

AC784xx_DFP RCM
3.1.0

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	Rcm_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	DelayTime	3
3.1.2.2	FilterValue	4
3.1.2.3	InterruptSource	4
3.1.2.4	RcmCallback	4
3.1.2.5	ResetSource	4
4	File Documentation	5
4.1	AC784xx_API_Reference_Manual_RCM.pdf File Reference	5
4.2	AC784xx_Rcm_Reg.h File Reference	5
4.2.1	Detailed Description	6
4.2.2	Function Documentation	6
4.2.2.1	Rcm_Reg_ClearResetInterruptStatus()	6
4.2.2.2	Rcm_Reg_ClearResetStatus()	6
4.2.2.3	Rcm_Reg_EnableExtResetFilter()	6
4.2.2.4	Rcm_Reg_EnableGlobalResetInterrupt()	7
4.2.2.5	Rcm_Reg_EnableResetInterrupts()	7

4.2.2.6	Rcm_Reg_EnableResetSources()	8
4.2.2.7	Rcm_Reg_GetResetInterruptSources()	8
4.2.2.8	Rcm_Reg_GetResetInterruptStatus()	9
4.2.2.9	Rcm_Reg_GetResetSources()	9
4.2.2.10	Rcm_Reg_GetResetStatus()	10
4.2.2.11	Rcm_Reg_SetExtResetFilter()	10
4.2.2.12	Rcm_Reg_SetResetDelayTime()	10
4.2.2.13	Rcm_Reg_SetResetState()	11
4.3	Rcm_Hal.c File Reference	11
4.3.1	Detailed Description	12
4.3.2	Function Documentation	12
4.3.2.1	ISR()	12
4.3.2.2	Rcm_Hal_ClearResetStatus()	12
4.3.2.3	Rcm_Hal_GetResetStatus()	13
4.3.2.4	Rcm_Hal_Init()	13
4.3.2.5	Rcm_Hal_SetResetState()	14
4.4	Rcm_Hal.h File Reference	14
4.4.1	Detailed Description	15
4.4.2	Macro Definition Documentation	15
4.4.2.1	RCM_IRQ_CONTROL_INTERNAL	15
4.4.2.2	RESET_FILTER_MAX_VALUE	16
4.4.3	Enumeration Type Documentation	16
4.4.3.1	Rcm_DelayType	16
4.4.3.2	Rcm_ResetIDType	16
4.4.3.3	Rcm_ResetStateType	17
4.4.4	Function Documentation	17
4.4.4.1	Rcm_Hal_ClearResetStatus()	17
4.4.4.2	Rcm_Hal_GetResetStatus()	18
4.4.4.3	Rcm_Hal_Init()	18
4.4.4.4	Rcm_Hal_SetResetState()	19

Chapter 1

Class Index

1.1 Class List

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

Rcm_ConfigType	
Rcm module configuration structure	3

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_RCM.pdf	5
AC784xx_Rcm_Reg.h This file provides extern Low level Rcm register api	5
Rcm_Hal.c This file provides Hal Rcm api	11
Rcm_Hal.h This file provides extern Hal Rcm api	14

Chapter 3

Class Documentation

3.1 Rcm_ConfigType Struct Reference

Rcm module configuration structure.

```
#include <Rcm_Hal.h>
```

Public Attributes

- uint32 [InterruptSource](#)
- uint32 [ResetSource](#)
- uint8 [FilterValue](#)
- Hal_CallbackType [RcmCallback](#)
- [Rcm_DelayType](#) [DelayTime](#)

3.1.1 Detailed Description

Rcm module configuration structure.

Definition at line 177 of file Rcm_Hal.h.

3.1.2 Member Data Documentation

3.1.2.1 DelayTime

[Rcm_DelayType](#) Rcm_ConfigType::DelayTime

delay time before reset

Definition at line 183 of file Rcm_Hal.h.

3.1.2.2 FilterValue

```
uint8 Rcm_ConfigType::FilterValue
```

filter value of external reset pid

Definition at line 181 of file Rcm_Hal.h.

