

복소수 승산기 코어의 파라미터화된 소프트 IP 설계

양대성*, 이승기*, 정회원 신경욱*

Parameterized Soft IP Design of Complex-number Multiplier Core

Dae-Sung Yang*, Seung-Ky Lee*, Kyung-Wook Shin* *Regular Members*

요약

디지털 통신 시스템 및 신호처리 회로의 핵심 연산블록으로 사용될 수 있는 복소수 승산기 코어의 파라미터화된 소프트 IP (Intellectual Property)를 설계하였다. 승산기는 응용분야에 따라 요구되는 비트 수가 매우 다양하므로, 승산기 코어 IP는 비트 수를 파라미터화하여 설계하는 것이 필요하다. 본 논문에서는 복소수 승산기의 비트 수를 파라미터화 함으로써 사용자의 필요에 따라 승수와 피승수를 8-b~24-b 범위에서 2-b 단위로 선택하여 사용할 수 있도록 하였으며, GUI 환경의 코어 생성기 PCMUL_GEN는 지정된 비트 크기를 갖는 복소수 승산기의 VHDL 모델을 생성한다. 복소수 승산기 코어 IP는 redundant binary (RB) 수치계와 본 논문에서 제안하는 새로운 radix-4 Booth 인코딩/디코딩 회로를 적용하여 설계되었으며, 이를 통해 기존의 방식 보다 단순화된 내부 구조와 고속/저전력 특성을 갖는다. 설계된 IP는 Xilinx FPGA로 구현하여 기능을 검증하였다.

ABSTRACT

This paper describes a parameterized soft IP (Intellectual Property) design of complex-number multiplier (CMUL) core, which can be used as a basic arithmetic unit in ASIC design of digital communication and digital signal processing systems. Since multipliers in various applications require different operand sizes, it is necessary to parameterize the bit-width of multiplier core IP. In this paper, the bit-width of the CMUL is parameterized in the range of 8-b~24-b and is user-selectable in 2-b step. The PCMUL_GEN, a core generator with GUI, generates VHDL code of a CMUL core for a specified bit-width. The CMUL IP was designed using redundant binary (RB) arithmetic and a new radix-4 Booth encoding/decoding scheme proposed in this paper, which result in a simplified internal structure, as well as high-speed/low-power and area-efficient implementation. The designed IP was verified using Xilinx FPGA board.

I. 서론

반도체 제조기술의 급속한 발전에 의해 단일 칩에 백만 게이트 이상의 소자를 집적시킬 수 있게 되었으며, 그 결과 단일 칩에 시스템을 집적시키는 SoC (System-on-Chip) 기술이 비메모리 반도체의 핵심으로 부각되고 있다. SoC급의 반도체는 설계의 복잡성으로 인한 개발 기간과 비용의 증가가 큰 문제점으로 대두되고 있으며, 최근 선진 반도체 업체

들은 설계자산의 재사용이라는 방법을 통해 설계기간을 단축함으로써 비메모리 반도체의 설계 생산성을 향상시키기 위한 시도를 활발히 추진하고 있다. 이와 같이 반도체 설계에 재 사용될 수 있도록 개발된 지적 재산권 (Intellectual Property; IP)을 반도체 설계자산이라 한다. 반도체 IP는 “반도체 디바이스 내에 구현되기 위해 미리 정의된 기능 블록”을 일컬으며, 가상부품 (Virtual Component; VC)으로 불리기도 한다.^[1] 반도체 IP는 아날로그 또는 디지털

* 금오공과대학교 전자공학부 VLSI 설계 연구실 (kwshin@knut.kumoh.ac.kr)

논문번호 : 010127-0530, 접수일자 : 2001년 5월 30일

* 본 연구는 시스템집적반도체기술개발사업의 선행핵심IP개발과제의 지원과 반도체설계교육센터(IDECE)의 CAD Tool 지원으로 수행되었습니다.

털의 물리적 라이브러리, 카운터나 멀티플렉서와 같은 기본 블록, 승산기나 가산기와 같은 연산 코어, 그리고 마이크로 프로세서/컨트롤러와 같은 시스템 레벨 마크로 등 시스템 설계에 재사용이 가능하도록 가공 및 표준화된 설계자산을 포함한다. 일부 기능을 파라미터화하여 사용자 필요에 따라 변형이 가능한 형태도 있다.^[1,2]

디지털 신호처리 및 통신의 응용분야가 확대되면서 복소수 데이터 처리의 필요성이 증대되고 있다. 예를 들어, 고속 푸리에 변환 (Fast Fourier Transform; FFT), QAM (Quadrature Amplitude Modulation), QPSK (Quadrature Phase-Shift Keying), OFDM (Orthogonal Frequency Division Multiplexing) 시스템의 기저대역 신호처리, DS/SS baseband 수신기의 IF 반송파 제거 등을 위해서는 복소수 연산이 필요하며, 복소수 승산기 및 승산-누적기는 이들 시스템을 구성하는 핵심 기능블록으로 사용되어 칩 면적, 동작속도, 전력소모 등의 성능에 중요한 영향을 미친다.

일반적으로, 승산기는 응용분야 또는 시스템에 따라 요구되는 비트 수가 매우 다양하므로, 비트 수가 고정된 승산기 IP는 활용범위가 극히 제한적이다. 따라서, 승수와 피승수의 비트 수를 파라미터화함으로써 사용자의 필요에 따라 원하는 크기로 변형하여 사용할 수 있는 parameterized IP의 개발이 필요하다.

