



Denglin HammingTM V2

dICU Extended API

DL-DG/SW-032C-06

2025-01-03

Copyright©苏州登临科技有限公司，2019 - 2025，版权所有。

未经苏州登临科技有限公司事先书面同意，不得以任何形式或方式复制或传播本文件的任何部分。

商标和许可



和其它苏州登临科技有限公司的其它登临科技的图标为苏州登临科技有限公司的商标。本手册中提及的所有其他商标均为其各自所有者的财产。

通知

所购买的产品、服务和特性由苏州登临科技有限公司与客户签订的合同规定。本文件中描述的所有或部分产品、服务和特性可能不在采购范围或使用范围内。除非合同中另有规定，本文件中的所有声明、信息和建议均按“原样”提供，无任何明示或暗示的保证或陈述。本手册中的信息如有更改，恕不另行通知。本文件在编制过程中已尽一切努力确保内容的准确性，本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼栋5号1101室，江苏，中国

<http://www.denglin.ai>

email : support@denglin.ai

Change History

| Version | Change Description |
|---------|--|
| 06 | Delete redundant content. |
| 05 | Added notes "Internal usage only" to <code>cudaSetClusterMask</code> 、 <code>cudaGetClusterMask</code> 、 <code>cudaEnterSpmSection</code> and <code>cudaLeavespmSection</code> . Removed <code>cudaMallocCluster</code> . Renamed <code>cudaGetDeviceProperties</code> to <code>cudaGetDeviceProperties_ext</code> . |
| 04 | Changed descriptions of <code>cudaSetClusterMask</code> . |
| 03 | Changed descriptions of <code>cudaEnterSpmSection</code> and <code>cudaLeavespmSection</code> . |
| 02 | Added <code>cuModuleGetFunctionAsync</code> 、 <code>cudaMallocCluster</code> 、 <code>cudaGetClusterMask</code> and <code>cudaGetDeviceProperties</code> . |
| 01 | Initial version. |

Table of Contents

Table of Contents

- 1 `cuMemAllocChannel`
- 2 `cudaSetClusterMask`
- 3 `cudaGetClusterMask`
- 4 `cudaGetSpm`
- 5 `cudaEnterSpmSection`
- 6 `cudaLeaveSpmSection`
- 7 `cuModuleGetFunctionAsync`
- 8 `cudaGetDeviceProperties_ext`

1 cuMemAllocChannel

```
CUresult CUDAAPI cuMemAllocChannel(CUdeviceptr* dptr, size_t bytesize, uint8_t cluster,
uint8_t channel, uint32_t alignment = 0)
```

Allocate memory on DDR channel.

Parameters

- `dptr`
Out.
Allocated memory.
- `bytesize`
In.
Allocated memory size in byte.
- `cluster`
In.
Cluster ID.
- `channel`
In.
DDR channel ID.
- `alignment`
In.
Memory alignment.

Returns

`cudaSuccess`, `cudaErrorOutOfMemory`

2 cudaSetClusterMask

```
cudaError_t CUDARTAPI cudaSetClusterMask(uint8_t mask)
```

Set the cluster mask to enable clusters for device executions.

Notes

Internal usage only

Parameters

- `mask`
the cluster mask to enable clusters.

Returns

- `cudaSuccess` if the clusters are successfully enabled.

- `cudaErrorInvalidValue` if the clusters are not successfully enabled (please check possible failing reasons below).

Description

The cluster mask specifies the selected clusters on which to execute the subsequent commands. A cluster will be enabled for device executions if the cluster index i satisfies the following condition:

$$((1 \ll i) \& \text{mask}) \neq 0$$

If there exists a cluster index i that is greater than or equal to 4 and satisfies the above condition, `cudaErrorInvalidValue` is returned. If any internal error happens that makes a specified cluster fail to be enabled for subsequent command executions, `cudaErrorInvalidValue` is also returned.

If `cudaSuccess` is not returned, the enabled clusters for device executions are not changed.

All clusters are enabled for device executions by default.

3 `cudaGetClusterMask`

```
cudaGetClusterMask(uint8_t *mask)
```

Get current cluster mask.

Notes

Internal usage only

Parameters

- `mask`
Returned cluster mask

Returns

`cudaSuccess`, `cuda_Error_Invalid_Device`, `cuda_Error_Not_Initialized`, `cuda_Error_Invalid_Context`

Description

Get current cluster mask.

4 `cudaGetSpm`

```
cudaError_t CUDARTAPI cudaGetSpm(void** devPtr)
```

Obtain SPM memory.

Parameters

- `devPtr`
Pointer to SPM memory

Returns

`cudaSuccess`, `cudaErrorInvalidValue`, `cudaErrorInitializationError`, `cudaErrorNotPermitted`

Description

SPM is a block memory on chip, the whole SPM memory is obtained each time when `cudaGetSpm` is called. `devPtr` points to the whole SPM. The SPM memory is not cleared when it is obtained.

`cudaErrorInvalidValue` is returned for invalid input parameter (`devPtr` is `null`).

