

論文98-35C-9-2

Radix-4 Booth Recoding과 RB 연산을 이용한 새로운 복소수 승산 알고리즘 및 10-bit CMAC 코어 설계

(A New Complex-Number Multiplication Algorithm using Radix-4 Booth Recoding and RB Arithmetic, and a 10-bit CMAC Core Design)

金 虎 河 * , 辛 卿 旭 *

(Ho-Ha Kim and Kyung-Wook Shin)

요 약

고속 복소수 연산장치는 채널등화, 동기신호 복원, 변조 및 복조 등 디지털 통신 시스템의 기저대역 신호처리에 필수적인 기능블록이다. 본 논문에서는 redundant binary (RB) 연산과 radix-4 Booth recoding을 결합한 새로운 복소수 승산 알고리즘을 제안한다. 제안되는 복소수 승산 방법은 실수 승산기를 사용하는 기존의 방법과 비교하여 부분곱의 수를 반으로 감소시키며, 단순화된 병렬구조로 구현되므로 고속 동작 및 저전력 소모를 가능하게 한다. 제안된 알고리즘을 적용하여 10-bit operand를 갖는 prototype 복소수 승산-누적기 (complex-number multiplier-accumulator ; CMAC) 코어를 0.8- μ m N-Well CMOS 공정으로 설계, 제작하였다. 제작된 CMAC 칩은 18,000여개의 트랜지스터로 구성되며, 코어부분의 면적은 약 $1.60 \times 1.93 \text{ mm}^2$ 이다. 제작된 칩을 테스트 보드에 실장하여 특성을 평가한 결과, 전원전압 $V_{DD}=3.3\text{-V}$ 에서 120-MHz의 속도로 동작함을 확인하였으며, 이때의 전력소모는 약 63-mW로 측정되었다.

Abstract

High-speed complex-number arithmetic units are essential to baseband signal processing of modern digital communication systems such as channel equalization, timing recovery, modulation and demodulation. In this paper, a new complex-number multiplication algorithm is proposed, which is based on redundant binary (RB) arithmetic combined with radix-4 Booth recoding scheme. The proposed algorithm reduces the number of partial product by one-half as compared with the conventional direct method using real-number multipliers and adders. It also leads to a highly parallel architecture and simplified circuit, resulting in high-speed operation and low power dissipation. To demonstrate the proposed algorithm, a prototype complex-number multiplier-accumulator (CMAC) core with 10-bit operands has been designed using 0.8- μ m N-Well CMOS technology. The designed CMAC core contains about 18,000 transistors on the area of about $1.60 \times 1.93 \text{ mm}^2$. The functional and speed test results show that it can operate with 120-MHz clock at $V_{DD}=3.3\text{-V}$, and its power consumption is given to about 63-mW.

* 正會員, 金烏工科大學校 電子工學部
(Department of Electronic Engineering, Kumoh
National University of Technology)

※ 이 논문은 1997년 한국학술진흥재단의 공모과제

연구비에 의하여 연구되었으며, 반도체설계교육센터
(IDEC)로 부터의 부분적인 지원을 받아 이루어 졌음.
接受日字: 1998年5月8日, 수정완료일: 1998年6月29日

I. 서론

최근, 디지털 신호처리 기술의 발전에 의해 무선 데이터 통신 시스템의 새로운 응용분야가 급속히 확대되고 있다. 예를 들어, High-Definition TV (HDTV), 차세대 무선 디지털 CATV 및 광대역 무선 데이터 모뎀 등의 실용화에 대한 연구가 활발히 진행되고 있다. 이와 같은 고비트율 디지털 통신 시스템들은 quadrature phase shift-keying (QPSK) 변조나 quadrature amplitude modulation (QAM) 등의 변조방식을 기반으로 하므로 이들 시스템의 기저대역 (baseband) 신호처리를 위해서는 in-phase와 quadrature (I-Q) 채널을 처리하기 위한 복소수 연산이 필수적이다. 예를 들어, quadrature 변·복조기 및 타이밍 복원기, inter-symbol interference (ISI) 제거를 위한 decision-feedback equalization (DFE) 등의 구현을 위해서는 복소수 승산기 또는 승산-누적기가 핵심 구성요소가 된다^[1,2]. 한편, CMOS 디지털 집적회로 기술의 발전에 의해 변·복조, 타이밍 복원, 채널등화 등 대부분의 송·수신기 기능을 디지털로 처리하는 추세이므로, 이들 시스템에 적합한 고속/저전력 복소수 연산장치에 대한 중요성이 더욱 증대되고 있다.

복소수 승산기를 구현하는 가장 일반적인 방법은 실수 승산기와 실수 가산기를 이용하는 것이다. 예를 들어, 4개의 실수 승산기와 2개의 실수 가산기를 사용하는 직접적인 방법과 3개의 실수 승산기와 5개의 실수 가산기를 사용하는 strength reduction 방법^[3]이 사용되고 있다. 이와 같은 고전적인 방법에서는 실수 승산기 및 가산기의 성능이 복소수 승산기의 성능을 결정하는 요인이 되며, 또한 실수 승산후의 가산/감산 과정이 전체 복소수 승산기의 속도를 제한하는 단점을 갖는다. 따라서, 디지털 통신 시스템의 기저대역 신호 처리에 적합한 고속/고집적/저전력 복소수 연산장치의 구현을 위해서는 기존의 방식과는 다른 새로운 복소수 승산 방식에 대한 연구가 필요하다.

본 논문에서는 고성능 복소수 승산-누적기의 구현에 적합한 새로운 복소수 승산 방법을 제안하고자 하며, 이를 적용한 prototype 회로설계, 제작 및 특성평가 결과를 기술하고자 한다. 본 논문에서 제안하는 복소수 승산방법은 기존의 2진 수치계 대신에 redundant binary (RB) 수치계를 사용하며, RB 수

치계와 radix-4 Booth recoding 기법을 혼합한 새로운 개념을 바탕으로 하고 있다. 이를 통해 복소수 승산의 연산량을 줄이고 회로를 단순화함으로써 전체적인 성능향상 및 전력소모 감소 등을 이루고자 한다.

