制御向き状態空間デジタルフィルタ用高性能 VLSI プロセッサの 設計と性能評価†

恒 川 佳 隆*・千 葉 晃 司*・三 浦 守*

Design and Performance Evaluation of High-Performance VLSI Processor Suitable for Control Systems for State-Space Digital Filters

Yoshitaka TSUNEKAWA*, Kohji CHIBA* and Mamoru MIURA*

State-space digital filters can synthesize the optimal filter structure with respect to overall quantization effects, but have many more multiplications than a canonical structure has. Therefore the sampling rate and the latency become important problems on implementing these filters. Especially for the very long latency the control system becomes unstable when these filters are used in the feedback control systems, such as the robot control system.

Previously we proposed a VLSI-oriented highly parallel architecture for state-space digital filters with high sampling rate and small latency. For the purpose of further speeding up and reducing hardware complexity, the distributed arithmetic, of which processing time depends on only word length, was applied to this architecture, making good use of high accuracy of state-space digital filters. Therefore, the very high sampling rate can be implemented independently of the filter order and the numbers of the inputs and outputs.

This paper designs and evaluates high-performance VLSI processor based on our proposed VLSI architecture for state-space digital filters using distributed arithmetic. As a result, we confirm that the processor can perform the sampling rate of about 1.67-MHz and the latency of 600ns using $0.8\mu\text{m}$ CMOS technology, in the case where the filter order is 16, the numbers of inputs and outputs are 1, respectively, and the word length is 14. Moreover, we show that the high sampling rate of about 1.6-MHz can be held constant, even if the filter order and the numbers of inputs and outputs increase.

Key Words: state-space digital filters, high accuracy, distributed arithmetic, high-performance VLSI architecture, VLSI evaluation

1. はじめに

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るようになってきた。特に制御の分野では、デジタルフィルタや制御装置の設計が状態方程式を用いて行われている。状態方程式で表されるデジタルフィルタを状態空間デジタルフィルタ（以下、SSDF）と呼ぶ。ここでは SSDF を、これまでの伝達関数表現による狭義のデジタルフィルタばかりでなく、カルマンフィルタ、オブザーバ、レギュレータなどを含む広義のデジタルフィルタとして扱う。

デジタルフィルタの設計においては、与えられた仕様が満足するフィルタ係数を求めること（近似）だけでなく、量子化誤差が小さなフィルタ構造を決定すること（合成）が必要である。デジタルフィルタを状態方程式で記述すれば、

システム理論的観点から近似と合成を統一に行うことが可能である。その結果、近似誤差と量子化誤差が最小な SSDF の設計が可能となる^{1), 2)}。しかし、この方法で設計された SSDF は係数行列が密となるため、出力当りの計算量が増加する。そのため、現在の高性能シグナルプロセッサで高次の SSDF を実現しても、音声程度の周波数領域の処理にしか使用できないという問題点があった。

これに対して、シストリックアレイに基づく SSDF の VLSI 向きアーキテクチャの提案が行われてきた³⁾。しかし、これは非常に高い処理速度が得られる反面、高次のフィルタに対して滞在時間（latency）とハードウェア量が非常に大きくなってしまう。センサ信号処理やロボット制御などでのフィードバック制御系で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。このため、処理速度ばかりでなく滞在時間が大きな問題となる⁴⁾。したがって、このような分野においてはシストリックアレイ実現は適当ではない。

われわれはこれまでに滞在時間とハードウェア量を大幅に減少したうえで、高い処理速度をもつ SSDF 用 VLSI 向き

† 計測自動制御学会東北支部第 160 回研究集会で発表（1996・5）

* 岩手大学工学部 盛岡市上田 4-3-5

* Faculty of Engineering, Iwate University, Morioka
(Received June 24, 1996)
(Revised June 5, 1997)

アーキテクチャを提案してきた⁵⁾。すなわち、滞在時間を大幅に減少させるために、状態変数の各要素および出力の計算が同時に同一命令で実行できることに着目して、共通バスによるブロードキャスト通信だけを用いた1次元のSIMDプロセッサアレイを考案してきた。

