

로그수체계 기반의 저전력/저면적 제산기 및 제곱근기 회로 설계

김채현^{*} · 김종환^{*} · 이용환^{*} · 신경욱^{*}

^{*}금오공과대학교 전자공학부

A Design of Low-power/Small-area Divider and Square-Root Circuits based on Logarithm Number System

Chay-Hyeun Kim^{*} · Jong-Hwan Kim^{*} · Yong-Hwan Lee^{*} · Kyung-Wook Shin^{*}

^{*}School of electronic Eng., Kumoh National Institute of Technology

E-mail : cogus7892@kumoh.ac.kr

요 약

본 논문에서는 그래픽 프로세싱 분야와 디지털 신호 처리 분야에 응용될 수 있는 로그수체계(Logarithm Number System; LNS) 기반의 제산기와 제곱근기를 설계하였다. 설계된 제산기와 제곱근기는 부동소수점 대신 16.16의 고정소수점 방식을 사용하여 모바일 환경에서 저전력/저면적으로 동작하도록 하였다. 설계된 제산기와 제곱근기는 이진수-로그 변환기, 감산기, 로그-이진수 변환기 등으로 구성되어 있다. 특히, 이진수-로그 변환시 룩업테이블(Look Up Table; LUT)을 사용하지 않고 6-영역의 근사화 방법을 이용한 조합회로로 구현함으로써, 기존의 룩업테이블로 구현한 방식에 비해 게이트 수가 감소되도록 하여, 제산기 3,130, 제곱근기 1,280 게이트로 구현되었다. 연산정밀도를 높이기 위해 여러 보상방법을 적용하였으며 연산 정밀도 분석결과 평균 퍼센트 에러가 각각 3.8% 와 4.2%로 평가되었다.

ABSTRACT

This paper describes a design of LNS-based divider and square-root circuits which are key arithmetic units in graphic processor and digital signal processor. To achieve area-efficient and low-power that is an essential consideration for mobile environment, a fixed-point format of 16.16 is adopted instead of conventional floating-point format. The designed divider and square-root units consist of binary-to-logarithm converter, subtractor, logarithm-to-binary converter. The binary to logarithm converter is designed using combinational logic based on six regions approximation method. As a result, gate count reduction is obtained when compared with conventional lookup approach. The designed units is 3,130 gate count and 1,280 gate count. To minimize average percent error to 3.8% and 4.2%, error compensation method is employed.

키워드

Logarithm Number System, Divider, Square-Root, 3D Graphic

1. 서 론

본 논문에서는 그래픽 프로세싱 분야와 디지털 신호 처리 분야에 응용될 수 있는 로그수체계 기반의 제산기와 제곱근기를 설계하였다. 로그수체

계에서 제산/제곱근 연산은 감산/쉬프트 연산만으로 구현할 수 있다. 이진수-로그 변환 및 로그-이진수 변환시 룩업테이블을 사용하지 않고 조합 회로로 구현하였으며, 일반적인 부동소수점 보다는 고정소수점 실수 표현 방식을 사용하여 면적

과 하드웨어 구조의 복잡도를 감소시켰다.

2장에서는 로그수체계 기반의 제산과 제공근에 대해 기술하고, 3장에서는 로그수체계 기반의 하드웨어 구현에 관하여, 4장에서는 설계 검증 및 성능 평가에 대해 기술한다.

II. LNS 기반의 제산기와 제공근기

본 논문에서는 16.16 고정소수점 실수 기반 제산기와 제공근기를 설계하였다. 부동소수점 연산은 고정소수점 연산보다 더 높은 정밀도와 큰 동적 범위를 제공하나, 추가적인 하드웨어가 필요하고, 전력소모가 크며, 연산속도가 느린 단점이 있다. 고정소수점 실수 표현을 사용하는 것이 저전력/저면적으로 구현할 수 있다.^[1]

