
hipCUB Documentation

Release 2.10.9

Advanced Micro Devices

Oct 27, 2021

CONTENTS:

| | | |
|----------|-------------------------------|------------|
| 1 | Introduction | 1 |
| 1.1 | Overview | 1 |
| 1.2 | rocPRIM backend | 1 |
| 2 | Library API | 3 |
| 2.1 | Full API | 3 |
| 2.1.1 | Namespaces | 3 |
| 2.1.2 | Classes and Structs | 8 |
| 2.1.3 | Enums | 88 |
| 2.1.4 | Functions | 92 |
| 2.1.5 | Defines | 111 |
| 2.1.6 | Typedefs | 116 |
| 3 | Indices and tables | 117 |
| | Index | 119 |

INTRODUCTION

1.1 Overview

hipCUB is a thin wrapper library on top of rocPRIM or CUB. It enables developers to port project using CUB library to the HIP layer and to run them on AMD hardware. In the ROCm environment, hipCUB uses rocPRIM library as the backend, however, on CUDA platforms it uses CUB instead.

- When using hipCUB you should only include `<hipcub/hipcub.hpp>` header.
- When rocPRIM is used as backend `HIPCUB_ROCPRIM_API` is defined.
- When CUB is used as backend `HIPCUB_CUB_API` is defined.
- Backends are automatically selected based on platform detected by HIP layer (`__HIP_PLATFORM_HCC__`, `__HIP_PLATFORM_NVCC__`).

1.2 rocPRIM backend

hipCUB with rocPRIM backend may not support all function and features CUB has because of the differences between ROCm (HIP) platform and CUDA platform.

Not-supported features and differences:

- Functions, classes and macros which are not in the public API or not documented are not supported.
- Device-wide primitives can't be called from kernels (dynamic parallelism is not supported in HIP on ROCm).
- `DeviceSpmv` is not supported.
- Fancy iterators: `CacheModifiedInputIterator`, `CacheModifiedOutputIterator`, and `TexRefInputIterator` are not supported.
- Thread I/O:
 - `CacheLoadModifier`, `CacheStoreModifier` cache modifiers are not supported.
 - `ThreadLoad`, `ThreadStore` functions are not supported.
- Storage management and debug functions:
 - `Debug`, `PtxVersion`, `SmVersion` functions and `CubDebug`, `CubDebugExit`, `_CubLog` macros are not supported.
- Intrinsic:
 - `ThreadExit`, `ThreadTrap` - not supported.
 - Warp thread masks (when used) are 64-bit unsigned integers.

- member_mask input argument is ignored in WARP_* functions.
- Arguments first_lane, last_lane, and member_mask are ignored in Shuffle* functions.
- Utilities:
 - SwizzleScanOp, ReduceBySegmentOp, ReduceByKeyOp, CastOp - not supported.

LIBRARY API

2.1 Full API

2.1.1 Namespaces

Namespace detail

Contents

- *Functions*

Functions

- *Function detail::to_BlockHistogramAlgorithm_enum*
- *Function detail::to_BlockReduceAlgorithm_enum*

Namespace hipcub

Contents

- *Namespaces*
- *Classes*
- *Enums*
- *Functions*
- *Typedefs*

Namespaces

- *Namespace hipcub::detail*
- *Namespace hipcub::internal*

Classes

- *Struct ArgMax*
- *Struct ArgMin*
- *Template Struct block_raking_layout*
- *Struct block_raking_layout::TempStorage*
- *Struct BlockRadixRank::PrefixCallBack*
- *Struct BlockRadixRank::TempStorage*
- *Struct BlockRadixRankMatch::TempStorage*
- *Struct CacheModifiedOutputIterator::Reference*
- *Struct CachingDeviceAllocator*
- *Struct CachingDeviceAllocator::BlockDescriptor*
- *Struct DeviceHistogram*
- *Struct DevicePartition*
- *Struct DeviceRadixSort*
- *Struct DeviceSegmentedRadixSort*
- *Struct DeviceSegmentedReduce*
- *Template Struct DeviceSpmv::SpmvParams*
- *Template Struct DigitExtractor*
- *Template Struct DoubleBuffer*
- *Struct Equality*
- *Template Struct GridEvenShare*
- *Template Struct If*
- *Struct Inequality*
- *Template Struct InequalityWrapper*
- *Template Struct Int2Type*
- *Template Struct IsPointer*
- *Template Struct IsVolatile*
- *Template Struct Log2*
- *Struct Max*
- *Struct Min*
- *Template Struct PowerOfTwo*

- *Template Struct RadixSortTwiddle*
- *Template Struct RemoveQualifiers*
- *Struct Sum*
- *Template Struct Uninitialized*
- *Template Class BlockAdjacentDifference*
- *Template Class BlockDiscontinuity*
- *Template Class BlockExchange*
- *Template Class BlockLoad*
- *Template Class BlockRadixRank*
- *Template Class BlockRadixRankMatch*
- *Template Class BlockRadixSort*
- *Template Class BlockScan*
- *Template Class BlockShuffle*
- *Template Class BlockStore*
- *Template Class CacheModifiedInputIterator*
- *Template Class CacheModifiedOutputIterator*
- *Class CachingDeviceAllocator::TotalBytes*
- *Class DeviceReduce*
- *Class DeviceRunLengthEncode*
- *Class DeviceScan*
- *Class DeviceSelect*
- *Class DeviceSpmv*
- *Template Class DiscardOutputIterator*
- *Class GridBarrier*
- *Class GridBarrierLifetime*
- *Template Class GridQueue*
- *Template Class TexObjInputIterator*
- *Template Class TexRefInputIterator*
- *Template Class WarpReduce*
- *Template Class WarpScan*

Enums

- Enum *BlockLoadAlgorithm*
- Enum *BlockScanAlgorithm*
- Enum *BlockStoreAlgorithm*
- Enum *GridMappingStrategy*

Functions

- Function *hipcub::BAR*
- Template Function *hipcub::BFE*
- Function *hipcub::BFI*
- Function *hipcub::CTA_SYNC*
- Function *hipcub::Debug*
- Template Function *hipcub::DivideAndRoundUp*
- Function *hipcub::IADD3*
- Function *hipcub::LaneId*
- Function *hipcub::LaneMaskGe*
- Function *hipcub::LaneMaskGt*
- Function *hipcub::LaneMaskLe*
- Function *hipcub::LaneMaskLt*
- Template Function *hipcub::LoadDirectBlocked(int, InputIteratorT, T(&))*
- Template Function *hipcub::LoadDirectBlocked(int, InputIteratorT, T(&), int)*
- Template Function *hipcub::LoadDirectBlocked(int, InputIteratorT, T(&), int, Default)*
- Template Function *hipcub::LoadDirectBlockedVectorized*
- Template Function *hipcub::LoadDirectStriped(int, InputIteratorT, T(&))*
- Template Function *hipcub::LoadDirectStriped(int, InputIteratorT, T(&), int)*
- Template Function *hipcub::LoadDirectStriped(int, InputIteratorT, T(&), int, Default)*
- Template Function *hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&))*
- Template Function *hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&), int)*
- Template Function *hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&), int, Default)*
- Function *hipcub::PRMT*
- Function *hipcub::RowMajorTid*
- Function *hipcub::SHL_ADD*
- Function *hipcub::SHR_ADD*
- Template Function *hipcub::ShuffleDown*
- Template Function *hipcub::ShuffleIndex*

- *Template Function hipcub::ShuffleUp*
- *Template Function hipcub::StoreDirectBlocked(int, OutputIteratorT, T(&))*
- *Template Function hipcub::StoreDirectBlocked(int, OutputIteratorT, T(&), int)*
- *Template Function hipcub::StoreDirectBlockedVectorized*
- *Template Function hipcub::StoreDirectStriped(int, OutputIteratorT, T(&))*
- *Template Function hipcub::StoreDirectStriped(int, OutputIteratorT, T(&), int)*
- *Template Function hipcub::StoreDirectWarpStriped(int, OutputIteratorT, T(&))*
- *Template Function hipcub::StoreDirectWarpStriped(int, OutputIteratorT, T(&), int)*
- *Function hipcub::WARP_ALL*
- *Function hipcub::WARP_ANY*
- *Function hipcub::WARP_BALLOT*
- *Function hipcub::WARP_SYNC*
- *Function hipcub::WarpId*

Typedefs

- *Typedef hipcub::KeyValuePair*
- *Typedef hipcub::NullType*

Namespace hipcub::detail

Contents

- *Functions*
- *Typedefs*

Functions

- *Template Function hipcub::detail::convert_result_type*
- *Template Function hipcub::detail::get_lowest_value*
- *Function hipcub::detail::get_lowest_value< __half >*
- *Template Function hipcub::detail::get_max_value*
- *Function hipcub::detail::get_max_value< __half >*
- *Function hipcub::detail::to_BlockLoadAlgorithm_enum*
- *Function hipcub::detail::to_BlockScanAlgorithm_enum*
- *Function hipcub::detail::to_BlockStoreAlgorithm_enum*
- *Template Function hipcub::detail::to_double_buffer*
- *Template Function hipcub::detail::unsigned_bit_extract*

- *Template Function hipcub::detail::update_double_buffer*

Typedefs

- *Typedef hipcub::detail::is_integral_or_enum*

Namespace hipcub::internal

Internal namespace (to prevent ADL mishaps between static functions when mixing different CUB installations)

Contents

- *Functions*

Functions

- *Function hipcub::internal::ThreadScanExclusive*
- *Function hipcub::internal::ThreadScanInclusive*

Namespace internal

Internal namespace (to prevent ADL mishaps between static functions when mixing different CUB installations)

Contents

- *Functions*

Functions

- *Template Function internal::ThreadReduce(T(&), ReductionOp)*
- *Template Function internal::ThreadReduce(T(&), ReductionOp, T)*
- *Template Function internal::ThreadReduce(T *, ReductionOp, T)*

2.1.2 Classes and Structs

Struct ArgMax

- Defined in file_hipcub_backend_rocprim_thread_thread_operators.hpp

Struct Documentation

struct hipcub: **ArgMax**

Public Functions

```
template<class Key, class Value>
__host__ __device__ inline constexpr KeyValuePair<Key, Value> operator()(const KeyValuePair<Key,
                                                                    Value> &a, const
                                                                    KeyValuePair<Key, Value>
                                                                    &b) const
```

Struct ArgMin

- Defined in file `hipcub_backend_rocprim_thread_thread_operators.hpp`

Struct Documentation

struct hipcub: **ArgMin**

Public Functions

```
template<class Key, class Value>
__host__ __device__ inline constexpr KeyValuePair<Key, Value> operator()(const KeyValuePair<Key,
                                                                    Value> &a, const
                                                                    KeyValuePair<Key, Value>
                                                                    &b) const
```

Template Struct `block_raking_layout`

- Defined in file `hipcub_backend_rocprim_block_block_raking_layout.hpp`

Nested Relationships

Nested Types

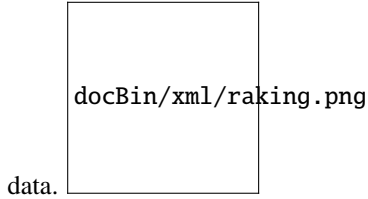
- *Struct `block_raking_layout::TempStorage`*

Struct Documentation

```
template<typename T, int BLOCK_THREADS, int ARCH = HIPCUB_ARCH>
```

```
struct hipcub::block_raking_layout
```

BlockRakingLayout provides a conflict-free shared memory layout abstraction for 1D raking across thread block



Overview This type facilitates a shared memory usage pattern where a block of CUDA threads places elements into shared memory and then reduces the active parallelism to one “raking” warp of threads for serially aggregating consecutive sequences of shared items. Padding is inserted to eliminate bank conflicts (for most data types).

tparam T The data type to be exchanged.

tparam BLOCK_THREADS The thread block size in threads.

tparam PTX_ARCH [optional]

Public Types

```
enum [anonymous]
```

Values:

enumerator **SHARED_ELEMENTS**

The total number of elements that need to be cooperatively reduced.

enumerator **MAX_RAKING_THREADS**

Maximum number of warp-synchronous raking threads.

enumerator **SEGMENT_LENGTH**

Number of raking elements per warp-synchronous raking thread (rounded up)

enumerator **RAKING_THREADS**

Never use a raking thread that will have no valid data (e.g., when **BLOCK_THREADS** is 62 and **SEGMENT_LENGTH** is 2, we should only use 31 raking threads)

enumerator **USE_SEGMENT_PADDING**

Pad each segment length with one element if segment length is not relatively prime to warp size and can't be optimized as a vector load.

enumerator **GRID_ELEMENTS**

Total number of elements in the raking grid.

enumerator **UNGUARDED**

Whether or not we need bounds checking during raking (the number of reduction elements is not a

multiple of the number of raking threads)

Public Functions

inline struct __align__ (16) _TempStorage
Shared memory storage type.

Public Static Functions

`__device__ static inline T *PlacementPtr(TempStorage &temp_storage, unsigned int linear_tid)`
Returns the location for the calling thread to place data into the grid.

`__device__ static inline T *RakingPtr(TempStorage &temp_storage, unsigned int linear_tid)`
Returns the location for the calling thread to begin sequential raking.

struct **TempStorage** : public hipcub::Uninitialized<_TempStorage>
Alias wrapper allowing storage to be unioned.

Public Types

enum [**anonymous**]
Values:

typedef UnitWord<_TempStorage>::DeviceWord **DeviceWord**
Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

`__host__ __device__ __forceinline__ inline _TempStorage &Alias()`
Alias.

Public Members

DeviceWord **storage**[WORDS]
Backing storage.

Struct `block_raking_layout::TempStorage`

- Defined in file `hipcub_backend_rocprim_block_block_raking_layout.hpp`

Nested Relationships

This struct is a nested type of *Template Struct block_raking_layout*.

Inheritance Relationships

Base Type

- `public hipcub::Uninitialized< _TempStorage >` (*Template Struct Uninitialized*)

Struct Documentation

struct hipcub::*block_raking_layout*::**TempStorage** : public hipcub::*Uninitialized*<_TempStorage>
Alias wrapper allowing storage to be unioned.

Public Types

enum [**anonymous**]
Values:

typedef UnitWord<_TempStorage>::DeviceWord **DeviceWord**
Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

`__host__ __device__ __forceinline__ inline _TempStorage &Alias()`
Alias.

Public Members

DeviceWord **storage**[WORDS]
Backing storage.

Struct BlockRadixRank::PrefixCallback

- Defined in file `_hipcub_backend_rocprim_block_block_radix_rank.hpp`

Nested Relationships

This struct is a nested type of *Template Class BlockRadixRank*.

Struct Documentation

```
struct hipcub::BlockRadixRank::PrefixCallback
    Block-scan prefix callback
```

Public Functions

```
__device__ inline PackedCounter operator() (PackedCounter block_aggregate)
```

Struct BlockRadixRank::TempStorage

- Defined in file `hipcub_backend_rocprim_block_block_radix_rank.hpp`

Nested Relationships

This struct is a nested type of *Template Class BlockRadixRank*.

Inheritance Relationships

Base Type

- `public hipcub::Uninitialized< _TempStorage >` (*Template Struct Uninitialized*)

Struct Documentation

```
struct hipcub::BlockRadixRank::TempStorage : public hipcub::Uninitialized<_TempStorage>
    {BlockScan}
```

Public Types

```
enum [anonymous]
    Values:
```

```
typedef UnitWord<_TempStorage>::DeviceWord DeviceWord
    Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.
```

Public Functions

`__host__ __device__ __forceinline__ inline _TempStorage &Alias()`
Alias.

Public Members

DeviceWord **storage**[WORDS]
Backing storage.

Struct BlockRadixRankMatch::TempStorage

- Defined in file `hipcub_backend_rocprim_block_block_radix_rank.hpp`

Nested Relationships

This struct is a nested type of *Template Class BlockRadixRankMatch*.

Inheritance Relationships

Base Type

- `public hipcub::Uninitialized< _TempStorage >` (*Template Struct Uninitialized*)

Struct Documentation

```
struct hipcub::BlockRadixRankMatch::TempStorage : public hipcub::Uninitialized<_TempStorage>
{BlockScan}
```

Public Types

enum **[anonymous]**
Values:

typedef UnitWord<_TempStorage>::DeviceWord **DeviceWord**
Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

`__host__ __device__ __forceinline__ inline _TempStorage &Alias()`
Alias.

Public Members

DeviceWord **storage**[WORDS]
Backing storage.

Struct CacheModifiedOutputIterator::Reference

- Defined in file_hipcub_backend_rocprim_iterator_cache_modified_output_iterator.hpp

Nested Relationships

This struct is a nested type of *Template Class CacheModifiedOutputIterator*.

Struct Documentation

struct hipcub::CacheModifiedOutputIterator::Reference

Public Functions

`__host__ __device__ __forceinline__ inline Reference(ValueType *ptr)`
Constructor.

`__device__ __forceinline__ inline ValueType operator=(ValueType val)`
Assignment.

Public Members

ValueType *ptr

Struct CachingDeviceAllocator

- Defined in file_hipcub_backend_cub_util_allocator.hpp

Nested Relationships

Nested Types

- *Struct CachingDeviceAllocator::BlockDescriptor*
- *Class CachingDeviceAllocator::TotalBytes*

Inheritance Relationships

Base Type

- public CachingDeviceAllocator

Struct Documentation

```
struct hipcub::CachingDeviceAllocator : public CachingDeviceAllocator
```

Public Types

```
typedef bool (*Compare)(const BlockDescriptor&, const BlockDescriptor&)  
BlockDescriptor comparator function interface.
```

```
typedef std::multiset<BlockDescriptor, Compare> CachedBlocks  
Set type for cached blocks (ordered by size)
```

```
typedef std::multiset<BlockDescriptor, Compare> BusyBlocks  
Set type for live blocks (ordered by ptr)
```

```
typedef std::map<int, TotalBytes> GpuCachedBytes  
Map type of device ordinals to the number of cached bytes cached by each device.
```

Public Functions

```
inline hipError_t SetMaxCachedBytes(size_t max_cached_bytes)
```

```
inline hipError_t DeviceAllocate(int device, void **d_ptr, size_t bytes, hipStream_t active_stream = 0)
```

```
inline hipError_t DeviceAllocate(void **d_ptr, size_t bytes, hipStream_t active_stream = 0)
```

```
inline hipError_t DeviceFree(int device, void *d_ptr)
```

```
inline hipError_t DeviceFree(void *d_ptr)
```

```
inline hipError_t FreeAllCached()
```

inline void **NearestPowerOf**(unsigned int &power, size_t &rounded_bytes, unsigned int base, size_t value)
Round up to the nearest power-of

inline **CachingDeviceAllocator**(unsigned int bin_growth, unsigned int min_bin = 1, unsigned int max_bin = *INVALID_BIN*, size_t max_cached_bytes = *INVALID_SIZE*, bool skip_cleanup = false, bool debug = false)

Set of live device allocations currently in use.

Constructor.

Parameters

- **bin_growth** – Geometric growth factor for bin-sizes
- **min_bin** – Minimum bin (default is bin_growth^1)
- **max_bin** – Maximum bin (default is no max bin)
- **max_cached_bytes** – Maximum aggregate cached bytes per device (default is no limit)
- **skip_cleanup** – Whether or not to skip a call to `FreeAllCached()` when the destructor is called (default is to deallocate)
- **debug** – Whether or not to print (de)allocation events to stdout (default is no stderr output)

inline **CachingDeviceAllocator**(bool skip_cleanup = false, bool debug = false)

Default constructor.

Configured with:

which delineates five bin-sizes: 512B, 4KB, 32KB, 256KB, and 2MB and sets a maximum of 6,291,455 cached bytes per device

- $\text{bin_growth} = 8$
- $\text{min_bin} = 3$
- $\text{max_bin} = 7$
- $\text{max_cached_bytes} = (\text{bin_growth}^{\text{max_bin}} * 3) - 1 = 6,291,455$ bytes

inline hipError_t **SetMaxCachedBytes**(size_t max_cached_bytes)

Sets the limit on the number bytes this allocator is allowed to cache per device.

Changing the ceiling of cached bytes does not cause any allocations (in-use or cached-in-reserve) to be freed. See `FreeAllCached()`.

inline hipError_t **DeviceAllocate**(int device, void **d_ptr, size_t bytes, hipStream_t active_stream = 0)

Provides a suitable allocation of device memory for the given size on the specified device.

Once freed, the allocation becomes available immediately for reuse within the `active_stream` with which it was associated with during allocation, and it becomes available for reuse within other streams when all prior work submitted to `active_stream` has completed.

Parameters

- **device** – Device on which to place the allocation
- **d_ptr** – Reference to pointer to the allocation
- **bytes** – Minimum number of bytes for the allocation
- **active_stream** – The stream to be associated with this allocation

inline hipError_t **DeviceAllocate**(void **d_ptr, size_t bytes, hipStream_t active_stream = 0)

Provides a suitable allocation of device memory for the given size on the current device.

Once freed, the allocation becomes available immediately for reuse within the `active_stream` with which it was associated with during allocation, and it becomes available for reuse within other streams when all prior work submitted to `active_stream` has completed.

Parameters

- **d_ptr** – Reference to pointer to the allocation
- **bytes** – Minimum number of bytes for the allocation
- **active_stream** – The stream to be associated with this allocation

inline hipError_t **DeviceFree**(int device, void *d_ptr)

Frees a live allocation of device memory on the specified device, returning it to the allocator.

Once freed, the allocation becomes available immediately for reuse within the `active_stream` with which it was associated with during allocation, and it becomes available for reuse within other streams when all prior work submitted to `active_stream` has completed.

inline hipError_t **DeviceFree**(void *d_ptr)

Frees a live allocation of device memory on the current device, returning it to the allocator.

Once freed, the allocation becomes available immediately for reuse within the `active_stream` with which it was associated with during allocation, and it becomes available for reuse within other streams when all prior work submitted to `active_stream` has completed.

inline hipError_t **FreeAllCached**()

Frees all cached device allocations on all devices.

inline virtual **~CachingDeviceAllocator**()

Destructor.