RB 수치계에 의한 연산은 가산과정시 최하위 비트에서 최상위 비트로 캐리신호의 전파가 없는 carry-propagation-free (CPF) 특성을 갖는다^[4]. 이와 같은 RB 연산의 CPF 특성을 이용하면 승산기 구현시 캐리전파 없이 부분곱을 가산할 수 있으므로, 최근에 이를 실수 승산기 설계에 적용한 연구결과가 일부 발표되고 있다^[5,6]. 최근, RB 수치계를 바탕으로 한 복소수 승산 알고리즘과 prototype chip 설계 및 성능평가에 대한 결과가 발표된 바 있으며^[7,8], 본 논문에서는 이를 개선한 새로운 알고리즘을 제안하고자 한다.

본 논문의 2장에서는 RB 수치계에 대해 간략히 소개하고, 3장에서는 RB 수치계와 radix-4 Booth recoding을 혼합한 새로운 복소수 승산방법을 제안한다. 4장에서는 제안된 알고리즘을 적용하여 실수부와 허수부의 operand가 각각 10-비트인 복소수 승산-누적기의 회로설계에 대해 언급하고, 5장에서는 0.8- μ m N-Well CMOS 공정을 이용한 prototype 칩 제작 및 제작된 칩의 특성평가 결과를 기술한다.

II. Redundant Binary (RB) 수치계

1. Signed-digit 수치계와 RB 수치계

RB 수치계는 Avizienis에 의해 제안된 signed-digit (SD) 수치계^[9]의 일종이다. 일반적으로 radix-r SD 수치계로 표현된 n-digit의 수 $Y=[y_{n-1} \dots y_1 y_0]_{SD}$ 의 각 digit y_i 는 $(2a+1)$ 개의 정수로 구성되는 digit set $\{-a, \dots, -1, 0, 1, \dots, a\}$ 중의 한 값을 가지며, 이때 양의 정수 a 는 식(1)로 주어진 범위의 값을 갖는다.

$$\lceil \frac{r-1}{2} \rceil \leq a \leq r-1 \tag{1}$$

한편, SD 수 Y 의 algebraic value Y_v 는 식(2)로 표현된다.

$$Y_v = \sum_{i=0}^{n-1} y_i \cdot r^i \tag{2}$$

SD 수치계는 수의 표현을 위해 식(1)에 정의된 범위의 양수와 음수를 사용하므로 임의의 수를 여러 가지 형태로 표현할 수 있어 redundant 수치계라고도 한다. RB 수치계는 digit set $\{-1, 0, 1\}$ 을 사용하여 수를 표현하는 radix-2 SD 수치계이며, 수 표현의 redundancy에 의해 임의의 수를 여러 가지 형태로 표현할 수 있다. 예를 들어, 정수 5는 $[0\ 1\ 0\ 1]_{RB}$, $[0\ 1\ 1\ -1]_{RB}$, $[1\ -1\ 1\ -1]_{RB}$, $[1\ -1\ 0\ 1]_{RB}$, 그리고 $[1\ 0\ -1\ -1]_{RB}$ 등 여러 가지 형태로 표현이 가능하다.

RB 수치계가 갖는 수 표현의 redundancy 특성을 이용하면 하위 digit에서 상위 digit로 캐리전파가 일어나지 않도록 가산 규칙을 정의할 수 있음이 입증되었다^[49]. 이는 RB 수치계로 표현된 두 수의 가산이 캐리전파 없이 처리될 수 있으며, 따라서 operand의 digit 수에 무관하게 일정한 시간 내에 가산이 완료될 수 있음을 의미한다. 승산은 부분곱의 연속적인 가산으로 구성되므로 부분곱을 RB 수치계로 표현하면 부분곱의 가산을 캐리전파 없이 고속으로 실현할 수 있다. 이와 같은 RB 수치계의 특성을 이용한 실수 승산기의 구현 사례가 발표되고 있다^[56].

2. 2의 보수 수치계와 RB 수치계의 상호변환

n-digit의 RB 수 $D = [d_{n-1} \dots d_1 d_0]_{RB}$ (단, $d_i \in \{-1, 0, 1\}$)의 algebraic value D_v 는 식(3)과 같이 정의된다.

$$D_v = \sum_{i=0}^{n-1} d_i 2^i \tag{3}$$

SD d_i 는 digit set $\{-1, 0, 1\}$ 중의 한 값을 가지므로 2-bit의 2진 코드 (x_i, y_i) 로 표현되며, 이때 d_i 와 (x_i, y_i) 사이의 관계는 식(4)와 같이 정의될 수 있다.

$$d_i \equiv (x_i, y_i) \equiv x_i - \overline{y_i} \tag{4}$$

단, $\overline{y_i}$ 는 y_i 의 반전을 의미함

식(4)를 확장하면, n-digit의 RB 수 D 와 n-bit의 이진수 X, Y 의 관계를 식(5)와 같이 정의할 수 있다.

$$D \equiv (X, Y) \equiv X - \overline{Y} \tag{5}$$

단, $D = [d_{n-1} d_{n-2} \dots d_1 d_0]_{RB}$

$$X = [x_{n-1} x_{n-2} \dots x_1 x_0]_{TC}$$

$$Y = [y_{n-1} y_{n-2} \dots y_1 y_0]_{TC}$$

식(5)에서 \overline{Y} 는 Y 의 모든 비트의 반전을 의미하며, 첨자 RB와 TC는 각각 RB 수와 two's complement 수를 나타낸다. 한편 X 와 Y 가 2의 보수 형태라고 하면, 두 수의 합 $X+Y$ 는 식(6)과 같이 RB 수 D 로 변환할 수 있다.

$$\begin{aligned} X+Y &= X - (\overline{Y} + 1) \\ &= (X, Y) - 1 \\ &= D - 1 \end{aligned} \tag{6}$$

식(6)으로부터, 2의 보수로 표현된 두 수의 합은 하나의 RB 수 D 로 변환될 수 있으며, 이와 같은 수치계의 변환은 연산과정 없이 직접 이루어짐을 알 수 있다.