본 연구에서는 디지털 통신 및 신호처리 시스템의 핵심 연산장치로 사용될 수 있도록 고속/고집적/저전력 특성을 갖는 복소수 승산기 IP를 설계하였으며, IP의 활용범위를 극대화하기 위하여 사용자의 필요에 따라 승수와 피승수의 비트 수를 8-b~24-b 범위에서 2-b 단위로 선택할 수 있도록 파라미터화하였다. 복소수 승산 방식은 기존의 2진 수치계 대신에 redundant binary (RB) 수치계를 이용한 복소수 승산방법을 적용하였으며, 새로운 Booth 인코딩/디코딩 방식과 회로구조를 개발하여 적용함으로써 면적과 전력소모를 최소화하였다. 개발된 복소수 승산기 (Parameterized Complex-number Multiplier; PCMUL) 소프트 IP는 전자부품연구원의 "IP 등록 표준안 V3.0"^[2]과 RMM (Reuse Methodology Manual)^[3]의 지침을 만족하도록 설계되었으며, FPGA를 이용한 하드웨어 구현을 통하여 검증하였다.

II. RB 복소수 승산기

1. 실수 승산기를 이용한 복소수 승산

실수부와 허수부가 각각 n-b 2의 보수로 표현된 복소수 $A = A_R + jA_I$ 와 $B = B_R + jB_I$ 의 곱 $Z = A \cdot B$ 는 다음과 같이 표현된다.

$$\begin{aligned} Z &= Z_R + jZ_I \\ &= (A_R B_R - A_I B_I) + j(A_R B_I + A_I B_R) \end{aligned} \quad (1)$$

승산기 구현에 널리 사용되고 있는 radix-4 Booth 인코딩은 승수를 3-b씩 묶어 그 비트 패턴에 따라 부분곱을 생성함으로써 부분곱의 수를 1/2로 줄이는 방법이다. 일반적으로, n-b 승수 Y에 대한 radix-4 Booth 인코딩은 식(2)와 같이 표현된다.

$$\begin{aligned} Y &= \sum_{k=0}^{n/2-1} (-2y_{2k+1} + y_{2k} + y_{2k-1}) \cdot 2^{2k} \\ &= \sum_{k=0}^{n/2-1} Q(k) \cdot 2^{2k} \end{aligned} \quad (2)$$

식(2)에서 $y_{-1}=0$ 이고, $Q(k) \in \{-2, -1, 0, 1, 2\}$ 는 승수 3-비트 묶음의 비트 조합에 따라 정의된다. 식(2)가 나타내는 radix-4 Booth 인코딩을 식(1)에 적용하면, 복소수 승산의 실수부와 허수부는 각각 식(3), 식(4)와 같이 표현된다.

$$\begin{aligned} Z_R &= A_R B_R - A_I B_I \\ &= \sum_{k=0}^{n/2-1} Q_R(k) A_R \cdot 2^{2k} - \sum_{k=0}^{n/2-1} Q_I(k) A_I \cdot 2^{2k} \end{aligned} \quad (3)$$

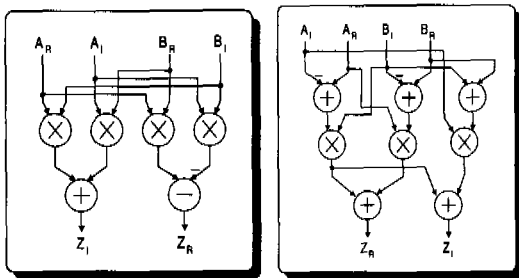
$$\begin{aligned} Z_I &= A_R B_I + A_I B_R \\ &= \sum_{k=0}^{n/2-1} Q_I(k) A_R \cdot 2^{2k} + \sum_{k=0}^{n/2-1} Q_R(k) A_I \cdot 2^{2k} \end{aligned} \quad (4)$$

여기서 $Q_R(k)$ 와 $Q_I(k)$ 는 각각 B_R 와 B_I 에 식(2)의 radix-4 Booth 인코딩을 적용하여 얻어지는 signed-digit (SD) 이다.

일반적인 복소수 승산기 구현 방법은 식(3)과 식(4)를 4개의 실수 승산기와 2개의 실수 가산기를 사용하여 그림 1-(a)와 같이 구현하는 것이며, 이는 총 2n개의 이진 부분곱 가산을 필요로 한다. 이와 같은 고전적인 구현 방법은 기존의 실수 승산기 구현에 관한 많은 연구 결과들을 직접 활용할 수 있다는 장점을 갖는다. 그러나, 실수승산 결과에 대한 가산/감산을 거쳐 최종 결과가 얻어지므로, 가산기의 최하위 비트에서 최상위 비트로 전달되는 캐리 전파 지연에 의해 승산 속도가 느려진다. 또한, 전

체적인 복소수 승산기의 구조가 복잡해지고, 설계 확장 및 변경이 용이하지 않다는 단점이 있다.

한편, 복소수 승산의 연산량을 줄이기 위해 strength reduction 기법이 제안되었으며, 이 경우 그림 1-(b)와 같은 구조로 구현이 가능하다. 이 방법은 그림 1-(a)의 방법과 비교할 때, 1개의 실수 승산기를 줄이는 대가로 3개의 실수 가산기가 추가로 필요하며, 회로 구현시 가산기 및 승산기 사이의 배선이 복잡해지는 단점을 가지며, 또한 실수 승산 전/후의 가산과정에 의한 지연시간 증가와 전체적인 구조의 복잡성에 기인한 설계 확장 및 변경의 어려움 등이 단점으로 지적되고 있다.



