

論文97-34C-2-4

Redundant Binary 연산을 이용한 고속 복소수 승산기

(A high-speed complex multiplier based on redundant binary arithmetic)

辛 卿 旭 *

(Kyung-Wook Shin)

요 약

본 논문에서는 Redundant binary (RB) 연산을 이용한 새로운 복소수 승산 알고리즘과 병렬 아키텍처를 제안하고, 이를 이용한 8-비트 복소수 승산기를 설계하였다. RB 연산을 적용함으로써 N-비트 복소수 승산이 실수부, 허수부를 담당하는 두 개의 RB승산(즉, N개의 RB 부분곱의 합)으로 간소화되었으며, 이를 통하여 기존의 실수 승산기를 이용한 방법 보다 고속 승산이 가능하도록 하였다. 또한, 효율적인 RB 인코딩을 제안함으로써 이진 부분곱 생성과 비교하여 추가적인 하드웨어 및 지연시간의 부담 없이 RB 부분곱이 생성되도록 하였다. 제안된 복소수 승산방법은 규칙성과 모듈성을 갖는 병렬구조로 구현되며, 따라서 기존의 방법 보다 구현이 용이하고 고속 승산이 가능하다. 제안된 알고리즘의 평가를 위해 prototype 8-비트 복소수 승산기를 0.8 μm CMOS 공정으로 제작하였다. 제작된 복소수 승산기 코어는 11,500개의 트랜지스터로 구성되며, 레이아웃 면적은 약 $1.05 \times 1.34 \text{ mm}^2$ 이다. 제작된 칩의 특성평가로부터, 전원전압 2.5 V에서 200 MHz의 클럭 주파수로 안전하게 동작함을 확인하였으며, 이때의 전력소모는 약 90 mW로 측정되었다.

Abstract

A new algorithm and parallel architecture for high-speed complex number multiplication is presented, and a prototype chip based on the proposed approach is designed. By employing redundant binary (RB) arithmetic, an N-bit complex number multiplication is simplified to two RB multiplications (i. e., an addition of N RB partial products), which are responsible for real and imaginary parts, respectively. Also, an efficient RB encoding scheme proposed in this paper enables to generate RB partial products without additional hardware and delay overheads compared with binary partial product generation. The proposed approach leads to a highly parallel architecture with regularity and modularity. As a result, it results in much simpler realization and higher performance than the classical method based on real multipliers and adders. As a test vehicle, a prototype 8-bit complex number multiplier core has been fabricated using 0.8 μm CMOS technology. It contains 11,500 transistors on the area of about $1.05 \times 1.34 \text{ mm}^2$. The functional and speed test results show that it can safely operate with 200 MHz clock at $V_{DD}=2.5 \text{ V}$, and consumes about 90 mW.

I. 서 론

디지털 신호처리 및 통신의 응용분야가 확대되면서 복소수 데이터 처리의 필요성이 증대되고, 따라서 복소수 승산기 및 승산-누적기 등 복소수 연산장치들이 이들 시스템의 핵심 구성요소가 되고 있다. 예를 들어, 고속 푸리에 변환 (Fast Fourier Transform; FFT),

* 正會員, 金鳥工科大学校 電子工學科

(Department of Electronic Engineering, Kumoh National University of Technology)

※ 본 연구는 1995년 전반기 한국과학재단 해외 Post Doc. 연수지원에 의한 연구결과의 일부임.

接受日字:1996年10月15日, 수정완료일:1997年2月3日

레이더 신호처리, Quadrature 변조 시스템의 기저대역 신호처리 등은 근본적으로 복소수 연산을 필요로 한다. 일반적으로, 실지산 신호처리 및 통신 시스템에서 연산 회로의 성능은 전체 시스템의 성능에 중요한 영향을 미치므로, 경제적인 고성능 연산장치의 설계가 중요 관심사항으로 대두되고 있다.

복소수 승산기를 구현하는 가장 기본적인 방법은 4개의 실수 승산기와 2개의 실수 가산기를 사용하는 것이다^[1,2]. 이와 같은 고전적인 방법에서는 실수 승산기의 성능이 복소수 승산기의 성능을 결정하는 요소가 되므로, 지금까지 대부분의 승산기 설계에 관한 연구가 실수 승산기의 성능 향상에 초점이 맞추어져 왔으며, 복소수 승산기 설계에 대한 관심이 적었다.

복소수 승산의 연산량을 줄이기 위한 고전적인 방법으로는 식(1)과 같은 변환식을 이용하는 것이다^[3]. 이 방법은 1개의 실수 승산기를 줄이는 대가로 3개의 실수 가산기가 추가로 필요하므로 하드웨어 구현의 측면에서 크게 바람직한 방법이라 할 수 없다.

$$(A_R + jA_I)(B_R + jB_I) = (A_R - A_I)B_I + A_R(B_R - B_I) + j[(A_R - A_I)B_I + A_I(B_R + B_I)] \quad (1)$$

또 다른 방법으로는 2차원 벡터의 회전을 계산하기 위한 CORDIC 알고리즘^[4]을 이용할 수 있다. 그러나, CORDIC 알고리즘의 하드웨어 구현을 위해서는 의사 scaling factor 값을 결정하는 문제와 수렴영역의 제한성, bit-recursive 연산특성 등에 의해 하드웨어의 병렬구현 측면에서 바람직하지 않은 것으로 평가되고 있다^[5]. 최근에 off-set 이진표현과 분산연산을 이용한 직렬형 복소수 승산기 구조^[6]가 제안되었으나, latency가 매우 길다는 단점이 있다. 한편, quadratic residue number system (QRNS)을 이용하면 복소수 승산이 2개의 실수 승산에 의해 구현 가능하지만^[7], 2진 수치계를 QRNS로 변환하기 위한 전처리와 QRNS를 2진 수치계로 역변환하기 위한 후처리 과정에 많은 양의 부가 하드웨어가 필요하다는 것이 문제점이다.