한편 식(6)으로부터, (X, Y) 로 표현된 RB 수 D 를 2의 보수 수치계 B 로 변환하기 위한 관계식은 식(7)과 같이 정의될 수 있다.

$$B = X + Y + 1 \tag{7}$$

식(7)에서 알 수 있듯이, RB 수치계로부터 2의 보수 수치계로의 변환은 RB 수를 표현하는 두 수의 가산에 의해 이루어지며, 이는 일반적인 2진 가산기를 사용하여 구현된다.

III. Radix-4 Booth Recoding과 RB 수치계를 이용한 복소수 승산

본 장에서는 복소수 승산을 보다 효율적으로 구현하기 위한 새로운 방법을 제안한다. 제안되는 복소수 승산 알고리즘은 radix-4 Booth recoding과 부분곱의 RB 수치계 변환을 바탕으로 하며, 이를 통하여 부분곱의 감소, 캐리전파 없는 부분곱의 가산 그리고 전체적인 복소수 승산기 구조의 단순화 등이 얻어지도록 한다.

실수부와 허수부가 각각 n-bit 2의 보수로 표현된 복소수 A 와 B , 그리고 그들의 곱 $Z = A \cdot B$ 에 대해 생각한다.

$$A = A_R + jA_I \tag{8}$$

$$B = B_R + jB_I \tag{9}$$

$$Z = Z_R + jZ_I$$

$$= (A_R B_R - A_I B_I) + j(A_R B_I + A_I B_R) \quad (10)$$

1. Radix-4 Booth recoding

잘 알려진 바와 같이, 승산기 구현에 널리 사용되고 있는 radix-4 Booth recoding은 승수를 3-bit씩 묶어 그 비트 패턴에 따라 부분곱을 생성함으로써 부분곱의 수를 1/2로 줄이는 방법이다. 일반적으로, n -bit 승수 Y 에 대한 Radix-4 Booth recoding은 식(11)과 같이 표현된다.

$$Y = \sum_{k=0}^{n/2-1} (-2y_{2k+1} + y_{2k} + y_{2k-1}) \cdot 2^{2k}$$

$$= \sum_{k=0}^{n/2-1} Q(k) \cdot 2^{2k} \quad (11)$$

식(11)에서 $y_{-1}=0$ 이며, $Q(k) \in \{-2, -1, 0, +1, +2\}$ 는 3-bit로 grouping된 승수의 비트 조합에 따라 표 1과 같이 정의된다. $Q(k)$ 는 표 1에 정의된 것과 같이 $S(k), X_1(k), X_2(k)$ 로 분해되어 부분곱 생성을 위한 제어신호로 사용된다. 표 1에서 $S(k)='1'$ 은 $Q(k)$ 가 음수 값을 가짐을 나타내며, 이는 3-3절에서 설명될 correction 항을 생성하는 데 사용된다. 식(11)을 이용하여 두 수의 곱 P 를 식(12)와 같이 표현할 수 있다.

$$P = X \cdot Y = \sum_{k=0}^{n/2-1} Q(k) X \cdot 2^{2k} \quad (12)$$

여기서 $Q(k) X \cdot 2^{2k}$ 는 Booth recoding에 의해 생성되는 이진 부분곱을 나타낸다.

식(11)과 식(12)가 나타내는 radix-4 Booth recoding을 식(10)에 적용하면, 복소수 승산의 실수부와 허수부는 각각 식(13)과 식(14)와 같이 표현된다.

$$Z_R = A_R B_R - A_I B_I$$

$$= \sum_{k=0}^{n/2-1} Q_R(k) A_R \cdot 2^{2k} - \sum_{k=0}^{n/2-1} Q_I(k) A_I \cdot 2^{2k} \quad (13)$$

$$Z_I = A_R B_I + A_I B_R$$

$$= \sum_{k=0}^{n/2-1} Q_I(k) A_R \cdot 2^{2k} + \sum_{k=0}^{n/2-1} Q_R(k) A_I \cdot 2^{2k} \quad (14)$$

여기서 $Q_R(k)$ 와 $Q_I(k)$ 는 식(11)을 B_R 와 B_I 에 각각 적용하여 얻어지는 signed-digit (SD)이며, 표 1과 동일하게 정의된다. 잘 알려진 바와 같이, 기존에 널리

사용되고 있는 복소수 승산기 구현방식은 식(13)과 식(14)를 4개의 실수 승산기와 2개의 실수 가산기를 사용하여 구현하는 것이며, 이는 총 $2n$ 개의 이진 부분곱 가산을 필요로 한다.

표 1. Radix-4 Booth 리코딩

Table 1. Radix-4 Booth recoding table.

y_{2k+1}	y_{2k}	y_{2k-1}	$S(k)$	$X_2(k)$	$X_1(k)$	$Q(k)$
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	0	1	1
0	1	1	0	1	0	2
1	0	0	1	1	0	-2
1	0	1	1	0	1	-1
1	1	0	1	0	1	-1
1	1	1	0	0	0	0

2. Booth recoded RB 부분곱을 이용한 복소수 승산 알고리즘

본 절에서는 2의 보수형태로 표현된 두 수의 합(승) 또는 차(差)를 RB 수치계로 변환하는 과정을 통해 식(13)과 식(14)에 정의된 이진 부분곱을 RB 부분곱으로 변환하는 과정에 대해 언급한다.