(a) direct method (b) strength reduction method

그림 1. 고전적인 복소수 승산 방법

2. RB 수치계를 이용한 복소수 승산^[4]

2의 보수 형태로 표현된 두 수의 합(合) 또는 차(差)를 RB 수치계로 변환하는 과정을 통해 식(3)과 식(4)에 정의된 이진 부분곱을 각각 식(5), 식(6)과 같이 RB 부분곱으로 변환할 수 있다^[4].

$$P_R(k) = \left[\gamma_{k,n}^* 2^n + \sum_{j=0}^{k-1} \gamma_{k,j} 2^j \right] \cdot 2^{2k} \quad (5)$$

$$P_I(k) = \left[\lambda_{k,n}^* 2^n + \sum_{j=0}^{k-1} \lambda_{k,j} 2^j \right] \cdot 2^{2k} \quad (6)$$

식(5), 식(6)에서 $\gamma_{k,j}$ 와 $\lambda_{k,j}$ (단, $\gamma_{k,j}, \lambda_{k,j} \in \{-1, 0, 1\}$)는 k -번째 이진 부분곱의 j -번째 비트로부터 생성된 SD number를 나타낸다. 따라서, n -b 복소수 승산은 RB 수치계로 변환된 부분곱 $P_R(k)$ 과 $P_I(k)$ 를 이용하여 식(7)과 식(8)로 표현된다.

$$Z_R = \sum_{k=0}^{n/2-1} P_R(k) \quad (7)$$

$$Z_I = \sum_{k=0}^{n/2-1} (P_I(k) - 2^{2k}) \quad (8)$$

식(7)과 식(8)은 각각 RB 수치계로 변환된 복소수 승산의 실수부와 허수부를 나타내며, 실수부와 허수부가 각각 $n/2$ 개의 RB 부분곱의 합으로 간략화되었다. 따라서 2진 수치계를 이용하는 식(3), 식(4)의 경우와 비교할 때, 부분곱의 수가 $1/2$ 로 감소되었음을 알 수 있다. 이와 같은 부분곱의 감소는 복소수 승산기 회로의 단순화와 함께 승산 속도의 향상을 가능하게 한다. 또한, 식(7)과 식(8)에는 부호 비트에 대한 정보가 고려되어 있으므로 RB 부분곱 가산 시에 부호 확장을 고려할 필요가 없다는 장점도 갖는다.

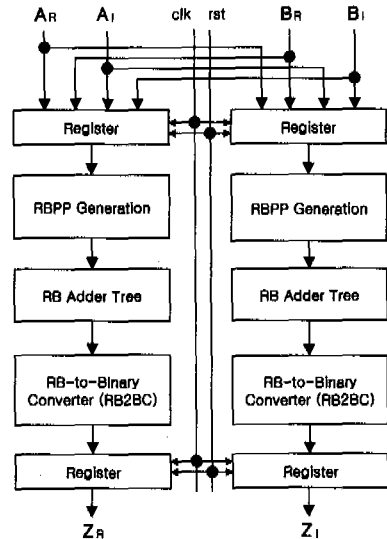


그림 2. RB 복소수 승산기의 구조

3. RB 수치계를 이용한 복소수 승산기 구조

3.1. 전체 구조

설계된 복소수 승산기의 구조는 그림 2와 같으며, 두개의 RB 승산기로 구성된다. 각각의 RB 승산기는 RB 부분곱 생성기, RB 가산기, 그리고 수치계 변환기 등의 기능 블록으로 구성되며, 복소수 연산의 실수부와 허수부 연산을 담당한다. RB 부분곱 생성기는 radix-4 Booth 인코딩을 통해 RB 부분곱과 보정항을 생성하며, 이진 부분곱으로부터 부가적인 하드웨어 없이 직접 RB 부분곱이 생성된다. RB 가산기는 생성된 부분곱과 보정항을 가산하는 복소수 승산 기능을 수행하며, RB 전가산기와 반가산기의 배열로 구성된다. 수치계 변환기는 RB 수치계로 연산된 승산 결과를 2진 수치계로 변환하는 기능을 수행한다.

3.2. 새로운 Booth 인코딩/디코딩 방식을 적용한 RB 부분곱 생성 블록

RB 부분곱 생성 블록은 피승수 A_R , A_I 와 승수 B_R , B_I 를 받아 radix-4 Booth 인코딩을 통해 이진 부분곱 $A_R B_R$, $A_I B_I$, $A_R B_I$, $A_I B_R$ 을 생성한 후, 이들을 signed-digit (SD) 표현으로 변환하여 식(5), 식(6)이 나타내는 RB 부분곱을 생성한다. 본 논문에서는 RB 부분곱 생성 블록의 효율적인 구현을 위해 기존의 방식과 다른 새로운 Booth 인코딩 방식을 고안하여 적용하였다.

