논문 2006-43SD-3-3

Newton-Raphson 방식의 제곱근 근사를 위한 초기값의 최적화

(Initial Point Optimization for Square Root Approximation based on Newton-Raphson Method)

최 창 순*, 이 진 용**, 김 영 록*

(Changsoon Choi, Jinyong Lee, and Younglok Kim)

요 약

본 논문은 Newton-Raphson 방법을 기반으로 하는 table-driven 알고리듬에 대해 연구되었다. 특히 본 논문에서는 Newton-Raphson 방법을 이용한 제곱근 근사에 중점을 두었다. Newton-Raphson방법에서 최적화된 초기근사해를 구하게 되면 제곱근 근사의 정확성을 높일 수 있으며, 연산 속도 또한 빨라지게 된다. 그러므로 Newton-Raphson 알고리듬에서 초기근사해를 어떻게 결정하느냐하는 것이 전체적인 알고리듬의 성능을 평가하게 되는 중요한 이슈이다. 본 논문에서는 Newton-Raphson 알고리듬의 초기 근사해를 기하평균을 기준으로 테이블에 저장, 연산의 속도와 최대 오차율을 줄일 수 있음을 확인하였다.

Abstract

A Newton-Raphson Method for table driven algorithm is presented in this paper. We concentrate the approximation of square root by using Newton-Raphson method. We confirm that this method has advantages of accurate and fast processing with optimized initial point. Hence the selection of the fitted initial points used in approximation of Newton-Raphson algorithm is important issue. This paper proposes that log scale based on geometric wean is most profitable initial point. It shows that the proposed method givemore accurate results with faster processing speed.

Keywords : Newton-Raphson, square root, iteration

I. 서 론

Digital signal processing에서 사용되는 제곱근 연산은 기본적이고 필수적인 과정인 반면 다른 기본 연산에 비해 큰 복잡도와 메모리를 요구하기 때문에 하드웨어의 구현 에 있어 신속성과 간소화에 큰 걸림돌이 된다[1]. 이에 대해 계산량을 줄이면서도 구성이 간단한 모듈 구현을 위한 방법 여러 방법이 개발되었는데, Newton-Raphson 알고리듬과 보간법 알고리듬, shift/subtract 알고리듬 방식이 널리 사용 되어진다[2].

Newton-Raphson 알고리듬 방식은 간단하면서도 효율적으로 근사해를 구하는 이점을 갖고 있다[1]. 그럼에도 불구하고 자체 정확도를 높이기 위한 과정이 포함되지 않는다는 단점을 안고 있다[3]. 본 논문은 look-up-table (LUT)과 초기 근사해의 최적화 구성을 통해 정확도를 높이는 과정을 설명한다. 본 논문의 구성은 다음과 같다. 제 II장 에서는 Newton-Raphson 알고리듬을 이용한 제곱근의 근사 방법에 대해 소개하며, 제 III장에서는 이 알고리듬을 이용한 제곱근 연산에 있어 최적의 초기근사해 선택과 LUT 구성의 최적화에 대해 기술 하였으며, 제 IV장에서는 결과 및 분석

* 정회원, ** 학생회원, 서강대학교 공과대학 전자공학과 (Dept. of Electronic Engineering, Program of Integrated Biotechnology and IDEC, Sogang University)

※ 본 논문은 2005년도 서강대학교 바이오융합 사업단의 지원을 받아 수행되었으며, 설계 Tool은 IDEC의 지원을 받았음.

접수일자: 2005년10월15일, 수정완료일: 2006년3월7일

에 대해 기술하였다. 마지막으로 제 V장에서는 본 논문의 결론을 제시하였다.

II. Newton-Raphson알고리듬을 이용한 제곱근 근사 방법

Newton-Raphson 알고리듬은 초기값에서 함수 $f(x)$와의 접선과 x축과의 교점을 구하는 방법을 반복하여 $f(x)=0$의 해의 근사값을 수치적으로 구하는 방법이다[1]. 초기값과 그 함수의 값들은 LUT에 저장해 놓으며, 초기값과 반복 횟수에 따라 근사의 정확도가 결정된다. 반복 횟수가 고정되어 있다고 가정한다면, 초기값을 최적화함으로써 근사 오차를 줄일 수 있거나 LUT의 크기를 줄일 수 있다.

