システムＬＳＩ設計技術　103－21
（2001. 11. 30）

乗算を使わないFIRフィルタの構成

河野 彰†　　柴田 裕一郎‡　　小栗 清‡

†長崎大学大学院生産科学研究科
〒852-8521 長崎市文教町1-14

E-mail: †akawano@cis.nagasaki-u.ac.jp, ‡{shibata,oguri}@ cis.nagasaki-u.ac.jp

あらまし
PCAにおけるビットシリアル方式のFIRフィルタの新しい構造を提案する．乗算器を使わないフィルタはシフト部分，加算部分，分岐部分の3つの基本部分から構成される．これらのオブジェクトがPlastic Cell Architecture (PCA)上でどのように相互に接続され，機能するかを詳細に示す．

キーワード
FIRフィルタ，PCA

The Structure of a Finite Impulse Response Filter without Multiplication

Akira Kawano†, Yuichiro Shibata‡, and Kiyoshi Oguri‡

† Department of Electrical Engineering and Computer Science,
Graduate School of Engineering Science, Nagasaki University
‡ Department of Computer and Information Sciences,
Faculty of Engineering, Nagasaki University
1-14 Bunkyo-machi, Nagasaki, 852-8521, Japan

Abstract
We propose novel structure of an FIR(Finite Impulse Response) filter for PCA(Plastic Cell Architecture) based on an asynchronous bit serial design style. Avoiding the use of multipliers, the filter consists of three types of basic objects; shift objects, addition objects and disjunction objects. We show how these objects are connected with each other and are implemented on PCA in detail. Theorical aspects of reduction of multiplication are also described.

Key words
FIR filter, PCA

1. はじめに

FIR フィルタはディジタル信号処理において広く使用されているフィルタである．通常，DSP(Digital Signal Processor)等を用いてソフトウェア的に実現されるが，フィルタをハードウェアとして構成すれば，より高い処理性能が期待できる．しかし，FIR フィルタをそのままハードウェア化すると，多くの乗算回路が必要となり，ハードウェア量と消費電力の点で不利となる．そこで本稿では，FIR フィルタを，乗算を使用せずに非同期ビットシリアル方式で設計する方法について述べる．これにより省スペース，低消費電力，高速動作が達成できることを述べる．また，これを PCA(Plastic Cell Architecture)上に構成する場合について説明する．

2. FIR フィルタの実現法

FIR フィルタにインパルスを入力した場合，インパルスはシフトレジスタを伝播して 1 箇所のタップのみが非ゼロ数になる．インパルス振幅は 1 なので係数がそのまま順番に出力される形になる．目的とするフィルタのインパルス応答がわかれば，それをそのまま係数とすることによって FIR フィルタが構成できる．

FIR フィルタは Z 変換後の Z 多項式として表すことができる．ここで，

$$a_5Z^{-5}+a_4Z^{-4}+a_3Z^{-3}+a_2Z^{-2}+a_1Z^{-1}+a_0$$

のような FIR フィルタのシステムを図 1 のように考える．

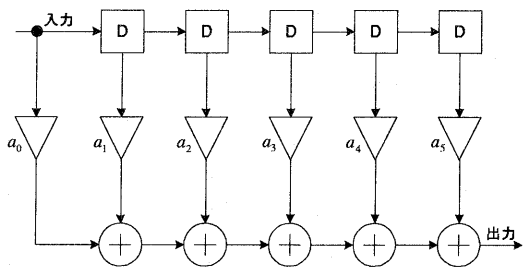

図 1：トランスバーサル・フィルタ

これはトランスバーサルフィルタと呼ばれる．D は遅延を表し，一般的にはシフトレジスタが使用される．最も標準的で式をそのままアルゴリズムに変えただけなのでわかりやすいが，最後に加算が集中するためにハードウェア化した場合に遅いシステムとなってしまう．今回はパイプライン処理するために図 2 のようなアルゴリズムを考える．この方法は精度を保つために途中経過のシフトレジスタの幅を広げる必要があり，メモリをそれだけ消費するという欠点がある．しかし，今回はパイプライン化に重点をおいているためにこの方法を用いる．