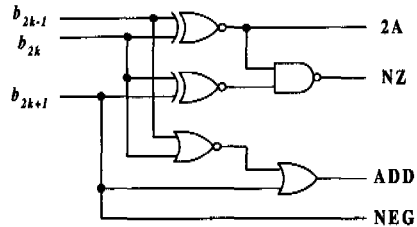
승산기 구현에 널리 사용되고 있는 Booth 알고리즘은 식(2)가 나타내는 바와 같이 승수를 3비트씩 묶어 그 비트 패턴에 따라 부분곱을 생성함으로써 부분곱의 수를 1/2로 줄이는 방법이다. 일반적으로, Booth 인코딩/디코딩 회로는 구현 방법에 따라 승산기의 면적, 전력소모 등에 영향을 미치며, 면적을 최소화하기 위한 compact 방식^[5-6], 전력소모를 줄이기 위한 race-free 방식^[7], 그리고 sign-select 방식^[8] 등이 제안되고 있다.

본 연구에서는 RB 부분곱 생성 블록의 효율적인 구현을 위해 기존의 방식과 다른 새로운 Booth 인코딩/디코딩 방식을 고안하여 IP 설계에 적용하였다. Radix-4 Booth 인코딩을 통해 생성되는 이진 부분곱은 $+A_R B_R$, $+A_R B_I$, $+A_I B_R$ 를 위해 '+'부호를 갖는 경우와 $-A_I B_I$ 를 위해 '-'부호를 갖는 경우로 구분된다. 표 1은 '+'부호를 갖는 이진 부분곱의 생성을 위한 Booth 인코딩 진리표이다. 이로부터 그림 3과 같은 Type-A의 인코딩/디코딩 회로가 얻어지며, 그림 3-(a)는 Booth 인코딩 셀 (BE-A)이고 그림 3-(b)는 디코딩 셀 (BD-A)을 나타낸다. 표 2는 '-'부호를 갖는 이진 부분곱을 생성하기 위한 Booth 인코딩 진리표이다. 표 2로부터 그림 4와 같은 Type-B의 회로가 얻어지며, 그림 4-(a), (b)는 각각 Booth 인코딩 셀 (BE-B)과 디코딩 셀 (BD-B)을 나타낸다.

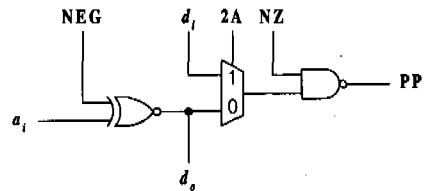
그림 3과 그림 4의 Booth 인코딩/디코딩 셀을 이용한 RB 부분곱 생성 블록의 내부 구성도는 그림 5와 같다. 그림 5-(a)는 복소수 승산의 실수부를 연산하기 위한 RB 부분곱 생성 블록이며, 식(5)가 나타내는 RB 부분곱을 생성한다. 그림 5-(b)는 복소수 승산의 허수부를 연산하기 위한 RB 부분곱 생성 블록이며, 식(6)이 나타내는 RB 부분곱을 생성한다.

표 1. $+A_R B_R$, $+A_R B_I$, $+A_I B_R$ 를 위한 Booth 인코딩 진리표

b_{2k+1}	b_{2k}	b_{2k-1}	$Q(k)$	2A	NZ	NEG	ADD
0	0	0	0	1	0	0	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	2	1	1	0	0
1	0	0	-2	1	1	1	1
1	0	1	-1	0	1	1	1
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	1



(a) Booth encoder cell (BE-A)



(b) Booth decoder cell (BD-A)

그림 3. $+A_R B_R$, $+A_R B_I$, $+A_I B_R$ 를 위한 Booth 인코딩/디코딩 셀 (Type-A)

표 2. $-A_I B_I$ 를 위한 Booth 인코딩 진리표

b_{2k+1}	b_{2k}	b_{2k-1}	$Q(k)$	2A	NZ	NEG	ADD
0	0	0	0	1	0	1	1
0	0	1	-1	0	1	1	1
0	1	0	-1	0	1	1	1
0	1	1	-2	1	1	1	1
1	0	0	2	1	1	0	0
1	0	1	1	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	1	0	0	1

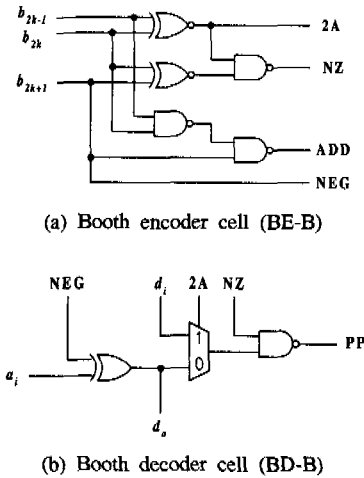


그림 4. $-A_I B_I$ 을 위한 Booth 인코딩/디코딩 셀 (Type-B)

