ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 21-04-08
j.inst.Korean.electr.electron.eng.Vol.25,No.4,634~643,December 2021
https://doi.org/10.7471/ikeee.2021.25.4.634
51

실시간 SAR 영상 생성을 위한 Range Doppler 알고리즘의 FPGA 기반 가속화

FPGA-Based Acceleration of Range Doppler Algorithm for Real-Time Synthetic Aperture Radar Imaging

정 동 민*, 이 우 경**, 정 윤 호*, **★

Dongmin Jeong*, Wookyung Lee**, Yunho Jung*, **★

Abstract

In this paper, an FPGA-based acceleration scheme of range Doppler algorithm (RDA) is proposed for the real time synthetic aperture radar (SAR) imaging. Hardware architectures of matched filter based on systolic array architecture and a high speed sinc interpolator to compensate range cell migration (RCM) are presented. In addition, the proposed hardware was implemented and accelerated on Xilinx Alveo FPGA. Experimental results for 4096×4096-size SAR imaging showed that FPGA-based implementation achieves 2 times acceleration compared to GPU-based design. It was also confirmed the proposed design can be implemented with 60,247 CLB LUTs, 103,728 CLB registers, 20 block RAM tiles and 592 DPSs at the operating frequency of 312 MHz.

요 약

본 논문에서는 실시간 SAR (synthetic aperture radar) 영상 생성을 위한 RDA (range Doppler algorithm)의 FPGA 기반 가속화 기법을 제안한다. RDA의 연산 과정인 거리 및 방위 압축 연산을 가속하기 위한 시스토릭 어레이 구조 기반 정합 필터와 RCM (range cell migration)을 보상해 주기 위한 고속의 sinc 보간 연산기의 하드웨어 구조를 제시하고, Xilinx Alveo FPGA에 다채널 커널 형태로 구현하여 가속을 진행하였다. 제안된 구조의 하드웨어를 사용하여 4096×4096 크기의 영상 생성 시간을 측정한 결과, Nvidia RTX3090 GPU를 사용하여 SAR 영상을 생성하는 시간보다 약 2배 가속이 가능함을 확인하였다. 또한, 제안된 가속 하드웨어는 60,247개의 CLB LUT, 103,728개의 CLB register, 20개의 block RAM tile과 592개의 DPS로 구현 가능하며, 최대 동작속도는 312 MHz임을 확인하였다.

Key words : FFT, FPGA, radar, RDA, SAR, signal processing, systolic array

* Department of Smart Air Mobility, Korea Aerospace University
** School of Electronics and Information Engineering, Korea Aerospace University
★ Corresponding author
E-mail : yjung@kau.ac.kr, Tel : +82-2-300-0133

※ Acknowledgment
The authors gratefully acknowledge the support from Next Generation SAR Research Laboratory at Korea Aerospace University, originally funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).
Manuscript received Nov. 29, 2021; revised Dec. 2, 2021; accepted Dec. 7, 2021.

Ⅰ. 서론

SAR(synthetic aperture radar)는 군사 및 민간 분야에서 지상 및 해양에 대해 공중에서 전파를 송/수신하여 지표를 관측하는 레이다 시스템으로, 작은 안테나의 크기로 높은 방위해상도를 얻을 수 있고, 전파를 사용하기 때문에 주/야간 및 비, 구름, 안개 등에 관계없이 영상 획득이 가능하다는 장점을 갖고 있어, 지속적인 연구가 활발하게 진행되고 있다[1].

SAR를 통해 대상을 관측할 때 신속한 응답을 위해서는 실시간으로 SAR 영상을 획득하는 것이 중요하다. 그러나, 실제로 실시간 SAR 영상 생성의 어려운 문제 중 하나는 많은 양의 원시 에코 데이터에서 최종 이미지를 생성하기 위한 엄청난 양의 신호처리이다. 전파를 송신하고 반사되어 돌아와 저장되는 에코 데이터를 이용해, 거리 방향과 방위 방향으로의 FFT(fast Fourier transform)와 IFFT (inverse fast Fourier transform)의 연산을 수행하고, 항공기나 위성에 탑재되어 이동하면서 얻어지는 데이터의 특성상 이미지의 왜곡 현상을 보정하는 연산도 필요하기 때문에, 원시 데이터가 커지면 연산에 걸리는 시간이 매우 크게 증가하기 때문이다.

신호처리 관점에서의 효율성 향상은 제한적이기 때문에, SAR 영상 생성을 가속하기 위한 아이디어에는 GPU(graphic processing unit)를 사용하거나 FPGA(field-programmable gate array)를 사용하여 가속하는 방법이 있다[2], [3]. GPU는 많은 코어를 활용한 높은 병렬성, 멀티 스레딩, 큰 대역폭 등의 특성이 있지만 여전히 큰 전력소모는 무시할 수 없다.

FPGA는 범용 CPU(center processing unit) 및 GPU와 비교할 때 컴퓨팅 집약적인 어플리케이션을 구현하기 위한 처리량, 짧은 지연 시간 및 에너지 효율성 면에서 상당한 이점을 제공한다. 수년 동안 FPGA의 급속한 발전에 따라 강력한 처리 능력을 갖춘 FPGA가 널리 사용되고 있고[4], 온 칩 CPU, 도메인별 프로그래밍 가능 가속기 및 여러 연결 옵션을 갖춘 SoC 형태로 발전했다.

최근에는 Xilinx의 Alveo FPGA 카드, Intel의 PAC 카드 같은 하드웨어 가속기 카드가 출시되었고, 이와 같은 FPGA 기반 하드웨어 가속기 카드는 GPU에 비해 훨씬 낮은 전력을 소비하는 것과 동시에, 풍부한 온 칩 메모리와 계산 리소스를 통해 높은 처리량을 가지며, 다채널 형태로 구성이 가능해서 GPU처럼 강력한 병렬 처리가 가능하다. 일반적인 목적으로 사용되는 GPU와는 달리, 특정 어플리케이션에 맞추어 최적화를 진행하기 용이하기 때문에 특정한 동작을 요구하는 상황에서는 GPU보다 더 빠른 연산속도를 보여준다[3]. 또한, 부동 소수점을 사용하는 GPU와는 다르게 고정 소수점을 사용하는 FPGA는 적은 규모로 적절한 정밀도를 만족시킬 수 있다. 이러한 이유로, 본 논문에서는 Xilnx Alveo FPGA 카드를 사용하여 SAR 영상 생성을 위한 고속의 하드웨어 프로세서를 제시하고, 다채널 형태로 구성하여 실시간 SAR 영상 생성을 가속화한다.

