A-001

高速微分演算器について

和田　平司　　三角田　秀実

On The High Speed Differential Caluculator.

Heiji WADA　　Hidemi MISUMIDA

（所属なし）

キーワード　　アルゴリズム、デジタル微分、アーキテクチュア、デジタル回路

あらまし　　　我々の研究チームは微分を高速に行う演算器を開発したので報告する。あるアルゴリズムに則って x と⊿x を一意に定めると、簡単な比較回路と割り算回路で演算器が構成される。

1.　はじめに

デジタル微分を行うのには色々な方法があり、現在も活発に研究が行われている。

我々の研究チームは数理的に関数 f(x)の変数 x をある特定の値に設定し、微小な⊿x をある一意の値に設定すると、瞬時に高速な微分が行えるアルゴリズムを見い出したので報告する。

2.　本論

2-1）一般的に微分は、ある点の傾きを求めたり、ある点の接線を求めたり、その点での値を求めたりする。その微分の公式は f(x)を関数とすると、①式で与えられる。

$$f'(x)=\lim_{⊿x\to 0}\{(f(x+⊿x)-f(x))/⊿x\}\cdot\cdot\cdot ①$$

2-2）アルゴリズムについて

変数 x を奇数の任意の一定の値とする。また、x の微小な値⊿x を偶数のある一意の値に設定すると、瞬時に微分が行える。

そのアルゴリズムの対応を表 1 に示す。

⊿x ＼ x	⊿x=hex2	⊿x=hex4	⊿x=hex8
	hex9	hex5	hex5
		hex9	hex7
		hexd	hex9
			hexb
			hexd

表 1.　⊿x と x の関係

このアルゴリズムは⊿x の値を表 1 のように定めると、一意に x の値が決定される。

条件として、⊿x を偶数に定めると、x は奇数であり、かつ、“0”bit が 1 つ存在するという条件を満たす必要がある。

x が奇数で、かつ 0 が 1bit 存在する必要がある。何故なら⊿x が偶数であるから下位 4bit が hex(⊿x・n)=hex(⊿x・m)となる n,m が存在する。

ただし、(n,m=1,2,・・・,n ≠ m)

両辺の hex の値が違う n と m が存在するとき、そのアルゴリズムは成り立たない。

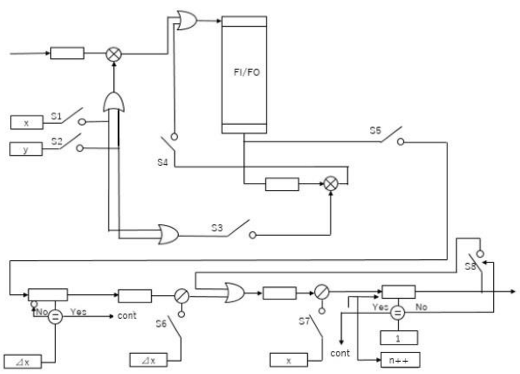

図 1　ブロックダイアグラムを示す。

2-3)図 1 の回路動作を簡単に説明する。上部の回路は関数 f(x+⊿x)を逐次各要素に展開する。すなわちx^{n-j}・⊿x^{j}（j=0、1、・・・、n）を求める回路である。

その要素を一旦 FI/FO に蓄積しておく。その次に逐次 f(x+⊿x)の要素x^{n-j}・⊿x^{j}（j=0、1、・・・、n）を比較回路（すなわち微分を行う回路）へ送り、x^{n-j}・⊿x^{j} (j=1) と⊿x の hex の値を比較し、一致すれば、Accept、不一致であれば Reject する。