3.1.2.3 InterruptSource

```
uint32 Rcm_ConfigType::InterruptSource
```

sources can generate interrupt before reset

Definition at line 179 of file Rcm_Hal.h.

3.1.2.4 RcmCallback

```
Hal_CallbackType Rcm_ConfigType::RcmCallback
```

callback function

Definition at line 182 of file Rcm_Hal.h.

3.1.2.5 ResetSource

```
uint32 Rcm_ConfigType::ResetSource
```

sources can generate reset

Definition at line 180 of file Rcm_Hal.h.

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

- [Rcm_Hal.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_RCM.pdf File Reference

4.2 AC784xx_Rcm_Reg.h File Reference

This file provides extern Low level Rcm register api.

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

Functions

- LOCAL_INLINE void [Rcm_Reg_EnableGlobalResetInterrupt](#) (boolean En)
Enable or disable global reset interrupt.
- LOCAL_INLINE void [Rcm_Reg_EnableResetInterrupts](#) (uint32 RstSources)
Set reset sources trigger interrupt.
- LOCAL_INLINE uint32 [Rcm_Reg_GetResetInterruptSources](#) (void)
Get reset sources trigger interrupt.
- LOCAL_INLINE void [Rcm_Reg_SetResetDelayTime](#) (uint32 Time)
Set reset delay time.
- LOCAL_INLINE void [Rcm_Reg_SetExtResetFilter](#) (uint32 FilterVal)
Set ext reset filter value.
- LOCAL_INLINE void [Rcm_Reg_EnableExtResetFilter](#) (boolean IsEnable)
Enable or disable external reset filter.
- LOCAL_INLINE void [Rcm_Reg_EnableResetSources](#) (uint32 RstSources)
Set reset sources trigger reset.
- LOCAL_INLINE uint32 [Rcm_Reg_GetResetSources](#) (void)
Get reset sources.
- LOCAL_INLINE void [Rcm_Reg_SetResetState](#) (uint32 ResetId, uint32 ResetState)
Assert or deassert the reset.
- LOCAL_INLINE uint32 [Rcm_Reg_GetResetStatus](#) (void)
Get reset and interrupt status.
- LOCAL_INLINE uint32 [Rcm_Reg_GetResetInterruptStatus](#) (void)
Get reset and interrupt status.
- LOCAL_INLINE void [Rcm_Reg_ClearResetInterruptStatus](#) (void)
Clear all reset interrupt status.
- LOCAL_INLINE void [Rcm_Reg_ClearResetStatus](#) (void)
Clear all reset status.

4.2.1 Detailed Description

This file provides extern Low level Rcm register api.

4.2.2 Function Documentation

4.2.2.1 Rcm_Reg_ClearResetInterruptStatus()

```
LOCAL_INLINE void Rcm_Reg_ClearResetInterruptStatus (
    void )
```

Clear all reset interrupt status.

Note

Function ID: DES_MCU_API_318

Returns

void

Definition at line 210 of file AC784xx_Rcm_Reg.h.

4.2.2.2 Rcm_Reg_ClearResetStatus()

```
LOCAL_INLINE void Rcm_Reg_ClearResetStatus (
    void )
```

Clear all reset status.

Note

Function ID: DES_MCU_API_319

Returns

void

Definition at line 220 of file AC784xx_Rcm_Reg.h.

4.2.2.3 Rcm_Reg_EnableExtResetFilter()

```
LOCAL_INLINE void Rcm_Reg_EnableExtResetFilter (
    boolean IsEnable )
```

Enable or disable external reset filter.

Note

Function ID: DES_MCU_API_311

Parameters

in	<i>IsEnable</i>	enable or disable external reset filter <ul style="list-style-type: none">• true: enable ext reset filter• false: disable ext reset filter
----	-----------------	---

Returns

void

Definition at line 132 of file AC784xx_Rcm_Reg.h.

4.2.2.4 Rcm_Reg_EnableGlobalResetInterrupt()

```
LOCAL_INLINE void Rcm_Reg_EnableGlobalResetInterrupt (
    boolean En )
```

Enable or disable global reset interrupt.

