

任意の関数の有限語長近似における演算回路最適化

小谷 光史[†] 佐々木孝雄[†] 豊嶋 久道[†]

[†] 神奈川大学工学部

あらまし ハードウェア記述言語を用いたデジタル回路設計において、初等関数やユーザ定義の関数など直接表現できないものがある。この場合、対象関数は多項式近似され、加算器と乗算器によって実装される。また許容範囲を超えた近似誤差は、テーブルによって補正される。しかしながら、演算誤差と回路規模の間の離散的、非線形的関係を考慮すると、全体のハードウェアコストの最適化は困難になる。本研究では、任意の関数に対して並列タブーサーチを用いることでハードウェアコストを最小化する方法を提案する。提案法は、関数の特徴を考慮した二つのハードウェアモデルを基に、複数の初期値から広域探索を行う。

Arithmetic Circuit Optimization in Finite Word Length Approximation of Arbitrary Functions

Koji KOTANI[†], Takao SASAKI[†], and Hisamichi TOYOSHIMA[†]

[†] Faculty of Engineering, Kanagawa University

Abstract In digital circuit design using a hardware description language, some elementary functions and user defined functions can not be expressed directly. In such cases, the target function is approximated as a polynomial to be implemented by additions and multiplications. If approximation error exceeds the permissible one, it can be corrected with a look-up table. However, considering discrete and nonlinear relation between the approximation error and the circuit area, optimization of total hardware cost would be more complex. In this research, for an arbitrary function, we propose the technique of minimizing the hardware cost using parallel tabu search. The proposed method starts with multiple initial solutions based on two hardware models, and make a global search.

1. はじめに

近年の回路設計において、動作仕様はC言語のようなソフトウェア的なアルゴリズムとして与えられる。しかし、実装する場合、ソフトウェア記述をVHDL/Verilog HDLなどのハードウェア記述言語[1][2]によりハードウェア記述に変換する必要がある。その一例として初等関数が挙げられる。例えば、ソフトウェア記述では、正弦関数を単に \sin と記述するだけでよいのに対して、ハードウェア記述では、加算・乗算・ビットシフトによる多項式やテーブルを用いて表現しなければならない[3]。初等関数のように一般的によく用いられる関数に対しては、それぞれの関数に特化したハードウェア記述方法が知られているが、ユーザーが独自に作成する任意の関数に対しては、ハードウェア記述に変換するための汎用的な手

法が必要となる。

一般に任意の関数をハードウェア実現するためには、多項式近似によって表現する手法[4]や、テーブルを用いて表現する手法[5]がある。しかし、多項式近似の表現手法では、チェビシェフ近似により誤差を許容範囲内に収めるためには係数が無限語長で表現されなければならない。そのため、演算回路が増大する問題点がある。一方、テーブルを用いる手法でも、入力値に対して出力の結果を全て与えておかなければならないため、精度を上げるためにはテーブルの規模が増大する。手法[6]では、これらの問題点を踏まえ、任意の関数を有限語長による多項式近似により表現し、演算誤差をテーブルによって補正する。その結果、回路規模を削減することが可能となる。

しかし、この手法においては、演算誤差（テーブルサイズ）と演算回路規模がトレードオフの関係となるため、

係数の値を変えることで最適となる回路を探索することが必要となる。係数の変化に対する演算誤差や回路規模の関係は離散的、非線形的影響により複雑であり、膨大な係数の組合せに対して最適値を探索することは容易ではない。そこで本研究では、任意の関数に対して回路の最適化を行う手法を提案する。最適化手法としては、このような数値最適化問題に対して有効である局所探索アルゴリズムのタブーサーチを用いるが、本研究では初期解を複数持つことで並列化し、広域探索も可能とする。そして、関数の特徴を考慮した2つの回路表現を基に最適解を探索する。さらに、初期解の設定、探索領域の制限により解空間を制限することで探索の効率を向上させる。

2. 任意の関数に対する演算回路の表現

任意の関数 $F(x)$ をハードウェアとして設計するためには、加算・乗算・ビットシフトによって表現しなければならない。ここで一般的な手法として多項式による近似 [6] がある。しかし、実際に回路表現するときには $F(x)$ の係数を有限語長で近似する必要がある。その誤差は関数の形状により大きく変わってくる。そこで、有限語長近似の手法として、ここでは二つの表現手法 (多項式表現、因数分解表現) を用いる。