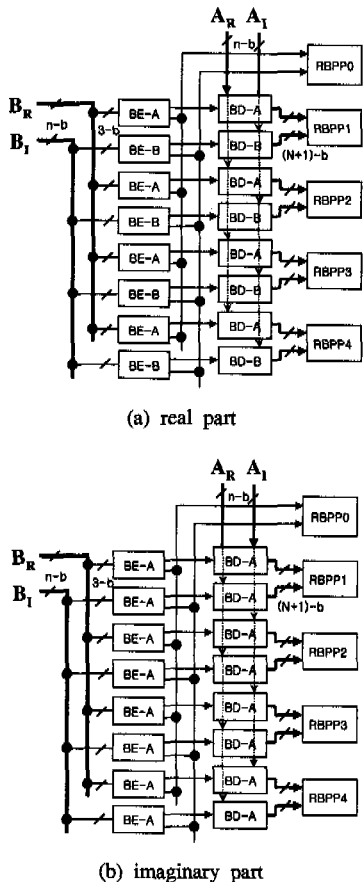


그림 5. 제안된 Booth 인코딩/디코딩 방식에 의한 RB 부분 곱 생성 블록

본 논문에서 제안된 Booth 인코딩/디코딩 방식의 특징은 다음과 같다. 첫째, Booth 인코더에서 디코더

로 연결되는 신호 선을 3개로 줄임으로써 배선에 의한 면적이 최소화된다. 둘째, Booth 인코더에서 신호가 발생하는 순서는 신호 NEG가 가장 먼저 생성되고, 그 다음으로는 신호 2A, 그리고 마지막으로 신호 NZ가 생성된다 (그림 3-(a)와 그림 4-(a) 참조). 따라서, 가장 먼저 생성되는 신호인 NEG를 디코더의 가장 앞단 게이트에 인가하고, 가장 늦게 생성되는 신호인 NZ를 디코더의 출력쪽 게이트에 인가함으로써 신호의 지연경로가 균일화되도록 하였다 (그림 3-(b)와 그림4-(b) 참조). 이는 디코더 출력신호의 불필요한 논리 천이 (spurious transition)를 방지하므로, 전력소모를 최소화하여 저전력 특성을 갖는다.

표 3은 본 논문에서 제안된 Booth 인코딩/디코딩 방식과 기존의 방식을 비교한 것이다. 지연 시간은 버퍼를 제외한 Booth 인코딩/디코딩 회로의 지연을 게이트 단위로 표시한 것이며, 2입력 게이트는 1, 인버터는 0.5, 그리고 3입력 게이트는 1.5로 계산하였다. 면적은 $24-b \times 24-b$ 복소수 승산기에 대해 Booth 인코딩/디코딩 회로의 면적을 계산한 것이다. 표 3에서 보는 바와 같이, 제안된 방식은 기존의 방식에 비해 배선의 복잡도가 낮고, 면적 및 지연 시간이 작으며, 또한 race-free 방식으로 동작하므로 저전력 특성을 가져 고집적/고속/저전력 복소수 승산기 구현에 매우 적합하다.

표 3. Booth 인코딩/디코딩 회로의 비교 ($24-b \times 24-b$ 승산의 경우)

	Compact [5,6]	Race-free [7]	Sign-select [8]	본 논문의 방법
인코더/디코더 연결 신호 수	3	4	4	3
Race-free	×	○	×	○
지연시간 [gate]	5	3.5	3.5	3
면적 [gate] (비율)	4992 (1.3)	8040 (2.1)	3360 (0.9)	3840 (1)

3.3. RB 부분곱 가산 블록

RB 가산기 블록은 RB 부분곱 생성 블록에서 생성된 부분곱을 가산하며, RB 전가산기 (RB Full Adder; RBFA)와 RB 반가산기 (RB Half Adder; RBHA)로 구성된다. RBFA는 두개의 SD 데이터와 캐리 입력을 받아, 그들의 합과 캐리 출력을 생성하는 기능을 수행한다. 문헌 [9]에 언급된 바와 같이, RB 수치계가 갖는 redundancy 특성을 이용하면 ($i-1$)-번째 digit에서 전파된 캐리신호가 i -번째

RBFA를 통해 $(i+1)$ -번째 digit로 전달되지 않도록 RBFA를 정의할 수 있으며, 따라서 RB 부분곱 가산기에 최하위 digit에서 최상위 digit로 캐리신호의 전파가 없는 고속가산을 실현할 수 있다. RBFA 회로는 그림 6과 같다.

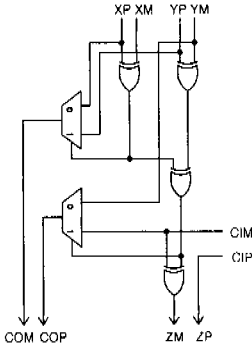


그림 6. RB 가산기 회로도

3.4. RB-to-Binary 변환 블록

RB-to-Binary 변환 블록(RB2BC)은 RB 수치계로 계산된 복소수 승산 결과를 2의 보수 수치계로 변환하는 기능을 수행한다. 이와 같은 수치계 변환은 N-digit RB 데이터를 인코딩하는 두개의 N-b 데이터 A, B의 가산에 의해 이루어지며^[4,9], 따라서 2진 가산기 회로로 구성되고 최하위 비트에서 최상위 비트로 캐리 신호의 전파가 발생된다.

RB2BC 블록의 면적과 속도 최적화를 위해 리플 캐리 가산기 (Ripple-Carry Adder; RCA)와 캐리 선택 가산기 (Carry-Select Adder; CSA)로 구성되는 혼합 구조를 채용하였다. 승산기의 비트 수가 가변 될 수 있으므로, RB2BC의 블록의 구조 (즉, RCA의 비트 수와 CSA의 구성)은 승산기의 출력 비트 수에 따라 서로 달라진다. 예를 들어, 승산기 출력이 33-b인 경우에 RB2BC 블록은 그림 7과 같다. RB 부분곱 가산기에서 출력되는 승산 결과 중 하위 비트가 상위 비트 보다 먼저 출력되므로, 33-b 중 하위 14-b는 RCA 구조를 사용하고 나머지 19-b는 3단 CSA 구조를 사용하여 가산되도록 하였다.