본 논문에서는 고성능 복소수 승산기의 구현을 목적으로 종래의 구현방법과 다른 새로운 복소수 승산방법을 연구하고 이에 적합한 효율적인 VLSI 구조를 제안하고자 한다. 이를 위해 기존의 2진 수치계 대신에 redundant binary (RB) 수치계를 이용함으로써 복소

수 승산을 RB 부분곱의 합으로 단순화시킬 수 있는 새로운 복소수 승산 알고리즘을 개발하였다. 본 논문에서 채택한 RB 수치계는 각 digit가 $\{-1, 0, 1\}$ 의 값을 갖는 radix-2 signed-digit (SD) 수치계^[8]에 속한다. 문헌에 잘 알려진 바와 같이, RB 수치계에 의한 연산은 carry-propagation-free (CPF) 가산 특성을 가지므로 고속 연산을 실현할 수 있다.^[9] 이와 같은 RB 연산의 CPF 특성을 이용하면 승산기 구현시 캐리전파 없이 부분곱을 빠르게 가산할 수 있으므로 이를 실수 승산기에 적용한 연구결과 등이 발표되고 있다.^[10-13]

본 논문의 2장에서는 RB 수치계를 이용한 새로운 복소수 승산 알고리즘과 RB 부분곱을 효율적으로 생성하기 위한 방법을 제안한다. 3장에서는 2장에서 제안된 알고리즘을 적용한 승산기 아키텍처 및 회로설계, 그리고 CMOS 공정을 이용한 칩 제작과 제작된 칩의 성능평가 등에 대해 언급한다. 마지막으로 4장에서는 결론과 고찰을 기술한다.

II. RB 수치계를 이용한 새로운 복소수 승산 알고리즘

1. RB 수치계

RB 표현은 digit set $\{-1, 0, 1\}$ 을 갖는 radix-2 SD 수치계^[8]에 속하며, 일반적으로 SD $d_i \in \{-1, 0, 1\}$ 는 두개의 이진 비트 (x_i, y_i) (단, $x_i, y_i \in \{0, 1\}$)로 인코딩된다. 식(2)와 같이 인코딩을 정의하는 경우, (x_i, y_i) 로 인코딩된 SD d_i 는 $x_i - \overline{y_i}$ 의 값을 갖는다.

$$d_i \equiv (x_i, y_i) \equiv x_i - \overline{y_i} \quad (2)$$

단, $\overline{y_i}$ 는 y_i 의 반전을 나타냄.

식(2)를 확장하면, N -digit RB 데이터 D 와 이를 인코딩하는 N -비트 이진수 X, Y 와의 관계를 식(3)과 같이 정의할 수 있다.

$$D \equiv (X, Y) \equiv X - \overline{Y} \quad (3)$$

$$\text{단, } D = [d_{N-1} \ d_{N-2} \ \cdots \ d_1 \ d_0]$$

$$X = [x_{N-1} \ x_{N-2} \ \cdots \ x_1 \ x_0]$$

$$Y = [y_{N-1} \ y_{N-2} \ \cdots \ y_1 \ y_0]$$

여기서 \overline{Y} 는 Y 의 모든 비트의 반전을 의미한다.

X와 Y가 2의 보수라면, 식(3)으로부터 다음의 유용한 관계를 얻을 수 있다.

$$X - (-Y) = X - (\overline{Y} + 1) = D - 1 \quad (4)$$

식(4)로부터, 2의 보수 X와 Y의 합 X+Y는 하나의 RB 수 D로 변환될 수 있으며, 역으로 RB 수 D는 2의 보수 X와 Y의 합으로 변환될 수 있음을 알 수 있다. 이와 같은 RB 수와 2의 보수 사이의 관계를 기초로 다음절에서는 RB 수치계를 이용한 새로운 복소수 곱셈 알고리즘을 제안한다.

2. RB 연산을 이용한 복소수 승산 알고리즘

실수부와 허수부가 각각 식(5)와 같이 2의 보수형태로 표현되는 두개의 복소수 A와 B, 그리고 그들의 곱 Z에 대해 생각한다.

$$A = A_R + jA_I = (-a_{N-1}2^{N-1} + \sum_{i=0}^{N-2} a_i 2^i) + j(-b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i) \quad (5.a)$$

$$B = B_R + jB_I = (-c_{N-1}2^{N-1} + \sum_{i=0}^{N-2} c_i 2^i) + j(-d_{N-1}2^{N-1} + \sum_{i=0}^{N-2} d_i 2^i) \quad (5.b)$$

$$Z = (A_R + jA_I)(B_R + jB_I) = Z_R + jZ_I \quad (6)$$

식(5)를 식(6)에 대입한 후 2의 보수에 관한 성질을 이용하여 정리하면, 복소수 곱의 실수부 Z_R와 허수부 Z_I는 각각 식(7)과 같이 된다.