SAR 영상 생성 기술의 발전으로 많은 알고리즘들이 실제로 적용되고 있다[5-7]. 이중 RDA (range Doppler algorithm)는 물리적인 개념이 직관적이어서, 이전부터 SAR 영상 생성 알고리즘으로 널리 사용되었고, 영상 품질과 계산 효율성의 교환관계가 가장 좋기 때문에 실시간 SAR 영상 생성을 위한 알고리즘으로 적합하다[5], [8].

RDA는 거리 압축, RCMC(range cell migration correction), 방위 압축의 연산으로 구성되며, 거리/방위 압축 연산은 FFT, 곱셈, IFFT로 구성된다. 따라서 실시간 영상 생성을 위해서는 거리/방위 압축 연산을 빠르게 하기 위해 고속의 FFT 프로세서를 설계하는 것이 매우 중요하다. 또한, RCMC 연산은 거리 방향으로 연산 복잡도가 높은 sinc 보간 연산을 수행하여 진행되기 때문에 고속의 sinc 보간 프로세서를 설계할 필요가 있다.

FFT 프로세서의 하드웨어 구조에는 단일 버터플라이 구조[9], 파이프라인 구조[10], 시스토릭 어레이 구조[11]로 구분된다. 단일 버터플라이 구조와 파이프라인 구조는 작은 면적으로 구현 가능하지만, 실시간 SAR 영상 생성과 같이 고속의 연산 속도가 필요한 응용에 적합하지 않다. 따라서 위 하드웨어 구조들 중 실시간 SAR 영상 생성 응용의 경우, 속도가 가장 빠른 시스토릭 어레이 구조 기반의 병렬구조가 가장 적합하다[12], [13].

다양한 시스토릭 어레이 구조들 중에서 base-4 기반의 시스토릭 어레이 구조가 구현의 용이성과 확장성이 좋고 면적과 수행시간의 교환관계를 가장 잘 만족하기 때문에[14], [15], base-4 기반의 시

스토릭 어레이 구조 기반 FFT 프로세서를 설계하여, 이를 기반으로 정합 필터를 설계하고, 고속의 RCMC 프로세서를 설계한 뒤, 이를 BUS 인터페이스를 포함한 Xilnx Alveo FPGA 카드에 다채널/병렬 형태로 구성하여 고속의 RDA 기반 SAR 영상 생성을 위한 하드웨어 가속기를 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 RDA와 base-4 기반 시스토릭 어레이 구조 기반 FFT 알고리즘에 대해 설명하고, 3장에서는 제안하는 RDA 가속화하기 위한 FFT 프로세서와 정합필터 및 RCMC의 하드웨어 구조에 대해 설명한다. 4장에서는 제안된 프로세서의 설계 및 구현과 Xilnx Alveo FPGA로 가속한 RDA의 결과를 제시하며, 비교 및 분석한다. 5장에서는 본 논문의 주요 결론을 요약하여 마무리한다.

II. Background

1. Range-Doppler algorithm

RDA는 SEASAT(sea satellite) SAR 데이터를 처리하기 위해 1976~1978년 사이에 개발되었다. 최초의 디지털 연산으로 처리된 우주 기반 SAR 이미지는 1978년에 이 알고리즘을 기반으로 만들어졌고, 오늘날에도 널리 사용되고 있다. 거리와 방위 모두 주파수 영역에서 연산을 진행하여 효율적인 연산 처리를 위해 설계되었다. 방위 주파수는 도플러 주파수와 동의어이고, RCMC가 방위 주파수 영역에서 수행되는 것이 이 알고리즘의 특징이기 때문에 RDA라고 불린다.

RDA는 구현 효율성을 위해 정합 필터 컨볼루션 연산을 주파수 도메인에서 곱셈으로 수행한다. 또한, 거리 방향의 처리와 방위 방향의 처리가 분리되어 진행될 수 있기 때문에 각각의 연산을 1차원으로 수행하여 처리가 간단하고 효율적이다.

RDA의 연산 과정은 그림 1처럼 크게 거리 방향 압축, 방위 방향 압축 과정으로 진행되는데, 방위 방향 압축 과정에는 RCMC 연산이 포함된다. 먼저 LFM(linear frequency modulation) 첩 신호를 송신하여 표적에 맞고 반사되어 돌아오는 수신 신호는 기저대역으로 복조되어 반송 주파수가 제거된 다음의 식 (1)로 표현할 수 있다.

$$s_0(\tau,\eta) = A_0 w_r\left[\tau - \frac{2R(\eta)}{c}\right] w_a(\eta-\eta_c) \cdot e^{-j\frac{4\pi f_0 R(\eta)}{c}} e^{j\pi K_r\left(\tau-\frac{2R(\eta)}{c}\right)^2} \quad (1)$$

식 (1)에서 A_0는 수신 신호의 크기, τ는 거리 시간, η는 방위 시간, η_c는 빔 센터 크로싱 시간(beam center crossing time), $w_r(\tau)$는 거리 시간으로의 포락선(envelope)으로 구형파로 근사화 가능하며, $w_a(\eta)$는 방위 시간으로의 포락선으로 sinc 함수로 근사가 가능하다. f_c는 중심 주파수, K_r는 거리 방향 주파수 변조율(range FM rate), $R(\eta)$는 레이다와 타겟 사이의 경사 거리(slant range)를 의미한다. 이후, 이렇게 수신된 신호 $s_0(\tau,\eta)$를 푸리에 변환하여 $S_0(f_\tau,\eta)$로 변환한 뒤 거리 참조 신호의 주파수 도메인 식인 $G(f_r)$을 곱하고, 역 푸리에 변환을 수행하여 거리 압축 과정을 진행하며, 식 (2)로 정리된다.