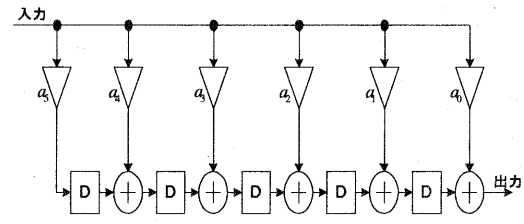

図 2：FIR フィルタのパイプライン処理

3. PCA

PCA(Plastic Cell Architecture)とは構成可能な回路である可変部と，これを制御し通信する組み込み部からなるプラスティックセルを 2 次元メッシュ状に並べたものである．プラスティックセルの可変部は，隣接するプラスティックセルの可変部と回路を作るために接続され，プラスティックセルの組み込み部は隣接するプラスティックセルの組み込み部と通信するために接続され，組み込み部と可変部は，組み込み部から可変部を制御するため，組み込み部が可変部に通信要求を送るため，可変部が組み込み部に通信要求を送るために接続される．図 3 に PCA の構造を示す．

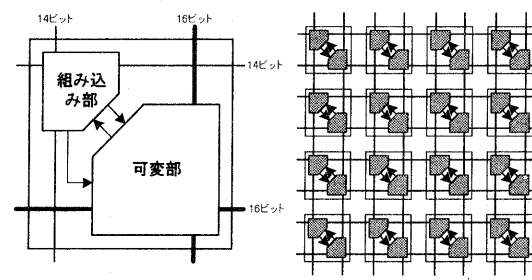

図 3：PCA の構造

3.1 可変部の構造

プラスティックセルの可変部はさらに 8×8 のメ

ッシュ構造となっていて，このメッシュの単位を基本セルと呼ぶ．基本セルは隣接する4つの基本セルとそれぞれ入力1ビット出力1ビットで接続されている．基本セルは合計4ビットの入力と4ビットの出力を持つ．基本セルには64ビットの記憶素子が含まれていて，これで4つのLUT(Look Up Table)を構成している．1つのLUTはアドレス入力4ビット，データ出力1ビットのメモリで，4入力変数までの任意の組み合わせ論理を表現することができる．4つのLUTはそれぞれ東西南北に出力を持ち，これが基本セルの4ビットの出力となっている．これらの4つのLUTの入力は共通に基本セルの4ビットの入力に接続されている．図4に可変部と組み込み部とを接続する場合のピンアサインを示す．

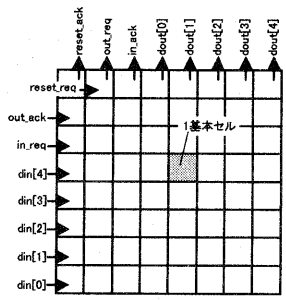

図4：可変部のピンアサイン

3.2 組み込み部の構造

プラスティックセルの組み込み部には，独立に動作する5つの入力ポートがあり，これらは図3右でルーティングスイッチを形成するとともに可変部の制御を行う．5つの入力ポートは同一構造でそれぞれ，東西南北あるいは可変部からの要求を受け付け，入力ポートの状態に応じた処理を行う．要求は10種のコマンドで表現され，これらは5ビットでコーディングされている．図5にコマンドを示す．

clear	0xxxx	経路開放
open	1000x	接続
close	1001x	切断
config_out	1010x	書き込み
config_in	1011x	読み出し
west	11000	経路設定西
north	11001	経路設定北
east	11010	経路設定東
south	11011	経路設定南
pp	111xx	経路設定可変部

図5：コマンド

3.3 コマンド

10種のコマンドのうちwestコマンド，northコマンド，eastコマンド，southコマンド，ppコマンドは経路を設定するためのコマンドである．入力ポートはこれらのコマンドを初期状態で受け取ると，これらのコマンドを消費するとともに後続のコマンドを指定された向きへ転送しつづけるように，すなわちシフトレジスタとなるように設定される．もし指定された向きへの経路がすでに経路として使われていた場合は，経路が開放されるまで待たされる．