이진수 N 이 $2^j \leq N \leq 2^k$ ($j=0, \pm 1, \pm 2, \dots$, $k=0, \pm 1, \pm 2, \dots, k \geq j$) 범위의 수일 때, 다음 식 (1), (2)와 같이 Mitchell의 로그 근사화 식으로 표현된다.^[2]

$$N = \sum_{i=j}^k 2^i z_i = 2^k + \sum_{i=j}^{k-1} 2^i z_i = 2^k \left(1 + \sum_{i=j}^{k-1} 2^{i-k} z_i \right) \quad (1)$$

$$= 2^k (1 + m)$$

$$N = 2^k (1 + m) \rightarrow \log_2 N = k + \log_2 (1 + m) \quad (2)$$

m 은 최상위 '1'의 오른쪽에 위치해 있는 모든 비트들을 포함하고 있고, k 는 로그의 특성(characteristic)으로 로그수체계로 변환된 수가 인접한 2의 거듭제곱 사이에 있는 것을 의미한다.

기존의 제산 알고리즘은 크게 2가지 부류로 나뉜다. 첫째 방식은 가산/감산과 쉬프트 동작을 조합하여 구현하는 방법이며, restoring 알고리즘, non-restoring 알고리즘, SRT 알고리즘이 이에 속한다. 둘째 방식은 승산을 이용하여 제산을 수행하는 방법으로, 분모·분자에 동일한 수를 곱하는 convergence 방식과 분모의 역수를 구해 곱하는 NR(Newton-Raphson) 방식등이 이에 속한다. 본 논문에서는 이와 달리 식(3-a), (4-a)을 식(3-b), (4-b)와 같이 로그수체계로 변환후 각각 감산과 쉬프트를 사용하여 제산과 제공근을 구현하였다.^[3]

$$P = \frac{A}{B} = 2^{(\log(A) - \log(B))} \quad (3-a)$$

$$\log(P) = \log\left(\frac{A}{B}\right) = \log(A) - \log(B) \quad (3-b)$$

$$Q = \sqrt{C} = 2^{\frac{1}{2} \log(C)} \quad (4-a)$$

$$\log(Q) = \log(\sqrt{C}) = \frac{1}{2} \log(C) \quad (4-b)$$

III. LNS 기반의 제산기와 제공근기 회로 설계

그림 1과 그림 2는 설계된 제산기와 제공근기의 전체 구조이며, 크게 나누어 이진수-로그 변환기, 감산기, 로그-이진수 변환기 등으로 구성된다. 설계된 제산기와 제공근기는 이진수-로그 변환기와 로그-이진수 변환기의 구조는 동일하며, 로그값으로 변환된 수를 제산기는 감산하고 제공근기는 1-비트 오른쪽 쉬프트 시킨다.

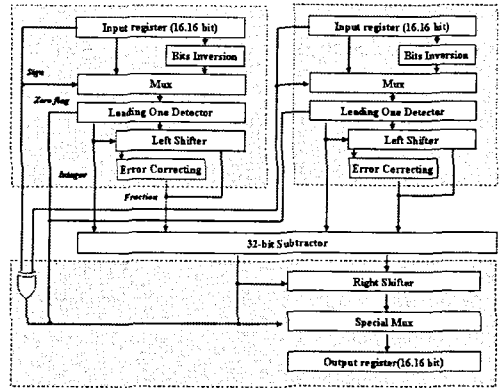


그림 1. 로그 수체계 기반의 제산기의 구조

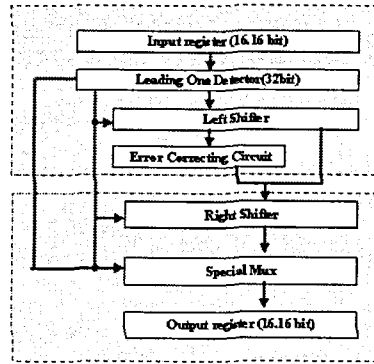


그림 2. 로그 수체계 기반의 제공근기의 구조

3.1 로그 변환기

일반적으로 로그수체계의 이용시 이진수에서 로그값의 변환은 룩업 테이블을 이용하였다. 이 방법은 필요한 정밀도까지 연산이 가능하나 그에 따른 하드웨어 크기가 지수적으로 증가한다. 따라서 본 논문에서는 룩업 테이블을 사용하는 대신 정밀도를 허용 오차 범위 내로 줄이는 조합회로로 구현함으로써, 면적이 대폭 감소되도록 설계하였다.

