新しい冗長数表現に基づくディジタル信号処理用FPGAの構成

澤田 善樹　　青木 孝文　　樋口 龍雄

東北大学大学院情報科学研究科

〒 980–8579 仙台市青葉区荒巻字青葉 05

TEL: 022-217-7169 FAX: 022-263-9406

Email: sawada@higuchi.ecei.tohoku.ac.jp

あらまし　本稿では，ディジタル信号処理などで用いられる高速な積和演算回路を構成する際に有用になる新たな数表現として，Signed-Weight Number System(SW 数系： 符号付き重み数系) を提案し，その基本性質および演算アルゴリズムを示す．SW 数系では，演算の各段階で，各桁の重みの符号を自由に設定することにより，規則的なハードウェア構成で正負の数を統一的に表現することが可能である．これにより，2 の補数表現の演算回路に見られるような符号拡張や補正ビットの挿入などの不規則な操作を不要とし，加算アルゴリズムの完全なモジュール化を実現することが可能である．さらに，本稿では，SW 数系の演算アルゴリズムを活用した高速かつコンパクトなディジタルフィルタ専用 FPGA “Field Programmable Digital Filter” の構成を示すとともに，基礎実験の結果についても述べる．

キーワード　FPGA，算術演算，ディジタルフィルタ，積和演算回路，VLSI

Design of Digital Signal Processing FPGA Architecture Using New Redundant Number Representation

Yoshiki SAWADA, Takafumi AOKI and Tatsuo HIGUCHI

Graduate School of Information Sciences, Tohoku University

Aoba-yama 05, Sendai 980–8579, Japan

TEL: 022-217-7169 FAX: 022-263-9406

Email: sawada@higuchi.ecei.tohoku.ac.jp

Abstract　This paper presents “Signed-Weight Number System (SW Number System)”, which is useful for designing high-speed multiply-add modules for digital signal processing applications. The SW number system makes possible the unified representation of positive and negative numbers by adjusting the signs of individual digit positions. Compared with the conventional two's complement representation, the SW number representation makes possible the construction of highly regular arithmetic circuits without introducing irregular arithmetic operations, such as sign extension and LSB compensation. This paper also presents the design of a Field Programmable Digital Filter (FPDF) IC – a special-purpose FPGA for high-speed FIR filtering – based on the proposed SW arithmetic. The result of the experimental fabrication of test circuits is also described.

key words　FPGA, Computer arithmetic, Digital Filter, Multiply-add operations, VLSI

1 まえがき

マルチメディア通信を中心とした現在のディジタル信号処理応用においては，汎用のマイクロプロセッサの演算能力では不十分な場合が多く，用途に応じたASIC (Application Specific IC) の開発がますます重要になりつつある．しかしながら，通常，ASICの開発には膨大なコストと開発期間が要求される．しかも，設計仕様の変更に対する柔軟性に欠けるという問題があった．これに対して，FPGA (Field Programmable Gate Array) などに代表されるようなユーザがハードウェアがハードウェアの構成を柔軟にプログラム可能なデバイスは，今後，上記のような問題に対する画期的ブレークスルーを与える可能性を有する．しかしながら，現在実用化されているFPGAは，任意のランダムロジックをマッピングすることを前提としているため，ディジタル信号処理応用においては，ASICと比較して十分な性能を達成できなかった．そこで，もしFPGAのような柔軟性とともにASICに近い高性能を有する信号処理FPGAが開発されれば，きわめて広範な応用において有用になるものと予想される [1],[2].

このようなディジタル信号処理用のFPGAの開発には，高速性とコンパクト性を兼ね備えたハードウェアアルゴリズムの考案が鍵になると考えられる．本稿では，特にディジタル信号処理などで用いられる高速な積和演算回路を構成する際に有用になる新たな数表現として，Signed-Weight Number System(SW 数系) を提案し，その基本性質および演算アルゴリズムを示す．SW 数系では，演算の各段階で，各桁の重みの符号を自由に設定することにより，規則的なハードウェア構成で正負の数を統一的に表現することが可能である．これにより，2の補数表現の演算回路に見られるような符号拡張や補正ビットの挿入などの不規則な操作を用いずに，加算アルゴリズムの完全なモジュール化を実現することが可能である．さらに，本稿では，SW 数系の演算アルゴリズムを活用した高速かつコンパクトなディジタルフィルタ専用FPGA “Field Programmable Digital Filter” の構成を示すとともに，基礎実験の結果についても述べる．提案するFPDFにおいては，SW 数系に基づく基本算術ブロックを用いることにより，規則的構造でかつ桁上げ伝播のないFIRフィルタリング動作を実行することが可能である．

2 Signed-Weight (SW) 数系とその加算アルゴリズム

2の補数表現された数を入力とするCanonic Signed-Digit (CSD) 係数乗算器を構成する場合，加算器の入力の任意の位置に正および負の数が不規則に入力される．このため，各加算器で入力符号を自由に設定できるような加算アルゴリズムが実現できれば効率的に乗算回路を構成できる．そこで，正負の数を統一的に扱うことを目的として，2進数系の重み2^iに$\lambda_i(\in\{-1,+1\})$のような符号を付加したSigned-Weight (SW) 数系と呼ぶ新しい数系を定義する．