Public Members

std::mutex **mutex**

unsigned int **bin_growth**

Mutex for thread-safety.

unsigned int **min_bin**

Geometric growth factor for bin-sizes.

unsigned int **max_bin**

Minimum bin enumeration.

size_t **min_bin_bytes**

Maximum bin enumeration.

size_t **max_bin_bytes**

Minimum bin size.

size_t **max_cached_bytes**

Maximum bin size.

const bool **skip_cleanup**

Maximum aggregate cached bytes per device.

bool **debug**

Whether or not to skip a call to FreeAllCached() when destructor is called. (The CUDA runtime may have already shut down for statically declared allocators)

GpuCachedBytes **cached_bytes**

Whether or not to print (de)allocation events to stdout.

CachedBlocks **cached_blocks**

Map of device ordinal to aggregate cached bytes on that device.

BusyBlocks **live_blocks**

Set of cached device allocations available for reuse.

Public Static Functions

static inline unsigned int **IntPow**(unsigned int base, unsigned int exp)

Integer pow function for unsigned base and exponent

Public Static Attributes

static const unsigned int **INVALID_BIN** = (unsigned int)-1

Out-of-bounds bin.

static const size_t **INVALID_SIZE** = (size_t)-1

Invalid size.

static const int **INVALID_DEVICE_ORDINAL** = -1

Invalid device ordinal.

struct **BlockDescriptor**

Descriptor for device memory allocations

Public Functions

inline **BlockDescriptor**(void *d_ptr, int device)

inline **BlockDescriptor**(int device)

Public Members

void ***d_ptr**
size_t **bytes**
unsigned int **bin**
int **device**
hipStream_t **associated_stream**
hipEvent_t **ready_event**

Public Static Functions

static inline bool **PtrCompare**(const *BlockDescriptor* &a, const *BlockDescriptor* &b)

static inline bool **SizeCompare**(const *BlockDescriptor* &a, const *BlockDescriptor* &b)

class **TotalBytes**

Public Functions

inline **TotalBytes**()

Public Members

size_t **free**
size_t **live**

Struct **CachingDeviceAllocator::BlockDescriptor**

- Defined in file `file_hipcub_backend_rocprim_util_allocator.hpp`

Nested Relationships

This struct is a nested type of *Struct CachingDeviceAllocator*.

Struct Documentation

struct hipcub::*CachingDeviceAllocator*::**BlockDescriptor**
 Descriptor for device memory allocations

Public Functions

inline **BlockDescriptor**(void *d_ptr, int device)

inline **BlockDescriptor**(int device)

Public Members

void ***d_ptr**

size_t **bytes**

unsigned int **bin**

int **device**

hipStream_t **associated_stream**

hipEvent_t **ready_event**

Public Static Functions

static inline bool **PtrCompare**(const *BlockDescriptor* &a, const *BlockDescriptor* &b)

static inline bool **SizeCompare**(const *BlockDescriptor* &a, const *BlockDescriptor* &b)

Struct DeviceHistogram

- Defined in file_hipcub_backend_cub_device_device_histogram.hpp

Struct Documentation

struct hipcub::**DeviceHistogram**

Public Static Functions

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
        SampleIteratorT d_samples, CounterT *d_histogram, int
        num_levels, LevelT lower_level, LevelT upper_level,
        OffsetT num_samples, hipStream_t stream = 0, bool
        debug_synchronous = false)
```

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
        SampleIteratorT d_samples, CounterT *d_histogram, int
        num_levels, LevelT lower_level, LevelT upper_level,
        OffsetT num_row_samples, OffsetT num_rows, size_t
        row_stride_bytes, hipStream_t stream = 0, bool
        debug_synchronous = false)
```

```
template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
        SampleIteratorT d_samples, CounterT
        *d_histogram[NUM_ACTIVE_CHANNELS], int
        num_levels[NUM_ACTIVE_CHANNELS], LevelT
        lower_level[NUM_ACTIVE_CHANNELS], LevelT
        upper_level[NUM_ACTIVE_CHANNELS], OffsetT
        num_pixels, hipStream_t stream = 0, bool
        debug_synchronous = false)
```

```
template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
        SampleIteratorT d_samples, CounterT
        *d_histogram[NUM_ACTIVE_CHANNELS], int
        num_levels[NUM_ACTIVE_CHANNELS], LevelT
        lower_level[NUM_ACTIVE_CHANNELS], LevelT
        upper_level[NUM_ACTIVE_CHANNELS], OffsetT
        num_row_pixels, OffsetT num_rows, size_t
        row_stride_bytes, hipStream_t stream = 0, bool
        debug_synchronous = false)
```

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramRange(void *d_temp_storage, size_t &temp_storage_bytes,
        SampleIteratorT d_samples, CounterT *d_histogram, int
        num_levels, LevelT *d_levels, OffsetT num_samples,
        hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
```

```

__host__ static inline hipError_t HistogramRange(void *d_temp_storage, size_t &temp_storage_bytes,
SampleIteratorT d_samples, CounterT *d_histogram, int
num_levels, LevelT *d_levels, OffsetT num_row_samples,
OffsetT num_rows, size_t row_stride_bytes, hipStream_t
stream = 0, bool debug_synchronous = false)

```

```

template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramRange(void *d_temp_storage, size_t
&temp_storage_bytes, SampleIteratorT d_samples,
CounterT
*d_histogram[NUM_ACTIVE_CHANNELS], int
num_levels[NUM_ACTIVE_CHANNELS], LevelT
*d_levels[NUM_ACTIVE_CHANNELS], OffsetT
num_pixels, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramRange(void *d_temp_storage, size_t
&temp_storage_bytes, SampleIteratorT d_samples,
CounterT
*d_histogram[NUM_ACTIVE_CHANNELS], int
num_levels[NUM_ACTIVE_CHANNELS], LevelT
*d_levels[NUM_ACTIVE_CHANNELS], OffsetT
num_row_pixels, OffsetT num_rows, size_t
row_stride_bytes, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
SampleIteratorT d_samples, CounterT *d_histogram, int
num_levels, LevelT lower_level, LevelT upper_level,
OffsetT num_samples, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
SampleIteratorT d_samples, CounterT *d_histogram, int
num_levels, LevelT lower_level, LevelT upper_level,
OffsetT num_row_samples, OffsetT num_rows, size_t
row_stride_bytes, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>

```

```
__host__ static inline hipError_t MultiHistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
    SampleIteratorT d_samples, CounterT
    *d_histogram[NUM_ACTIVE_CHANNELS], int
    num_levels[NUM_ACTIVE_CHANNELS], LevelT
    lower_level[NUM_ACTIVE_CHANNELS], LevelT
    upper_level[NUM_ACTIVE_CHANNELS], OffsetT
    num_pixels, hipStream_t stream = 0, bool
    debug_synchronous = false)
```

```
template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramEven(void *d_temp_storage, size_t &temp_storage_bytes,
    SampleIteratorT d_samples, CounterT
    *d_histogram[NUM_ACTIVE_CHANNELS], int
    num_levels[NUM_ACTIVE_CHANNELS], LevelT
    lower_level[NUM_ACTIVE_CHANNELS], LevelT
    upper_level[NUM_ACTIVE_CHANNELS], OffsetT
    num_row_pixels, OffsetT num_rows, size_t
    row_stride_bytes, hipStream_t stream = 0, bool
    debug_synchronous = false)
```

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramRange(void *d_temp_storage, size_t &temp_storage_bytes,
    SampleIteratorT d_samples, CounterT *d_histogram, int
    num_levels, LevelT *d_levels, OffsetT num_samples,
    hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename SampleIteratorT, typename CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t HistogramRange(void *d_temp_storage, size_t &temp_storage_bytes,
    SampleIteratorT d_samples, CounterT *d_histogram, int
    num_levels, LevelT *d_levels, OffsetT num_row_samples,
    OffsetT num_rows, size_t row_stride_bytes, hipStream_t
    stream = 0, bool debug_synchronous = false)
```

```
template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
__host__ static inline hipError_t MultiHistogramRange(void *d_temp_storage, size_t
    &temp_storage_bytes, SampleIteratorT d_samples,
    CounterT
    *d_histogram[NUM_ACTIVE_CHANNELS], int
    num_levels[NUM_ACTIVE_CHANNELS], LevelT
    *d_levels[NUM_ACTIVE_CHANNELS], OffsetT
    num_pixels, hipStream_t stream = 0, bool
    debug_synchronous = false)
```

```
template<int NUM_CHANNELS, int NUM_ACTIVE_CHANNELS, typename SampleIteratorT, typename
CounterT, typename LevelT, typename OffsetT>
```

```

__host__ static inline hipError_t MultiHistogramRange(void *d_temp_storage, size_t
&temp_storage_bytes, SampleIteratorT d_samples,
CounterT
*d_histogram[NUM_ACTIVE_CHANNELS], int
num_levels[NUM_ACTIVE_CHANNELS], LevelT
*d_levels[NUM_ACTIVE_CHANNELS], OffsetT
num_row_pixels, OffsetT num_rows, size_t
row_stride_bytes, hipStream_t stream = 0, bool
debug_synchronous = false)

```

Struct DevicePartition

- Defined in file `hipcub_backend_cub_device_device_partition.hpp`

Struct Documentation

```
struct hipcub::DevicePartition
```

Public Static Functions

```

template<typename InputIteratorT, typename FlagIterator, typename OutputIteratorT, typename
NumSelectedIteratorT>
__host__ __forceinline__ static inline hipError_t Flagged(void *d_temp_storage, size_t
&temp_storage_bytes, InputIteratorT d_in,
FlagIterator d_flags, OutputIteratorT d_out,
NumSelectedIteratorT d_num_selected_out, int
num_items, hipStream_t stream = 0, bool
debug_synchronous = false)

```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to `temp_storage_bytes` and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of `d_temp_storage` allocation
- **d_in** – Pointer to the input sequence of data items
- **d_flags** – Pointer to the input sequence of selection flags
- **d_out** – Pointer to the output sequence of partitioned data items
- **d_num_selected_out** – Pointer to the output total number of items selected (i.e., the offset of the unselected partition)
- **num_items** – Total number of items to select from
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.

```

template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT,
typename SelectOp>

```

```
__host__ __forceinline__ static inline hipError_t If(void *d_temp_storage, size_t &temp_storage_bytes,
                                                    InputIteratorT d_in, OutputIteratorT d_out,
                                                    NumSelectedIteratorT d_num_selected_out, int
                                                    num_items, SelectOp select_op, hipStream_t stream = 0,
                                                    bool debug_synchronous = false)
```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to **temp_storage_bytes** and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of **d_temp_storage** allocation
- **d_in** – Pointer to the input sequence of data items
- **d_out** – Pointer to the output sequence of partitioned data items
- **d_num_selected_out** – Pointer to the output total number of items selected (i.e., the offset of the unselected partition)
- **num_items** – Total number of items to select from
- **select_op** – Unary selection operator
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.

```
template<typename InputIteratorT, typename FlagIterator, typename OutputIteratorT, typename NumSelectedIteratorT>
```

```
__host__ __forceinline__ static inline hipError_t Flagged(void *d_temp_storage, size_t
                                                         &temp_storage_bytes, InputIteratorT d_in,
                                                         FlagIterator d_flags, OutputIteratorT d_out,
                                                         NumSelectedIteratorT d_num_selected_out, int
                                                         num_items, hipStream_t stream = 0, bool
                                                         debug_synchronous = false)
```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to **temp_storage_bytes** and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of **d_temp_storage** allocation
- **d_in** – Pointer to the input sequence of data items
- **d_flags** – Pointer to the input sequence of selection flags
- **d_out** – Pointer to the output sequence of partitioned data items
- **d_num_selected_out** – Pointer to the output total number of items selected (i.e., the offset of the unselected partition)
- **num_items** – Total number of items to select from
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.

```
template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT,
typename SelectOp>
```

```
__host__ __forceinline__ static inline hipError_t If(void *d_temp_storage, size_t &temp_storage_bytes,
            InputIteratorT d_in, OutputIteratorT d_out,
            NumSelectedIteratorT d_num_selected_out, int
            num_items, SelectOp select_op, hipStream_t stream = 0,
            bool debug_synchronous = false)
```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to **temp_storage_bytes** and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of **d_temp_storage** allocation
- **d_in** – Pointer to the input sequence of data items
- **d_out** – Pointer to the output sequence of partitioned data items
- **d_num_selected_out** – Pointer to the output total number of items selected (i.e., the offset of the unselected partition)
- **num_items** – Total number of items to select from
- **select_op** – Unary selection operator
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.

Struct DeviceRadixSort

- Defined in file `hipcub_backend_cub_device_device_radix_sort.hpp`

Struct Documentation

```
struct hipcub::DeviceRadixSort
```

Public Static Functions

```
template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
            *d_keys_in, KeyT *d_keys_out, const ValueT *d_values_in,
            ValueT *d_values_out, int num_items, int begin_bit = 0, int
            end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
            debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes,
            DoubleBuffer<KeyT> &d_keys, DoubleBuffer<ValueT>
            &d_values, int num_items, int begin_bit = 0, int end_bit =
            sizeof(KeyT) * 8, hipStream_t stream = 0, bool
            debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT>
```

```
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, const KeyT *d_keys_in,
KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int begin_bit
= 0, int end_bit = sizeof(KeyT) * 8, hipStream_t
stream = 0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, DoubleBuffer<KeyT>
&d_keys, DoubleBuffer<ValueT> &d_values, int
num_items, int begin_bit = 0, int end_bit =
sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, int num_items, int begin_bit = 0,
int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int begin_bit = 0,
int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
const KeyT *d_keys_in, KeyT *d_keys_out, int
num_items, int begin_bit = 0, int end_bit =
sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int
begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous =
false)
```

```
template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int begin_bit = 0, int
end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT>
```



```

__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes,
      DoubleBuffer<KeyT> &d_keys, DoubleBuffer<ValueT>
      &d_values, int num_items, int begin_bit = 0, int end_bit =
      sizeof(KeyT) * 8, hipStream_t stream = 0, bool
      debug_synchronous = false)

```

```

template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
      &temp_storage_bytes, const KeyT *d_keys_in,
      KeyT *d_keys_out, const ValueT *d_values_in,
      ValueT *d_values_out, int num_items, int begin_bit
      = 0, int end_bit = sizeof(KeyT) * 8, hipStream_t
      stream = 0, bool debug_synchronous = false)

```

```

template<typename KeyT, typename ValueT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
      &temp_storage_bytes, DoubleBuffer<KeyT>
      &d_keys, DoubleBuffer<ValueT> &d_values, int
      num_items, int begin_bit = 0, int end_bit =
      sizeof(KeyT) * 8, hipStream_t stream = 0, bool
      debug_synchronous = false)

```

```

template<typename KeyT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
      *d_keys_in, KeyT *d_keys_out, int num_items, int begin_bit = 0,
      int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
      debug_synchronous = false)

```

```

template<typename KeyT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes,
      DoubleBuffer<KeyT> &d_keys, int num_items, int begin_bit = 0,
      int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
      debug_synchronous = false)

```

```

template<typename KeyT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
      const KeyT *d_keys_in, KeyT *d_keys_out, int
      num_items, int begin_bit = 0, int end_bit =
      sizeof(KeyT) * 8, hipStream_t stream = 0, bool
      debug_synchronous = false)

```

```

template<typename KeyT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
      DoubleBuffer<KeyT> &d_keys, int num_items, int
      begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
      hipStream_t stream = 0, bool debug_synchronous =
      false)

```

Struct DeviceSegmentedRadixSort

- Defined in file `hipcub_backend_cub_device_device_segmented_radix_sort.hpp`

Struct Documentation

struct `hipcub::DeviceSegmentedRadixSort`

Public Static Functions

```
template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int num_segments,
OffsetIteratorT d_begin_offsets, OffsetIteratorT d_end_offsets,
int begin_bit = 0, int end_bit = sizeof(KeyT) * 8, hipStream_t
stream = 0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, DoubleBuffer<ValueT>
&d_values, int num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int begin_bit =
0, int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, const KeyT *d_keys_in,
KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets,
OffsetIteratorT d_end_offsets, int begin_bit = 0, int
end_bit = sizeof(KeyT) * 8, hipStream_t stream =
0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, DoubleBuffer<KeyT>
&d_keys, DoubleBuffer<ValueT> &d_values, int
num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int
begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous =
false)
```

```
template<typename KeyT, typename OffsetIteratorT>
```

```

__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets, OffsetIteratorT
d_end_offsets, int begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous = false)

```

```

template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets, OffsetIteratorT
d_end_offsets, int begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous = false)

```

```

template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
const KeyT *d_keys_in, KeyT *d_keys_out, int
num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int
begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous =
false)

```

```

template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets,
OffsetIteratorT d_end_offsets, int begin_bit = 0, int
end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0,
bool debug_synchronous = false)

```

```

template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int num_segments,
OffsetIteratorT d_begin_offsets, OffsetIteratorT d_end_offsets,
int begin_bit = 0, int end_bit = sizeof(KeyT) * 8, hipStream_t
stream = 0, bool debug_synchronous = false)

```

```

template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairs(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, DoubleBuffer<ValueT>
&d_values, int num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int begin_bit =
0, int end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<typename KeyT, typename ValueT, typename OffsetIteratorT>

```

```
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, const KeyT *d_keys_in,
KeyT *d_keys_out, const ValueT *d_values_in,
ValueT *d_values_out, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets,
OffsetIteratorT d_end_offsets, int begin_bit = 0, int
end_bit = sizeof(KeyT) * 8, hipStream_t stream =
0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename ValueT, typename OffsetIteratorT>
__host__ static inline hipError_t SortPairsDescending(void *d_temp_storage, size_t
&temp_storage_bytes, DoubleBuffer<KeyT>
&d_keys, DoubleBuffer<ValueT> &d_values, int
num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int
begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous =
false)
```

```
template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes, const KeyT
*d_keys_in, KeyT *d_keys_out, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets, OffsetIteratorT
d_end_offsets, int begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeys(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets, OffsetIteratorT
d_end_offsets, int begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
const KeyT *d_keys_in, KeyT *d_keys_out, int
num_items, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, int
begin_bit = 0, int end_bit = sizeof(KeyT) * 8,
hipStream_t stream = 0, bool debug_synchronous =
false)
```

```
template<typename KeyT, typename OffsetIteratorT>
__host__ static inline hipError_t SortKeysDescending(void *d_temp_storage, size_t &temp_storage_bytes,
DoubleBuffer<KeyT> &d_keys, int num_items, int
num_segments, OffsetIteratorT d_begin_offsets,
OffsetIteratorT d_end_offsets, int begin_bit = 0, int
end_bit = sizeof(KeyT) * 8, hipStream_t stream = 0,
bool debug_synchronous = false)
```

Struct DeviceSegmentedReduce

- Defined in file `hipcub_backend_cub_device_device_segmented_reduce.hpp`

Struct Documentation

struct `hipcub::DeviceSegmentedReduce`

Public Static Functions

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT,
typename ReductionOp, typename T>
__host__ static inline hipError_t Reduce(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, ReductionOp
reduction_op, T initial_value, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Sum(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Min(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t ArgMin(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream
= 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Max(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t ArgMax(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream
= 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT,
typename ReductionOp, typename T>
__host__ static inline hipError_t Reduce(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, ReductionOp
reduction_op, T initial_value, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Sum(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Min(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t ArgMin(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream
= 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t Max(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename OffsetIteratorT>
__host__ static inline hipError_t ArgMax(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_segments, OffsetIteratorT
d_begin_offsets, OffsetIteratorT d_end_offsets, hipStream_t stream
= 0, bool debug_synchronous = false)
```

Template Struct DeviceSpmv::SpmvParams

- Defined in file `hipcub_backend_cub_device_device_spmv.hpp`

Nested Relationships

This struct is a nested type of *Class DeviceSpmv*.

Struct Documentation

```
template<typename ValueT, typename OffsetT>
```

```
struct hipcub::DeviceSpmv::SpmvParams
    < Signed integer type for sequence offsets
```

Public Members

ValueT ***d_values**

Pointer to the array of `num_nonzeros` values of the corresponding nonzero elements of matrix **A**.

OffsetT ***d_row_end_offsets**

Pointer to the array of `m` offsets demarcating the end of every row in `d_column_indices` and `d_values`.

OffsetT ***d_column_indices**

Pointer to the array of `num_nonzeros` column-indices of the corresponding nonzero elements of matrix **A**. (Indices are zero-valued.)

ValueT ***d_vector_x**

Pointer to the array of `num_cols` values corresponding to the dense input vector *x*

ValueT ***d_vector_y**

Pointer to the array of `num_rows` values corresponding to the dense output vector *y*

int **num_rows**

Number of rows of matrix **A**.

int **num_cols**

Number of columns of matrix **A**.

int **num_nonzeros**

Number of nonzero elements of matrix **A**.

ValueT **alpha**

Alpha multiplicand.

ValueT **beta**

Beta addend-multiplicand.

