hitex

Hitex Tech Tips

CMSIS DSP for Cortex-M Microcontrollers

CMSIS-DSP for Cortex-M Microcontrollers

By Trevor Martin

Introduction

The CMSIS-DSP (Cortex Microcontroller Software Interface Standard - Digital Signal Processing) library is a comprehensive source code library providing essential DSP algorithms tailored for all Cortex-M based microcontrollers. The CMSIS-DSP library covers a wide array of algorithms divided into several major groups:

- Basic Math Functions: Primitives for vector maths
- Fast Math Functions: Trigonometric and square root functions with optimized performance.
- Complex Math Functions: Operations for complex numbers.
- Filters: Finite Impulse Response (FIR), Infinite Impulse Response (IIR), and other filter designs.
- Matrix Functions: Operations including addition, multiplication, and transposition.
- Transform Functions: Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT).
- Control Functions: Including, PID control loop and Motor Control (Vector Park Transform and Vector Clark Transform).
- Statistical Functions: Mean, variance, standard deviation, and other statistical measures.
- Support Functions: Convert between integer to Q numbers and additional vector functions
- Interpolation Functions: Linear and cubic interpolation functions.
- Classic Machine Learning algorithms: Support vector Machine (SVM), Bayesian inference and clustering distance functions
- Quaternion Functions: To calculate rotations and orientations in a 3D space.

Optimization for Cortex-M Processors

The CMSIS-DSP library is highly optimized for various Cortex-M processors and uses specific architectural features to enhance performance and efficiency.

- Hardware Floating Point: On processors with hardware floating point units (e.g., Cortex-M4, Cortex-M7, Cortex-M33), the library uses hardware-accelerated floating-point operations to achieve significant performance gains.
- SIMD Instructions: Single Instruction, Multiple Data (SIMD) instructions available in Cortex-M4 and Cortex-M7 processors are utilized to perform parallel operations, thus boosting the throughput of mathematical computations.
- Helium Vector Processor: For the next generation of Cortex-M processors like Cortex-M55, the library takes advantage of the Helium vector processing technology, which significantly enhances the processing of dense mathematical computations and DSP tasks.

Python Wrapper for Simulation

While the CMSIS-DSP library is C source code it is also available with a python wrapper. This allows engineers to prototype and simulate DSP algorithms in a high-level Python environment before deploying them on actual hardware. This integration ensures a seamless transition from algorithm development to implementation, enabling rapid testing and iteration.

CMSIS-Stream for Optimized Data Flow

In a real-world system, the DSP system may consist of several algorithms arranged as a pipeline. In such a system each algorithm may process different data block sizes. This leads difficulties optimising the data flow through the pipeline.

The CMSIS-Stream framework complements the CMSIS-DSP library by optimizing the flow of data through such a pipeline. CMSIS-Stream provides:

- Efficient Data Management: It handles data movement between different processing stages, ensuring minimal latency and efficient resource utilization.
- Modular Design: Allows for easy configuration and adjustment of DSP pipelines to meet specific application needs.
- Synchronization and Buffering: Manages synchronization and buffering of data streams, which is crucial for real-time signal processing applications.
- Python Utility: To calculate the optimum execution pattern for a DSP pipeline.

In Summary

The CMSIS-DSP library is a powerful tool that brings advanced DSP and ML capabilities to small microcontrollers, making it ideal for a wide range of applications. Optimised for Cortex-M processors, it ensures high performance while maintaining efficiency. With the added benefits of a Python wrapper for simulation and CMSIS-Stream for efficient data flow management, engineers can confidently design and implement sophisticated DSP systems.

Related Information

- [CMSIS-DSP Library Documentation]

(https://arm-software.github.io/CMSIS_5/DSP/html/index.html)

- [CMSIS DSP repository]

(https://github.com/ARM-software/CMSIS-DSP)

- [CMSIS-Stream User Guide]

(https://github.com/ARM-software/CMSIS-Stream)

Further Information

For more information visit our website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: LinkedIn

Sign up to receive Tech Tips directly to your inbox. Never miss a trick!