2.1 SW 数系の定義と性質

まず，本稿で提案するSigned-Weight (SW) と呼ぶ数系を記述するため，ベクトルによる数系の表記方法について述べる [3]．一般に重み数系は，各桁のとり得る値の集合D，各桁の符号のベクトル$\boldsymbol{\Lambda}=(\lambda_{n-1},\cdots,\lambda_i,\cdots,\lambda_0)$，各桁の重みの絶対値のベクトル$\boldsymbol{W}=(w_{n-1},\cdots,w_i,\cdots,w_0)$の組$\langle D,\boldsymbol{\Lambda},\boldsymbol{W}\rangle$で表現できる．すなわち$\langle D,\boldsymbol{\Lambda},\boldsymbol{W}\rangle$の数表現をとる$n$桁の数$\boldsymbol{X}=(x_{n-1}\cdots x_i\cdots x_0)$は次式で表される．

$$\boldsymbol{X}=\sum_{i=0}^{n-1}x_i\lambda_i w_i \tag{1}$$

例えば2の補数表現された数の場合 ($n=4$とする) では，

$$\begin{aligned} D_{2C} &= \{0,1\} \\ \boldsymbol{\Lambda}_{2C} &= (-1,1,1,1) \\ \boldsymbol{W}_{2C} &= (2^3,2^2,2^1,2^0) \end{aligned} \tag{2}$$

のように表現され，その数$\boldsymbol{X}=(x_3,x_2,x_1,x_0)$は，

$$\boldsymbol{X}=-x_3 2^3+x_2 2^2+x_1 2^1+x_0 2^0 \tag{3}$$

となる．2の補数は最上位ビットの符号が-1，それ以下のビットの符号が$+1$であり，語長によって一意に定まる符号ベクトル$\boldsymbol{\Lambda}_{2C}$を持った数であることがわかる．

次に，本稿で提案するSigned-Weight (SW) 数系と呼ばれる数表現を上記の表記方法に従って定義する．

定義 1 Signed-Weight 数系$\langle D_{SW},\boldsymbol{\Lambda}_{SW},\boldsymbol{W}_{SW}\rangle$は，次のように定義される重み数系である．

$$\begin{aligned} D_{SW} &= \{0,1\} \\ \boldsymbol{\Lambda}_{SW} &= (\lambda_{n-1},\lambda_{n-2},\cdots,\lambda_1,\lambda_0) \\ &\quad \text{ただし}\lambda_i\in\{-1,+1\} \\ \boldsymbol{W}_{SW} &= (2^{n-1},2^{n-2},\cdots,2^1,2^0) \end{aligned} \tag{4}$$

ここで，SW 数系$\langle D_{SW},\boldsymbol{\Lambda}_{SW},\boldsymbol{W}_{SW}\rangle$で表現された数$\boldsymbol{X}=(x_{n-1}\cdots x_i\cdots x_0)$を表記上以下のように表わす．

Λ_{SW}	λ_{n-1}	λ_{n-2}	$\cdots$	λ_i	$\cdots$	λ_2	λ_1	λ_0
W_{SW}	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^2	2^1	2^0
	x_{n-1}	x_{n-2}	$\cdots$	x_i	$\cdots$	x_2	x_1	x_0

(5)

ここで$\lambda_i\in\{-1,+1\}, x_i\in D_{SW}(=\{0,1\})$である．すなわち数$\boldsymbol{X}$の実際の値は次式で与えられる．

$$\boldsymbol{X}=\sum_{i=0}^{n-1}x_i\lambda_i 2^i \tag{6}$$

□

このように，SW 数系は各桁の重みの符号$\lambda_i(\in\{-1,+1\})$が独立に調整可能な2進数系として定義される．このため2の補数表現は，最上位の重みの符号が負であるようなSW 数系の特殊な場合と考えることができる．SW 数系の各桁の重みに付加される符号λ_iは，演算の格段において，自由に設定可能であるとする．

以上で，SW 数系の形式的な定義を与えた．次にSW 数$\boldsymbol{X}$の取りうる値の範囲 (ダイナミックレンジ) について述べる．

性質 1 SW 数系 $\langle D_{SW}, \boldsymbol{\Lambda}_{SW}, \boldsymbol{W}_{SW} \rangle$ で表現された数$\boldsymbol{X}$のダイナミックレンジを$[DR_{min}, DR_{max}]$とすると，最小値 DR_{min}，最大値 DR_{max}は以下のように与えられる．

$$DR_{min} = -\sum_{i=0}^{n-1} \delta(\lambda_i + 1)2^i \tag{7}$$

$$DR_{max} = \sum_{i=0}^{n-1} \delta(\lambda_i - 1)2^i \tag{8}$$

$$\text{ただし} \delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$$

□

すなわち，SW 数系のダイナミックレンジは符号ベクトル $\boldsymbol{\Lambda} = (\lambda_{n-1}, \cdots, \lambda_i, \cdots, \lambda_0)$ によって決定される．

本数系の最も重要な特長は，演算の各段階で数系の符号ベクトル$\boldsymbol{\Lambda}_{SW}$を自由に設定することにより，規則的なハードウェア構成で正負の数を統一的に表現できる点にある．これにより，結果的に 2 の補数表現に基づく多入力加減算器に見られるような符号拡張や補正ビットのなどの演算回路の規則性を乱す要因を全く不要とし，完全なモジュール化を実現することが可能である．