```
::cub::TexRefInputIterator<ValueT, 66778899, OffsetT> t_vector_x
```

```
::hipcub::TexRefInputIterator<ValueT, 66778899, OffsetT> t_vector_x
```

Template Struct DigitExtractor

- Defined in file_hipcub_backend_rocprim_block_radix_rank_sort_operations.hpp

Struct Documentation

```
template<typename UnsignedBits>
```

```
struct hipcub::DigitExtractor  
    Stateful abstraction to extract digits.
```

Public Functions

```
__device__ inline __inline__ DigitExtractor()
```

```
__device__ inline __inline__ DigitExtractor(int current_bit, int num_bits)
```

```
inline __device__ __inline__ int Digit (UnsignedBits key)
```

Public Members

```
int current_bit
```

```
int mask
```

Template Struct DoubleBuffer

- Defined in file_hipcub_backend_rocprim_util_type.hpp

Struct Documentation

```
template<typename T>
```

```
struct hipcub::DoubleBuffer
```


Public Functions

```
__host__ __device__ inline DoubleBuffer()
```

```
__host__ __device__ inline DoubleBuffer(T *d_current, T *d_alternate)
```

```
__host__ __device__ inline T *Current()
```

```
__host__ __device__ inline T *Alternate()
```

Public Members

```
T *d_buffers[2]
```

```
int selector
```

Struct Equality

- Defined in file `hipcub_backend_rocprim_thread_thread_operators.hpp`

Struct Documentation

```
struct hipcub::Equality
```

Public Functions

```
template<class T>
__host__ __device__ inline constexpr bool operator() (const T &a, const T &b) const
```

Template Struct GridEvenShare

- Defined in file `hipcub_backend_rocprim_grid_grid_even_share.hpp`

Struct Documentation

```
template<typename OffsetT>
```

```
struct hipcub::GridEvenShare
```

GridEvenShare is a descriptor utility for distributing input among CUDA thread blocks in an “even-share” fashion. Each thread block gets roughly the same number of input tiles.

Overview Each thread block is assigned a consecutive sequence of input tiles. To help preserve alignment and eliminate the overhead of guarded loads for all but the last thread block, *GridEvenShare* assigns one of three different amounts of work to a given thread block: “big”, “normal”, or “last”. The “big” workloads are one scheduling grain larger than “normal”. The “last” work unit for the last thread block may be partially-full if the input is not an even multiple of the scheduling grain size.

Before invoking a child grid, a parent thread will typically construct an instance of *GridEvenShare*. The instance can be passed to child thread blocks which can initialize their per-thread block offsets using *BlockInit()*.

Public Functions

`__host__ __device__ __forceinline__ inline GridEvenShare()`

Constructor.

`__host__ __device__ __forceinline__ inline void DispatchInit(OffsetT num_items_, int max_grid_size, int tile_items)`

Dispatch initializer. To be called prior to kernel launch.

Parameters

- **num_items_** – Total number of input items
- **max_grid_size** – Maximum grid size allowable (actual grid size may be less if not warranted by the the number of input items)
- **tile_items** – Number of data items per input tile

`template<int TILE_ITEMS>`

`__device__ __forceinline__ inline void BlockInit(int block_id, Int2Type<GRID_MAPPING_RAKE>)`

Initializes ranges for the specified thread block index. Specialized for a “raking” access pattern in which each thread block is assigned a consecutive sequence of input tiles.

`template<int TILE_ITEMS>`

`__device__ __forceinline__ inline void BlockInit(int block_id, Int2Type<GRID_MAPPING_STRIP_MINE>)`

Block-initialization, specialized for a “raking” access pattern in which each thread block is assigned a consecutive sequence of input tiles.

`template<int TILE_ITEMS, GridMappingStrategy STRATEGY>`

`__device__ __forceinline__ inline void BlockInit()`

Block-initialization, specialized for “strip mining” access pattern in which the input tiles assigned to each thread block are separated by a stride equal to the the extent of the grid.

`template<int TILE_ITEMS>`

`__device__ __forceinline__ inline void BlockInit(OffsetT block_offset, OffsetT block_end)`

Block-initialization, specialized for a “raking” access pattern in which each thread block is assigned a consecutive sequence of input tiles.

Parameters

- **block_offset** – Threadblock begin offset (inclusive)
- **block_end** – Threadblock end offset (exclusive)

Public Members

OffsetT **num_items**

Total number of input items.

int **grid_size**

Grid size in thread blocks.

OffsetT **block_offset**

OffsetT into input marking the beginning of the owning thread block's segment of input tiles.

OffsetT **block_end**

OffsetT into input of marking the end (one-past) of the owning thread block's segment of input tiles.

OffsetT **block_stride**

Stride between input tiles.

Template Struct If

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Struct Documentation

```
template<bool B, typename T, typename F>
```

```
struct hipcub::If
```

Public Types

```
using Type = typename std::conditional<B, T, F>::type
```

Struct Inequality

- Defined in file `hipcub_backend_rocprim_thread_thread_operators.hpp`

Struct Documentation

```
struct hipcub::Inequality
```

Public Functions

```
template<class T>
__host__ __device__ inline constexpr bool operator()(const T &a, const T &b) const
```

Template Struct InequalityWrapper

- Defined in file_hipcup_backend_rocprim_thread_thread_operators.hpp

Struct Documentation

```
template<class EqualityOp>
struct hipcup::InequalityWrapper
```

Public Functions

```
__host__ __device__ inline InequalityWrapper(EqualityOp op)
```

```
template<class T>
__host__ __device__ inline bool operator()(const T &a, const T &b)
```

Public Members

EqualityOp **op**

Template Struct Int2Type

- Defined in file_hipcup_backend_rocprim_util_type.hpp

Struct Documentation

```
template<int A>
struct hipcup::Int2Type
```

Public Types

enum **[anonymous]**

Values:

enumerator **VALUE**

Template Struct IsPointer

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Struct Documentation

```
template<typename T>
```

```
struct hipcub::IsPointer
```

Public Static Attributes

```
static constexpr bool VALUE = std::is_pointer<T>::value
```

Template Struct IsVolatile

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Struct Documentation

```
template<typename T>
```

```
struct hipcub::IsVolatile
```

Public Static Attributes

```
static constexpr bool VALUE = std::is_volatile<T>::value
```

Template Struct Log2

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Struct Documentation

template<int N >

struct hipcub::Log2

Public Static Attributes

static constexpr int **VALUE** = detail::Log2Impl< N >::VALUE

Struct Max

- Defined in file_hipcub_backend_rocprim_thread_thread_operators.hpp

Struct Documentation

struct hipcub::Max

Public Functions

template<class T >
__host__ __device__ inline constexpr T **operator**() (const T &a, const T &b) const

Struct Min

- Defined in file_hipcub_backend_rocprim_thread_thread_operators.hpp

Struct Documentation

struct hipcub::Min

Public Functions

template<class T >
__host__ __device__ inline constexpr T **operator**() (const T &a, const T &b) const

Template Struct PowerOfTwo

- Defined in file_hipcub_backend_rocprim_util_type.hpp

Struct Documentation

```
template<int N>
```

```
struct hipcub::PowerOfTwo
```

Public Static Attributes

```
static constexpr bool VALUE = ::rocprim::detail::is_power_of_two<N>()
```

Template Struct RadixSortTwiddle

- Defined in file_hipcub_backend_rocprim_block_radix_rank_sort_operations.hpp

Struct Documentation

```
template<bool IS_DESCENDING, typename KeyT>
```

```
struct hipcub::RadixSortTwiddle
    Twiddling keys for radix sort.
```

Public Types

```
typedef Traits<KeyT> TraitsT
```

```
typedef TraitsT::UnsignedBits UnsignedBits
```

Public Static Functions

```
__host__ __device__ __forceinline__ static inline UnsignedBits In(UnsignedBits key)
```

```
__host__ __device__ __forceinline__ static inline UnsignedBits Out(UnsignedBits key)
```

```
__host__ __device__ __forceinline__ static inline UnsignedBits DefaultKey()
```

Template Struct RemoveQualifiers

- Defined in file_hipcub_backend_rocprim_util_type.hpp

Struct Documentation

```
template<typename T>
```

```
struct hipcub::RemoveQualifiers
```

Public Types

```
using Type = typename std::remove_cv<T>::type
```

Struct Sum

- Defined in file_hipcub_backend_rocprim_thread_thread_operators.hpp

Struct Documentation

```
struct hipcub::Sum
```

Public Functions

```
template<class T>  
__host__ __device__ inline constexpr T operator() (const T &a, const T &b) const
```

Template Struct Uninitialized

- Defined in file_hipcub_backend_rocprim_util_type.hpp

Struct Documentation

```
template<typename T>
```

```
struct hipcub::Uninitialized
```

A storage-backing wrapper that allows types with non-trivial constructors to be aliased in unions.

Public Types

enum [anonymous]

Values:

enumerator **WORDS**

typedef UnitWord<T>::DeviceWord **DeviceWord**

Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

`__host__ __device__ __forceinline__ inline T &Alias()`

Alias.

Public Members

DeviceWord **storage**[*WORDS*]

Backing storage.

Template Class BlockHistogram

- Defined in file_hipcub_backend_rocprim_block_block_histogram.hpp

Inheritance Relationships

Base Type

- private rocprim::block_histogram< T, BLOCK_DIM_X, ITEMS_PER_THREAD, BINS, static_cast<::rocprim::block_histogram_algorithm >(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename T, int BLOCK_DIM_X, int ITEMS_PER_THREAD, int BINS, BlockHistogramAlgorithm ALGORITHM
= BLOCK_HISTO_SORT, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class BlockHistogram : private rocprim::block_histogram<T, BLOCK_DIM_X, ITEMS_PER_THREAD, BINS,
static_cast<::rocprim::block_histogram_algorithm>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockHistogram()
```

```
__device__ inline BlockHistogram(TempStorage &temp_storage)
```

```
template<class CounterT>  
__device__ inline void InitHistogram(CounterT histogram[BINS])
```

```
template<class CounterT>  
__device__ inline void Composite(T (&items)[ITEMS_PER_THREAD], CounterT histogram[BINS])
```

```
template<class CounterT>  
__device__ inline void Histogram(T (&items)[ITEMS_PER_THREAD], CounterT histogram[BINS])
```

Template Class BlockReduce

- Defined in file_hipcub_backend_rocprim_block_block_reduce.hpp

Inheritance Relationships

Base Type

- private rocprim::block_reduce< T, BLOCK_DIM_X, static_cast<::rocprim::block_reduce_algorithm>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename T, int BLOCK_DIM_X, BlockReduceAlgorithm ALGORITHM =  
BLOCK_REDUCE_WARP_REDUCTIONS, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH =  
HIPCUB_ARCH>  
class BlockReduce : private rocprim::block_reduce<T, BLOCK_DIM_X,  
static_cast<::rocprim::block_reduce_algorithm>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockReduce()
```

```
__device__ inline BlockReduce(TempStorage &temp_storage)
```

```
__device__ inline T Sum(T input)
```

```
__device__ inline T Sum(T input, int valid_items)
```

```
template<int ITEMS_PER_THREAD>
```

```
__device__ inline T Sum(T (&input)[ITEMS_PER_THREAD])
```

```
template<typename ReduceOp>
```

```
__device__ inline T Reduce(T input, ReduceOp reduce_op)
```

```
template<typename ReduceOp>
```

```
__device__ inline T Reduce(T input, ReduceOp reduce_op, int valid_items)
```

```
template<int ITEMS_PER_THREAD, typename ReduceOp>
```

```
__device__ inline T Reduce(T (&input)[ITEMS_PER_THREAD], ReduceOp reduce_op)
```

Template Class BlockAdjacentDifference

- Defined in file_hipcup_backend_rocprim_block_block_adjacent_difference.hpp

Inheritance Relationships

Base Type

- private rocprim::block_adjacent_difference< T, BLOCK_DIM_X, BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename T, int BLOCK_DIM_X, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockAdjacentDifference : private rocprim::block_adjacent_difference<T, BLOCK_DIM_X,
BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockAdjacentDifference()
```

```
__device__ inline BlockAdjacentDifference(TempStorage &temp_storage)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeads(FlagT (&head_flags)[ITEMS_PER_THREAD], T
(&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeads(FlagT (&head_flags)[ITEMS_PER_THREAD], T
(&input)[ITEMS_PER_THREAD], FlagOp flag_op, T
tile_predecessor_item)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagTails(FlagT (&tail_flags)[ITEMS_PER_THREAD], T
(&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagTails(FlagT (&tail_flags)[ITEMS_PER_THREAD], T
(&input)[ITEMS_PER_THREAD], FlagOp flag_op, T
tile_successor_item)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], FlagT
(&tail_flags)[ITEMS_PER_THREAD], T
(&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], FlagT
(&tail_flags)[ITEMS_PER_THREAD], T tile_successor_item,
T (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
```

```

__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    tile_predecessor_item, FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op)

template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    tile_predecessor_item, FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T tile_successor_item,
    T (&input)[ITEMS_PER_THREAD], FlagOp flag_op)

```

Template Class BlockDiscontinuity

- Defined in file `hipcub_backend_rocprim_block_block_discontinuity.hpp`

Inheritance Relationships

Base Type

- private `rocprim::block_discontinuity< T, BLOCK_DIM_X, BLOCK_DIM_Y, BLOCK_DIM_Z >`

Class Documentation

```

template<typename T, int BLOCK_DIM_X, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockDiscontinuity : private rocprim::block_discontinuity<T, BLOCK_DIM_X,
    BLOCK_DIM_Y, BLOCK_DIM_Z>

```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```

__device__ inline BlockDiscontinuity()

__device__ inline BlockDiscontinuity(TempStorage &temp_storage)

template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeads(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op)

template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>

```

```
__device__ inline void FlagHeads(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op, T
    tile_predecessor_item)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagTails(FlagT (&tail_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagTails(FlagT (&tail_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op, T
    tile_successor_item)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T tile_successor_item,
    T (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    tile_predecessor_item, FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T
    (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

```
template<int ITEMS_PER_THREAD, typename FlagT, typename FlagOp>
__device__ inline void FlagHeadsAndTails(FlagT (&head_flags)[ITEMS_PER_THREAD], T
    tile_predecessor_item, FlagT
    (&tail_flags)[ITEMS_PER_THREAD], T tile_successor_item,
    T (&input)[ITEMS_PER_THREAD], FlagOp flag_op)
```

Template Class BlockExchange

- Defined in file_hipcub_backend_rocprim_block_block_exchange.hpp

Inheritance Relationships

Base Type

- private rocprim::block_exchange< InputT, BLOCK_DIM_X, ITEMS_PER_THREAD, BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename InputT, int BLOCK_DIM_X, int ITEMS_PER_THREAD, bool WARP_TIME_SLICING = false, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockExchange : private rocprim::block_exchange<InputT, BLOCK_DIM_X,
ITEMS_PER_THREAD, BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockExchange()
```

```
__device__ inline BlockExchange(TempStorage &temp_storage)
```

```
template<typename OutputT>
__device__ inline void StripedToBlocked(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
(&output_items)[ITEMS_PER_THREAD])
```

```
template<typename OutputT>
__device__ inline void BlockedToStriped(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
(&output_items)[ITEMS_PER_THREAD])
```

```
template<typename OutputT>
__device__ inline void WarpStripedToBlocked(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
(&output_items)[ITEMS_PER_THREAD])
```

```
template<typename OutputT>
__device__ inline void BlockedToWarpStriped(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
(&output_items)[ITEMS_PER_THREAD])
```

```
template<typename OutputT, typename OffsetT>
```

```
__device__ inline void ScatterToBlocked(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
                                        (&output_items)[ITEMS_PER_THREAD], OffsetT
                                        (&ranks)[ITEMS_PER_THREAD])
```

```
template<typename OutputT, typename OffsetT>
__device__ inline void ScatterToStriped(InputT (&input_items)[ITEMS_PER_THREAD], OutputT
                                        (&output_items)[ITEMS_PER_THREAD], OffsetT
                                        (&ranks)[ITEMS_PER_THREAD])
```

```
template<typename OutputT, typename OffsetT>
__device__ inline void ScatterToStripedGuarded(InputT (&input_items)[ITEMS_PER_THREAD],
                                                OutputT (&output_items)[ITEMS_PER_THREAD],
                                                OffsetT (&ranks)[ITEMS_PER_THREAD])
```

```
template<typename OutputT, typename OffsetT, typename ValidFlag>
__device__ inline void ScatterToStripedFlagged(InputT (&input_items)[ITEMS_PER_THREAD],
                                                OutputT (&output_items)[ITEMS_PER_THREAD],
                                                OffsetT (&ranks)[ITEMS_PER_THREAD], ValidFlag
                                                (&is_valid)[ITEMS_PER_THREAD])
```

Template Class BlockLoad

- Defined in file `hipcub_backend_rocprim_block_block_load.hpp`

Inheritance Relationships

Base Type

- `private rocprim::block_load< T, BLOCK_DIM_X, ITEMS_PER_THREAD, static_cast<::rocprim::block_load_method >(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z >`

Class Documentation

```
template<typename T, int BLOCK_DIM_X, int ITEMS_PER_THREAD, BlockLoadAlgorithm ALGORITHM =
BLOCK_LOAD_DIRECT, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockLoad: private rocprim::block_load<T, BLOCK_DIM_X, ITEMS_PER_THREAD,
static_cast<::rocprim::block_load_method>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z>
```


Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockLoad()
```

```
__device__ inline BlockLoad(TempStorage &temp_storage)
```

```
template<class InputIteratorT>
```

```
__device__ inline void Load(InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD])
```

```
template<class InputIteratorT>
```

```
__device__ inline void Load(InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
```

```
template<class InputIteratorT, class Default>
```

```
__device__ inline void Load(InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items,
                             Default oob_default)
```

Template Class BlockRadixRank

- Defined in file `hipcub_backend_rocprim_block_block_radix_rank.hpp`

Nested Relationships

Nested Types

- *Struct* `BlockRadixRank::PrefixCallback`
- *Struct* `BlockRadixRank::TempStorage`

Class Documentation

```
template<int BLOCK_DIM_X, int RADIX_BITS, bool IS_DESCENDING, bool MEMOIZE_OUTER_SCAN = false,
         BlockScanAlgorithm INNER_SCAN_ALGORITHM = BLOCK_SCAN_WARP_SCANS, hipSharedMemConfig
SMEM_CONFIG = hipSharedMemBankSizeFourByte, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = 1>
class hipcub::BlockRadixRank
```

BlockRadixRank provides operations for ranking unsigned integer types within a CUDA thread block.

Overview Blah...

- Keys must be in a form suitable for radix ranking (i.e., unsigned bits).
-

Performance Considerations

•

Examples

- **Example 1:** Simple radix rank of 32-bit integer keys

```
#include <hipcub/hipcub.hpp>

template <int BLOCK_THREADS>
__global__ void ExampleKernel(...)
{
```

tparam **BLOCK_DIM_X** The thread block length in threads along the X dimension

tparam **RADIX_BITS** The number of radix bits per digit place

tparam **IS_DESCENDING** Whether or not the sorted-order is high-to-low

tparam **MEMOIZE_OUTER_SCAN** [optional] Whether or not to buffer outer raking scan partials to incur fewer shared memory reads at the expense of higher register pressure (default: true for architectures SM35 and newer, false otherwise). See `BlockScanAlgorithm::BLOCK_SCAN_RAKING_MEMOIZE` for more details.

tparam **INNER_SCAN_ALGORITHM** [optional] The `hipcub::BlockScanAlgorithm` algorithm to use (default: `hipcub::BLOCK_SCAN_WARP_SCANS`)

tparam **SMEM_CONFIG** [optional] Shared memory bank mode (default: `hipSharedMemBankSizeFourByte`)

tparam **BLOCK_DIM_Y** [optional] The thread block length in threads along the Y dimension (default: 1)

tparam **BLOCK_DIM_Z** [optional] The thread block length in threads along the Z dimension (default: 1)

tparam **ARCH** [optional]

Collective constructors

`__device__ inline BlockRadixRank()`

Collective constructor using a private static allocation of shared memory as temporary storage.

`__device__ inline BlockRadixRank(TempStorage &temp_storage)`

Collective constructor using the specified memory allocation as temporary storage.

Parameters `temp_storage` – Reference to memory allocation having layout type *TempStorage*

Raking

```
template<typename UnsignedBits, int KEYS_PER_THREAD>
__device__ inline void RankKeys(UnsignedBits (&keys)[KEYS_PER_THREAD], int
                                (&ranks)[KEYS_PER_THREAD], int current_bit, int num_bits)
```

Rank keys.

Parameters

- **keys** – Keys for this tile
- **ranks** – For each key, the local rank within the tile
- **current_bit** – The least-significant bit position of the current digit to extract
- **num_bits** – The number of bits in the current digit

```
template<typename UnsignedBits, int KEYS_PER_THREAD>
__device__ inline void RankKeys(UnsignedBits (&keys)[KEYS_PER_THREAD], int
                                (&ranks)[KEYS_PER_THREAD], int current_bit, int num_bits, int
                                (&exclusive_digit_prefix)[BINS_TRACKED_PER_THREAD])
```

Rank keys. For the lower RADIX_DIGITS threads, digit counts for each digit are provided for the corresponding thread.

Parameters

- **keys** – Keys for this tile
- **ranks** – For each key, the local rank within the tile (out parameter)
- **current_bit** – The least-significant bit position of the current digit to extract
- **num_bits** – The number of bits in the current digit
- **exclusive_digit_prefix** – The exclusive prefix sum for the digits $[(\text{threadIdx.x} * \text{BINS_TRACKED_PER_THREAD}) \dots (\text{threadIdx.x} * \text{BINS_TRACKED_PER_THREAD}) + \text{BINS_TRACKED_PER_THREAD} - 1]$

Public Types

```
enum [anonymous]
```

Values:

```
enumerator BINS_TRACKED_PER_THREAD
```

Number of bin-starting offsets tracked per thread.

```
struct TempStorage : public hipcub::Uninitialized<_TempStorage>
{ BlockScan }
```

Public Types

enum [**anonymous**]

Values:

typedef UnitWord<_TempStorage>::DeviceWord **DeviceWord**

Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

`__host__ __device__ __forceinline__ inline _TempStorage &Alias()`

Alias.

Public Members

DeviceWord **storage**[WORDS]

Backing storage.

