

AC784xx_DFP FLASH

9.1.0

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	Flash_Config Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	Erase_Timeout	3
3.1.2.2	LVD_Enable	4
3.1.2.3	Write_Timeout	4
3.2	Flash_CseType Struct Reference	4
3.2.1	Detailed Description	4
3.2.2	Member Data Documentation	4
3.2.2.1	KeySize	4
3.2.2.2	Sfe	4

4 File Documentation	5
4.1 AC784xx_API_Reference_Manual_FLASH.pdf File Reference	5
4.2 AC784xx_Fls_Reg.h File Reference	5
4.2.1 Detailed Description	7
4.2.2 Macro Definition Documentation	7
4.2.2.1 FLASH_STAT_ERROR_BITS	7
4.2.3 Function Documentation	7
4.2.3.1 FLASH_ClearFRAMECC1BitFlag()	7
4.2.3.2 FLASH_ClearFRAMECC2BitFlag()	8
4.2.3.3 FLASH_GetFRAMECC1BitFlag()	8
4.2.3.4 FLASH_GetFRAMECC2BitFlag()	8
4.2.3.5 Flash_Reg_AbortPageEraseReg()	9
4.2.3.6 Flash_Reg_ClearCollIRQFlag()	9
4.2.3.7 Flash_Reg_ClearECCErrorAddress()	9
4.2.3.8 Flash_Reg_ClearECCIRQFlag()	10
4.2.3.9 Flash_Reg_ClearStatus()	10
4.2.3.10 Flash_Reg_ECCErrorAddr()	10
4.2.3.11 Flash_Reg_EnableCollisionIRQ()	10
4.2.3.12 Flash_Reg_EnableCompleteIRQ()	11
4.2.3.13 Flash_Reg_EnableECCIRQ()	11
4.2.3.14 Flash_Reg_ForceECCErrorDetect()	12
4.2.3.15 Flash_Reg_GetAbortPageEraseStatusReg()	12
4.2.3.16 Flash_Reg_GetBackdoorKeyStatus()	12
4.2.3.17 Flash_Reg_GetBackdoorStatusReg()	13
4.2.3.18 Flash_Reg_GetCnfg()	13
4.2.3.19 Flash_Reg_GetCollIRQFlag()	13
4.2.3.20 Flash_Reg_GetCommandCompleteStatusReg()	14
4.2.3.21 Flash_Reg_GetCSEPartition()	14
4.2.3.22 Flash_Reg_GetDFlashProtection()	14
4.2.3.23 Flash_Reg_GetDPartition()	15
4.2.3.24 Flash_Reg_GetECCIRQFlag()	15

4.2.3.25	Flash_Reg_GetEPartition()	15
4.2.3.26	Flash_Reg_GetHighDataReg()	16
4.2.3.27	Flash_Reg_GetLockStatusReg()	16
4.2.3.28	Flash_Reg_GetLowDataReg()	16
4.2.3.29	Flash_Reg_GetPFlashProtection()	16
4.2.3.30	Flash_Reg_GetReadProtectStatus()	17
4.2.3.31	Flash_Reg_GetStatusReg()	17
4.2.3.32	Flash_Reg_GetTrigCtrlCmdRegStatus()	18
4.2.3.33	Flash_Reg_Lock()	18
4.2.3.34	Flash_Reg_SetAddressReg()	18
4.2.3.35	Flash_Reg_SetCommandReg()	19
4.2.3.36	Flash_Reg_SetDFlashProtection()	19
4.2.3.37	Flash_Reg_SetEMDFIE()	20
4.2.3.38	Flash_Reg_SetHighDataReg()	20
4.2.3.39	Flash_Reg_SetLengthReg()	21
4.2.3.40	Flash_Reg_SetLowDataReg()	21
4.2.3.41	Flash_Reg_SetLVDDetection()	21
4.2.3.42	Flash_Reg_SetPFlashProtection()	22
4.2.3.43	Flash_Reg_TrigCtrlCmdReg()	22
4.2.3.44	Flash_Reg_UnLock()	23
4.3	Fls_Hal.c File Reference	23
4.3.1	Detailed Description	24
4.3.2	Macro Definition Documentation	25
4.3.2.1	FLASH_INVALID_ADDR	25
4.3.2.2	FLASH_NOT_USE_ADDR	25
4.3.3	Function Documentation	25
4.3.3.1	Flash_Hal_AllErase()	25
4.3.3.2	Flash_Hal_AllVerify()	25
4.3.3.3	Flash_Hal_BackdoorSet()	26
4.3.3.4	Flash_Hal_BackdoorVerify()	26
4.3.3.5	Flash_Hal_Erase()	27

