

科学研究費助成事業 研究成果報告書

平成 27 年 4 月 28 日現在

機関番号：32689

研究種目：挑戦的萌芽研究

研究期間：2013～2014

課題番号：25540021

研究課題名(和文) セレクタ論理を利用し部分積項数を半減する差積演算回路設計とその画像処理応用

研究課題名(英文) Efficient Subtract-Multiply Operation Circuit Design Using Selector Logics

研究代表者

戸川 望 (TOGAWA, NOZOMU)

早稲田大学・理工学術院・教授

研究者番号：30298161

交付決定額(研究期間全体)：(直接経費) 2,900,000円

研究成果の概要(和文)：差積演算 $(a-b)*c$ は、高速フーリエ変換や超解像処理など重要な画像処理の基本演算であるが、減算と乗算の順序関係のため演算時間が増大するという問題点がある。差積演算を2進表現し展開し、途中項を2つずつペアにすると「セレクタ論理」に帰着されることを示す。そして、画像の拡大・縮小など画素補間演算に使用されるバイリニア補間を取り上げ、バイリニア補間演算をセレクタ論理に帰着させることで、桁上げ伝搬遅延を削減し高速化したセレクタ論理帰着型バイリニア補間演算器を提案する。バイリニア補間演算器について、セレクタ論理帰着型バイリニア補間演算器をはじめ、種々の方法で実装し評価・比較した結果を示す。

研究成果の概要(英文)：A subtract-multiply operation, $(a-b)*c$, is a basic and important operation in image processing but it requires much computation time since we first calculate $t=(a-b)$ and, after that, we calculate $t*c$. In this report, we first show that the subtract-multiply operation can be effectively implemented by “selectors.” After that, we pick up bi-linear interpolation and apply the selector-logics to it. Selector logics can reduce the carry-propagations and then we can realize area-efficient and fast dedicated circuits. We have implemented our proposed bi-linear interpolation circuit in several ways and evaluated each of them.

研究分野：集積システム設計

キーワード：セレクタ演算 差積演算 画像処理

1. 研究開始当初の背景

差積演算とは2変数 a, b の差に別の1変数 c を乗算する演算で $(a - b) \times c$ と表される専用演算の1つである。高速フーリエ変換 (FFT) をはじめ実用上最も重要な演算回路の基本演算として頻出する演算である。差積演算は「2変数の差をとった後に、さらに別変数と乗算する」という順序関係が存在するため、2変数減算と2変数乗算それぞれの桁上げ伝播遅延を直列に合計したものが全体の演算時間となる。結果、差積演算は基本演算であるが、演算時間が極めて大きいあるいは電力消費が極めて大きくなる。長年これは不可避の問題として棚上げされているのが実情であった。

一方、専用演算の代表例かつ歴史的に最もその構成が最適化されたものが積和演算である。積和演算は $(a + b) \times c$ によって表される。信号処理で頻繁に利用される演算であり、これを極めて効率よく計算するのがMAC演算回路 (Multiply-ACcumulate 演算回路) である。MAC演算回路は構成や方式について多数の提案がなされたが、我々の知る限り国内外を問わず差積演算に特化した演算回路の最適設計の報告はない。

また近年、アプリケーションの規模の増大に伴い、システムLSI (集積回路) はより効率的で高速な演算が期待されている。それに伴い、特定の演算に特化した高速かつ効率的な演算器の設計が求められている。補間演算はこうした効率化・高速化が要求される演算の一つである。補間演算とは、得られたデータ列から範囲内の値を推定する演算であり、画像の拡大・縮小や、歪みの補正などに用いられる。バイリニア補間は周囲4つの点から線形的に値を補間する演算であり、2つの値を直線的に結ぶことで補間を実行する線形補間の組み合わせとして表すことができる。したがって、線形補間演算を効率化することでバイリニア補間の効率化が実現される。バイリニア補間演算の中心的な役割を果たすのが前述した差積演算であり、その効率的な実現がバイリニア補間演算そのものの効率化を実現すると考えられる。