Template Class BlockRadixRankMatch

- Defined in file_hipcup_backend_rocprim_block_block_radix_rank.hpp

Nested Relationships

Nested Types

- *Struct* `BlockRadixRankMatch::TempStorage`

Class Documentation

template<int **BLOCK_DIM_X**, int **RADIX_BITS**, bool **IS_DESCENDING**, *BlockScanAlgorithm* **INNER_SCAN_ALGORITHM**
= **BLOCK_SCAN_WARP_SCANS**, int **BLOCK_DIM_Y** = 1, int **BLOCK_DIM_Z** = 1, int **ARCH** = 1>

class hipcub:::**BlockRadixRankMatch**

Radix-rank using match.any

Collective constructors

`__device__ inline BlockRadixRankMatch(TempStorage &temp_storage)`

Collective constructor using the specified memory allocation as temporary storage.

Parameters `temp_storage` – Reference to memory allocation having layout type *TempStorage*

Raking

```
template<typename UnsignedBits, int KEYS_PER_THREAD>
__device__ inline void RankKeys(UnsignedBits (&keys)[KEYS_PER_THREAD], int
                                (&ranks)[KEYS_PER_THREAD], int current_bit, int num_bits)
```

Rank keys.

Parameters

- **keys** – Keys for this tile
- **ranks** – For each key, the local rank within the tile
- **current_bit** – The least-significant bit position of the current digit to extract
- **num_bits** – The number of bits in the current digit

```
template<typename UnsignedBits, int KEYS_PER_THREAD>
__device__ inline void RankKeys(UnsignedBits (&keys)[KEYS_PER_THREAD], int
                                (&ranks)[KEYS_PER_THREAD], int current_bit, int num_bits, int
                                (&exclusive_digit_prefix)[BINS_TRACKED_PER_THREAD])
```

Rank keys. For the lower RADIX_DIGITS threads, digit counts for each digit are provided for the corresponding thread.

Parameters

- **keys** – Keys for this tile
- **ranks** – For each key, the local rank within the tile (out parameter)
- **current_bit** – The least-significant bit position of the current digit to extract
- **num_bits** – The number of bits in the current digit
- **exclusive_digit_prefix** – The exclusive prefix sum for the digits $[(\text{threadIdx.x} * \text{BINS_TRACKED_PER_THREAD}) \dots (\text{threadIdx.x} * \text{BINS_TRACKED_PER_THREAD}) + \text{BINS_TRACKED_PER_THREAD} - 1]$

Public Types

```
enum [anonymous]
```

Values:

```
enumerator BINS_TRACKED_PER_THREAD
```

Number of bin-starting offsets tracked per thread.

```
struct TempStorage : public hipcub::Uninitialized<_TempStorage>
{ BlockScan }
```

Public Types

enum [**anonymous**]

Values:

typedef UnitWord<_TempStorage>::DeviceWord **DeviceWord**

Biggest memory-access word that T is a whole multiple of and is not larger than the alignment of T.

Public Functions

__host__ __device__ __forceinline__ inline _TempStorage &**Alias**()

Alias.

Public Members

DeviceWord **storage**[WORDS]

Backing storage.

Template Class BlockRadixSort

- Defined in file_hipcub_backend_rocprim_block_block_radix_sort.hpp

Inheritance Relationships

Base Type

- private rocprim::block_radix_sort< KeyT, BLOCK_DIM_X, ITEMS_PER_THREAD, ValueT, BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename KeyT, int BLOCK_DIM_X, int ITEMS_PER_THREAD, typename ValueT = NullType, int
RADIX_BITS = 4, bool MEMOIZE_OUTER_SCAN = true, BlockScanAlgorithm INNER_SCAN_ALGORITHM =
BLOCK_SCAN_WARP_SCANS, hipSharedMemConfig SMEM_CONFIG = hipSharedMemBankSizeFourByte, int
BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int PTX_ARCH = HIPCUB_ARCH>
class hipcub:: BlockRadixSort : private rocprim::block_radix_sort<KeyT, BLOCK_DIM_X,
ITEMS_PER_THREAD, ValueT, BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

using **TempStorage** = typename base_type::storage_type

Public Functions

```
__device__ inline BlockRadixSort()
```

```
__device__ inline BlockRadixSort(TempStorage &temp_storage)
```

```
__device__ inline void Sort(KeyT (&keys)[ITEMS_PER_THREAD], int begin_bit = 0, int end_bit =
    sizeof(KeyT) * 8)
```

```
__device__ inline void Sort(KeyT (&keys)[ITEMS_PER_THREAD], ValueT
    (&values)[ITEMS_PER_THREAD], int begin_bit = 0, int end_bit = sizeof(KeyT)
    * 8)
```

```
__device__ inline void SortDescending(KeyT (&keys)[ITEMS_PER_THREAD], int begin_bit = 0, int
    end_bit = sizeof(KeyT) * 8)
```

```
__device__ inline void SortDescending(KeyT (&keys)[ITEMS_PER_THREAD], ValueT
    (&values)[ITEMS_PER_THREAD], int begin_bit = 0, int end_bit =
    sizeof(KeyT) * 8)
```

```
__device__ inline void SortBlockedToStriped(KeyT (&keys)[ITEMS_PER_THREAD], int begin_bit = 0,
    int end_bit = sizeof(KeyT) * 8)
```

```
__device__ inline void SortBlockedToStriped(KeyT (&keys)[ITEMS_PER_THREAD], ValueT
    (&values)[ITEMS_PER_THREAD], int begin_bit = 0, int
    end_bit = sizeof(KeyT) * 8)
```

```
__device__ inline void SortDescendingBlockedToStriped(KeyT (&keys)[ITEMS_PER_THREAD], int
    begin_bit = 0, int end_bit = sizeof(KeyT) * 8)
```

```
__device__ inline void SortDescendingBlockedToStriped(KeyT (&keys)[ITEMS_PER_THREAD],
    ValueT (&values)[ITEMS_PER_THREAD], int
    begin_bit = 0, int end_bit = sizeof(KeyT) * 8)
```

Template Class BlockScan

- Defined in file_hipcub_backend_rocprim_block_block_scan.hpp

Inheritance Relationships

Base Type

- private rocprim::block_scan< T, BLOCK_DIM_X, static_cast<::rocprim::block_scan_algorithm>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z >

Class Documentation

```
template<typename T, int BLOCK_DIM_X, BlockScanAlgorithm ALGORITHM = BLOCK_SCAN_RAKING, int
BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockScan : private rocprim::block_scan<T, BLOCK_DIM_X,
static_cast<::rocprim::block_scan_algorithm>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockScan()
```

```
__device__ inline BlockScan(TempStorage &temp_storage)
```

```
__device__ inline void InclusiveSum(T input, T &output)
```

```
__device__ inline void InclusiveSum(T input, T &output, T &block_aggregate)
```

```
template<typename BlockPrefixCallbackOp>
```

```
__device__ inline void InclusiveSum(T input, T &output, BlockPrefixCallbackOp
&block_prefix_callback_op)
```

```
template<int ITEMS_PER_THREAD>
```

```
__device__ inline void InclusiveSum(T (&input)[ITEMS_PER_THREAD], T
(&output)[ITEMS_PER_THREAD])
```

```
template<int ITEMS_PER_THREAD>
```

```
__device__ inline void InclusiveSum(T (&input)[ITEMS_PER_THREAD], T
(&output)[ITEMS_PER_THREAD], T &block_aggregate)
```

```
template<int ITEMS_PER_THREAD, typename BlockPrefixCallbackOp>
```



```

__device__ inline void InclusiveSum(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], BlockPrefixCallbackOp
                                     &block_prefix_callback_op)

template<typename ScanOp>
__device__ inline void InclusiveScan(T input, T &output, ScanOp scan_op)

template<typename ScanOp>
__device__ inline void InclusiveScan(T input, T &output, ScanOp scan_op, T &block_aggregate)

template<typename ScanOp, typename BlockPrefixCallbackOp>
__device__ inline void InclusiveScan(T input, T &output, ScanOp scan_op, BlockPrefixCallbackOp
                                     &block_prefix_callback_op)

template<int ITEMS_PER_THREAD, typename ScanOp>
__device__ inline void InclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], ScanOp scan_op)

template<int ITEMS_PER_THREAD, typename ScanOp>
__device__ inline void InclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], ScanOp scan_op, T
                                     &block_aggregate)

template<int ITEMS_PER_THREAD, typename ScanOp, typename BlockPrefixCallbackOp>
__device__ inline void InclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], ScanOp scan_op,
                                     BlockPrefixCallbackOp &block_prefix_callback_op)

__device__ inline void ExclusiveSum(T input, T &output)

__device__ inline void ExclusiveSum(T input, T &output, T &block_aggregate)

template<typename BlockPrefixCallbackOp>
__device__ inline void ExclusiveSum(T input, T &output, BlockPrefixCallbackOp
                                     &block_prefix_callback_op)

template<int ITEMS_PER_THREAD>
__device__ inline void ExclusiveSum(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD])

template<int ITEMS_PER_THREAD>
__device__ inline void ExclusiveSum(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], T &block_aggregate)

template<int ITEMS_PER_THREAD, typename BlockPrefixCallbackOp>

```

```
__device__ inline void ExclusiveSum(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], BlockPrefixCallbackOp
                                     &block_prefix_callback_op)

template<typename ScanOp>
__device__ inline void ExclusiveScan(T input, T &output, T initial_value, ScanOp scan_op)

template<typename ScanOp>
__device__ inline void ExclusiveScan(T input, T &output, T initial_value, ScanOp scan_op, T
                                     &block_aggregate)

template<typename ScanOp, typename BlockPrefixCallbackOp>
__device__ inline void ExclusiveScan(T input, T &output, ScanOp scan_op, BlockPrefixCallbackOp
                                     &block_prefix_callback_op)

template<int ITEMS_PER_THREAD, typename ScanOp>
__device__ inline void ExclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], T initial_value, ScanOp
                                     scan_op)

template<int ITEMS_PER_THREAD, typename ScanOp>
__device__ inline void ExclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], T initial_value, ScanOp
                                     scan_op, T &block_aggregate)

template<int ITEMS_PER_THREAD, typename ScanOp, typename BlockPrefixCallbackOp>
__device__ inline void ExclusiveScan(T (&input)[ITEMS_PER_THREAD], T
                                     (&output)[ITEMS_PER_THREAD], ScanOp scan_op,
                                     BlockPrefixCallbackOp &block_prefix_callback_op)
```

Template Class BlockShuffle

- Defined in file `hipcub_backend_rocprim_block_block_shuffle.hpp`

Inheritance Relationships

Base Type

- `public rocprim::block_shuffle< T, BLOCK_DIM_X, BLOCK_DIM_Y, BLOCK_DIM_Z >`

Class Documentation

```
template<typename T, int BLOCK_DIM_X, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockShuffle : public rocpim::block_shuffle<T, BLOCK_DIM_X, BLOCK_DIM_Y,
BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockShuffle()
```

```
__device__ inline BlockShuffle(TempStorage &temp_storage)
```

Parameters `temp_storage` – Reference to memory allocation having layout type TempStorage

```
__device__ inline void Offset(T input, T &output, int distance = 1)
```

Each *thread* obtains the input provided by *thread*₊. The offset distance may be negative.

•

Parameters

- **input** – The input item from the calling thread (*thread*)
- **output** – The input item from the successor (or predecessor) thread *thread*₊ (may be aliased to **input**). This value is only updated for *thread* when $0 \leq (i + \text{distance}) < \text{BLOCK_THREADS} - 1$
- **distance** – Offset distance (may be negative)

```
__device__ inline void Rotate(T input, T &output, unsigned int distance = 1)
```

Each *thread* obtains the input provided by *thread*₊.

•

Parameters

- **input** – The calling thread's input item
- **output** – The input item from thread *thread*_{(+)%} (may be aliased to **input**). This value is not updated for *thread*_{BLOCK_THREADS-1}
- **distance** – Offset distance ($0 < \text{distance} < \text{BLOCK_THREADS}$)

```
template<int ITEMS_PER_THREAD>
```

```
__device__ inline void Up(T (&input)[ITEMS_PER_THREAD], T (&prev)[ITEMS_PER_THREAD])
```

The thread block rotates its of `input` items, shifting it up by one item.

-
-
-

Parameters

- **input** – The calling thread’s input items
- **prev** – The corresponding predecessor items (may be aliased to `input`). The item `prev[0]` is not updated for `thread0`.

```
template<int ITEMS_PER_THREAD>
__device__ inline void Up(T (&input)[ITEMS_PER_THREAD], T (&prev)[ITEMS_PER_THREAD], T
    &block_suffix)
```

The thread block rotates its of `input` items, shifting it up by one item. All threads receive the `input` provided by `thread`.

-
-
-

Parameters

- **input** – The calling thread’s input items
- **prev** – The corresponding predecessor items (may be aliased to `input`). The item `prev[0]` is not updated for `thread0`.
- **block_suffix** – The item `input[ITEMS_PER_THREAD-1]` from `thread`, provided to all threads

```
template<int ITEMS_PER_THREAD>
__device__ inline void Down(T (&input)[ITEMS_PER_THREAD], T (&next)[ITEMS_PER_THREAD])
```

The thread block rotates its of `input` items, shifting it down by one item.

-
-
-

Parameters

- **input** – The calling thread’s input items

- **next** – The corresponding predecessor items (may be aliased to `input`). The value `next[0]` is not updated for `threadBLOCK_THREADS-1`.

```
template<int ITEMS_PER_THREAD>
__device__ inline void Down(T (&input)[ITEMS_PER_THREAD], T (&next)[ITEMS_PER_THREAD], T
    &block_prefix)
```

The thread block rotates its of input items, shifting it down by one item. All threads receive `input[0]` provided by `thread`.

-
-
-

Parameters

- **input** – The calling thread’s input items
- **next** – The corresponding predecessor items (may be aliased to `input`). The value `next[0]` is not updated for `threadBLOCK_THREADS-1`.
- **block_prefix** – The item `input[0]` from `thread`, provided to all threads

Template Class BlockStore

- Defined in file `hipcub_backend_rocprim_block_block_store.hpp`

Inheritance Relationships

Base Type

- `private rocprim::block_store< T, BLOCK_DIM_X, ITEMS_PER_THREAD, static_cast<::rocprim::block_store_method >(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z >`

Class Documentation

```
template<typename T, int BLOCK_DIM_X, int ITEMS_PER_THREAD, BlockStoreAlgorithm ALGORITHM =
BLOCK_STORE_DIRECT, int BLOCK_DIM_Y = 1, int BLOCK_DIM_Z = 1, int ARCH = HIPCUB_ARCH>
class hipcub::BlockStore : private rocprim::block_store<T, BLOCK_DIM_X, ITEMS_PER_THREAD,
static_cast<::rocprim::block_store_method>(ALGORITHM), BLOCK_DIM_Y, BLOCK_DIM_Z>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline BlockStore()
```

```
__device__ inline BlockStore(TempStorage &temp_storage)
```

```
template<class OutputIteratorT>  
__device__ inline void Store(OutputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD])
```

```
template<class OutputIteratorT>  
__device__ inline void Store(OutputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int  
    valid_items)
```

Template Class CacheModifiedInputIterator

- Defined in file `hipcub_backend_rocprim_iterator_cache_modified_input_iterator.hpp`

Class Documentation

```
template<CacheLoadModifier MODIFIER, typename ValueType, typename OffsetT = ptrdiff_t>  
class hipcub::CacheModifiedInputIterator
```

Public Types

```
typedef CacheModifiedInputIterator self_type  
    My own type.
```

```
typedef OffsetT difference_type  
    Type to express the result of subtracting one iterator from another.
```

```
typedef ValueType value_type  
    The type of the element the iterator can point to.
```

```
typedef ValueType *pointer  
    The type of a pointer to an element the iterator can point to.
```

```
typedef ValueType reference  
    The type of a reference to an element the iterator can point to.
```

```
typedef std::random_access_iterator_tag iterator_category  
    The iterator category.
```

Public Functions

`__host__ __device__ __forceinline__ inline CacheModifiedInputIterator(ValueType *ptr)`
 Constructor.

Parameters `ptr` – Native pointer to wrap

`__host__ __device__ __forceinline__ inline self_type operator++(int)`
 Postfix increment.

`__host__ __device__ __forceinline__ inline self_type operator++()`
 Prefix increment.

`__device__ __forceinline__ inline reference operator*() const`
 Indirection.

template<typename **Distance**>
`__host__ __device__ __forceinline__ inline self_type operator+(Distance n) const`
 Addition.

template<typename **Distance**>
`__host__ __device__ __forceinline__ inline self_type &operator+=(Distance n)`
 Addition assignment.

template<typename **Distance**>
`__host__ __device__ __forceinline__ inline self_type operator-(Distance n) const`
 Subtraction.

template<typename **Distance**>
`__host__ __device__ __forceinline__ inline self_type &operator-=(Distance n)`
 Subtraction assignment.

`__host__ __device__ __forceinline__ inline difference_type operator-(self_type other) const`
 Distance.

template<typename **Distance**>
`__device__ __forceinline__ inline reference operator[] (Distance n) const`
 Array subscript.

`__device__ __forceinline__ inline pointer operator->()`
 Structure dereference.

`__host__ __device__ __forceinline__ inline bool operator==(const self_type &rhs)`
 Equal to.

`__host__ __device__ __forceinline__ inline bool operator!=(const self_type &rhs)`
 Not equal to.

Public Members

ValueType ***ptr**
 Wrapped native pointer.

Friends

```
inline friend friend std::ostream & operator<< (std::ostream &os, const self_type &)
    ostream operator
```

Template Class CacheModifiedOutputIterator

- Defined in file `hipcub_backend_rocprim_iterator_cache_modified_output_iterator.hpp`

Nested Relationships

Nested Types

- *Struct* `CacheModifiedOutputIterator::Reference`

Class Documentation

```
template<CacheStoreModifier MODIFIER, typename ValueType, typename OffsetT = ptrdiff_t>
class hipcub::CacheModifiedOutputIterator
```

Public Types

```
typedef CacheModifiedOutputIterator self_type
    My own type.
```

```
typedef OffsetT difference_type
    Type to express the result of subtracting one iterator from another.
```

```
typedef void value_type
    The type of the element the iterator can point to.
```

```
typedef void pointer
    The type of a pointer to an element the iterator can point to.
```

```
typedef Reference reference
    The type of a reference to an element the iterator can point to.
```

```
typedef std::random_access_iterator_tag iterator_category
    The iterator category.
```


Public Functions

```
template<typename QualifiedValueType>
__host__ __device__ __forceinline__ inline CacheModifiedOutputIterator(QualifiedValueType *ptr)
    Constructor.
```

Parameters *ptr* – Native pointer to wrap

```
__host__ __device__ __forceinline__ inline self_type operator++(int)
    Postfix increment.
```

```
__host__ __device__ __forceinline__ inline self_type operator++()
    Prefix increment.
```

```
__host__ __device__ __forceinline__ inline reference operator*() const
    Indirection.
```

```
template<typename Distance>
__host__ __device__ __forceinline__ inline self_type operator+(Distance n) const
    Addition.
```

```
template<typename Distance>
__host__ __device__ __forceinline__ inline self_type &operator+=(Distance n)
    Addition assignment.
```

```
template<typename Distance>
__host__ __device__ __forceinline__ inline self_type operator-(Distance n) const
    Subtraction.
```

```
template<typename Distance>
__host__ __device__ __forceinline__ inline self_type &operator--(Distance n)
    Subtraction assignment.
```

```
__host__ __device__ __forceinline__ inline difference_type operator-(self_type other) const
    Distance.
```

```
template<typename Distance>
__host__ __device__ __forceinline__ inline reference operator[]( Distance n) const
    Array subscript.
```

```
__host__ __device__ __forceinline__ inline bool operator==(const self_type &rhs)
    Equal to.
```

```
__host__ __device__ __forceinline__ inline bool operator!=(const self_type &rhs)
    Not equal to.
```

Friends

```
inline friend friend std::ostream & operator<< (std::ostream &os,
const self_type &itr)
    ostream operator
```

Class `CachingDeviceAllocator::TotalBytes`

- Defined in file `hipcub_backend_rocprim_util_allocator.hpp`

Nested Relationships

This class is a nested type of *Struct* `CachingDeviceAllocator`.