$$Z_R = A_R B_R - A_I B_I = (p_{N-1, N-1} - q_{N-1, N-1})2^{2N-2} + \sum_{i=0}^{N-2} \sum_{j=0}^{N-2} (p_{i, j} - q_{i, j}) 2^{i+j} + 2^{N-1} \sum_{i=0}^{N-2} (\overline{p_{i, N-1}} - \overline{q_{i, N-1}}) 2^i + 2^{N-1} \sum_{j=0}^{N-2} (\overline{p_{N-1, j}} - \overline{q_{N-1, j}}) 2^j \quad (7.a)$$

$$Z_I = A_R B_I + A_I B_R = (u_{N-1, N-1} - \overline{v_{N-1, N-1}})2^{2N-2} + \sum_{i=0}^{N-2} \sum_{j=0}^{N-2} (u_{i, j} - \overline{v_{i, j}}) 2^{i+j} + 2^{N-1} \sum_{i=0}^{N-2} (\overline{u_{i, N-1}} - v_{i, N-1}) 2^i + 2^{N-1} \sum_{j=0}^{N-2} (\overline{u_{N-1, j}} - v_{N-1, j}) 2^j + 1 \quad (7.b)$$

식(7)에서, $p_{i,j} = a_i c_j$, $q_{i,j} = b_i d_j$, $u_{i,j} = a_i d_j$ 그리고 $v_{i,j} = b_i c_j$ 는 비트 레벨 부분곱을 나타낸다.

식(7.a)와 (7.b)를 RB 수치계로 변환하기 위하여, RB 수 $a_{i,j}$ 와 $\beta_{i,j}$ (단, $a_{i,j}, \beta_{i,j} \in \{-1, 0, 1\}$), 그리고 그들의 부호 반전 $a_{i,j}^*$, $\beta_{i,j}^*$ (즉, $a_{i,j}=1$ 이면 $a_{i,j}^*=-1$)을 식(8)과 같이 정의한다.

$$a_{i,j} = p_{i,j} - q_{i,j} \quad (8.a)$$

$$a_{i,j}^* = \overline{p_{i,j}} - \overline{q_{i,j}} \quad (8.b)$$

$$\beta_{i,j} = u_{i,j} - \overline{v_{i,j}} \quad (8.c)$$

$$\beta_{i,j}^* = \overline{u_{i,j}} - v_{i,j} \quad (8.d)$$

식(8)을 식(7)에 대입하여 정리하면, 식(9)와 같이 RB 수치계로 변환된 부분곱 (RB partial products ; RB_PP) 을 얻을 수 있다.

$$P_R(i) = a_{i, N-1}^* 2^{N-1+i} + \sum_{j=0}^{N-2} a_{i, j} 2^{i+j} \quad (\text{단, } 0 \leq i \leq N-2) \quad (9.a)$$

$$P_R(N-1) = a_{N-1, N-1} 2^{2N-2} + \sum_{j=0}^{N-2} a_{N-1, j}^* 2^{N-1+j} \quad (\text{단, } i = N-1) \quad (9.b)$$

$$P_I(i) = \beta_{i, N-1}^* 2^{N-1+i} + \sum_{j=0}^{N-2} \beta_{i, j} 2^{i+j} \quad (\text{단, } 0 \leq i \leq N-2) \quad (9.c)$$

$$P_I(N-1) = \beta_{N-1, N-1} 2^{2N-2} + \sum_{j=0}^{N-2} \beta_{N-1, j}^* 2^{N-1+j} \quad (\text{단, } i = N-1) \quad (9.d)$$

마지막으로 식(9)를 정리하면, 복소수 승산의 실수부와 허수부는 각각 식(10)과 같이 새롭게 정의될 수 있다.

$$Z_R = \sum_{i=0}^{N-1} P_R(i) \quad (10.a)$$

$$Z_I = \sum_{i=0}^{N-1} P_I(i) + 1 \quad (10.b)$$

결론적으로 식(10)이 의미하는 바는 다음과 같다. 첫째, N-비트 복소수 승산을 RB 수치계로 표현함으로써 실수부 Z_R와 허수부 Z_I가 각각 N개의 RB_PP의 합으로 변환되었으며, 둘째, 식(10)은 한 짝의 RB_PP가 산회로로 구현될 수 있으므로 전체적인 승산기의 구조가 단순화되며, 셋째, RB 연산의 특징인 CPF 가산에 의해 RB_PP의 합을 고속으로 실현할 수 있다. 또한, 식(9)에는 부호 비트에 대한 정보가 고려되었으므로 RB_PP의 가산과정에서 부호 확장을 고려할 필요가 없다는 장점도 갖는다.

3. RB 부분곱 생성

2절에서 유도된 RB 부분곱 (RB_PP) 의 각 digit는

$\{-1, 0, 1\}$ 중의 한 값을 가지므로 이들은 2 비트의 이진코드로 인코딩된다. 본 절에서는 효율적인 부분곱 생성을 위해 식(8)에 정의된 $a_{i,j}$, $\beta_{i,j}$ 를 식(11)과 같이 인코딩한다.

$$a_{i,j} \equiv p_{i,j} - q_{i,j} \equiv (s_{i,j}^-, \overline{s_{i,j}^+}) \quad (11.a)$$

$$a_{i,j}^* \equiv \overline{p_{i,j}} - q_{i,j} \equiv (s_{i,j}^-, \overline{s_{i,j}^+}) \quad (11.b)$$

$$\beta_{i,j} \equiv u_{i,j} - v_{i,j} \equiv (t_{i,j}^-, \overline{t_{i,j}^+}) \quad (11.c)$$

$$\beta_{i,j}^* \equiv \overline{u_{i,j}} - v_{i,j} \equiv (t_{i,j}^-, \overline{t_{i,j}^+}) \quad (11.d)$$

단, $s_{i,j}^-, s_{i,j}^+, t_{i,j}^-, t_{i,j}^+ \in \{0, 1\}$

식(11)로부터, RB 부분곱의 인코딩을 위한 이진 비트 $s_{i,j}^-, s_{i,j}^+, t_{i,j}^-, t_{i,j}^+$ 와 비트 레벨 부분곱 $p_{i,j}, q_{i,j}, u_{i,j}, v_{i,j}$ 사이의 상호관계를 표 1과 같이 유도할 수 있다. 표 1은 2진 수치계로부터 RB 수치계로의 변환 즉, 식(7)의 2진 부분곱을 식(9)의 RB 부분곱으로 변환하기 위한 관계를 나타내며, 단순한 2-입력 NAND 및 AND 게이트를 사용하여 RB 부분곱이 생성될 수 있음을 알 수 있다. 따라서 기존의 2진 수치계를 이용한 승산방식과 비교할 때, 부가적인 하드웨어나 지연시간 없이 효율적인 RB 부분곱의 생성이 가능하다.

