計測自動制御学会東北支部 第160回研究集会 (1996.5.24)
資料番号 160-15

制御向き状態空間ディ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

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

○千葉晃司*，恒川佳隆*，吉田等明*，三浦　守*

○Kohji Chiba*, Yoshitaka Tsunekawa*, Hitoaki Yoshida*, Mamoru Miura*

*岩手大学

*Iwate University

キーワード： 状態空間ディジタルフィルタ (state-space digital filters)，高精度 (high accuracy)，分散演算 (distributed arithmetic)，高性能VLSIアーキテクチャ (high-performance VLSI architecture)，VLSI評価 (VLSI evaluation)

連絡先： 〒020　盛岡市上田4-3-5 岩手大学 工学部 情報工学科 恒川研究室
千葉晃司，Tel.: (0196)21-6468，Fax.: (0196)23-5491，E-mail: kohji@cis.iwate-u.ac.jp

1. はじめに

最近，計測・制御や通信など様々な分野で，ディ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，ディ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ィジタルフィルタを状態空間ディジタルフィルタ（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ィ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ィジタルフィルタとして扱う．

ディジタルフィルタの設計においては，与えられた仕様を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．ディ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的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．

そこで，われわれはSSDFの高精度性に着目して，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．

本研究では，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 状態空間ディジタルフィルタ

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]．

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = Cx(k) + Du(k) \tag{2}$$

ここで，$x(k)$はn次の状態ベクトル，$u(k)$はr次の入力ベクトル，$y(k)$はm次の出力ベクトルである．また，A，B，C，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ィジタル信号処理システムや制御システムを状態空間ディジタルフィルタ（State-Space Digital Filters）と呼び，これをSSDF[A, B, C, D]と記述する．SSDF[A, B, C, D]の伝達関数$H(z)$はつぎのように得られる．

$$H(z) = C(zI - A)^{-1}B + D \tag{3}$$

ここで，状態ベクトル$x(k)$がシステムの遅延要素の出力を表し，行列A，B，C，Dの各要素がシステムの係数に対応するとき，SSDF[A, B, C, D]は伝達関数$H(z)$を有する状態空間ディ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．

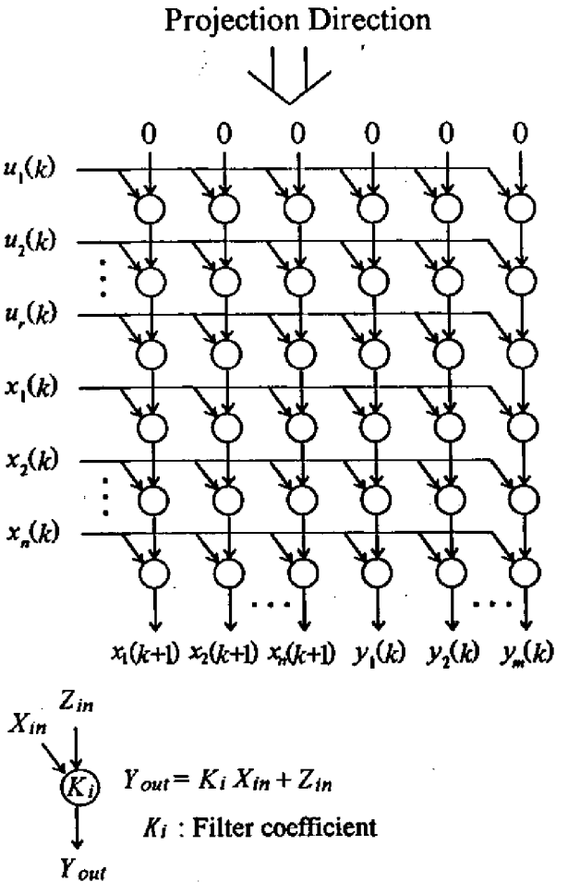

Fig. 1 Data dependence graph for SSDF.