本アーキテクチャを用いることによって、高次および多入力多出力のSSDFに対してもこれまでにない高速処理が可能となった。しかし、今後の制御系においては対象となるシステムの複雑化に伴い、これまで以上に高い次数および入出力数を高速に処理することが要求されてくる。その場合、次数および入出力数に依存して膨大な乗算回数を実行しなければならず、この実現法を用いた場合においても処理速度は大きく低下する。また、これをVLSI実現する場合、膨大なハードウェア量が必要となる。

そこで、われわれはSSDFの高精度性に着目して、先のアーキテクチャにさらに分散演算を適用した高性能VLSIアーキテクチャを提案してきた⁶⁾。SSDFにおいては量子化誤差に対して最適合成が可能となるため²⁾、実現の際には必要語長を他の構造に対して最小化できる。すなわち、計算時間が語長のみに依存する分散演算を用いた本実現法は、高次および多入力多出力の場合でも非常に高い処理速度が実現可能となる。

本論文では、分散演算に基づくSSDF用プロセッサをVLSI設計システムPARTHENONおよびCOMPASSを用いて設計し、その性能評価を行う。その結果、非常に高次な場合においても高速処理が実現可能となることを明らかにする。さらに、フィルタの次数あるいは入出力数が増加した場合、他の実現法では処理速度が大きく低下してしまうのに対して、本実現法では高速性がほぼ一定に保たれることをVLSI評価によって明らかにする。

2. 分散演算を適用した高性能VLSIアーキテクチャ

2.1 状態空間デジタルフィルタ

線形システム理論によれば、有限次数の線形時不変システムはつぎの状態および出力方程式によって記述される²⁾。

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2)$$

ここで、 $\mathbf{x}(k)$ は n 次の状態ベクトル、 $\mathbf{u}(k)$ は r 次の入力ベクトル、 $\mathbf{y}(k)$ は m 次の出力ベクトルである。また、 $\mathbf{A}$ 、 $\mathbf{B}$ 、 $\mathbf{C}$ 、 $\mathbf{D}$ はそれぞれ $n \times n$ 、 $n \times r$ 、 $m \times n$ 、 $m \times r$ の実係数行列である。(1)、(2)式によって記述されるデジタル信号処理システムや制御システムを状態空間デジタルフィルタ(State-Space Digital Filters)と呼び、これをSSDF[$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$]と記述する。SSDF[$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$]の伝達関数 $\mathbf{H}(z)$ はつぎのように得られる。

$$\mathbf{H}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (3)$$

ここで、状態ベクトル $\mathbf{x}(k)$ がシステムの遅延要素の出力を表し、行列 $\mathbf{A}$ 、 $\mathbf{B}$ 、 $\mathbf{C}$ 、 $\mathbf{D}$ の各要素がシステムの係数に対応

Fig. 1 Data dependence graph for SSDF.

するとき、SSDF[$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$]は伝達関数 $\mathbf{H}(z)$ を有する状態空間デジタルフィルタの一つの構造(乗算, 加算, 遅延の組み合わせ)を表している。

伝達関数 $\mathbf{H}(z)$ を有する状態空間デジタルフィルタの構造は無限に存在する。すなわち、 $n \times n$ の任意の正則行列 $\mathbf{T}$ に対して、 $\mathbf{x}' = \mathbf{T}^{-1}\mathbf{x}$ と状態ベクトルの変換を行うとき、 $\mathbf{H}(z)$ を有する他の構造はSSDF[$\mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \mathbf{T}^{-1}\mathbf{B}, \mathbf{C}\mathbf{T}, \mathbf{D}$]によって一般的に記述される。状態空間デジタルフィルタでは、適当な等価変換行列 $\mathbf{T}$ を選ぶことによって、係数量子化誤差と演算誤差が共に最小であり、かつリミットサイクルが発生しない最適合成が可能となる²⁾。特に、他のフィルタ構造においては周波数特性の通過帯域幅が狭帯域になるほど誤差利得が大きく増加するのに対して、最適構造では通過帯域幅に無関係に一定の誤差利得をもつ。したがって、状態空間デジタルフィルタを用いれば、非常に高精度なフィルタリングが可能となる(ここでは、これを状態空間デジタルフィルタの高精度性と呼んでいる)。その結果、仕様に対して必要語長を最小化できる。