2.1 多項式表現

係数量子化の影響の小さい関数に対しては、通常多項式表現を用いる。多項式表現では、任意の関数 $F(x)$ を式 (1) のように x に関する N 次の多項式として近似して $F'(x)$ とする。ここで、 C_i は係数の値を示す。

$$F'(x) = \sum_{i=0}^N C_i x^i \quad (1)$$

この手法ではチェビシェフ近似により誤差を許容範囲内に収めることができるが⁵、係数の値を有限語長で近似する必要があるため量子化誤差の影響を受ける。そこで、テーブルにより誤差を補正する。多項式表現による演算回路表現を図 1 に示す。

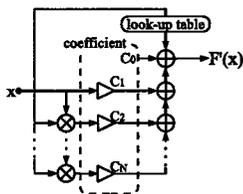


図 1 多項式表現

この表現手法では、図 1 のように入力値と係数を乗算し、最後にテーブルによる誤差補正を含めて加算する。

2.2 因数分解表現

振動を含む関数など多項式表現による量子化誤差の影響が大きい関数に対しては因数分解表現を用いる。ここでは直線成分を

$$\beta(x) = ax + b \quad (2)$$

とおき、任意の関数 $F(x)$ を次式のように $F''(x)$ として表現する。

$$F''(x) = \alpha \prod_{i=0}^N (x + C_i) + \beta(x) \quad (3)$$

ここで、 C_i は係数、 α はスケーリングを表す。この表現では、 $\beta(x)$ を除いた成分が x 軸を交差して振動する。 $-C_i$ が各交点と一致することから近似誤差の発散を防ぐことができる。演算回路は図 2 のように表される。多項式表現と同様に係数量子化による誤差はテーブルで補正する。

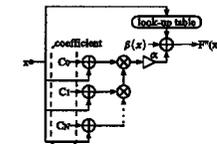


図 2 因数分解表現

3. 最適化手法

前述のような演算回路の表現では、演算誤差と演算回路規模がトレードオフの関係になるため、係数の値を変えることで最適となる回路を探索する必要がある。係数の変化に対する演算誤差や回路規模の関係は離散的、非線形的影響により複雑であり、決定的手法により回路規模を最適化する係数を探索することは困難である。そこで、本研究ではこの問題を組合せ最適化問題として扱う。

一般的には組合せ最適化手法として遺伝的アルゴリズム、シミュレーテッドアニーリング、タブーサーチなどが知られている。今回の問題は、チェビシェフ近似による係数を初期解の候補とすることから本研究では、局所探索に有効であるタブーサーチを用いる [7] [8]。但し、タブーサーチは、初期解に依存する傾向がある。そこで、一つの解のみで単点探索をするのではなく、いくつかの初期解を並列的に用いた多点探索により、広域的な探索も含めた最適化をする。

3.1 解表現

本研究の最適化における解は、次数分の係数を並べることで表現する。解表現は演算回路の表現手法ごとに異なる。多項式表現では、係数 C_i を全て並べて一つの解とする。因数分解表現では、係数 C_i とスケーリング係数 α を並べて解とする。但し、直線性分 $\beta(x)$ は、解に含める

と、解空間が広がるだけでなく、関数値が大きく変わってしまうことから最適化が困難となる。そこで、 $\beta(x)$ は解には含まず、関数に合わせて設定することとする。

3.2 タブーサーチ

タブーサーチの探索手法は、現在の解の局所的近傍を探索し、その近傍内の最良解を選択して、新しい解とする。そして、新しい解が現在の解よりも悪くなったとしても、近傍における最良解を現在の解に換え探索を続ける。ここで、処理のループを避けるためにタブーリストと呼ばれる探索の制約条件を用いる。これによって同一解を探索し続けることを防いでいる。本研究では初期解として、以下のように複数の解を用いてタブーサーチを並列化させる。

まず、それぞれの表現手法に応じた初期解を設定する。多項式表現では、チェビシェフ近似の結果を丸めた値、因数分解表現では、因数分解をした結果を丸めた値を求めて初期解とする。これに加えて以下の4種類の解をそれぞれの表現における初期値として設定する。