그림1에서 보는 바와 같이 초기 근사값 $x=x_0$에서 함수와의 접선, 즉 $p_0(x_0, f(x_0))$를 지나고 기울기가 $f'(x_0)$인 직선과 x축과의 교점을 구하여 새로운 근사해 x_1을 결정한다. 이러한 과정을 반복하여 $f(x)=0$의 해를 근사적으로 구할 수 있다. 반복 과정에서 현재의 근사값과 다음 근사값의 관계는 수식(1)로 표현된다.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1}$$

임의의 수 p의 제곱근 $\sqrt{p}$의 근사값은 함수를 $f(x)=x^2-p$로 정의하여 구할 수 있으며, 수식(1)은 수식(2)와 같이 표현된다.

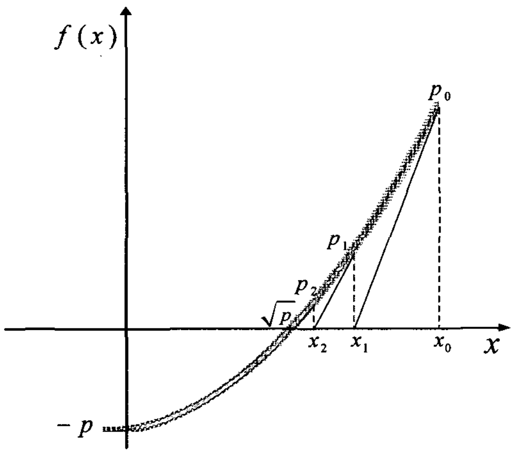

그림 1. Newton-Raphson 알고리듬을 이용한 근사 방법
Fig. 1. Approximation method take Newton-Raphson algorithm.

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{p}{x_n}\right) \tag{2}$$

이 결과는 초기 근사값이 구하고자 하는 제곱근 $\sqrt{p}$보다 작은 경우와 큰 경우에 상관없이 만족된다.

III. 초기 근사해의 최적화

1. 최적의 초기값 선택

주어진 p값이 LUT의 두 개의 인접한 a와 b 사이에 존재하는 경우, $a<p<b$, 초기 근사값을 a로 정한 경우와 b로 정한 경우의 오차율을 그림2에서 보여 준다. 여기서, 계산 범위[1, 2]를 일정한 간격으로 나누었으며, 각 반복단계에서의 오차율은 수식(3)과 같이 정의된다[1].

$$E_n = \frac{\left|x_n - \sqrt{p}\right|}{\sqrt{p}} \tag{3}$$

기존의 일반적인 방식에서는 주어진 범위 내에서 초기 근사값이 일정한 간격으로 주어지며, 구하려는 값과 제일 가까운 값을 초기값으로 정하게 된다. 즉, p값이 LUT의 두 개의 인접한 a와 b 사이에 존재하는 경우, 두 값의 산술평균값 $\frac{a+b}{2}$을 기준으로 하여 초기 근사값을 $\sqrt{a}$나 $\sqrt{b}$로 결정하게 된다. 하지만, 이 기준은 그림2에서 보여주는 두 곡선의 교점이 되지 않으므로, 제곱근 계산을 위한 최적의 초기값을 선택하는 최적의 선택 기준이 될 수가 없다.

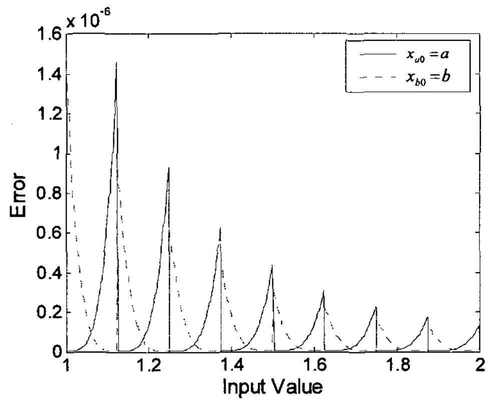

그림 2. 2회의 반복을 통한 제곱근의 오차율 비교
Fig. 2. Error rate of 2 times repeated square root.