4.3.3.6	Flash_Hal_EraseCseInfo()	27
4.3.3.7	Flash_Hal_ErrorStatusGetAndClear()	27
4.3.3.8	Flash_Hal_InfoErase()	28
4.3.3.9	Flash_Hal_InfoRead()	28
4.3.3.10	Flash_Hal_InfoWrite()	29
4.3.3.11	Flash_Hal_Init()	29
4.3.3.12	Flash_Hal_LockCtrl()	30
4.3.3.13	Flash_Hal_ProgramCseInfo()	30
4.3.3.14	Flash_Hal_ReadProtectSet()	31
4.3.3.15	Flash_Hal_SetPartition()	31
4.3.3.16	Flash_Hal_Swap()	31
4.3.3.17	Flash_Hal_UnLockCtrl()	32
4.3.3.18	Flash_Hal_Verify()	32
4.3.3.19	Flash_Hal_WPInfoErase()	33
4.3.3.20	Flash_Hal_Write()	33
4.3.3.21	Flash_Hal_WriteOtp()	34
4.3.3.22	Flash_Hal_WriteProtectSet()	34
4.4	Fls_Hal.h File Reference	35
4.4.1	Detailed Description	37
4.4.2	Macro Definition Documentation	37
4.4.2.1	DFLASH_ENABLE	37
4.4.2.2	FLASH_ERASE_ALL_BLOCK_UNSECURE	37
4.4.2.3	FLASH_ERASE_BLOCK	38
4.4.2.4	FLASH_ERASE_PAGE	38
4.4.2.5	FLASH_ERASE_PARTITION	38
4.4.2.6	FLASH_OPTION_PAGE_ERASE	38
4.4.2.7	FLASH_PROGRAM_CSE	38
4.4.2.8	FLASH_PROGRAM_ONCE	38
4.4.2.9	FLASH_PROGRAM_PARTITION	39
4.4.2.10	FLASH_PROGRAM_PHRASE	39
4.4.2.11	FLASH_PROGRAM_SECTION	39

