

ディジットシリアル演算器を用いた FIR フィルタの設計と初期評価

山辺 裕樹[†] 谷川 一哉[†] 弘中 哲夫[†][†] 広島市立大学大学院 情報科学研究科 情報工学専攻, 731-3194 広島県広島市安佐南区大塚東 3-4-1
E-mail: tyamabe@csys.ce.hiroshima-cu.ac.jp, ††{tanigawa, hironaka}@ce.hiroshima-cu.ac.jp

あらまし FIR フィルタを従来手法であるパラレル演算器を用いて設計した場合、フィルタ長や演算ビット幅を大きくするとハードウェア資源を使いすぎるといった問題があった。そこで本稿では、高精度な FIR フィルタを可能な限り小面積で実現する方法として、ディジットシリアル演算器を用いた手法を紹介し、従来手法との比較評価を行った。

キーワード ディジットシリアル, ビットシリアル, FIR フィルタ

Initial Evaluation of FIR Filter Based on Digit-Serial Computation

Yuhki YAMABE[†], Kazuya TANIGAWA[†], and Tetsuo HIRONAKA[†][†] Graduate School of Information Sciences, Hiroshima City University
3-4-1, Ohzuka-higasi, AsaMinami-ku, Hiroshima-shi, 731-3194 JapanE-mail: tyamabe@csys.ce.hiroshima-cu.ac.jp, ††{tanigawa, hironaka}@ce.hiroshima-cu.ac.jp

Abstract Increasing the filter-length and bit-width of the FIR filter designed using the conventional parallel operation units, will experience a rapid increase in hardware resources. Therefore, we introduce digit-serial operation units on implementing a FIR filter, as a method to implement a high precision FIR with a long filter-length. With the digit-serial operation units, we can keep the required hardware resources of the FIR filter small. In this paper, we evaluated the difference of the FIR filter using the bit-serial operation units between the conventional one as initial evaluation one.

Key words Digit-Serial, Bit-Serial, FIR Filter

1. はじめに

ディジタル信号処理の技術は通信機器や映像機器などの幅広い分野で使用されており、今日我々が生活するうえで欠かすことができないものとなっている。中でも FIR フィルタは、その構成が単純でありながら信号処理を行う上での重要な役割を担っており、これまで様々な研究が行われている。FIR フィルタのアルゴリズムでは時間変数に対して無限長のデータを用いて算出しているが、これをハードウェアで実装する場合には、無限長のデータを有限長で打ち切らなければならない。そのため、急峻な遮断特性を得るには一般に高い次数をとる必要がある。FIR フィルタでは加算器、乗算器、遅延素子を用いて構成されており、次数を上げるためにはこれらのエレメントを増やす必要があるが、高い次数になればなるほど乗算器の面積が膨大なものになるという深刻な問題があった。

本稿では、このハードウェア量の増加という問題を解決する手法として、ディジットシリアル演算器を FIR フィルタに適用した場合の設計例について紹介する。そして、従来手法であるパラレル演算器を用いた FIR フィルタと面積あたりの動作周

波数を比較評価を行う。

2. ディジットシリアル演算

ディジットシリアル演算とは、1ワードのデータをそれよりも細かい単位に分割し、時間軸に展開して処理を行う演算方法のことである。図1にディジットシリアル加算器の演算例を示す。ここでは1ワード4ビット、ディジット幅1ディジットの演算例を示している。ディジットシリアル演算器は1サイクルの間にディジット幅分だけデータを取り込み演算を行う。演算途中で発生するキャリーはD-FFを用いて同一の演算器に入力している。このようにすることで演算器の共有化を図り小面積化を実現している。なお、本稿で紹介するディジットシリアル演算器のデータ入力は全てLSB側から入力するものである。