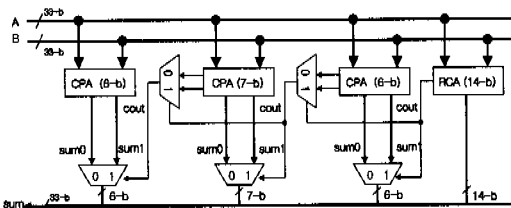


그림 7. RB-to-Binary 변환 블록 (33-b의 경우)

이와 같이 hybrid 구조를 채용함으로써 CSA와 같은 병렬 가산기만을 사용하는 경우 보다 회로를 단순화하여 RB2BC 블록이 차지하는 면적을 최소화할 수 있다.

III. 파라미터화된 복소수 승산기 생성기

본 장에서는 2장에서 설명된 RB 복소수 승산 알고리즘 및 구조를 적용한 파라미터화된 복소수 승산기 (Parameterized Complex-number MULtiplier; PCMUL)의 소프트 IP 구현에 대해 설명한다.

일반적으로, 승산기는 응용분야 또는 시스템에 따라 요구되는 비트 수가 매우 다양하므로, 특정 비트 수의 승산기 IP는 활용가치가 매우 제한적이다. 따라서, 승수와 피승수의 비트 수를 파라미터화 함으로써 필요에 따라 원하는 비트 수로 합성하여 사용할 수 있는 parameterized IP의 개발이 필요하다. 이를 위해, 사용자의 필요에 따라 승수와 피승수의 비트 수를 8-b~24-b 범위에서 2-b 단위로 선택하면 지정된 비트 수를 갖는 복소수 승산기의 VHDL 모델이 자동으로 생성되도록 복소수 승산기 생성기 (PCMUL GENerator; PCMUL_GEN)를 C++ 언어를 사용하여 개발하였다.

PCMUL_GEN에서 복소수 승산기의 VHDL 모델을 생성하는 과정은 그림 8과 같다. 승수 (A_R, A_I)의 비트 수 W_A 와 피승수 (B_R, B_I)의 비트 수 W_B 가 지정되면, 이들 중 작은 값을 승수의 비트 수로, 큰 값을 피승수의 비트 수로 결정한다. 예를 들어, 승수의 비트 수가 $W_A = 12 - b$ 이고, 피승수의 비트 수가 $W_B = 8 - b$ 로 지정되는 경우, 내부 알고리즘에 의해 $W_A = 8 - b, W_B = 12 - b$ 로 승수와 피승수의 비트 수를 교환한다. 이는 승수와 피승수를 서로 바꾸도 승산 결과는 동일하다는 사실을 기반으로 한다. 잘 알려진 바와 같이, radix-4 Booth 인코딩을 이용하는 승산기에서 생성되는 부분곱의 수는 $\frac{1}{2} W_A$ 이 되므로, 승수의 비트 수를 작은 값으로 취하면 부분곱 가산기 트리의 depth가 최소화되어 승산 속도를 향상시킬 수 있다.

승수와 피승수의 비트 수가 확정되면, 미리 개발된 VHDL Component Library로부터 RB 부분곱 생성 블록, 부분곱 가산 블록 및 수치계 변환 블록 등에 필요한 VHDL Component들을 조합하여 복소수 승산기의 VHDL 모델이 생성된다. PCMUL_GEN은 사용자의 필요에 따라 $(8-b \times 8-b) \sim (24-b \times$

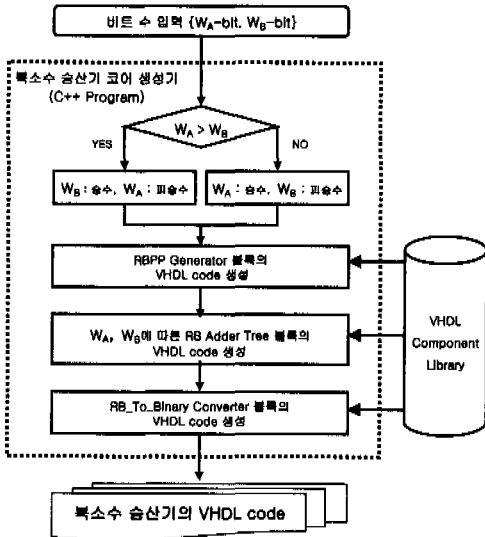


그림 8. PCMUL_GEN의 흐름도

24-b)의 비트 수 범위에서 45가지 종류의 복소수 승산기의 VHDL 코드를 자동으로 생성할 수 있다. 그림 9는 PCMUL_GEN의 실행 화면이며, (16-b x 16-b) 복소수 승산기 코어를 생성하는 화면이다.

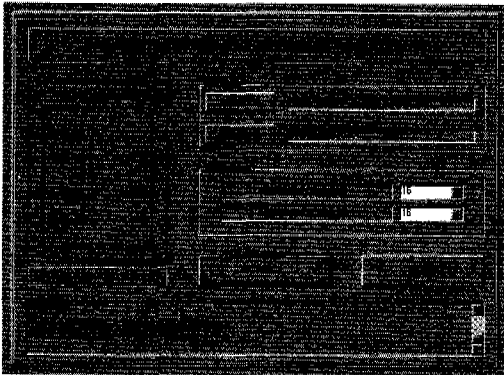


그림 9. PCMUL_GEN을 이용한 CMUL 코어의 생성

IV. IP 검증

본 장에서는 PCMUL_GEN에서 생성되는 복소수 승산기 VHDL 모델의 검증에 대해 기술한다. 승수와 피승수의 비트 수에 따라 PCMUL_GEN에서 생성되는 45가지 종류의 복소수 승산기를 모두 검증하였으며, 시뮬레이션을 통한 검증과 FPGA (Field Programmable Gate Array) 디바이스를 이용한 하드웨어 구현 검증을 실시하였다. 또한, Synopsys를 이용한 논리합성을 통해 45가지 복소수 승산기의

게이트 수를 추정하여 IP의 사용자에게 유용한 정보가 되도록 하였다.