2.2 高並列システムアーキテクチャ

本節では、先に提案した状態空間デジタルフィルタ(以下SSDF)の高並列システムアーキテクチャの概要について述べる⁵⁾。われわれは、(1)、(2)式の状態変数 $\mathbf{x}(k+1)$ および $\mathbf{y}(k)$ の各要素が同一データを用いた同時処理により実行されることに着目して、SSDFの並列化の手法としてFig. 1に

Fig. 2 1-dimensional SIMD array.

Fig. 3 Function of PE.

示すようなグローバルバス通信を用いたデータ依存グラフを考える。この並列化の手法は、PE数と滞在時間の大きさを極力抑えて、高いスループットを持つVLSIアーキテクチャを得ることを目的としている。このグローバルバスを用いたデータ依存グラフに対して、Fig. 1のように射影方向を取り、Fig. 2に示す1次元のシステムアーキテクチャを提案する。このシステムアーキテクチャは、任意の次数および多入力多出力に対して容易に拡張可能なモジュール構造になっている。以下にその構成法を示す。

- (1) この実現法は、 n 次の状態ベクトルの各要素に対して $PE_1, PE_2, \dots, PE_n$ を設け、 m 次の出力ベクトルの各要素に対して $PE_{n+1}, PE_{n+2}, \dots, PE_{n+m}$ を設けた1次元のプロセッサアレイ実現である。
- (2) 並列度を $(n+m)$ とすることによって、 $x(k+1)$ および $y(k)$ のいずれの処理時間も、各PEで同時に実行される $(n+r)$ 項の内積演算を求める時間でよい。
- (3) 更に、各PEにおける $(n+r)$ 項の内積演算に対しては、パイプライン処理を用いて処理の高速化を図る。Fig. 3にPEの機能を示す。PEにはフィルタ係数用のメモリを内蔵させる。いま、PEのパイプラインのステージ数を S 、各ステージの実行時間であるピッチを τ 、内積演算の項数を L とする場合、処理時間 T は

$$T = (L + S - 1)\tau \quad (4)$$

となる。出力はラッチ回路によってつぎのデータが出力されるまで保持される。当然、ステージ数を大きくするほどピッチが減少し、より短い時間で出力結果が得られる。

- (4) 各PEで内積演算を実行する場合、各PE間では同

Fig. 4 Basic structure of distributed arithmetic.

一データを必要とするだけなので、グローバルバスを用いたデータ通信により実現できる。バスの利用形態はブロードキャスト（1対多の通信）形である。このバスを用いることによって全PEでは同一命令を実行すればよく、したがって1次元のSIMD形プロセッサアレイ構成とすることができる。

2.3 分散演算を適用したVLSIアーキテクチャ

われわれは、先のアーキテクチャに分散演算（Distributed Arithmetic）を適用して、大幅にハードウェア量を減少した上で、次数および入出力数に依存することなく高速処理可能な高性能VLSIアーキテクチャを提案した⁶⁾。

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^{7), 8)}。いま、項数 N の係数ベクトル $\mathbf{a} = (a_1, \dots, a_N)$ と変数ベクトル $\mathbf{v} = (v_1, \dots, v_N)$ との内積

$$y = \mathbf{a}\mathbf{v} = \sum_{i=1}^N a_i v_i \quad (5)$$

を考える。ただし、 $-1 \leq v_i < 1$ 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。つまり、

$$v_i = -v_i^0 + \sum_{k=1}^{B-1} 2^{-k} v_i^k \quad (6)$$

と表される。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。(6) 式を (5) 式に代入すれば、内積演算 $\mathbf{a}\mathbf{v}$ は次式で示される。

$$y = -\Phi(v_1^0, \dots, v_N^0) + \sum_{k=1}^{B-1} 2^{-k} \Phi(v_1^k, \dots, v_N^k) \quad (7)$$