伝達関数$H(z)$を有する状態空間ディ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Tに対して，$x' = T^{-1}x$と状態ベクトルの変換を行うとき，$H(z)$を有する他の構造はSSDF[$T^{-1}AT$, $T^{-1}B$, CT, D]によって一般的に記述される．状態空間ディジタルフィルタでは，適当な等価変換行列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]．従って，状態空間ディ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．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ィジタルフィルタ（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)，

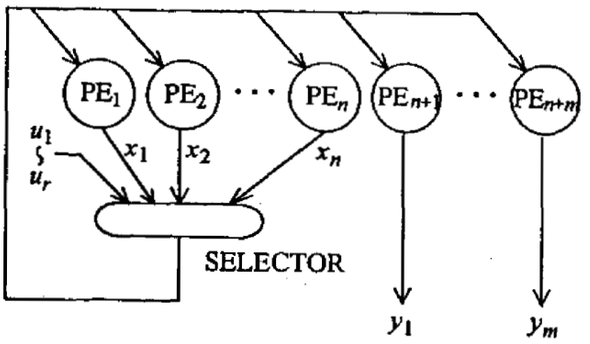

Fig. 2　1-dimensional SIMD array.

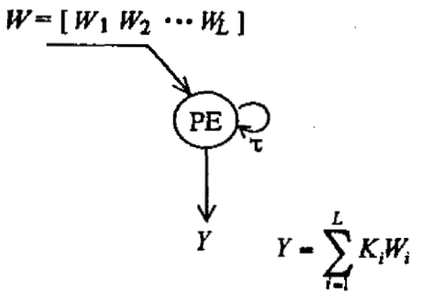

Fig. 3　Function of PE.

(2) 式の状態変数$x(k+1)$および$y(k)$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，…，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}，…，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．出力はラッチ回路によって次のデータが出力されるまで保持される．

(4) 各PEで内積演算を実行する場合，各PE間では同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，従って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$\boldsymbol{a}=(a_1,\ldots,a_N)$と変数ベクトル$\boldsymbol{v}=(v_1,\ldots,v_N)$との内積

$$y=\boldsymbol{a}\boldsymbol{v}=\sum_{i=1}^{N}a_iv_i \tag{4}$$

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．つまり，

$$v_i=-v_i^0+\sum_{k=1}^{B-1}2^{-k}v_i^k \tag{5}$$

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．(5) 式を (4) 式に代入すれば，内

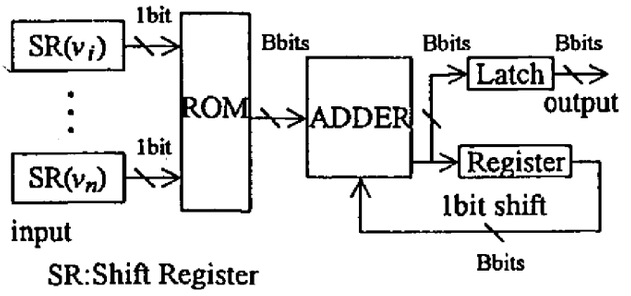

Fig. 4 Basic structure of distributed arithmetic.

積演算avは次式で示される.

$$y = -\Phi(v_1^0, \ldots, v_N^0) + \sum_{k=1}^{B-1} 2^{-k}\Phi(v_1^k, \ldots, v_N^k) \quad (6)$$

ただし, Φは

$$\Phi(v_1^k, \ldots, v_N^k) = \sum_{i=1}^{N} a_i v_i^k \quad (7)$$

である.

(6) 式を実現するには, $(v_1^k, \ldots, v_N^k)$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.