C 언어 모델링으로부터 10,000개의 랜덤 테스트 벡터를 생성하고, 이를 VHDL Testbench에서 읽어 논리 시뮬레이션을 수행하였다. 시뮬레이션 결과와 C 언어 모델링으로 생성된 결과를 비교하여 복소수 승산기 VHDL 모델의 논리동작을 검증하였다. 시뮬레이션 결과, PCMUL_GEN으로 생성된 45가지 복소수 승산기가 모두 올바르게 동작함을 확인하였다. 그림 10은 Synopsys를 이용한 시뮬레이션 결과의 일부이다.

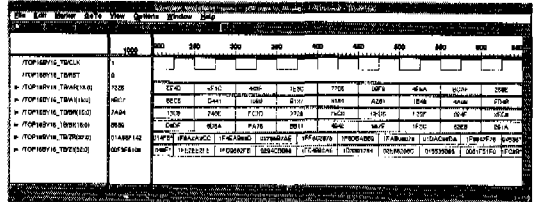


그림 10. 설계된 PCMUL IP의 시뮬레이션 결과 (16-b x 16-b 승산기의 경우)

그림 11은 개발된 IP의 하드웨어 검증 시스템을 보인 것이며, FPGA 보드, PC 및 ISA 인터페이스 보드, 구동 소프트웨어 등으로 구성된다. FPGA는 Xilinx XCV1000E를 사용하였으며, Visual C++ 언어로 테스트 프로그램을 작성하였다.

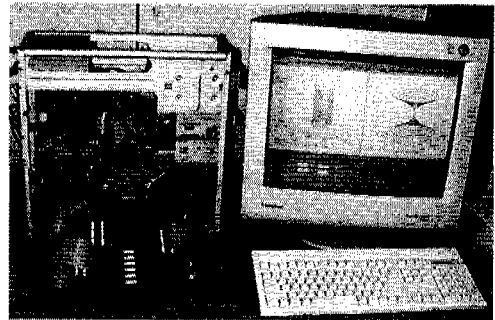


그림 11. PCMUL IP의 검증 시스템 사진

그림 12는 PCMUL IP 검증 시스템의 실행 화면이다. 그림 12-(a)에서 화면의 왼쪽은 승산기에 입력되는 승수 (크기를 고정하고 위상을 변화시킨 원통 모양의 벡터 집합)와 피승수 (위상을 고정하고 크기를 변화시킨 포물선 모양의 벡터 집합)를 나타내는 1,500개의 복소수 벡터이며, 오른쪽은 승산 결과로 얻어진 복소수 벡터로서 크기와 위상이 함께 변하는 Tornado 모양을 나타내고 있다. 그림 12-(b)

는 화면 왼쪽의 승수 벡터 성장도와 $1 + j1$ 로 고정된 피승수 벡터의 승산을 보인 것이며, 승수 벡터 성장도가 45° 회전된 결과가 출력됨을 보이고 있다. 이와 같은 FPGA 구현 검증을 통해 개발된 IP가 정상적으로 동작함을 확인하였다.

PCMUL_GEN에 의해 생성되는 45종류의 복소수 승산기를 Synopsys와 0.25- μm 표준 셀 라이브러리를 이용한 논리 합성을 통하여 게이트 복잡도를 분석하였다. 그림 13은 승수와 피승수의 비트 수에 따른 복소수 승산기의 게이트 복잡도를 나타낸 것이다.

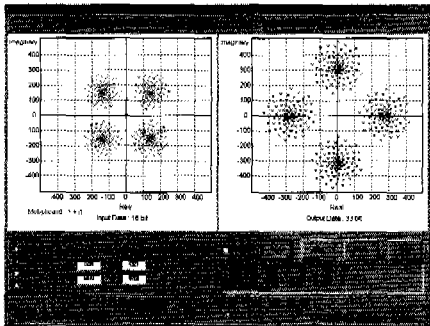
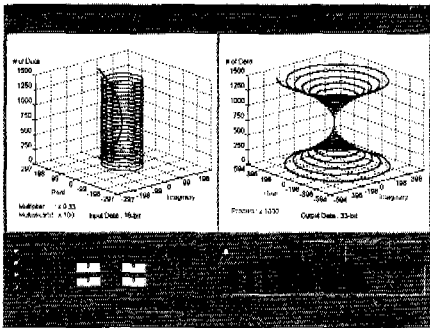


그림 12. PCMUL IP의 검증 시스템 실행 화면

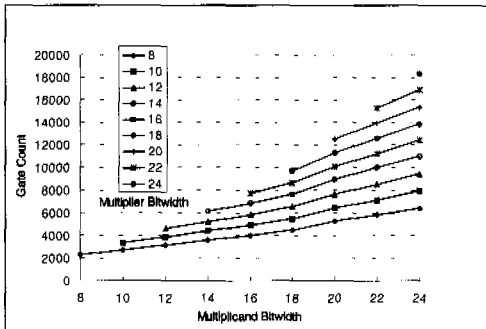


그림 13. 승수와 피승수의 비트 수에 따른 게이트 복잡도

예를 들어, (8-b \times 8-b)의 경우 2,280 게이트, (16-b \times 16-b)의 경우 7,680 게이트, 그리고 (20-b \times 20-b)의 경우 12,520 게이트로 구현된다.