표 1. RB 코드 비트 생성

Table 1. Generation of RB code bit.

$(0 \leq i, j \leq N-2)$ or $(i = j = N-1)$	$(i = N-1, 0 \leq j \leq N-2)$ or $(j = N-1, 0 \leq i \leq N-2)$
$s_{i,j}^- = p_{i,j} = a_i c_j$	$s_{i,j}^- = \overline{p_{i,j}} = \overline{a_i c_j}$
$s_{i,j}^+ = \overline{q_{i,j}} = \overline{b_i d_j}$	$s_{i,j}^+ = q_{i,j} = b_i d_j$
$t_{i,j}^- = u_{i,j} = a_i d_j$	$t_{i,j}^- = \overline{u_{i,j}} = \overline{a_i d_j}$
$t_{i,j}^+ = v_{i,j} = b_i c_j$	$t_{i,j}^+ = \overline{v_{i,j}} = \overline{b_i c_j}$

III. 8-비트 복소수 승산기 설계, 제작 및 성능평가

II장에서 제안된 알고리즘을 적용한 복소수 승산기의 prototype을 설계하였다. 본 논문에서는 성능평가를 위해 편의상 8-비트 승산기를 설계하였으며, 전체 승산기의 구조가 매우 규칙적이므로 승산기 비트 수를 쉽게 확장할 수 있다. 1절에서는 전체적인 승산기 구조에 대해 기술하고, 2절과 3절에서는 RB 가산기 및 RB-binary 변환블록의 회로설계를 언급한다. 4절에서는 8-비트 복소수 승산기 코어의 레이아웃 설계를 언

급하고, 마지막으로 5절에서는 제작된 칩의 테스트 및 성능평가를 설명한다.

1. 아키텍처 개요

2장에서 제안된 알고리즘에 의하면, N -비트 복소수 승산의 실수부와 허수부가 각각 N 개의 RB_PP의 합으로 표현되었다. 본 논문에서는 이를 효율적으로 구현하기 위해 그림 1과 같은 승산기 구조를 채택하였다. 전체적인 승산기 구조는 서로 대칭인 한 짝 (pair) 의 가산기 트리 (tree) 로 구성되며, 왼쪽의 가산기 트리는 복소수 승산의 실수부를 담당하고, 오른쪽의 트리는 허수부를 담당한다. 가산기 트리는 $\lceil \log_2 N \rceil$ (단, N 은 피승수의 비트 수) stage의 RB 부분곱 가산 블록 (RBA) 으로 구성되며, RBA는 RB 전가산기 (Redundant Binary Full Adder ; RBFA) 셀의 1차원 배열로 구성된다. 한편, 가산기 트리의 첫 번째 stage에 위치한 RBA들은 표 1에서 정의된 RB부분곱 생성 회로 (즉, 2-입력 AND 및 NAND 게이트) 를 포함한다. 승산기 출력단의 RB2BC (RB-to-Binary Conversion) 블록은 RB 수치계를 2의 보수 수치계로 변환하여 최종 승산결과가 2의 보수형태로 외부에 출력되도록 한다. 전체 승산기는 가산기 트리의 stage 단위로 파이프라인 동작하며, 파이프라인 클럭 주기는 단일 RBFA 셀의 지연시간에 의해 결정된다. 본 논문의 복소수 승산기는 동일한 RBFA 셀의 규칙적인 배열로 구성되므로 설계가 용이하고 확장성이 뛰어나며, 병렬/파이프라인 방식으로 동작하여 throughput이 높다는 특징을 갖는다.

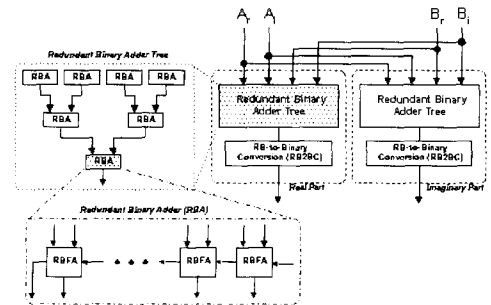


그림 1. 제안된 복소수 승산기의 구조

Fig. 1. Architecture of the proposed complex number multiplier.

2. RB 가산기 회로

본 절에서는 복소수 승산기를 구성하는 기본 셀인

RBFA 셀을 설계한다. 설계되는 RBFA는 RB 부분곱 가산시에 최하위 digit에서 최상위 digit로 캐리신호의 전파가 없는 CPF 가산기이며, 따라서 N -digit RB 부분곱의 가산시간은 단위 RBFA 셀의 지연시간에 의해 결정된다.