2.1 ディジットシリアル演算器の構成

この節では、FIR フィルタで使用するディジットシリアル加算器と乗算器の構成について説明する。

2.1.1 ディジットシリアル加算器

ディジットシリアル加算器の構成を図2に示す[2]。ディジットシリアル加算器の加算部分は全加算器とD-FFという基本的

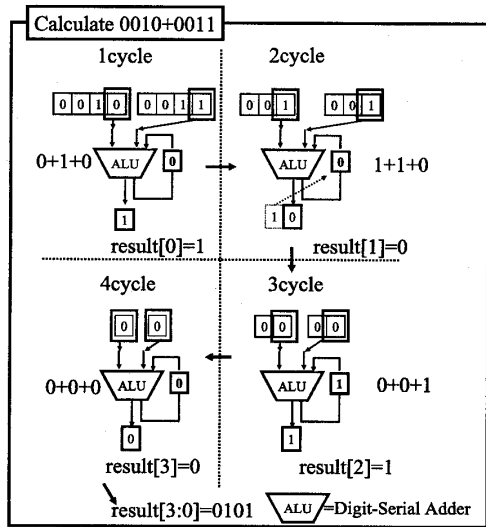


図1 デジットシリアル演算の例 (デジット幅=1)

な構成で実現されている。加算後に発生するキャリーを D-FF で保持し、1 サイクル後に入力されるデータと同時に全加算器へデータを渡すことでデジットシリアル演算を実現している。演算開始はデータの開始位置を示す制御信号 (control_i) をきっかけに行う。詳細なタイミングチャートは 2.2.1 節で述べる。図 2 の右側はオーバーフローを検知する部分である。

2.1.2 デジットシリアル乗算器

デジットシリアル乗算器はデータ入力方法の違いや演算方法の違いにより複数の実現手法が存在する。デジットシリアル乗算器は大別すると“シリアル-シリアル入力”と“シリアル-パラレル入力”の二通りに分けられる。まずシリアル-シリアル入力型の乗算器の構成を図 3 に示す [2]。この構成では、入力される被乗数は 1 サイクル毎に隣の PE へシフトされる。このとき、ctr_{in} 信号に合わせてレジスタへの格納を行い乗数との部分積を行う。乗数は隣の PE にシフトし、そこで再び被乗数との部分積を行う。PE には LSB セル, Inner, MSB という三つのモジュールを使用する。PE の数は、1 ワードのデータサイズと等しい。この乗算器は、データの上位側と下位側の両方を出力可能である。シリアルデータで入力される片方のデータを内部で保持して上位側を出力する構造をとっているため、データを連続して入力することができないが、上位側の出力結