ただし、 Φ は

$$\Phi(v_1^k, \dots, v_N^k) = \sum_{i=1}^N a_i v_i^k \quad (8)$$

である。

(7) 式を実現するには、 $(v_1^k, \dots, v_N^k)$ をアドレスとするROMに、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き、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えばよい。分散演算の基本的な構成法をFig. 4に示す。

Fig. 5 Configuration of SSDF system using distributed arithmetic.

Fig. 6 Basic structure of PE.

前述した SSDF 用システムアーキテクチャでは、SIMD 形のプロセッサレイ構造になっているため、状態変数 $x(k)$ の各要素は同時に求められる。したがって、このアーキテクチャに分散演算を容易に適用可能となる。今、状態変数 $x(k)$ と出力 $y(k)$ の各要素がすべて $(n+r)$ 項の内積演算になっていることに着目して、(1), (2) 式の r 入力 m 出力 n 次の SSDF に対して分散演算を次式のように適用する。

$$x(k+1) = \sum_{j=1}^{B-1} 2^{-j} \Phi_p^j - \Phi_p^0$$

$$y(k) = \sum_{j=1}^{B-1} 2^{-j} \Phi_q^j - \Phi_q^0 \quad (9)$$

ただし、

$$\Phi_p^j = Ax^j(k) + Bu^j(k)$$

$$\Phi_q^j = Cx^j(k) + Du^j(k) \quad (10)$$

Table 1 Filter specifications.

Filter order	16
Number of inputs	1
Number of outputs	1
Data format	14 bit fixed-point

Fig. 7 Design process.

r 入力 m 出力 n 次の場合、乗算回数は $(n+r)(n+m)$ 回であり、 r, m, n が増加した場合、他の実現法においては処理速度は大きく低下する。しかし、(9) 式から明らかなように、分散演算を適用した場合、多入力多出力あるいは高次の場合に対しても、原理的には処理時間は語長 B のみに依存した実現が可能である。また、SSDF は、最適合成法を用いることによって、設計仕様に対して必要語長は最小でよい。したがって、出力の誤差特性を含めて VLSI 化を考えた場合、SSDF においては分散演算実現は非常に有効となる。

Fig. 5 にこの場合のシステム構成を示す。また、Fig. 6 には PE の基本構成を示す。このシステム構成は、次数および出力数の増加に対して、単に PE を共通バスに対して拡張するだけでよく、したがって VLSI 化に適したモジュール構造となっている。

3. フィルタの仕様と設計プロセス

分散演算を用いた SSDF の構成はフィルタ仕様に依存する。そこで、本章ではまず性能評価を行う VLSI プロセッサのフィルタ仕様を与える。つぎに、プロセッサの設計プロセスについて述べる。

3.1 フィルタ仕様

われわれが設計する VLSI プロセッサのフィルタ仕様を Table 1 に示す。プロセッサのフィルタ次数は 16 であり、このような高次の SSDF を実現する場合、他の実現法では乗算回数が非常に膨大となるため高い処理速度は望めない。フィルタの入力および出力数は 1 とし、数値のデータ形式は語長 14 ビットの固定小数点の 2 の補数表示とする。

3.2 VLSI プロセッサの設計プロセス

VLSI プロセッサの設計プロセスは、Fig. 7 に示すように

Fig. 8 Block diagram of VLSI processor.

