

AC784xx_DFP CORE

6.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	2
2.1	File List	2
3	Class Documentation	3
3.1	Core_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	Irq_Callback	3
3.1.2.2	Irq_Src	4
3.2	Core_EccErrorInfoType Struct Reference	4
3.2.1	Detailed Description	4
3.2.2	Member Data Documentation	4
3.2.2.1	Err_Addr	4
3.2.2.2	Err_Status	5
3.3	Core_FpuErrorInfoType Struct Reference	5
3.3.1	Detailed Description	5
3.3.2	Member Data Documentation	5
3.3.2.1	Err_Status	5

4	File Documentation	6
4.1	AC784xx_API_Reference_Manual_CORE.pdf File Reference	6
4.2	AC784xx_Mcm_Reg.h File Reference	6
4.2.1	Detailed Description	6
4.2.2	Function Documentation	6
4.2.2.1	Mcm_Reg_EnableNmi()	6
4.2.2.2	Mcm_Reg_GetInterruptSources()	7
4.2.2.3	Mcm_Reg_GetInterruptStatus()	7
4.2.2.4	Mcm_Reg_GetMbistStatus()	8
4.2.2.5	Mcm_Reg_SetInterruptSources()	8
4.3	Core_Hal.c File Reference	9
4.3.1	Detailed Description	9
4.3.2	Function Documentation	10
4.3.2.1	Core_Hal_ClearPendingIrq()	10
4.3.2.2	Core_Hal_DeInit()	10
4.3.2.3	Core_Hal_DisableIrq()	10
4.3.2.4	Core_Hal_EnableIrq()	11
4.3.2.5	Core_Hal_EnableNMI()	11
4.3.2.6	Core_Hal_GetChipID()	12
4.3.2.7	Core_Hal_GetIrqPriority()	12
4.3.2.8	Core_Hal_GetIrqPriorityGrouping()	13
4.3.2.9	Core_Hal_GetMBistExecStatus()	13
4.3.2.10	Core_Hal_GetUUID()	13
4.3.2.11	Core_Hal_Init()	14
4.3.2.12	Core_Hal_IsIrqActive()	14
4.3.2.13	Core_Hal_IsIrqEnable()	15
4.3.2.14	Core_Hal_IsIrqPending()	15
4.3.2.15	Core_Hal_PerformReset()	16
4.3.2.16	Core_Hal_SetIrqPriority()	16
4.3.2.17	Core_Hal_SetIrqPriorityGrouping()	16
4.3.2.18	Core_Hal_SetPendingIrq()	17

4.3.2.19	ISR()	17
4.4	Core_Hal.h File Reference	18
4.4.1	Detailed Description	18
4.4.2	Function Documentation	19
4.4.2.1	Core_Hal_ClearPendingIrq()	19
4.4.2.2	Core_Hal_DeInit()	19
4.4.2.3	Core_Hal_DisableIrq()	19
4.4.2.4	Core_Hal_EnableIrq()	20
4.4.2.5	Core_Hal_EnableNMI()	20
4.4.2.6	Core_Hal_GetChipID()	21
4.4.2.7	Core_Hal_GetIrqPriority()	21
4.4.2.8	Core_Hal_GetIrqPriorityGrouping()	22
4.4.2.9	Core_Hal_GetMBistExecStatus()	22
4.4.2.10	Core_Hal_GetUUID()	22
4.4.2.11	Core_Hal_Init()	23
4.4.2.12	Core_Hal_IsIrqActive()	23
4.4.2.13	Core_Hal_IsIrqEnable()	24
4.4.2.14	Core_Hal_IsIrqPending()	24
4.4.2.15	Core_Hal_PerformReset()	25
4.4.2.16	Core_Hal_SetIrqPriority()	25
4.4.2.17	Core_Hal_SetIrqPriorityGrouping()	25
4.4.2.18	Core_Hal_SetPendingIrq()	26
4.5	Core_Hal_Types.h File Reference	26
4.5.1	Detailed Description	26

Chapter 1

Class Index

1.1 Class List

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

Core_ConfigType		
Core configuration structure	3
Core_EccErrorInfoType		
Ecc error information structure	4
Core_FpuErrorInfoType		
FPU error information structure	5

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_CORE.pdf	6
AC784xx_Mcm_Reg.h This file provides extern Low level Mcm register api	6
Core_Hal.c This file provides Hal Core api	9
Core_Hal.h This file provides extern Hal Core api	18
Core_Hal_Types.h This file provides Core hal types header	26

Chapter 3

Class Documentation

3.1 Core_ConfigType Struct Reference

Core configuration structure.

```
#include <Core_Hal_Types.h>
```

Public Attributes

- uint32 [Irq_Src](#)
- Hal_CallbackType [Irq_Callback](#)

3.1.1 Detailed Description

Core configuration structure.

