

# 고속 여분 부동 소수점 이진수의 제산/스퀘어-루트 설계 및 제작

김 종 섭, \*이 중 화, \*\*조 상 복  
서라벌대학, \*울산대학교, \*\*울산대학교  
전화 : 052-259-2202 / 핸드폰 : 017-588-2202

## A Design and Fabrication of the High-Speed Division/Square-Root using a Redundant Floating Point Binary Number

Jong-Seop Kim, \*Jong-Hwa Lee, \*\*Sang-Bock Cho  
School of Electricity, Electronics & Computer Science, Sorabol Junior College /  
School of Electrical Engineering, University of Ulsan  
E-mail : sjsuus@sorabol.ac.kr / sbcho@uou2.ulsan.ac.kr

### Abstract

This paper described a design and implementation of the division/square-root for a redundant floating point binary number using high-speed quotient selector. This division/square-root used the method of a redundant binary addition with 25MHz clock speed. The addition of two numbers can be performed in a constant time independent of the word length since carry propagation can be eliminated. We have developed a 16-bit VLSI circuit for division and square-root operations used extensively in each iterative step. It performed the division and square-root by a redundant binary addition to the shifted binary number every 16 cycles. Also the circuit uses the nonrestoring method to obtain a quotient. The quotient selection logic used a leading three digits of partial remainders in order to be implemented in a simple circuit. As a result, the performance of the proposed scheme is further enhanced in the speed of operation process by applying new quotient selection addition

※ 본 논문은 산업자원부에서 지원을 받는 반도체설계교육센터(IDECS)의 후원으로 이루어졌습니다.

logic which can be parallelly process the quotient decision field. It showed the speed-up of 13% faster than previously presented schemes used the same algorithms.

### I. 서론

본 논문은 제산(division)과 스퀘어-루트(square-root)에 대한 산술 연산을 위해 동일한 형태의 알고리즘을 사용하여 공유된 조합 논리 회로를 설계하였다. 이 회로는 여분 가산 배열(redundant addition array)과 지수 선택 논리(quotient selection logic), 그리고 진행 변환<sup>[1]</sup>(on-the-fly conversion)의 세 가지 주요한 요소로 구성된다. 여분 가산 배열은 일반적인 여분 가산 셀을 배열하여 피연산자와 지수 값을 병렬 가산 처리하여 나머지를 얻는다. 그리고 지수 선택 논리는 지수 숫자를 선택하여 여분 가산 배열과 진행 변환으로 의하여 지수를 결정한다.

부동소수점에 대한 고속 제산 알고리즘은 여분 이진수 체계를 사용한다. 제산 알고리즘을 기초로한 스퀘어-루트 알고리즘은 여분 이진수 중에 가장 탁월한 성능을 갖는다. 이와 같은 두 개의 알고리즘은 비복원(non-restoring) 방법을 사용하여 정상적인 피연산자를 지수 숫자와 나머지 부분의 여분 표현으로 만들고, 또한 나머지 부분의 최상위 숫자의 정수에 의해 지수 숫

자 선택을 수행한다.

이 회로의 주요 요소 중에 하나인 지수 선택 논리는 피연산자의 개수에 독립적인 모든 반복 연산에서 나머지 부분의 몇 개의 최상위 비트를 사용하여 지수를 선택한다. 본 논문에서는 지수 선택에 필요한 나머지 부분의 최상위 비트에 대해 세 개의 부호화된 형태를 사용하였으며, 이것은 제산/스퀘어-루트의 지수 최대 값을 줄임으로써 연산 반복 시간을 단축하는데 중요한 역할을 한다. 이러한 관점에서, 제안된 설계는 공유된 제산과 스퀘어-루트에 대하여 지수 숫자는 최대 여분 숫자인  $\{-1, 0, +1\}$ 에 속한 알고리즘에 의해 만들어진다. 또한 만들어진 지수 숫자는 진행 변환 알고리즘을 사용하여 이진수 형태로 변환된다.