3.1.1 N-bit Leading One Detector

부동소수점 연산이나 로그수체계에서 사용되는 LOD는 비트열들 중에 최상위 '1'의 위치를 찾는다. 최근 발표된 문헌에 의하면, Schmoockler는 부

동소수점에 쓰이는 leading one/zero anticipator 나 leading one/zero predictor 등에 관하여 연구하였다.^[4] Oklobdzija는 leading-zero detector에 관하여, Bruguera와 Lang은 leading-one predictor에 관하여 기술해 놓았다.^[5-6] 일반적으로 LOD의 회로구조는 직/병렬 구조로서 동작속도가 느린 단점이 있으며, 또한 최상위 '1'의 비트수를 찾은 후 ROM을 거쳐 자리수를 표현하는 2단계 과정을 거친다. 본 논문은 Suzuki가 제안한 병렬 구조인, 최상위 1의 자리수를 1단계에 표현하는 방법을 사용한다.^[7] 그림 3은 8-비트 LOD 회로 이고 그림 4는 16-비트 LOD 회로도이다. 본 논문에서는 32-비트 LOD 회로를 사용하였으며 작은 비트의 LOD로 N-비트 LOD를 구현할 수 있다.

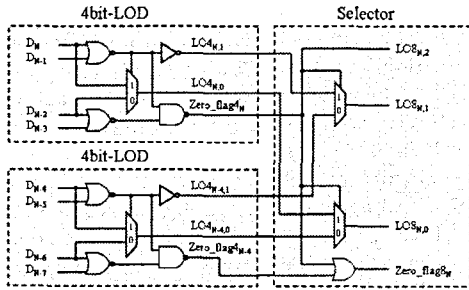


그림 3. 8-비트 LOD 회로도

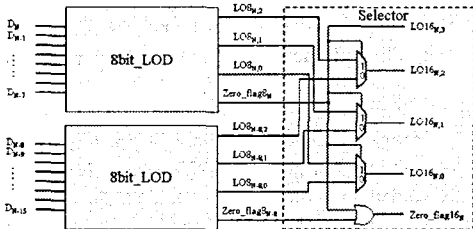


그림 4. 16-비트 LOD 회로도

3.1.2 N-bit Shifter

N-비트만큼의 자리이동은 이진수-로그변환과 로그-이진수 변환 시에 최상위 '1'의 오른쪽 위치에 비트열을 제공하는 역할을 한다. 배럴 쉬프트 (barrel shifter)는 로그변환 시나 부동소수점 연산에서 흔히 사용되며, 왼쪽, 오른쪽 자리이동과 왼쪽, 오른쪽 회전의 4가지 기능을 제공한다. 그러나 로그변환 시에는 왼쪽, 오른쪽 쉬프트 기능만이 필요하므로 배럴 쉬프트를 이용하지 않고 구현하는 방법으로 설계하였다. 설계된 쉬프트는 배럴 쉬프트와 마찬가지로 입력 데이터 워드, 출력 데이터 워드, 제어 입력으로 구성되며 배선의 배치로 왼쪽, 오른쪽 쉬프트로 구현할 수 있다. 제어 입력의 비트 수 n에 따라 n개의 스테이지로 구성된다. 그림 5는 왼쪽 16-비트 쉬프트이며, 본 논문에서는 32-비트 왼쪽 쉬프트와 32-비트 오른쪽 쉬프트를 사용하였다.