Note

Function ID: DES_MCU_API_306

Parameters

in	<i>En</i>	enable state <ul style="list-style-type: none">• true: enable global reset interrupt• false: disable global reset interrupt
----	-----------	--

Returns

void

Definition at line 73 of file AC784xx_Rcm_Reg.h.

4.2.2.5 Rcm_Reg_EnableResetInterrupts()

```
LOCAL_INLINE void Rcm_Reg_EnableResetInterrupts (
    uint32 RstSources )
```

Set reset sources trigger interrupt.

Note

Function ID: DES_MCU_API_307

Parameters

in	<i>RstSources</i>	reset sources which can trigger RCM interrupt
----	-------------------	---

Returns

void

Definition at line 84 of file AC784xx_Rcm_Reg.h.

4.2.2.6 Rcm_Reg_EnableResetSources()

```
LOCAL_INLINE void Rcm_Reg_EnableResetSources (
    uint32 RstSources )
```

Set reset sources trigger reset.

Note

Function ID: DES_MCU_API_312

Parameters

in	<i>RstSources</i>	reset sources which can trigger RCM reset
----	-------------------	---

Returns

void

Definition at line 144 of file AC784xx_Rcm_Reg.h.

4.2.2.7 Rcm_Reg_GetResetInterruptSources()

```
LOCAL_INLINE uint32 Rcm_Reg_GetResetInterruptSources (
    void )
```

Get reset sources trigger interrupt.

Note

Function ID: DES_MCU_API_308

Returns

reset sources which can trigger RCM interrupt

Definition at line 95 of file AC784xx_Rcm_Reg.h.

4.2.2.8 Rcm_Reg_GetResetInterruptStatus()

```
LOCAL_INLINE uint32 Rcm_Reg_GetResetInterruptStatus (
    void )
```

Get reset and interrupt status.

Note

Function ID: DES_MCU_API_317

Returns

value of reset and interrupt status

- BIT0:RCM_RESET_INT_STATUS_SW_RST
- BIT1:RCM_RESET_INT_STATUS_LOCKUP_RST
- BIT2:RCM_RESET_INT_STATUS_ACK_ERR_RST
- BIT3:RCM_RESET_INT_STATUS_WDG_RST
- BIT5:RCM_RESET_INT_STATUS_XOSC_LOSS_RST
- BIT6:RCM_RESET_INT_STATUS_PLL_UNLOCK_RST
- BIT7:RCM_RESET_INT_STATUS_VHSI_LOSS_RST
- BIT9:RCM_RESET_INT_STATUS_SMU_ERR_RST

Definition at line 200 of file AC784xx_Rcm_Reg.h.

4.2.2.9 Rcm_Reg_GetResetSources()

```
LOCAL_INLINE uint32 Rcm_Reg_GetResetSources (
    void )
```

Get reset sources.

Note

Function ID: DES_MCU_API_313

Returns

reset sources which can trigger RCM reset.

Definition at line 154 of file AC784xx_Rcm_Reg.h.

4.2.2.10 Rcm_Reg_GetResetStatus()

```
LOCAL_INLINE uint32 Rcm_Reg_GetResetStatus (
    void )
```

Get reset and interrupt status.

Note

Function ID: DES_MCU_API_317

Returns

value of reset and interrupt status

Definition at line 182 of file AC784xx_Rcm_Reg.h.

4.2.2.11 Rcm_Reg_SetExtResetFilter()

```
LOCAL_INLINE void Rcm_Reg_SetExtResetFilter (
    uint32 FilterVal )
```

Set ext reset filter value.

Note

Function ID: DES_MCU_API_310

Parameters

in	<i>FilterVal</i>	filter value: 0~127, means filter width value is N * 128KHz.
----	------------------	--

Returns

void

Definition at line 118 of file AC784xx_Rcm_Reg.h.

4.2.2.12 Rcm_Reg_SetResetDelayTime()

```
LOCAL_INLINE void Rcm_Reg_SetResetDelayTime (
    uint32 Time )
```

Set reset delay time.