## II. 알고리즘 및 설계

### 1. 제산 알고리즘

제산 알고리즘은 나머지 부분과 제수의 값을 판단하여 비복원 제산 방법과 네 개의 지수를 기초로 한다. 비복원 제산은 시프트, 감산, 그리고 비교 동작에 의해 실현된 제산 방법으로 널리 사용되는 알고리즘이다. 4개의 지수 숫자  $q_j$ 는  $\{-1, -0, +0, +1\}$ 을 갖는다. 지수 선택 숫자  $q_j$ 를 기초로 한 제산의 연산은 다음 관계식을 만족한다.

$$R_{j+1} = rR_j - q_j D, \quad j=0, 1, 2, \dots, n-1$$

여기서,

$$\begin{aligned} R_j &= j\text{번째 지수 숫자의 선택 후 나머지 부분} \\ R_0 &= \text{피제수 (강제조건 } |R_0| < |D| \text{)} \\ r &= \text{기수 (이진수 = 2)} \\ q_j &\in \{-1, 0, 1\}, j\text{번째 이진 지수 부호 숫자(BSD)} \\ D &= \text{제수} \end{aligned}$$

이고,  $j$ 가 0일 때,  $n$  개 비트의 피제수와 제수를 갖는 지수  $q_0$ 과 나머지  $R_1$ 을 구하면 다음과 같은 초기치를 갖는다.

$$\begin{aligned} q_0 &= 1 \\ R_1 &= R_0 - D \end{aligned}$$

그리고, 피제수  $R_0$ 은 다음과 같다.

$$\begin{aligned} R_0(\text{피제수}) &= R_0^1 R_0^0 \cdot R_0^1 R_0^0 \cdot R_0^1 R_0^0 \cdots R_0^{n-1} \\ \text{여기서 } R_0^0 &= 0 \end{aligned}$$

비복원 제산에서 다음 관계식에 의해  $n-1$ 번을 반복적으로 지수 선택하여 나머지 부분에 대한 결과를 계산한다. 여기서 지수  $q_j$ 는 시프트된  $j$ 번째의 나머지 부분  $R_j$  여분 이진 형태에 대한 세 개 주요 숫자인  $(R_j^1 R_j^0, R_j^1)$ 의 크기 범위에 따라 네 가지 값을 갖는다.

$$q_j = \begin{cases} 1 \\ 0 \\ -0 \\ -1 \end{cases}, \quad R_{j+1} = \begin{cases} 2(R_j - D) \\ 2(R_j + 0) \\ 2(R_j + 0) \\ 2(R_j + D) \end{cases}$$

$$\text{if } \begin{cases} 1 \leq (R_j^1 R_j^0, R_j^1)_{BSD} \\ 0 \leq (R_j^1 R_j^0, R_j^1)_{BSD} < 1 \\ -1 < (R_j^1 R_j^0, R_j^1)_{BSD} < 1 \\ (R_j^1 R_j^0, R_j^1)_{BSD} \leq -1 \end{cases}$$

여기서

$$\begin{aligned} 1 &\leq j \leq n-1 \\ 0 &\equiv 00.00 \cdots 0 \\ 0\bar{0} &\equiv 11.11 \cdots 1 \text{ 초기 올림수 입력} = 1(2\text{의 보수}) \end{aligned}$$

이고, 그리고 나머지 부분의 연산 속도를 높이기 위하여,  $R_{j+1}$  나머지 부분에 대한 갱신을 이진 부호 숫자를 갖는 여분 이진수로 표현한다. 반복 형태의 연산 방식은 이진수에 대한 여분 이진수의 가산이다. 지수의  $j$ 번째 최상위 비트는 만약  $R_j$ 의 부호가 제수의 부호와 일치하면 1이고, 그리고 만약 일치하지 않는 부호이면 0이다.

### 2. 스퀘어-루트 알고리즘

스퀘어-루트 알고리즘은 제산과 스퀘어-루트 반복 연산에 대한 지수 숫자  $q_j = \{-1, -0, 0, 1\}$ 을 동일하게 사용한다. 여기서 나머지 부분이 복원되지 않는 비복원 알고리즘으로 두 개의 비트를 한 개의 비트로 전환한다. 이전의 나머지 부분이 복원되지 않는 이와 같은 알고리즘을 비복원 스퀘어-루트 알고리즘<sup>[2]</sup>이라 한다. 만약 마지막 반복에서 나머지 부분이 음(-)이 아니면, 최종 나머지가 된다. 만약 그렇지 않으면, 나머지를 가산하여 정확한 최종 나머지를 얻을 수 있다. 지수 초기치  $q_0$ 가 1일 때, 초기치  $Q_0$ 는 다음과 같다.