2ステップに分けらる。第1ステップではNTTが開発したVLSI設計システムPARTHENONを用いてプロセッサの高位設計を行う。PARTHENONは、ハードウェア記述言語を用いた設計から回路合成まで一貫して作業を行えるため、効率的なプロセッサの設計を可能にする⁹⁾。

PARTHENONによる作業は4つのフェーズに分けられる。最初のフェーズでは、動作記述言語SFLを用いてVLSIプロセッサの設計を行う。SFL言語は、記述の基本単位をモジュールと呼び、複数のサブモジュールを包含する階層表現が可能である。このフェーズでは、プロセッサの機能とI/O、そしてサブモジュール間のインタフェースを設計する。第2のフェーズでは、シミュレータSECONDSを用いてモジュールの動作をソースレベルで検証する。第3のフェーズでは、シンセサイザSFLEXPを用いて動作記述からゲート接続回路へ展開する。ここで論理的な単純化が行われ、テクノロジーに依存しないネットリストが合成される。最後のフェーズでは、オプティマイザOPT_MAPを用いてテクノロジー・マッピングと回路の最適化を行い、テクノロジー依存の実部品からなるネットリストを合成する。実部品の情報は、実際のテクノロジーに依存した物理特性や制約条件を記述したセル・ライブラリによって与えられる。

第2のステップでは、COMPASSを用いたレイアウト設計を行う。COMPASSで処理するネットリストは、PARTHENONから出力されたネットリストに対する簡単なフォーマット変換によって得られるため、設計プロセスの移行が容易である。ここでパッドを配置したフロアプランを検討し、チップを生成する。

4. VLSIプロセッサの設計

4.1 VLSIプロセッサの構成

分散演算に基づくVLSIプロセッサのブロック図をFig. 8に示す。

Table 1のフィルタ仕様から、プロセッサは状態変数およ

Fig. 9 Memory which divided the function Φ .

び出力の各要素を計算するために17個のPEが必要となる。プロセッサの動作は、1個のカウンタを用いたステート・マシンとして設計することで容易に制御できる。つまり、動作制御のために複雑な回路を必要としない。

各PEに対するデータ通信はブロードキャスト方式となるため、共通のデータバスを用いた実現が可能となる。また、PEの入力はビット・シリアルで十分である。したがって、PE間の相互接続に必要な配線数は大きく抑えられる。

VLSIプロセッサの入力(u_1)は、PEへビット・シリアルで送信されるため、入力のために必要な端子の数は最小限に抑えられる。また、出力(y_1)の流れは滞在時間を減少するためにビット・パラレルとする。

4.2 関数用メモリの実現

PEの構成を考えた場合、Table 1の仕様を満たす関数用メモリをROMを用いてハードウェア実現するためには、 2^{17} ワードの膨大な容量が必要となる。この問題に対しては、メモリ容量を大幅に減少する方法として、関数 Φ の分割化法が有効な手段となる。すなわち関数 Φ に対して、分割数を Q として以下の処置を施す。

$$\begin{aligned}\Phi(v_1^k, \dots, v_{n+r}^k) &= \sum_{i=1}^{n+r} a_i \cdot v_i^k \\ &= \sum_{i=1}^{\frac{n+r}{Q}} a_i \cdot v_i^k + \dots + \sum_{i=Q'}^{n+r} a_i \cdot v_i^k \\ &= \Phi(v_1^k, \dots, v_{\frac{n+r}{Q}}^k) + \dots \\ &\quad + \Phi(v_{Q'}^k, \dots, v_{n+r}^k) \quad (11)\end{aligned}$$

ここで、

$$Q' = \frac{(Q-1)(n+r)}{Q} + 1 \quad (12)$$

これによって、容量の大きな関数用メモリをいくつかの小容量のメモリの加算として実現でき、大幅にメモリ容量を減少できる。われわれは、関数 Φ の実現に必要なROMの面積を検討した上で、関数メモリの分割数を2とする。この場合の関数メモリの構成をFig. 9に示す。これをFig. 6のROMに単に置き換えて使用すればよい。

また関数用メモリの実現を考えた場合、ROMを用いるほかに論理ゲートによる実現が考えられる。しかし、種々のフィルタ係数に対してROMでは記憶内容を変更するだけでよいが、論理ゲート実現ではプロセッサの設計をやり直す必要が

Table 3 Parameters of the VLSI processor.