제곱근을 위한 최적의 초기값 선택을 위하여 첫 번째의 근사 오차율을 비교하여 적은 오차율을 보이는 값을 초기 근사해로 선택한다. 초기값 $x_{a0} = \sqrt{a}$또는 $x_{b0} = \sqrt{b}$와 수식(2)에서 얻어진 첫 번째 근사해를 각각 x_{a1} 또는 x_{b1}라고 하자. 이 두 값들은 반드시 $\sqrt{p}$ 보다 큰 값을 갖게 되며, 둘 중 작은 값이 적은 오차율을 의미하게 된다. 따라서 초기 근사값의 기준은 수식(4)가 된다.

$$x_{a1} - x_{b1} = 0 \tag{4}$$

여기서,

$$x_{a1} - x_{b1} = \frac{1}{2\sqrt{ab}}(p - \sqrt{ab})(\sqrt{b} - \sqrt{a}) \tag{5}$$

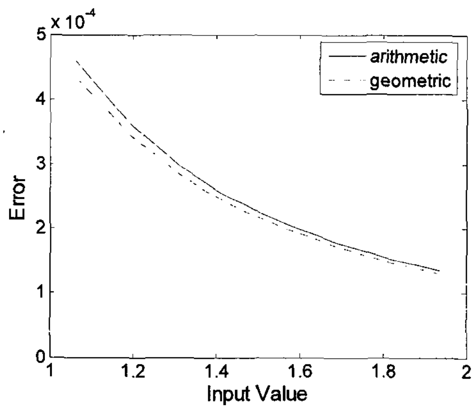

그림 3. 초기 근사해 선택에 따른 오차율 비교 (1회 반복)

Fig. 3. Comparison of error rate followed at selecting initial point. (repeat once)

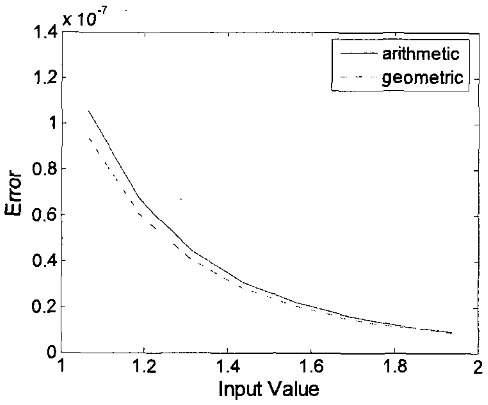

그림 4. 초기 근사해 선택에 따른 오차율 비교 (2회 반복)

Fig. 4. Comparison of error rate followed at selecting initial point(repeat twice).

이며, $\sqrt{ab}$와 $(\sqrt{b} - \sqrt{a})$은 항상 양의 값을 가지므로, 결국 $p = \sqrt{ab}$가 수식(4)를 만족하는 기준이 된다. 즉, $p < \sqrt{ab}$인 경우 초기 근사해 값을 a로 결정하고, $p > \sqrt{ab}$이면, 초기근사해를 b로 결정하면, 오차율을 최소화 할 수 있다.

근사해 선택을 위한 기준으로 기존의 산술평균과 제안된 기하평균을 이용했을 때의 오차율의 비교를 그림3과 4에서 보여 준다.

2. 최적의 LUT 구성

오차율을 줄이기 위한 근사해 선택의 기준이 된 기하평균은 LUT의 구성에 있어서도 테이블의 크기와 오차율을 줄이는 최적화의 기준이 된다. 구간별 최대 오차율을 보이는 지점에 새로운 테이블 값을 추가시킴에 따라 가장 효율적으로 오차율을 줄일 수 있다..

구하고자 하는 제곱근의 범위가 $[a, b]$이며, LUT에 a, b의 초기근사해가 포함되어 있다면, a, b에 따른 근사값은 q_{a1}, q_{b1}이 되며, 수식(3)을 통해 각각 다음과 같은 오차율을 얻는다.

$$E_a = f(a,p) = \frac{\left|q_{a1} - \sqrt{p}\right|}{\sqrt{p}}$$

$$E_b = f(b,p) = \frac{\left|q_{b1} - \sqrt{p}\right|}{\sqrt{p}} \tag{6}$$

그림 5에서 보는 바와 같이 근사값의 오차율은 $E_a = E_b$인 점 c에서 최대가 되며, 이점에서 각각의 오차율은 다음과 같다.

$$E_a = \frac{1}{2}\left(\sqrt{a} + \sqrt{c}\right)\left(\frac{1}{\sqrt[4]{ac}}\right) - 1$$

$$E_b = \frac{1}{2}\left(\sqrt{c} + \sqrt{b}\right)\left(\frac{1}{\sqrt[4]{cb}}\right) - 1 \tag{7}$$