$$\begin{aligned} s_{rc}(\tau,\eta) &= IFFT\{S_0(f_\tau,\eta)\,G(f_\tau)\} \\ &= A_0 p_r\left[\tau - \frac{2R(\eta)}{c}\right] w_a(\eta-\eta_c) e^{-j\frac{\pi f_0 R(\eta)}{c}} \end{aligned} \quad (2)$$

식 (2)에서 p_r은 sinc-like 거리 포락선이며 내부의 $2R(\eta)/c$는 방위 시간에 대한 함수로 표적의 RCM에 영향을 끼친다. 이후, 방위 압축 과정을 진행해야 하는데, 항공기나 위성에 탑재된 레이더의 이동에 의해 관측하고자 하는 지표로부터 레이더 사이의 거리가 변하게 되는데, 이를 보정해주기 위해 sinc 보간을 사용하여 RCMC 연산이 수행된다. RDA는 RCMC 연산을 방위 주파수 영역에서 수행하는 것이 특징이기 때문에 거리 압축이 완료된 신호를 주파수 도메인으로 변환하는 푸리에 변환을 수행해야 한다. 이렇게 얻어지는 식은 식 (3)으로 정리된다.

$$S_1(\tau,f_\eta) = A_0 p_r\left[\tau - \frac{2R_{rd}(f_\eta)}{c}\right] W_a(f_\eta - f_{\eta_c}) \cdot e^{-j\frac{4\pi f_0 R_0}{c}} e^{j\pi\frac{f_\eta^2}{K_a}} \quad (3)$$

다음으로는, 이미지의 보정을 위한 과정인 RCMC 과정을 진행하게 된다. 이는 거리 방향으로 sinc 보간을 수행함으로써 진행되고, RCMC를 수행한 뒤 방위 참조 신호를 곱해주게 되는데, 방위 참조 신호는 식 (4)와 같이 표현된다.

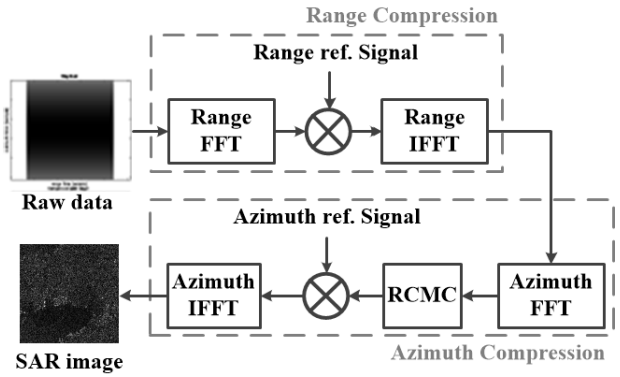


Fig. 1. RDA operation flow.
그림 1. RDA 연산 흐름

$$H(f_\eta)=e^{-j\frac{f_\eta^2}{K_a}} \tag{4}$$

따라서, 식 (3)의 마지막 항과 곱해져 1이 되고, IFFT를 수행하면 최종적으로, 다음과 같이 식이 구해지게 되며 SAR 영상을 생성할 수 있다.

$$s_{ac}(\tau,\eta)=A_0 p_r\left(\tau-\frac{2R_0}{c}\right)p_a(\eta)e^{-j\frac{4\pi f_0 R_0}{c}}e^{j2\pi f_{\eta_c}\eta} \tag{5}$$

이러한 과정을 통해 알 수 있듯이 RDA의 대부분의 시간은 FFT 연산에서 소요되기 때문에 실시간 SAR 영상 생성을 위해서는 고속의 FFT 프로세서를 사용할 필요성이 있다.

2. Base-b FFT algorithm

N 길이의 DFT(discrete Fourier transform)는 식 (6)과 같이 정의된다.

$$Z(k)=\sum_{n=0}^{N-1}W_N^{nk}X(n),\ W_N^{nk}=e^{-j\frac{2\pi nk}{N}} \tag{6}$$

식 (6)에서 W_N^{nk}는 회전인자(twiddle factor)라 불리고, $X(n)$은 시간 영역에서의 입력값을 의미하며, $Z(k)$는 주파수 영역의 출력값을 의미한다. $N=N_1N_2$로 분해할 수 있다면, n과 k는 각각 식 (7)과 같이 나타낼 수 있다.

$$\begin{aligned} n&=n_1+N_1n_2,\ (0\le n_1,k_1\le N_1-1)\\ k&=k_1+N_1k_2,\ (0\le n_2,k_2\le N_2-1)\end{aligned} \tag{7}$$

이후, 식 (7)을 DFT 정의식인 식 (6)에 대입하여 식(8)과 같이 정리할 수 있다.

$$\begin{aligned} &Z(k_1+N_1k_2)\\ &=\sum_{n_1=0}^{N_1-1}\left(W_N^{n_1k_1}\sum_{n_2=0}^{N_2-1}W_{N_2}^{n_2k_1}X(n_1+N_1n_2)\right)W_{N_2}^{n_1k_2}\end{aligned} \tag{8}$$

이때, 식 (8)에서 괄호 내부 연산 결과의 값을 $Y(k_1, n_1)$라 하고, 시그마 연산을 행렬 연산으로 풀어서 나타내면 식 (9)와 같이 나타낼 수 있다.

$$\begin{aligned} &Y(k_1,n_1)=\\ &W_N^{n_1k_1}[W_{N_2}^0\ W_{N_2}^{k_1}\cdots W_{N_2}^{(N_2-1)k_1}]\times\begin{bmatrix}X(n_1)\\X(n_1+N_1)\\\vdots\\X(n_1+(N_2-1)N_1)\end{bmatrix}\end{aligned} \tag{9}$$