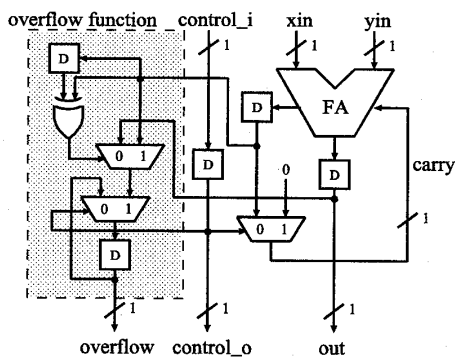


図2 デジットシリアル加算器

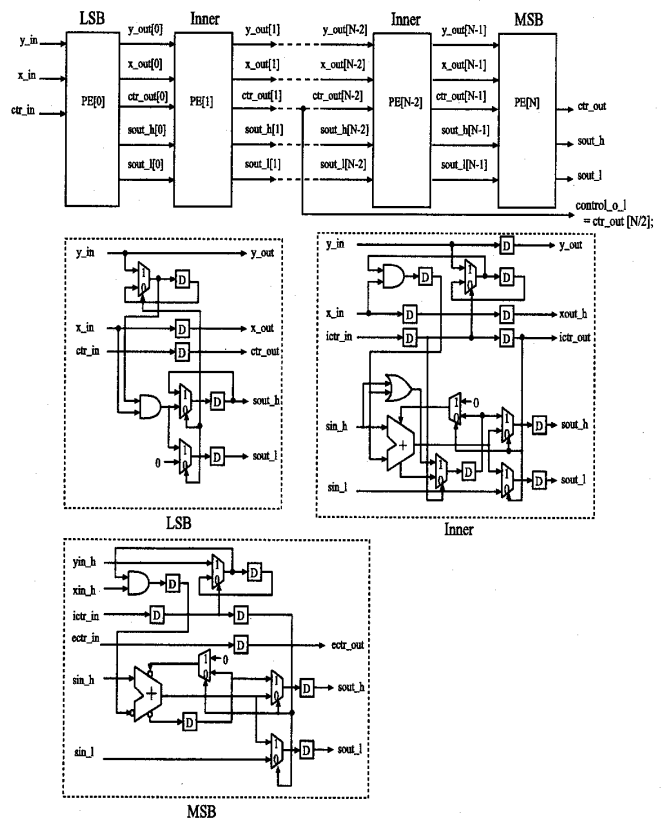


図3 シリアル-シリアル入力型デジットシリアル乗算器

果が必要でない場合にはデータを連続して入力することができる。この場合、終了位置を示す制御信号を次のデータの開始位置を示す信号として共有させる必要がある。この乗算器の構成では、上位側を計算するロジックと下位側を計算するロジックが共有されているため、下位側のみ出力が良い場合でも、上位側のロジックを削除することはできない。

次にシリアル-パラレル入力型デジットシリアル乗算器の構成を図 4 に示す [1]。この乗算器は片方の入力をパラレルデータ、もう片方の入力をシリアルデータで入力する。乗数と被乗数との部分積を隣のセルへ入力し順次処理を行う。図 4 の制御信号 (control_i) はデータが入力される 1 サイクル前に High になり、全ての D-FF の値を初期化する役割を果たしている。デジットシリアル演算器では入力の開始位置と終了位置は制御信号 (control_i) によって制御される。しかしこの乗算器の構成の場合、単純に制御信号をデータの開始と終了で High にしてしまうと、データの終了位置でレジスタの値が初期化され、符号ビットの処理が正常に行われなくなる。そのため点線で囲まれた部分で、開始位置を示す部分のみ High になるように信号を加工する必要がある。この乗算器はシリアル-シリアル入力型の乗算器と異なり、下位側のみ出力するロジックだけで構成することができる。そのため、上位側の計算を行うのに必要なハードウェア量を削除でき、他の乗算器と比較して非常に小面積で構成することができる。この乗算器のタイミングチャートを節 2.2.3 で説明する。

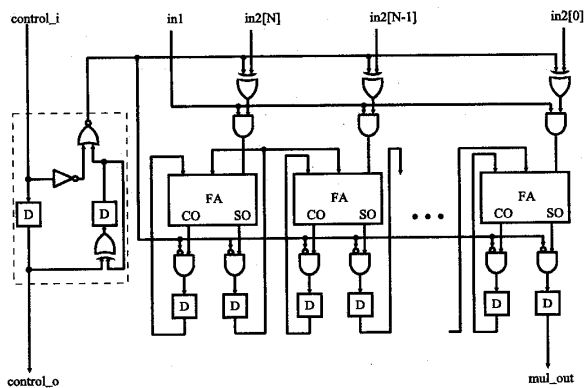


図4 シリアル-パラレル入力型ディジットシリアル乗算器

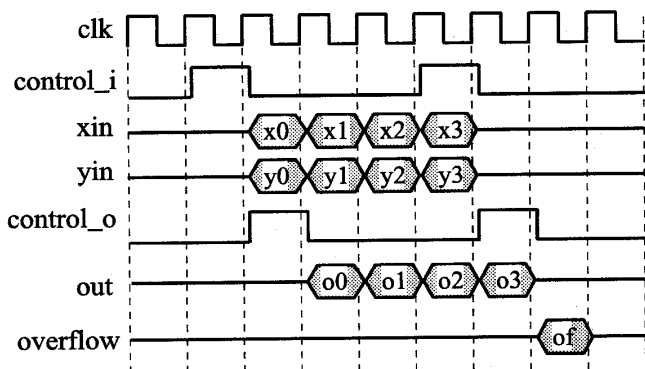


図5 デジットシリアル加算器のタイミングチャート

2.2 デジットシリアル演算器のタイミングチャート

次に、前述した演算器のタイミングチャートについて説明する。今回は紙面の都合上、FIR フィルタで使用するワードサイズのタイミングチャートではなく、1ワード4bitのタイミングチャートを用いて説明する。