$$\begin{aligned} q_0 &= 1, \quad Q_0 = 01, \quad R_1 = R_0 - 1, \\ \text{여기서 } R_0 &= \text{스퀘어-루트의 연산자} \end{aligned}$$

스퀘어-루트 반복 단계에서도 제산 방법과 동일하게 시프트된  $j$ 번째의 나머지 부분  $R_j$  여분 이진 형태에 대한 세 개 주요 숫자인  $(R_j^1 R_j^0, R_j^1)_{BCD}$ 의 크기 범위에 따라 지수  $q_j$ 의 네 가지 값의 변화에 의해 나머지 갱신 부분  $R_{j+1}$ 의 연산 값이 결정된다.

$$q_j = \begin{cases} 1 \\ 0 \\ -0 \\ -1 \end{cases}, \quad R_{j+1} = \begin{cases} 2(R_j - (Q_j + 2^{-j-2})) \\ 2(R_j + 0) \\ 2(R_j + 0) \\ 2(R_j + (Q_j - 2^{-j-2})) \end{cases}$$

$$\text{if } \begin{cases} 1 \leq (R_j^1 R_j^0, R_j^1)_{BSD} \\ 0 \leq (R_j^1 R_j^0, R_j^1)_{BSD} < 1 \\ -1 < (R_j^1 R_j^0, R_j^1)_{BSD} < 1 \\ (R_j^1 R_j^0, R_j^1)_{BSD} \leq -1 \end{cases}$$

## 고속 여분 부동 소수점 이진수의 제산/스퀘어-루트 설계 및 제작

결국, 여기서 사용된 스퀘어-루트 알고리즘은 음(-)일 때도 다음 반복 연산에 사용되어 각각의 나머지 부분을 만든다. 그래서 복원 알고리즘을 실행한 이전의 나머지를 사용하기 때문에 복원되지 않는다. 이와 같은 점에서 비복원 스퀘어-루트 알고리즘은 비복원 제산 알고리즘과 매우 동일한 형태를 갖는다.

### 3. 지수 선택 논리 회로

지수 숫자는 각 단계에서  $(R_i^j, R_i^0, R_i^1)_{BCD}$ 를 바탕으로 두고 있다. 네 개의 가능한 지수 값  $q_j = \{-1, -0, 0, 1\}$ 을 선택한다. 그래서  $(R_i^j, R_i^0, R_i^1)_{BCD}$ 는 a(1), b(0), c(-0), d(-1)의 네 개의 경우로 디코드된다. 이것은 표 1에 의해서 다음과 일치하는 논리 값  $S_2, S_1, S_0$ 을 갖는다.

표 1. 지수 선택  $q_j$ 의 진리표

$S_2$	$S_1$	$S_0$	$q_j$	value
0	0	0	b	0
0	0	1	a	1
0	1	0	a	1
0	1	1	a	1
1	0	0	c	-0
1	0	1	d	-1
1	1	0	d	-1
1	1	1	d	-1

제산과 스퀘어-루트의 지수 선택 논리는  $(R_i^j, R_i^0, R_i^1)_{BCD}$  값을 입력으로 하여 그 합을 그림 1의 조합 논리 회로에서 구한다. 여분 이진 표현의 주요 숫자에 대한 합은 디코더에 입력되어 네 개의 가능한 지수 선택 값 a, b, c, d가 디코드된다.

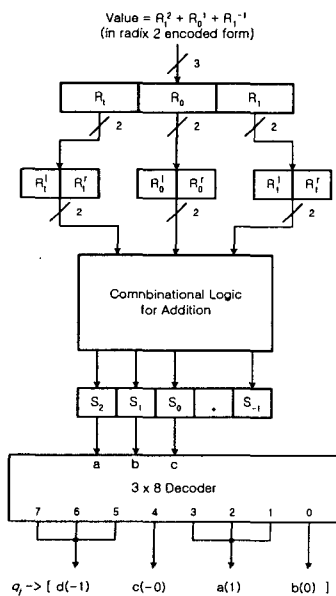


그림 1. 지수 선택 구성도

디코더의 출력  $q_j$ 의 지수 선택에서 알 수 있듯이  $S_n = (R_i)^2 + (R_0)^1 + (R_1)^{-1}$  가산에 대한 논리식  $S_2, S_1, S_0$ 의 이진 값에 의해 지수 선택 논리는  $(R_i^j, R_i^0, R_i^1)_{BCD}$  값을 네 개의 가능한 지수 값 a, b, c, d를 3x8 디코더를 사용하여 그림 1과 같이 선택할 수 있다.