점 c는 $E_a = E_b$의 관계를 통해 도출되며, 더욱 정확한 근사값을 위한 초기근사해의 추가 시 가장 효율적인 위치가 된다.

$$c = \sqrt{ab} \tag{8}$$

근사값의 정확도를 더욱 높이기 위해 초기근사해의

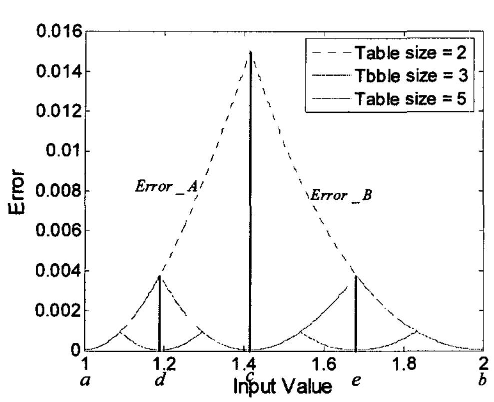

그림 5. LUT의 크기에 따른 최대 오차율의 비교
Fig. 5. Comparison of error followed Table sizes.

값을 추가 한다면, 추가되는 초기근사해는 각각 $d=\sqrt{ac}$, $e=\sqrt{cb}$이며, 그림 5는 LUT에 저장되는 초기근사해 수에 따른 1회 근사 시 오차율 비교를 나타낸다.

요구되는 제곱근의 범위와 LUT의 크기 및 초기근사해의 구성은 다음과 같은 관계를 보인다.

추가되는 초기근사해는 LUT 내에 인접한 두 값 사이에 생성되므로 LUT의 address 수는 $N=2^r+1$이 된다. 제곱근의 요구 범위가 $[a,b]$인 경우 초기근사해 $q(n)$은 수식 (9)와 같으며, 초기근사해 $q(n)$을 선택하는 기준 $f(n)$은 수식(10)과 같다.

$$q(n)=a^{1-\frac{n}{N}}b^{\frac{n}{N}} \tag{9}$$

$$f(n)=q(n)a^{\frac{1}{2N}}b^{\frac{1}{2N}} \tag{10}$$

그림 6과 그림 7은 p의 범위가 [1,2]이고, LUT의

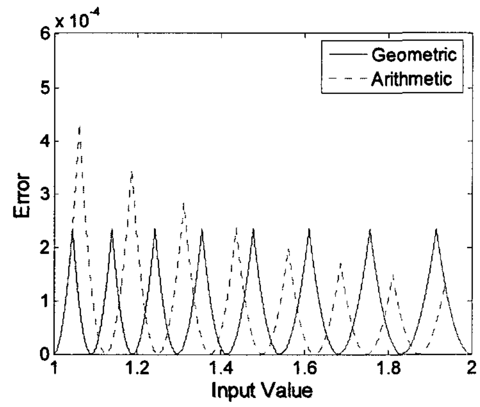

그림 6. LUT 구성에 따른 오차율의 비교(1회 반복)
Fig. 6. Comparison of error by selecting criterion method (repeat once).

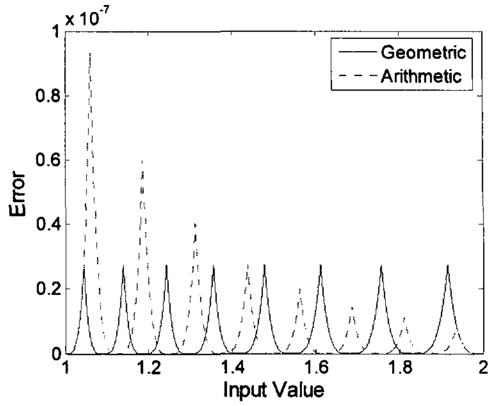

그림 7. LUT 구성에 따른 오차율의 비교(2회 반복)
Fig. 7. Comparison of error by selecting criterion method (repeat twice).

address 수가 $2^3+1=9$인 경우 LUT 구성의 방법에 따른 오차율을 비교한 것이다. 근사 횟수가 증가하여도 구간별 최대오차율이 일정하게 유지된다.

IV. 결과 및 분석

표 1은 LUT의 크기가 $(2^3+1)*m(bit)=9m$ 인 경우를 기준으로 LUT의 구성 및 초기 근사해의 선택 방법에 따른 오차율 비교 결과로 제안하는 구성이 가장 작은 오차율을 보인다.