以上のように SW 数系は規則的な演算回路を構成するために理想的な性質を備えているが，その一方で符号ベクトル$\boldsymbol{\Lambda}_{SW}$に応じて格段の加減算アルゴリズムに若干の修正が必要になる．後で述べるように，SW 数系の典型的な加算アルゴリズムは，(i) SW 数系から通常の符号なし 2 進数系へのダイナミックレンジの変換，(ii) 通常の符号なし 2 進数系による加算，(iii) 通常の 2 進数から SW 数系への逆変換，という形式を取る．このため，(i)，(iii) のオーバーヘッドが低いことが実用上非常に重要になるが，これを保証する性質として次があげられる．

性質 2 次式で与えられる SW 数$\boldsymbol{X}$

Λ_{SW}	λ_{n-1}	λ_{n-2}	$\cdots$	λ_i	$\cdots$	λ_1	λ_0
W_{SW}	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	x_{n-1}	x_{n-2}	$\cdots$	x_i	$\cdots$	x_1	x_0

(9)

および，SW 数$\boldsymbol{X}'$

Λ'_{SW}	λ_{n-1}	λ_{n-2}	$\cdots$	$-\lambda_i$	$\cdots$	λ_1	λ_0
W_{SW}	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	x_{n-1}	x_{n-2}	$\cdots$	$\overline{x_i}$	$\cdots$	x_1	x_0

(10)

の間には，

$$\boldsymbol{X} = \boldsymbol{X}' + \lambda_i 2^i \tag{11}$$

なる関係がある．ただし，$\overline{x}_i$はx_iの否定 (not) である．ここで，$\lambda_i 2^i$のことを$\boldsymbol{X}$から$\boldsymbol{X}'$への変換に伴う「バイアス」と呼ぶ．

(証明)

$$\begin{aligned} \boldsymbol{X} - \boldsymbol{X}' &= \sum_{j=-1}^{n-1} x_j \lambda_j 2^j \\ &\quad - \left(\sum_{k=-1}^{i-1} x_k \lambda_k 2^k - \overline{x}_i \lambda_i 2^i + \sum_{l=i+1}^{n-1} x_l \lambda_l 2^l \right) \\ &= (x_i + \overline{x}_i)\lambda_i 2^i \\ &= \lambda_i 2^i \end{aligned}$$

□

このことは，SW 数系$\boldsymbol{\Lambda} = (\lambda_{n-1}, \cdots, \lambda_i, \cdots, \lambda_0)$ の数 $\boldsymbol{X}$ はその i 桁目の符号λ_iを反転させた SW 数$\boldsymbol{X}'$に変数 x_iの否定を施すことにより容易に変換することができることを意味している．ただし，その際に変換に伴う「バイアス」$\boldsymbol{X} - \boldsymbol{X}' = \lambda_i 2^i$が問題になるが，これは適宜再変換プロセスを加えることにより，補正することが可能である．

例 1 SW 数$\boldsymbol{X}$を

Λ_{SW}	$+1$	$+1$	$+1$	$+1$
W_{SW}	2^3	2^2	2^1	2^0
	1	1	1	1

(12)

とし，3 桁目の符号とディジットを反転させた SW 数$\boldsymbol{X}'$を

Λ'_{SW}	$+1$	-1	$+1$	$+1$
W_{SW}	2^3	2^2	2^1	2^0
	1	0	1	1

(13)

とする．このとき$\boldsymbol{X}$および$\boldsymbol{X}'$は，次式で与えられる．

$$\begin{aligned} \boldsymbol{X} &= 2^3 + 2^2 + 2^1 + 2^0 \\ \boldsymbol{X}' &= 2^3 + 2^1 + 2^0 \end{aligned}$$

$\boldsymbol{X}$から$\boldsymbol{X}'$への変換に伴うバイアスは，

$$\boldsymbol{X} - \boldsymbol{X}' = 2^2$$

であり，**性質 2** を満していることがわかる．□

2.2 SW 数系の加算アルゴリズム

これまで，SW 数の定義および基本性質について述べてきた．以下では，SW 数系の多入力加算アルゴリズムについて述べ，さらに，本加算アルゴリズムを通常の 2 進数系の加算器で実現するための方法を示す．

定義 2 SW 数系の 3-2 カウンタは，入力される 3 つの SW 数$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$をそれぞれ

Λ_x	λ^x_{n-1}	λ^x_{n-2}	$\cdots$	λ^x_i	$\cdots$	λ^x_1	λ^x_0
W_x	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	x_{n-1}	x_{n-2}	$\cdots$	x_i	$\cdots$	x_1	x_0

(14)

Λ_y	λ^y_{n-1}	λ^y_{n-2}	$\cdots$	λ^y_i	$\cdots$	λ^y_1	λ^y_0
W_y	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	y_{n-1}	y_{n-2}	$\cdots$	y_i	$\cdots$	y_1	y_0

(15)

Λ_z	λ^z_{n-1}	λ^z_{n-2}	$\cdots$	λ^z_i	$\cdots$	λ^z_1	λ^z_0
W_z	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	z_{n-1}	z_{n-2}	$\cdots$	z_i	$\cdots$	z_1	z_0

(16)