3.4 PCASIM

FIRフィルタ設計においては，PCASEと呼ばれる回路設計ソフトを用いた．デバッグ作業はPCASIMと呼ばれるシミュレーター上で行った．PCASIMでは，klファイルを作成し，そのファイル上にPCASE上で作成されたサーキットファイルの読み込みや経路設定，入力データなどの情報を記述することによりシミュレーションを制御することができる．

4. 乗算を使わないFIRフィルタ

4.1 乗算の消去

FIRフィルタを作成する場合には，フィルタの係数とある時間での信号との乗算部分が必ず存在する．この乗算を全て乗算器で演算すると，ハードウェア量が大きくなりすぎてしまう．

今回考案する方法はデータがシリアルに移動することを利用して，もともと加算とシフトで構成される乗算器を分解し，FIRフィルタのための加算を含め全てをシフトと加算で構成するものである．

図6に7×6の動作例を示す．

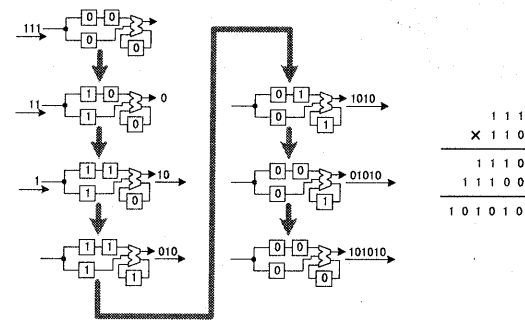

図6：7×6システム図

乗数部分の6は2のべき乗である4と2の和として表すことができる．2進数において，4を掛けるという事は2ビットシフトすることであり，2を掛けるという事は1ビットシフトするという事であるので，シフトした被乗数を加算することにより6を掛けたものと同じことになる．以上のことから，どの様な乗算もシフトと加算で置き換えることができる．

4.2 フィルタの構成

21タップのFIRフィルタを非同期ビットシリアル方式で構成する方法について述べる．

FIRフィルタの式が，

$Z^{-20}-2Z^{-18}Z+8Z^{-14}-16Z^{-12}+18Z^{-10}-16Z^{-8}+8Z^{-6}-2Z^{-2}+1$

で与えられ，入力は16ビットであるとする．この時，演算結果は桁上がりの結果23ビットとなる．このためにシフトレジスタは全て23ビットとした．図7にフィルタの構成図を示す．

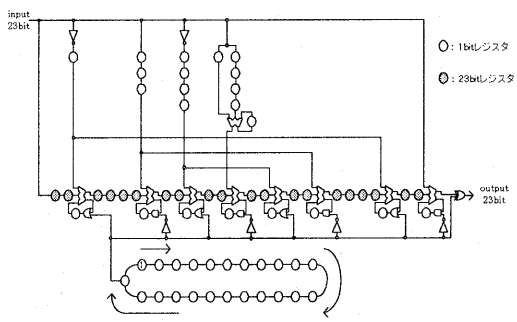

図7：構成図

入力はLSBから1ビットずつ入力されるものとする．網掛けの丸部分は23ビットのレジスタで，入力や演算結果を遅延させるシフトレジスタとなっている．白丸は1ビットのレジスタで，全加算器の部分では桁上げの保持を行う．また，下の環状の23ビットレジスタは，1つだけ1を保持，他はリセットされていて全加算器の桁上げ部分に接続されている．この構造によりフィルタの式の減算部分を表現することが可能となっている．

5. PCAにおける構成

PCAにおいて図7の構成をそのままひとつの回路として設計した場合には，回路が大きくなるために動作速度が非常に遅くなる．そのために回路をシフト部分，合流してからの加算部分，分岐部分の3種類の基本部品(オブジェクト)に分け，それらを複数組み合わせることによりFIRフィルタとして機能させる．今回はシフト部分，加算部分，分岐部分のPCAにおける設計について述べる．