Definition at line 66 of file Core_Hal_Types.h.

3.1.2 Member Data Documentation

3.1.2.1 Irq_Callback

```
Hal_CallbackType Core_ConfigType::Irq_Callback
```

interrupt callback function.

Definition at line 69 of file Core_Hal_Types.h.

3.1.2.2 Irq_Src

```
uint32 Core_ConfigType::Irq_Src
```

Core interrupt sources.

Definition at line 68 of file Core_Hal_Types.h.

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

- [Core_Hal_Types.h](#)

3.2 Core_EccErrorInfoType Struct Reference

Ecc error information structure.

```
#include <Core_Hal_Types.h>
```

Public Attributes

- uint32 [Err_Status](#)
- uint32 [Err_Addr](#)

3.2.1 Detailed Description

Ecc error information structure.

Definition at line 75 of file Core_Hal_Types.h.

3.2.2 Member Data Documentation

3.2.2.1 Err_Addr

```
uint32 Core_EccErrorInfoType::Err_Addr
```

Error address.

Definition at line 78 of file Core_Hal_Types.h.

3.2.2.2 Err_Status

```
uint32 Core_EccErrorInfoType::Err_Status
```

Error Status, 1bit or 2bits ECC error.

Definition at line 77 of file Core_Hal_Types.h.

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

- [Core_Hal_Types.h](#)

3.3 Core_FpuErrorInfoType Struct Reference

FPU error information structure.

```
#include <Core_Hal_Types.h>
```

Public Attributes

- uint32 [Err_Status](#)

3.3.1 Detailed Description

FPU error information structure.

Definition at line 84 of file Core_Hal_Types.h.

3.3.2 Member Data Documentation

3.3.2.1 Err_Status

```
uint32 Core_FpuErrorInfoType::Err_Status
```

Float point exceptions.

Definition at line 86 of file Core_Hal_Types.h.

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

- [Core_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_CORE.pdf File Reference

4.2 AC784xx_Mcm_Reg.h File Reference

This file provides extern Low level Mcm register api.

```
#include "Device_Register.h"
```

Functions

- LOCAL_INLINE uint32 [Mcm_Reg_GetInterruptStatus](#) (void)
Get MCM exceptions.
- LOCAL_INLINE void [Mcm_Reg_SetInterruptSources](#) (uint32 IntSources)
Set which MCM exception trigger interrupt.
- LOCAL_INLINE uint32 [Mcm_Reg_GetInterruptSources](#) (void)
Get MCM exception sources trigger interrupt.
- LOCAL_INLINE uint32 [Mcm_Reg_GetMbistStatus](#) (void)
Get MBIST status.
- LOCAL_INLINE void [Mcm_Reg_EnableNmi](#) (uint32 Enable)
Enable or Disable NMI.

4.2.1 Detailed Description

This file provides extern Low level Mcm register api.

4.2.2 Function Documentation

4.2.2.1 Mcm_Reg_EnableNmi()

```
LOCAL_INLINE void Mcm_Reg_EnableNmi (  
    uint32 Enable )
```

Enable or Disable NMI.

Note

Function ID: DES_STP_API_541

Parameters

in	<i>Enable</i>	NMI enable status
----	---------------	-------------------

Returns

NMI enable status.

Definition at line 168 of file AC784xx_Mcm_Reg.h.

4.2.2.2 Mcm_Reg_GetInterruptSources()

```
LOCAL_INLINE uint32 Mcm_Reg_GetInterruptSources (  
    void )
```

Get MCM exception sources trigger interrupt.

