HissTools FFT

1.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

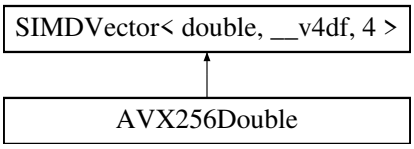
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 AVX256Double Struct Reference

Inheritance diagram for AVX256Double:



Public Member Functions

- **AVX256Double** (__v4df a)
- AVX256Double **operator+** (const AVX256Double a) const
- AVX256Double **operator-** (const AVX256Double a) const
- AVX256Double **operator*** (const AVX256Double a) const

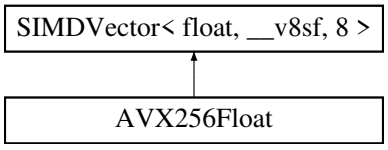
Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.2 AVX256Float Struct Reference

Inheritance diagram for AVX256Float:



Public Member Functions

- **AVX256Float** (__v8sf a)
- AVX256Float **operator+** (AVX256Float a) const
- AVX256Float **operator-** (AVX256Float a) const
- AVX256Float **operator*** (AVX256Float a) const

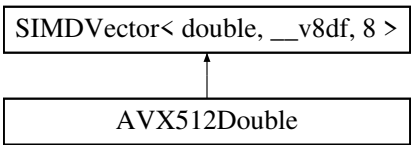
Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.3 AVX512Double Struct Reference

Inheritance diagram for AVX512Double:



Public Member Functions

- **AVX512Double** (__v8df a)
- AVX512Double **operator+** (const AVX512Double a) const
- AVX512Double **operator-** (const AVX512Double a) const
- AVX512Double **operator*** (const AVX512Double a) const

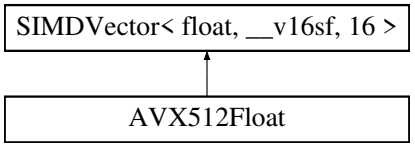
Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.4 AVX512Float Struct Reference

Inheritance diagram for AVX512Float:



Public Member Functions

- **AVX512Float** (__v16sf a)
- AVX512Float **operator+** (const AVX512Float a) const
- AVX512Float **operator-** (const AVX512Float a) const
- AVX512Float **operator*** (const AVX512Float a) const

Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.5 Scalar< T > Struct Template Reference

Public Types

- typedef T **scalar_type**
- typedef Split< scalar_type > **split_type**
- typedef Setup< scalar_type > **setup_type**

Public Member Functions

- **Scalar** (T a)
- Scalar **operator+** (const Scalar a) const
- Scalar **operator-** (const Scalar a) const
- Scalar **operator*** (const Scalar a) const

Static Public Member Functions

- static int **size** ()

Public Attributes

- T **mVal**

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.6 Setup< T > Struct Template Reference

Public Attributes

- unsigned long **max_fft_log2**
- Split< T > **tables** [28]

The documentation for this struct was generated from the following file:

- HISSTools_FFT.h

4.7 SIMDVector< T, U, vec_size > Struct Template Reference

Public Types

- typedef T **scalar_type**
- typedef Split< scalar_type > **split_type**
- typedef Setup< scalar_type > **setup_type**

Public Member Functions

- **SIMDVector** (U a)

Static Public Member Functions

- static int **size** ()

Public Attributes

- U **mVal**

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.8 Split< T > Struct Template Reference

Public Attributes

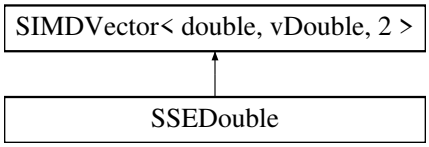
- T * **realp**
- T * **imagp**

The documentation for this struct was generated from the following file:

- HISSTools_FFT.h

4.9 SSEDouble Struct Reference

Inheritance diagram for SSEDouble:



Public Member Functions

- **SSEDouble** (vDouble a)
- SSEDouble **operator+** (const SSEDouble a) const
- SSEDouble **operator-** (const SSEDouble a) const
- SSEDouble **operator*** (const SSEDouble a) const

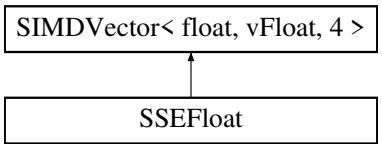
Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