그 후, 식(5)의 전체 시그마 연산을 식 (8)에서 구한 $Y(k_1,n_1)$을 사용하면 다음의 식 (10)과 같이 행렬 연산으로 나타낼 수 있다.

$$\begin{aligned} &Z(k_1+N_1k_2)=\\ &[W_{N_2}^0\ W_{N_2}^{k_2}\cdots W_{N_2}^{(N_2-1)k_2}]\times\begin{bmatrix}Y(k_1,0)\\Y(k_1,1)\\\vdots\\Y(k_1,N_1-1)\end{bmatrix}\end{aligned} \tag{10}$$

최종적으로, 앞서 구한 식 (9)와 식 (10)의 과정을 각각의 행렬식으로 표현하면 식 (11)로 나타낼 수 있다.

$$\begin{aligned} Y&=W_M\bullet C_{M1}X\\ Z&=C_{M2}Y^t\end{aligned} \tag{11}$$

여기서, $W_M=W_N^{n_1k_1}$이며, $N_1\times N_1$ 행렬이고, $\bullet$은 요소별 곱셈, $C_{M1}=W_{N_2}^{n_2k_1}$이며, $N_1\times N_2$ 행렬이다. X는 $N_2\times N_1$ 행렬이기 때문에 Y는 $N_1\times N_1$ 행렬이 된다. $C_{M2}=W_{N_2}^{n_1k_2}$이고, 이는 C_{M1}^t와 같다. 따라서, $N_2\times N_1$ 행렬이 되며 Z는 $N_2\times N_1$ 행렬이 된다. Base-4 FFT 알고리즘에서 $N_2=4$로 고정되며, 이는 응용에 따라 바뀔 수 있다.

Base-4 FFT는 N 길이의 1차원 데이터를 총 2단계의 부분 분리(factorization)과정을 사용해 수행된다. 첫 번째 부분 분리는 행과 열로 분리하여 $N=N_rN_c$가 되며, 이때 N_r은 행의 길이, N_c는 열의 길이가 된다. 이후, 두 번째 부분 분리는 $N_r=N_{1r}N_2$와 $N_c=N_{1c}N_2$로 한 번 더 분리가 되며, 식 (11)을 이용해 FFT를 진행하게 된다. 총 3단계로 진행되며, 먼저, N_c 길이의 열 데이터를 사용하여 식 (11)의 FFT 연산을 행의 방향으로 N_r번 수행한다. 이

후, 각 요소에 W_N을 곱해준 뒤, 마지막으로, N_r 길이의 행 데이터를 사용하여 FFT 연산을 열의 방향으로 N_c번 수행한다. 즉, 1차원 데이터를 $N_r \times N_c$의 2차원 행렬로 바꾼 뒤 열 FFT, W_N 곱셈, 행 FFT의 3단계로 진행되어 최종 결과가 도출된다 [15].

III. 제안된 프로세서 하드웨어 구조

RDA를 수행하기 위해 거리 압축 및 방위 압축 연산이 필요하며, 이는 정합 필터를 사용하여 구현된다. 정합 필터 연산은 FFT, 참조 신호 곱셈, IFFT 연산으로 구성되며, 시스토릭 어레이 구조 기반 FFT 프로세서를 사용해 구현되고, RCMC 프로세서와 함께 AXI(advanced extensible interface) BUS를 포함하여 그림 2와 같이 구성된다.

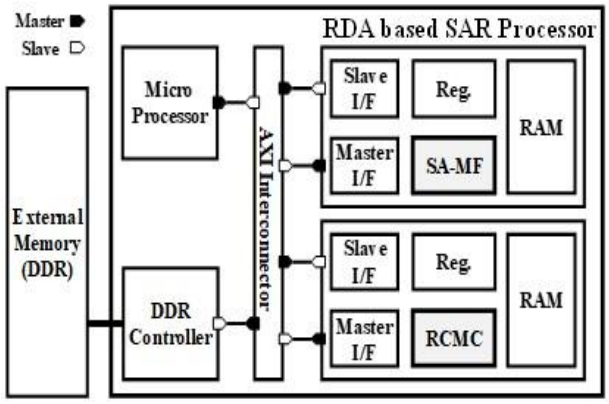


Fig. 2. Hardware configuration for RDA acceleration.
그림 2. RDA 가속을 위한 하드웨어 구성

1. 시스토릭 어레이 구조 기반 FFT 프로세서

시스토릭 어레이 구조는 아래 그림 3과 같이 각각의 PE(process element) 셀이 지역적으로 연결되어 각각의 PE 셀에서 연산을 수행하고 연결된 PE 셀에 데이터를 전달하는 형태로 구성된다. 규칙적이고, 지역적인 데이터 흐름을 갖고, 여러 개의 PE 셀이 동시에 연산을 처리하기 때문에 많은 연산이 요구되는 알고리즘에 적합하다[12].

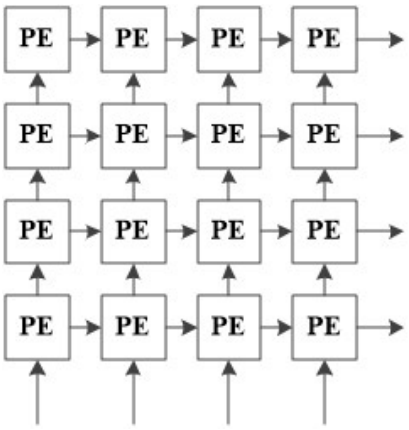


Fig. 3. Example of systolic array architecture.
그림 3. 시스토릭 어레이 구조 예시

시스토릭 어레이 구조를 사용하면, 행렬 연산을 빠르게 처리할 수 있다. 아래에서 입력이 들어오면, 각 PE 셀에서 곱셈과 덧셈 연산을 수행한 뒤 연결된 PE 셀로 값을 전달하며, 모든 PE 셀을 지나게 된다. A 행렬의 입력이 아래에서 순차적으로 들어온다 할 때, PE 셀 내부에 B 행렬값이 존재하게 되