2.2.1 デジットシリアル加算器のタイミングチャート

まず、ディジットシリアル加算器のタイミングチャートを図5に示す。この図の信号名は図2と対応している。制御信号(control_i)は図5で示すように、データが入力される1サイクル前とデータの終了時に High にする。出力データは入力データの1サイクル後に出てくるため、出力データの有効範囲を示す制御信号(control_o)は control_i を1サイクル遅らせたものを使用する。

2.2.2 シリアル-シリアル入力型ディジットシリアル乗算器のタイミングチャート

次にシリアル-シリアル入力型ディジットシリアル乗算器のタイミングチャートを図6に示す。乗算結果が出力されているかどうかの判定は、下位側の有効範囲を示す信号は control_o_l, 上位側の有効範囲を示す信号は control_o_h である。乗算結果は入力データが全て入力された後に出力される。これは、図3の乗算器の構造上データの出力が一番最後のセルからになっているためである。よって出力の先頭ビットは入力データ先頭ビットから (PE の数) サイクル後である。

2.2.3 シリアル-パラレル入力型ディジットシリアル乗算器のタイミングチャート

最後にシリアル-パラレル入力型ディジットシリアル乗算器のタイミングチャートを図7に示す。片方のデータをパラレルデータで供給することで、内部的にデータを溜めるような措置が不要になり、演算結果をすぐに出力することができる。そのため、制御信号(control_i)がデータの終了時で High になると、図7のデータ(o3)が0になってしまう。内部では、制御信号(control_i)を加工した control_first を使用する。

3. FIR フィルタの構成

本節では、パラレル FIR フィルタとディジットシリアル FIR フィルタの構成について説明する。FIR フィルタで採用する構成は転置形を採用する。これについての議論は [4] で行っている。

本稿では、パラレル FIR フィルタとディジットシリアル FIR フィルタを面積当たりのスループットで比較を行うため、FIR フィルタと係数データを供給する部分を合わせた全体での面積と動作周波数が必要である。よって今回の論文ではこれらを求められるように、係数データの供給に [4] で紹介したメモリを使用せず、D-FF を用いて実現する。

3.1 パラレル FIR フィルタの構成

図8にパラレル FIR フィルタの構成について示す。

パラレル FIR フィルタで使用する演算器にはシノプス社の DesignWare に登録されている演算器構成の中で、最も面積効率が良い "pparch" を採用した。この実装方法は柔軟性が高く、面積とタイミングの制約条件といったコンテキスト、およびテクノロジライブラリに基づいてダイナミックに生成され