- (1) 全ての係数の値が0(テーブルのみでの生成)
- (2) 全ての係数の値が最大値
- (3) 全ての係数の値が最小値
- (4) ランダムによる値

(1)の初期解は係数を用いていない、つまり演算回路を用いずにテーブルのみにより回路を生成することを意味する。このことにより、最適化結果は、最悪でもテーブルのみで設計した場合と同じになるすることができる。次に、(1), (2), (3)の初期解により、探索範囲の中間値、最大値、最小値を評価する。このような初期解を用いることで、ランダムな初期解設定では得ることが困難な解空間の端の値を網羅することができ、探索の有用性を高める効果がある。最後に(4)では、規則性を持たないことで全く違った点に最良解があるときに効果がある。

さらに、従来の並列タブーサーチの手法では、並列にしている解に関連性が薄いためタブーリストを別にして探索を行うが、本研究では、このタブーリストを共有する。タブーリストの共有は2つの表現手法それぞれで行い、関連性を重視させながら探索をする。タブーリストの共有を図3に示す。

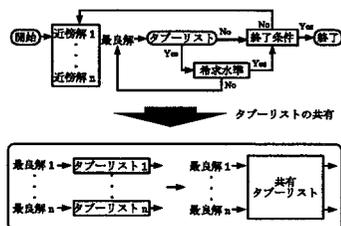


図3 タブーリストの共有化

この手法を用いるとタブーとなっている点が値同士で関連性を持ったまま大きく変化することが期待でき、一般的なタブーサーチでは近傍解から得ることが困難な解も探索することができ、大域的な探索と局所探索の両面からの探索が可能となる。

3.3 近傍解の生成手法

タブーサーチを用いた探索で重要となるのが近傍解の生成手法である。本研究では、近傍解は全ての係数に対して生成するのではなく、タブーサーチによる短期メモリと長期メモリを用いることによって確率的に選択された係数でのみ生成する[9]。短期メモリは過去にそれぞれの係数が連続して近傍を生成されなかった数を示し、長期メモリは近傍を生成された累積の回数を示す。これを用いて、選択された度合の低い係数が選ばれやすくなっている。そして、選択された係数から最下位ビットの値を変化させて近傍解を求める。この時、選ばれた係数によって作れる全てのパターンで近傍解を生成する。例として、係数 C_0, C_1, C_{N-1} を選択した時の近傍解の生成を図4に示す。

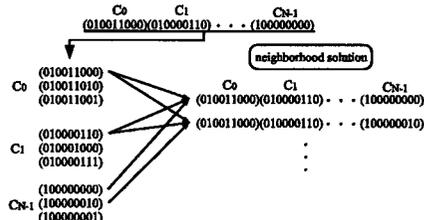


図4 近傍解生成

3.4 評価値

本研究では、演算回路のコストを、テーブルコストと演算コストの2つの視点から評価する。まず、テーブルは関数の演算誤差を補正するためのなので、誤差の最大ビット幅がテーブルの入力ビット幅となる。テーブルコストは図5のように入力ビット幅に応じて変化する。

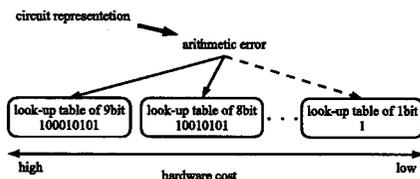


図5 テーブルのハードウェアコスト

次に、演算コストは加算器の数で評価する。加算器数は、係数ごとの非零ビットの数を数えることで求める。ここでは、 $\{0,1\}$ からなる2進数表現の代わりに、 $\{-1,0,1\}$ を用いて表現する方法で非零ビットが連続して現れないようにしたCSD(Canonical Signed Digit)を用いる。

このような2つのコストを考慮して、本研究での演算回路の評価手法を示す。始めに、テーブルコストの評価のために、対象関数と求めた関数の誤差を評価する。そこで、誤差の最大値の逆数を評価値とする。これにより誤差の評価ができ、必要なテーブルビット幅が計算できる。評価値 TEV を式(4)として求める。ここで $error$ は任意の関数と実際に求める関数の誤差を示す。

$$TEV = \frac{1}{\max\{error\}} \quad (4)$$

TEV の計算によってテーブルコストの評価をすることができる。しかし、演算回路全体のコストを考慮した場合、演算器でのデータの削減を考慮することも必要である。そこで、演算回路で必要とする加算器数の数を $adder$ としたときの評価値 EV を式(5)に示す。

$$EV = 2^{\lceil \log_2 TEV \rceil} + \frac{2^{\lceil \log_2 TEV + 1 \rceil} - 2^{\lceil \log_2 TEV \rceil}}{adder} \quad (5)$$