これに対しそれぞれ n ビット3変数 a, b, c を2進数で表記し、差積演算を展開すると約 $2n^2$ 個の途中演算項 (これを部分積項と呼ぶ) が出現するが、部分積項を2つずつペアにすると、部分積項の大部分を $x_i z_i + y_i z_i$ という形式で表現することに成功した。 x_i, y_i, z_i は入力変数中の適当な1ビットを表す。 $x_i z_i + y_i z_i$ を評価すると、 z_i の値が1のとき出力は x_i 、 z_i の値が0のとき出力が y_i になるというセレクトア論理を表す。これは桁上げを必要とせず、極めて高速かつ低電力にその演算を実行できる。即ちセレクトア論理ペアを事前演算 (pre-computation) し、2つの部分積項から極めて高速に1つの演算結果を生成すれば、差積演算に必要な部分積項数は半減する。原理的に差積演算の演算時間や消費電力を最

大で半減できることを意味している。

2. 研究の目的

上記のアイデアのもと、本研究ではセレクトア論理を用いて差積演算回路を実現することで遅延・電力を削減することを考える。応用例として画像処理に含まれる各種の演算、とりわけバイリニア補間演算を取り上げ、ここに含まれる差積演算に着目し遅延・電力の削減を図る。

3. 研究の方法

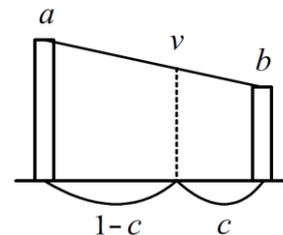
(1) 線形補間

補間とは既知のデータの範囲内で値を推定することである。線形補間は2つの端点を直線で結んで補間する演算であり、補間演算の中でも実用的に用いられることが多い。

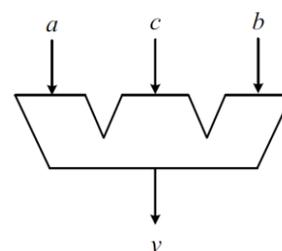
2つの端点を示す入力信号を a, b とし、その間の値を $(1 - c) : c$ の割合で線形補間したときの出力を v とする。ただし a, b は整数で、 c は $0 < c < 1$ の小数とする。このとき線形補間は下式で表現できる。

$$v = ac + b(1 - c)$$

線形補間の概要を図1(a)に、ブロック図を図1(b)に示す。



(a) 線形補間の概要



(b) 線形補間器のブロック図

図1: 線形補間

(2) セレクトア論理と線形補間演算器設計

さて、ここで線形補間演算をビットレベル式変形しよう。入力信号 a, b は n ビットの符号付き整数であり、これをビットレベルで表現すると下式となる。

$$\begin{aligned} a &= [a_{n-1}a_{n-2} \cdots a_1a_0] \\ &= -a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_02^0 \end{aligned}$$

$$= -a_{n-1}2^{n-1} + \sum_{j=0}^{n-2} a_j 2^j$$

$$\begin{aligned} b &= [b_{n-1}b_{n-2} \cdots b_1b_0] \\ &= -b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots + b_02^0 \\ &= -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j \end{aligned}$$

また、補間の割合を示す c は $0 < c < 1$ の n ビットの小数によって表すと、そのビットレベル式表現は下式となる。

$$\begin{aligned} c &= [0.c_{n-1}c_{n-2} \cdots c_1c_0] \\ &= c_{n-1}2^{-1} + c_{n-2}2^{-2} + \cdots + c_02^{-n} \\ &= \sum_{i=0}^{n-1} c_i 2^{-(n-i)} \end{aligned}$$

これらの式から線形補間を表す式をビットレベルで記述すると以下ようになる。

$$\begin{aligned} v &= \left(-a_{n-1}2^{n-1} + \sum_{j=0}^{n-2} a_j 2^j \right) \left(\sum_{i=0}^{n-1} c_i 2^{-(n-i)} \right) \\ &+ \left(-b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j \right) \left(1 - \sum_{i=0}^{n-1} c_i 2^{-(n-i)} \right) \end{aligned}$$