### 4. 제산 및 스퀘어-루트 $R_{j+1}$ 연산의 선택

제산에서, 모든 제수에 대하여 4x1 멀티플렉서를 연결하고 제어 입력에 지수 선택  $q_j$ 를 입력하면 지수 값에 의해 제수의 여분 이진 값이 선택된다. 표 2는 여분 제산  $R_{j+1}$ 의 선택에 대한 네 가지 연산을 나타내고 있다.

표 2. 여분 이진수 제산  $R_{j+1}$  선택에 대한 연산

Quotient selection				$R_{j+1}$ selection		Operation	Selected D
a	b	c	d	A	B		
1	0	0	0	0	0	$2(R_i - D)$	D
0	1	0	0	0	1	$2(R_i + 0_i)$	1
0	0	1	0	1	0	$2(R_i + 0)$	0
0	0	0	1	1	1	$2(R_i + D)$	-D

위와 동일한 방법으로 스퀘어-루트  $R_{j+1}$ 의 네 가지 연산을 수행하기 위해 지수 선택  $q_j$ 를 사용하여 스퀘어-루트의 여분 이진수  $Q_j$ 를 선택한다. 선택된 지수의 연산은 표 3과 같다.

표 3. 여분 이진수 스퀘어-루트  $R_{j+1}$  선택에 대한 연산

Quotient select				$R_{j+1}$ select		Operation	Selected $Q_j$
a	b	c	d	A	B		
1	0	0	0	0	0	$2(R_i - (Q_i + 2^{j-2}))$	$(Q_i + 2^{j-2})$
0	1	0	0	0	1	$2(R_i + 0_i)$	1
0	0	1	0	1	0	$2(R_i + 0)$	0
0	0	0	1	1	1	$2(R_i + (Q_i - 2^{j-2}))$	$-(Q_i - 2^{j-2})$

제산 및 스퀘어-루트  $R_{j+1}$  연산의 선택<sup>[3]</sup> 회로를 나타내면 그림 2와 같다. 이 회로는 n 개의 제수와 스퀘어-루트 진행 변환에 대한 각각의 출력에 연결되어 지수 선택 a, b, c, d 값에 의하여 여분 이진수 D 또는  $Q_j$ 를 선택한다. 선택된 여분 이진수는 여분 이진 셀로 보내어 가산을 처리한다.

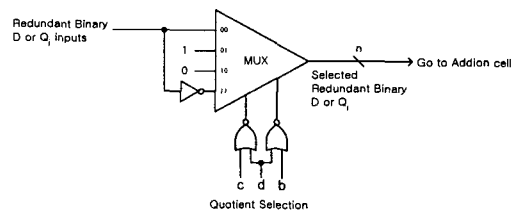


그림 2. 제산/스퀘어-루트  $R_{j+1}$  연산의 선택 회로

5. 제산/스퀘어-루트의 전체 구조

지금까지 기술된 제산과 스퀘어-루트 알고리즘을 그림 3과 같이 규격화된 셀로 구현할 수 있다. 그림과 같이 전체 회로는 여분 가산 배열과 지수 선택 논리 그리고 진행 변환의 세 가지 중요한 요소로 구성된다.