Note

Function ID: DES_MCU_API_309

Parameters

in	<i>Time</i>	system reset delay time set
----	-------------	-----------------------------

Returns

void

Definition at line 107 of file AC784xx_Rcm_Reg.h.

4.2.2.13 Rcm_Reg_SetResetState()

```
LOCAL_INLINE void Rcm_Reg_SetResetState (
    uint32 ResetId,
    uint32 ResetState )
```

Assert or deassert the reset.

Note

Function ID: DES_MCU_API_315

Parameters

in	<i>ResetId</i>	Reset ID
in	<i>ResetState</i>	Reset state, assert or deassert

Returns

void

Definition at line 166 of file AC784xx_Rcm_Reg.h.

4.3 Rcm_Hal.c File Reference

This file provides Hal Rcm api.

```
#include "Rcm_Hal.h"
#include "Core_Hal.h"
#include "AC784xx_Rcm_Reg.h"
```

Functions

- [ISR](#) (RCM_IRQHandler)
reset interrupt.
- void [Rcm_Hal_Init](#) (const [Rcm_ConfigType](#) *ConfigPtr)

Initialize reset module.

- uint32 [Rcm_Hal_GetResetStatus](#) (void)

Get reset statue.

- void [Rcm_Hal_ClearResetStatus](#) (void)

Clear All reset status.

- void [Rcm_Hal_SetResetState](#) ([Rcm_ResetIDType](#) ResetId, [Rcm_ResetStateType](#) ResetState)

Assert or deassert the reset.

4.3.1 Detailed Description

This file provides Hal Rcm api.

4.3.2 Function Documentation

4.3.2.1 ISR()

```
ISR (
    RCM_IRQHandler )
```

reset interrupt.

Note

Function ID: DES_MCU_API_405

Returns

void

Definition at line 164 of file Rcm_Hal.c.

4.3.2.2 Rcm_Hal_ClearResetStatus()

```
void Rcm_Hal_ClearResetStatus (
    void )
```

Clear All reset status.

Note

Function ID:DES_MCU_API_403

Returns

void

Definition at line 142 of file Rcm_Hal.c.

4.3.2.3 Rcm_Hal_GetResetStatus()

```
uint32 Rcm_Hal_GetResetStatus (
    void )
```

Get reset statue.

Note

Function ID: DES_MCU_API_402

Returns

value of reset and interrupt status

- BIT0:RCM_RESET_STATUS_POR_RST
- BIT1:RCM_RESET_STATUS_LVR_RST
- BIT2:RCM_RESET_STATUS_SW_RST
- BIT3:RCM_RESET_STATUS_LOCKUP_RST
- BIT4:RCM_RESET_STATUS_ACK_ERR_RST
- BIT5:RCM_RESET_STATUS_WDG_RST
- BIT7:RCM_RESET_STATUS_XOSC_LOSS_RST
- BIT8:RCM_RESET_STATUS_PLL_UNLOCK_RST
- BIT9:RCM_RESET_STATUS_VHSI_LOSS_RST
- BIT10:RCM_RESET_STATUS_EXT_RST
- BIT11:RCM_RESET_STATUS_SMU_ERR_RST
- BIT12:RCM_RESET_STATUS_ECC2_ERR_RST

Definition at line 132 of file Rcm_Hal.c.

4.3.2.4 Rcm_Hal_Init()

```
void Rcm_Hal_Init (
    const Rcm_ConfigType * ConfigPtr )
```

Initialize reset module.

Note

Function ID: DES_MCU_API_401

Parameters

in	<i>ConfigPtr</i>	Rcm configuration.
----	------------------	--------------------

Returns

void

Definition at line 68 of file Rcm_Hal.c.

4.3.2.5 Rcm_Hal_SetResetState()

```
void Rcm_Hal_SetResetState (
    Rcm_ResetIDType ResetId,
    Rcm_ResetStateType ResetState )
```

Assert or deassert the reset.

Note

Function ID:DES_MCU_API_404

Parameters

in	<i>ResetId</i>	Reset ID
in	<i>ResetState</i>	Reset state, assert or deassert

Returns

void

Definition at line 154 of file Rcm_Hal.c.