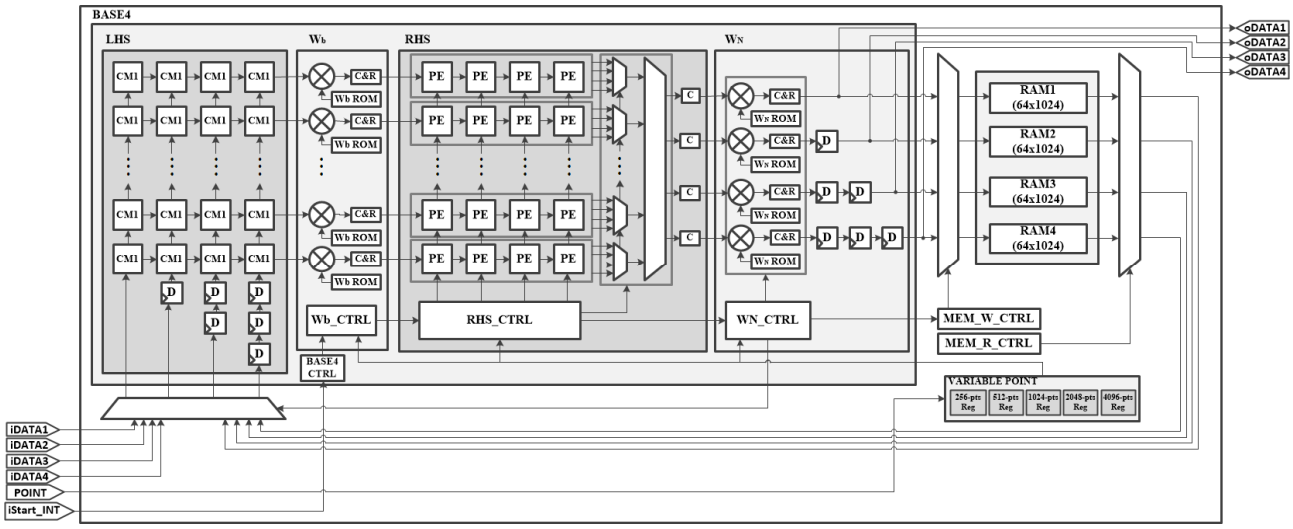


Fig. 4. Proposed FFT processor architecture based on systolic array architecture.
그림 4. 제안된 시스토릭 어레이 구조 기반 FFT 프로세서 구조

면, $B \times A$의 행렬값을 출력하게 된다. 행렬의 곱으로 정리되어 있는 식 (11)을 사용해서 FFT 연산을 진행할 때, 행렬 연산을 빠르게 수행할 수 있어, 고속으로 FFT를 처리할 수 있다.

제안된 $N=N_rN_c$ 길이의 FFT를 수행하는 시스토릭 어레이 구조 기반 FFT 프로세서의 하드웨어는 'base-b'에서 하드웨어 복잡도와 처리량의 교환 관계가 가장 좋은 $b=4$를 기반으로 4채널의 구조를 선택하였다. 구성은 크게 5가지 부분으로, 그림 4와 같이 LHS(left hand side)라 불리는 왼쪽의 $(N_r/4)\times4$ 배열의 PE 셀들의 묶음과 W_M을 곱해주는 복소곱셈기가 $(N_r/4)\times1$ 배열로 구성되며, RHS(right hand side)라 불리는 오른쪽의 $(N_r/4)\times4$ PE 셀들의 묶음, 그리고 W_N을 곱해주는 복소곱셈기 4개와 $N/4$ 크기의 메모리 4개로 구성된다[16].

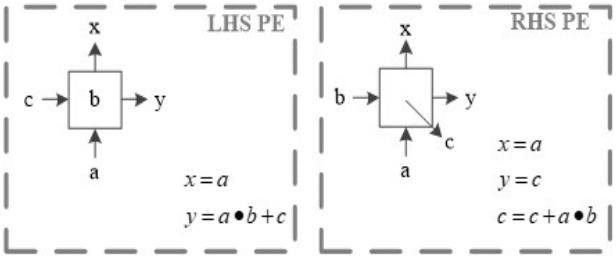


Fig. 5. Internal architecture of LHS and RHS.
그림 5. LHS와 RHS의 내부 구조

LHS와 RHS의 PE 셀 내부는 각각 그림 5와 같다. 4096 길이의 FFT를 수행하는 base-4 시스토릭 어레이 구조 기반 FFT 프로세서의 경우, 하드웨어 구조는 그림 5과 같다. LHS에는 16×4 배열의 PE 셀이 존재하고 RHS에도 16×4 배열의 PE 셀이 존재하며, LHS와 RHS 사이에 W_M을 곱해주는 복소곱셈기가 16×1 배열로 존재하고 RHS의 오른쪽에 W_N을 곱해주는 복소곱셈기 4개와 64 크기의 메모리 4개로 구성된다.

FFT의 과정은 열 FFT와 행 FFT의 순서로 진행된다. 먼저, 열 FFT 과정은 입력 X가 LHS 아래에서 들어오면 PE 셀 내부에 존재하는 C_{M1}값과 X값이 서로 행렬 곱 연산을 하고, $C_{M1}\times X$를 W_M 복소곱셈기로 전달하여 W_M을 곱해준 뒤 식 (11)의 Y의 결과인 $W_M \bullet C_{M1}X$를 RHS로 전달하면, 밑에서 C_{M2}값이 들어오면서 식 (11)의 Z의 결과인 $C_{M2}\times Y^t$를 계산하여, 열 FFT의 결과를 출력하게 되고, 이후, 이 값을 W_N 복소곱셈기로 전달하여, W_N을 곱해준 뒤 메모리에 저장하게 된다. 열 FFT 연산이 끝나면, 메모리에 저장된 열 FFT 결과값들을 행 방향으로 LHS에 입력되고, W_M, RHS, W_N (이때, W_N에서는 1을 곱해줌으로써 W_N 연산은 수행하지 않는 것과 같다.)의 각 모듈의 연산을 열 FFT와 동일한 방식으로 진행하게 되면, 행 FFT까지 완료되며 최종 결과값이 출력된다.