그림 1에서 보듯이 N -digit RBA는 N 개의 RBFA 셀로 구성되며, i -번째 RBFA 셀의 연산기능은 식(12)와 같이 정의된다.

$$2c_i + s_i = x_i + y_i \quad (12.a)$$

$$z_i = s_i + c_{i-1} \quad (12.b)$$

여기서, SD들은 $x_i = (x_i^-, x_i^+)$, $y_i = (y_i^-, y_i^+)$, $c_{i-1} = (c_{i-1}^-, c_{i-1}^+)$, $z_i = (z_i^-, z_i^+)$, $c_i = (c_i^-, c_i^+)$ 로 인코딩된다. 식(12)의 RBFA는 이진 전가산기와 유사한 정의식을 갖으나, RBFA가 이진 전가산기와 다른 점은 입력 데이터가 이진수가 아닌 signed-digit 표현이며, 가산 과정이 CPF 방식으로 이루어진다는 점이다. i -번째 RBFA 셀은 RB 부분곱의 i -번째 digit인 x_i , y_i 와 $(i-1)$ -번째 digit로부터 전파된 캐리신호 c_{i-1} 를 입력받아 합 digit z_i 와 캐리 digit c_i 를 생성한다. 이때, CPF 가산을 이루기 위해 $(i-1)$ -번째 digit에서 전파된 캐리신호 c_{i-1} 이 i -번째 RBFA 셀을 통해 $(i+1)$ -번째 digit로 전달되지 않도록 RBFA 셀이 정의된다. 이와 같은 CPF 가산특성은 RB 수치계가 갖는 redundancy를 이용하여 쉽게 얻어진다^[9]. 본 논문에서는 Takagi가 제안한 CPF 가산규칙^[9]을 기본으로 하여 RBFA와 RB 반가산기(RB half adder; RBHA)의 논리기능을 표 2와 같이 정의하였다.

표 2. RB 전가산기와 RB 반가산기의 논리식
Table 2. Logic functions of RBFA and RBHA.

RBFA Cell	RBHA Cell
$g_i = (x_i^- \oplus x_i^+) \oplus (y_i^- \oplus y_i^+)$	$g_i = (x_i^- \oplus x_i^+)$
$h_i = x_i^- x_i^+ + y_i^- y_i^+$	$h_i = x_i^- x_i^+$
$c_i^- = (x_i^- + x_i^+) (y_i^- + y_i^+)$	$c_i^- = (x_i^- + x_i^+)$
$c_i^+ = g_i c_{i-1}^- + \bar{g}_i h_i$	$c_i^+ = \bar{g}_i c_{i-1}^- + g_i h_i$
$z_i^- = g_i \oplus c_{i-1}^-$	$z_i^- = \bar{g}_i \oplus c_{i-1}^-$
$z_i^+ = c_{i-1}^+$	$z_i^+ = c_{i-1}^+$

설계된 복소수 승산기는 RBFA 셀의 규칙적인 배열로 구성되므로 기본 셀인 RBFA의 성능(즉, 면적 및 속도)이 전체 승산기의 성능을 결정하는 중요한 요소

가 된다. 본 논문에서는 RBFA 회로의 최적설계를 위해 고전적인 CMOS 회로 대신에 전달게이트(transmission gate) 회로구조를 사용하였다. 그림 2는 표 2에서 정의된 RBFA 셀의 회로도이며, AND-OR-XOR(AOX) 회로를 기본으로 하여 구성된다. 그림 3은 RBFA를 구성하는 AOX-1 단위 셀의 회로도이며, 전달게이트를 사용하여 13개의 트랜지스터로 구현되었다. 설계된 RBFA 셀은 54개의 트랜지스터로 구성되며, 이는 표 3에서 보는 바와 같이, 문헌에 발표된 RBFA 회로들 보다 가장 적은 수의 트랜지스터로 구현되었다.

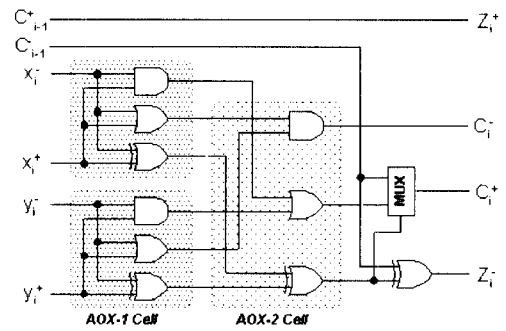


그림 2. RB 전가산기(RBFA)의 회로도
Fig. 2. Circuit diagram of RB full adder (RBFA).

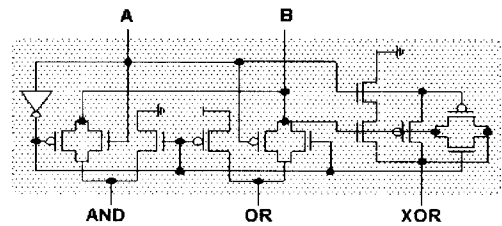


그림 3. AOX-1 셀의 회로도
Fig. 3. Circuit diagram of AOX-1 cell.

표 3. RBFA 셀의 트랜지스터 수 비교
Table 3. Comparison of transistor count of RBFA cell.

Ref. [10,11]	Ref. [12]	Ref. [13]	Ref. [14]	This Paper
62	56	58	98	54

3. RB-Binary 변환(RB2BC) 회로
RB2BC는 RB 수치계로 계산된 복소수 승산결과를

2의 보수 수치계로 변환하는 기능을 수행하는 블록이다. 이와 같은 수치계 변환은 식(4)에 정의된 바와 같이 N-digit RB 데이터를 인코딩하는 두개의 N-bit 데이터 X, Y 의 가산에 의해 이루어진다. 따라서, RB2BC 블록은 2진 가산기 회로로 구성되며, 최하위 비트에서 최상위 비트로 캐리신호의 전파가 발생된다.