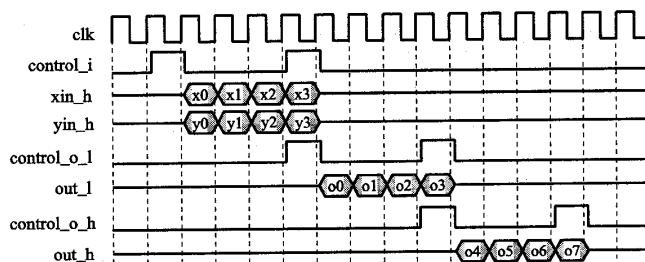


図6 シリアル-シリアル入力型の乗算器のタイミングチャート

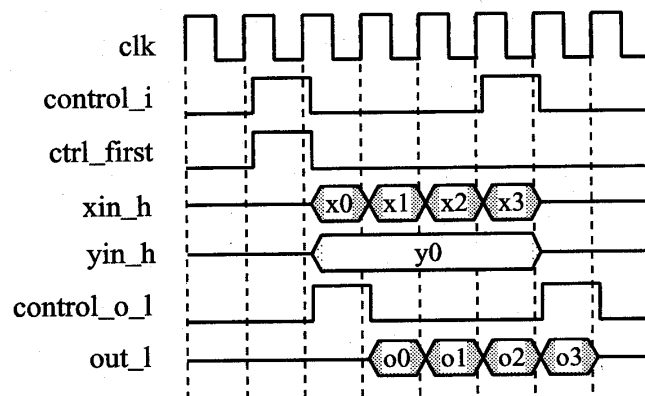


図7 シリアル-パラレル入力型の乗算器のタイミングチャート

る。“pparch”はさまざまな実装方法の特性を利用して、最適なアーキテクチャを生成するものである。

3.2 転置形ディジットシリアル FIR フィルタの構成

次に転置形のディジットシリアル FIR フィルタの構成について説明する。転置形 FIR フィルタの構成例を図 11 に示す。

転置形ディジットシリアル FIR フィルタの種類を入力形式で以下のように識別することができる。これは FIR フィルタの入力が乗算器の入力と等価なためである。

- (1) typeS:シリアル-シリアル入力型
- (2) typeP:シリアル-パラレル入力型

この入力形式の違いにより、図 11 に示すように、FIR フィルタで使用する遅延素子 $[Z^{-1}]$ の数も異なってくる。これはデータを連続的に入力することが可能かそうでないかという違いが関係している。typeS で使用する乗算器は、入力するデータに間隔を入れなければ連続的にデータを処理することができる。逆に typeP で使用する乗算器は、制御信号 (図 4 中の control_i) が内部の D-FF をリセットする機能も兼ねているため、連続的なデータの投入はできず、制御信号 (control_i) の間隔を 1 サイクルあける必要がある (図 10 参照)。以上の理由で、typeS と typeP で遅延素子 $[Z^{-1}]$ の個数に違いが生じている。

typeS と typeP でデータを連続的に入力する場合の乗算器の入出力タイミングチャートをそれぞれ図 9 と図 10 に示す。typeS では、制御信号の開始と終了を共有することができるため、データを間隔をあけることなく入力し続けることができる。この動作はディジットシリアル加算器でも可能なので、出力側の信号が入力側と同じ形式になっても問題ない。

次に、データの投入形式についてだが、typeS の投入形式はシリアル-シリアル乗算器を使用した場合、係数データとフィル

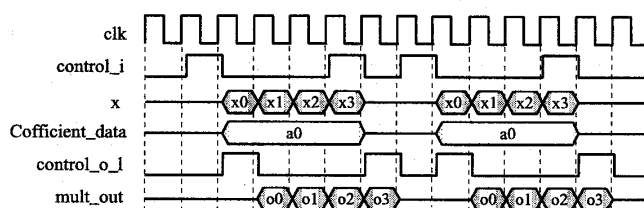


図 10 連続的にデータを投入する場合:typeP の乗算器のタイミングチャート

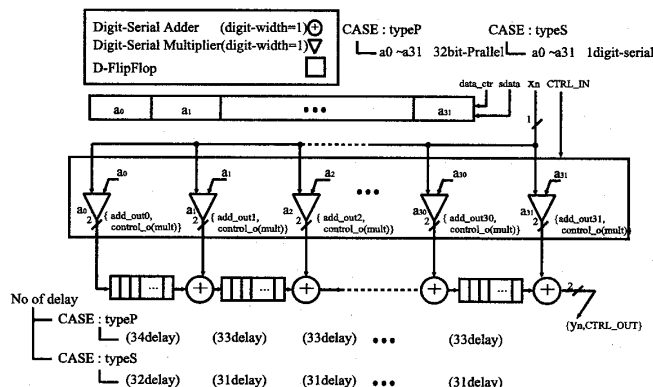


図 11 転置形ディジットシリアル FIR フィルタの構成図

タしたい信号の両方がシリアルデータ転送になる。typeP の場合、シリアル-パラレル乗算器を使用しているため、入力のどちらをどのデータ転送方式にするかを検討する必要がある。係数データの読み込みをシリアルデータ転送にした場合、一演算毎にパラレルデータをシリアルデータに変換しなくてはならない。逆にパラレルデータ転送にした場合、一度メモリにデータを入力後は固定のデータが出力され続けることになる。シリアルデータ転送では毎回読み出しを行わなければならないので、仮にこの部分の動作周波数が FIR フィルタ全体の動作周波数よりも低ければ、この部分がシステムのボトルネックになってしまうという問題がある。以上のことから typeP の FIR フィルタの入力は、係数データをパラレルデータ転送で行い、フィルタしたい信号の方をシリアルデータ転送で行う。