と表し，2 つの出力$\boldsymbol{U}, \boldsymbol{V}$を

Λ_u	λ^u_n	λ^u_{n-1}	$\cdots$	λ^u_{i+1}	$\cdots$	λ^u_2	λ^u_1
W_u	2^n	2^{n-1}	$\cdots$	2^{i+1}	$\cdots$	2^2	2^1
	u_n	u_{n-1}	$\cdots$	u_{i+1}	$\cdots$	u_2	u_1

(17)

Λ_v	λ^v_{n-1}	λ^v_{n-2}	$\cdots$	λ^v_i	$\cdots$	λ^v_1	λ^v_0
W_v	2^{n-1}	2^{n-2}	$\cdots$	2^i	$\cdots$	2^1	2^0
	v_{n-1}	v_{n-2}	$\cdots$	v_i	$\cdots$	v_1	v_0

(18)

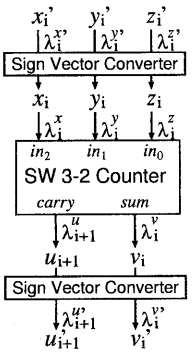

図 1: SW 数 3-2 カウンタ (性質 3 の説明)

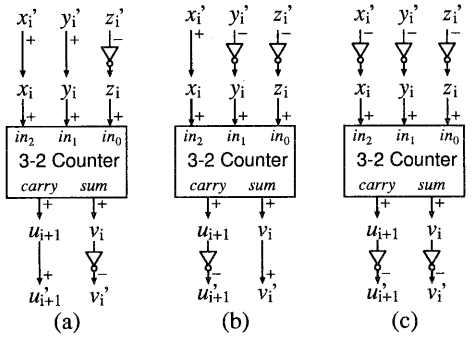

図 2: 通常の 3-2 カウンタによる SW 数 3-2 カウンタの実現

としたとき，次式で定められる加算を実行する．

$$2u_{i+1}\lambda^u_{i+1} + v_i\lambda^v_i = x_i\lambda^x_i + y_i\lambda^y_i + z_i\lambda^z_i \qquad (19)$$

ここで，$x_i, y_i, z_i, u_i, v_i \in \{0,1\}$，かつ，$\lambda^x_i, \lambda^y_i, \lambda^z_i, \lambda^u_i, \lambda^v_i \in \{-1,+1\}$ である．ただし，式 (19) においてカウンタ出力 $\boldsymbol{U},\boldsymbol{V}$ の符号ベクトルは入力 $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$ の符号ベクトルにより，一意に決定され，以下の条件を満すことが必要である．

$$2\lambda^u_{i+1} + \lambda^v_i = \lambda^x_i + \lambda^y_i + \lambda^z_i \qquad (20)$$

□

たとえば，通常の符号なし 2 進数系の場合，$\lambda^x_i = \lambda^y_i = \lambda^z_i = \lambda^u_{i+1} = \lambda^v_i = +1$ であり，常に上式の条件を満す特別な場合と考えられる．

このように，SW 数系の 3-2 カウンタにおいては，入力変数の符号ベクトルに応じて出力変数の符号ベクトルが決定されるが，これは物理的には入力変数のダイナミックレンジに応じて出力変数のダイナミックレンジが決定されることに対応する．式 (20) を満すような入出力の符号の組合わせは，実際には以下の 4 通りである．

- $\lambda^x_i, \lambda^y_i, \lambda^z_i$ がすべて $+1$ のとき，$\lambda^u_{i+1} = \lambda^v_i = +1$
- $\lambda^x_i, \lambda^y_i, \lambda^z_i$ のうち 1 つのみが -1 のとき，$\lambda^u_{i+1} = +1, \lambda^v_i = -1$
- $\lambda^x_i, \lambda^y_i, \lambda^z_i$ のうち 2 つが -1 のとき，$\lambda^u_{i+1} = -1, \lambda^v_i = +1$
- $\lambda^x_i, \lambda^y_i, \lambda^z_i$ がすべて -1 のとき，$\lambda^u_{i+1} = \lambda^v_i = -1$

性質 3 符号ベクトルが $\boldsymbol{\Lambda}_x, \boldsymbol{\Lambda}_y, \boldsymbol{\Lambda}_z$ で与えられる 3 つの SW 数 $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$ を入力変数とする特定の 3-2 カウンタが存在する場合，これを用いて任意の符号ベクトル $\boldsymbol{\Lambda}'_x, \boldsymbol{\Lambda}'_y, \boldsymbol{\Lambda}'_z$ を有する 3 つの SW 数 $\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{Z}'$ を入力する 3-2 カウンタを構成することができる．図 1にその i 桁目の構成を示す．

(証明)

カウンタの入力において，性質 2 を用いて符号ベクトル変換を行い，入力変数の符号ベクトル $\boldsymbol{\Lambda}_x, \boldsymbol{\Lambda}_y, \boldsymbol{\Lambda}_z$ を，$\boldsymbol{\Lambda}'_x, \boldsymbol{\Lambda}'_y, \boldsymbol{\Lambda}'_z$ に変換 (ダイナミックレンジ変換) し，さらに，この変換のバイアス (ダイナミックレンジの差) を出力側でキャンセルするように符号ベクトルの再変換を行う．カウンタ回路の場合，このようなバイアスをキャンセルするような再変換が常に可能であることが証明できる．□