본 논문에서는 RB2BC 블록이 1 클럭내에 가산을 완료하여 전체 승산기가 파이프라인 방식으로 동작하도록 설계하였으며, 면적과 속도의 최적화를 위해 그림 4와 같이 ripple-carry adder (RCA) 와 carry-select adder (CSA) 로 구성되는 hybrid 구조를 채용하였다. RB 가산기 트리를 통해 출력되는 승산 결과 중 하위 비트가 상위 비트 보다 먼저 출력되므로, 하위 4-비트는 RB2BC 블록 보다 1클럭 이전에 RCA에서 미리 계산되고, 나머지 상위 14-비트는 RB2BC 블록에서 단일 클럭 내에 가산되도록 설계하였다. 한편, RB2BC 블록이 단일 클럭 내에 가산을 완료할 수 있도록 하기 위해 14-비트중 하위 4-비트는 RCA 구조를 사용하고 나머지 10-비트는 CSA 구조를 사용하여 처리하였다. 이와 같이 hybrid 구조를 채용함으로써 CSA와 같은 병렬 가산기만을 사용하는 경우 보다 회로를 단순화하여 RB2BC 블록이 차지하는 면적을 최소화하였다.

한편, 복소수 승산결과가 직접 외부로 출력되지 않고 가산/감산/비교 등 다른 연산에 계속적으로 사용되는 응용분야에서는 필요한 모든 연산을 RB 수치계로 처리한 후 최종 출력단에 RB2BC 블록을 사용하면 되므로, 승산기 코어의 출력단에 RB2BC 블록을 사용하지 않고 최종 출력단에만 수치계 변환회로를 사용하도록 설계하면 된다.

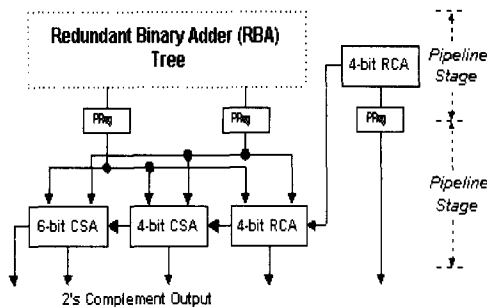


그림 4. RB2BC 회로의 구성도
Fig. 4. Block diagram of RB2BC circuit.

4. 레이아웃 설계
설계된 복소수 승산기는 크게 실수부와 허수부로 구성되며, 이들은 각각 서로 대칭인 이진트리구조를 갖는다. 이진트리는 7개의 부분곱 가산기 블록 (RBA) 이 4-2-1의 3 stage로 구성되며, 고밀도 레이아웃이 가능하도록 7개의 RBA를 수직으로 배열하여 그림 5와 같은 floorplan을 설계하였다. 클럭방식은 TSPC (true single phase clocking) 을 사용하였으며, 파이프라인 레지스터는 11개의 트랜지스터로 구성된다. 클럭신호는 그림 5에 나타난 것과 같이 2단 버퍼를 통해 파이프라인 레지스터에 공급되며, 클럭 skew를 최소화 하기 위해 실수부와 허수부 사이의 중앙부분에서 공급 되도록 하였다. 전체 승산기의 레이아웃은 full custom 방식으로 설계되었으며, 레이아웃의 aspect 비 (즉, 가로 : 세로의 비) 가 거의 1:1이 되도록 RBFA 셀의 aspect 비를 미리 예측하여 레이아웃에 반영하였다. 또한, 배선에 의한 면적소모를 최소화함과 동시에 설계의 효율성을 높이기 위해 RBFA 셀의 외부 연결단자를 표준화하였으며, 이를 통하여 셀들의 규칙적인 배열에 의해 부분곱 가산기 블록이 완성되도록 하였다. RBFA 셀의 레이아웃 면적은 약 $84.4 \times 51.2 \mu\text{m}^2$ 이다.

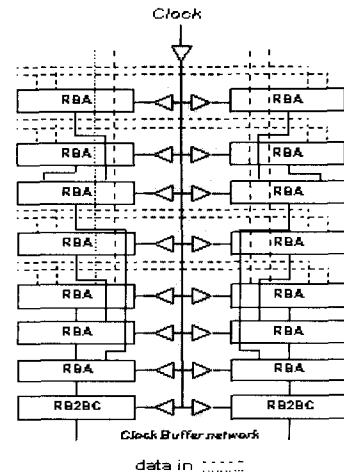


그림 5. 설계된 복소수 승산기의 floorplan
Fig. 5. Floorplan of the complex number multiplier.

5. 칩 제작 및 특성 측정
설계된 회로를 0.8 μm 이중금속배선 CMOS 공정으로 제작하였으며, 제작된 복소수 승산기 코어의 확대사진은 그림 6과 같다. 8-비트 승산기 코어는 11,500여

개의 트랜지스터로 구성되며 약 $1.05 \times 1.34 \text{ mm}^2$ 의 면적을 갖는다.