次に、転置形 FIR フィルタで使用するディジットシリアル加算器が使用する制御信号について説明する。この構成で、出力位置を示す制御信号 (control_o) を加算器の方からとると、加算結果と共に遅延素子 $[Z^{-1}]$ をもう一つ用意しなくてはならない。加算器から出力される制御信号 (control_o) が遅延素子を通して次の加算器へ入力する場合のタイミングと、乗算器からの制御信号の High になるタイミングは全く同じである。そのため、加算器側で生成したものを使用せず、乗算器からの制御信号を使用することで遅延素子 $[Z^{-1}]$ をもう一つ用意する必要がなくなる。なお、ディジットシリアル乗算器の点線枠で囲まれた制御回路を全ての乗算器で使用方法を [4] で紹介したが、このような構成にすると制御信号 (図 11, data_ctr) がクリティカルパスとなり、フィルタ長を長くするに従って動作周波数は低下してしまう恐れがある。フィルタ長、回路全体の面積、動作周波数等を考慮して、ディジットシリアル乗算器の点

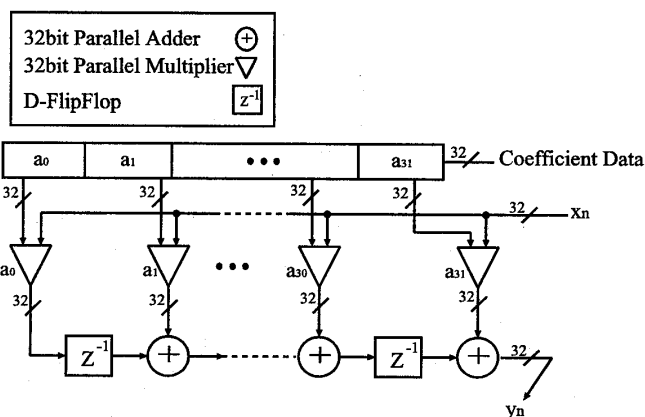


図 8 パラレル FIR フィルタの構成

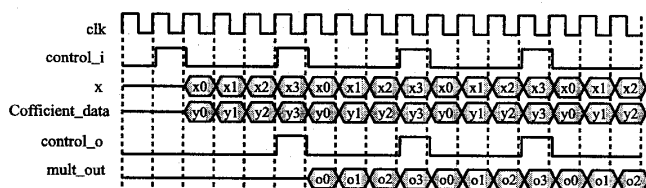


図 9 連続的にデータを投入する場合:typeS の乗算器のタイミングチャート

線枠で囲まれた部分を共有化するよりも個々に制御回路を設けた方が性能が良いと判断したため、今回は各々の乗算器に制御回路を設ける方法を選択した。係数メモリは、コントロール信号の High になるパルスを検知してシリアルデータを取り込み、データを全て取り込んだ後はそのデータを出力し続けるという動作を行う。

4. 評価

本節では、これまで述べてきたディジタルシリアル FIR フィルタと従来手法であるパラレル FIR フィルタの設計評価を行う。評価で使用する環境を表 1 に示す。設計時間の関係上最適化を行うことができなかつたため、自動レイアウトで設定する“Core Utilization”の値は、ディジタルシリアルとパラレルの両方 0.7(70%) にした。

次に評価で使用する FIR フィルタの設計仕様は、ビット幅を 16bit, 32bit, 64bit, 128bit とし、フィルタ長は 31, 63, 127 の 3 通りで行う。また、パラレル FIR フィルタの入出力はパラレルデータ転送、ディジタルシリアル FIR フィルタの入出力はディジタル幅 1 のデータ転送で行うものとする。64bit, 128bit と高いビット幅でも評価を行う理由としては、FIR フィルタで係数データのビット幅が小さい場合、フィルタリングして得られる周波数特性に影響を与えてしまうという問題があり、データのビット幅が増やすことによってフィルタの性能が向上するという結果が得られたためである [3]。よって本稿では、前述したようにビット幅を増やした場合についても同様に比較を行うことにする。