먼저, 복소수 승산의 실수부를 나타내는 식(13)은 다음과 같이 다시 표현할 수 있다.

$$Z_R = \sum_{k=0}^{n/2-1} [P_{rr}(k) - P_{ii}(k)] \cdot 2^{2k}$$

$$= \sum_{k=0}^{n/2-1} \left[-(e_{k,n} - f_{k,n}) 2^n + \sum_{j=0}^{n-1} (e_{k,j} - f_{k,j}) 2^j \right] \cdot 2^{2k} \quad (15)$$

식(15)에서, $P_{rr}(k) \equiv A_R \cdot Q_R(k)$ 와 $P_{ii}(k) \equiv A_I \cdot Q_I(k)$ 는 radix-4 Booth recoding에 의해 생성된 k -번째 이진 부분곱을 나타내며, $e_{k,j}$ 와 $f_{k,j}$ (단, $e_{k,j}, f_{k,j} \in \{0, 1\}$)는 각각 $P_{rr}(k)$ 와 $P_{ii}(k)$ 의 j -번째 비트를 나타낸다. 한편, SD 수 $\gamma_{k,j}$ 와 $\gamma_{k,j}^*$ (단, $\gamma_{k,j}, \gamma_{k,j}^* \in \{-1, 0, 1\}$)를 식(16)과 같이 정의하면, 복소수 승산의 실수부를 구성하는 RB 부분곱 $P_R(k)$ 는 식(17)과 같이 표현된다.

$$\gamma_{k,j} \equiv e_{k,j} - f_{k,j} \quad (16.a)$$

$$\gamma_{k,j}^* \equiv -(e_{k,j} + f_{k,j}) \quad (16.b)$$

$$P_R(k) = \left[\gamma_{k,n}^* 2^n + \sum_{j=0}^{n-1} \gamma_{k,j} 2^j \right] \cdot 2^{2k} \quad (17)$$

마찬가지로, 식(14)가 나타내는 복소수 승산의 허수부는 다음과 같이 다시 표현할 수 있다.

$$\begin{aligned}
 Z_I &= [P_{ri}(k) + P_{ir}(k)] \cdot 2^{2k} \\
 &= [P_{ri}(k) - \overline{P_{ir}(k)} - 1] \cdot 2^{2k} \\
 &= \left[-(g_{k,n} - \overline{h_{k,n}}) 2^n + \sum_{j=0}^{n-1} (g_{k,j} - \overline{h_{k,j}}) 2^j - 1 \right] \cdot 2^{2k}
 \end{aligned}
 \tag{18}$$

식(18)에서 $P_{ri}(k) \equiv A_R \cdot Q(k)$ 와 $P_{ir}(k) \equiv A_I \cdot Q(k)$ 는 radix-4 Booth recoding에 의해 생성된 k-번째 이진 부분곱을 나타내며, $g_{k,j}$ 와 $h_{k,j}$ (단, $g_{k,j}, h_{k,j} \in \{0, 1\}$) 는 각각 $P_{ri}(k)$ 와 $P_{ir}(k)$ 의 j-번째 비트를 나타낸다. 한편, SD 수 $\lambda_{k,j}$ 와 $\lambda_{k,j}^*$ (단, $\lambda_{k,j}, \lambda_{k,j}^* \in \{-1, 0, 1\}$) 를 식(19)와 같이 정의하면, 복소수 승산의 허수부를 구성하는 RB 부분곱 $P_I(k)$ 는 식(20)과 같이 표현된다.

$$\lambda_{k,j} \equiv g_{k,j} - \overline{h_{k,j}} \tag{19.a}$$

$$\lambda_{k,j}^* \equiv -(g_{k,j} - \overline{h_{k,j}}) \tag{19.b}$$

$$P_I(k) = \left[\lambda_{k,n}^* 2^n + \sum_{j=0}^{n-1} \lambda_{k,j} 2^j \right] \cdot 2^{2k} \tag{20}$$

따라서, n-bit 복소수 승산은 RB 수치계로 변환된 부분곱 $P_R(k)$ 과 $P_I(k)$ 를 이용하여 식(21), 식(22)와 같이 표현될 수 있다.

$$Z_R = \sum_{k=0}^{n/2-1} P_R(k) \tag{21}$$

$$Z_I = \sum_{k=0}^{n/2-1} (P_I(k) - 2^{2k}) \tag{22}$$

식(21)과 식(22)는 각각 RB 수치계로 변환된 복소수 승산의 실수부와 허수부를 나타내며, 실수부와 허수부가 각각 n/2개의 RB 부분곱의 합으로 간략화되었다. 따라서 2진 수치계를 이용하는 식(13), (14)의 경우와 비교할 때, 부분곱의 수가 1/2로 감소되었음을 알 수 있다. 이와 같은 부분곱의 감소는 복소수 승산기 회로의 단순화와 함께 승산 속도의 향상, 전력소모의 감소 등을 가능하게 한다. 또한 II장에서 언급된 바와 같이, RB 수치계가 갖는 연산특성에 의해 RB 부분곱을 캐리전과 없이 고속으로 가산할 수 있어 전체적인 승산속도가 개선되는 장점을 갖는다. 한편, 식(21), (22)에는 부호 비트에 대한 정보가 고려되어 있으므로 RB 부분곱 가산 시에 부호확장을 고려할 필요가 없다는 장점도 갖는다.