Note

Function ID: DES_STP_API_534

Parameters

in	<i>none</i>	
----	-------------	--

Returns

MCM exception sources which can trigger Mcm interrupt.

Definition at line 101 of file AC784xx_Mcm_Reg.h.

4.2.2.3 Mcm_Reg_GetInterruptStatus()

```
LOCAL_INLINE uint32 Mcm_Reg_GetInterruptStatus (  
    void )
```

Get MCM exceptions.

Note

Function ID: DES_STP_API_532

Parameters

in	<i>none</i>	
----	-------------	--

Returns

MCM exceptions.

Definition at line 75 of file AC784xx_Mcm_Reg.h.

4.2.2.4 Mcm_Reg_GetMbistStatus()

```
LOCAL_INLINE uint32 Mcm_Reg_GetMbistStatus (
    void )
```

Get MBIST status.

Note

Function ID: DES_STP_API_538

Parameters

in	<i>none</i>	
----	-------------	--

Returns

MBIST status.

Definition at line 114 of file AC784xx_Mcm_Reg.h.

4.2.2.5 Mcm_Reg_SetInterruptSources()

```
LOCAL_INLINE void Mcm_Reg_SetInterruptSources (
    uint32 IntSources )
```

Set which MCM exception trigger interrupt.

Note

Function ID: DES_STP_API_533

Parameters

in	<i>IntSources</i>	MCM exception sources which can trigger MCM interrupt.
----	-------------------	--

Returns

none

Definition at line 88 of file AC784xx_Mcm_Reg.h.

4.3 Core_Hal.c File Reference

This file provides Hal Core api.

```
#include "Core_Hal.h"
#include "AC784xx_Mcm_Reg.h"
```

Functions

- [ISR](#) (MCM_IRQHandler)
MCM handler.
- void [Core_Hal_EnableIrq](#) (IRQn_Type IrqNumber)
Enable Irq.
- void [Core_Hal_DisableIrq](#) (IRQn_Type IrqNumber)
Disable Irq.
- boolean [Core_Hal_IsIrqEnable](#) (IRQn_Type IrqNumber)
Get Irq Enable status.
- void [Core_Hal_SetIrqPriority](#) (IRQn_Type IrqNumber, uint32 Priority)
Set Irq priority.
- uint32 [Core_Hal_GetIrqPriority](#) (IRQn_Type IrqNumber)
Get Irq priority.
- void [Core_Hal_PerformReset](#) (void)
Perform software reset.
- void [Core_Hal_SetPendingIrq](#) (IRQn_Type IrqNumber)
Set pending Irq.
- void [Core_Hal_ClearPendingIrq](#) (IRQn_Type IrqNumber)
Clear pending Irq.
- boolean [Core_Hal_IsIrqPending](#) (IRQn_Type IrqNumber)
Get pending Irq.
- boolean [Core_Hal_IsIrqActive](#) (IRQn_Type IrqNumber)
Get Active Irq.
- void [Core_Hal_SetIrqPriorityGrouping](#) (uint32 PriorityGroup)
Set Irq priority grouping.
- uint32 [Core_Hal_GetIrqPriorityGrouping](#) (void)
Get Irq priority grouping.
- void [Core_Hal_GetUUID](#) (uint32 *uuidBuffer)
Get UUID from device.
- uint32 [Core_Hal_GetChipID](#) (void)
Get Chip ID from device.
- void [Core_Hal_EnableNMI](#) (boolean En)
Enable NMI, after set pinmux.
- Core_BistStatusType [Core_Hal_GetMBistExecStatus](#) (void)
Get the MBIST execute status.
- void [Core_Hal_Init](#) (const [Core_ConfigType](#) *ConfigPtr)
Initialize MCM module.
- void [Core_Hal_DeInit](#) (void)
De-Initialize MCM module.

4.3.1 Detailed Description

This file provides Hal Core api.

4.3.2 Function Documentation

4.3.2.1 Core_Hal_ClearPendingIrq()

```
void Core_Hal_ClearPendingIrq (
    IRQn_Type IrqNumber )
```

Clear pending Irq.

Note

Function ID: DES_CORE_API_207

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 150 of file Core_Hal.c.