Class Documentation

```
class hipcub::CachingDeviceAllocator::TotalBytes
```

Public Functions

```
inline TotalBytes()
```

Public Members

```
size_t free
```

```
size_t live
```

Class `DeviceReduce`

- Defined in file `hipcub_backend_cub_device_device_reduce.hpp`

Class Documentation

```
class hipcub::DeviceReduce
```

Public Static Functions

```
template<typename InputIteratorT, typename OutputIteratorT, typename ReduceOpT, typename T>  
__host__ static inline hipError_t Reduce(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT  
d_in, OutputIteratorT d_out, int num_items, ReduceOpT  
reduction_op, T init, hipStream_t stream = 0, bool  
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>  
__host__ static inline hipError_t Sum(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT  
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,  
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
```

```
__host__ static inline hipError_t Min(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t ArgMin(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t Max(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t ArgMax(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename KeysInputIteratorT, typename UniqueOutputIteratorT, typename
ValuesInputIteratorT, typename AggregatesOutputIteratorT, typename
NumRunsOutputIteratorT, typename ReductionOpT>
__host__ static inline hipError_t ReduceByKey(void *d_temp_storage, size_t &temp_storage_bytes,
KeysInputIteratorT d_keys_in, UniqueOutputIteratorT
d_unique_out, ValuesInputIteratorT d_values_in,
AggregatesOutputIteratorT d_aggregates_out,
NumRunsOutputIteratorT d_num_runs_out, ReductionOpT
reduction_op, int num_items, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename ReduceOpT, typename T>
__host__ static inline hipError_t Reduce(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, ReduceOpT
reduction_op, T init, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t Sum(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t Min(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
```

```
__host__ static inline hipError_t ArgMin(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t Max(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t ArgMax(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, int num_items, hipStream_t stream = 0,
bool debug_synchronous = false)
```

```
template<typename KeysInputIteratorT, typename UniqueOutputIteratorT, typename
ValuesInputIteratorT, typename AggregatesOutputIteratorT, typename
NumRunsOutputIteratorT, typename ReductionOpT>
__host__ static inline hipError_t ReduceByKey(void *d_temp_storage, size_t &temp_storage_bytes,
KeysInputIteratorT d_keys_in, UniqueOutputIteratorT
d_unique_out, ValuesInputIteratorT d_values_in,
AggregatesOutputIteratorT d_aggregates_out,
NumRunsOutputIteratorT d_num_runs_out, ReductionOpT
reduction_op, int num_items, hipStream_t stream = 0, bool
debug_synchronous = false)
```

Class DeviceRunLengthEncode

- Defined in file_hipcub_backend_cub_device_device_run_length_encode.hpp

Class Documentation

```
class hipcub::DeviceRunLengthEncode
```

Public Static Functions

```
template<typename InputIteratorT, typename UniqueOutputIteratorT, typename
LengthsOutputIteratorT, typename NumRunsOutputIteratorT>
__host__ static inline hipError_t Encode(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, UniqueOutputIteratorT d_unique_out, LengthsOutputIteratorT
d_counts_out, NumRunsOutputIteratorT d_num_runs_out, int
num_items, hipStream_t stream = 0, bool debug_synchronous =
false)
```

```
template<typename InputIteratorT, typename OffsetsOutputIteratorT, typename
LengthsOutputIteratorT, typename NumRunsOutputIteratorT>
```

```

__host__ static inline hipError_t NonTrivialRuns(void *d_temp_storage, size_t &temp_storage_bytes,
InputIteratorT d_in, OffsetsOutputIteratorT
d_offsets_out, LengthsOutputIteratorT d_lengths_out,
NumRunsOutputIteratorT d_num_runs_out, int
num_items, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<typename InputIteratorT, typename UniqueOutputIteratorT, typename
LengthsOutputIteratorT, typename NumRunsOutputIteratorT>
__host__ static inline hipError_t Encode(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, UniqueOutputIteratorT d_unique_out, LengthsOutputIteratorT
d_counts_out, NumRunsOutputIteratorT d_num_runs_out, int
num_items, hipStream_t stream = 0, bool debug_synchronous =
false)

```

```

template<typename InputIteratorT, typename OffsetsOutputIteratorT, typename
LengthsOutputIteratorT, typename NumRunsOutputIteratorT>
__host__ static inline hipError_t NonTrivialRuns(void *d_temp_storage, size_t &temp_storage_bytes,
InputIteratorT d_in, OffsetsOutputIteratorT
d_offsets_out, LengthsOutputIteratorT d_lengths_out,
NumRunsOutputIteratorT d_num_runs_out, int
num_items, hipStream_t stream = 0, bool
debug_synchronous = false)

```

Class DeviceScan

- Defined in file_hipcub_backend_cub_device_device_scan.hpp

Class Documentation

```
class hipcub::DeviceScan
```

Public Static Functions

```

template<typename InputIteratorT, typename OutputIteratorT>
__host__ static inline hipError_t InclusiveSum(void *d_temp_storage, size_t &temp_storage_bytes,
InputIteratorT d_in, OutputIteratorT d_out, int num_items,
hipStream_t stream = 0, bool debug_synchronous = false)

```

```

template<typename InputIteratorT, typename OutputIteratorT, typename ScanOpT>
__host__ static inline hipError_t InclusiveScan(void *d_temp_storage, size_t &temp_storage_bytes,
InputIteratorT d_in, OutputIteratorT d_out, ScanOpT
scan_op, int num_items, hipStream_t stream = 0, bool
debug_synchronous = false)

```

```

template<typename InputIteratorT, typename OutputIteratorT>

```

```
__host__ static inline hipError_t ExclusiveSum(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, int num_items,  
                                                hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename ScanOpT, typename  
InitValueT>  
__host__ static inline hipError_t ExclusiveScan(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, ScanOpT  
scan_op, InitValueT init_value, int num_items,  
                                                hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>  
__host__ static inline hipError_t InclusiveSum(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, int num_items,  
                                                hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename ScanOpT>  
__host__ static inline hipError_t InclusiveScan(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, ScanOpT  
scan_op, int num_items, hipStream_t stream = 0, bool  
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT>  
__host__ static inline hipError_t ExclusiveSum(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, int num_items,  
                                                hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename ScanOpT, typename  
InitValueT>  
__host__ static inline hipError_t ExclusiveScan(void *d_temp_storage, size_t &temp_storage_bytes,  
                                                InputIteratorT d_in, OutputIteratorT d_out, ScanOpT  
scan_op, InitValueT init_value, int num_items,  
                                                hipStream_t stream = 0, bool debug_synchronous = false)
```

Class DeviceSelect

- Defined in file `hipcub_backend_cub_device_device_select.hpp`

Class Documentation

```
class hipcub::DeviceSelect
```

Public Static Functions

```
template<typename InputIteratorT, typename FlagIterator, typename OutputIteratorT, typename
NumSelectedIteratorT>
__host__ static inline hipError_t Flagged(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, FlagIterator d_flags, OutputIteratorT d_out,
NumSelectedIteratorT d_num_selected_out, int num_items,
hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT,
typename SelectOp>
__host__ static inline hipError_t If(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT d_in,
OutputIteratorT d_out, NumSelectedIteratorT d_num_selected_out, int
num_items, SelectOp select_op, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT>
__host__ static inline hipError_t Unique(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, NumSelectedIteratorT
d_num_selected_out, int num_items, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename FlagIterator, typename OutputIteratorT, typename
NumSelectedIteratorT>
__host__ static inline hipError_t Flagged(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, FlagIterator d_flags, OutputIteratorT d_out,
NumSelectedIteratorT d_num_selected_out, int num_items,
hipStream_t stream = 0, bool debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT,
typename SelectOp>
__host__ static inline hipError_t If(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT d_in,
OutputIteratorT d_out, NumSelectedIteratorT d_num_selected_out, int
num_items, SelectOp select_op, hipStream_t stream = 0, bool
debug_synchronous = false)
```

```
template<typename InputIteratorT, typename OutputIteratorT, typename NumSelectedIteratorT>
__host__ static inline hipError_t Unique(void *d_temp_storage, size_t &temp_storage_bytes, InputIteratorT
d_in, OutputIteratorT d_out, NumSelectedIteratorT
d_num_selected_out, int num_items, hipStream_t stream = 0, bool
debug_synchronous = false)
```

Class DeviceSpmv

- Defined in file `hipcub_backend_cub_device_device_spmv.hpp`

Nested Relationships

Nested Types

- *Template Struct* `DeviceSpmv::SpmvParams`

Class Documentation

class hipcub::DeviceSpmv

Public Static Functions

```
template<typename ValueT>
__host__ static inline hipError_t CsrMV(void *d_temp_storage, size_t &temp_storage_bytes, ValueT
                                         *d_values, int *d_row_offsets, int *d_column_indices, ValueT
                                         *d_vector_x, ValueT *d_vector_y, int num_rows, int num_cols, int
                                         num_nonzeros, hipStream_t stream = 0, bool debug_synchronous =
                                         false)
```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to **temp_storage_bytes** and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of **d_temp_storage** allocation
- **d_values** – Pointer to the array of **num_nonzeros** values of the corresponding nonzero elements of matrix **A**.
- **d_row_offsets** – Pointer to the array of $m + 1$ offsets demarcating the start of every row in **d_column_indices** and **d_values** (with the final entry being equal to **num_nonzeros**)
- **d_column_indices** – Pointer to the array of **num_nonzeros** column-indices of the corresponding nonzero elements of matrix **A**. (Indices are zero-valued.)
- **d_vector_x** – Pointer to the array of **num_cols** values corresponding to the dense input vector x
- **d_vector_y** – Pointer to the array of **num_rows** values corresponding to the dense output vector y
- **num_rows** – number of rows of matrix **A**.
- **num_cols** – number of columns of matrix **A**.
- **num_nonzeros** – number of nonzero elements of matrix **A**.
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.


```
template<typename ValueT>
__global__ static inline void CsrMVKernel(SpmvParams<ValueT, int> spmv_params)
```

```
template<typename ValueT>
__host__ static inline hipError_t CsrMV(void *d_temp_storage, size_t &temp_storage_bytes, ValueT
*d_values, int *d_row_offsets, int *d_column_indices, ValueT
*d_vector_x, ValueT *d_vector_y, int num_rows, int num_cols, int
num_nonzeros, hipStream_t stream = 0, bool debug_synchronous =
false)
```

Parameters

- **d_temp_storage** – Device-accessible allocation of temporary storage. When NULL, the required allocation size is written to **temp_storage_bytes** and no work is done.
- **temp_storage_bytes** – Reference to size in bytes of **d_temp_storage** allocation
- **d_values** – Pointer to the array of **num_nonzeros** values of the corresponding nonzero elements of matrix **A**.
- **d_row_offsets** – Pointer to the array of **m + 1** offsets demarcating the start of every row in **d_column_indices** and **d_values** (with the final entry being equal to **num_nonzeros**)
- **d_column_indices** – Pointer to the array of **num_nonzeros** column-indices of the corresponding nonzero elements of matrix **A**. (Indices are zero-valued.)
- **d_vector_x** – Pointer to the array of **num_cols** values corresponding to the dense input vector *x*
- **d_vector_y** – Pointer to the array of **num_rows** values corresponding to the dense output vector *y*
- **num_rows** – number of rows of matrix **A**.
- **num_cols** – number of columns of matrix **A**.
- **num_nonzeros** – number of nonzero elements of matrix **A**.
- **stream** – [optional] hip stream to launch kernels within. Default is `stream0`.
- **debug_synchronous** – [optional] Whether or not to synchronize the stream after every kernel launch to check for errors. May cause significant slowdown. Default is `false`.

Public Static Attributes

```
static constexpr uint32_t CsrMVKernel_MaxThreads = 256
```

```
template<typename ValueT, typename OffsetT>
```

```
struct SpmvParams
< Signed integer type for sequence offsets
```

Public Members

ValueT ***d_values**

Pointer to the array of `num_nonzeros` values of the corresponding nonzero elements of matrix **A**.

OffsetT ***d_row_end_offsets**

Pointer to the array of `m` offsets demarcating the end of every row in `d_column_indices` and `d_values`.

OffsetT ***d_column_indices**

Pointer to the array of `num_nonzeros` column-indices of the corresponding nonzero elements of matrix **A**. (Indices are zero-valued.)

ValueT ***d_vector_x**

Pointer to the array of `num_cols` values corresponding to the dense input vector *x*

ValueT ***d_vector_y**

Pointer to the array of `num_rows` values corresponding to the dense output vector *y*

int **num_rows**

Number of rows of matrix **A**.

int **num_cols**

Number of columns of matrix **A**.

int **num_nonzeros**

Number of nonzero elements of matrix **A**.

ValueT **alpha**

Alpha multiplicand.

ValueT **beta**

Beta addend-multiplicand.

`::cub::TexRefInputIterator<ValueT, 66778899, OffsetT> t_vector_x`

`::hipcub::TexRefInputIterator<ValueT, 66778899, OffsetT> t_vector_x`

Template Class DiscardOutputIterator

- Defined in file `hipcub_backend_rocprim_iterator_discard_output_iterator.hpp`

Class Documentation

```
template<typename OffsetT = ptrdiff_t>
```

```
class hipcub::DiscardOutputIterator  
    A discard iterator.
```

Public Types

```
typedef DiscardOutputIterator self_type
```

My own type.

ostream operator

Not equal to.

Equal to.

Array subscript.

Distance.

Subtraction assignment.

Addition assignment.

Addition.

Indirection.

Postfix increment.

```
typedef OffsetT difference_type
```

Type to express the result of subtracting one iterator from another.

```
typedef void value_type
```

The type of the element the iterator can point to.

```
typedef void pointer
```

The type of a pointer to an element the iterator can point to.

```
typedef void reference
```

The type of a reference to an element the iterator can point to.

```
typedef std::random_access_iterator_tag iterator_category
```

The iterator category.

Public Functions

__host__ __device__ __forceinline__ inline **DiscardOutputIterator**(*OffsetT* offset = 0)
Constructor.

Parameters *offset* – Base offset

__host__ __device__ __forceinline__ inline *self_type* **operator++**(int)

__host__ __device__ __forceinline__ inline *self_type* **operator++**()

__host__ __device__ __forceinline__ inline *self_type* &**operator***()

template<typename **Distance**>
__host__ __device__ __forceinline__ inline *self_type* **operator+**(*Distance* n) const

template<typename **Distance**>
__host__ __device__ __forceinline__ inline *self_type* &**operator+=**(*Distance* n)

template<typename **Distance**>
__host__ __device__ __forceinline__ inline *self_type* **operator-**(*Distance* n) const

template<typename **Distance**>
__host__ __device__ __forceinline__ inline *self_type* &**operator--**(*Distance* n)

__host__ __device__ __forceinline__ inline *difference_type* **operator-**(*self_type* other) const

template<typename **Distance**>
__host__ __device__ __forceinline__ inline *self_type* &**operator**[](*Distance*)

__host__ __device__ __forceinline__ inline *pointer* **operator->**()
Structure dereference.

template<typename **T**>
__host__ __device__ __forceinline__ inline void **operator=**(*T* const&)
Assignment to anything else (no-op)

__host__ __device__ __forceinline__ inline **operator** void*() const
Cast to void* operator.

__host__ __device__ __forceinline__ inline bool **operator==**(const *self_type* &rhs)

__host__ __device__ __forceinline__ inline bool **operator!=**(const *self_type* &rhs)

Friends

```
inline friend friend std::ostream & operator<< (std::ostream &os,
const self_type &itr)
```

Class GridBarrier

- Defined in file `hipcub_backend_rocprim_grid_grid_barrier.hpp`

Inheritance Relationships

Derived Type

- public `hipcub::GridBarrierLifetime` (*Class `GridBarrierLifetime`*)

Class Documentation

class `hipcub::GridBarrier`

GridBarrier implements a software global barrier among thread blocks within a hip grid.

Subclassed by *hipcub::GridBarrierLifetime*

Public Functions

```
inline GridBarrier()
```

Constructor

```
__device__ __forceinline__ inline void Sync() const
```

Protected Types

```
typedef unsigned int SyncFlag
```

Synchronize.

Protected Attributes

```
SyncFlag *d_sync
```

Class GridBarrierLifetime

- Defined in file `hipcub_backend_rocprim_grid_grid_barrier.hpp`

Inheritance Relationships

Base Type

- `public hipcub::GridBarrier` (*Class GridBarrier*)

Class Documentation

class `hipcub::GridBarrierLifetime` : public `hipcub::GridBarrier`
GridBarrierLifetime extends *GridBarrier* to provide lifetime management of the temporary device storage needed for cooperation.

Uses RAII for lifetime, i.e., device resources are reclaimed when the destructor is called.

Public Functions

inline `GridBarrierLifetime()`

Constructor

inline `hipError_t HostReset()`

DeviceFrees and resets the progress counters

inline virtual `~GridBarrierLifetime()`

Destructor

inline `hipError_t Setup(int sweep_grid_size)`

Sets up the progress counters for the next kernel launch (lazily allocating and initializing them if necessary)

`__device__ __forceinline__ inline void Sync() const`

Protected Types

typedef unsigned int `SyncFlag`

Synchronize.

Protected Attributes

size_t `sync_bytes`

SyncFlag *`d_sync`

Template Class GridQueue

- Defined in file_hipcub_backend_rocprim_grid_grid_queue.hpp

Class Documentation

template<typename **OffsetT**>

class hipcub::GridQueue

GridQueue is a descriptor utility for dynamic queue management.

Overview *GridQueue* descriptors provides abstractions for “filling” or “draining” globally-shared vectors.

A “filling” *GridQueue* works by atomically-adding to a zero-initialized counter, returning a unique offset for the calling thread to write its items. The *GridQueue* maintains the total “fill-size”. The fill counter must be reset using *GridQueue::ResetFill* by the host or kernel instance prior to the kernel instance that will be filling.

Similarly, a “draining” *GridQueue* works by works by atomically-incrementing a zero-initialized counter, returning a unique offset for the calling thread to read its items. Threads can safely drain until the array’s logical fill-size is exceeded. The drain counter must be reset using *GridQueue::ResetDrain* or *GridQueue::FillAndResetDrain* by the host or kernel instance prior to the kernel instance that will be filling. (For dynamic work distribution of existing data, the corresponding fill-size is simply the number of elements in the array.)

Iterative work management can be implemented simply with a pair of flip-flopping work buffers, each with an associated set of fill and drain *GridQueue* descriptors.

tparam OffsetT Signed integer type for global offsets

Public Functions

__host__ __device__ __forceinline__ inline **GridQueue**()

Constructs an invalid *GridQueue* descriptor.

__host__ __device__ __forceinline__ inline **GridQueue**(void *d_storage)

Constructs a *GridQueue* descriptor around the device storage allocation.

Parameters d_storage – Device allocation to back the *GridQueue*. Must be at least as big as *AllocationSize()*.

__device__ inline hipError_t **FillAndResetDrain**(*OffsetT* fill_size, hipStream_t stream = 0)

This operation sets the fill-size and resets the drain counter, preparing the *GridQueue* for draining in the next kernel instance. To be called by the host or by a kernel prior to that which will be draining.

__host__ inline hipError_t **FillAndResetDrain**(*OffsetT* fill_size, hipStream_t stream = 0)

__device__ inline hipError_t **ResetDrain**(hipStream_t stream = 0)

This operation resets the drain so that it may advance to meet the existing fill-size. To be called by the host or by a kernel prior to that which will be draining.

__host__ inline hipError_t **ResetDrain**(hipStream_t stream = 0)

`__device__ inline hipError_t ResetFill(hipStream_t stream = 0)`
This operation resets the fill counter. To be called by the host or by a kernel prior to that which will be filling.

`__host__ inline hipError_t ResetFill(hipStream_t stream = 0)`

`__device__ inline hipError_t FillSize(OffsetT &fill_size, hipStream_t stream = 0)`
Returns the fill-size established by the parent or by the previous kernel.

`__host__ inline hipError_t FillSize(OffsetT &fill_size, hipStream_t stream = 0)`

`__device__ inline OffsetT Drain(OffsetT num_items)`
Drain `num_items` from the queue. Returns offset from which to read items. To be called from hip kernel.

`__device__ inline OffsetT Fill(OffsetT num_items)`
Fill `num_items` into the queue. Returns offset from which to write items. To be called from hip kernel.

Public Static Functions

`__host__ __device__ __forceinline__ static inline size_t AllocationSize()`
Returns the device allocation size in bytes needed to construct a *GridQueue* instance.

Template Class `TexObjInputIterator`

- Defined in file `hipcub_backend_rocprim_iterator_tex_obj_input_iterator.hpp`

Inheritance Relationships

Base Type

- `public rocprim::texture_cache_iterator< T, OffsetT >`

Class Documentation

```
template<typename T, typename OffsetT = std::ptrdiff_t>
```

```
class hipcub::TexObjInputIterator : public rocprim::texture_cache_iterator<T, OffsetT>
```

Public Functions

```
template<class Qualified>  
inline hipError_t BindTexture(Qualified *ptr, size_t bytes = size_t(-1), size_t texture_offset = 0)
```

```
inline hipError_t UnbindTexture()
```

```
__host__ __device__ inline ~TexObjInputIterator() = default
```

```
__host__ __device__ inline TexObjInputIterator()
```



```
__host__ __device__ inline TexObjInputIterator(const ::rocprim::texture_cache_iterator<T, OffsetT>
                                               other)
```

Template Class TexRefInputIterator

- Defined in file_hipcub_backend_rocprim_iterator_tex_ref_input_iterator.hpp

Inheritance Relationships

Base Type

- public rocprim::texture_cache_iterator< T, OffsetT >

Class Documentation

```
template<typename T, int UNIQUE_ID, typename OffsetT = std::ptrdiff_t>
```

```
class hipcub:: TexRefInputIterator : public rocprim::texture_cache_iterator<T, OffsetT>
```

Public Functions

```
template<class Qualified>
```

```
inline hipError_t BindTexture(Qualified *ptr, size_t bytes = size_t(-1), size_t texture_offset = 0)
```

```
inline hipError_t UnbindTexture()
```

```
__host__ __device__ inline ~TexRefInputIterator() = default
```

```
__host__ __device__ inline TexRefInputIterator()
```

```
__host__ __device__ inline TexRefInputIterator(const ::rocprim::texture_cache_iterator<T, OffsetT>
                                               other)
```

Template Class WarpReduce

- Defined in file_hipcub_backend_rocprim_warp_warp_reduce.hpp

Inheritance Relationships

Base Type

- private rocprim::warp_reduce< T, LOGICAL_WARP_THREADS >

Class Documentation

```
template<typename T, int LOGICAL_WARP_THREADS = ::rocprim::device_warp_size(), int ARCH = 1>
```

```
class hipcub::WarpReduce : private rocprim::warp_reduce<T, LOGICAL_WARP_THREADS>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline WarpReduce(TempStorage &temp_storage)
```

```
__device__ inline T Sum(T input)
```

```
__device__ inline T Sum(T input, int valid_items)
```

```
template<typename FlagT>  
__device__ inline T HeadSegmentedSum(T input, FlagT head_flag)
```

```
template<typename FlagT>  
__device__ inline T TailSegmentedSum(T input, FlagT tail_flag)
```

```
template<typename ReduceOp>  
__device__ inline T Reduce(T input, ReduceOp reduce_op)
```

```
template<typename ReduceOp>  
__device__ inline T Reduce(T input, ReduceOp reduce_op, int valid_items)
```

```
template<typename ReduceOp, typename FlagT>  
__device__ inline T HeadSegmentedReduce(T input, FlagT head_flag, ReduceOp reduce_op)
```

```
template<typename ReduceOp, typename FlagT>  
__device__ inline T TailSegmentedReduce(T input, FlagT tail_flag, ReduceOp reduce_op)
```