2. RCMC 프로세서 하드웨어

제안된 RCMC 프로세서는 그림 6과 같이 레지스터, RAM, 보간 모듈로 구성되어 있다. 입력되는 데이터는 4개가 동시에 레지스터에 입력된다. 저장되어 있는 데이터는 다음 클락 사이클에 4개의 입력 데이터가 다시 들어오게 되면서, 다음 레지스터 4개로 시프트되는 방식으로 이동하게 된다. 8-tap

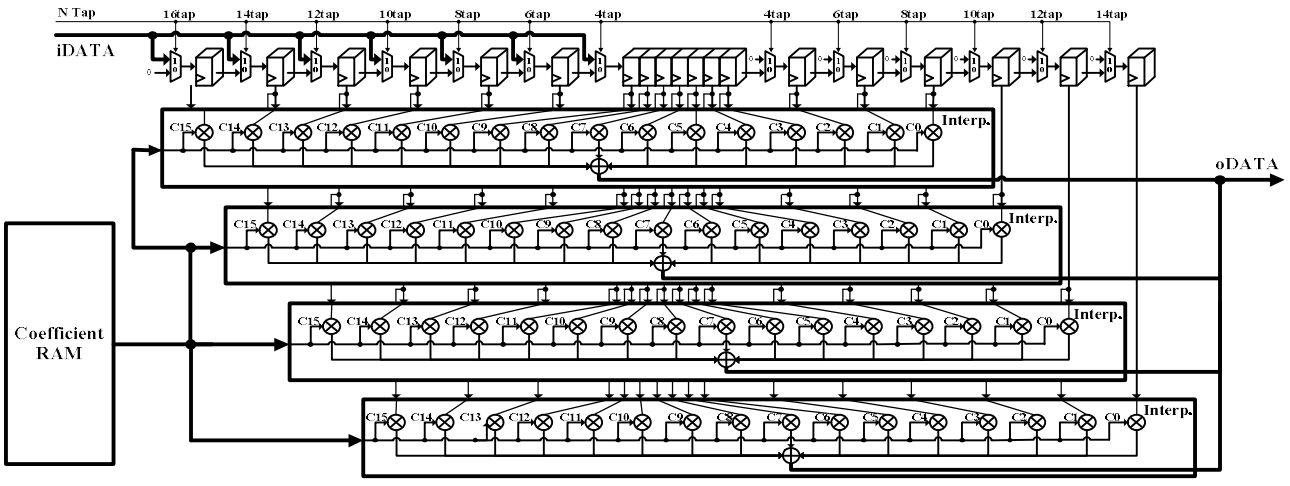


Fig. 6. Proposed RCMC processor architecture.
그림 6. 제안된 RCMC 프로세서 구조

구조기준으로 입력 데이터가 3회 입력되면서, 총 12개의 입력 데이터를 저장하게 된다면, 4개의 출력 데이터가 동시에 생성된다.

출력 데이터가 4개 동시에 생성되는 과정은 다음과 같다. 1번부터 8번까지의 입력 데이터는 동시에 보간 모듈로 입력되어 RAM에서 출력되는 커널(kernel) 상수값과 내적 연산(dot product)을 진행하여 첫 번째 출력 데이터를 생성한다. 마찬가지로, 2~9번 데이터는 두 번째 출력 데이터를 생성하고, 3~10번 데이터는 세 번째 출력, 4~11번 데이터는 네 번째 출력을 생성한다.

이와 같이, 레지스터에 저장되어 있는 입력값과 커널 상수와의 내적 연산을 진행하며, 클락 주기마다 4개씩 입력되는 데이터를 병렬 연산이 가능하도록 4개의 보간 모듈을 사용한다. slave 인터페이스를 통해 응용에 맞도록 RAM에 커널 상수를 가변적으로 저장이 가능하도록 설계하였으며, 레지스터 사이에는 멀티플렉서(multiplexer)를 추가해줌으로써, 보간의 tap 수를 다양한 응용 분야에 맞출 수 있도록 4, 6, 8, 10, 12, 14, 16-tap을 가변적으로 설정 가능하도록 설계하였다.

Ⅳ. 제안된 프로세서 하드웨어 설계 및 구현

RDA 기반 SAR 영상 생성 과정에 있어 필요한 연산인 거리 및 방위 압축과 RCM 보상 연산을 위해, 시스토릭 어레이 구조 기반 정합 필터와 RCMC 프로세서를 설계하였다. 설계된 정합 필터와 RCMC 프로세서는 Xilinx Alveo FPGA를 사용하여 구현되었으며, Alveo FPGA 내부에 다중 커널의 형태로 탑재되었다. 그림 7은 전체적인 실험 환경을 도시한다.

Xilinx Alveo FPGA에 커널 형태로 구현된 정합 필터와 RCMC 프로세서는 python 언어를 통해 제어되며, 구현된 커널을 사용하여 SAR 영상을 생성하는 과정은 그림 8과 같다.

표 1은 256×256부터 4096×4096까지 다양한 크기의 점표적 데이터에 대해 정합 필터 및 RCMC 가속 하드웨어 커널의 개수를 변경하면서 실행 시간을 측정해 가속화 실험을 진행한 결과를 보여준다. 실험 결과, 5 커널 이상 구성에서 수렴하는 것을 확인할 수 있으며, Nvidia RTX3090 GPU를 사용하여 연산한 시간과 비교하면, 4096×4096 이미지를 생성하는 데에 GPU는 2.612 sec이 소요되고, FPGA 가속의 경우, 7 커널 사용 기준 1.321 sec이 소요되어 GPU 대비 약 2배 빠른 속도를 확인할 수 있다. 또한, GPU와 FPGA로 구현한 점표적 결과의 차이가 없음을 그림 9에서 확인할 수 있다.