4.3.2.2 Core_Hal_DeInit()

```
void Core_Hal_DeInit (
    void )
```

De-Initialize MCM module.

Note

Function ID: DES_CORE_API_229

Returns

void

Definition at line 714 of file Core_Hal.c.

4.3.2.3 Core_Hal_DisableIrq()

```
void Core_Hal_DisableIrq (
    IRQn_Type IrqNumber )
```

Disable Irq.

Note

Function ID: DES_CORE_API_201

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 81 of file Core_Hal.c.

4.3.2.4 Core_Hal_EnableIrq()

```
void Core_Hal_EnableIrq (  
    IRQn_Type IrqNumber )
```

Enable Irq.

Note

Function ID: DES_CORE_API_200

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 69 of file Core_Hal.c.

4.3.2.5 Core_Hal_EnableNMI()

```
void Core_Hal_EnableNMI (  
    boolean En )
```

Enable NMI, after set pinmux.

Note

Function ID: DES_CORE_API_214

Parameters

in	<i>En</i>	enable state
----	-----------	--------------

Returns

void

Definition at line 608 of file Core_Hal.c.

4.3.2.6 Core_Hal_GetChipID()

```
uint32 Core_Hal_GetChipID (  
    void )
```

Get Chip ID from device.

Note

Function ID: DES_CORE_API_213

Returns

uint32: Chip ID

Definition at line 597 of file Core_Hal.c.

4.3.2.7 Core_Hal_GetIrqPriority()

```
uint32 Core_Hal_GetIrqPriority (  
    IRQn_Type IrqNumber )
```

Get Irq priority.

Note

Function ID: DES_CORE_API_204

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

uint32: Irq priority

Definition at line 118 of file Core_Hal.c.

4.3.2.8 Core_Hal_GetIrqPriorityGrouping()

```
uint32 Core_Hal_GetIrqPriorityGrouping (
    void )
```

Get Irq priority grouping.

Note

Function ID: DES_CORE_API_211

Returns

uint32: Irq priority grouping.

Definition at line 197 of file Core_Hal.c.

4.3.2.9 Core_Hal_GetMBistExecStatus()

```
Core_BistStatusType Core_Hal_GetMBistExecStatus (
    void )
```

Get the MBIST execute status.

Note

Function ID: DES_CORE_API_217

Returns

Core_BistStatusType: BIST execute status. BIST_NORUN: BIST not run BIST_ERROR: There is a BIST fault occurred. BIST_OK: BIST success. BIST_BUSY: BIST is running

Definition at line 635 of file Core_Hal.c.

4.3.2.10 Core_Hal_GetUUID()

```
void Core_Hal_GetUUID (
    uint32 * uuidBuffer )
```

Get UUID from device.

Note

Function ID: DES_CORE_API_212

Parameters

out	<i>uuidBuffer</i>	UUID buffer
-----	-------------------	-------------

Returns

void

Definition at line 579 of file Core_Hal.c.

4.3.2.11 Core_Hal_Init()

```
void Core_Hal_Init (
    const Core_ConfigType * ConfigPtr )
```

Initialize MCM module.

Note

Function ID: DES_CORE_API_228

Parameters

in	<i>ConfigPtr</i>	Core Configuration.
----	------------------	---------------------

Returns

void

Definition at line 679 of file Core_Hal.c.

4.3.2.12 Core_Hal_IsIrqActive()

```
boolean Core_Hal_IsIrqActive (
    IRQn_Type IrqNumber )
```

Get Active Irq.

Note

Function ID: DES_CORE_API_209

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq is active or not. FALSE: Interrupt status is not active. TRUE: Interrupt status is active.

Definition at line 176 of file Core_Hal.c.

4.3.2.13 Core_Hal_IsIrqEnable()

```
boolean Core_Hal_IsIrqEnable (
    IRQn_Type IrqNumber )
```

Get Irq Enable status.

Note

Function ID: DES_CORE_API_202

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq Enable status. FALSE: Interrupt status is not enable. TRUE: Interrupt status is enable.

Definition at line 94 of file Core_Hal.c.

4.3.2.14 Core_Hal_IsIrqPending()

```
boolean Core_Hal_IsIrqPending (
    IRQn_Type IrqNumber )
```

Get pending Irq.