前述したSSDF用システムアーキテクチャでは, SIMD形のプロセッサア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. 従って, 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$y(k)$の各要素が全て$(n+r)$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.

$$x(k+1) = \sum_{j=1}^{B-1} 2^{-j}\,\Phi_p^j - \Phi_p^0$$
$$y(k) = \sum_{j=1}^{B-1} 2^{-j}\,\Phi_q^j - \Phi_q^0 \quad (8)$$

ただし,

$$\Phi_p^j = A\,x^j(k) + B\,u^j(k)$$
$$\Phi_q^j = C\,x^j(k) + D\,u^j(k) \quad (9)$$

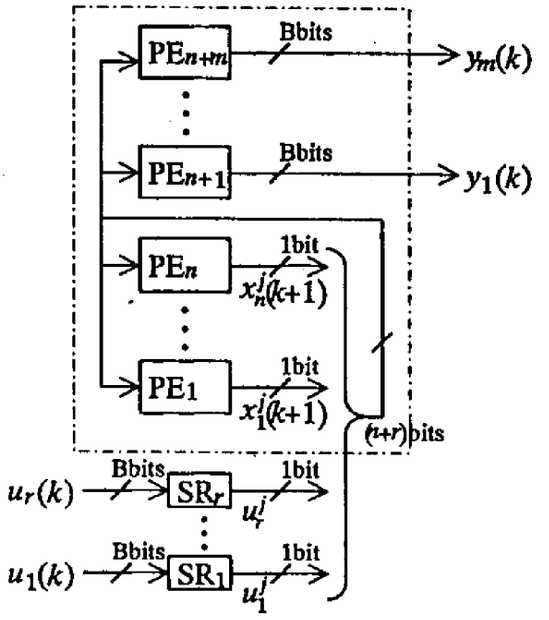

Fig. 5 Configuration of SSDF system using distributed arithmetic.

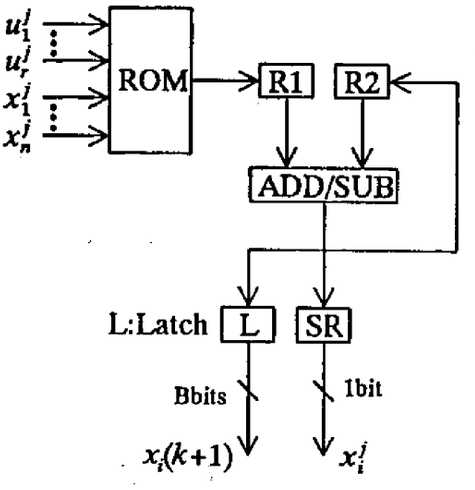

Fig. 6 Basic structure of PE.

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, (8) 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. 従って, 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

Table 1　Filter specifications.

Order	16
Number of input	1
Number of output	1
Data format	14 bit fixed-point

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, 従って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. 次に, 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.

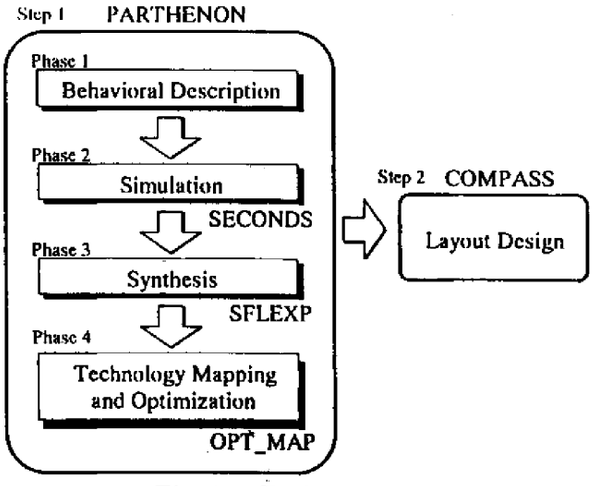

Fig. 7　Design process.