1.0 e-15의 정확도를 기준으로 많이 사용되는 LUT을 이용한 2차 보간법과 연산 복잡도 및 LUT 크기를 비교한 결과는 표 2와 같으며, 연산복잡도 크기에 비해 LUT 크기에 있어 크게 유리함을 참고문헌을 통해 확인하였다[4]. 요구되는 정확도가 낮은 경우 근사 횟수가 줄어들어 연산복잡도 또한 줄어들게 된다.

표 1. LUT구성과 초기근사해 선택에 따른 오차율 비교
Table 1. Comparison of error by method of LUT compose and initial value select.

Table구성	Equal.	Equal.	Geometric
초기근사해선택	Arithmetic	Geometric	Geometric
근사 1회	4.8e-4	4.2e-4	2.3e-4
근사 2회	9.7e-8	9.1e-8	2.9e-10
근사 3회	5.0e-15	4.3e-15	2.2e-16

표 2 제안하는 알고리듬과 2차 보간법의 성능 비교

Table 2. Comparison of performance between proposed algorithm and 2nd order interpolation.

		제안된 알고리듬 (근사 3회)	2차 보간법
오차값(정확도)		2.2 e-16	3.56 e-15
LUT	Table size (word)	9 m	512 m
	m (bits/word)	32	48
	Total bits	288 bits	27.5k bits
HW 복잡도		2 Multipliers 1 Addition	2 Multipliers 1 Addition
연산 시간		6 MAC	2 MAC

V. 결 론

본 논문에서는 Newton-Raphson 알고리듬을 바탕으로 제곱근의 계산에 대한 연산 복잡도와 LUT의 크기를 줄일 수 있는 최적화 방법에 대해 설명하였다.

Newton-Raphson 알고리듬은 다른 알고리듬에 비해 연산복잡도에 있어 커다란 이점을 갖고 있지만 상대적으로 낮은 정확도와 LUT 포함에 따른 크기의 증가가 문제점으로 지적되고 있다.

제안된 최적화 과정은 Newton-Raphson 알고리듬의 정확도를 향상 시킴과 동시에 LUT의 크기를 줄여, 연산속도 및 크기에 있어서 효율성을 도모한다.

더 높은 정확도가 요구 되는 경우 근사 횟수가 늘어나거나 LUT의 크기가 증가되어야 함 또한 확인하였는데, 이와 같은 문제에 대한 대안으로 Newton-Raphson 알고리듬 자체의 정확도를 개선하는 연구가 추가되어야 한다.

참 고 문 헌

[1] Behrooz Parhami "Computer Arithmetic : Algorithms and Hardware Designs" Oxford University Press, New York, 2000.

[2] Ryuki Kawamata and Shugang Wei "Suqare-Rooting Algorithm Using Signed-Digit Arithmetic" ITC-CSCC2005, Vol.2. pp.559-560, July, 2005.

[3] Reza Hashemian "Square Rooting Algorithm for Integer and Floating-Point Numbers", IEEE Transactions on Computers, Vol.39. No.8. August, 1990.

[4] M.J.Schulte and E.E.Swartzlander Jr, " Hardware desing for exactly rounded elementary functions", IEEE Transactions on Computers 43.8. pp.964-973, 1994.

저 자 소 개

최 창 순(정회원)
2003년 목원대학교 전자공학과 학사 졸업.
2005년 서강대학교 전자공학과 석사 졸업
2005년~현재 트루모바일 연구원.
<주관심분야 : VLSI 설계를 위한 DSP 알고리즘>

이 진 용(정회원)
1999년 서강대학교 물리학과 학사 졸업.
2005년~현재 서강대학교 전자공학과 석사과정.
<주관심분야 : VLSI 설계를 위한 DSP 알고리즘, 무선 통신 신호처리>

김 영 록(정회원)
1991년 서강대학교 전자공학과 학사 졸업.
1993년 Polytechnic Univ. 공학석사 졸업.
1998년 Polytechnic Univ. 공학박사 졸업.
1998년~1999년 AT&T연구소 연구원
1999년~2003년 InterDigital Comm. Crop.,연구원
2003년~현재 서강대학교 전자공학과 조교수
<주관심분야 : VLSI 설계를 위한 DSP 알고리즘, 레이다 및 무선 통신 신호처리>