High-Performance Floating-Point FFT Accelerator with Dynamic Twiddle Factor Generation Using CORDIC

Team Member Details

- Tadikonda Venkata Sai Chaitanya
  Email ID: TVS.chaitanya@iiitb.ac.in
  Current Status: IMTech Third Year Student, ECE
  Institute Name: International Institute of Information Technology Bangalore (IIIT-B)

- Akula Nishith
  Email ID: A.Nishith@iiitb.ac.in
  Current Status: IMTech Third Year Student, ECE
  Institute Name: International Institute of Information Technology Bangalore (IIIT-B)

- Komaragiri Sai Vishwanath Rohit
  Email ID: komaragiri.sai@iiitb.ac.in
  Current Status: IMTech Third Year Student, ECE
  Institute Name: International Institute of Information Technology Bangalore (IIIT-B)

- Lohitaksh Maruvada
  Email ID: lohitaksh.maruvada@iiitb.ac.in
  Current Status: IMTech Third Year Student, ECE
  Institute Name: International Institute of Information Technology Bangalore (IIIT-B)

Abstract :

The Fast Fourier Transform (FFT) is a fundamental technique in signal processing, essential for efficient analysis in the frequency domain. While traditional FFT accelerators rely on pre-computed and memory-intensive twiddle factors, this project proposes a **High-Performance Floating-Point FFT Accelerator** integrated with a real-time, hardware-efficient twiddle factor generation module using the **Coordinate Rotation Digital Computer (CORDIC)** algorithm.

The CORDIC algorithm, renowned for its iterative and hardware-friendly computation of trigonometric functions, eliminates the need for large memory banks to store pre-computed twiddle factors. Instead, twiddle factors are dynamically generated during runtime, significantly reducing the memory footprint while maintaining computational accuracy. By leveraging floating-point arithmetic, the proposed design ensures precision and scalability for high-performance applications such as radar systems, 5G communications, and real-time signal analysis.

The architecture is optimized for throughput and efficiency through pipeline implementation, enabling parallel processing of FFT stages. This pipelined design minimizes latency, making it suitable for real-time applications. Additionally, the use of floating-point representation ensures compatibility with a wide range of input signal magnitudes, avoiding precision loss common in fixed-point implementations.

This project focuses on key challenges such as optimizing the CORDIC-based module for high-speed operation and integrating it seamlessly with the FFT datapath. The design also incorporates advanced memory management techniques to handle input and output buffers efficiently.

By reducing the dependency on static memory for twiddle factor storage and enhancing computational performance, this project offers a cutting-edge solution for FFT acceleration. The proposed design is highly versatile, with potential applications in embedded systems, digital signal processing, and edge computing platforms.This approach enables more resource-efficient and scalable FFT designs, especially for applications requiring real-time, high-precision computations.

Project Description

The proposed project, **"High-Performance Floating-Point FFT Accelerator with Dynamic Twiddle Factor Generation Using CORDIC,"** aims to design and implement a cutting-edge floating-point FFT (Fast Fourier Transform) accelerator that integrates a real-time, hardware-efficient twiddle factor generation module based on the CORDIC (Coordinate Rotation Digital Computer) algorithm.