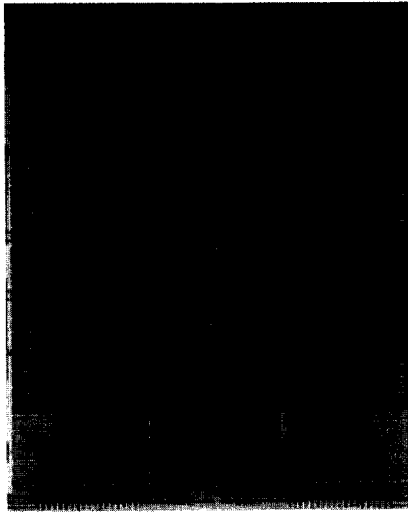
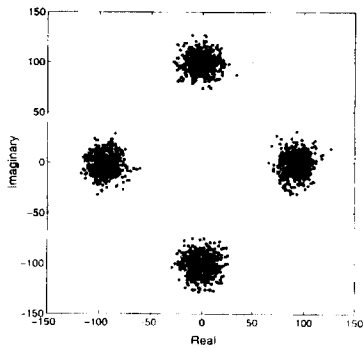
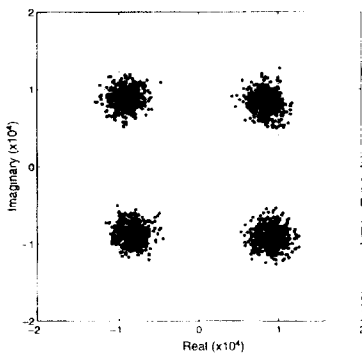


그림 6. 제작된 복소수 승산기 코어의 확대사진
Fig. 6. Microphotograph of the fabricated complex number multiplier.



(a) 테스트 입력 데이터 constellation



(b) 테스트 응답 데이터 constellation

그림 7. 제작된 복소수 승산기의 논리동작 테스트 결과
Fig. 7. Functional test results of the fabricated complex number multiplier.

제작된 승산기 칩은 디지털 I/O 보드와 논리 분석기, 펄스 발생기 등의 장비를 이용하여 논리기능과 동작속도 특성을 평가였다. 그림 7은 제작된 승산기의 승산동작을 테스트한 결과이다. 그림(7-a)는 승산기에 입력으로 인가된 복소수 데이터들의 constellation이며, 그림(7-b)는 칩으로부터 얻어진 승산결과 (즉, 입력 복소수 데이터에 $90 + j90$ 을 곱한 승산결과) 를 나타내는 constellation이다. 2,000개의 복소수 데이터로 구성되는 입력 데이터 constellation이 정확히 45° 만큼 회전되어 출력됨을 볼 수 있으며, 이로부터 설계된 복소수 승산기의 논리기능이 정상적으로 동작함을 확인하였다. 이밖에 여러 가지 테스트 패턴들을 입력시켜 제작된 칩이 정상적으로 동작함을 확인하였다.

그림 8은 제작된 승산기의 동작특성을 측정한 Schmoop plot이며, 전원전압에 따른 최대 동작 클럭 주파수의 관계를 보이고 있다.

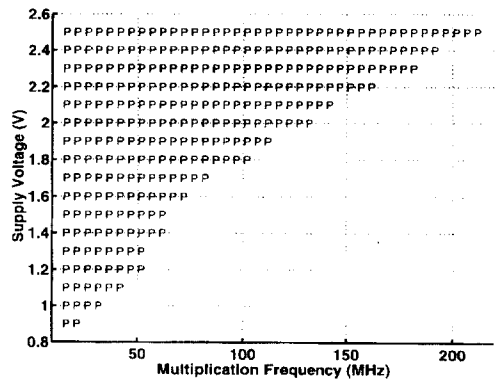


그림 8. 제작된 복소수 승산기의 Schmoop plot
Fig. 8. Schmoop plot of the fabricated complex number multiplier.

표 4. 제작된 8-비트 복소수 승산기의 특성
Table 4. Summary of the fabricated 8-bit complex number multiplier.

Technology	0.8 μm double metal CMOS
Transistor count	11,500
Layout area (core part)	$1.05 \times 1.34 \text{ mm}^2$
Layout density	$8,200 \text{ Trs./mm}^2$
Latency	4 clock cycles
Operating clock frequency	200 MHz @ 2.5 V
Power consumption	90 mW @ 2.5 V, 200 MHz
I/O pin count	77 including 6 VDD/GND Pads

제작된 복소수 승산기는 $V_{dd}=2.5\text{ V}$ 에서 200 MHz의 클럭 주파수로 안전하게 동작함을 알 수 있으며, 이때의 전력소모는 약 90 mW로 측정되었다. 그림 9는 전원전압, 동작속도에 따른 전력소모를 그래프로 나타낸 것이며, 표 4는 제작된 복소수 승산기의 특성을 요약한 것이다.

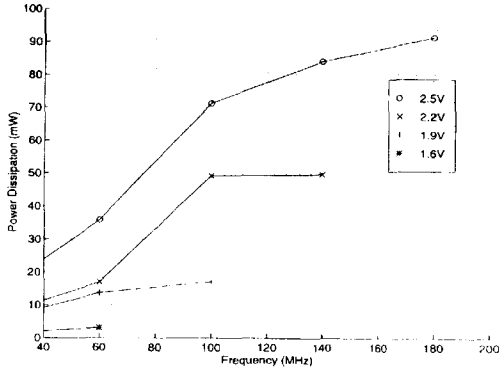


그림 9. 제작된 복소수 승산기의 전력소모 특성
Fig. 9. Power consumption of the fabricated complex number multiplier.

IV. 결론

본 논문에서는 고성능 복소수 승산기 구현을 위한 새로운 승산 알고리즘과 병렬 아키텍처를 제안하였다. 제안된 알고리즘 및 아키텍처를 적용하여 8 비트 복소수 승산기를 설계, 제작하였으며, $V_{dd}=2.5\text{ V}$ 에서 200 MHz의 클럭 주파수로 안전하게 동작함이 확인되었다.