この式において、 a_i, b_i, c_i は 1 ビットの変数であり、0 か 1 を表す。そこでこれらの変数に対して、論理演算を定義することができ、これを用いると、上記の線形補間を表す式は以下のように変形することができる。

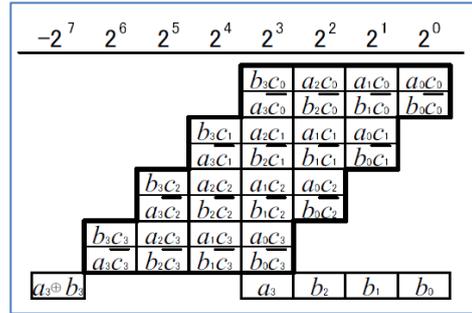
$$\begin{aligned} v &= -(a_{n-1} \oplus b_{n-1}) + a_{n-1}2^{-1} \\ &+ \sum_{i=0}^{n-1} (b_{n-1}c_i + a_{n-1}\bar{c}_i) 2^{i-1} \\ &+ \sum_{i=0}^{n-1} \sum_{j=0}^{n-2} (a_i c_j + b_j \bar{c}_i) 2^{-(n-i-j)} \\ &+ \sum_{j=0}^{n-2} b_j 2^{-(n-j)} \end{aligned}$$

上記の式は、内部に $a_i c_i + b_i \bar{c}_i$ といった項を含む。これはセレクタ論理そのものを表し、この部分を予めセレクタ論理によって計算してもその値は 1 ビットで表現でき、算術結果と等価な値を得ることができる。

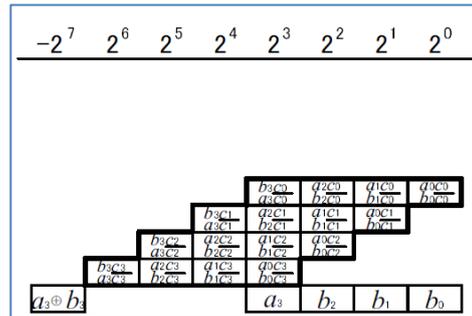
上記の式をそのまま評価した場合、その部分積項の数は $2n^2 + n + 1$ 個となるが、セレクタ論理によって 2 つの部分積項を予め計算

しておくと、その数は $n^2 + n + 1$ 個に削減される。この数は n が十分に大きければ、部分積項の数約半数に削減されることを表している。

例えば、 $n = 4$ としたとき、線形補間演算の部分積項数を図 2 に表す。実線で囲んだ部分がセレクタ論理に帰着可能な部分を表している。この例の場合、セレクタ論理を適用する前に 37 個必要な部分積項が、セレクタ論理を適用することによって 21 個に削減できる。



(a) セレクタ論理を適用しない場合



(b) セレクタ論理を適用した場合

図 2: $n = 4$ の場合の線形補間演算に必要な部分積項数

(3) バイリニア補間

バイリニア補間は、画素の補間位置を包括する正方形領域を構成する 4 画素の画素値から線形的に値を補間することで、補間位置における画素値を算出する手法である。縦横に対してそれぞれ線形補間が行われるためバイリニア法と呼ばれる。図 3 にバイリニア法による画素値の補間を示す。

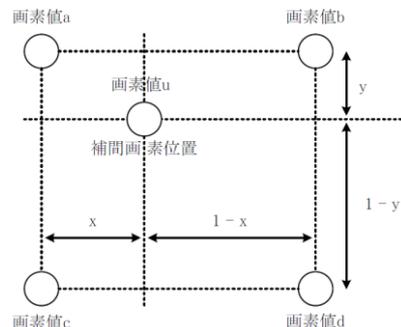


図 3: バイリニア補間

図3に示したバイリニア補間を考える。ここで補間位置の周囲の画素の画素値を図3のように a, b, c, d とし、横方向の補間の割合を x 、縦方向の補間の割合 y とする。ただし a, b, c, d は整数、 x, y は $0 < x, y < 1$ を満たす小数とする。算出する補間位置の画素値を u とすると、 u は下式で表せる。

$$u = \{ax + b(1-x)\} + \{cx + d(1-x)\}(1-y)$$

上式は下線部に着目するとそれぞれ入力 $\{a, b, x\}$, $\{c, d, x\}$ の線形補間の形式となっており、また式全体としても $\{ax + b(1-x), cx + d(1-x), y\}$ の3項を入力とした線形補間の形式となっている。したがって、このバイリニア補間のブロック図は図4のようなブロック図によって表すことができる。