4.10 SSEFloat Struct Reference

Inheritance diagram for SSEFloat:



Public Member Functions

- **SSEFloat** (vFloat a)
- SSEFloat **operator+** (const SSEFloat a) const
- SSEFloat **operator-** (const SSEFloat a) const
- SSEFloat **operator*** (const SSEFloat a) const

Additional Inherited Members

The documentation for this struct was generated from the following file:

- FFT_Core.h

Chapter 5

File Documentation

5.1 HISSTools_FFT.h File Reference

```
#include <stdint.h>
```

Classes

- struct Split< T >
- struct Setup< T >

Typedefs

- typedef Setup< float > * **FFT_SETUP_F**
- typedef Setup< double > * **FFT_SETUP_D**
- typedef Split< float > **FFT_SPLIT_COMPLEX_F**
- typedef Split< double > **FFT_SPLIT_COMPLEX_D**

Functions

- void hisstools_create_setup (FFT_SETUP_D *setup, uintptr_t max_fft_log_2)
- void hisstools_create_setup (FFT_SETUP_F *setup, uintptr_t max_fft_log_2)
- void hisstools_destroy_setup (FFT_SETUP_D setup)
- void hisstools_destroy_setup (FFT_SETUP_F setup)
- void hisstools_fft (FFT_SETUP_D setup, FFT_SPLIT_COMPLEX_D *input, uintptr_t log2n)
- void hisstools_fft (FFT_SETUP_F setup, FFT_SPLIT_COMPLEX_F *input, uintptr_t log2n)
- void hisstools_rfft (FFT_SETUP_D setup, FFT_SPLIT_COMPLEX_D *input, uintptr_t log2n)
- void hisstools_rfft (FFT_SETUP_F setup, FFT_SPLIT_COMPLEX_F *input, uintptr_t log2n)
- void hisstools_ifft (FFT_SETUP_D setup, FFT_SPLIT_COMPLEX_D *input, uintptr_t log2n)
- void hisstools_ifft (FFT_SETUP_F setup, FFT_SPLIT_COMPLEX_F *input, uintptr_t log2n)
- void hisstools_rifft (FFT_SETUP_D setup, FFT_SPLIT_COMPLEX_D *input, uintptr_t log2n)
- void hisstools_rifft (FFT_SETUP_F setup, FFT_SPLIT_COMPLEX_F *input, uintptr_t log2n)
- void **hisstools_unzip_zero** (double *input, FFT_SPLIT_COMPLEX_D *output, uintptr_t in_length, uintptr_t log2n)
- void **hisstools_unzip_zero** (float *input, FFT_SPLIT_COMPLEX_F *output, uintptr_t in_length, uintptr_t log2n)
- void **hisstools_unzip_zero** (float *input, FFT_SPLIT_COMPLEX_D *output, uintptr_t in_length, uintptr_t log2n)
- void **hisstools_unzip** (double *input, FFT_SPLIT_COMPLEX_D *output, uintptr_t log2n)
- void **hisstools_unzip** (float *input, FFT_SPLIT_COMPLEX_F *output, uintptr_t log2n)
- void **hisstools_zip** (FFT_SPLIT_COMPLEX_D *input, double *output, uintptr_t log2n)
- void **hisstools_zip** (FFT_SPLIT_COMPLEX_F *input, float *output, uintptr_t log2n)

5.1.1 Function Documentation

5.1.1.1 hisstools_create_setup() [1/2]

```
void hisstools_create_setup (
            FFT_SETUP_D * setup,
            uintptr_t max_fft_log_2 )
```

hisstools_create_setup() creates an FFT setup suitable for double-precision FFTs and iFFTs up to a maximum specified size.

Parameters

setup	A pointer to an uninitialised FFT_SETUP_D.
max_fft_log↩ _2	The log base 2 of the FFT size of the maimum FFT size you wish to support..

Remarks

On return the object pointed to by setup will be intialsed,

5.1.1.2 hisstools_create_setup() [2/2]

```
void hisstools_create_setup (
            FFT_SETUP_F * setup,
            uintptr_t max_fft_log_2 )
```

hisstools_create_setup() creates an FFT setup suitable for single-precision FFTs and iFFTs up to a maximum specified size.