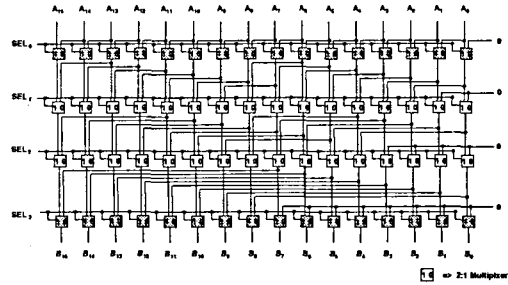


그림 5. 16-bit left shifter

3.1.3 Error Correcting Circuit

초기에 제안된 Mitchell의 방법은 소수 3번째 자리의 정밀도만 가지고 있으므로 정밀도를 높이기 위한 방법들이 Combet^[8], Hall^[9], Sangregory^[10] 등에 의해서 제안되었다. 제안된 방법들 중 Khalid가 제안한 6-영역의 에러 보상방법은 소수 7번째 자리의 정밀도를 가지고 있다.^[2] 이를 이용하여 로그변환 시 에러를 줄일 수 있는 방법을 사용하였다. 다음 식(5)는 Khalid의 6-영역의 에러 보상 방법이다. 그림 6는 식(5)의 회로 구현이다.

$$\text{Fraction} = \begin{cases} (m + 2^{-2} m_{\text{MSB}6}) & \text{for } m \in [0.0, 0.625) \\ (m + 2^{-2} m_{\text{MSB}6} + 2^{-4}) & \text{for } m \in [0.625, 0.25) \\ (m + 2^{-4} + 2^{-7} + 2^{-8}) & \text{for } m \in [0.25, 0.375) \\ (m + 2^{-4} + 2^{-6} + 2^{-7}) & \text{for } m \in [0.375, 0.625) \\ (m + 2^{-4} + 2^{-7}) & \text{for } m \in [0.625, 0.75) \\ (m + 2^{-2} m_{\text{MSB}6}) & \text{for } m \in [0.75, 1.0) \text{ at } \bar{m} = (1 - m_{\text{MSB}6} \cdot 2^{-4}) \end{cases} \quad (5)$$

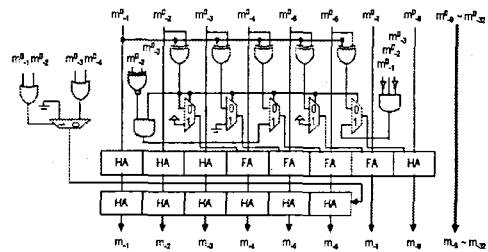


그림 6. 에러 정정 회로

3.2 N-bit Subtractor, 1-bit Right Shifter

로그수체계에서는 제산과 제곱근 연산을 감산과 1-bit 오른쪽 쉬프트 연산으로 대체할 수 있다. 본 논문에서는 carry select 방식의 감산기를 사용하였으며, 1-bit 쉬프트는 로그값으로 변환된 입력 비트들의 배선으로 구현하였다.

3.3 로그-이진수 변환기

로그값을 이진수로 변환하기 위해서는 이진수-로그 변환 과정의 역변환을 거친다. characteristic에서 우선 leading one의 자리를 결정하고 분수 (fraction) 부분을 leading one의 오른쪽에 위치시킨다.

IV. 설계 검증 및 성능 평가

설계된 연산기의 검증은 Matlab 모델링으로부터 각각 50,000개의 랜덤 테스트 벡터를 생성하고, 이를 verilog testbench에서 읽어 논리 시뮬레이션을 수행하였다. 시뮬레이션 결과와 Matlab 모델링으로 생성된 결과를 비교하여 논리 동작을 검증하였으며, Matlab을 이용하여 연산 정밀도를 분석하였다.