5.1 シフトオブジェクト

図8にPCAにおける4ビットシフトの構成を示す．

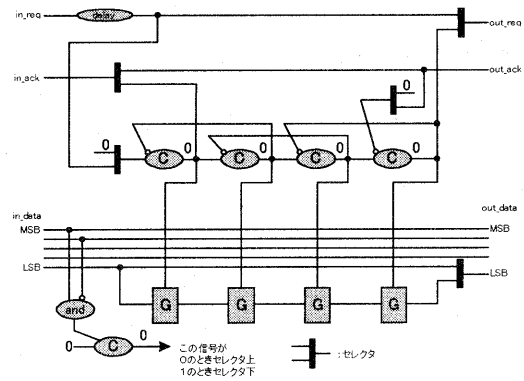

図8：シフト部分

この回路は，入力に10xxxが来るまでは入力を出力に直結，つまり11xxxである経路設定命令はそのまま入力を出力に渡す構造となっている．ただし，動作中に0xxxxは来ないものとしている．入力に10xxxが来るとLSBをシフトさせるシフトレジスタとする．このオブジェクトを複数個連結させ，任意ビットのシフトレジスタを作ることができる．

5.2 加算部分

図9にPCAにおける加算オブジェクトを示す．

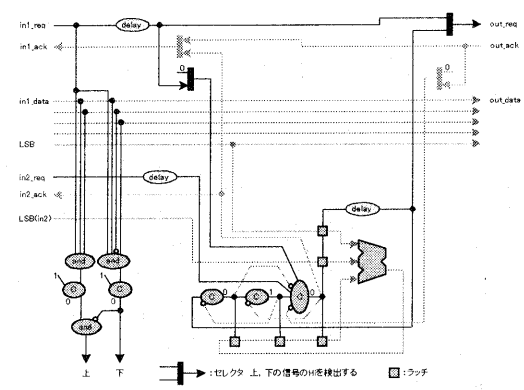

図9：加算オブジェクト

この回路は入力が11xxxのときは，経路設定命令

と判断して入力を出力に直結させる．101xx が入力されたときは2つの入力のLSBを全加算器の入力として加算し，結果をLSBに出力する構成となっている．フィルタの加算部分は全てこの形で表すことができる．

5.3 分岐オブジェクト

まず，分岐オブジェクトの動作の説明をする．始めは経路を南(下)に設定，100xx が入力されると経路を東(左)に設定，そして 101xx が入力されると経路を南と東の両方に設定し，後続のデータを流すという構造になっている．ただし，101xx は東のみに流れるとする．分岐オブジェクトの構造によって分岐の方向は自由に決めることができる．図に分岐オブジェクト動作例を示す．

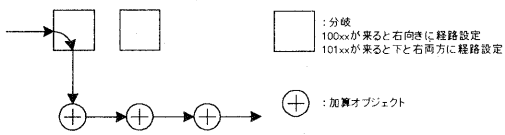

図 10：分岐部分(1)

経路設定コマンド(11xxx)により図 10 の経路を作る．次に 100xx を入力し後続のデータで東向きの経路設定命令を流す(図 11)．

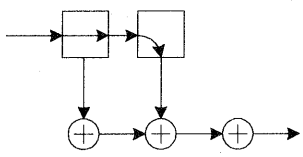

図 11：分岐部分(2)

100xx を入力して 2 つ目の分岐部分の経路を東に向け，後続のデータで南向きの経路設定命令を与える．(図 12)．

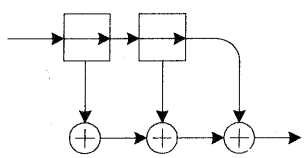

図 12：分岐部分(3)

101xx を入力して 1 つ目の分岐の経路を東と南に設定し，次に 101xx を入力することにより 2 つ目の分岐の経路を東と南に設定する(図 13)．以上で経路の設定は終了である．後はデータを入力すればよい．

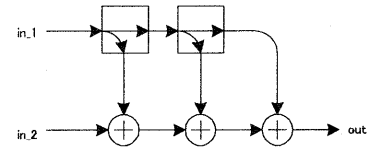

図 13：分岐部分(4)