Template Class WarpScan

- Defined in file `hipcub_backend_rocprim_warp_warp_scan.hpp`

Inheritance Relationships

Base Type

- `private rocprim::warp_scan< T, LOGICAL_WARP_THREADS >`

Class Documentation

```
template<typename T, int LOGICAL_WARP_THREADS = ::rocprim::device_warp_size(), int ARCH = 1>
```

```
class hipcub::WarpScan : private rocprim::warp_scan<T, LOGICAL_WARP_THREADS>
```

Public Types

```
using TempStorage = typename base_type::storage_type
```

Public Functions

```
__device__ inline WarpScan(TempStorage &temp_storage)
```

```
__device__ inline void InclusiveSum(T input, T &inclusive_output)
```

```
__device__ inline void InclusiveSum(T input, T &inclusive_output, T &warp_aggregate)
```

```
__device__ inline void ExclusiveSum(T input, T &exclusive_output)
```

```
__device__ inline void ExclusiveSum(T input, T &exclusive_output, T &warp_aggregate)
```

```
template<typename ScanOp>
```

```
__device__ inline void InclusiveScan(T input, T &inclusive_output, ScanOp scan_op)
```

```
template<typename ScanOp>
```

```
__device__ inline void InclusiveScan(T input, T &inclusive_output, ScanOp scan_op, T
&warp_aggregate)
```

```
template<typename ScanOp>
```

```
__device__ inline void ExclusiveScan(T input, T &exclusive_output, ScanOp scan_op)
```

```
template<typename ScanOp>
```

```
__device__ inline void ExclusiveScan(T input, T &exclusive_output, T initial_value, ScanOp scan_op)
```

```
template<typename ScanOp>
__device__ inline void ExclusiveScan(T input, T &exclusive_output, ScanOp scan_op, T
&warp_aggregate)
```

```
template<typename ScanOp>
__device__ inline void ExclusiveScan(T input, T &exclusive_output, T initial_value, ScanOp scan_op, T
&warp_aggregate)
```

```
template<typename ScanOp>
__device__ inline void Scan(T input, T &inclusive_output, T &exclusive_output, ScanOp scan_op)
```

```
template<typename ScanOp>
__device__ inline void Scan(T input, T &inclusive_output, T &exclusive_output, T initial_value, ScanOp
scan_op)
```

```
__device__ inline T Broadcast(T input, unsigned int src_lane)
```

2.1.3 Enums

Enum BlockHistogramAlgorithm

- Defined in file `hipcub_backend_rocprim_block_block_histogram.hpp`

Enum Documentation

enum **BlockHistogramAlgorithm**

Values:

enumerator **BLOCK_HISTO_ATOMIC**

enumerator **BLOCK_HISTO_SORT**

Enum BlockReduceAlgorithm

- Defined in file `hipcub_backend_rocprim_block_block_reduce.hpp`

Enum Documentation

enum **BlockReduceAlgorithm**

Values:

enumerator **BLOCK_REDUCE_RAKING_COMMUTATIVE_ONLY**

enumerator **BLOCK_REDUCE_RAKING**

enumerator **BLOCK_REDUCE_WARP_REDUCTIONS**

Enum CacheLoadModifier

- Defined in file_hipcub_backend_rocprim_thread_thread_load.hpp

Enum Documentation

enum **CacheLoadModifier**

Values:

enumerator **LOAD_DEFAULT**

Default (no modifier)

enumerator **LOAD_CA**

Cache at all levels.

enumerator **LOAD_CG**

Cache at global level.

enumerator **LOAD_CS**

Cache streaming (likely to be accessed once)

enumerator **LOAD_CV**

Cache as volatile (including cached system lines)

enumerator **LOAD_LDG**

Cache as texture.

enumerator **LOAD_VOLATILE**

Volatile (any memory space)

Enum CacheStoreModifier

- Defined in file_hipcub_backend_rocprim_thread_thread_store.hpp

Enum Documentation

enum **CacheStoreModifier**

Values:

enumerator **STORE_DEFAULT**

Default (no modifier)

enumerator **STORE_WB**

Cache write-back all coherent levels.

enumerator **STORE_CG**

Cache at global level.

enumerator **STORE_CS**
Cache streaming (likely to be accessed once)

enumerator **STORE_WT**
Cache write-through (to system memory)

enumerator **STORE_VOLATILE**
Volatile shared (any memory space)

Enum BlockLoadAlgorithm

- Defined in file `hipcub_backend_rocprim_block_block_load.hpp`

Enum Documentation

enum `hipcub::BlockLoadAlgorithm`

Values:

enumerator **BLOCK_LOAD_DIRECT**

enumerator **BLOCK_LOAD_VECTORIZE**

enumerator **BLOCK_LOAD_TRANSPOSE**

enumerator **BLOCK_LOAD_WARP_TRANSPOSE**

enumerator **BLOCK_LOAD_WARP_TRANSPOSE_TIMESLICED**

Enum BlockScanAlgorithm

- Defined in file `hipcub_backend_rocprim_block_block_scan.hpp`

Enum Documentation

enum `hipcub::BlockScanAlgorithm`

Values:

enumerator **BLOCK_SCAN_RAKING**

enumerator **BLOCK_SCAN_RAKING_MEMOIZE**

enumerator **BLOCK_SCAN_WARP_SCANS**

Enum BlockStoreAlgorithm

- Defined in file_hipcub_backend_rocprim_block_block_store.hpp

Enum Documentation

enum hipcub::BlockStoreAlgorithm

Values:

enumerator BLOCK_STORE_DIRECT

enumerator BLOCK_STORE_VECTORIZE

enumerator BLOCK_STORE_TRANSPOSE

enumerator BLOCK_STORE_WARP_TRANSPOSE

enumerator BLOCK_STORE_WARP_TRANSPOSE_TIMESLICED

Enum GridMappingStrategy

- Defined in file_hipcub_backend_rocprim_grid_grid_mapping.hpp

Enum Documentation

enum hipcub::GridMappingStrategy

cub::GridMappingStrategy enumerates alternative strategies for mapping constant-sized tiles of device-wide data onto a grid of CUDA thread blocks.

Values:

enumerator GRID_MAPPING_RAKE

An a “raking” access pattern in which each thread block is assigned a consecutive sequence of input tiles.

Overview The input is evenly partitioned into p segments, where p is constant and corresponds loosely to the number of thread blocks that may actively reside on the target device. Each segment is comprised of consecutive tiles, where a tile is a small, constant-sized unit of input to be processed to completion before the thread block terminates or obtains more work. The kernel invokes p thread blocks, each of which iteratively consumes a segment of n/p elements in tile-size increments.

enumerator GRID_MAPPING_STRIP_MINE

An a “strip mining” access pattern in which the input tiles assigned to each thread block are separated by a stride equal to the the extent of the grid.

Overview The input is evenly partitioned into p sets, where p is constant and corresponds loosely to the number of thread blocks that may actively reside on the target device. Each set is comprised of data tiles separated by stride `tiles`, where a tile is a small, constant-sized unit of input to be processed to completion before the thread block terminates or obtains more work. The kernel invokes p thread blocks, each of which iteratively consumes a segment of n/p elements in tile-size increments.

enumerator **GRID_MAPPING_DYNAMIC**

A dynamic “queue-based” strategy for assigning input tiles to thread blocks.

Overview The input is treated as a queue to be dynamically consumed by a grid of thread blocks. Work is atomically dequeued in tiles, where a tile is a unit of input to be processed to completion before the thread block terminates or obtains more work. The grid size p is constant, loosely corresponding to the number of thread blocks that may actively reside on the target device.

2.1.4 Functions

Template Function `AsmThreadLoad`

- Defined in file `hipcub_backend_rocprim_thread_thread_load.hpp`

Function Documentation

```
template<CacheLoadModifier MODIFIER = LOAD_DEFAULT, typename T>  
__device__ __forceinline__ T AsmThreadLoad(void *ptr)
```

Template Function `AsmThreadStore`

- Defined in file `hipcub_backend_rocprim_thread_thread_store.hpp`

Function Documentation

```
template<CacheStoreModifier MODIFIER = STORE_DEFAULT, typename T>  
__device__ __forceinline__ void AsmThreadStore(void *ptr, T val)
```

Function detail::`to_BlockHistogramAlgorithm_enum`

- Defined in file `hipcub_backend_rocprim_block_block_histogram.hpp`

Function Documentation

```
inline constexpr std::underlying_type<::rocprim::block_histogram_algorithm>::type detail::to_BlockHistogramAlgorithm_enum
```


Function detail::to_BlockReduceAlgorithm_enum

- Defined in file_hipcub_backend_rocprim_block_block_reduce.hpp

Function Documentation

```
inline constexpr std::underlying_type<::rocprim::block_reduce_algorithm>::type detail::to_BlockReduceAlgorithm_enum(::rocprim::block_reduce_algorithm v)
```

Function hipcub::BAR

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline void hipcub::BAR(int count)
```

Template Function hipcub::BFE

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
template<typename UnsignedBits>
__device__ inline unsigned int hipcub::BFE(UnsignedBits source, unsigned int bit_start, unsigned int num_bits)
```

Function hipcub::BFI

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline void hipcub::BFI(unsigned int &ret, unsigned int x, unsigned int y, unsigned int bit_start, unsigned int num_bits)
```

Function `hipcub::CTA_SYNC`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

```
__device__ inline void hipcub::CTA_SYNC()
```

Function `hipcub::Debug`

- Defined in `file_hipcub_config.hpp`

Function Documentation

```
inline hipError_t hipcub::Debug(hipError_t error, const char *filename, int line)  
hipCUB error reporting macro (prints error messages to stderr)
```

Template Function `hipcub::detail::convert_result_type`

- Defined in `file_hipcub_backend_rocprim_thread_thread_operators.hpp`

Function Documentation

```
template<class InputIteratorT, class OutputIteratorT, class BinaryFunction>  
inline convert_result_type_wrapper<InputIteratorT, OutputIteratorT, BinaryFunction> hipcub::detail::convert_result_type(  
(
```

Template Function `hipcub::detail::get_lowest_value`

- Defined in `file_hipcub_backend_rocprim_device_device_reduce.hpp`

Function Documentation

```
template<class T>  
inline T hipcub::detail::get_lowest_value()
```

Function hipcub::detail::get_lowest_value< __half >

- Defined in file_hipcub_backend_rocprim_device_device_reduce.hpp

Function Documentation

```
template<>
inline __half hipcub::detail::get_lowest_value<__half>()
```

Template Function hipcub::detail::get_max_value

- Defined in file_hipcub_backend_rocprim_device_device_reduce.hpp

Function Documentation

```
template<class T>
inline T hipcub::detail::get_max_value()
```

Function hipcub::detail::get_max_value< __half >

- Defined in file_hipcub_backend_rocprim_device_device_reduce.hpp

Function Documentation

```
template<>
inline __half hipcub::detail::get_max_value<__half>()
```

Function hipcub::detail::to_BlockLoadAlgorithm_enum

- Defined in file_hipcub_backend_rocprim_block_block_load.hpp

Function Documentation

```
inline constexpr std::underlying_type<::rocprim::block_load_method>::type hipcub::detail::to_BlockLoadAlgorithm_enum(
v)
```

Function `hipcub::detail::to_BlockScanAlgorithm_enum`

- Defined in file `hipcub_backend_rocprim_block_block_scan.hpp`

Function Documentation

```
inline constexpr std::underlying_type<::rocprim::block_scan_algorithm>::type hipcub::detail::to_BlockScanAlgorithm_enum(
```

Function `hipcub::detail::to_BlockStoreAlgorithm_enum`

- Defined in file `hipcub_backend_rocprim_block_block_store.hpp`

Function Documentation

```
inline constexpr std::underlying_type<::rocprim::block_store_method>::type hipcub::detail::to_BlockStoreAlgorithm_enum(
```

Template Function `hipcub::detail::to_double_buffer`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Function Documentation

```
template<typename T>  
inline ::rocprim::double_buffer<T> hipcub::detail::to_double_buffer(DoubleBuffer<T> &source)
```

Template Function `hipcub::detail::unsigned_bit_extract`

- Defined in file `hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

```
template<typename UnsignedBits>  
__device__ inline auto hipcub::detail::unsigned_bit_extract(UnsignedBits source, unsigned int bit_start,  
unsigned int num_bits) -> typename  
std::enable_if<sizeof(UnsignedBits) == 8,  
unsigned int>::type
```

Template Function `hipcub::detail::update_double_buffer`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Function Documentation

```
template<typename T>
inline void hipcub::detail::update_double_buffer(DoubleBuffer<T> &target, ::rocprim::double_buffer<T>
&source)
```

Template Function `hipcub::DivideAndRoundUp`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Function Documentation

```
template<typename NumeratorT, typename DenominatorT>
__host__ __device__ __forceinline__ constexpr NumeratorT hipcub::DivideAndRoundUp(NumeratorT n,
DenominatorT d)
```

Function `hipcub::IADD3`

- Defined in file `hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

```
__device__ inline unsigned int hipcub::IADD3(unsigned int x, unsigned int y, unsigned int z)
```

Function `hipcub::internal::ThreadScanExclusive`

- Defined in file `hipcub_backend_rocprim_thread_thread_scan.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “`hipcub::internal::ThreadScanExclusive`” with arguments () in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int LENGTH, typename T, typename ScanOp> __device__ __forceinline__ T
↳ ThreadScanExclusive(T inclusive, T exclusive, T *input, T *output, ScanOp scan_op,
↳ Int2Type<LENGTH>)
```

Function `hipcub::internal::ThreadScanInclusive`

- Defined in file `hipcub_backend_rocprim_thread_thread_scan.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::internal::ThreadScanInclusive” with arguments () in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int LENGTH, typename T, typename ScanOp> __device__ __forceinline__ T  
  ↳ ThreadScanInclusive(T inclusive, T *input, T *output, ScanOp scan_op, Int2Type  
  ↳ <LENGTH>)
```

Function `hipcub::LaneId`

- Defined in file `hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__` inline unsigned int `hipcub::LaneId()`

Function `hipcub::LaneMaskGe`

- Defined in file `hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__` inline uint64_t `hipcub::LaneMaskGe()`

Function `hipcub::LaneMaskGt`

- Defined in file `hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__` inline uint64_t `hipcub::LaneMaskGt()`

Function hipcub::LaneMaskLe

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline uint64_t hipcub::LaneMaskLe()
```

Function hipcub::LaneMaskLt

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline uint64_t hipcub::LaneMaskLt()
```

Template Function hipcub::LoadDirectBlocked(int, InputIteratorT, T(&))

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectBlocked” with arguments (int, InputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectBlocked(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Template Function hipcub::LoadDirectBlocked(int, InputIteratorT, T(&), int)

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectBlocked” with arguments (int, InputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void_
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void_
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename_
↳InputIteratorT> __device__ void LoadDirectBlocked(int linear_id, InputIteratorT_
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Template Function hipcub::LoadDirectBlocked(int, InputIteratorT, T(&), int, Default)

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectBlocked” with arguments (int, InputIteratorT, T (&), int, Default) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void_
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void_
↳LoadDirectBlocked(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_PER_
↳THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename_
↳InputIteratorT> __device__ void LoadDirectBlocked(int linear_id, InputIteratorT_
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Template Function hipcub::LoadDirectBlockedVectorized

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectBlockedVectorized” with arguments (int, T*, T (&)) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD> __device__ void
↳LoadDirectBlockedVectorized(int linear_id, T *block_iter, T (&items)[ITEMS_PER_
↳THREAD])
```

Template Function hipcub::LoadDirectStriped(int, InputIteratorT, T(&))

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectStriped” with arguments (int, InputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD])
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
- template<int BLOCK_THREADS, typename T, typename Default, int ITEMS_PER_THREAD,
↳typename InputIteratorT> __device__ void LoadDirectStriped(int linear_id,
↳InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default
↳oob_default)
```

Template Function hipcub::LoadDirectStriped(int, InputIteratorT, T(&), int)

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectStriped” with arguments (int, InputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD])
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
- template<int BLOCK_THREADS, typename T, typename Default, int ITEMS_PER_THREAD,
↳typename InputIteratorT> __device__ void LoadDirectStriped(int linear_id,
↳InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default
↳oob_default)
```

Template Function `hipcub::LoadDirectStriped(int, InputIteratorT, T(&), int, Default)`

- Defined in file `hipcub_backend_rocprim_block_block_load_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectStriped” with arguments (int, InputIteratorT, T (&), int, Default) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD])
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
- template<int BLOCK_THREADS, typename T, typename Default, int ITEMS_PER_THREAD,
↳typename InputIteratorT> __device__ void LoadDirectStriped(int linear_id,
↳InputIteratorT block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default
↳oob_default)
```

Template Function `hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&))`

- Defined in file `hipcub_backend_rocprim_block_block_load_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectWarpStriped” with arguments (int, InputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectWarpStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Template Function hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&), int)

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectWarpStriped” with arguments (int, InputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectWarpStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Template Function hipcub::LoadDirectWarpStriped(int, InputIteratorT, T(&), int, Default)

- Defined in file_hipcub_backend_rocprim_block_block_load_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::LoadDirectWarpStriped” with arguments (int, InputIteratorT, T (&), int, Default) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename InputIteratorT> __device__ void
↳LoadDirectWarpStriped(int linear_id, InputIteratorT block_iter, T (&items)[ITEMS_
↳PER_THREAD], int valid_items)
- template<typename T, typename Default, int ITEMS_PER_THREAD, typename
↳InputIteratorT> __device__ void LoadDirectWarpStriped(int linear_id, InputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items, Default oob_default)
```

Function `hipcub::PRMT`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__ inline int hipcub::PRMT(unsigned int a, unsigned int b, unsigned int index)`

Function `hipcub::RowMajorTid`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__ inline int hipcub::RowMajorTid(int block_dim_x, int block_dim_y, int block_dim_z)`

Function `hipcub::SHL_ADD`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__ inline unsigned int hipcub::SHL_ADD(unsigned int x, unsigned int shift, unsigned int addend)`

Function `hipcub::SHR_ADD`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

`__device__ inline unsigned int hipcub::SHR_ADD(unsigned int x, unsigned int shift, unsigned int addend)`

Template Function `hipcub::ShuffleDown`

- Defined in `file_hipcub_backend_rocprim_util_ptx.hpp`

Function Documentation

```
template<int LOGICAL_WARP_THREADS, typename T>
__device__ inline T hipcub::ShuffleDown(T input, int src_offset, int last_thread, unsigned int member_mask)
```

Template Function hipcub::ShuffleIndex

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
template<int LOGICAL_WARP_THREADS, typename T>
__device__ inline T hipcub::ShuffleIndex(T input, int src_lane, unsigned int member_mask)
```

Template Function hipcub::ShuffleUp

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
template<int LOGICAL_WARP_THREADS, typename T>
__device__ inline T hipcub::ShuffleUp(T input, int src_offset, int first_thread, unsigned int member_mask)
```

Template Function hipcub::StoreDirectBlocked(int, OutputIteratorT, T(&))

- Defined in file_hipcub_backend_rocprim_block_block_store_func.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectBlocked” with arguments (int, OutputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
  ↪void StoreDirectBlocked(int linear_id, OutputIteratorT block_iter, T (&items)[ITEMS_
  ↪PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
  ↪void StoreDirectBlocked(int linear_id, OutputIteratorT block_iter, T (&items)[ITEMS_
  ↪PER_THREAD], int valid_items)
```

Template Function `hipcub::StoreDirectBlocked(int, OutputIteratorT, T(&), int)`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectBlocked” with arguments (int, OutputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
  ↪void StoreDirectBlocked(int linear_id, OutputIteratorT block_iter, T (&items)[ITEMS_
  ↪PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
  ↪void StoreDirectBlocked(int linear_id, OutputIteratorT block_iter, T (&items)[ITEMS_
  ↪PER_THREAD], int valid_items)
```

Template Function `hipcub::StoreDirectBlockedVectorized`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectBlockedVectorized” with arguments (int, T*, T (&)) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD> __device__ void
  ↪StoreDirectBlockedVectorized(int linear_id, T *block_iter, T (&items)[ITEMS_PER_
  ↪THREAD])
```

Template Function `hipcub::StoreDirectStriped(int, OutputIteratorT, T(&))`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectStriped” with arguments (int, OutputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
  ↪OutputIteratorT> __device__ void StoreDirectStriped(int linear_id, OutputIteratorT
  ↪block_iter, T (&items)[ITEMS_PER_THREAD])
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
  ↪OutputIteratorT> __device__ void StoreDirectStriped(int linear_id, OutputIteratorT
  ↪block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
```

Template Function `hipcub::StoreDirectStriped(int, OutputIteratorT, T(&), int)`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectStriped” with arguments (int, OutputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳OutputIteratorT> __device__ void StoreDirectStriped(int linear_id, OutputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD])
- template<int BLOCK_THREADS, typename T, int ITEMS_PER_THREAD, typename
↳OutputIteratorT> __device__ void StoreDirectStriped(int linear_id, OutputIteratorT
↳block_iter, T (&items)[ITEMS_PER_THREAD], int valid_items)
```

Template Function `hipcub::StoreDirectWarpStriped(int, OutputIteratorT, T(&))`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectWarpStriped” with arguments (int, OutputIteratorT, T (&)) in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
↳void StoreDirectWarpStriped(int linear_id, OutputIteratorT block_iter, T (&
↳items)[ITEMS_PER_THREAD])
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__
↳void StoreDirectWarpStriped(int linear_id, OutputIteratorT block_iter, T (&
↳items)[ITEMS_PER_THREAD], int valid_items)
```

Template Function `hipcub::StoreDirectWarpStriped(int, OutputIteratorT, T(&), int)`