설계된 제산기와 제공근기의 평균 오차는 식(6)과 같이 정의되며, PM은 Matlab에서 생성된 결과값, PV는 verilog에서 출력된 결과값을 나타낸다. 평균 퍼센트 오차는 식(7)과 같이 정의 되며, 정확한 결과 값에 비해 오차가 차지하는 비중의 척도를 나타낸다.

$$E_V = \frac{\sum |PM - PV|}{M} \quad (6)$$

$$PE_V = \frac{\sum \left| \frac{PM - PV}{PM} \times 100 \right|}{M} \quad (7)$$

설계된 제산기와 제공근기의 면적은 Synopsys와 0.25- μ m 표준 셀 라이브러리를 이용한 논리합성을 통하여 분석하였다. 다음 표 1은 설계된 제산기와 제공근기의 연산정밀도와 게이트 수를 평가한 것이다.

표 1. 설계된 제산기와 제공근기의 연산 정밀도 및 게이트 수

	평균 에러	평균 퍼센트 에러	게이트 수
제산기	736	3.8%	3,130
제공근기	5.4	4.2%	1,280

V. 결 론

본 논문에서는 로그 수체계 기반의 제산기와 제공근기를 설계하였다. 면적의 감소를 위하여 록업테이블 대신 조합회로로 구현함으로써 게이트 수 감소를 이루었고, 이진수-로그 변환시 에러보상을 하였다. 측정된 연산정밀도는 각각 평균 퍼센트 오차가 3.8%, 4.2%로 평가 되었다.

일반적으로 휴대형 환경의 경우 부동소수점 연산보다는 고정소수점 표현을 이용하여 저전력/저면적으로 구현하는 것이 바람직하다. 따라서, 설계된 연산기들은 게이트 수와 평균 오차 특성이 우수하여, 그래픽 프로세싱 및 DSP 분야의 저전력 설계에 폭 넓게 이용 가능할 것이다.

참고문헌

- [1] "Using Fixed-Point Instead of Floating Point for Better 3D Performance", <http://www.devx.com/Intel/Article/16478>
- [2] H. A. Khalid, E. S. Raymond, "CMOS VLSI Implementation of a Low-Power Logarithmic Converter", IEEE Trans. on Computer vol. 52, no. 11, pp. 1421-1433, Nov. 2003.
- [3] J. N. Mitchell, "Computer Multiplication and Division Using Binary Logarithms", IRE Trans. electronic Computer, vol. 11, pp. 512-517, Aug. 1962.
- [4] M. Schmookler, K. Nowka, "Leading Zero Anticipator and Detection : A Comparison of Methods.", Proc. 15th Symp. Computer Arithmetic, Jun. 2001.
- [5] V. Oklobdzija, "Comments on Leading-Zero Anticipatory Logic for High-speed Floating Point Addition", IEEE J. Solid-State Circuits, pp. 292-293, 1997.
- [6] J. Bruguera, T. Lang, "Leading-One Prediction with Concurrent Position Correction", IEEE Trans. on Computer, vol. 48, no. 10, pp. 298-305, Oct. 1999.
- [7] H. Suzuki, H. Morinaka, H. Makino, Y. Nakase, K. Mashiko, T. Sumi, "Leading-Zero Anticipatory Logic for High-Speed Floating Point Addition", IEEE J. Solid-state Circuits, pp. 1157-1164, 1996.
- [8] M. Combet, H. Zonneveld, L. Verbeđa, "Computation of the Base Two Logarithm of Binary Numbers", IEEE Trans. Electronic Computers, vol. 14, pp. 863-867, Dec. 1965.
- [9] E.L. Hall, D.D. Lynch, S.J.Dwyer III, "Generation of products and quotients Using Approximate Binary Logarithms for Digital Filtering Applications", IEEE Trans. Computers, vol. 19, pp. 97-105, Feb. 1970.
- [10] S.L. Sangregory, R.E. Siferd, C. Brother, D. Gallagher, "A Fast, Low-Power Logarithm Approximation with CMOS VLSI Implementation", Proc. IEEE Midwest Symp. Circuits and Systems, Aug. 1999.

※2005년도 IT SoC 핵심설계인력양성 사업의 SoC 전공인증과정 지원에 의한 연구 결과의 일부임.
 ※반도체설계교육센터(IDECS)의 CAD Tool 지원에 감사드립니다.