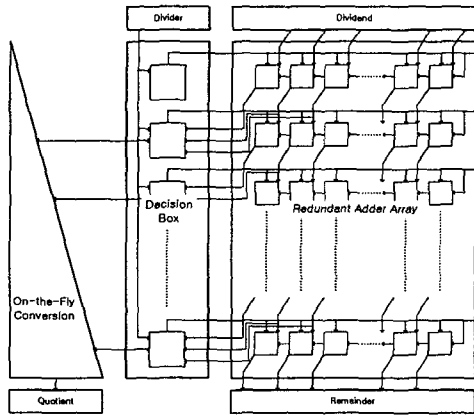


그림 3. 제산/스퀘어-루트의 전체 구조

구성된다. 이 회로에 대한 성능 결과는 제산의 연산 속도가 41ns이고, 스퀘어-루트의 연산 속도는 43ns를 갖는다. 이것은 기존에 있는 동일한 제산과 스퀘어-루트 알고리즘을 사용한 설계보다 약 13% 회로 속도가 향상된 것이다. 그림 5는 제산 및 스퀘어-루트 모델<sup>[4][5]</sup>의 연산 속도를 비교한 것이다. 비교 모델이 동일한 비트 크기로 비교되지 않았지만, 본 논문이 제안한 제산 및 스퀘어-루트 연산은 올림수 지연을 제거함으로써 비트 크기에 상관없이 항상 일정한 지연 시간을 갖기 때문에 연산 속도가 다른 모델에 비해 성능이 우수하다는 것을 알 수 있다.

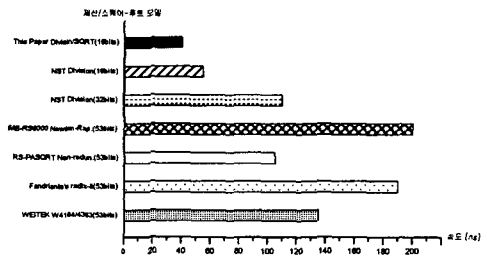


그림 5. 연산 속도 비교

III. 결론 및 평가

본 논문에서 제안한 고속 여분 부동소수점 이진수의 제산/스퀘어-루트 회로는 0.6 $\mu$ m 공정, 전원 전압 3.3V를 사용하여 설계하였으며, 지수 선택 가산 회로에 대한 SPICE 시뮬레이션 결과를 그림 4에 나타내었다. 이 그림에서 알 수 있듯이, 지수 선택 가산 회로의 전달지연시간은 2ns를 갖는다. 시뮬레이션 결과에서 신호①은 지수 선택 회로에 대한 여분 이진 형태의 세 개의 주요 숫자 중 첫 번째 왼쪽 숫자인  $R'_1$  입력이고, 신호②는 지수 선택에 대한 가산 값  $S_2$ 의 출력이다.

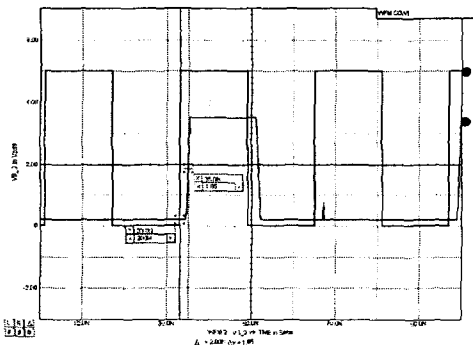


그림 4. 지수 선택 가산 회로의 시뮬레이션 결과

본 논문의 제산/스퀘어-루트는 16-비트 VLSI 회로로 설계하였으며, 총 트랜지스터 수는 약 17,000개로

참고 문헌

- [1] Milos D. Ercegovac and Tomas Lang, "On-the-Fly Conversion of Redundant into Conventional Representations", *IEEE Trans. on Computers*, Vol. C-36, No. 7, pp. 895-897, July 1987.
- [2] Y. Li and W. Chu, "A New Non-Restoring Square Root Algorithm and Its VLSI Implementations", *Proceeding of 1996 IEEE International Conference on Computer Design: VLSI in Computer and Processor*, Austin, TX, pp. 538-544, October 1996.
- [3] H. R. Srinivas, K. K. Parhi and L. A. Montalvo, "Radix-2 Division with Over-Redundant Quotient Selection", *IEEE Transactions on Computer*, vol. 46, No. 1, January 1997.
- [4] Y. Li and W. Chu, "A Parallel-Array Implementations of A Non-Restoring Square Root Algorithm", *Proceeding of 1997 IEEE International Conference on Computer Design*: pp. 690-695, 1997.
- [5] L. A. Montalvo, K. K. Parhi and A. Guyot, "New Svoboda-Tung Division", *IEEE Transactions on Computer*, vol. 47, No. 9, September 1998.