例 2 性質 3 において $\boldsymbol{\Lambda}_x = \boldsymbol{\Lambda}_y = \boldsymbol{\Lambda}_z = (+1,\cdots,+1)$ であるような 3-2 カウンタ，つまり，通常の 2 進数 3-2 カウンタを用いて任意の符号ベクトル $\boldsymbol{\Lambda}'_x, \boldsymbol{\Lambda}'_y, \boldsymbol{\Lambda}'_z$ を持った 3 つの SW 数 $\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{Z}'$ を入力とする 3-2 カウンタを構成するための符号ベクトル変換器の構成を示す．すでに述べたように，$\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{Z}'$ の i 桁目の符号の組合わせは 4 通りある．このうち，$\lambda^{x'}_i = \lambda^{y'}_i = \lambda^{z'}_i = +1$，つまり符号変換が必要ない場合を除いた 3 通りを図 2に示す．□

定義 3 SW 数系の 4-2 カウンタは，入力を $\boldsymbol{W},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$，出力を $\boldsymbol{U},\boldsymbol{V}$ とし，さらにキャリーを $\boldsymbol{C}$ としたとき，次式で定められる加算を実行する．

$$\begin{aligned} &2c_{i+1}\lambda^c_{i+1} + 2u_{i+1}\lambda^u_{i+1} + v_i\lambda^v_i \\ &\quad = c_i\lambda^c_i + w_i\lambda^w_i + x_i\lambda^x_i + y_i\lambda^y_i + z_i\lambda^z_i \end{aligned} \qquad (21)$$

ただし，各桁において入出力の符号は以下の条件を満すことが必要である．

$$2\lambda^c_{i+1} + 2\lambda^u_{i+1} + \lambda^v_i = \lambda^c_i + \lambda^w_i + \lambda^x_i + \lambda^y_i + \lambda^z_i \qquad (22)$$

□

性質 4 符号ベクトルが $\boldsymbol{\Lambda}_w, \boldsymbol{\Lambda}_x, \boldsymbol{\Lambda}_y, \boldsymbol{\Lambda}_z$ で与えられる 4 つの SW 数 $\boldsymbol{W},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$ を入力変数とする特定の 4-2 カウンタが存在する場合，これを用いて任意の符号ベクトル $\boldsymbol{\Lambda}'_w, \boldsymbol{\Lambda}'_x, \boldsymbol{\Lambda}'_y, \boldsymbol{\Lambda}'_z$ を有する 4 つの SW 数 $\boldsymbol{W}',\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{Z}'$ を入力する 4-2 カウンタを構成することができる．□

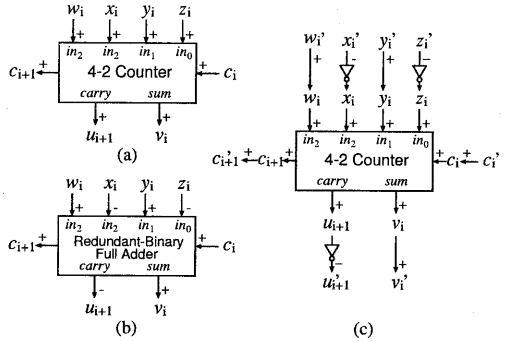

図 3: SW 数 4-2 カウンタ

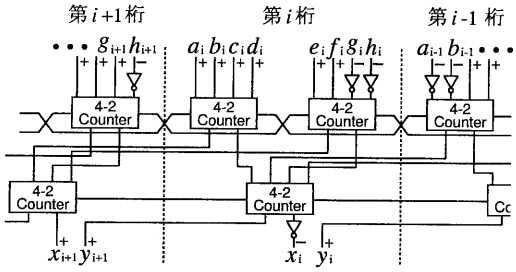

図 4: 2 進数 4-2 カウンタによる SW 数加算木の構成

例 3 以上のような SW 数系の表現を用いると，従来，まったく別個のものとして議論されていた，各種の carry-free 加算器の原理を統一的に議論することが可能となる．例えば，式 (21) のような 4 入力 2 出力カウンタの形式をとる加算器として，通常の 2 進数系の 4-2 カウンタと冗長 2 進全加算器 (Redundant Binary Full Adder) があげられる．これらは，ともに，SW 数 4-2 カウンタの特殊な場合として位置づけることができる．具体的には，図 3(a) に示す通常の 4-2 カウンタは，4 つの入力変数の符号がすべて正 (+1) の場合であり，一方，同図 (b) に示す冗長 2 進全加算器は，2 入力の符号が正 (+1) で，他の 2 入力の符号が負 (−1) であるような特殊な SW 数 4-2 カウンタとみなすことが可能である．さらに，符号ベクトル変換のテクニックを用いると両者は可換であることがわかる．具体例として，図 3(c) に，通常の 4-2 カウンタに符号変換を施すことによって実現された冗長 2 進全加算器の構成を示す．
□

以上のように，ある特定の符号パターンの入力データに対する加算器が与えられたときに，その入出力において符号変換を行うことにより，任意の符号パターンの SW 数入力データに対する加算器を構成することができる．これまでに述べたのは，基本的な 3-2 カウンタおよび 4-2 カウンタの例であるが，この原理自体は，大規模な加算木 (桁上げ伝播のない多入力加算器) に拡張することが可能である．すなわち，図 4に示すように，通常の加算木の入出力において符号変換を施すことにより，任意の符号ベクトルを有する SW 数に対する加算木を自由自在に構成することが可能である．