5.4 PCA への実装

PCASIM への実装の方法について述べる．

まず，多数ビットのシフトはシフト数分のシフト回路を作る方法もあるが，この方法ではオブジェクトの面積が大きくなり，動作速度が遅くなってしまう．別の方法として，複数個のシフト回路を接続することにより多数ビットのシフトを実現させる方法がある．この方法を用いると，オブジェクトの面積を縮小することが可能となり，ビット数が多いオブジェクトにおいては動作速度も速くなる．また，PCA の組み込み部は同時に実行されるコマンドが競合すると待たされるコマンドが発生する．待たされたコマンドは先のコマンドが処理されて空きができると実行される．コマンドは 5 ビットであるが今回の演算ではコマンドの LSB をデータとして処理するので，コマンドが処理 1 つ分待たされることは 1 ビットのシフトとして考えることができる．このことより，組み込み部を 1 ビットのシフトレジスタとして用いることが可能となるので，任意のビット数のシフトも可能となる．以上のように多数ビットを表現するためには一括して作成する方法と分割して組み合わせる方法の 2 種類がある．ビットシフト数，消費電力そして動作速度等を考慮すると一概にどちらの方法が良いとは言えない．目的とするハードウェアに適した方法でシフト回路を構成すべきである．また，組み合わせる場合もどのようなビット数に分割するかによって結果は変わるので，この場合も最適な組み合わせを見つける必要がある．

FIR フィルタの構成は図 7 にもあるように，ほとんどが信号分岐，シフト，加算の繰り返しとなる．これを実現させるためには，シフトオブジェクトの接続と同じようにオブジェクトを複数個接続すればよい．部品は省スペースで収まるように，そして経

路が短くなるように配置して高速動作,省スペース,低消費電力になるようにする.

このように階層的な構成にすることでフィルタを1つの回路として作成した場合よりも高速に動作させ,かつハードウェア量を削減することができる.

6. まとめ

本稿では,FIR フィルタは乗算を用いず加算とシフトで表現できることを示した.

また,PCA における構成を3つの基本部品に分けて,それを組み合わせることによる実現方法を示した.

図 14 に 4 ビットシフトのオブジェクトを示す.サイズとしては可変部4個を使用している.小さな一つの四角と基本セル一つが対応している.

今後は PCA において FIR フィルタを表現して,動作時間の短縮,省電力,省スペース化を実現させる予定である.

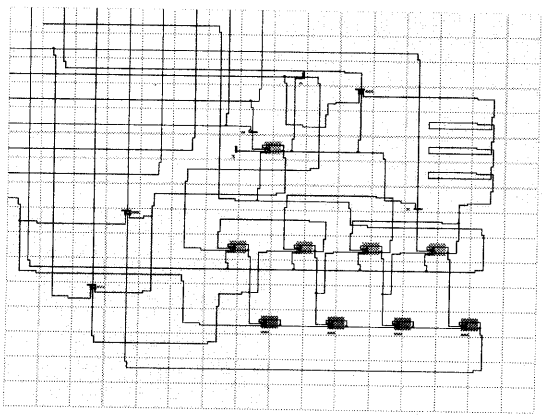

図 14 : 4 ビットシフト

謝辞

本研究を進めるにあたって,貴重な議論や情報の提供をしてくださった(株)エーオーアール 九州 R&D の西村芳一氏に感謝致します.

文献

[1] 小栗清,"布線論理による新しい汎用情報処理アーキテクチャ PCA" bit,Vol.32,No.1,pp.27-35,January 2000.

[2] 小西隆介,小栗清,伊藤秀之,永見康一"LUT アレイ用非同期論理回路の合成法" 情報処理学会研究報告,98-DA-90,pp.107-114,December 1998

[3] 塩澤恒道, Norbert Imlig, 永見康一, 小栗清, "Communicating Logi による汎用情報処理機構の実現" 情報処理学会研究報告,
99-SLDM-93,pp.95-102,November 1999.

[4] 西村芳一,"FIR フィルタ"DSP 処理のノウハウ,pp.37-46, CQ 出版, May 2000.