4.4 Rcm_Hal.h File Reference

This file provides extern Hal Rcm api.

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

Classes

- struct [Rcm_ConfigType](#)
Rcm module configuration structure.

Macros

- #define [RESET_FILTER_MAX_VALUE](#) (127UL)
- #define [RCM_IRQ_CONTROL_INTERNAL](#) (STD_ON)
rcm irq control internal(STD_ON) or external(STD_OFF)

Enumerations

- enum `Rcm_DelayType` { `RCM_RESET_DELAY_8_128KHz` = 0U, `RCM_RESET_DELAY_32_128KHz`, `RCM_RESET_DELAY_128_128KHz`, `RCM_RESET_DELAY_512_128KHz` }
Reset delay time enumeration.
- enum `Rcm_ResetIDType` {
`RCM_RESET_ID_UART0` = 0U, `RCM_RESET_ID_UART1` = 1U, `RCM_RESET_ID_UART2` = 2U, `RCM_RESET_ID_UART3` = 3U,
`RCM_RESET_ID_DMA` = 34U, `RCM_RESET_ID_GPIO` = 37U, `RCM_RESET_ID_WDG` = 38U, `RCM_RESET_ID_EWDG` = 39U,
`RCM_RESET_ID_CRC` = 40U, `RCM_RESET_ID_CAN0` = 41U, `RCM_RESET_ID_CAN1` = 42U, `RCM_RESET_ID_CAN2` = 43U,
`RCM_RESET_ID_CAN3` = 44U, `RCM_RESET_ID_CTU` = 65U, `RCM_RESET_ID_PDT0` = 73U, `RCM_RESET_ID_PDT1` = 74U,
`RCM_RESET_ID_ADC0` = 75U, `RCM_RESET_ID_ADC1` = 76U, `RCM_RESET_ID_TIMER` = 77U, `RCM_RESET_ID_EIO` = 78U,
`RCM_RESET_ID_MAX` }
- enum `Rcm_ResetStateType` { `RCM_RESET_STATE_ASSERT` = 0U, `RCM_RESET_STATE_DEASSERT` = 1U }

Functions

- void `Rcm_Hal_Init` (const `Rcm_ConfigType` *ConfigPtr)
Initialize reset module.
- uint32 `Rcm_Hal_GetResetStatus` (void)
Get reset statue.
- void `Rcm_Hal_ClearResetStatus` (void)
Clear All reset status.
- void `Rcm_Hal_SetResetState` (`Rcm_ResetIDType` ResetId, `Rcm_ResetStateType` ResetState)
Assert or deassert the reset.

4.4.1 Detailed Description

This file provides extern Hal Rcm api.

4.4.2 Macro Definition Documentation

4.4.2.1 RCM_IRQ_CONTROL_INTERNAL

```
#define RCM_IRQ_CONTROL_INTERNAL (STD_ON)
```

rcm irq control internal(STD_ON) or external(STD_OFF)

Definition at line 62 of file Rcm_Hal.h.

4.4.2.2 RESET_FILTER_MAX_VALUE

```
#define RESET_FILTER_MAX_VALUE (127UL)
```

< rcm reset filter max value

Definition at line 59 of file Rcm_Hal.h.

4.4.3 Enumeration Type Documentation

4.4.3.1 Rcm_DelayType

```
enum Rcm_DelayType
```

Reset delay time enumeration.

Enumerator

RCM_RESET_DELAY_8_128KHz	Reset delay 8 cycles * 128K
RCM_RESET_DELAY_32_128KHz	Reset delay 32 cycles * 128K
RCM_RESET_DELAY_128_128KHz	Reset delay 128 cycles * 128K
RCM_RESET_DELAY_512_128KHz	Reset delay 512 cycles * 128K

Definition at line 68 of file Rcm_Hal.h.