このような SW 数加算木は，特に大規模な多入力加算を必要とする定係数乗算器や積和演算器などにおいて有用である．通常これらの回路では，部分積数の圧縮のために Canonic Signed-Digit (CSD) 数による乗数のリコーディングを行うが，これにより乗数は不規則な正および負のディジットの系列に変換される．このため，部分積も不規則な正および負の符号パターンを有しており，加算木はこれらのデータを入力することが必要となる．SW 数加算木は，このような用途に対して有用であると考えられる．具体的には，2 の補数演算における符号拡張や補正ビットなどの不規則な処理を用いずに正・負のデータを統一的に扱うことが可能であり，これにより加算器の完全なモジュール化を行うことができる．このことは，FPGA などのプログラマブルなデバイスとして，高速な演算回路を実現する際に有用になると予想される．

3 SW 数系に基づく Field Programmable Digital Filter の構成

ディジタル信号処理においては，高速な積和演算回路が必須となる．特に，高速性を要求される通信系のディジタルフィルタなどにおいては，係数を固定し，完全にハードウェアに展開した積和演算回路が用いられる．しかし，このような構成では，フィルタ係数の変更に伴い，ハードウェアの再設計が必要となるため，用途が限られるという問題がある．これに対して，FPGA のように，プログラマブルにその係数や構造をマッピングでき，しかも ASIC に近い高速性を兼ね備えたデバイスが実現できれば，その応用範囲は広いものと考えられる．SW 数系によるデータ表現は，その規則性とモジュール性により，上記のようなディジタル信号処理用 FPGA の構成の際に有用である．そこで，本章では，SW 数系の加算アルゴリズムを採用した構成可能算術ブロック (Configurable Arithmetic Block: CAB) をバス構造のプログラマブル配線で結合したディジタルフィルタ用の FPGA — Field Programmable Digital Filter (FPDF) — の構成を提案する．

3.1 FPDF のアーキテクチャ

FPDF のアーキテクチャは，図 5からわかるように Configurable Arithmetic Block (CAB), Connection Block (CB) および Switch Block (SB) から構成される．従来の FPGA と比較した場合，構成可能な基本ブロックをプログラマブルな配線で接続して回路を構成するという基本的なアーキテクチャは変わらない．しかし，基本ブロックに桁上げ伝搬のない加算器を使用し，さらにそのブロックを 8 ビットを単位とするプログラマブル配線で結合したという点が従来の FPGA と大きく異なる．このことによっ

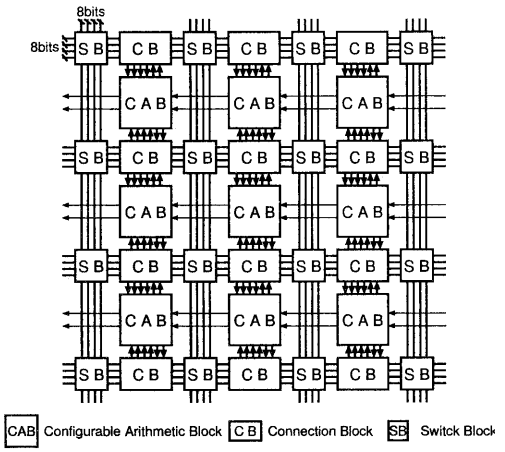

図 5: Field Programmable Digital Filter の構成

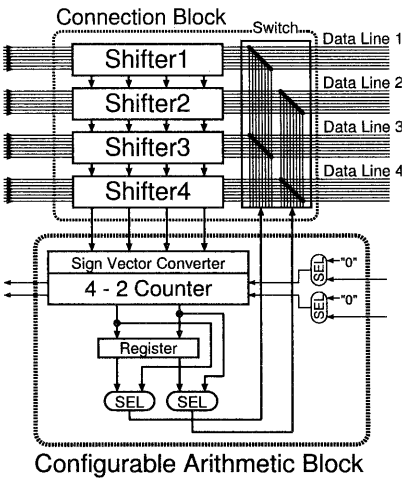

図 6: Configurable Arithmetic Block と Connection Block の構成

て，演算回路をより規則的な構造で実現でき，さらに，プログラマブル配線の構造・制御が単純になるため，高速かつコンパクトな回路が実現可能になる．

3.2 各ブロックの構成

FPDF の基本ブロックとなる Configurable Arithmetic Block (CAB) および Connection Block (CB) の構成を図 6 に示す．CAB は，基本的には桁上げ伝搬がない高速な SW 数 4-2 カウンタにより構成され，入力データの符号は符号変換器（Sign Vector Converter: SVC）で任意に設定可能となっている．CAB の 4 つのオペランドは，8 ビット幅のデータライン 1 – 4 から CB を介して入力される．CB は，基本的に (i)CAB へ入力されるデータラインのルーティング，(ii) シフトによる CSD 係数の乗算，(iii)CAB 出力からデータラインへのルーティングを実行する．特に，上記 (ii) を実現するために，CB 内部では図 7のような多数のスイッチマトリクスが存在し，それぞれ 0 – 7 ビットのシフトを実行する．