Parameters

setup	A pointer to an uninitialised FFT_SETUP_F.
max_fft_log↩ _2	The log base 2 of the FFT size of the maimum FFT size you wish to support..

Remarks

On return the object pointed to by setup will be intialsed,

5.1.1.3 hisstools_destroy_setup() [1/2]

```
void hisstools_destroy_setup (
            FFT_SETUP_D setup )
```

hisstools_destroy_setup() destroys a double-precision FFT setup.

Parameters

setup	A FFT_SETUP_D (double-precision setup).

Remarks

After calling this routine the setup is destroyed.

5.1.1.4 hisstools_destroy_setup() [2/2]

```
void hisstools_destroy_setup (
            FFT_SETUP_F setup )
```

hisstools_destroy_setup() destroys a single-precision FFT setup.

Parameters

setup	A FFT_SETUP_F (single-precision setup).

Remarks

After calling this routine the setup is destroyed.

5.1.1.5 hisstools_fft() [1/2]

```
void hisstools_fft (
            FFT_SETUP_D setup,
            FFT_SPLIT_COMPLEX_D * input,
            uintptr_t log2n )
```

hisstools_fft() performs an in-place complex Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_D that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_D structure containing the complex input.
log2n	The log base 2 of the FFT size.

Remarks

The FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned

5.1.1.6 hisstools_fft() [2/2]

```
void hisstools_fft (
            FFT_SETUP_F setup,
            FFT_SPLIT_COMPLEX_F * input,
            uintptr_t log2n )
```

hisstools_fft() performs an in-place complex Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_F that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_F structure containing the complex input.
log2n	The log base 2 of the FFT size.

Remarks

The FFT may be performed with scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned.

5.1.1.7 hisstools_ifft() [1/2]

```
void hisstools_ifft (
            FFT_SETUP_D setup,
            FFT_SPLIT_COMPLEX_D * input,
            uintptr_t log2n )
```

hisstools_ifft() performs an in-place inverse complex Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_D that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_D structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The inverse FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned.

5.1.1.8 hisstools_ifft() [2/2]

```
void hisstools_ifft (
            FFT_SETUP_F setup,
```

```
            FFT_SPLIT_COMPLEX_F * input,
            uintptr_t log2n )
```

hisstools_ifft() performs an in-place inverse complex Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_D that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_F structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The inverse FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_F are sixteen byte aligned.

5.1.1.9 hisstools_rfft() [1/2]

```
void hisstools_rfft (
            FFT_SETUP_D setup,
            FFT_SPLIT_COMPLEX_D * input,
            uintptr_t log2n )
```

hisstools_rfft() performs an in-place real Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_D that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_D structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned. Note that the input should first be unzipped into the complex input structure using hisstools_unzip() or hisstools_unzip_zero).

5.1.1.10 hisstools_rfft() [2/2]

```
void hisstools_rfft (
            FFT_SETUP_F setup,
            FFT_SPLIT_COMPLEX_F * input,
            uintptr_t log2n )
```

hisstools_rfft() performs an in-place real Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_F that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_F structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned. Note that the input should first be unzipped into the complex input structure using hisstools_unzip() or hisstools_unzip_zero).

5.1.1.11 hisstools_rifft() [1/2]

```
void hisstools_rifft (
            FFT_SETUP_D setup,
            FFT_SPLIT_COMPLEX_D * input,
            uintptr_t log2n )
```

hisstools_rifft() performs an in-place inverse real Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_D that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_D structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The inverse FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned. Note that the output will need to be zipped from the complex output structure using hisstools_zip().

5.1.1.12 hisstools_rifft() [2/2]

```
void hisstools_rifft (
            FFT_SETUP_F setup,
            FFT_SPLIT_COMPLEX_F * input,
            uintptr_t log2n )
```

hisstools_rifft() performs an in-place inverse real Fast Fourier Transform.

Parameters

setup	A FFT_SETUP_F that has been created to deal with an appropriate maximum size of FFT.
input	A pointer to a FFT_SPLIT_COMPLEX_F structure containing a complex input.
log2n	The log base 2 of the FFT size.

Remarks

The inverse FFT may be performed with either scalar or SIMD instructions. SIMD instuctions will be used when the pointers within the FFT_SPLIT_COMPLEX_D are sixteen byte aligned. Note that the output will need to be zipped from the complex output structure using hisstools_zip().