CMSIS DSP Software Library

Introduction

This user manual describes the CMSIS DSP software library, a suite of common signal processing functions for use on Cortex-M processor based devices.

The library is divided into a number of functions each covering a specific category:

- Basic math functions
- Fast math functions
- Complex math functions
- Filters
- Matrix functions
- Transforms
- Motor control functions
- Statistical functions
- Support functions
- Interpolation functions

The library has separate functions for operating on 8-bit integers, 16-bit integers, 32-bit integer and 32-bit floating-point values.

Using the Library

The library installer contains prebuilt versions of the libraries in the `Lib` folder.

- arm_cortexM7lfdp_math.lib (Cortex-M7, Little endian, Double Precision Floating Point Unit)
- arm_cortexM7bfdp_math.lib (Cortex-M7, Big endian, Double Precision Floating Point Unit)
- arm_cortexM7lfsp_math.lib (Cortex-M7, Little endian, Single Precision Floating Point Unit)
- arm_cortexM7bfsp_math.lib (Cortex-M7, Big endian and Single Precision Floating Point Unit on)
- arm_cortexM7l_math.lib (Cortex-M7, Little endian)
- arm_cortexM7b_math.lib (Cortex-M7, Big endian)
- arm_cortexM4lf_math.lib (Cortex-M4, Little endian, Floating Point Unit)
- arm_cortexM4bf_math.lib (Cortex-M4, Big endian, Floating Point Unit)
- arm_cortexM4l_math.lib (Cortex-M4, Little endian)
- arm_cortexM4b_math.lib (Cortex-M4, Big endian)
- arm_cortexM3l_math.lib (Cortex-M3, Little endian)
- arm_cortexM3b_math.lib (Cortex-M3, Big endian)
- arm_cortexM0l_math.lib (Cortex-M0 / Cortex-M0+, Little endian)
- arm_cortexM0b_math.lib (Cortex-M0 / Cortex-M0+, Big endian)
- arm_ARMv8MBLl_math.lib (Armv8-M Baseline, Little endian)
- arm_ARMv8MMLl_math.lib (Armv8-M Mainline, Little endian)
- arm_ARMv8MMLlfsp_math.lib (Armv8-M Mainline, Little endian, Single Precision Floating Point Unit)
- arm_ARMv8MMLld_math.lib (Armv8-M Mainline, Little endian, DSP instructions)
- arm_ARMv8MMLldfsp_math.lib (Armv8-M Mainline, Little endian, DSP instructions, Single Precision Floating Point Unit)

The library functions are declared in the public file `arm_math.h` which is placed in the `Include` folder. Simply include this file and link the appropriate library in the application and begin calling the library functions. The Library supports single public header file `arm_math.h` for Cortex-M cores with little endian and big endian. Same header file will be used for floating point unit(FPU) variants. Define the appropriate preprocessor macro ARM_MATH_CM7 or ARM_MATH_CM4 or ARM_MATH_CM3 or ARM_MATH_CM0 or ARM_MATH_CM0PLUS depending on the target processor in the application. For Armv8-M cores define preprocessor macro ARM_MATH_ARMV8MBL or ARM_MATH_ARMV8MML. Set preprocessor macro __DSP_PRESENT if Armv8-M Mainline core supports DSP instructions.

Examples

The library ships with a number of examples which demonstrate how to use the library functions.

Toolchain Support

The library has been developed and tested with MDK version 5.14.0.0 The library is being tested in GCC and IAR toolchains and updates on this activity will be made available shortly.

Building the Library

The library installer contains a project file to rebuild libraries on MDK toolchain in the `CMSIS\DSP_Lib\Source\ARM` folder.

- arm_cortexM_math.uvprojx

The libraries can be built by opening the arm_cortexM_math.uvprojx project in MDK-ARM, selecting a specific target, and defining the optional preprocessor macros detailed above.

Preprocessor Macros

Each library project have different preprocessor macros.

- UNALIGNED_SUPPORT_DISABLE:

Define macro UNALIGNED_SUPPORT_DISABLE, If the silicon does not support unaligned memory access

- ARM_MATH_BIG_ENDIAN:

Define macro ARM_MATH_BIG_ENDIAN to build the library for big endian targets. By default library builds for little endian targets.

- ARM_MATH_MATRIX_CHECK:

Define macro ARM_MATH_MATRIX_CHECK for checking on the input and output sizes of matrices

- ARM_MATH_ROUNDING:

Define macro ARM_MATH_ROUNDING for rounding on support functions

- ARM_MATH_CMx:

Define macro ARM_MATH_CM4 for building the library on Cortex-M4 target, ARM_MATH_CM3 for building library on Cortex-M3 target and ARM_MATH_CM0 for building library on Cortex-M0 target, ARM_MATH_CM0PLUS for building library on Cortex-M0+ target, and ARM_MATH_CM7 for building the library on cortex-M7.

- ARM_MATH_ARMV8MxL:

Define macro ARM_MATH_ARMV8MBL for building the library on Armv8-M Baseline target, ARM_MATH_ARMV8MML for building library on Armv8-M Mainline target.

- __FPU_PRESENT:

Initialize macro __FPU_PRESENT = 1 when building on FPU supported Targets. Enable this macro for floating point libraries.

- __DSP_PRESENT:

Initialize macro __DSP_PRESENT = 1 when Armv8-M Mainline core supports DSP instructions.

CMSIS-DSP in ARM::CMSIS Pack

The following files relevant to CMSIS-DSP are present in the **ARM::CMSIS** Pack directories:

File/Folder	Content
CMSIS\Documentation\DSP	This documentation
CMSIS\DSP_Lib	Software license agreement (license.txt)
CMSIS\DSP_Lib\Examples	Example projects demonstrating the usage of the library functions
CMSIS\DSP_Lib\Source	Source files for rebuilding the library

Revision History of CMSIS-DSP

Please refer to **Revision History of CMSIS-DSP**.

Copyright Notice