 100MHz 64-tap FIR Digital Filter in 0.8 μ m BiCMOS Gate Array", IEEE J. Solid-State Circuits, Vol.25, No.6, pp.1494-1501, (1990)
- (6) 野崎 剛, 恒川佳隆, 三浦 守: 「高スループット形 FIR フィルタの滞在時間最小化 VLSI アーキテクチャ」, 電子情報通信学会技術研究報告, DSP97-116, pp.87-94, (1997)
- (7) 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・低消費電力形アーキテクチャ」, 電学論(c), Vol. 118-C, No. 7/8, pp. 1098-1107, (1998)
- (8) H. H. Lu, E. A. Lee, and G. David: "Fast Recursive Filtering with Multiple Slow Processing Elements", IEEE Trans., Circuits & Syst., Vol. CAS-32, No.11, (1985)
- (9) C. F. Chen: "Implementing FIR Filters with Distributed Arithmetic", IEEE Trans., Acoust. Speech & Signal Process., Vol. ASSP-33, No.4, pp.1318-1321, (1985)
- (10) C. S. Burrus: "Digital Filter Structures Described by Distributed Arithmetic", IEEE Trans., Circuits & Syst., Vol. CAS-24, pp.674-680, (1977)
- (11) A. Peled and B. Liu: "A New Hardware Realization of Digital Filters", IEEE Trans., Acoust. Speech & Signal Process., Vol. ASSP-22, No.12, pp.456-462, (1974)
- (12) 名古屋彰, 中村行宏, 小栗 清, 野村 亮: 「高位記述からの大規模論理合成における多段論理最適化」, 信学論(A), Vol.J74-A, No.2, pp.206-217, (1991)
- (13) R. K. Brayton and C. McClellan: "The decomposition and factorization of Boolean expression", ISCAS-82, pp.49-54, (1982)
- (14) NTT データ通信株式会社: "PARTHENON User's Manual", (1990)
- (15) G. C. Christos and Z. P. Kiamal, "Low-latency Bit-Parallel Systolic VLSI Implementation of FIR Digital Filters", IEEE Trans. Circuits Syst. II, Vol.43, No.7, pp.529-534, (1996)
- (16) H. R. Lee, C. W. Jen and C. M. Liu, "A New Hardware-Efficient Architecture for Programmable Filters", IEEE Trans. Circuits Syst. II, Vol.43, No.9, pp.637-644, (1996)

恒川 佳隆

(非会員)昭55岩手大・工・電気卒。昭58東北大学院修士課程了。同年岩手大・工・助手。平6同大講師。工博。デジタル信号処理, デジタル制御の研究に従事。IEEE, 計測自動制御学会会員。

野崎 剛

(非会員)平8岩手大・工・情報卒。現在, 同年同大学大学院工学研究科博士前期課程在学中。デジタルフィルタの高速実現に関する研究に従事。

三浦 守

(正員)昭37岩手大・工・電気卒。同年東北大・工・助手。昭43秋田工業高専助教授。昭47岩手大・工・助教授。昭60同大教授。工博。この間, マグネティックス, デジタル信号処理などに関する研究に従事。平1年10月より10カ月間、ウインザ大(カナダ)およびカリフォルニア大サンタバーバラ校(米国)の客員研究員。計測自動制御学会, IEEE, 情報処理学会, 日本ME学会各会員。