한편, 식(16)과 식(19)에서 정의된 SD 수 $\gamma_{k,j}$ 와 $\lambda_{k,j}$ 는 $\{-1, 0, 1\}$ 중의 한 값을 가지므로 이들은

2-bit의 이진코드로 표현되어야 한다. 본 논문에서는 효율적인 RB 부분곱의 생성을 위해 이진 부분곱 $e_{k,j}, f_{k,j}, g_{k,j}, h_{k,j}$ 와 SD 수 $\gamma_{k,j}, \lambda_{k,j}$ 의 관계를 식(23)과 같이 정의하였다. 식(23)에서 볼 수 있듯이, RB 부분곱은 추가적인 하드웨어나 지연시간 없이 이진 부분곱으로부터 직접 생성할 수 있다.

$$\gamma_{k,j} \equiv (e_{k,j}, f_{k,j}) \tag{23.a}$$

$$\gamma_{k,j}^* \equiv (\overline{e_{k,j}}, \overline{f_{k,j}}) \tag{23.b}$$

$$\lambda_{k,j} \equiv (g_{k,j}, \overline{h_{k,j}}) \tag{23.c}$$

$$\lambda_{k,j}^* \equiv (\overline{g_{k,j}}, h_{k,j}) \tag{23.d}$$

3. 효율적인 Correction 항 처리

본 절에서는 RB 부분곱을 생성하는 과정에서 발생되는 상수항들을 효율적으로 처리할 수 있도록 고안된 correction 항에 대해 언급한다.

correction 항으로 처리되는 상수항들은 다음과 같은 두 가지 요인에 의해 발생된다. 첫째, radix-4 Booth recoding에 의해 생성된 2진 부분곱이 음수(즉, 식(13)과 식(14)에서 $Q_R(k), Q_I(k)$ 가 음수)인 경우에는 피승수의 모든 비트를 반전시킨 후 최하위 비트에 '1'을 더해야 한다. 따라서, 상수항의 발생여부는 승수의 비트 패턴에 의존하게 된다. 둘째, radix-4 Booth recoding에 의해 생성된 이진 부분곱을 RB 부분곱으로 변환하는 과정중 허수부의 경우에 식(18)에서 보는 바와 같이 n/2개의 -2^{2k} 항이 발생된다. 따라서, 이와 같은 두 가지 요인에 의해 생성되는 상수항은 실수부와 허수부가 서로 다르며, 또한 승수의 비트 패턴에 따라 달라진다.

표 2. Correction 항

Table 2. Correction terms.

$S_R(k)$	$S_I(k)$	$C_R(k)$	$C_I(k)$
0	0	0	-1
0	1	-1	0
1	0	1	0
1	1	0	1

본 논문에서는 이를 효율적으로 처리하기 위해 correction 항 $C_R(k), C_I(k)$ 를 식(24)와 같이 정의하였으며, $S_R(k), S_I(k)$ 와의 관계는 표 2와 같다. 표 2에서 $S_R(k)$ 와 $S_I(k)$ 는 각각 $Q_R(k)$ 와 $Q_I(k)$ 가 음수일 때 '1'이 되며, 표 1의 진리표에 의해 생성된다. 따라

서, correction 항 $C_R(k)$, $C_I(k)$ 는 부가적인 하드웨어 없이 승수 B_R 와 B_I 에 대한 Booth으로 recoding 결과로부터 직접 생성될 수 있다.

$$C_R(k) \equiv (S_R(k), \overline{S_I(k)}) \quad (24.a)$$

$$C_I(k) \equiv (S_R(k), S_I(k)) \quad (24.b)$$

4. 제안된 알고리즘의 평가

본 논문에서 제안된 새로운 복소수 승산 알고리즘을 기존의 방식과 비교한 결과를 표 3에 요약하였다.

표 3. 복소수 승산방식의 비교
Table 3. Comparison of complex-number multiplication methods.

	Direct Method† (방법-①)	Strength Reduction† (방법-②)	문헌 [6,7]의 방법 (방법-③)	본 논문의 방법 (방법-④)
Number Representation	Two's Complement	Two's Complement	Redundant Binary	Redundant Binary
Total # of Partial Product	2n	1.5n	2n	n
Carry Propagation	Yes	Yes	No	No
Post-Additions	Yes	Yes	No	No
Complexity	4 Multi-plications 2 Additions	3 Multi-plications 5 Additions	2 RB Multi-plications 2 RB Additions*	2 RB Multi-plications 2 RB Additions*

† using real-number multipliers based on radix-4 Booth recoding

※ including RB-to-Binary Conversion (RB2BC) circuit

방법-①과 방법-②^[3]는 기존의 2진 수치계를 바탕으로 한 구현방법이고, 방법-③과 방법-④는 RB 수치계를 바탕으로 한 구현방법이다. 일반적으로 실수 승산기는 radix-4 Booth recoding 기법을 이용하여 부분 곱의 수를 줄이므로, 방법-①과 방법-②도 radix-4 Booth recoding을 적용하여 구현하는 것으로 가정하였다. 방법-①은 복소수 승산을 구현하는 가장 일반적인 형태로서 식(13)과 식(14)를 직접 구현하는 것이며, 그림 1-(a)에서 보는 바와 같이 4개의 실수 승산기와 2개의 실수 가산기를 필요로 한다. 한편, strength reduction 기법을 이용한 방법-②는 그림 1-(b)에서 보는 바와 같이 한 개의 실수 승산기를 줄이는 대가로 3개의 실수 가산기가 추가로 필요하므로 승산기의 critical path가 길어지는 단점을 갖는다. 한편, 문헌

[7,8]에 발표된 방법-③은 RB 수치계를 이용하여 복소수 승산을 구현하는 방법이며, 본 논문에서 제안된 방법-④는 RB 수치계와 radix-4 Booth recoding을 적용하여 방법-③을 개선한 것이다. 표 3에서 볼 수 있듯이, 본 논문에서 제안된 복소수 승산 방법은 n개의 부분곱 (단, n은 승수의 비트 수) 만을 필요로 하므로 기존의 방법들에 비해 부분곱 가산이 단순화된다. 또한, 방법-①, 방법-②와 같이 2진 수치계를 이용하는 방법에서는 실수 승산 후에 post-addition 과정이 필요한 반면에, 본 논문의 방법은 RB 승산을 바탕으로 하므로 post-addition 과정이 필요 없이 승산기의 critical path가 짧아지며, 전체적인 승산기 구조가 단순화되는 장점을 갖는다. 또한, 부분곱이 RB 형태로 생성되므로 부분곱을 캐리전파 없이 고속으로 가산할 수 있다는 특징도 갖는다.