線形補間はセレクトラ論理を適用可能であるため、図4からセレクトラ論理帰着型線形補間演算器を3つ用いることでセレクトラ論理帰着型のバイリニア補間演算器を設計することが可能であることが分かる。

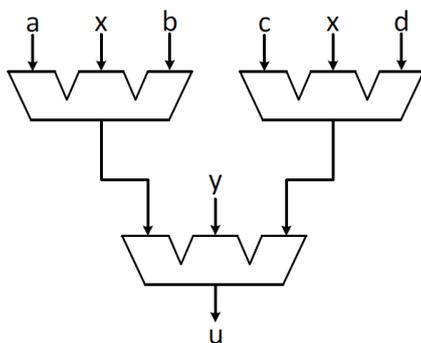


図4: バイリニア補間演算器のブロック図

4. 研究成果

提案したセレクトラ論理帰着型バイリニア補間演算器をいくつかの方法で実装し、論理合成ツールによって評価・比較する。なお、セレクトラ論理帰着型バイリニア補間演算器の設計にはハードウェア記述言語 VHDL を用いた。

論理合成ツールによる評価では、Synopsys社のDesign Compiler D-2010.03-SP5のトポグラフィカルモードを用いた。セルライブラリにはSTARC (CMOS 90nm) の設計ルールを用いた。4ビット、8ビット、16ビット入力のバイリニア補間演算器をそれぞれ以下の4通りの手法によって設計した：

- セレクトラ論理を用いずすべての部分積を生成し、HDLの算術演算子によって加算する[Op].
- セレクトラ論理を用いずすべての部分積を生成し、Wallace treeによって加算する[Wa+CLA].
- セレクトラ論理で部分積を生成し、HDLの算術演算子によって加算する[Sel+Op].

- セレクトラ論理で部分積を生成し、Wallace treeによって加算する[Sel+Wa+CLA].