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. ここで論理的な簡単化が行われ, テクノロジ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・マッピングと回路の最適化を行い, テクノロジ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の検討を行

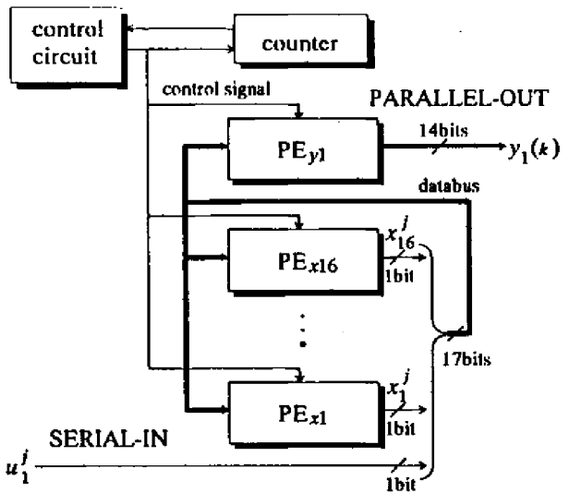

Fig. 8 Block diagram of VLSI processor.

い，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．全てのPEの動作は，1個のカウンタを用いたステート・マシンとしてプロセッサを設計することで容易に制御できる．つまり，プロセッサの動作は規則的であるため，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．従って，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

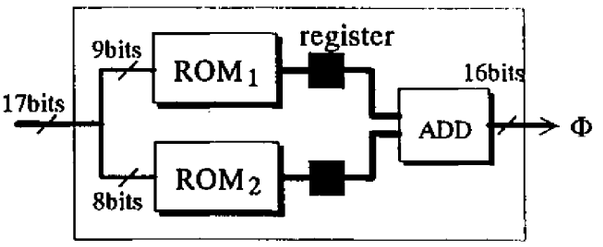

Fig. 9 Memory which devided the function Φ.

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．

$$\begin{aligned}\Phi(v_1^k,\ldots,v_{n+r}^k) &= \sum_{i=1}^{n+r} a_i\cdot v_i^k \\ &= \sum_{i=1}^{\frac{n+r}{Q}} a_i\cdot v_i^k+\cdots+\sum_{i=Q'}^{n+r} a_i\cdot v_i^k \\ &= \Phi(v_1^k,\ldots,v_{\frac{n+r}{Q}}^k)+\cdots \\ &\quad +\Phi(v_{Q'}^k,\ldots,v_{n+r}^k) \quad (10)\end{aligned}$$

ここで，

$$Q'=\frac{(Q-1)(n+r)}{Q}+1 \qquad (11)$$

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の実現に必要となる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ある．従って，PEの関数用メモリはROMを用いて実現する．

4.3 PEの構成

PEの処理は，ビット・シリアルで入力される各桁に対して連続的に行われるため，パイプライン処理を用いて高速化させることが有効である．PEのブロック図をFig. 10に示す．なお，計算時のオーバーフローを防ぐために，PE内部の語長は整数部を2ビット設けた16ビットとしている．

ここで，PEのスケジューリングについて述べる．PEは，アドレス生成部，ROMのアクセス部，加算部，2の補数生成部，演算部の5段のステージで構成される．関数用メモリとして用いるROMは，クロックの立ち上がりでアドレスを内部でラッチし，そのアドレスが示す番地のデータをアクセスタイム分遅れて出力する．そのため，ROMはアドレス生成部とアクセス部による2サイクルで実行される．加算部では，2個のROMの出力値を足し合わせてΦを計算する．なお，加算器はモジュール化した4ビット桁上げ先見加算器を4個用いて実現する．2の補数生成部では，符号ビットをアドレスとして読み出されたΦの値に対して2の補数を計算し，それ以外では処理を行わない．そして演算部では，その値とアキュムレータの値をシフト加算する．ここで，算術シフトの際には有限語長に起因する量子化誤差を考慮して丸め処理を施す．丸めの計算は，加算器の桁上げ入力を利用して実現できる．

このようにPEを設計することによって，パイプラインのピッチはほぼ加算器の遅延時間まで抑えられる．

4.4 回路合成とレイアウト設計

動作記述された各モジュールをSECONDSを用いて動作確認した後，SFLEXPを用いて論理合成する．次に，OPT_MAPを用いてテクノロジ・マッピングと回路の最適化を行う．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，