- Defined in file `hipcub_backend_rocprim_block_block_store_func.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “hipcub::StoreDirectWarpStriped” with arguments (int, OutputIteratorT, T (&), int) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__  
  ↪void StoreDirectWarpStriped(int linear_id, OutputIteratorT block_iter, T (&  
  ↪items)[ITEMS_PER_THREAD])  
- template<typename T, int ITEMS_PER_THREAD, typename OutputIteratorT> __device__  
  ↪void StoreDirectWarpStriped(int linear_id, OutputIteratorT block_iter, T (&  
  ↪items)[ITEMS_PER_THREAD], int valid_items)
```

Function hipcub::WARP_ALL

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline int hipcub::WARP_ALL(int predicate, uint64_t member_mask)
```

Function hipcub::WARP_ANY

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline int hipcub::WARP_ANY(int predicate, uint64_t member_mask)
```

Function hipcub::WARP_BALLOT

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline int64_t hipcub::WARP_BALLOT(int predicate, uint64_t member_mask)
```


Function hipcub::WARP_SYNC

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline void hipcub::WARP_SYNC(unsigned int member_mask)
```

Function hipcub::WarpId

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Function Documentation

```
__device__ inline unsigned int hipcub::WarpId()
```

Template Function internal::ThreadReduce(T *, ReductionOp, T)

- Defined in file_hipcub_backend_rocprim_thread_thread_reduce.hpp

Function Documentation

```
template<int LENGTH, typename T, typename ReductionOp, bool NoPrefix = false>
__device__ __forceinline__ T internal::ThreadReduce(T *input, ReductionOp reduction_op, T prefix = T(0))
```

Template Function internal::ThreadReduce(T(&), ReductionOp, T)

- Defined in file_hipcub_backend_rocprim_thread_thread_reduce.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve function “internal::ThreadReduce” with arguments (T (&), ReductionOp, T) in doxygen xml output for project “hipCUB” from directory: ./docBin/xml. Potential matches:

```
- template<int LENGTH, typename T, typename ReductionOp, bool NoPrefix = false> __
  ↳device__ __forceinline__ T ThreadReduce(T *input, ReductionOp reduction_op, T
  ↳prefix = T(0))
- template<int LENGTH, typename T, typename ReductionOp> __device__ __forceinline__ T
  ↳ThreadReduce(T (&input)[LENGTH], ReductionOp reduction_op)
- template<int LENGTH, typename T, typename ReductionOp> __device__ __forceinline__ T
  ↳ThreadReduce(T (&input)[LENGTH], ReductionOp reduction_op, T prefix)
```

Template Function `internal::ThreadReduce(T&), ReductionOp`

- Defined in file `hipcub_backend_rocprim_thread_thread_reduce.hpp`

Function Documentation

Warning: doxygenfunction: Unable to resolve function “`internal::ThreadReduce`” with arguments `(T (&), ReductionOp)` in doxygen xml output for project “hipCUB” from directory: `./docBin/xml`. Potential matches:

```
- template<int LENGTH, typename T, typename ReductionOp, bool NoPrefix = false> __
↳device__ __forceinline__ T ThreadReduce(T *input, ReductionOp reduction_op, T
↳prefix = T(0))
- template<int LENGTH, typename T, typename ReductionOp> __device__ __forceinline__ T
↳ThreadReduce(T (&input)[LENGTH], ReductionOp reduction_op)
- template<int LENGTH, typename T, typename ReductionOp> __device__ __forceinline__ T
↳ThreadReduce(T (&input)[LENGTH], ReductionOp reduction_op, T prefix)
```

Template Function `ThreadLoad(InputIteratorT)`

- Defined in file `hipcub_backend_rocprim_thread_thread_load.hpp`

Function Documentation

```
template<CacheLoadModifier MODIFIER = LOAD_DEFAULT, typename InputIteratorT>
__device__ __forceinline__ std::iterator_traits<InputIteratorT>::value_type ThreadLoad(InputIteratorT itr)
```

Template Function `ThreadLoad(T *)`

- Defined in file `hipcub_backend_rocprim_thread_thread_load.hpp`

Function Documentation

```
template<CacheLoadModifier MODIFIER = LOAD_DEFAULT, typename T>
__device__ __forceinline__ T ThreadLoad(T *ptr)
```

Template Function `ThreadStore(OutputIteratorT, T)`

- Defined in file `hipcub_backend_rocprim_thread_thread_store.hpp`

Function Documentation

```
template<CacheStoreModifier MODIFIER = STORE_DEFAULT, typename OutputIteratorT, typename T>
__device__ __forceinline__ void ThreadStore(OutputIteratorT itr, T val)
```

Template Function ThreadStore(T *, T)

- Defined in file_hipcub_backend_rocprim_thread_thread_store.hpp

Function Documentation

```
template<CacheStoreModifier MODIFIER = STORE_DEFAULT, typename T>
__device__ __forceinline__ void ThreadStore(T *ptr, T val)
```

2.1.5 Defines

Define _HipcubLog

- Defined in file_hipcub_backend_rocprim_util_allocator.hpp

Define Documentation

`_HipcubLog`(format, ...)

Define BEGIN_HIPCUB_NAMESPACE

- Defined in file_hipcub_config.hpp

Define Documentation

`BEGIN_HIPCUB_NAMESPACE`

Define END_HIPCUB_NAMESPACE

- Defined in file_hipcub_config.hpp

Define Documentation

END_HIPCUB_NAMESPACE

Define HIPCUB_ARCH

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Define Documentation

HIPCUB_ARCH

Define HIPCUB_DEVICE

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_DEVICE

Define HIPCUB_DEVICE_WARP_THREADS

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Define Documentation

HIPCUB_DEVICE_WARP_THREADS

Define HIPCUB_HOST

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_HOST

Define HIPCUB_HOST_DEVICE

- Defined in file_hipcub_config.hpp

Define Documentation

hipCUB_HOST_DEVICE

Define HIPCUB_HOST_WARP_THREADS

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Define Documentation

hipCUB_HOST_WARP_THREADS

Define HIPCUB_MAIN_HEADER_INCLUDED

- Defined in file_hipcub_hipcub.hpp

Define Documentation

hipCUB_MAIN_HEADER_INCLUDED

Define HIPCUB_MAX_WARP_SIZE

- Defined in file_hipcub_config.hpp

Define Documentation

hipCUB_MAX_WARP_SIZE

Define HIPCUB_NAMESPACE

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_NAMESPACE

Define HIPCUB_ROCPRIM_API

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_ROCPRIM_API

Define HIPCUB_RUNTIME_FUNCTION

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_RUNTIME_FUNCTION

Define HIPCUB_SHARED_MEMORY

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_SHARED_MEMORY

Define HIPCUB_THREAD_LOAD_USE_CACHE_MODIFIERS

- Defined in file_hipcub_thread_thread_load.hpp

Define Documentation

HIPCUB_THREAD_LOAD_USE_CACHE_MODIFIERS

Define HIPCUB_THREAD_STORE_USE_CACHE_MODIFIERS

- Defined in file_hipcub_thread_thread_store.hpp

Define Documentation

HIPCUB_THREAD_STORE_USE_CACHE_MODIFIERS

Define HIPCUB_WARP_SIZE_32

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_WARP_SIZE_32

Supported warp sizes.

Define HIPCUB_WARP_SIZE_64

- Defined in file_hipcub_config.hpp

Define Documentation

HIPCUB_WARP_SIZE_64

Define HIPCUB_WARP_THREADS

- Defined in file_hipcub_backend_rocprim_util_ptx.hpp

Define Documentation

HIPCUB_WARP_THREADS

Define HipcubDebug

- Defined in file_hipcub_config.hpp

Define Documentation

HipcubDebug(e)

2.1.6 Typedefs

Typedef `hipcub::detail::is_integral_or_enum`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Typedef Documentation

```
using hipcub::detail::is_integral_or_enum = std::integral_constant<bool, std::is_integral<T>::value ||
std::is_enum<T>::value>
```

Typedef `hipcub::KeyValuePair`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Typedef Documentation

```
using hipcub::KeyValuePair = ::rocprim::key_value_pair<Key, Value>
```

Typedef `hipcub::NullType`

- Defined in file `hipcub_backend_rocprim_util_type.hpp`

Typedef Documentation