評価値 EV は、 $\lceil \log_2 TEV \rceil$ によって求められるテーブルのビット幅が同じ場合は、加算器数によって値を評価する。つまり、テーブルのコストを重視して演算器のコストを評価する。

4. シミュレーション

本手法をいくつかの関数に適用して設計した結果を示す。適用した関数は1入力1出力のものとする。パラメータは以下のように設定する。

TS: タブーリストの数: 7, 近傍解の数: 27, 次数 N : 8
 タブーサーチの実行回数: 20000
 並列するタブーサーチの数: 12
 x の範囲: $0 \leq x \leq 8$, 入出力ビット: 9
 対象関数: $F_1(x) = \cos(x)\log(x+1)$
 $F_2(x) = (x+1)\log(x+1)$

並列するタブーサーチの数は初期解の数でもあり、前述で述べたように設定する。さらに、タブーリストの共有をするのでそれぞれに複数の同一初期解を用いることで最適化をする。

4.1 任意の関数のハードウェアコスト比較

$F_1(x)$ の特性を図6に、 $F_2(x)$ の特性を図7に示す。

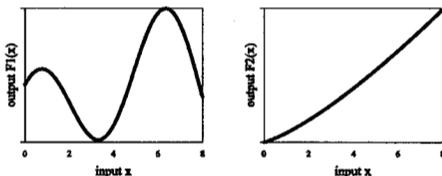


図6 $\cos(x)\log(x+1)$ 図7 $(x+1)\log(x+1)$
 係数のビット幅を9ビットとし、本手法とチェビシェフ近似の値を丸めた値に対して、加算器数とテーブルのビット幅との比較を表1に示す。

表1 ハードウェアコストの比較

function	$\cos(x)\log(x+1)$		$(x+1)\log(x+1)$	
	proposed	Chebyshev	proposed	Chebyshev
adders	15	19	11	11
table	1	8	5	8

図6は、振動を含むことから、因数分解を用いた結果で最適化をしている。一方、図7は、係数量子化の影響が小さいことから、多項式表現を用いた結果で最適化をしている。

関数 $F_1(x)$ に対しては、加算器数を約20%削減でき、テーブルのビット幅を約90%削減できた。一方、 $F_2(x)$ に対しては、加算器数は変わっていないが、テーブルビット幅は約40%削減されている。従って、全体のハードウェアコストは削減されている。

5. むすび

本研究では、任意の関数を回路として実装する際の最適化を行う手法を提案した。最適化には、係数の変化に対する演算誤差や回路規模のトレードオフの関係に対して、多項式表現、因数分解表現の複数の回路表現を持つことで関数の特徴にあった回路を設計することができた。そして、複数の初期解を持たせることで並列タブーサーチを行い、局所探索だけではなく広域探索も可能とし、初期解の値によって解空間を制限することで探索の効率の向上となった。

この結果、任意の関数に対して本手法を用いることで、テーブルのみでの回路設計やチェビシェフ近似による丸めの結果よりも、関数の特徴を考慮した最適な演算回路を探索することができた。

文 献

- [1] 並木 秀明, 永井 亘道, “VHDLによるデジタル回路入門”, 技術評論社, 2006.
- [2] 並木 秀明, 宮尾 正大, 前田 智美, “実用入門 デジタル回路と Verilog HDL”, 1996.
- [3] 高橋 進一, 豊嶋 久道, “デジタル回路設計入門”, 培風館, 2000.
- [4] 浜田 穂積, “近似式のプログラミング”, 培風館, 1995.
- [5] P.T.P. Tang, “Table-Lookup Algorithms for Elementary Functions and Their Error Analysis” Proc. 10th Symp. Computer Arithmetic, vol.40, pp.1,030-1,037,1990.
- [6] M.J.Schulte and J.E.Stine, “Approximating elementary functions with symmetric bipartite tables”, IEEE Trans. on Comput., vol.48, no.9, pp.842, 1999.
- [7] Glover,F., “Tabu Search - Part I”, ORSA Journal on Computing, Vol. 1, No. 3, 1989.
- [8] Glover,F., “Tabu Search - Part II”, ORSA Journal on Computing, Vol. 2, No. 1, 1990.
- [9] 豊嶋, 松本, 佐々木: “確率的タブーサーチによる係数乗算ブロックで構成された FIR デジタルフィルタの設計”, 電子情報通信学会論文誌, VOL. J88-A, 1, pp.99-103, January 2005.