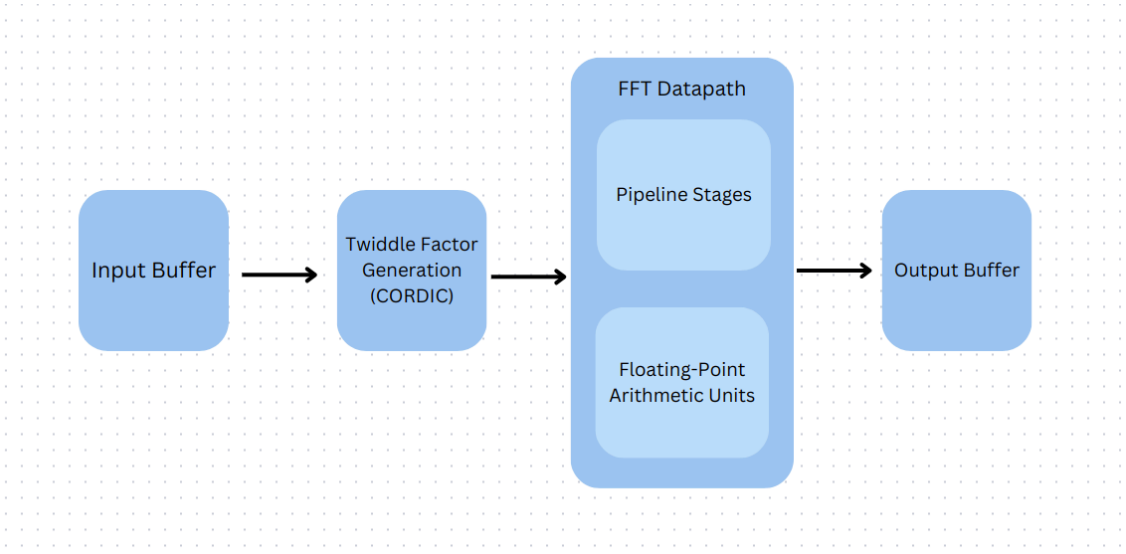
Traditional FFT architectures often rely on pre-computed and stored twiddle factors, which demand significant memory resources and introduce latency during memory access. Our innovative approach eliminates the dependency on pre-computed data by dynamically generating twiddle factors in real-time using the CORDIC algorithm.

This project combines the advantages of dynamic computation with high-speed FFT operations to cater to applications in telecommunications, radar systems, image processing, and real-time analytics.

Impacts

- **Reduced Memory Footprint:** By eliminating the need to store pre-computed twiddle factors, the proposed design significantly reduces memory requirements, freeing up hardware resources for other functionalities.
- **Enhanced Performance:** Real-time computation of twiddle factors avoids memory access bottlenecks, enabling faster and more efficient FFT operations.
- **Cost and Energy Efficiency:** The hardware-efficient CORDIC algorithm reduces power consumption and chip area, contributing to a lower overall cost for ASIC implementations.
- **Broad Applicability:** This accelerator is particularly valuable for real-time signal processing in applications like wireless communication, medical imaging, and defense systems where performance and precision are critical.

FlowChart:



4-Week Plan:

Week 1: Initial Research and Design Specification

- **Objective: Define the project's functional and technical specifications, and establish the baseline framework.**
    - Conduct a literature review on FFT and CORDIC algorithms, focusing on hardware-efficient methods and real-time twiddle factor generation.
    - Design the system architecture, defining modules for FFT computation, twiddle factor generation, and their integration with CORDIC.
    - Determine the number format (floating-point precision) and bit-width considerations.

Week 2: Development of the CORDIC-Based Twiddle Factor Generator

- **Objective: Implement the CORDIC algorithm for dynamic twiddle factor generation.**
    - Implement the CORDIC algorithm for sine and cosine computation, focusing on fixed-point arithmetic.
    - Optimize the CORDIC module for accuracy and hardware efficiency.
    - Integrate the CORDIC module with the FFT pipeline to enable real-time twiddle factor generation.
    - Validate the functionality through simulation and perform an error analysis.
    - Optimize resource usage by minimizing hardware overhead and latency.

Week 3: Integration of the FFT Accelerator

- **Objective: Develop the FFT processing pipeline and integrate it with the CORDIC module.**
    - Implement the FFT core, focusing on the butterfly operation and floating-point arithmetic optimizations.
    - Replace pre-computed twiddle factors with the dynamic CORDIC-based generator.
    - Use Cadence tools to synthesize and Analyze performance metrics, including throughput, latency, and resource utilization.

Week 4: Optimization, Validation, and Documentation

- **Objective: Optimize the overall design, validate the system, and document the results.**
    - Optimize the FFT accelerator for higher clock frequencies, lower power consumption, and minimal resource usage using Cadence tools for power and timing analysis.
    - Implement the design on ASIC and validate real-time performance using synthetic or real-world signal data.
    - Conduct extensive validation for numerical accuracy and latency across varying input sizes using MATLAB/Simulink and Cadence verification tools.
    - Document the project, including system architecture, simulation results, and implementation benchmarks, and deliver the final project presentation.