Note

Function ID: DES_CORE_API_208

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq is pending or not. FALSE: Interrupt status is not active. TRUE: Interrupt status is active.

Definition at line 163 of file Core_Hal.c.

4.3.2.15 Core_Hal_PerformReset()

```
void Core_Hal_PerformReset (
    void )
```

Perform software reset.

Note

Function ID: DES_CORE_API_205

Returns

void

Definition at line 128 of file Core_Hal.c.

4.3.2.16 Core_Hal_SetIrqPriority()

```
void Core_Hal_SetIrqPriority (
    IRQn_Type IrqNumber,
    uint32 Priority )
```

Set Irq priority.

Note

Function ID: DES_CORE_API_203

Parameters

in	<i>IrqNumber</i>	Irq Number.
in	<i>Priority</i>	Irq Priority.

Returns

void

Definition at line 107 of file Core_Hal.c.

4.3.2.17 Core_Hal_SetIrqPriorityGrouping()

```
void Core_Hal_SetIrqPriorityGrouping (
    uint32 PriorityGroup )
```

Set Irq priority grouping.

Note

Function ID: DES_CORE_API_210

Parameters

in	<i>PriorityGroup</i>	Irq Priority grouping.
----	----------------------	------------------------

Returns

void

Definition at line 187 of file Core_Hal.c.

4.3.2.18 Core_Hal_SetPendingIrq()

```
void Core_Hal_SetPendingIrq (
    IRQn_Type IrqNumber )
```

Set pending Irq.

Note

Function ID: DES_CORE_API_206

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 139 of file Core_Hal.c.

4.3.2.19 ISR()

```
ISR (
    MCM_IRQHandler )
```

MCM handler.

Note

Function ID: DES_CORE_API_230

Returns

void

Definition at line 744 of file Core_Hal.c.

4.4 Core_Hal.h File Reference

This file provides extern Hal Core api.

```
#include "Core_Hal_Types.h"
```

Functions

- void [Core_Hal_EnableIrq](#) (IRQn_Type IrqNumber)
Enable Irq.
- void [Core_Hal_DisableIrq](#) (IRQn_Type IrqNumber)
Disable Irq.
- boolean [Core_Hal_IsIrqEnable](#) (IRQn_Type IrqNumber)
Get Irq Enable status.
- void [Core_Hal_SetIrqPriority](#) (IRQn_Type IrqNumber, uint32 Priority)
Set Irq priority.
- uint32 [Core_Hal_GetIrqPriority](#) (IRQn_Type IrqNumber)
Get Irq priority.
- void [Core_Hal_PerformReset](#) (void)
Perform software reset.
- void [Core_Hal_SetPendingIrq](#) (IRQn_Type IrqNumber)
Set pending Irq.
- void [Core_Hal_ClearPendingIrq](#) (IRQn_Type IrqNumber)
Clear pending Irq.
- boolean [Core_Hal_IsIrqPending](#) (IRQn_Type IrqNumber)
Get pending Irq.
- boolean [Core_Hal_IsIrqActive](#) (IRQn_Type IrqNumber)
Get Active Irq.
- void [Core_Hal_SetIrqPriorityGrouping](#) (uint32 PriorityGroup)
Set Irq priority grouping.
- uint32 [Core_Hal_GetIrqPriorityGrouping](#) (void)
Get Irq priority grouping.
- void [Core_Hal_GetUUID](#) (uint32 *uuidBuffer)
Get UUID from device.
- uint32 [Core_Hal_GetChipID](#) (void)
Get Chip ID from device.
- void [Core_Hal_EnableNMI](#) (boolean En)
Enable NMI, after set pinmux.
- Core_BistStatusType [Core_Hal_GetMBistExecStatus](#) (void)
Get the MBIST execute status.
- void [Core_Hal_Init](#) (const [Core_ConfigType](#) *ConfigPtr)
Initialize MCM module.
- void [Core_Hal_DeInit](#) (void)
De-Initialize MCM module.

4.4.1 Detailed Description

This file provides extern Hal Core api.

4.4.2 Function Documentation

4.4.2.1 Core_Hal_ClearPendingIrq()

```
void Core_Hal_ClearPendingIrq (
    IRQn_Type IrqNumber )
```

Clear pending Irq.

