International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 12 - Issue 09 || September 2026 || PP. 09-12

Research on the Application of AI Engine in the Design of Multi-Channel Coherent Accumulator using FFT of Digital Phased Array Radar

Nguyen Xuan Luong[1], Dang Thi Thanh Thuy[1]*, Nguyen Phung Bao[2]
[1]*Department of Radio Physics, Faculty of Physics, VNU University of Science, 334 Nguyen Trai, Thanh Xuan, Hanoi, Vietnam*
[2]*Institute of System Integration, Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi*
[*]*Corresponding Author: dangthithanhthuy@hus.edu.vn*

Abstract: The application of an AI Engine in designing a multi-channel coherent accumulator using Fast Fourier transform of phased array radar is investigated. The efficiency of signal accumulation in improving the detection of small-footprint targets in advanced phased array radars is analyzed. An overview of the AI Engine is provided. Outline the architectural design of the Versal series of components of Advanced Micro Devices, illustrating its integration with the AI Engine. The capabilities of the AI Engine and graphics processors are compared. A multi-channel signal accumulator utilizing the Fast Fourier transform in digital phased array radars, employing the Vitis Model Composer design tool, is designed. A multi-channel correlation accumulator using the Fast Fourier transform in multi-channel radar signal accumulation is simulated and compiled.
Keywords: AI Engine, Fast Fourier transform, moving target detector, digital phased array radar.

1. Introduction

Phased array radar (PAA) represents a state-of-the-art technology that employs antenna arrays to detect and track targets [1, 2]. Signal processing plays an integral role in this system, with the moving target detector (MTD) utilising the Fast Fourier transform (FFT) being a prevalent technique. In modern radar systems, field-programmable gate arrays (FPGAs) and graphics processing units (GPUs) are often used to perform FFT algorithms with a minimal number of cumulative cycles (typically, the number of FFT points is less than 256) and a restricted number of processing channels (1 to 4). The number of FFT calculation points is limited by the resources of FPGA and GPU processors, as well as the brief target irradiation time. This results in a restricted number of reflected pulses in the reflected signal beam from the target (in the case where the antenna is continuously rotating during space observation). In the current era, the advent of sophisticated radar evasion techniques employed by small radar footprint targets, such as stealth aircraft and attack UAVs, has rendered the detection of such targets by radars an increasingly challenging endeavour. In order to enhance the capacity to differentiate between moving targets with minimal radar signatures, it is typical to employ techniques that extend the target irradiation time and utilise MTD configurations with numerous signal accumulation cycles. Moreover, in PAAs, particularly digital PAAs, the number of processing channels is considerable, reaching thousands. This posed significant challenges for radar system design engineers in selecting technical solutions for multi-channel MTD processing using FFT algorithms. In contemporary radar installations, GPUs are frequently employed to facilitate these operations. Recently, a novel technology has emerged, offering numerous advantages in the design of radar signal processors, namely the AI Engine. The AI Engine forms part of AMD's Adaptive Compute Acceleration Platform, conferring significant benefits for the design and implementation of FFT processors with a substantial number of points and a multitude of larger processing channels in modern digital PAAs [3-5]. The present article is concerned solely with the application of the AI Engine in the design of processors utilizing FFT with a length of up to 1024 points and 16 processing channels.

2. The application of cumulative signal processing facilitates the detection of small radar footprint targets in modern phased array radars

It is known that the reflected received signal is very weak when small-track targets are involved. One method currently being applied to increase the ability to detect small-track targets is to process the accumulation using a long-length FFT algorithm [6]. In correlation accumulation, complex signal samples are accumulated after correlation demodulation. Provided that the samples are accumulated in phase, the correlation accumulation process increases the SNR with the number of accumulated pulses. However, the actual accumulation will also depend on the fluctuations of the target's radar cross section, which will have a relatively large impact on the signal accumulation results for radars with short irradiation times. It should be noted that the

FFT algorithm itself is a technique for converting signals from the time domain to the frequency domain. In the case of a reflected signal from a target in a pulsed radar system, it is important to recognize that the reflected signal will always contain a Doppler shift. The transformation from the time domain to the frequency domain can therefore be considered to be equivalent to a correlation signal accumulation. In addition to the correlated accumulation method, the non-correlated accumulation method is also employed in radar systems. In contrast to the correlated accumulation method, the signal amplification factor in the non-correlated accumulation method is dependent upon the probability of correct detection and the probability of false alarm.

3. An overview of AI Engine

3.1. AI Engine

The AI Engine represents a powerful computing platform, comprising multiple high-performance vector cores that have been optimized for use in AI and digital signal processing applications. The AI Engine offers high parallelism and high computational efficiency, rendering it an optimal selection for compute-intensive applications such as multi-channel FFT processors in PAAs. The figure below illustrates the component architecture of the Versal family. AMD's Versal family encompasses three programmable processor systems: the Arm® processor subsystem (PS), programmable logic (PL), and AI Engines.