4.4.3.2 Rcm_ResetIDType

```
enum Rcm_ResetIDType
```

Enumerator

RCM_RESET_ID_UART0	Soft reset UART0
RCM_RESET_ID_UART1	Soft reset UART1
RCM_RESET_ID_UART2	Soft reset UART2
RCM_RESET_ID_UART3	Soft reset UART3
RCM_RESET_ID_DMA	Soft reset DMA
RCM_RESET_ID_GPIO	Soft reset GPIO
RCM_RESET_ID_WDG	Soft reset WDG
RCM_RESET_ID_EWDG	Soft reset EWDG
RCM_RESET_ID_CRC	Soft reset CRC
RCM_RESET_ID_CAN0	Soft reset CAN0
RCM_RESET_ID_CAN1	Soft reset CAN1
RCM_RESET_ID_CAN2	Soft reset CAN2
RCM_RESET_ID_CAN3	Soft reset CAN3
RCM_RESET_ID_CTU	Soft reset CTU
RCM_RESET_ID_PDT0	Soft reset PDT0

Enumerator

RCM_RESET_ID_PDT1	Soft reset PDT1
RCM_RESET_ID_ADC0	Soft reset ADC0
RCM_RESET_ID_ADC1	Soft reset ADC1
RCM_RESET_ID_TIMER	Soft reset TIMER
RCM_RESET_ID_EIO	Soft reset EIO
RCM_RESET_ID_MAX	

Definition at line 76 of file Rcm_Hal.h.

4.4.3.3 Rcm_ResetStateType

```
enum Rcm_ResetStateType
```

Enumerator

RCM_RESET_STATE_ASSERT	Assert
RCM_RESET_STATE_DEASSERT	Deassert

Definition at line 167 of file Rcm_Hal.h.

4.4.4 Function Documentation**4.4.4.1 Rcm_Hal_ClearResetStatus()**

```
void Rcm_Hal_ClearResetStatus (
    void )
```

Clear All reset status.

Note

Function ID:DES_MCU_API_403

Returns

void

Definition at line 142 of file Rcm_Hal.c.

4.4.4.2 Rcm_Hal_GetResetStatus()

```
uint32 Rcm_Hal_GetResetStatus (
    void )
```

Get reset statue.

Note

Function ID: DES_MCU_API_402

Returns

value of reset and interrupt status

- BIT0:RCM_RESET_STATUS_POR_RST
- BIT1:RCM_RESET_STATUS_LVR_RST
- BIT2:RCM_RESET_STATUS_SW_RST
- BIT3:RCM_RESET_STATUS_LOCKUP_RST
- BIT4:RCM_RESET_STATUS_ACK_ERR_RST
- BIT5:RCM_RESET_STATUS_WDG_RST
- BIT7:RCM_RESET_STATUS_XOSC_LOSS_RST
- BIT8:RCM_RESET_STATUS_PLL_UNLOCK_RST
- BIT9:RCM_RESET_STATUS_VHSI_LOSS_RST
- BIT10:RCM_RESET_STATUS_EXT_RST
- BIT11:RCM_RESET_STATUS_SMU_ERR_RST
- BIT12:RCM_RESET_STATUS_ECC2_ERR_RST

Definition at line 132 of file Rcm_Hal.c.

4.4.4.3 Rcm_Hal_Init()

```
void Rcm_Hal_Init (
    const Rcm_ConfigType * ConfigPtr )
```

Initialize reset module.

Note

Function ID: DES_MCU_API_401

Parameters

in	<i>ConfigPtr</i>	Rcm configuration.
----	------------------	--------------------

Returns

void

Definition at line 68 of file Rcm_Hal.c.

4.4.4.4 Rcm_Hal_SetResetState()

```
void Rcm_Hal_SetResetState (
    Rcm_ResetIDType ResetId,
    Rcm_ResetStateType ResetState )
```

Assert or deassert the reset.

Note

Function ID:DES_MCU_API_404

Parameters

in	<i>ResetId</i>	Reset ID
in	<i>ResetState</i>	Reset state, assert or deassert

Returns

void

Definition at line 154 of file Rcm_Hal.c.