Filter order	8	16	24	16
Numbers of inputs and outputs	1	1	1	8
Word length [bits]	14	14	14	14
Division number of ROMs	1	2	3	3
Machine cycle [ns]	34	34	34	34
Number of cycles	17	18	19	19
Sampling frequency [MHz]	1.73	1.63	1.55	1.55
Power dissipation [W/MHz]	0.091	0.325	0.708	0.677
Number of gates	19,170	57,095	118,897	110,163
Number of signal pins	18	18	18	123

を用いた逐次処理によって実現した場合（サンプリング周波数で約15KHz程度）と比較して、100倍以上（ $\tau=150\text{ns}$ のDSPの場合）も高速である¹⁰⁾。また、本プロセッサの滞在時間は、サンプリング周期と一致するため600nsと非常に減少できる。また、これらの値は、COMPASSにおける設計ターゲットの動作周波数である30MHzにおいて得られたサンプリング周波数と滞在時間である。なお、このターゲットとした動作周波数30MHzは、PARTHENONで設計されたプロセッサのマシン・サイクル34nsから決定された値である。

また、プロセッサはROMを含めて55,000ゲート以下で実現され、そのチップ面積は $8.6 \times 6.6\text{mm}^2$ と1チップ実現が可能な大きさである。なお、ゲート数は2入力NAND換算値である。

ここで消費電力の面からチップ性能を検討する。Fig. 11のレイアウトは動作周波数30MHzを想定して設計しており、プロセッサの消費電力は9.3Wとなる。本プロセッサをパッケージで実現することを考えた場合、この消費電力が問題となりうる。しかし、プロセッサの消費電力を3.0Wとして設計した場合でも、500KHz以上のサンプリング周波数を実現でき、単一のDSPを用いた逐次処理実現と比べても30倍以上の高速化が達成できる。

5.2 次数および入出力数の変化に対する性能評価

われわれが提案したSSDFのVLSIアーキテクチャは、分散演算を適用したことによって、次数あるいは入出力数の変化に対して処理速度が大きく低下することはない。このような特性を明らかにするために、フィルタの次数および入出力数

を変化させたVLSIプロセッサの性能評価をPARTHENONを用いて行う。PARTHENONによって得られる評価結果は、セル・ライブラリに物理情報として仮想配線容量が含まれているため、COMPASSを用いてレイアウト設計したVLSIプロセッサの性能とほぼ一致する。つまり、PARTHENONを用いたVLSI評価によって、プロセッサの性能を十分知ることができる。

次数および入出力数の変化に対するVLSIプロセッサの性能をTable 3に示す。ここでは、設計したプロセッサの次数16および入出力数1を基準として、フィルタの次数を8および24とした場合、また入出力数が8の場合について検討を行った。各フィルタ仕様に対するROMの分割数は、実現に要するROMの容量および最大遅延時間を考慮して決定する。特に、ROMの遅延時間を加算器の遅延時間以下にすることによって、プロセッサの動作マシン・サイクルは一定に保たれる。その結果、プロセッサのサンプリング周波数は、ROMの出力を加算するステージの増減のみに依存する。すなわち、フィルタの次数あるいは入出力数が増加した場合、分散演算を適用しない実現法では大きく処理速度が低下してしまうのに対して、本実現法ではほぼ一定に保たれる。

また、信号線数に着目すると、1入力1出力の場合では18本だけでプロセッサを実現できる。8入力8出力数の場合においても123本の信号線で実現可能となる。したがって、分散演算を用いた本実現法では、大幅に信号線数を減少できる。

なお、Table 2とTable 3のフィルタ次数16における結果の若干の違いは、サンプリング周波数についてはCOMPASSとPARTHENONにおける動作周波数が多少異なってくるためである。また、消費電力、ゲート数については使用しているセル・ライブラリのデータの違いとそれらの算出法にある。

6. おわりに

本論文では、われわれが先に提案した分散演算を適用したVLSIアーキテクチャに基づく状態空間デジタルフィルタ用VLSIプロセッサを設計し、その性能評価を行なった。その結果、フィルタの次数16、入出力数1、語長14ビットのSSDFを1チップ実現した場合、非常に高次なフィルタにもかかわらずサンプリング周波数1.67MHz（ $0.8\mu\text{m}$ CMOSスタンダードセル）の高速処理が実現可能となることが明らか

Table 2 Key performance of the VLSI processor.