4.1 入力形式毎のディジタルシリアル FIR フィルタの面積、動作周波数の比較

本節ではディジタルシリアル FIR フィルタの面積と動作周波数について比較する。まず、typeS と typeP の FIR フィルタ全体の面積や動作周波数を求め、それらを比較し性能の良い方の type と従来手法との比較を行う。

typeS の面積と動作周波数を表 2 に、typeP の面積と動作周波数を表 3 を示す。この図から、typeS の方が動作周波数は高いが面積の方も大きくなっていることがわかる。ビット幅 16 の場合では、typeP の面積は typeS の面積のおよそ 48% の面積、ビット幅 32 の場合ではおよそ 44% の面積で実現できていることがわかる。typeS の方が動作周波数が 150MHz 以上高くなっているという評価結果がでているが、これは、片方の入力をパラレルデータ転送にしているため、乗算器単体の動作周波数が低下してしまったことが主な原因であると考えられる。

この二つのディジタルシリアル FIR フィルタを、フィルタ長を 32bit と固定した場合の単位面積当たりのスループットで評価したところ、図 12 のようになった。これらの図から、ビッ

表 1 設計使用環境

設計ツール	simvision
論理合成ツール	design_vision
配置配線ツール	Milkyway, Astro
ライブラリ	HITACHI0.18 μ m プロセス

表 2 typeS の場合の面積と動作周波数

ビット幅	フィルタ長	動作周波数 (MHz)	面積 (μm^2)
16bit	31	1111	1839817
32bit		1031	3450675

表 3 typeP の場合の面積と動作周波数

ビット幅	フィルタ長	動作周波数 (MHz)	面積 (μm^2)
16bit	31	952	887356
32bit		826	1510925

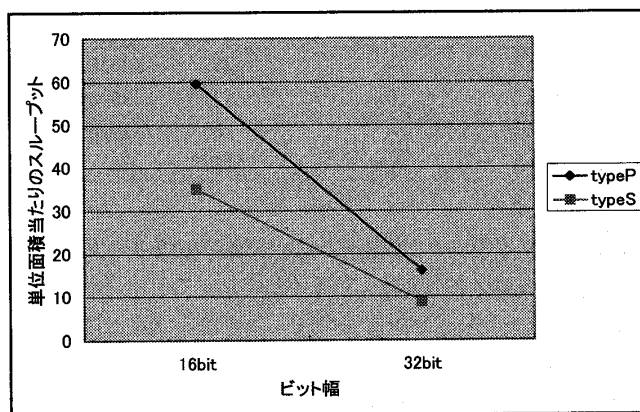


図 12 フィルタ長 31, ビット幅を変更したときの単位面積当たりのスループット

表 4 typeP: フィルタ長, ビット幅を変更した場合の面積と動作周波数

ビット幅	フィルタ長	動作周波数 (MHz)	面積 (μm^2)
16bit	31	952	887356
32bit		826	1510925
64bit		833	3176947
128bit		826	6247973
16bit	63	885	1593622
32bit		813	3141398
64bit		833	6498420
128bit		826	12865126
16bit	127	840	3252972
32bit		833	6672920
64bit		826	13089920
128bit		781	25786083

ト幅を大きくした場合でも変わらず typeP の方が性能が良いといえる。

4.2 パラレル FIR フィルタと

ディジタルシリアル FIR フィルタの性能比較

typeP のディジタルシリアル FIR フィルタで、フィルタ長やビット幅を変えたときの面積と動作周波数を表 4 に示す。次に、パラレル FIR フィルタの場合についての結果を表 5 を示す。

これらの結果から、16bit の場合の面積当たりのスループットを求めてみたところ図 13 のようになった。図 13 のグラフを見ると、従来手法であるパラレル FIR フィルタの方が全てのビット幅で単位面積当たりのスループットが高いということがわかる。ビット幅が高くなるに従って、一見パラレル FIR フィルタとディジタルシリアル FIR フィルタの性能差は縮まってい

表 5 パラレル:フィルタ長, ビット幅を変更した場合の面積と動作周波数

ビット幅	フィルタ長	動作周波数 (MHz)	面積 (μm^2)
16bit	31	262	1408486
32bit		198	4289867
64bit		159	14351971
128bit		138	51563883
16bit	63	262	2911801
32bit		198	8792411
64bit		161	29302715
128bit		138	105152709
16bit	127	258	5823526
32bit		197	18500314
64bit		160	75432695
128bit		138	—

るように思えるが, その減少傾向の比に違いがあまり見られないため, 今後ビット幅を高くすることによってディジットシリアル FIR フィルタの方が性能が良くなるということはないと予想される。