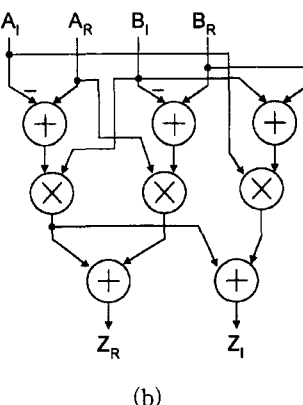
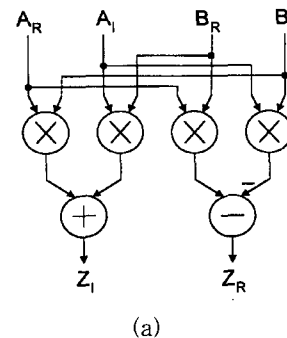


그림 1. 실수 승산기를 이용한 복소수 승산 방식
Fig. 1. Complex-number multiplication using real-number multipliers.

IV. 복소수 승산-누적기 코어 설계

본 장에서는 III장에서 제안된 복소수 승산 알고리

들을 적용한 10-bit 복소수 승산-누적기 코어의 설계에 대해 설명한다. 복소수 승산-누적기는 두 개의 복소수 데이터를 입력받아 한 개의 복소수 데이터를 출력하므로 많은 입·출력 핀을 필요로 하며, operand 비트 수는 사용가능한 입·출력 핀 수에 제한을 받는다. 본 논문에서는 반도체설계교육센터(IDECC)에서 제공하는 현대 0.8- μm CMOS 공정 (사용 가능한 입·출력 핀 수는 100핀)^[10]을 이용하였으며, 입·출력 핀 수의 제한을 고려하여 실수부와 허수부 operand를 모두 10-bit로 결정하였다. 한편, operand의 비트 수를 확장하는 경우에도 III장에서 설명된 알고리즘을 적용하여 동일한 방법으로 구현될 수 있다.

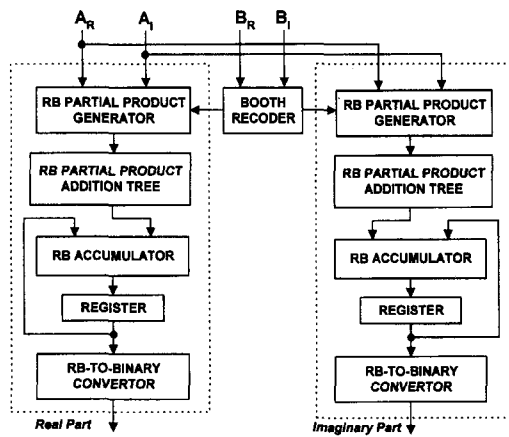


그림 2. 복소수 승산-누적기의 구조
Fig. 2. Architecture of the complex-number multiplier-accumulator.

설계된 복소수 승산-누적기의 구조는 그림 2와 같으며, 서로 대칭인 두개의 RB 승산-누적기로 구성된다. 각각의 RB 승산-누적기는 RB 부분곱 생성기, RB 가산기 및 누적기, 그리고 수치계 변환기 등 4개의 기능블록으로 구성되며, 복소수 연산의 실수부(허수부) 연산을 담당한다. RB 부분곱 생성기는 radix-4 Booth recoding을 통해 5개의 RB 부분곱과 correction 항을 생성하며, III장에서 언급된 바와 같이 이전 부분곱으로부터 부가적인 하드웨어 없이 직접 RB 부분곱이 생성된다. RB 가산기는 생성된 부분곱과 correction 항을 가산하는 복소수 승산기능을 수행하며, RB 전가산기(RBFA) 셀과 RB 반가산기(RBHA) 셀들의 배열로 구성된다. RB 누적기는 24-digit RB 가산기와 레지스터로 구성되며 승산결과를 누적하는 기능을 수행한다. 마지막으로, 수치계 변

환기는 RB 누적기의 출력을 2진 수치계로 변환하는 기능을 수행한다. 전체 회로의 동작속도를 향상시키기 위하여 RB 부분곱 가산기 내부에 1 stage의 파이프라인을 삽입하였으며, 누산기 레지스터와 입력 및 출력 레지스터를 포함하여 전체 4 clock cycles의 latency를 갖는다.

RBFA는 두개의 SD 데이터와 캐리입력을 받아, 그들의 합과 캐리출력을 생성하는 기능을 수행한다. 문헌 [4]에 언급된 바와 같이, RB 수치계가 갖는 redundancy 특성을 이용하면 (i-1)-번째 digit에서 전파된 캐리신호가 i-번째 RBFA 셀을 통해 (i+1)-번째 digit로 전달되지 않도록 RBFA 셀을 정의할 수 있으며, 따라서 RB 부분곱 가산기에 최하위 digit에서 최상위 digit로 캐리신호의 전파가 없는 고속가산을 실현할 수 있다. RBFA와 RBHA의 논리기능은 문헌 [8]과 동일하게 설계되었으며, 회로성능의 최적화를 위해 전달게이트 구조를 사용하여 그림 3과 같이 설계하였다.

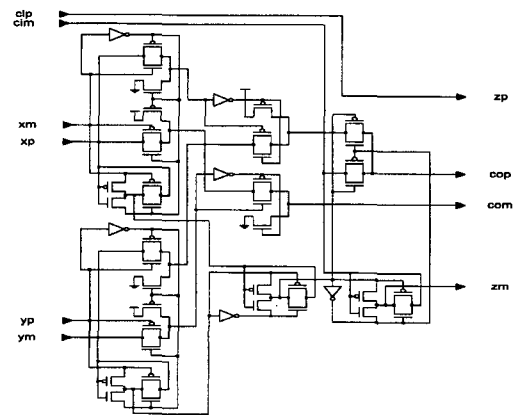


그림 3. RB 전가산기(RBFA) 회로
Fig. 3. RB full adder (RBFA) circuit.