Technology	$0.8\mu\text{m}$ double-metal CMOS standard cell
System clock	30 [MHz]
Sampling frequency	1.67 [MHz]
Latency	600 [ns]
Chip size	$8.6 \times 6.6 [\text{mm}^2]$
Number of gates	54,223.5 [gates]
Power dissipation	0.31 [W/MHz]
Supply voltage	5.0 [V]
Number of pins	115

となった。また、滞在時間も 600ns と非常に減少させることができた。

さらに、次数および入出力数の変化に対する性能評価を行った結果、分散演算に基づく VLSI プロセッサでは処理速度の低下が極力抑えられ、約 1.6MHz とほぼ一定になることが明らかとなった。

以上のことから、われわれが提案した分散演算を適用した高並列 VLSI アーキテクチャは、今後更に高いフィルタ次数および入出力数で高速化が要求される分野において、非常に有効な実現法であると言える。

本研究を行うにあたり、貴重なご助言を頂いた東北大学大学院情報科学研究科樋口龍雄教授に心より感謝申し上げます。また、PARTHENON をご提供いただいた NTT データ通信(株)、および有益なご助言をいただいた(株)ADC の横川隆氏、山本和行氏、入江洋氏に深く感謝いたします。

参 考 文 献

- 1) S. Y. Hwang: Minimum uncorrelated unit noise in state-space digital filtering, IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-25-4**, 273/281 (1977)
- 2) 樋口龍雄, 川又政征: 状態空間デジタルフィルタの有限語長実現問題, 計測と制御, **22-12**, 991/1004 (1983)
- 3) H. H. Lu, E. A. Lee and D. G. Messerschmitt: Fast recursive filtering with multiple slow processing elements, IEEE Trans., Circuits and System, **CAS-32-11**, 1119/1129 (1985)
- 4) 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, **6-4**, 332/338 (1988)
- 5) 恒川佳隆, 志田純一, 川又政征, 樋口龍雄: 滞在時間がきわめて小さい状態空間デジタルフィルタ用高並列 VLSI アーキテクチャ, 計測自動制御学会論文集, **27-9**, 1034/1040 (1991)
- 6) 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, 電子情報通信学会論文誌 A, **J78-A-3**, 357/364 (1995)
- 7) P. Peled and B. Liu: A new hardware realization of digital filters, IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-22-6**, 456/462 (1974)
- 8) C. S. Burrus: Digital filter structures described by distributed arithmetic, IEEE Trans. Circuits and System, **CAS-24-12**, 674/680 (1977)
- 9) NTT データ通信株式会社: PARTHENON User's Manual
- 10) 大町慎太郎, 恒川佳隆, 関 享士郎, 志田純一: 状態空間デジタルフィルタのマルチ DSP 実現の検討, 電気関係学会東北支部連合大会予稿集, 107/107 (1989)

[著 者 紹 介]

恒 川 佳 隆 (正会員)

1980 年岩手大学工学部電気工学科卒業。83 年東北大学大学院修士課程修了。同年岩手大学工学部助手。94 年同大講師。デジタル信号処理、デジタル制御の研究に従事(工学博士)。IEEE、電子情報通信学会の会員。

千 葉 晃 司

1995 年岩手大学工学部情報工学科卒業。同年同大学大学院工学研究科修士課程情報工学専攻入学。現在に至る。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 (正会員)

1962 年岩手大学工学部電気工学科卒業。同年東北大学工学部助手。68 年秋田工業高専助教授。72 年岩手大学工学部助教授を経て、85 年同大教授。この間、マグネティックス、デジタル信号処理およびニューラルネットワークなどに関する研究に従事(工学博士)。電気学会、電子情報通信学会、IEEE などの会員。