V. 결론

승산기는 응용분야에 따라 요구되는 비트 수가 매우 다양하므로, 특정 비트 수를 갖는 승산기 IP는 활용 범위가 매우 제한적이다. 본 논문에서는 복소수 승산기 IP의 활용 범위를 극대화하기 위하여 승수와 피승수의 비트 수를 사용자의 필요에 따라 8-b~24-b 범위에서 2-b 단위로 선택할 수 있도록 파라미터화 하였다. 또한, GUI 환경의 복소수 승산기 생성기 (PCMUL_GEN)를 개발하여 승수와 피승수의 비트 수를 8-b~24-b 범위에서 2-b 단위로 선택하면 해당 비트 수를 갖는 복소수 승산기의 VHDL 모델이 생성되도록 하였다. PCMUL_GEN은 총 45 종류의 복소수 승산기에 대한 VHDL 모델을 생성한다.

개발된 복소수 승산기 코어 IP는 실수 승산기를 이용하는 고전적인 방법 대신에 RB 수치제와 Booth 알고리즘을 혼합한 방식을 적용하여 설계되었으며, 또한 새로운 Booth 인코딩/디코딩 방식 및 회로를 개발하여 적용함으로써 기존의 방식에 비해 배선의 복잡도, 면적, 지연시간이 적은 특성을 가지며, race-free 방식으로 동작하여 저전력 특성을 갖도록 하였다. 개발된 복소수 승산기 IP는 Synopsys를 이용한 논리합성과 시뮬레이션을 통해 검증하였으며, Xilinx FPGA 구현을 통해 검증하였다.

개발된 IP는 DS/SS 기저대역 수신기의 IF 반송파 제거, ADSL 수신기의 복소수 필터, QAM/QPSK 시스템의 적응 결정위환 동기화, 고속 푸리에 변환 (FFT), 벡터 회전 연산 등 각종 디지털 신호처리 및 디지털 통신 시스템의 회로 설계에 폭넓게 이용 가능할 것이다.

참고 문헌

- [1] 임영이, 이재환, "반도체 IP 기술 및 시장동향", <http://etlars.etri.re.kr>, 한국전자통신연구원, 2001.
- [2] "IP Standard Specification V3.0", 전자부품연구원, 2001. 3. 19.
- [3] M. Keating and P. Bricaud, *Reuse Methodology Manual for System-On-A-Chip Designs*, Kluwer Academic Publishers, 1998.

[4] K.W. Shin, H.W. Jeon, "High-speed complex-number multiplications based on redundant binary representation of partial products", *Int. Journal of Electronics*, vol. 87, no. 6, pp. 683-702, 2000. 6.

[5] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54×54-b regularly structured tree multiplier", *IEEE J. of Solid-State Circuits*, vol. 27, no. 9, pp. 1229-1236, Sep. 1992.

[6] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, "A 4.4-ns CMOS 54×54-b multiplier using pass-transistor multiplexer", *IEEE J. of Solid-State Circuits*, vol. 30, no. 3, Mar. 1995.

[7] R. Frid, "Minimizing energy dissipation in high-speed multipliers", *International Sym. on Low-Power Electronics and Design*, pp. 214-219, 1997.

[8] G. Goto, A. Inoue, R. Ohe, S. Kashiwakura, S. Mitarai, T. Tsuru, and T. Izawa, "A 4.1-ns compact 54×54-b multiplier utilizing Sign-select Booth encoders", *IEEE J. of Solid-State Circuits*, vol. 32, no. 11, Nov. 1997.

[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.

이 승 기(Seung-Ky Lee)

비회원



2001년 2월 : 금오공과대학교
전자공학과 졸업
2001년 3월~현재 :
금오공과대학교
전자공학과
석사과정 재학 중

<주관심 분야> 통신 및 신호처리용 집적회로 설계,
로봇 제어

신 경 욱(Kyung-Wook Shin)

정회원



1984년 2월 : 한국항공대학교
전자공학과 졸업
1986년 2월 : 연세대학교 대학원
전자공학과 (공학석사)
1990년 8월 : 연세대학교 대학원
전자공학과 (공학박사)

1990년 9월~1991년 6월 : 한국전자통신연구소
반도체연구단 (선임연구원)

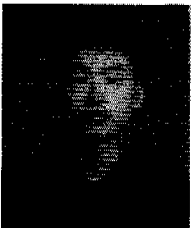
1991년 7월~현재 : 금오공과대학교 전자공학부
(부교수)

1995년 8월~1996년 7월 : University of Illinois
at Urbana-Champaign (방문교수)

<주관심 분야> 통신 및 신호처리용 집적회로 설계,
암호 프로세서 설계, 반도체 IP 설계

양 대 성(Dae-Sung Yang)

비회원



2000년 2월 : 금오공과대학교
전자공학과 졸업
2000년 3월~현재 :
금오공과대학교
전자공학과
석사과정 재학 중

<주관심 분야> 통신 및 신호처리용 집적회로 설계,
암호 프로세서 설계