Note

Function ID: DES_CORE_API_207

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 150 of file Core_Hal.c.

4.4.2.2 Core_Hal_DeInit()

```
void Core_Hal_DeInit (
    void )
```

De-Initialize MCM module.

Note

Function ID: DES_CORE_API_229

Returns

void

Definition at line 714 of file Core_Hal.c.

4.4.2.3 Core_Hal_DisableIrq()

```
void Core_Hal_DisableIrq (
    IRQn_Type IrqNumber )
```

Disable Irq.

Note

Function ID: DES_CORE_API_201

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 81 of file Core_Hal.c.

4.4.2.4 Core_Hal_EnableIrq()

```
void Core_Hal_EnableIrq (
    IRQn_Type IrqNumber )
```

Enable Irq.

Note

Function ID: DES_CORE_API_200

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 69 of file Core_Hal.c.

4.4.2.5 Core_Hal_EnableNMI()

```
void Core_Hal_EnableNMI (
    boolean En )
```

Enable NMI, after set pinmux.

Note

Function ID: DES_CORE_API_214

Parameters

in	<i>En</i>	enable state
----	-----------	--------------

Returns

void

Definition at line 608 of file Core_Hal.c.

4.4.2.6 Core_Hal_GetChipID()

```
uint32 Core_Hal_GetChipID (
    void )
```

Get Chip ID from device.

Note

Function ID: DES_CORE_API_213

Returns

uint32: Chip ID

Definition at line 597 of file Core_Hal.c.

4.4.2.7 Core_Hal_GetIrqPriority()

```
uint32 Core_Hal_GetIrqPriority (
    IRQn_Type IrqNumber )
```

Get Irq priority.

Note

Function ID: DES_CORE_API_204

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

uint32: Irq priority

Definition at line 118 of file Core_Hal.c.

4.4.2.8 Core_Hal_GetIrqPriorityGrouping()

```
uint32 Core_Hal_GetIrqPriorityGrouping (
    void )
```

Get Irq priority grouping.

Note

Function ID: DES_CORE_API_211

Returns

Irq priority grouping.

Note

Function ID: DES_CORE_API_211

Returns

uint32: Irq priority grouping.

Definition at line 197 of file Core_Hal.c.

4.4.2.9 Core_Hal_GetMBistExecStatus()

```
Core_BistStatusType Core_Hal_GetMBistExecStatus (
    void )
```

Get the MBIST execute status.

Note

Function ID: DES_CORE_API_217

Returns

Core_BistStatusType: BIST execute status. BIST_NORUN: BIST not run BIST_ERROR: There is a BIST fault occurred. BIST_OK: BIST success. BIST_BUSY: BIST is running

Definition at line 635 of file Core_Hal.c.

4.4.2.10 Core_Hal_GetUUID()

```
void Core_Hal_GetUUID (
    uint32 * uuidBuffer )
```

Get UUID from device.

Note

Function ID: DES_CORE_API_212

Parameters

out	<i>uuidBuffer</i>	UUID buffer
-----	-------------------	-------------

Returns

void

Definition at line 579 of file Core_Hal.c.

4.4.2.11 Core_Hal_Init()

```
void Core_Hal_Init (
    const Core_ConfigType * ConfigPtr )
```

Initialize MCM module.

Note

Function ID: DES_CORE_API_228

Parameters

in	<i>ConfigPtr</i>	Core Configuration.
----	------------------	---------------------

Returns

void

Definition at line 679 of file Core_Hal.c.

4.4.2.12 Core_Hal_IsIrqActive()

```
boolean Core_Hal_IsIrqActive (
    IRQn_Type IrqNumber )
```

Get Active Irq.

Note

Function ID: DES_CORE_API_209

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq is active or not. FALSE: Interrupt status is not active. TRUE: Interrupt status is active.

Definition at line 176 of file Core_Hal.c.

4.4.2.13 Core_Hal_IsIrqEnable()

```
boolean Core_Hal_IsIrqEnable (
    IRQn_Type IrqNumber )
```

Get Irq Enable status.

Note

Function ID: DES_CORE_API_202

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq Enable status. FALSE: Interrupt status is not enable. TRUE: Interrupt status is enable.