図5～図7に評価実験結果として、各設計について面積・遅延値の関係を示す。

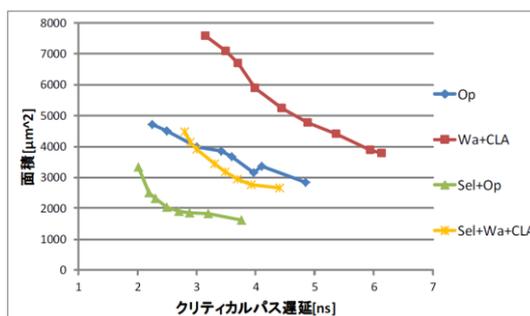


図5:4 ビットのバイリニア補間演算器設計

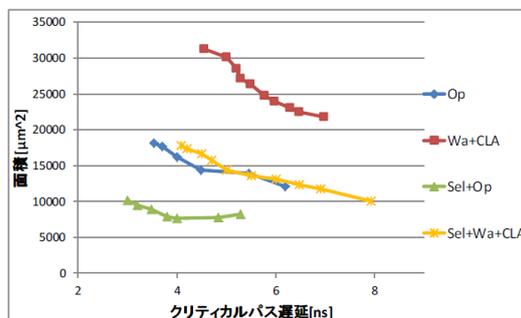


図6:8 ビットのバイリニア補間演算器設計

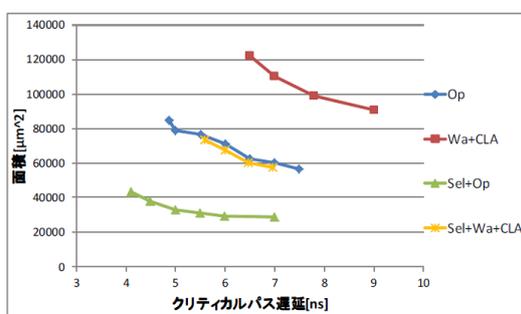


図7:16 ビットのバイリニア補間演算器設計

[Op]と[Sel+Op], [Wa+CLA]と[Sel+Wa+CLA]をそれぞれ比較すると、ともにセレクトラ論理を適用した結果が遅延、面積の側面で削減されていることが分かる。[Op]に対して[Sel+Op]は遅延が平均約17%、面積が平均約42%削減されており、[Wa+CLA]に対して[Sel+Wa+CLA]は遅延が平均約14%、面積が平均約41%削減されている。削減率が最も大きかったのはすべてのビット幅において[Sel+Op]で遅延が最大18%、面積が最大48%削減された。また、同様に[Op]と[Sel+Op], [Wa+CLA]と[Sel+Wa+CLA]をそれぞれ比較すると、セレクトラ論理を適用した結果[Sel+Op], [Sel+Wa+CLA]の方が効率的であることが分かる。

以上から、バイリニア補間にセレクトラ論理を適用することで遅延ならびに面積が削減され効率化が実現できた。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 8 件)

- [1] 塩雅史, 柳澤政生, 戸川望, セレクトラ論理を利用した線形補間演算器設計と評価, 電子情報通信学会 VLSI 設計技術研究会, 2013 年 5 月 16 日, 北九州国際会議場 (福岡県・北九州市).
- [2] 塩 雅史, 柳澤政生, 戸川望, セレクトラ論理を利用した線形補間演算器の実装と評価, 電子情報通信学会第 26 回回路とシステムワークショップ, 2013 年 7 月 29 日, 淡路夢舞台国際会議場 (兵庫県・淡路市).
- [3] 塩雅史, 柳澤政生, 戸川望, セレクトラ論理を利用したバイリニア補間演算器設計と評価, 電子情報通信学会 VLSI 設計技術研究会, 2013 年 10 月 8 日, 弘前大学 (青森県・弘前市).
- [4] Masashi Shio, Masao Yanagisawa, and Nozomu Togawa, Linear and bi-linear interpolation circuits using selector logics and their evaluations, 2014 IEEE International Symposium on Circuits and Systems, 2014 年 6 月 3 日, Melbourne (Australia).
- [5] 五十嵐啓太, 柳澤政生, 戸川望, セレクトラ論理に帰着させたバイリニア補間演算器を用いた画像拡大縮小回路の FPGA 実装, 情報処理学会 DA シンポジウム 2014, 2014 年 8 月 29 日, ホテル下呂温泉水明館 (岐阜県・下呂市).
- [6] 塩雅史, 柳澤政生, 戸川望, 4-to-1 セレクトラ論理及び 2-to-1 セレクトラ論理を利用したバイリニア補間演算器の設計と評価, 情報処理学会 DA シンポジウム 2014, 2014 年 8 月 29 日, ホテル下呂温泉水明館 (岐阜県・下呂市).
- [7] 五十嵐啓太, 柳澤政生, 戸川望, FPGA を対象としたセレクトラ論理型アルファブレンディング回路の実装と評価, 電子情報通信学会総合大会, 2015 年 3 月 13 日, 立命館大学 (滋賀県・草津市).
- [8] Keita Igarashi, Masao Yanagisawa, and Nozomu Togawa, FPGA implementation and evaluation of image scaling circuits using selector-logic-based bi-linear interpolation, 19th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI 2015), 2015 年 3 月 17 日, Yilan (Taiwan).

[その他]

ホームページ等

<http://www.togawa.cs.waseda.ac.jp/research.html>

6. 研究組織

(1) 研究代表者

戸川 望 (TOGAWA NOZOMU)

早稲田大学・理工学術院・教授

研究者番号 : 30298161