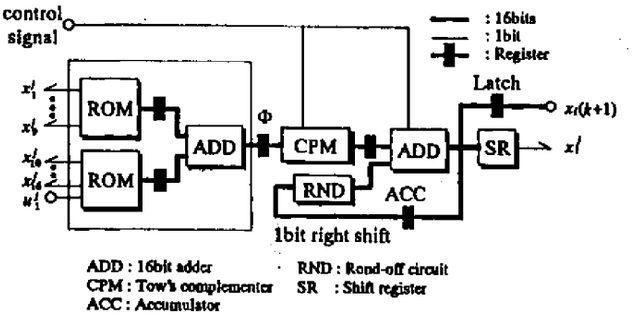

Fig. 10 Block diagram of PE module.

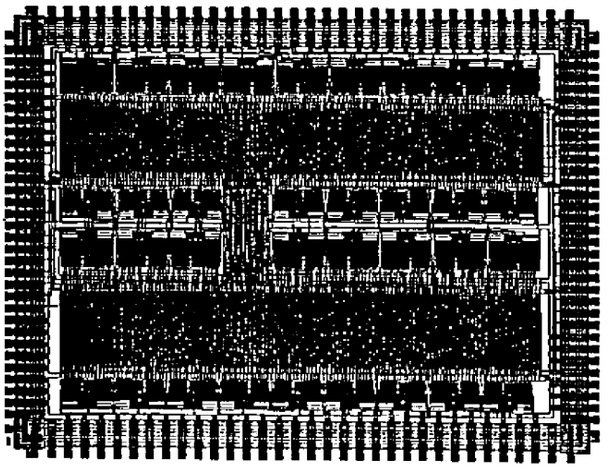

Fig. 11 Chip layout of the VLSI processor.

0.8μmCMOSスタンダードセル（VLSIテクノロジー社）である．回路の最適化では，実部品の物理および制約条件と設計者が設定する設計条件をターゲットとしてセルの置換が行われる．われわれは高速なプロセッサを合成するために，動作マシンサイクルの値をターゲット違反が残らない程度まで小さく設定する．その結果，マシン・サイクル34nsで動作するプロセッサが合成された．

COMPASSでは，PARTHENONによって得られたネットリストを基にして，プロセッサのレイアウト設計を行う．その結果，チップ化されたプロセッサの性能がVLSI評価できる．

5. 性能評価

5.1 レイアウト評価

VLSIプロセッサのレイアウトをFig. 11に示す．また，チップの性能をTable 2に示す．

Table 2　Key performance of the VLSI processor.

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

設計されたVLSIプロセッサは，フィルタの次数が16と高次にもかかわらず，サンプリング周波数1.67MHzという高速処理が可能である．これは，単一のDSPを用いた逐次処理によって実現した場合と比較して，100倍以上（τ =150nsのDSPの場合）も高速である[9]．また，滞在時間も600nsと非常に減少されている．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 × 6.6mm^{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，他の実現法と比べて30倍以上の高速化を達成している．

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えた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．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．従って，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．

6.　おわりに

本研究では，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ィジタルフィルタ用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

Table 3 Parameters of the VLSI processor.

Order	8	16	24	16
Number 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
Execution cycle	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 [gates]	19,170	57,095	118,897	110,163
Number of signal pins	18	18	18	123

わらずサンプリング周波数1.67MHz（0.8μmCMOS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，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．

本研究を進めるにあたり，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，入江洋氏に心より感謝申し上げます．

参考文献

1) S. Y. Hwang: Minimum uncorrelanted 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: 状態空間ディ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ィ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ィ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) 大町慎太郎，恒川佳隆，関　享士郎，志田純一: 状態空間ディジタルフィルタのマルチDSP実現の検討，平成元年度電気関係学会東北支部連合大会