Definition at line 94 of file Core_Hal.c.

4.4.2.14 Core_Hal_IsIrqPending()

```
boolean Core_Hal_IsIrqPending (
    IRQn_Type IrqNumber )
```

Get pending Irq.

Note

Function ID: DES_CORE_API_208

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

boolean: Irq is pending or not. FALSE: Interrupt status is not active. TRUE: Interrupt status is active.

Definition at line 163 of file Core_Hal.c.

4.4.2.15 Core_Hal_PerformReset()

```
void Core_Hal_PerformReset (
    void )
```

Perform software reset.

Note

Function ID: DES_CORE_API_205

Returns

void

Definition at line 128 of file Core_Hal.c.

4.4.2.16 Core_Hal_SetIrqPriority()

```
void Core_Hal_SetIrqPriority (
    IRQn_Type IrqNumber,
    uint32 Priority )
```

Set Irq priority.

Note

Function ID: DES_CORE_API_203

Parameters

in	<i>IrqNumber</i>	Irq Number.
in	<i>Priority</i>	Irq Priority.

Returns

void

Definition at line 107 of file Core_Hal.c.

4.4.2.17 Core_Hal_SetIrqPriorityGrouping()

```
void Core_Hal_SetIrqPriorityGrouping (
    uint32 PriorityGroup )
```

Set Irq priority grouping.

Note

Function ID: DES_CORE_API_210

Parameters

in	<i>PriorityGroup</i>	Irq Priority grouping.
----	----------------------	------------------------

Returns

void

Definition at line 187 of file Core_Hal.c.

4.4.2.18 Core_Hal_SetPendingIrq()

```
void Core_Hal_SetPendingIrq (
    IRQn_Type IrqNumber )
```

Set pending Irq.

Note

Function ID: DES_CORE_API_206

Parameters

in	<i>IrqNumber</i>	Irq Number.
----	------------------	-------------

Returns

void

Definition at line 139 of file Core_Hal.c.

4.5 Core_Hal_Types.h File Reference

This file provides Core hal types header.

#include "Device_Register.h"

Classes

- struct [Core_ConfigType](#)
Core configuration structure.
- struct [Core_EccErrorInfoType](#)
Ecc error information structure.
- struct [Core_FpuErrorInfoType](#)
FPU error information structure.

4.5.1 Detailed Description

This file provides Core hal types header.

Index

AC784xx_API_Reference_Manual_CORE.pdf, [6](#)
AC784xx_Mcm_Reg.h, [6](#)
 Mcm_Reg_EnableNmi, [6](#)
 Mcm_Reg_GetInterruptSources, [7](#)
 Mcm_Reg_GetInterruptStatus, [7](#)
 Mcm_Reg_GetMbistStatus, [8](#)
 Mcm_Reg_SetInterruptSources, [8](#)