Accept された要素だけを 1/⊿x をする。それが終わるとx^{n-j}の(n-j)の値を求める為に x の hex で割ってあげる。

答えは x と(n-j)の値が求まりそれらのx^{n-j}・⊿x^{j}(j=1) の和を求めて係数（n+1-j）を求める。

答えは各レジスタに蓄積しておいて全ての演算が終わると、DISP にて（n+1-j）・x^{n-j}の微分の結果を表示する。

2-4)三角関数の微分の場合は以下の様に行う。

f(θ+⊿θ)=sin(θ+⊿θ)=sinθ・cos⊿θ+cosθ・sin⊿θ

に展開する。そして、f(x+⊿θ)と同じ様に処理を行う。

⊿θ=sin⊿θ=hex4、sinθ=hex9、cos⊿θ=hex2、cosθ

=hexd で与えられる。

2-5)指数関数については、以下の様に展開すると、
f(x)=e^x=(1+x/1!+x^2/2!+・・・)
で表される。
各 x の要素を f(x+⊿x)と同様に求めると f(x)=e^{ax}の微分が求まる。
注)この様に逐次比較一致の処理を行い、一致の場合には Accept 不一致の場合には Reject する。

2-6)図 1 に本システムののブロックダイヤグラムを示す。

2-7)一致不一致の回路構成について

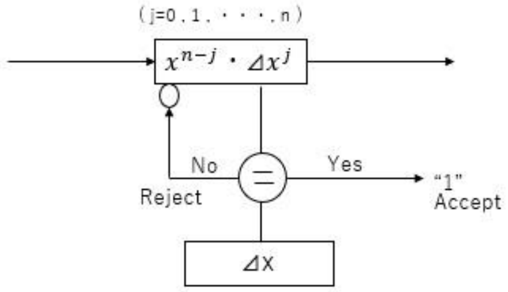

図 2.　Accept or Reject の回路構成

図 2 の説明を行うことにする。関数 f（x+⊿x）の因数分解する。その各要素の hex の値と⊿x の hex の値と等しいか。等しくないかの比較を行う。等しい場合は Accept で次の演算部に DATA が送られる。等しくない場合は Reject する。即ち、DATA を"0"にクリアする。

この操作を遂次行う。このことより⊿x の一次の項だけが遂次、次の演算部に送られる。

3.　考察

3-1)　例えば Ax^3の微分を求める場合、係数 A の値は別のレジスタに保持しておいて、x^3の微分を求める。
そして最終的に微分された値に係数を掛ける。
即ち、3x^2が求まると係数 A をかけて
f'(x)＝（Ax^3）'＝A・3・x^2が求まることになる。

3-2)　三角関数の微分や指数の場合の微分についても同様に係数は別のレジスタに保存しておく。

3-3)　関数 f(x)が f(x)＝x・sinθ で与えられた場合、f'(x)＝g'(x)・h(x)+g(x)・h'(x)の様に展開されることから、此の場合のアルゴリズムは⊿x と（⊿x を hex8 として）x を hex5、sinθを hexb、cosθを hex9、⊿θ=sin⊿θを hex4、cos⊿θを hex2 とするアルゴリズムを与えることによって後は逐次 h(x)・g'(x)の場合は h(x)を別のレジスタに保存しておく。又、g(x)・h'(x)の場合は g(x)を別のレジスタに保存して、h'(x)を逐次求める。
この事により関数 f(x)=x・sinθの微分
f'(x)=sinθ+x・cosθで求まることになる。

4.　結論

ある関数 f(x)の変数 x を、ある奇数の hex の値に設定し、微小な⊿x をある特定の偶数に hex の値に設定すると、比較回路だけで、⊿x の一次の項だけを Accept し、他は Reject 出来るので、高速の微分が可能になる。

この変数 x の hex の値を奇数の hex 値、微小な⊿x の hex 値を偶数の hex 値に設定すると、微分ができるアルゴリズムを見い出した。

残された課題として、特別なまたは、特殊な関数の微分を行う場合にどうするかが残されている。

参考文献

(1)　鈴木"デジタル論理回路・機能入門"
日刊工業出版、2007 年
(2)　Tohmas.H.C 訳　長尾"アルゴリズムの基本"
日経 BP 社、2016 年