설계된 10-bit 복소수 승산-누적기 회로는 18,000여개의 트랜지스터로 구성되며, 약 1.60 x 1.93-mm²의 면적을 갖는다. 전체회로의 레이아웃 설계는 Cadence 설계환경을 사용하여 full custom 방식으로 수행하였다.

설계된 복소수 승산-누적기는 부분곱 가산과 승산 결과의 누적연산이 모두 RB 수치계로 처리되므로 모든 가산과정이 캐리전파 없이 이루어지며, 따라서 기존의 2진 수치계를 이용하는 방법보다 고속연산이 가

능하고, 부분곱 가산시 부호확장을 고려할 필요가 없다는 특징을 갖는다.

V. 제작된 칩의 성능평가

설계된 10-bit 복소수 승산-누적기 회로를 0.8- μ m N-Well CMOS 공정으로 제작하였으며, 코어 부분의 확대사진은 그림 4와 같다. 제작된 칩을 테스트하기 위해 테스트 보드를 제작하였으며, PC와 디지털 I/O 보드, 논리분석기, 펄스 발생기 등을 이용하여 논리기능과 동작속도 특성을 평가하였다. 그림 5는 테스트 장비의 구성도이며, 그림 6은 제작된 칩이 실장된 테스트 보드의 사진이다.

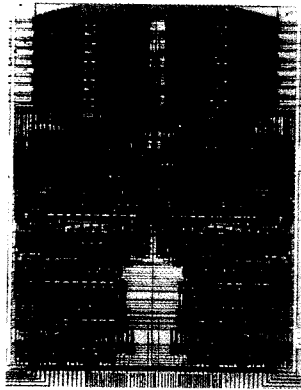


그림 4. 제작된 CMAC 코어의 확대사진
Fig. 4. Microphotograph of the core part of the fabricated CMAC chip.

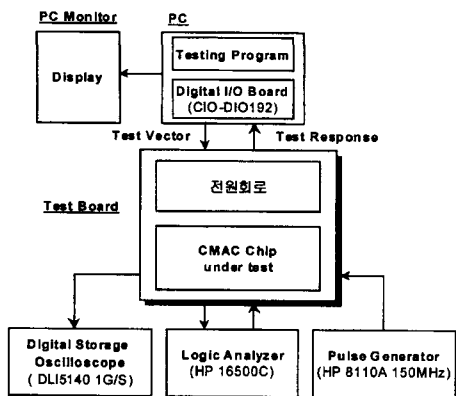


그림 5. CMAC 칩의 테스트 장비 구성도
Fig. 5. Test equipments for testing CMAC chip.

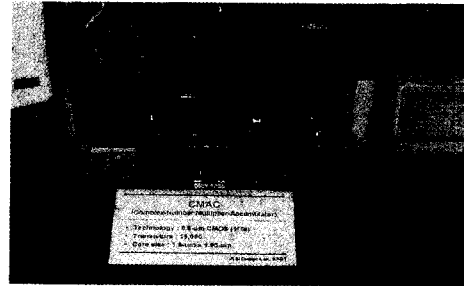


그림 6. CMAC 칩의 테스트 보드
Fig. 6. Test board for testing the CMAC chip.

제작된 칩의 논리기능을 확인하기 위해 PC와 C언어로 작성된 테스트 프로그램을 사용하여 다음과 같이 테스트하였다. PC에서 생성된 테스트 벡터를 디지털 I/O 보드를 통해 테스트 보드에 실장된 칩으로 인가하고, 칩에서 출력되는 데이터를 다시 디지털 I/O 보드를 통해 PC로 받은 후, PC에서 계산된 값과 비교함으로써 제작된 칩의 논리기능을 확인하였다. 또한, 논리분석기를 사용한 테스트도 수행하였으며, 그림 7은 HP16500C 논리분석기를 사용하여 칩의 논리기능을 테스트한 출력파형이다. 그림 7의 논리분석기 출력파형은 제작된 칩의 누적동작을 테스트한 결과이며, 클럭신호(CLK)의 상승 edge에서 복소수 승산-누적기의 실수부 출력(R_OUT)과 허수부 출력(L_OUT)이 각각 '+1'씩 증가되어 누적되는 결과를 보이고 있다. 이상과 같은 테스트 결과로부터 설계된 복소수 승산-누적기의 논리기능이 정상적으로 동작함을 확인할 수 있었다.

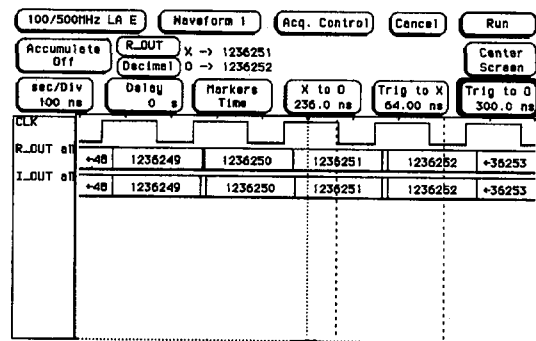


그림 7. 논리 분석기를 이용한 테스트 결과
Fig. 7. Test results using logic analyzer.

한편, 제작된 칩의 동작속도 및 전력소모 특성은 펄

스발생기와 디지털 오실로스코프를 사용하여 측정하였으며, 전원전압에 따른 최대 동작 주파수와 전력소모를 평가하였다. 제작된 복소수 승산-누적기는 $V_{dd}=3.3-V$ 에서 120-MHz의 클럭 주파수로 안전하게 동작함을 확인하였으며, 이때의 전력소모는 약 63-mW로 측정되었다.