Core_ConfigType, [3](#)
 Irq_Callback, [3](#)
 Irq_Src, [3](#)
Core_EccErrorInfoType, [4](#)
 Err_Addr, [4](#)
 Err_Status, [4](#)
Core_FpuErrorInfoType, [5](#)
 Err_Status, [5](#)
Core_Hal.c, [9](#)
 Core_Hal_ClearPendingIrq, [10](#)
 Core_Hal_DeInit, [10](#)
 Core_Hal_DisableIrq, [10](#)
 Core_Hal_EnableIrq, [11](#)
 Core_Hal_EnableNMI, [11](#)
 Core_Hal_GetChipID, [12](#)
 Core_Hal_GetIrqPriority, [12](#)
 Core_Hal_GetIrqPriorityGrouping, [12](#)
 Core_Hal_GetMBistExecStatus, [13](#)
 Core_Hal_GetUUID, [13](#)
 Core_Hal_Init, [14](#)
 Core_Hal_IsIrqActive, [14](#)
 Core_Hal_IsIrqEnable, [15](#)
 Core_Hal_IsIrqPending, [15](#)
 Core_Hal_PerformReset, [15](#)
 Core_Hal_SetIrqPriority, [16](#)
 Core_Hal_SetIrqPriorityGrouping, [16](#)
 Core_Hal_SetPendingIrq, [17](#)
 ISR, [17](#)
Core_Hal.h, [18](#)
 Core_Hal_ClearPendingIrq, [19](#)
 Core_Hal_DeInit, [19](#)
 Core_Hal_DisableIrq, [19](#)
 Core_Hal_EnableIrq, [20](#)
 Core_Hal_EnableNMI, [20](#)
 Core_Hal_GetChipID, [21](#)
 Core_Hal_GetIrqPriority, [21](#)
 Core_Hal_GetIrqPriorityGrouping, [21](#)
 Core_Hal_GetMBistExecStatus, [22](#)
 Core_Hal_GetUUID, [22](#)
 Core_Hal_Init, [23](#)
 Core_Hal_IsIrqActive, [23](#)
 Core_Hal_IsIrqEnable, [24](#)
 Core_Hal_IsIrqPending, [24](#)
 Core_Hal_PerformReset, [24](#)
 Core_Hal_SetIrqPriority, [25](#)
 Core_Hal_SetIrqPriorityGrouping, [25](#)
 Core_Hal_SetPendingIrq, [26](#)
Core_Hal_ClearPendingIrq
 Core_Hal.c, [10](#)
 Core_Hal.h, [19](#)
Core_Hal_DeInit
 Core_Hal.c, [10](#)
 Core_Hal.h, [19](#)
Core_Hal_DisableIrq
 Core_Hal.c, [10](#)
 Core_Hal.h, [19](#)
Core_Hal_EnableIrq
 Core_Hal.c, [11](#)
 Core_Hal.h, [20](#)
Core_Hal_EnableNMI
 Core_Hal.c, [11](#)
 Core_Hal.h, [20](#)
Core_Hal_GetChipID
 Core_Hal.c, [12](#)
 Core_Hal.h, [21](#)
Core_Hal_GetIrqPriority
 Core_Hal.c, [12](#)
 Core_Hal.h, [21](#)
Core_Hal_GetIrqPriorityGrouping
 Core_Hal.c, [12](#)
 Core_Hal.h, [21](#)
Core_Hal_GetMBistExecStatus
 Core_Hal.c, [13](#)
 Core_Hal.h, [22](#)
Core_Hal_GetUUID
 Core_Hal.c, [13](#)
 Core_Hal.h, [22](#)
Core_Hal_Init
 Core_Hal.c, [14](#)
 Core_Hal.h, [23](#)
Core_Hal_IsIrqActive
 Core_Hal.c, [14](#)
 Core_Hal.h, [23](#)
Core_Hal_IsIrqEnable
 Core_Hal.c, [15](#)
 Core_Hal.h, [24](#)
Core_Hal_IsIrqPending
 Core_Hal.c, [15](#)
 Core_Hal.h, [24](#)
Core_Hal_PerformReset
 Core_Hal.c, [15](#)
 Core_Hal.h, [24](#)
Core_Hal_SetIrqPriority
 Core_Hal.c, [16](#)
 Core_Hal.h, [25](#)
Core_Hal_SetIrqPriorityGrouping

- Core_Hal.c, [16](#)
- Core_Hal.h, [25](#)
- Core_Hal_SetPendingIrq
 - Core_Hal.c, [17](#)
 - Core_Hal.h, [26](#)
- Core_Hal_Types.h, [26](#)
- Err_Addr
 - Core_EccErrorInfoType, [4](#)
- Err_Status
 - Core_EccErrorInfoType, [4](#)
 - Core_FpuErrorInfoType, [5](#)
- ISR
 - Core_Hal.c, [17](#)
- Irq_Callback
 - Core_ConfigType, [3](#)
- Irq_Src
 - Core_ConfigType, [3](#)
- Mcm_Reg_EnableNmi
 - AC784xx_Mcm_Reg.h, [6](#)
- Mcm_Reg_GetInterruptSources
 - AC784xx_Mcm_Reg.h, [7](#)
- Mcm_Reg_GetInterruptStatus
 - AC784xx_Mcm_Reg.h, [7](#)
- Mcm_Reg_GetMbistStatus
 - AC784xx_Mcm_Reg.h, [8](#)
- Mcm_Reg_SetInterruptSources
 - AC784xx_Mcm_Reg.h, [8](#)