4.4.2.12	FLASH_SECURITY_BY_PASS	39
4.4.2.13	FLASH_SET_FLEXRAM	39
4.4.2.14	FLASH_VERIFY_ALL_BLOCK	39
4.4.2.15	FLASH_VERIFY_BLOCK	40
4.4.2.16	FLASH_VERIFY_SECTION	40
4.4.2.17	PFLASH_ENABLE	40
4.4.3	Enumeration Type Documentation	40
4.4.3.1	Flash_CseSize	40
4.4.4	Function Documentation	40
4.4.4.1	DFlash_Hal_BlockErase()	41
4.4.4.2	DFlash_Hal_BlockVerify()	41
4.4.4.3	DFlash_Hal_PageErase()	41
4.4.4.4	DFlash_Hal_PageWrite()	42
4.4.4.5	DFlash_Hal_Read()	42
4.4.4.6	DFlash_Hal_SectionVerify()	43
4.4.4.7	DFlash_Hal_SectionWrite()	43
4.4.4.8	Flash_Hal_AllErase()	44
4.4.4.9	Flash_Hal_AllVerify()	44
4.4.4.10	Flash_Hal_BackdoorSet()	44
4.4.4.11	Flash_Hal_BackdoorVerify()	45
4.4.4.12	Flash_Hal_Erase()	46
4.4.4.13	Flash_Hal_EraseCseInfo()	46
4.4.4.14	Flash_Hal_ErrorStatusGetAndClear()	47
4.4.4.15	Flash_Hal_InfoErase()	48
4.4.4.16	Flash_Hal_InfoRead()	48
4.4.4.17	Flash_Hal_InfoWrite()	49
4.4.4.18	Flash_Hal_Init()	50
4.4.4.19	Flash_Hal_LockCtrl()	50
4.4.4.20	Flash_Hal_ProgramCseInfo()	51
4.4.4.21	Flash_Hal_ReadProtectSet()	52
4.4.4.22	Flash_Hal_SetPartition()	52
4.4.4.23	Flash_Hal_Swap()	53
4.4.4.24	Flash_Hal_UnLockCtrl()	54
4.4.4.25	Flash_Hal_Verify()	54
4.4.4.26	Flash_Hal_WPInfoErase()	55
4.4.4.27	Flash_Hal_Write()	56
4.4.4.28	Flash_Hal_WriteOtp()	56
4.4.4.29	Flash_Hal_WriteProtectSet()	57
4.4.4.30	PFlash_Hal_BlockErase()	58
4.4.4.31	PFlash_Hal_BlockVerify()	58
4.4.4.32	PFlash_Hal_PageErase()	59
4.4.4.33	PFlash_Hal_PageWrite()	59
4.4.4.34	PFlash_Hal_Read()	59
4.4.4.35	PFlash_Hal_SectionVerify()	60
4.4.4.36	PFlash_Hal_SectionWrite()	60

[Index](#)

62

Chapter 1

Class Index

1.1 Class List

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

Flash_Config	Flash init configuration information	3
Flash_CseType	Flash cse information about size and sfe	4

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_FLASH.pdf	5
AC784xx_Fls_Reg.h	
This file provides extern Reg Flash api	5
Fls_Hal.c	
This file provides sdk flash api function	23
Fls_Hal.h	
This file provides sdk flash api function	35

Chapter 3

Class Documentation

3.1 Flash_Config Struct Reference

flash init configuration information

```
#include <Fls_Hal.h>
```

Public Attributes

- uint32 [Write_Timeout](#)
- uint32 [Erase_Timeout](#)
- boolean [LVD_Enable](#)

3.1.1 Detailed Description

flash init configuration information

Definition at line 96 of file Fls_Hal.h.

3.1.2 Member Data Documentation

3.1.2.1 Erase_Timeout

```
uint32 Flash_Config::Erase_Timeout
```

flash erase working the longest time

Definition at line 99 of file Fls_Hal.h.

3.1.2.2 LVD_Enable

```
boolean Flash_Config::LVD_Enable
```

flash lvd detect enable/disable

Definition at line 100 of file Fls_Hal.h.

3.1.2.3 Write_Timeout

```
uint32 Flash_Config::Write_Timeout
```

flash write working the longest time

Definition at line 98 of file Fls_Hal.h.

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

- [Fls_Hal.h](#)

3.2 Flash_CseType Struct Reference

flash cse information about size and sfe

```
#include <Fls_Hal.h>
```

Public Attributes

- [Flash_CseSize](#) [KeySize](#)
- boolean [Sfe](#)

3.2.1 Detailed Description

flash cse information about size and sfe

Definition at line 65 of file Fls_Hal.h.

3.2.2 Member Data Documentation

3.2.2.1 KeySize

```
Flash\_CseSize Flash_CseType::KeySize
```

Definition at line 67 of file Fls_Hal.h.

3.2.2.2 Sfe

```
boolean Flash_CseType::Sfe
```

Definition at line 68 of file Fls_Hal.h.

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

- [Fls_Hal.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_FLASH.pdf File Reference

4.2 AC784xx_Fls_Reg.h File Reference

This file provides extern Reg Flash api.