積和演算回路の入力にマッピングされた CAB は，符号ベクトル変換器で入力の持つ符号ベクトルを CAB 内部の 4-2 カウンタで扱えるような符号ベクトルに変換する．符号ベクトル変換の操作は入力の反転 (not) なので，図 8のように排他的論理和 (Exclusive OR) をフィルタの構成情報を持った SRAM で制御することで実現する．

一方，Switch Box (SB) は図 5においてデータラインが交差する点に置かれたスイッチマトリクスであり，図 9 のように構成される．図 9(c) のように，4 方向から来るデータラインを 6 つの NMOS Switch で短絡・開放させることによって，配線構造をプログラムする．

以下で，シフタと符号ベクトル変換器を用いた CSD 係数乗算器の構成例を示す．

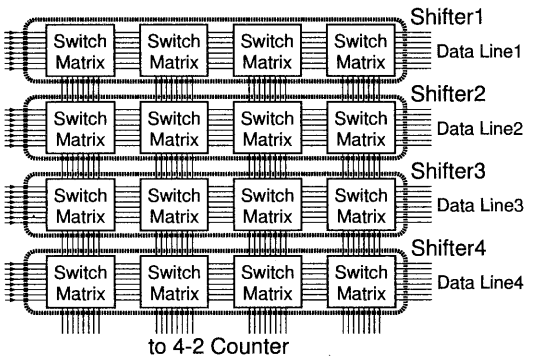

図 7: シフタの構造

例 4 シフタと符号ベクトル変換器を用いた 8 ビット幅の CSD 係数乗算器の構成例を示す．係数が $0.1011011_2(\approx 0.71)$ とする．

1. 係数 0.1011011_2 を CSD 数に変換する．係数に含まれる non-zero ビットの数を最小になるよう変換すると，$1.0(-1)00(-1)0(-1)_{CSD}$ となる．
2. CSD 係数からシフタ 1 – 4 でシフトする量が決まる．シフタ 1 – 4 でシフトする量はそれぞれ

$$\begin{array}{ll} \text{入力 1} & : \ 0 \text{ ビットシフト (シフタ 1)} \cdots (2^0 \text{倍}) \\ \text{入力 2} & : \ 2 \text{ ビットシフト (シフタ 2)} \cdots (2^{-2} \text{倍}) \\ \text{入力 3} & : \ 5 \text{ ビットシフト (シフタ 3)} \cdots (2^{-5} \text{倍}) \\ \text{入力 4} & : \ 7 \text{ ビットシフト (シフタ 4)} \cdots (2^{-7} \text{倍}) \end{array} \tag{23}$$

である．

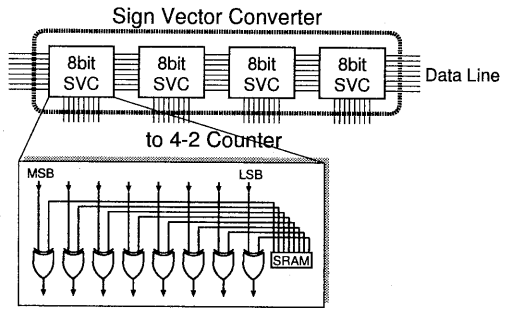

図 8: 符号ベクトル変換器の構造

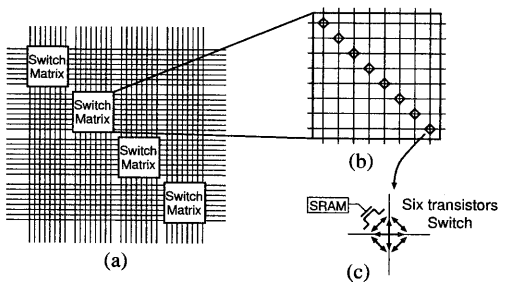

図 9: Switch Block の構成

3. 4 つの部分積の符号ベクトル$\boldsymbol{\Lambda}'_1, \boldsymbol{\Lambda}'_2, \boldsymbol{\Lambda}'_3, \boldsymbol{\Lambda}'_4$を求め，それらを$\boldsymbol{\Lambda}_{bin} = (+1, \cdots, +1)$ に変換する．$\boldsymbol{\Lambda}'_1, \boldsymbol{\Lambda}'_2, \boldsymbol{\Lambda}'_3, \boldsymbol{\Lambda}'_4$はそれぞれ

$$\begin{aligned}
\boldsymbol{\Lambda}'_1 &= (-,+,+,+,+,+,+,+,+,+,+,+,+,+,+,+,+) \\
\boldsymbol{\Lambda}'_2 &= (+,+,+,-,-,-,-,-,-,-,+,+,+,+,+,+,+) \\
\boldsymbol{\Lambda}'_3 &= (+,+,+,+,+,+,+,-,-,-,-,-,-,-,+,+,+) \\
\boldsymbol{\Lambda}'_4 &= (+,+,+,+,+,+,+,+,+,-,-,-,-,-,-,-,+)
\end{aligned} \tag{24}$$

である．式中の 1 は省略した．$\boldsymbol{\Lambda}'_1, \boldsymbol{\Lambda}'_2, \boldsymbol{\Lambda}'_3, \boldsymbol{\Lambda}'_4$に含まれる$-1$ に対応した桁で符号変換を行う．