Index

AC784xx_API_Reference_Manual_RCM.pdf, 5
AC784xx_Rcm_Reg.h, 5
 Rcm_Reg_ClearResetInterruptStatus, 6
 Rcm_Reg_ClearResetStatus, 6
 Rcm_Reg_EnableExtResetFilter, 6
 Rcm_Reg_EnableGlobalResetInterrupt, 7
 Rcm_Reg_EnableResetInterrupts, 7
 Rcm_Reg_EnableResetSources, 8
 Rcm_Reg_GetResetInterruptSources, 8
 Rcm_Reg_GetResetInterruptStatus, 8
 Rcm_Reg_GetResetSources, 9
 Rcm_Reg_GetResetStatus, 9
 Rcm_Reg_SetExtResetFilter, 10
 Rcm_Reg_SetResetDelayTime, 10
 Rcm_Reg_SetResetState, 11
DelayTime
 Rcm_ConfigType, 3
FilterValue
 Rcm_ConfigType, 3
ISR
 Rcm_Hal.c, 12
InterruptSource
 Rcm_ConfigType, 4
RCM_IRQ_CONTROL_INTERNAL
 Rcm_Hal.h, 15
RESET_FILTER_MAX_VALUE
 Rcm_Hal.h, 15
Rcm_ConfigType, 3
 DelayTime, 3
 FilterValue, 3
 InterruptSource, 4
 RcmCallback, 4
 ResetSource, 4
Rcm_DelayType
 Rcm_Hal.h, 16
Rcm_Hal.c, 11
 ISR, 12
 Rcm_Hal_ClearResetStatus, 12
 Rcm_Hal_GetResetStatus, 12
 Rcm_Hal_Init, 13
 Rcm_Hal_SetResetState, 14
Rcm_Hal.h, 14
 RCM_IRQ_CONTROL_INTERNAL, 15
 RESET_FILTER_MAX_VALUE, 15
 Rcm_DelayType, 16
 Rcm_Hal_ClearResetStatus, 17
 Rcm_Hal_GetResetStatus, 17
 Rcm_Hal_Init, 18
 Rcm_Hal_SetResetState, 19
 Rcm_ResetIDType, 16
 Rcm_ResetStateType, 17
Rcm_Hal_ClearResetStatus
 Rcm_Hal.c, 12
 Rcm_Hal.h, 17
Rcm_Hal_GetResetStatus
 Rcm_Hal.c, 12
 Rcm_Hal.h, 17
Rcm_Hal_Init
 Rcm_Hal.c, 13
 Rcm_Hal.h, 18
Rcm_Hal_SetResetState
 Rcm_Hal.c, 14
 Rcm_Hal.h, 19
Rcm_Reg_ClearResetInterruptStatus
 AC784xx_Rcm_Reg.h, 6
Rcm_Reg_ClearResetStatus
 AC784xx_Rcm_Reg.h, 6
Rcm_Reg_EnableExtResetFilter
 AC784xx_Rcm_Reg.h, 6
Rcm_Reg_EnableGlobalResetInterrupt
 AC784xx_Rcm_Reg.h, 7
Rcm_Reg_EnableResetInterrupts
 AC784xx_Rcm_Reg.h, 7
Rcm_Reg_EnableResetSources
 AC784xx_Rcm_Reg.h, 8
Rcm_Reg_GetResetInterruptSources
 AC784xx_Rcm_Reg.h, 8
Rcm_Reg_GetResetInterruptStatus
 AC784xx_Rcm_Reg.h, 8
Rcm_Reg_GetResetSources
 AC784xx_Rcm_Reg.h, 9
Rcm_Reg_GetResetStatus
 AC784xx_Rcm_Reg.h, 9
Rcm_Reg_SetExtResetFilter
 AC784xx_Rcm_Reg.h, 10
Rcm_Reg_SetResetDelayTime
 AC784xx_Rcm_Reg.h, 10
Rcm_Reg_SetResetState
 AC784xx_Rcm_Reg.h, 11
Rcm_ResetIDType
 Rcm_Hal.h, 16
Rcm_ResetStateType
 Rcm_Hal.h, 17
RcmCallback
 Rcm_ConfigType, 4
ResetSource
 Rcm_ConfigType, 4