본 논문의 방법은 다음과 같은 특징을 갖는다. 첫째, RB 수치계를 적용함으로써 N-비트 복소수 승산이 실수부와 허수부를 담당하는 두 개의 RB승산 (N개의 RB 부분곱의 합)으로 간소화되었다. 이를 통하여 종래의 2진 수치계를 이용하는 방법 (즉, 실수 승산기를 이용하는 방법) 보다 승산성능의 향상이 얻어지며, 승산기 구조의 단순화를 이루었다. 둘째, 제안된 복소수 승산기는 구조적인 규칙성을 가지므로 설계 및 확장이 용이하고 고밀도 레이아웃이 가능하므로 compact한 승산기 코어의 구현에 적합하다. 셋째, RB 전가산기 회로를 최적화된 전달게이트 방식으로 설계하였으며, 문헌에 발표된 회로들과 비교하여 최소의 트랜지스터로 구현되어 면적 및 동작속도의 최적화를 이루었다. 넷째, 전체 승산기가 병렬/파이프라인 방식으로 동작하

므로 데이터 처리율이 높으며, latency가 $O(\log_2 N)$ 이므로 승산기의 비트 수가 증가해도 latency가 매우 작다. 다섯째, RB 연산의 캐리전파 없는 가산특성에 의해 고속승산이 가능하다. 여섯째, 2진 부분곱과 동일하게 RB 부분곱이 생성되므로, 2진 수치계를 RB 수치계로 변환하기 위한 하드웨어나 지연시간의 추가부담이 없다.

결론적으로, 본 논문에서 제안된 방법은 실수 승산기를 사용하는 종래의 방법에 비해 고집적, 고성능 복소수 승산기의 구현이 가능하며, 따라서 모뎀의 adaptive equalizer, complex filter 등 통신 및 신호처리 시스템의 핵심 연산장치 구현에 적합할 것으로 평가된다.

※ Acknowledgments

The author would like to acknowledge Prof. Song from University of Illinois at Urbana-Champaign for his suggestion and discussion throughout this work. The author is also greatly indebted to Kanti Bacrania from Harris for fabricating the chip and Yunho Kim for chip testing. This work was supported in part by Korea Science and Engineering Foundation (KOSEF).

참고 문헌

- [1] R.B. Joshi, H. Samuelli, "A 100MHz, 5MBaud QAM decision-feedback equalizer for digital television applications" in *ISSCC Dig. Tech. Papers*, pp. 68-69, 1994
- [2] F. Lu, H. Samuelli, "A 60-MBd, 480-Mb/s, 256-QAM decision-feedback equalizer in 1.2-um CMOS" *IEEE J. of Solid-State Circuits*, vol. 28, no. 3, pp. 330-338, Mar., 1993.
- [3] R.E. Blahut, *Fast Algorithm for Digital Signal Processing*, Addison-Wesley, 1987
- [4] J.E. Volder, "The CORDIC trigonometric computing technique", *IEEE Trans. on Electronic Computers*, vol. EC8, pp. 330, Sep., 1959.
- [5] H.M. Ahmed, "Alternative arithmetic unit architecture for VLSI signal processors", in *VLSI and Modern Signal Processing*, Chap. 16, Prentice-Hall, Inc. 1985.

- [6] S. He, M. Torkelson, "A pipelined bit-serial complex multiplier using distributed arithmetic", *Proc. of ISCAS'95*, pp. 2313-2316, 1995.
- [7] R. Krishnan, G.A. Jullien, W.C. Miller, "Complex digital signal processing using quadratic residue number systems", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-34, no. 1, Feb., 1985.
- [8] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic", *IRE Trans. Electron. Compu.*, vol. EC-10, pp. 389-400, Sep., 1961.
- [9] N. Takagi, H. Yasuura and S. Yajima, "High-speed VLSI multiplication algorithm with a redundant binary addition tree", *IEEE Trans. on Computers*, vol. C-34, no. 9, pp. 789-796, Sep. 1985.
- [10] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation", *Dig. Tech. Papers of 1988 ISSCC*, pp. 152-153, Feb. 1988.
- [11] X. Huang, W.J. Liu, Belle W.Y. Wei, "A high-performance CMOS redundant binary multiplication-and-accumulation(MAC) unit", *IEEE Trans. on Circuits and Systems-I*, vol. 41, no. 1, pp. 33-39, Jan., 1994.
- [12] H. Makino, Y. Nakase, H. Suzuki, et al, "An 8.8-ns 54 x 54-bit multiplier with high speed redundant binary architecture", *IEEE J. of Solid-State Circuits*, vol. 31, no. 6, pp. 773-782, Jun., 1996.
- [13] W. Balakrishnan, N. Burgess, "Very-high-speed VLSI 2's complement multiplier using signed binary digits", *IEE Proceedings, Part-E*, vol. 139, no. 1, pp. 29-34, Jan., 1992.
- [14] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, N. Takagi, "A high-speed multiplier using a redundant binary adder tree", *IEEE J. of Solid-State Circuits*, vol. 22, no. 1, pp. 28-33, Feb., 1987.

 저 자 소 개



辛 卿 旭(正會員)

1961년 10월 26일생. 1984년 2월 한국항공대학교 전자공학과 졸업(공학사). 1986년 2월 연세대학교 대학원 전자공학과 졸업(공학석사). 1990년 8월 연세대학교 대학원 전자공학과 졸업(공학박사). 1990년 9월 ~ 1991년 6월 한국전자통신연구소 반도체 연구단 선임연구원. 1991년 7월 ~ 현재 금오공과

대학교 전자공학과 조교수. 1995년 8월 ~ 1996년 7월 Univ. of Illinois at Urbana-Champaign 방문연구. 주관심 분야는 통신 및 신호처리용 집적회로 설계, 저전압/저전력 집적회로 설계.