図 10にこの乗算回路の構成を示す．符号ベクトル変換器はインバータで略記した．□

3.3 テスト回路の試作

FPDF の動作原理を確認するために，8 ビット 4 次程度の小規模な FIR フィルタをプログラムできるテストチップを 0.5μmCMOS 1 層ポリシリコン 2 層配線技術で試作した．このテストチップは，ブロック間配線が固定されているためフィルタの構造を変えることはできないが，シフタ・符号ベクトル変換器によってフィルタの係数をプログラム可能である．テストチップの回路構成を図 11に，仕様を表 1に，チップ写真を図 12に示す．

表 1: テストチップの仕様

入出力語長	8 ビット入出力
フィルタ次数	4 次
動作周波数	約 100MHz (I/O バッファを除いた HSPICE 上でのシミュレーション)
トランジスタ数	10027
実効面積	$1.0mm \times 1.0mm$
プロセス	0.5μm CMOS 1 層ポリシリコン 2 層メタル配線

4 むすび

本稿では SW 数系の加算アルゴリズムを採用した構成可能算術ブロック (CAB) で FPGA を構成した FIR フィルタ用 FPGA — FPDF の構成を示した．FPDF は実時間処理が求められるマルチメディア通信を中心とした高速なディジタル信号処理において有用になると考える．

本稿で提案した FPDF は加算器を基本ブロックにした積和演算専用の FPGA であるが，除算や FFT などさらに多くの算術演算に対応した算術演算用 FPGA が開発されればその用途はさらに大きく広がるものと考えられる．

謝辞

本チップ試作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，NTT エレクトロニクス株式会社および大日本印刷株式会社の協力で行われたものである．

参考文献

[1] T. Arslan, H. I. Eskikurt, and H. D. Horrocks,"Configurable structures for a primitive operator digital filter FPGA," *IEEE Workshop on Signal Processing Systems(SiPS): Design and Implementation*, pp. 532–540, November 1997.

[2] C. Chen and J. M. Rabaey,"A reconfigurable multiprocessor ic for rapid prototyping of algorithmic-specific high-speed DSP data paths," *IEEE J. Solid-State Circuits*, Vol. 27, No. 12, pp. 1895–1904, December 1992.

[3] I. Koren,*Computer Arithmetic Algorithms*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

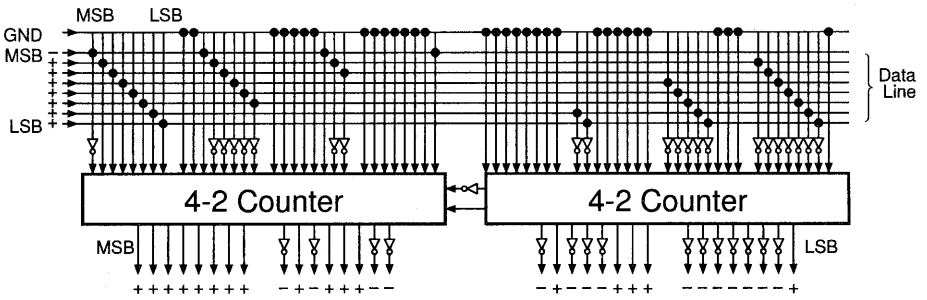

図 10: CSD 係数乗算器の構成例

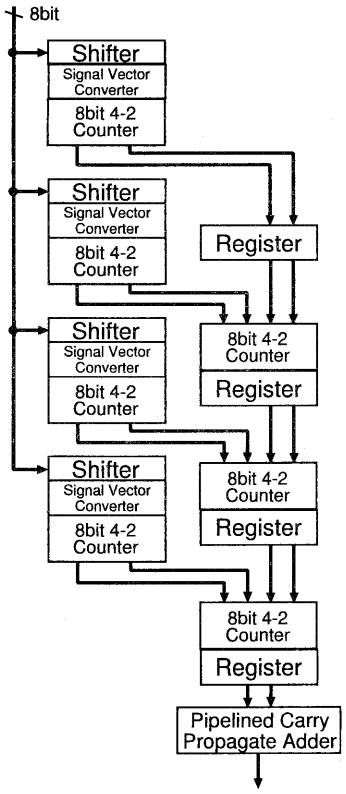

図 11: テスト回路の構成

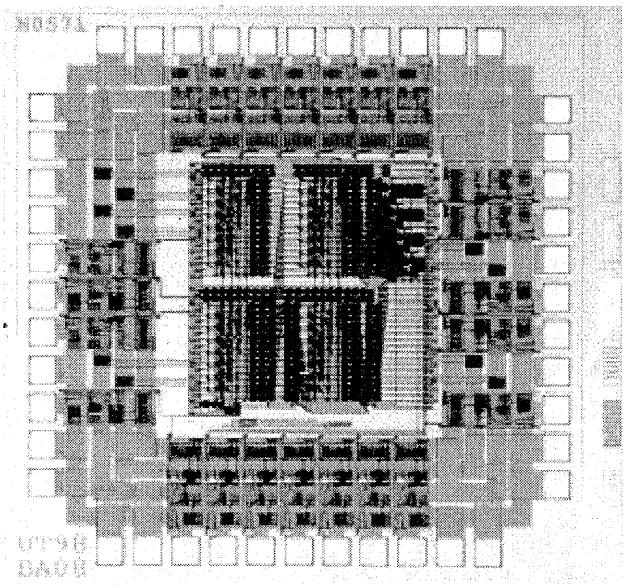

図 12: テストチップ (係数可変 FIR フィルタ)