Fig. 7. Xilinx Alveo FPGA-based experimental setup.
그림 7. Xilinx Alveo FPGA 기반 실험 구성

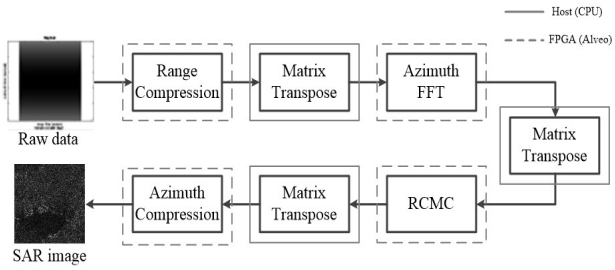
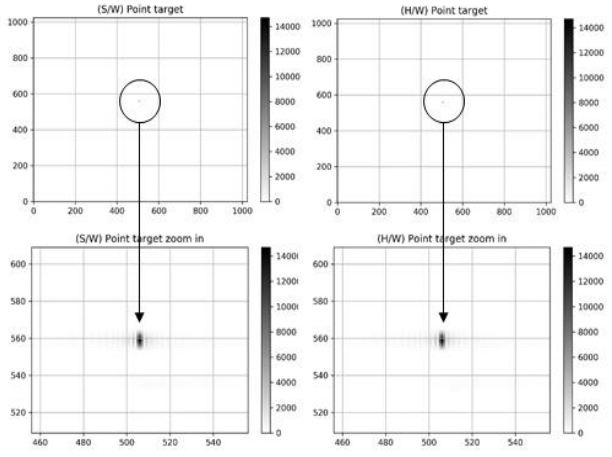


Fig. 8. Xilinx Alveo FPGA-based RDA operation flow.
그림 8. Xilinx Alveo RPGA 기반 RDA 연산 흐름



(a) GPU (b) FPGA

Fig. 9. SAR imaging results for point target by Nvidia GPU and Xilinx Alveo FPGA.
그림 9. Nvidia GPU 및 Xilinx Alveo FPGA 기반 점표적 SAR 영상 생성 결과

Table 1. RDA-based point target execution time according to the number of kernels on Xilinx Alveo FPGA.
표 1. Xilinx Alveo FPGA 내 커널 수에 따른 RDA 기반 점표적 수행시간

Image Size	FPGA Execution Time (sec)							GPU Exe. Time (sec)
	1-kernel	2-kernel	3-kernel	4-kernel	5-kernel	6-kernel	7-kernel	
256×256	0.112	0.063	0.045	0.036	0.032	0.029	0.028	0.153
512×512	0.228	0.126	0.097	0.075	0.064	0.062	0.060	0.294
1024×1024	0.441	0.238	0.198	0.156	0.140	0.137	0.134	0.556
2048×2048	1.092	0.686	0.483	0.421	0.392	0.377	0.368	1.069
4096×4096	2.752	1.993	1.662	1.511	1.379	1.342	1.321	2.612

Table 2. Synthesis results of the proposed matched filter processor and RCMC processor.
표 2. 제안된 정합 필터 프로세서와 RCMC 프로세서 합성 결과

Resource	Matched filter	RCMC
CLB register	59,333	914
CLB LUT	100,263	3,465
Block RAM tile	16	4
DSP	336	256
Maximum Clock Speed : 312MHz		

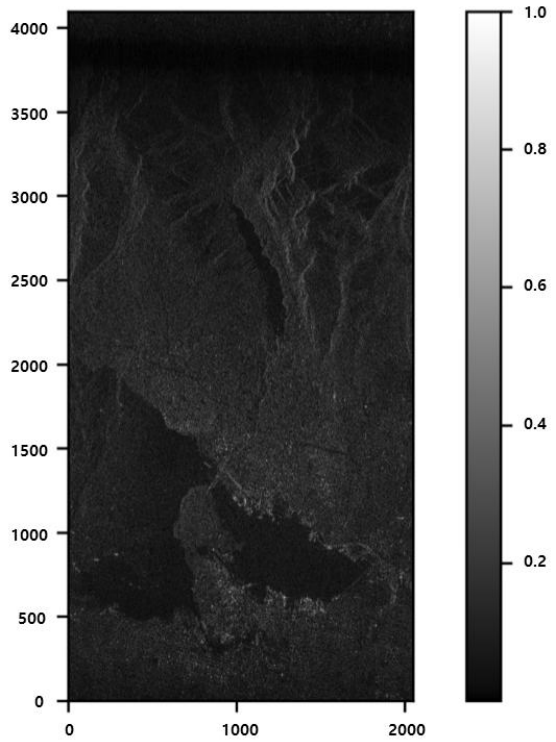


Fig. 10. SAR imaging results for RADARSAT-1 SAR data based on the proposed RDA hardware.
그림 10. RADARSAT-1 SAR 데이터에 대한 제안된 RDA 가속 하드웨어 기반 영상 생성 결과

표 2는 제안된 정합 필터 프로세서와 RCMC 프로세서의 FPGA 기반 구현 결과를 도시한다. 제안된 정합 필터 프로세서는 59,333개의 CLB register, 100,263개의 CLB LUT, 16개의 block RAM tile과 336개의 DSP로 합성 가능함을 확인하였으며, 최대 동작 주파수는 312MHz인 것을 확인하였고, 제안된 RCMC 프로세서의 FPGA 기반 구현 결과는 914개의 CLB register, 3,456개의 CLB LUT, 4개의 block RAM tile과 256개의 DSP로 합성 가능함을 확인하였으며, 최대 동작 주파수는 312MHz인 것을 확인했다.

제안된 정합 필터 및 RCMC 프로세서 기반 RDA 가속 하드웨어를 사용하여 RADARSAT-1에서 측정한 캐나다 밴쿠버 SAR 데이터에 대한 영상생성 실험을 수행하였다. 실험 결과, 영상생성에 소요되는 시간은 총 0.743 sec이며, 그림 10에 제시된 바와 같이, 실제 SAR 데이터에 대해서도 실시간 영상 생성이 가능함을 확인할 수 있다.