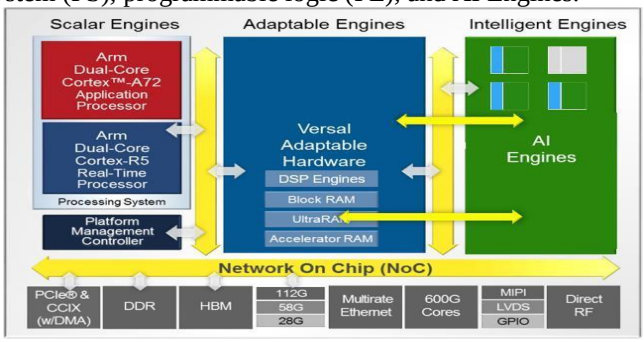

Figure 1: AMD Varsal family architecture diagram including AI Engines

3.2. AI Engine Tile

Each AI Engine tile consists of a Very Long Instruction Word (VLIW), Single Instruction Multiple Data (SIMD) vector processor optimized for machine learning and advanced signal processing applications. The AI Engine processor can run up to 1.3GHz enabling very efficient, high throughput and low latency functions.

As with the VLIW Vector processor, each cell contains program memory for storing the necessary instructions, local data memory for storing data, weights, activations, and coefficients, RISC scalar processors, and a variety of interconnect modes for handling different types of data transmission.

3.3. Parallel processing

The AI engine incorporates multiple vector processing cores, which are capable of performing calculations in parallel on multiple signal channels. This contributes to a reduction in latency and an enhancement of overall system performance. The AI Engine has been designed with the objective of optimizing computational performance, thereby facilitating the rapid and precise execution of complex calculations. The AI Engine is compatible with a multitude of programming languages and development tools, thereby facilitating the programming and optimization of algorithms for engineers. This results in a reduction in the time required for the development process and enables engineers to rapidly test and refine their designs. Furthermore, the system is readily adaptable to alterations in requirements and operational conditions. The AI Engine can be readily incorporated into the system alongside other components, including digital signal processors, FPGAs, and peripherals. This enables the construction of complex systems with high compatibility and straightforward expansion.

3.4. Comparing AI Engine and GPU

GPUs are software-programmable and are therefore capable of handling a wide variety of workloads. However, the data flow in a GPU is largely software-defined and governed by the GPU's complex and rigid memory hierarchy. In order for the computational potential and efficiency of the GPU to be fully realized, it is essential that the data flow of the workload is mapped with precision to the GPU's memory hierarchy. As a result of these data flow and memory hierarchy constraints, GPUs are unable to deliver deterministic throughput and latency simultaneously. Furthermore, due to the varying memory hierarchy and cache levels, GPUs are not

the most efficient in terms of resource and power utilization.

Similarly, AI Engine are programmable in a manner analogous to that of GPU. Furthermore, they are hardware-adaptive, which enables them to handle a diverse range of workloads due to the ability to remap the workload data stream directly on the device. Furthermore, Versal ACAP devices that incorporate an AI Engine are equipped with local memory situated in close proximity to the compute core, enabling high energy efficiency and eliminating cache misses, as observed in GPUs. Additionally, as the hardware can be adapted to align with the specific demands of the workload, the AI Engine is capable of delivering low latency and high throughput while achieving superior device performance and power efficiency compared to GPUs [7].

4. Design of multi-channel signal accumulator using FFT for digital phased array radar with Vitis Model Composer design tool

We can design DSP algorithms when Vitis™ Model Composer. This is a model-based design tool that enables rapid design exploration within the MathWorks

MATLAB® and Simulink® environment and accelerates the path to production on AMD devices through automatic code generation.

Figure 2 illustrates the design of signal accumulators utilizing 1024-point. For the case, the signal accumulator utilizing 64-point FFT algorithms is designed similarly. The designs were implemented on Vitis™ Model Composer with Versal AI Core Series components (xcvc1902-viva1596-2MP-i-L).

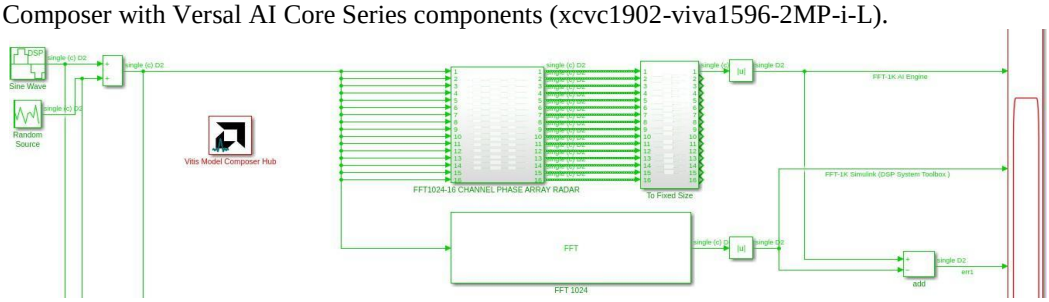

Figure 2: Correlated Accumulators Using FFT Algorithm on AI Engine Model with Vitis Composer