4.3 評価結果に対する考察

本稿では, 単位面積当たりのスループットで従来手法の方が良いという結果が得られた。しかし, 今回評価に使用した対象であるパラレル FIR フィルタで使用する乗算器は, データのワードサイズに合わせて構成を最適なものに実装する方式であった。そのため, 演算器構成が一意に定まっていないという点で, 評価対象として適切ではなかったと考えられる。配置配線の段階で設計時間の関係上”Core Utilization”の値をパラレル FIR フィルタと同じ値で設定していたが, ディジットシリアル FIR フィルタでは, パラレル FIR フィルタに比べてかなり小面積で実現できるため, 配線に使用する領域も少なくすることができる。これにより, さらにこの値をより大きく設定することが可能であると考えられる。そしてディジットシリアル演算器に関しても, 今回はディジット幅を1に設定して評価を行ったため, ビット幅を高くするに従ってサイクル数が非常に多くかかってしまう。ディジット幅の値を増やした場合, ディジット幅1のときよりも遥かにサイクル数を少なくすることができるため, スループットをあげることができると思われる。

5. まとめと今後の課題

本稿では HITACHI0.18 μm プロセスを用いて設計しディジットシリアル演算器を用いた FIR フィルタの設計を行い, 従来手法であるパラレル演算器を用いた FIR フィルタとの比較評価を行った。その結果, パラレル FIR フィルタの方が, typeP のディジットシリアル FIR フィルタよりも, 面積当たりのスループットで優れているということがわかった。本稿ではディジット幅を1に設定したときのみで評価を行ったため, 良い結果が得られなかった。

今後の課題としては以下の4点を行い, 再度設計をして評価を取り直すことが挙げられる。

- (1) 比較対象として用いる乗算器の構成を明確化

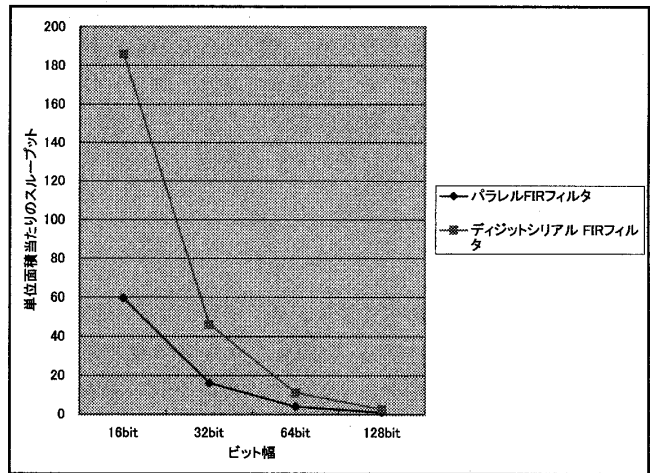


図 13 ビット幅を変更したときの単位面積当たりのスループット

- (2) "Core Utilization"の設定値を見直すこと
- (3) ディジット幅を増やした場合の演算器設計
- (4) 最適なディジット幅の比較調査

謝 辞

本研究は東京大学大規模集積システム設計教育研究センターを通し, シノプシス株式会社, 日本ケイデンス株式会社の協力で行われたものである。

また, 本研究の一部は科学研究費補助金(研究種目:基盤研究(B), 課題番号:18300016)の助成による。

文 献

- [1] Richard I. Hartley, Keshab K. Parhi "DIGIT-SERIAL COMPUTATION", Kluwer Academic Publishers, 1995
- [2] Tsuyoshi Issiki, "High-Performance Bit-Serial Datapath Implementation for Larger-Scale configurable Systems", PhD Dissertation at UCSC, 1996
- [3] 山辺裕樹, 谷川一哉, 弘中哲夫, "高精度な FIR フィルタの実現方法に関する検討", 第9回 IEEE 広島支部 学生シンポジウム 論文集, 2007.
- [4] 山辺裕樹, 谷川一哉, 弘中哲夫, "ディジット幅変換機能付きメモリとその応用", 信学技報, vol. 107, no. 225, RECONF2007-15~31, pp. 1-6, 2007年9月