`cudaErrorInitializationError` is returned for the failure of platform initialization.

`cudaErrorNotPermitted` is returned if `cudaGetSpm` is set to `callback` by `cudaStreamAddCallback`.

Note:

You should make a plan to decide which segment of SPM to use, because the whole SPM memory is obtained every time.

5 `cudaEnterSpmSection`

`cudaError_t CUDARTAPI cudaEnterSpmSection(cudaStream_t stream)`

Enqueues an SPM Lock command to the stream, to lock the SPM resources on the specified clusters.

Notes

Internal usage only

Parameters

- `stream`

The stream where to enter the SPM section, and where the SPM Lock command is enqueued.

Returns

- `cudaSuccess` if the SPM Lock command is successfully submitted.
- `cudaErrorNotPermitted` if the stream is currently being captured.

Description

An SPM section is a command sequence in the stream which begins at the SPM Lock command by `cudaEnterSpmSection` and ends at the SPM Unlock command by `cudaLeaveSpmSection`. The SPM Lock is applied on each of the selected clusters that can be specified by `cudaSetClusterMask`. The SPM Lock ensures that for each cluster only one command from the SPM section is accessing the cluster's SPM at any time.

When called with a `stream_0` and on `cluster_mask_0` (from `cudaSetClusterMask`), `cudaEnterSpmSection` will be blocked if any of the following conditions is true:

- Another call to `cudaEnterSpmSection` with a `stream_1` and on `cluster_mask_1`, where the bitwise AND of `cluster_mask_1` and `cluster_mask_0` is non-zero, has successfully returned, and the corresponding `cudaLeaveSpmSection` has not been called or has not returned `cudaSuccess`.
- Another call to `cudaEnterSpmSection` with the `stream_0` has successfully returned, and the corresponding `cudaLeaveSpmSection` has not been called or has not returned `cudaSuccess`.

If blocked by one of the conditions above, `cudaEnterSpmSection` will be unblocked only after the blocking condition is no longer true.

Constraint

- Any of the following APIs, including `cudaFree/cuMemFree` , `cudaDeviceSynchronize` , `cuCtxSynchronize` and `cudaStreamSynchronize/cuStreamSynchronize` (when called on a different stream) , should not be called in an SPM section from the same thread where the section's `cudaLeaveSpmSection` is to be called, otherwise, deadlocks can potentially occur.
- When called in an SPM section, `cudaStreamWaitEvent` will fail if the recording stream of the event is not the stream where the event is to be waited for, and the recording stream contains SPM commands.
- The stream synchronization behavior between the legacy default stream and blocking streams is turned off in an SPM section.
- `cudaStreamBeginCapture` will fail if called in an SPM section.
- A graph launched in an SPM section can potentially fail if the graph contains some nodes that are to be executed on a cluster whose SPM is not locked in the SPM section.

6 `cudaLeaveSpmSection`

```
cudaError_t CUDARTAPI cudaLeaveSpmSection(cudaStream_t stream)
```

Enqueues an SPM Unlock command to the stream, to unlock the SPM resources previously locked by `cudaEnterSpmSection`.

Notes

Internal usage only

Parameters

- `stream`

The stream where to leave the SPM section, and where the SPM Unlock command is enqueued.

Returns

- `cudaSuccess` if the SPM Unlock command is successfully submitted.
- `cudaErrorInvalidValue` if called from outside of any SPM section.

Description

The SPM Unlock command releases the SPM that are locked by the corresponding SPM Lock command enqueued by `cudaEnterSpmSection`.

Constraint

See **Constraint** in [cudaEnterSpmSection](#).

7 `cuModuleGetFunctionAsync`

```
cuModuleGetFunctionAsync(CUfunction hfunc, CUmodule hmod, const char *name, CUstream hStream)
```

Get function from module, and execute async in given stream.

Parameters

- `Hfunc`

Returned CUFunction

- `Hmod`

Given hc module

- `Name`

Given function name

- `Hstream`

Given hc stream

Returns

`cudaSuccess`, `cuda_Error_Not_Found`, `cuda_Error_Context_Is_Destroyed`
`cuda_Error_Not_Initialized`, `cuda_Error_Invalid_Value`, `cuda_Error_Invalid_Handle`

Description

This is the async version of `cuModuleGetFunction`. Normally, get function from module is blocked, if we need better performance when execution, we can use this aysnc version.

8 `cudaGetDeviceProperties_ext`

```
__host__ __cuda_builtin__ cudaError_t CUDARTAPI cudaGetDeviceProperties ( struct
cudaDeviceProp *prop, int device )
```

Returns information about the compute-device. Returns in *prop the properties of device dev.

Parameters

- `prop`
Properties for the specified device
- `device`
Device number to get properties for

Returns

`cudaSuccess`, `cudaErrorInvalidDevice`

Description

Get information about the compute-device.

Enum

- `Clustercount`
Number of clusters for current device.
- `spmSizePerCluster`
SPM size per cluster.

- ddrChannelNum

DDR channel number of each cluster.

登临科技保密材料