It is assumed that the input signal is a Doppler frequency shift signal reflected from a small radar footprint target, with an SNR approximately of 1. The input signal is the composite received signal, which includes the target Doppler signal and the receiver noise. The outcomes of signal accumulation processing utilizing the FFT 1024 and FFT 64 algorithms are illustrated in Fig. 3. The FFT-1K AI Engine represents the result of correlation accumulating signals through the implementation of the AI Engine. In contrast, the FFT-1K Simulink depicts the outcome of correlation accumulating signals through the utilization of the FFT calculation library of Matlab (DSP System Toolbox). ERR1 is the result of a comparative analysis of the calculation errors between the two methods. Similarly, the FFT-64 AI Engine, FFT-64 Simulink, and ERR2 represent the results of accumulating 64 points using the AI Engine, Matlab, and the calculation errors of the two methods, respectively. The design simulation results demonstrate that as the number of accumulated points increases, the SNR after accumulation also rises. This is exemplified by the 1024-point FFT, which exhibits an SNR approximately 20 dB higher than the 64-point FFT.

After simulation and design compilation, the Vitis Model Composer automatically calls the Vitis compiler to compile the design. The design of floating-point FFTs is implemented on AI Engines with 16 channels running at 1250MHz, with a design resource consumption of less than 10%. This allows for an increase in the number of design channels as well as the integration of complex processing algorithms, particularly in digital PAA designs. The result shows that a total of 16 tiles are employed for the combined functions of computing, 30 tiles for local buffering, and 43 tiles for interconnecting. The design utilizes a total of 89 AI Engine tiles. A total of 32 PLIO stream resources are utilized for transfer between the AI Engine array and PL.

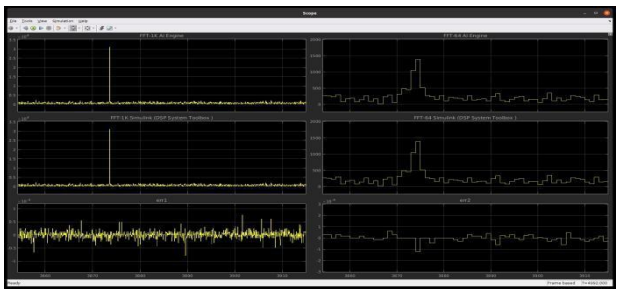

Figure 3: Simulation and compilation results on Vitis Model Composer

5. Conclusions

The key factors that make the AI Engine an ideal solution for modern phased array radar applications are its powerful parallel processing, high computing performance, programming flexibility, and energy efficiency. The Vitis Model Composer design tool facilitates comprehensive system simulation, thereby assisting system designers in reducing product design time. These advantages enable AI Engine to not only optimally address the multi-channel processing challenge for PAAs, but also to unlock numerous prospective development avenues in the future for designs of radio systems utilizing SDR.

6. References

[1]. P. Wang, H. Meng and Y. Wei, "FMCW Radar Imaging with Multi-channel Antenna Array via Sparse Recovery Technique," *2010 International Conference on Electrical and Control Engineering*, Wuhan, China, 2010, pp. 1018-1021, doi: 10.1109/iCECE.2010.258.

[2]. C. A. Sessions, R. A. Romero and D. J. Fouts, "A Field-Programmable Gate Array Implementation of a Cognitive Radar Target Recognition System," *2022 IEEE Radar Conference (Radar Conf22)*, New York City, NY, USA, 2022, pp. 1-6, doi: 10.1109/RadarConf2248738.2022.9764250.

[3]. Phased-Array Radar Design: Application of Radar Fundamentals. Jeffrey, Tom. Published by Scitech Publishing, 2009. ISBN 10: 1891121693/ISBN 13: 9781891121692

[4]. AI Engine: Meeting the Compute Demands of Next-Generation Applications. https://www.xilinx.com/products/technology/ai-engine.html (accessed 1 July 2024).

[5]. R. V. Chakaravarthy, H. Kwon and H. Jiang, "Vision Control Unit in Fully Self Driving Vehicles using Xilinx MPSoC and Opensource Stack,"*2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Tokyo, Japan, 2021, pp. 311-317.

[6]. Barton, David Knox. *Radar Equations for Modern Radar*. Artech House, 2013.

[7]. Xilinx AI Engines and Their Applications (WP506). https://docs.amd.com/v/u/en-US/wp506-ai-engine (accessed 1 July 2024).