Ⅴ. 결론

본 논문에서는 SAR 알고리즘 중 영상의 품질과 속도의 교환 관계를 가장 잘 만족하는 RDA를 가속화하기 위한 FPGA 기반 하드웨어 구조를 제시하고, 구현 및 실험 결과가 제시되었다. RDA의 연산 과정 중 거리 및 방위 압축 연산을 가속하기 위해 시스토릭 어레이 구조 기반 FFT 프로세서를 사용하여 정합 필터를 설계하였고, RCM을 보상해 주기 위해 고속의 sinc 보간 연산을 수행하는 RCMC 프로세서를 설계하였다. 설계된 정합 필터와 RCMC 프로세서를 Xilnx Alveo FPGA 카드에 커널 형태로 탑재하여 다채널 구성으로 가속을 진행하였으

며, Nvidia RTX3090 GPU를 사용해 연산한 시간과 비교하면 4096×4096 크기의 이미지 기준 FPGA 가속이 GPU 대비 약 2배 더 빠른 것을 확인할 수 있었고, 실제 측정된 데이터를 사용하여 영상을 생성할 수 있음을 확인할 수 있었다.

References

[1] W. M. Brown and L. J. Porcello, "An introduction to synthetic-aperture radar," *in IEEE Spectrum*, vol.6, no.9, pp.52-62, 1969.
DOI: 10.1109/MSPEC.1969.5213674

[2] B. Liu, K. Wang, X. Liu and W. Yu, "An Efficient SAR Processor Based on GPU via CUDA," *2009 2nd International Congress on Image and Signal Processing*, pp.1-5, 2009.
DOI: 10.1109/CISP.2009.5304418

[3] M. Wielage, F. Cholewa, C. Fahnemann, P. Pirsch and H. Blume, "High Performance and Low Power Architectures: GPU vs. FPGA for Fast Factorized Backprojection," *2017 Fifth International Symposium on Computing and Networking (CANDAR)*, pp.351-357, 2017.
DOI: 10.1109/CANDAR.2017.101

[4] X. Zhou, Z. J. Yu, Y. Cao and S. Jiang, "SAR Imaging Realization *with FPGA Based on VIVADO HLS," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pp.1-4, 2019.
DOI: 10.1109/ICSIDP47821.2019.9173161

[5] W. M. Brown and R. J. Fredricks, "Range-Doppler Imaging with Motion through Resolution Cells," *in IEEE Transactions on Aerospace and Electronic Systems*, vol.AES-5, no.1, pp.98-102, 1969. DOI: 10.1109/TAES.1969.309826

[6] A. F. Yegulalp, "Fast backprojection algorithm for synthetic aperture radar," *Proceedings of the 1999 IEEE Radar Conference. Radar into the Next Millennium (Cat. No.99CH36249)*, pp.60-65, 1999. DOI: 10.1109/NRC.1999.767270

[7] R. K. Raney, H. Runge, R. Bamler, I. G. Cumming and F. H. Wong, "Precision SAR processing using chirp scaling," *in IEEE Transactions on Geoscience and Remote Sensing*, vol.32, no.4, pp.786-799, 1994. DOI: 10.1109/36.298008

[8] Y. L. Neo, F. H. Wong and I. G. Cumming, "Processing of Azimuth-Invariant Bistatic SAR Data Using the Range Doppler Algorithm," *in IEEE Transactions on Geoscience and Remote Sensing*, vol.46, no.1, pp.14-21, 2008.
DOI: 10.1109/TGRS.2007.909090

[9] Baas, Bevan M. "A 9.5 mW 330 sec 1024-point FFT Processor," *Proceedings of the 1998 Custom Integrated Circuits Conference (CICC),* 1998. DOI: 10.1109/CICC.1998.694921

[10] Shousheng He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference*, pp.131-134, 1998. DOI: 10.1109/CICC.1998.694922

[11] M. Lee, K. Shin and J. Lee, "A VLSI array processor for 16-point FFT," *in IEEE Journal of Solid-State Circuits*, vol.26, no.9, pp.1286-1292, 1991. DOI: 10.1109/4.84946

[12] Kung, Hsiang-Tsung. "Why systolic architectures?," *IEEE computer 15.1,* 37-46, 1982. DOI: 10.1109/mc.1982.1653825

[13] Kung, Sun Yuan. "VLSI array processors," *Englewood Cliffs,* 1988.
DOI: 10.1109/massp.1985.1163741

[14] Hyesook Lim and E. E. Swartzlander, "Multidimensional systolic arrays for the implementation of discrete Fourier transforms," *in IEEE Transactions on Signal Processing*, vol.47, no.5, pp.1359-1370, 1999. DOI: 10.110978.757223

[15] J. G. Nash, "Computationally efficient systolic architecture for computing the discrete Fourier transform," *in IEEE Transactions on Signal Processing*, vol.53, no.12, pp.4640-4651, 2005. DOI: 10.1109/TSP.2005.859216

[16] Jeong, Dongmin, et al. "Design and Implementation Systolic Array FFT Processor Based on Shared Memory," *Journal of IKEEE,* Vol.24, No.3, pp.797-802, 2020.
DOI: 10.7471/ikeee.2020.24.3.797

BIOGRAPHY

Dongmin Jeong (Member)

2021 : BS degree in School of Electronics and Information Engineering, Korea Aerospace University.
2021~present : MS degree course in Department of Smart Air Mobility, Korea Aerospace University

Wookyung Lee (Member)

1994 : BS degree in Electrical Engineering, KAIST.
1996 : MS degree in Electrical Engineering, KAIST.
1999 : Ph.D degree in Electrical Engineering, University College London.
1999~2002 : Research professor, Satellite Technology Research Center, KAIST.
2003~2004 : Research Engineer, Samsung Advanced Institute of Technology
2004~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University

Yunho Jung (Member)

1998 : BS degree in Department of Electrical and Electronic Engineering, Yonsei University.
2000 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University.
2005 : Ph.D degree in Department of Electrical and Electronic Engineering, Yonsei University.
2005~2007 : Senior Engineer, Samsung Electronics.
2007~2008 : Research professor, Institute of Information Engineering, Yonsei University.
2008~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University