```
using hipcub::NullType = ::rocprim::empty_type
```


INDICES AND TABLES

- genindex
- search

Symbols

`_HipcubLog` (*C macro*), 111

A

`AsmThreadLoad` (*C++ function*), 92
`AsmThreadStore` (*C++ function*), 92

B

`BEGIN_HIPCUB_NAMESPACE` (*C macro*), 111
`BlockHistogram` (*C++ class*), 45
`BlockHistogram::BlockHistogram` (*C++ function*), 46
`BlockHistogram::Composite` (*C++ function*), 46
`BlockHistogram::Histogram` (*C++ function*), 46
`BlockHistogram::InitHistogram` (*C++ function*), 46
`BlockHistogram::TempStorage` (*C++ type*), 46
`BlockHistogramAlgorithm` (*C++ enum*), 88
`BlockHistogramAlgorithm::BLOCK_HISTO_ATOMIC` (*C++ enumerator*), 88
`BlockHistogramAlgorithm::BLOCK_HISTO_SORT` (*C++ enumerator*), 88
`BlockReduce` (*C++ class*), 46
`BlockReduce::BlockReduce` (*C++ function*), 47
`BlockReduce::Reduce` (*C++ function*), 47
`BlockReduce::Sum` (*C++ function*), 47
`BlockReduce::TempStorage` (*C++ type*), 47
`BlockReduceAlgorithm` (*C++ enum*), 88
`BlockReduceAlgorithm::BLOCK_REDUCE_RAKING` (*C++ enumerator*), 88
`BlockReduceAlgorithm::BLOCK_REDUCE_RAKING_COMMUNICATIVE_ONLY` (*C++ enumerator*), 88
`BlockReduceAlgorithm::BLOCK_REDUCE_WARP_REDUCTIONS` (*C++ enumerator*), 88

C

`CacheLoadModifier` (*C++ enum*), 89
`CacheLoadModifier::LOAD_CA` (*C++ enumerator*), 89
`CacheLoadModifier::LOAD_CG` (*C++ enumerator*), 89
`CacheLoadModifier::LOAD_CS` (*C++ enumerator*), 89
`CacheLoadModifier::LOAD_CV` (*C++ enumerator*), 89
`CacheLoadModifier::LOAD_DEFAULT` (*C++ enumerator*), 89

`CacheLoadModifier::LOAD_LDG` (*C++ enumerator*), 89
`CacheLoadModifier::LOAD_VOLATILE` (*C++ enumerator*), 89
`CacheStoreModifier` (*C++ enum*), 89
`CacheStoreModifier::STORE_CG` (*C++ enumerator*), 89
`CacheStoreModifier::STORE_CS` (*C++ enumerator*), 89
`CacheStoreModifier::STORE_DEFAULT` (*C++ enumerator*), 89
`CacheStoreModifier::STORE_VOLATILE` (*C++ enumerator*), 90
`CacheStoreModifier::STORE_WB` (*C++ enumerator*), 89
`CacheStoreModifier::STORE_WT` (*C++ enumerator*), 90

D

`detail::to_BlockHistogramAlgorithm_enum` (*C++ function*), 92
`detail::to_BlockReduceAlgorithm_enum` (*C++ function*), 93

E

`END_HIPCUB_NAMESPACE` (*C macro*), 112

H

`hipcub::ArgMax` (*C++ struct*), 9
`hipcub::ArgMax::operator()` (*C++ function*), 9
`hipcub::ArgMin` (*C++ struct*), 9
`hipcub::ArgMin::operator()` (*C++ function*), 9
`hipcub::BAR` (*C++ function*), 93
`hipcub::BFE` (*C++ function*), 93
`hipcub::BFI` (*C++ function*), 93
`hipcub::block_raking_layout` (*C++ struct*), 10
`hipcub::block_raking_layout::PlacementPtr` (*C++ function*), 11
`hipcub::block_raking_layout::RakingPtr` (*C++ function*), 11
`hipcub::block_raking_layout::TempStorage` (*C++ struct*), 11, 12

hipcub::block_raking_layout::TempStorage::Alias (C++ function), 51
 (C++ function), 11, 12 hipcub::BlockExchange::ScatterToStriped
 hipcub::block_raking_layout::TempStorage::DeviceWord (C++ function), 52
 (C++ type), 11, 12 hipcub::BlockExchange::ScatterToStripedFlagged
 hipcub::block_raking_layout::TempStorage::storage (C++ function), 52
 (C++ member), 11, 12 hipcub::BlockExchange::ScatterToStripedGuarded
 hipcub::block_raking_layout::TempStorage::[anonymous] (C++ function), 52
 (C++ enum), 11, 12 hipcub::BlockExchange::StripedToBlocked
 hipcub::block_raking_layout::[anonymous] (C++ function), 51
 (C++ enum), 10 hipcub::BlockExchange::TempStorage (C++ type),
 hipcub::block_raking_layout::[anonymous]::GRID_ELEMENTS
 (C++ enumerator), 10 hipcub::BlockExchange::WarpStripedToBlocked
 hipcub::block_raking_layout::[anonymous]::MAX_RAKING_THREADS (C++ function), 51
 (C++ enumerator), 10 hipcub::BlockLoad (C++ class), 52
 hipcub::block_raking_layout::[anonymous]::RAKING_THREADS BlockLoad::BlockLoad (C++ function), 53
 (C++ enumerator), 10 hipcub::BlockLoad::Load (C++ function), 53
 hipcub::block_raking_layout::[anonymous]::SEGMENT_LENGTH BlockLoad::TempStorage (C++ type), 53
 (C++ enumerator), 10 hipcub::BlockLoadAlgorithm (C++ enum), 90
 hipcub::block_raking_layout::[anonymous]::SHARED_ELEMENTS BlockLoadAlgorithm::BLOCK_LOAD_DIRECT
 (C++ enumerator), 10 (C++ enumerator), 90
 hipcub::block_raking_layout::[anonymous]::UNGUARDED BlockLoadAlgorithm::BLOCK_LOAD_TRANSPOSE
 (C++ enumerator), 10 (C++ enumerator), 90
 hipcub::block_raking_layout::[anonymous]::USE_SEGMENT_PADDING BlockLoadAlgorithm::BLOCK_LOAD_VECTORIZE
 (C++ enumerator), 10 (C++ enumerator), 90
 hipcub::BlockAdjacentDifference (C++ class), 48 hipcub::BlockLoadAlgorithm::BLOCK_LOAD_WARP_TRANSPOSE
 hipcub::BlockAdjacentDifference::BlockAdjacentDifference (C++ enumerator), 90
 (C++ function), 48 hipcub::BlockLoadAlgorithm::BLOCK_LOAD_WARP_TRANSPOSE_TIME
 hipcub::BlockAdjacentDifference::FlagHeads (C++ enumerator), 90
 (C++ function), 48 hipcub::BlockRadixRank (C++ class), 53
 hipcub::BlockAdjacentDifference::FlagHeadsAndTails hipcub::BlockRadixRank::BlockRadixRank (C++
 (C++ function), 48, 49 function), 54
 hipcub::BlockAdjacentDifference::FlagTails hipcub::BlockRadixRank::PrefixCallBack (C++
 (C++ function), 48 struct), 13
 hipcub::BlockAdjacentDifference::TempStorage hipcub::BlockRadixRank::PrefixCallBack::operator()
 (C++ type), 48 (C++ function), 13
 hipcub::BlockDiscontinuity (C++ class), 49 hipcub::BlockRadixRank::RankKeys (C++ func-
 hipcub::BlockDiscontinuity::BlockDiscontinuity tion), 55
 (C++ function), 49 hipcub::BlockRadixRank::TempStorage (C++
 hipcub::BlockDiscontinuity::FlagHeads (C++ struct), 13, 55
 function), 49 hipcub::BlockRadixRank::TempStorage::Alias
 hipcub::BlockDiscontinuity::FlagHeadsAndTails (C++ function), 14, 56
 (C++ function), 50 hipcub::BlockRadixRank::TempStorage::DeviceWord
 hipcub::BlockDiscontinuity::FlagTails (C++ (C++ type), 13, 56
 function), 50 hipcub::BlockRadixRank::TempStorage::storage
 hipcub::BlockDiscontinuity::TempStorage (C++ member), 14, 56
 (C++ type), 49 hipcub::BlockRadixRank::TempStorage::[anonymous]
 hipcub::BlockExchange (C++ class), 51 (C++ enum), 13, 56
 hipcub::BlockExchange::BlockedToStriped hipcub::BlockRadixRank::[anonymous] (C++
 (C++ function), 51 enum), 55
 hipcub::BlockExchange::BlockedToWarpStriped hipcub::BlockRadixRank::[anonymous]::BINS_TRACKED_PER_THR
 (C++ function), 51 (C++ enumerator), 55
 hipcub::BlockExchange::BlockExchange (C++ hipcub::BlockRadixRankMatch (C++ class), 56
 function), 51 hipcub::BlockRadixRankMatch::BlockRadixRankMatch
 hipcub::BlockExchange::ScatterToBlocked (C++ function), 56

hipcub::BlockRadixRankMatch::RankKeys (C++ function), 57
 hipcub::BlockRadixRankMatch::TempStorage (C++ struct), 14, 57
 hipcub::BlockRadixRankMatch::TempStorage::Alias (C++ function), 15, 58
 hipcub::BlockRadixRankMatch::TempStorage::DeviceWord (C++ type), 14, 58
 hipcub::BlockRadixRankMatch::TempStorage::storage (C++ member), 15, 58
 hipcub::BlockRadixRankMatch::TempStorage::[anonymous] (C++ enumerator), 91
 hipcub::BlockRadixRankMatch::TempStorage::[anonymous] (C++ enum), 14, 58
 hipcub::BlockRadixRankMatch::[anonymous] (C++ enum), 57
 hipcub::BlockRadixRankMatch::[anonymous]::BINS_TRACKED_PER_THREAD (C++ enumerator), 57
 hipcub::BlockRadixSort (C++ class), 58
 hipcub::BlockRadixSort::BlockRadixSort (C++ function), 59
 hipcub::BlockRadixSort::Sort (C++ function), 59
 hipcub::BlockRadixSort::SortBlockedToStriped (C++ function), 59
 hipcub::BlockRadixSort::SortDescending (C++ function), 59
 hipcub::BlockRadixSort::SortDescendingBlockedToStriped (C++ function), 59
 hipcub::BlockRadixSort::TempStorage (C++ type), 59
 hipcub::BlockScan (C++ class), 60
 hipcub::BlockScan::BlockScan (C++ function), 60
 hipcub::BlockScan::ExclusiveScan (C++ function), 62
 hipcub::BlockScan::ExclusiveSum (C++ function), 61
 hipcub::BlockScan::InclusiveScan (C++ function), 61
 hipcub::BlockScan::InclusiveSum (C++ function), 60
 hipcub::BlockScan::TempStorage (C++ type), 60
 hipcub::BlockScanAlgorithm (C++ enum), 90
 hipcub::BlockScanAlgorithm::BLOCK_SCAN_RAKING (C++ enumerator), 90
 hipcub::BlockScanAlgorithm::BLOCK_SCAN_RAKING_MEMOIZE (C++ function), 67
 hipcub::BlockScanAlgorithm::BLOCK_SCAN_RAKING_MEMOIZE (C++ enumerator), 90
 hipcub::BlockScanAlgorithm::BLOCK_SCAN_WARP_SCANS (C++ function), 67
 hipcub::BlockScanAlgorithm::BLOCK_SCAN_WARP_SCANS (C++ enumerator), 90
 hipcub::BlockShuffle (C++ class), 63
 hipcub::BlockShuffle::BlockShuffle (C++ function), 63
 hipcub::BlockShuffle::Down (C++ function), 64, 65
 hipcub::BlockShuffle::Offset (C++ function), 63
 hipcub::BlockShuffle::Rotate (C++ function), 63
 hipcub::BlockShuffle::TempStorage (C++ type), 63
 hipcub::BlockShuffle::Up (C++ function), 63, 64
 hipcub::BlockStore (C++ class), 65
 hipcub::BlockStore::BlockStore (C++ function), 66
 hipcub::BlockStore::Store (C++ function), 66
 hipcub::BlockStore::TempStorage (C++ type), 66
 hipcub::BlockStoreAlgorithm (C++ enum), 91
 hipcub::BlockStoreAlgorithm::BLOCK_STORE_DIRECT (C++ enumerator), 91
 hipcub::BlockStoreAlgorithm::BLOCK_STORE_TRANSPOSE (C++ enumerator), 91
 hipcub::BlockStoreAlgorithm::BLOCK_STORE_VECTORIZE (C++ enumerator), 91
 hipcub::BlockStoreAlgorithm::BLOCK_STORE_WARP_TRANSPOSE (C++ enumerator), 91
 hipcub::BlockStoreAlgorithm::BLOCK_STORE_WARP_TRANSPOSE_T (C++ enumerator), 91
 hipcub::CacheModifiedInputIterator (C++ class), 66
 hipcub::CacheModifiedInputIterator::CacheModifiedInputIterator (C++ function), 67
 hipcub::CacheModifiedInputIterator::difference_type (C++ type), 66
 hipcub::CacheModifiedInputIterator::iterator_category (C++ type), 66
 hipcub::CacheModifiedInputIterator::operator!= (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator* (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator+ (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator++ (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator+= (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator== (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator- (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator-= (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator-> (C++ function), 67
 hipcub::CacheModifiedInputIterator::operator[] (C++ function), 67
 hipcub::CacheModifiedInputIterator::pointer (C++ type), 66
 hipcub::CacheModifiedInputIterator::ptr (C++ member), 67
 hipcub::CacheModifiedInputIterator::reference (C++ type), 66
 hipcub::CacheModifiedInputIterator::self_type (C++ type), 66
 hipcub::CacheModifiedInputIterator::value_type (C++ type), 66

(C++ type), 66
 hipcub::CacheModifiedOutputIterator (C++ class), 68
 hipcub::CacheModifiedOutputIterator::CacheModifiedOutputIterator (C++ member), 20, 21
 hipcub::CacheModifiedOutputIterator::CacheModifiedOutputIterator (C++ function), 69
 hipcub::CacheModifiedOutputIterator::difference_type (C++ member), 20, 21
 hipcub::CacheModifiedOutputIterator::difference_type (C++ type), 68
 hipcub::CacheModifiedOutputIterator::iterator_category (C++ member), 20, 21
 hipcub::CacheModifiedOutputIterator::iterator_category (C++ type), 68
 hipcub::CacheModifiedOutputIterator::operator!= (C++ function), 20, 21
 hipcub::CacheModifiedOutputIterator::operator!= (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator* (C++ member), 20, 21
 hipcub::CacheModifiedOutputIterator::operator* (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator+ (C++ function), 20, 21
 hipcub::CacheModifiedOutputIterator::operator+ (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator++ (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator++ (C++ type), 16
 hipcub::CacheModifiedOutputIterator::operator+= (C++ member), 19
 hipcub::CacheModifiedOutputIterator::operator+= (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator== (C++ member), 19
 hipcub::CacheModifiedOutputIterator::operator== (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator- (C++ type), 16
 hipcub::CacheModifiedOutputIterator::operator- (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator-- (C++ function), 69
 hipcub::CacheModifiedOutputIterator::operator-- (C++ function), 17
 hipcub::CacheModifiedOutputIterator::operator[] (C++ type), 16
 hipcub::CacheModifiedOutputIterator::operator[] (C++ function), 69
 hipcub::CacheModifiedOutputIterator::pointer (C++ type), 68
 hipcub::CacheModifiedOutputIterator::Reference (C++ function), 16, 17
 hipcub::CacheModifiedOutputIterator::Reference (C++ struct), 15
 hipcub::CacheModifiedOutputIterator::reference (C++ function), 16, 18
 hipcub::CacheModifiedOutputIterator::reference (C++ type), 68
 hipcub::CacheModifiedOutputIterator::Reference::operator!= (C++ function), 16, 18
 hipcub::CacheModifiedOutputIterator::Reference::operator!= (C++ function), 15
 hipcub::CacheModifiedOutputIterator::Reference::ptr (C++ type), 16
 hipcub::CacheModifiedOutputIterator::Reference::ptr (C++ member), 15
 hipcub::CacheModifiedOutputIterator::Reference::Reference (C++ function), 19
 hipcub::CacheModifiedOutputIterator::Reference::Reference (C++ function), 15
 hipcub::CacheModifiedOutputIterator::self_type (C++ member), 19
 hipcub::CacheModifiedOutputIterator::self_type (C++ type), 68
 hipcub::CacheModifiedOutputIterator::value_type (C++ member), 19
 hipcub::CacheModifiedOutputIterator::value_type (C++ type), 68
 hipcub::CachingDeviceAllocator (C++ struct), 16
 hipcub::CachingDeviceAllocator::~CachingDeviceAllocator (C++ function), 18
 hipcub::CachingDeviceAllocator::bin_growth (C++ member), 18
 hipcub::CachingDeviceAllocator::BlockDescriptor (C++ struct), 19, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::BlockDescriptor (C++ function), 19, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::bytes (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::d_ptr (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::device (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::PtrComparator (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::ready_event (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BlockDescriptor::SizeComparator (C++ member), 20, 21
 hipcub::CachingDeviceAllocator::BusyBlocks (C++ type), 16
 hipcub::CachingDeviceAllocator::cached_blocks (C++ member), 19
 hipcub::CachingDeviceAllocator::cached_bytes (C++ member), 19
 hipcub::CachingDeviceAllocator::CachedBlocks (C++ type), 16
 hipcub::CachingDeviceAllocator::CachingDeviceAllocator (C++ function), 17
 hipcub::CachingDeviceAllocator::Compare (C++ type), 16
 hipcub::CachingDeviceAllocator::debug (C++ member), 19
 hipcub::CachingDeviceAllocator::DeviceAllocator (C++ function), 16, 17
 hipcub::CachingDeviceAllocator::DeviceFree (C++ function), 16, 18
 hipcub::CachingDeviceAllocator::FreeAllCached (C++ function), 16, 18
 hipcub::CachingDeviceAllocator::GpuCachedBytes (C++ type), 16
 hipcub::CachingDeviceAllocator::IntPow (C++ function), 19
 hipcub::CachingDeviceAllocator::INVALID_BIN (C++ member), 19
 hipcub::CachingDeviceAllocator::INVALID_DEVICE_ORDINAL (C++ member), 19
 hipcub::CachingDeviceAllocator::INVALID_SIZE (C++ member), 19
 hipcub::CachingDeviceAllocator::live_blocks (C++ member), 19
 hipcub::CachingDeviceAllocator::max_bin (C++ member), 18
 hipcub::CachingDeviceAllocator::max_bin_bytes (C++ member), 18
 hipcub::CachingDeviceAllocator::max_bin_bytes (C++ member), 18
 hipcub::CachingDeviceAllocator::max_cached_bytes (C++ member), 18
 hipcub::CachingDeviceAllocator::min_bin (C++ member), 18
 hipcub::CachingDeviceAllocator::min_bin (C++ member), 18

hipcub::CachingDeviceAllocator::min_bin_bytes (C++ member), 18

hipcub::CachingDeviceAllocator::mutex (C++ member), 18

hipcub::CachingDeviceAllocator::NearestPowerOfTwo (C++ function), 16

hipcub::CachingDeviceAllocator::SetMaxCachedBytes (C++ function), 16, 17

hipcub::CachingDeviceAllocator::skip_cleanup (C++ member), 18

hipcub::CachingDeviceAllocator::TotalBytes (C++ class), 20, 70

hipcub::CachingDeviceAllocator::TotalBytes::free (C++ member), 20, 70

hipcub::CachingDeviceAllocator::TotalBytes::live (C++ member), 20, 70

hipcub::CachingDeviceAllocator::TotalBytes::TotalBytes (C++ function), 20, 70

hipcub::CTA_SYNC (C++ function), 94

hipcub::Debug (C++ function), 94

hipcub::detail::convert_result_type (C++ function), 94

hipcub::detail::get_lowest_value (C++ function), 94

hipcub::detail::get_lowest_value<__half> (C++ function), 95

hipcub::detail::get_max_value (C++ function), 95

hipcub::detail::get_max_value<__half> (C++ function), 95

hipcub::detail::is_integral_or_enum (C++ type), 116

hipcub::detail::to_BlockLoadAlgorithm_enum (C++ function), 95

hipcub::detail::to_BlockScanAlgorithm_enum (C++ function), 96

hipcub::detail::to_BlockStoreAlgorithm_enum (C++ function), 96

hipcub::detail::to_double_buffer (C++ function), 96

hipcub::detail::unsigned_bit_extract (C++ function), 96

hipcub::detail::update_double_buffer (C++ function), 97

hipcub::DeviceHistogram (C++ struct), 21

hipcub::DeviceHistogram::HistogramEven (C++ function), 22, 23

hipcub::DeviceHistogram::HistogramRange (C++ function), 22, 24

hipcub::DeviceHistogram::MultiHistogramEven (C++ function), 22–24

hipcub::DeviceHistogram::MultiHistogramRange (C++ function), 23, 24

hipcub::DevicePartition (C++ struct), 25

hipcub::DevicePartition::Flagged (C++ function), 25, 26

hipcub::DevicePartition::If (C++ function), 25, 26

hipcub::DeviceRadixSort (C++ struct), 27

hipcub::DeviceRadixSort::SortKeys (C++ function), 28, 29

hipcub::DeviceRadixSort::SortKeysDescending (C++ function), 28, 29

hipcub::DeviceRadixSort::SortPairs (C++ function), 27, 28

hipcub::DeviceRadixSort::SortPairsDescending (C++ function), 27–29

hipcub::DeviceReduce (C++ class), 70

hipcub::DeviceReduce::ArgMax (C++ function), 71, 72

hipcub::DeviceReduce::ArgMin (C++ function), 71

hipcub::DeviceReduce::Max (C++ function), 71, 72

hipcub::DeviceReduce::Min (C++ function), 70, 71

hipcub::DeviceReduce::Reduce (C++ function), 70, 71

hipcub::DeviceReduce::ReduceByKey (C++ function), 71, 72

hipcub::DeviceReduce::Sum (C++ function), 70, 71

hipcub::DeviceRunLengthEncode (C++ class), 72

hipcub::DeviceRunLengthEncode::Encode (C++ function), 72, 73

hipcub::DeviceRunLengthEncode::NonTrivialRuns (C++ function), 72, 73

hipcub::DeviceScan (C++ class), 73

hipcub::DeviceScan::ExclusiveScan (C++ function), 74

hipcub::DeviceScan::ExclusiveSum (C++ function), 73, 74

hipcub::DeviceScan::InclusiveScan (C++ function), 73, 74

hipcub::DeviceScan::InclusiveSum (C++ function), 73, 74

hipcub::DeviceSegmentedRadixSort (C++ struct), 30

hipcub::DeviceSegmentedRadixSort::SortKeys (C++ function), 30–32

hipcub::DeviceSegmentedRadixSort::SortKeysDescending (C++ function), 31, 32

hipcub::DeviceSegmentedRadixSort::SortPairs (C++ function), 30, 31

hipcub::DeviceSegmentedRadixSort::SortPairsDescending (C++ function), 30–32

hipcub::DeviceSegmentedReduce (C++ struct), 33

hipcub::DeviceSegmentedReduce::ArgMax (C++ function), 33, 34

hipcub::DeviceSegmentedReduce::ArgMin (C++ function), 33, 34

hipcub::DeviceSegmentedReduce::Max (C++ function), 33, 34

hipcub::DeviceSegmentedReduce::Min (C++ function), 33, 34
 hipcub::DeviceSegmentedReduce::Reduce (C++ function), 33
 hipcub::DeviceSegmentedReduce::Sum (C++ function), 33, 34
 hipcub::DeviceSelect (C++ class), 74
 hipcub::DeviceSelect::Flagged (C++ function), 75
 hipcub::DeviceSelect::If (C++ function), 75
 hipcub::DeviceSelect::Unique (C++ function), 75
 hipcub::DeviceSpmv (C++ class), 76
 hipcub::DeviceSpmv::CsrMV (C++ function), 76, 77
 hipcub::DeviceSpmv::CsrMVKernel (C++ function), 76
 hipcub::DeviceSpmv::CsrMVKernel_MaxThreads (C++ member), 77
 hipcub::DeviceSpmv::SpmvParams (C++ struct), 35, 77
 hipcub::DeviceSpmv::SpmvParams::alpha (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::beta (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::d_column_indices (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::d_row_end_offsets (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::d_values (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::d_vector_x (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::d_vector_y (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::num_cols (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::num_nonzeros (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::num_rows (C++ member), 35, 78
 hipcub::DeviceSpmv::SpmvParams::t_vector_x (C++ member), 35, 78
 hipcub::DigitExtractor (C++ struct), 36
 hipcub::DigitExtractor::current_bit (C++ member), 36
 hipcub::DigitExtractor::DigitExtractor (C++ function), 36
 hipcub::DigitExtractor::mask (C++ member), 36
 hipcub::DiscardOutputIterator (C++ class), 79
 hipcub::DiscardOutputIterator::difference_type (C++ type), 79
 hipcub::DiscardOutputIterator::DiscardOutputIterator (C++ function), 80
 hipcub::DiscardOutputIterator::iterator_category (C++ type), 79
 hipcub::DiscardOutputIterator::operator void* (C++ function), 80
 hipcub::DiscardOutputIterator::operator!= (C++ function), 80
 hipcub::DiscardOutputIterator::operator* (C++ function), 80
 hipcub::DiscardOutputIterator::operator+ (C++ function), 80
 hipcub::DiscardOutputIterator::operator++ (C++ function), 80
 hipcub::DiscardOutputIterator::operator+= (C++ function), 80
 hipcub::DiscardOutputIterator::operator= (C++ function), 80
 hipcub::DiscardOutputIterator::operator== (C++ function), 80
 hipcub::DiscardOutputIterator::operator- (C++ function), 80
 hipcub::DiscardOutputIterator::operator-= (C++ function), 80
 hipcub::DiscardOutputIterator::operator-> (C++ function), 80
 hipcub::DiscardOutputIterator::operator[] (C++ function), 80
 hipcub::DiscardOutputIterator::pointer (C++ type), 79
 hipcub::DiscardOutputIterator::reference (C++ type), 79
 hipcub::DiscardOutputIterator::self_type (C++ type), 79
 hipcub::DiscardOutputIterator::value_type (C++ type), 79
 hipcub::DivideAndRoundUp (C++ function), 97
 hipcub::DoubleBuffer (C++ struct), 36
 hipcub::DoubleBuffer::Alternate (C++ function), 37
 hipcub::DoubleBuffer::Current (C++ function), 37
 hipcub::DoubleBuffer::d_buffers (C++ member), 37
 hipcub::DoubleBuffer::DoubleBuffer (C++ function), 37
 hipcub::DoubleBuffer::selector (C++ member), 37
 hipcub::Equality (C++ struct), 37
 hipcub::Equality::operator() (C++ function), 37
 hipcub::GridBarrier (C++ class), 81
 hipcub::GridBarrier::d_sync (C++ member), 81
 hipcub::GridBarrier::GridBarrier (C++ function), 81
 hipcub::GridBarrier::Sync (C++ function), 81
 hipcub::GridBarrier::SyncFlag (C++ type), 81
 hipcub::GridBarrierLifetime (C++ class), 82
 hipcub::GridBarrierLifetime::~~GridBarrierLifetime (C++ function), 82
 hipcub::GridBarrierLifetime::d_sync (C++

member), 82
hipcub::GridBarrierLifetime::GridBarrierLifetime (C++ function), 82
hipcub::GridBarrierLifetime::HostReset (C++ function), 82
hipcub::GridBarrierLifetime::Setup (C++ function), 82
hipcub::GridBarrierLifetime::Sync (C++ function), 82
hipcub::GridBarrierLifetime::sync_bytes (C++ member), 82
hipcub::GridBarrierLifetime::SyncFlag (C++ type), 82
hipcub::GridEvenShare (C++ struct), 37
hipcub::GridEvenShare::block_end (C++ member), 39
hipcub::GridEvenShare::block_offset (C++ member), 39
hipcub::GridEvenShare::block_stride (C++ member), 39
hipcub::GridEvenShare::BlockInit (C++ function), 38
hipcub::GridEvenShare::DispatchInit (C++ function), 38
hipcub::GridEvenShare::grid_size (C++ member), 39
hipcub::GridEvenShare::GridEvenShare (C++ function), 38
hipcub::GridEvenShare::num_items (C++ member), 39
hipcub::GridMappingStrategy (C++ enum), 91
hipcub::GridMappingStrategy::GRID_MAPPING_DYNAMIC (C++ enumerator), 92
hipcub::GridMappingStrategy::GRID_MAPPING_RAKE (C++ enumerator), 91
hipcub::GridMappingStrategy::GRID_MAPPING_STRIP_MINE (C++ enumerator), 91
hipcub::GridQueue (C++ class), 83
hipcub::GridQueue::AllocationSize (C++ function), 84
hipcub::GridQueue::Drain (C++ function), 84
hipcub::GridQueue::Fill (C++ function), 84
hipcub::GridQueue::FillAndResetDrain (C++ function), 83
hipcub::GridQueue::FillSize (C++ function), 84
hipcub::GridQueue::GridQueue (C++ function), 83
hipcub::GridQueue::ResetDrain (C++ function), 83
hipcub::GridQueue::ResetFill (C++ function), 83, 84
hipcub::IADD3 (C++ function), 97
hipcub::If (C++ struct), 39
hipcub::If::Type (C++ type), 39
hipcub::Inequality (C++ struct), 39
hipcub::Inequality::operator() (C++ function), 40
hipcub::InequalityWrapper (C++ struct), 40
hipcub::InequalityWrapper::InequalityWrapper (C++ function), 40
hipcub::InequalityWrapper::op (C++ member), 40
hipcub::InequalityWrapper::operator() (C++ function), 40
hipcub::Int2Type (C++ struct), 40
hipcub::Int2Type::[anonymous] (C++ enum), 41
hipcub::Int2Type::[anonymous]::VALUE (C++ enumerator), 41
hipcub::IsPointer (C++ struct), 41
hipcub::IsPointer::VALUE (C++ member), 41
hipcub::IsVolatile (C++ struct), 41
hipcub::IsVolatile::VALUE (C++ member), 41
hipcub::KeyValuePair (C++ type), 116
hipcub::LaneId (C++ function), 98
hipcub::LaneMaskGe (C++ function), 98
hipcub::LaneMaskGt (C++ function), 98
hipcub::LaneMaskLe (C++ function), 99
hipcub::LaneMaskLt (C++ function), 99
hipcub::Log2 (C++ struct), 42
hipcub::Log2::VALUE (C++ member), 42
hipcub::Max (C++ struct), 42
hipcub::Max::operator() (C++ function), 42
hipcub::Min (C++ struct), 42
hipcub::Min::operator() (C++ function), 42
hipcub::NullType (C++ type), 116
hipcub::PowerOfTwo (C++ struct), 43
hipcub::PowerOfTwo::VALUE (C++ member), 43
hipcub::PRMT (C++ function), 104
hipcub::RadixSortTwiddle (C++ struct), 43
hipcub::RadixSortTwiddle::DefaultKey (C++ function), 43
hipcub::RadixSortTwiddle::In (C++ function), 43
hipcub::RadixSortTwiddle::Out (C++ function), 43
hipcub::RadixSortTwiddle::TraitsT (C++ type), 43
hipcub::RadixSortTwiddle::UnsignedBits (C++ type), 43
hipcub::RemoveQualifiers (C++ struct), 44
hipcub::RemoveQualifiers::Type (C++ type), 44
hipcub::RowMajorTid (C++ function), 104
hipcub::SHL_ADD (C++ function), 104
hipcub::SHR_ADD (C++ function), 104
hipcub::ShuffleDown (C++ function), 105
hipcub::ShuffleIndex (C++ function), 105
hipcub::ShuffleUp (C++ function), 105
hipcub::Sum (C++ struct), 44
hipcub::Sum::operator() (C++ function), 44
hipcub::TexObjInputIterator (C++ class), 84
hipcub::TexObjInputIterator::~TexObjInputIterator (C++ function), 84

hipcub::TexObjInputIterator::BindTexture (C++ function), 84
 hipcub::TexObjInputIterator::TexObjInputIterator (C++ function), 84
 hipcub::TexObjInputIterator::UnbindTexture (C++ function), 84
 hipcub::TexRefInputIterator (C++ class), 85
 hipcub::TexRefInputIterator::~TexRefInputIterator (C++ function), 85
 hipcub::TexRefInputIterator::BindTexture (C++ function), 85
 hipcub::TexRefInputIterator::TexRefInputIterator (C++ function), 85
 hipcub::TexRefInputIterator::UnbindTexture (C++ function), 85
 hipcub::Uninitialized (C++ struct), 44
 hipcub::Uninitialized::Alias (C++ function), 45
 hipcub::Uninitialized::DeviceWord (C++ type), 45
 hipcub::Uninitialized::storage (C++ member), 45
 hipcub::Uninitialized::[anonymous] (C++ enum), 45
 hipcub::Uninitialized::[anonymous]::WORDS (C++ enumerator), 45
 hipcub::WARP_ALL (C++ function), 108
 hipcub::WARP_ANY (C++ function), 108
 hipcub::WARP_BALLOT (C++ function), 108
 hipcub::WARP_SYNC (C++ function), 109
 hipcub::WarpId (C++ function), 109
 hipcub::WarpReduce (C++ class), 86
 hipcub::WarpReduce::HeadSegmentedReduce (C++ function), 86
 hipcub::WarpReduce::HeadSegmentedSum (C++ function), 86
 hipcub::WarpReduce::Reduce (C++ function), 86
 hipcub::WarpReduce::Sum (C++ function), 86
 hipcub::WarpReduce::TailSegmentedReduce (C++ function), 86
 hipcub::WarpReduce::TailSegmentedSum (C++ function), 86
 hipcub::WarpReduce::TempStorage (C++ type), 86
 hipcub::WarpReduce::WarpReduce (C++ function), 86
 hipcub::WarpScan (C++ class), 87
 hipcub::WarpScan::Broadcast (C++ function), 88
 hipcub::WarpScan::ExclusiveScan (C++ function), 87, 88
 hipcub::WarpScan::ExclusiveSum (C++ function), 87
 hipcub::WarpScan::InclusiveScan (C++ function), 87
 hipcub::WarpScan::InclusiveSum (C++ function), 87
 hipcub::WarpScan::Scan (C++ function), 88
 hipcub::WarpScan::TempStorage (C++ type), 87
 hipcub::WarpScan::WarpScan (C++ function), 87
 HIPCUB_ARCH (C macro), 112
 HIPCUB_DEVICE (C macro), 112
 HIPCUB_DEVICE_WARP_THREADS (C macro), 112
 HIPCUB_HOST (C macro), 112
 HIPCUB_HOST_DEVICE (C macro), 113
 HIPCUB_HOST_WARP_THREADS (C macro), 113
 HIPCUB_MAIN_HEADER_INCLUDED (C macro), 113
 HIPCUB_MAX_WARP_SIZE (C macro), 113
 HIPCUB_NAMESPACE (C macro), 114
 HIPCUB_ROCPRIM_API (C macro), 114
 HIPCUB_RUNTIME_FUNCTION (C macro), 114
 HIPCUB_SHARED_MEMORY (C macro), 114
 HIPCUB_THREAD_LOAD_USE_CACHE_MODIFIERS (C macro), 114
 HIPCUB_THREAD_STORE_USE_CACHE_MODIFIERS (C macro), 115
 HIPCUB_WARP_SIZE_32 (C macro), 115
 HIPCUB_WARP_SIZE_64 (C macro), 115
 HIPCUB_WARP_THREADS (C macro), 115
 HipcubDebug (C macro), 116

I

internal::ThreadReduce (C++ function), 109

T

ThreadLoad (C++ function), 110
 ThreadStore (C++ function), 111