표 4. CMAC 칩의 전원전압과 동작주파수에 따른 전력소모 특성

Table 4. Power dissipation characteristics of the fabricated CMAC chip.

전원전압 [V]	동작 주파수 [MHz]	전력소모 [mW]
3.3	120	63.03
3.0	110	52.50
2.5	100	34.75
2.2	90	28.38
1.8	40	6.3

표 5. 10-bit CMAC 칩의 주요 특성

Table 5. Summary of the 10-bit CMAC.

Technology	0.8- μ m double metal N-Well CMOS
Total transistor integrated	18,000
Layout area	1.60 x 1.93 mm ²
Latency	4 clock cycles
Operating clock frequency	120-MHz @3.3-V
Power dissipation	63-mW @3.3-V, 120-MHz
Layout density	5.8k Transistors/mm ²
I/O pins	97

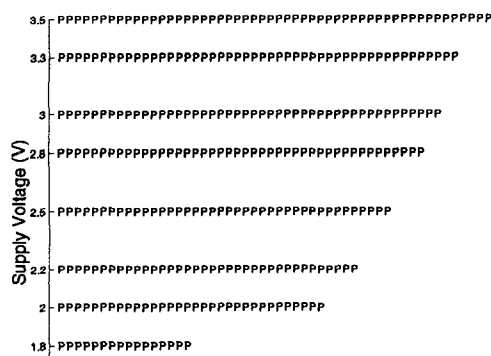


그림 8. CMAC 칩의 Schmoo plot.

Fig. 8. Schmoo plot of the CMAC chip.

그림 8은 측정된 데이터로부터 얻어진 Schmoo plot이며, 표 4는 전원전압과 동작 주파수에 따른 전력소모 특성을 측정한 결과이다. 표 5는 제작된 복소

수 승산-누적기의 특성을 요약한 것이다. 문헌에 발표된 13-bit 50-MHz@5-V 복소수 승산기^[11], 16-bit 100-MHz@5-V 복소수 승산기^[12] 등과 비교할때, 본 논문에서 설계된 회로는 복소수 승산과 누적연산을 120-MHz@3.3-V의 속도로 처리할 수 있으므로 이들보다 우수한 성능을 갖는다.

VI. 결 론

본 논문에서는 고성능 복소수 승산기의 구현에 적합한 새로운 복소수 승산 알고리즘을 제안하고, 이를 적용한 10-bit 복소수 승산-누적기를 설계, 제작하였다. 제안된 복소수 승산방법은 기존의 2진 수치계 대신에 RB 수치계와 radix-4 Booth recoding 기법을 혼합한 새로운 개념을 바탕으로 하며, 기존의 실수 승산기를 이용하는 방법에 비해 부분곱의 감소와 post-addition 과정의 제거를 통해 승산기의 동작속도 개선과 아울러 전체적인 복소수 승산기 구조의 단순화가 얻어진다. 특히, 승산 및 누적연산이 모두 RB 수치계로 처리되므로 모든 가산과정이 캐리전파 없이 이루어지며, 따라서 기존의 2진 수치계를 이용하는 방법보다 고속연산이 가능하고, 부분곱 가산시 부호확장을 고려할 필요가 없다는 특징을 갖는다. 제안된 방법을 적용하여 설계된 10-bit 복소수 승산-누적기는 18,000여개의 트랜지스터로 구성되며, 전원전압 3.3-V에서 120-MHz의 클럭 주파수로 동작하며, 약 63-mW의 전력소모 특성을 갖는다.

결론적으로, 본 논문에서 제안된 방법은 실수 승산기를 사용하는 종래의 고전적인 방법에 비해 고속/고집적/저전력 복소수 승산기의 구현이 가능하며, 디지털 통신 및 신호처리 시스템의 핵심 연산장치 구현에 적합할 것으로 평가된다.

참 고 문 헌

- [1] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.
- [2] K. Feher, *Wireless Digital Communications : Modulation & Spread Spectrum Applications*, Prentice Hall, 1995.
- [3] R.E. Blahut, *Fast Algorithm for Digital Signal Processing*, Addison-Wesley, 1987.

[4] 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.

[5] 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.

[6] 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.

[7] K. W. Shin and B. S. Song, "A complex-multiplier architecture based on redundant binary arithmetic", *Proc. of ISCAS'97*, pp. 1944-1947, 1997.

[8] K. W. Shin, B. S. Song, K. Bacrania, "A 200-MHz complex-number multiplier using redundant binary arithmetic", *IEEE J. of Solid-State Circuits*, vol. 33, no. 6, pp. 904-909, Jun., 1998.

[9] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic", *IRE Trans. Electron. Comp.*, vol. EC-10, pp. 389-400, Sep., 1961.

[10] 반도체설계교육센터, '97 IDEC MPW full custom design rule, 1997. 6

[11] T. Soulas, D. Villeger, V.G. Oklobdzija, "An ASIC macro cell multiplier for complex numbers", *Proc. of the European Conf. on Design Automation with the European Event in ASIC Design*, pp. 589-593, 1993.

[12] B.W.Y Wei, H. Du, H. Chen, "A complex-number multiplier using radix-4 digits", *Proc. of the 12-th Symposium on Computer Arithmetic*, pp. 84-90, 1995.

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 ~ 1991

년 6월 한국전자통신연구소 반도체연구단 선임연구원. 1991년 7월 ~ 현재 금오공과대학교 전자공학부 부교수. 1995년 8월 ~ 1996년 7월 Univ. of Illinois at Urbana-champaign 방문연구. 주관심분야는 통신 및 신호처리용 집적회로 설계, 저전압/저전력 집적회로 설계, 적외선 센서용 Readout 회로 설계



金 虎 河(正會員)

1974년 8월 17일생. 1997년 2월 금오공과대학교 전자공학과 졸업(공학사). 1998년 현재 금오공과대학교 대학원 석사과정 재학중. 주관심분야는 통신 및 신호처리용 집적회로 설계