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

Macros

- #define [FLASH_STAT_ERROR_BITS](#)

Functions

- LOCAL_INLINE void [Flash_Reg_Lock](#) (void)
Set eflash lock.
- LOCAL_INLINE void [Flash_Reg_UnLock](#) (void)
Set eflash unlock.
- LOCAL_INLINE void [Flash_Reg_ClearStatus](#) (void)
- LOCAL_INLINE uint8 [Flash_Reg_GetLockStatusReg](#) (void)
Get the can eflash controler lock status.
- LOCAL_INLINE void [Flash_Reg_SetPFlashProtection](#) (uint8 ProtectPos, uint32 ProtectStatus)
Set P-Flash write protection status.
- LOCAL_INLINE void [Flash_Reg_SetDFlashProtection](#) (uint16 ProtectStatus)
Set D-Flash write protection status.
- LOCAL_INLINE void [Flash_Reg_EnableCompleteIRQ](#) (boolean enable)
Set command complete interrupt register.
- LOCAL_INLINE void [Flash_Reg_EnableCollisionIRQ](#) (boolean enable)
Set collision error interrupt register.
- LOCAL_INLINE uint8 [Flash_Reg_GetECCIRQFlag](#) (void)
Get Flash ECC 2-bit error interrupt flag.
- LOCAL_INLINE void [Flash_Reg_ClearECCIRQFlag](#) (void)
Clear Flash ECC 2-bit error interrupt flag.
- LOCAL_INLINE void [Flash_Reg_SetAddressReg](#) (uint32 Addr)

- Set program or erase start address.*

 - LOCAL_INLINE void [Flash_Reg_SetLengthReg](#) (uint32 Len)
- Set program or erase length.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetAbortPageEraseStatusReg](#) (void)
- Get abort page erase status.*

 - LOCAL_INLINE void [Flash_Reg_AbortPageEraseReg](#) (void)
- Flash abort page erase.*

 - LOCAL_INLINE uint32 [Flash_Reg_GetStatusReg](#) (void)
- Get the error status.*

 - LOCAL_INLINE void [Flash_Reg_SetCommandReg](#) (uint32 Cmd)
- Set command.*

 - LOCAL_INLINE void [Flash_Reg_TrigCtrlCmdReg](#) (void)
- Trigger the command to start.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetTrigCtrlCmdRegStatus](#) (void)
- Get Trigger the FLASH operation command excute status.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetCommandCompleteStatusReg](#) (void)
- Get command completed status.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetBackdoorStatusReg](#) (void)
- Get backdoor verify status.*

 - LOCAL_INLINE void [Flash_Reg_EnableECCIRQ](#) (boolean enable)
- Set ECC 2-bit error interrupt register.*

 - LOCAL_INLINE uint32 [Flash_Reg_GetPFlashProtection](#) (uint8 ProtectPos)
- Get P-Flash write protection status.*

 - LOCAL_INLINE uint32 [Flash_Reg_GetDFlashProtection](#) (void)
- Get the error status.*

 - LOCAL_INLINE void [Flash_Reg_SetLowDataReg](#) (uint32 Data)
- Set low 32-bit data.*

 - LOCAL_INLINE uint32 [Flash_Reg_GetLowDataReg](#) (void)
- get low 32-bit data register value.*

 - LOCAL_INLINE uint32 [Flash_Reg_GetHighDataReg](#) (void)
- Get high 32-bit data register value.*

 - LOCAL_INLINE void [Flash_Reg_SetHighDataReg](#) (uint32 Data)
- Set high 32-bit data.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetDPartition](#) (void)
- Get DFLASH Partition.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetEPartition](#) (void)
- Get FlexRAM Partition.*

 - LOCAL_INLINE uint8 [Flash_Reg_GetCSEPartition](#) (void)
- Get CSE Partition.*

 - LOCAL_INLINE uint8 [FLASH_GetFRAMECC1BitFlag](#) (void)
- Get FlexRAM ECC 1-bit error flag.*

 - LOCAL_INLINE void [FLASH_ClearFRAMECC1BitFlag](#) (void)
- Clear FlexRAM ECC 1-bit error flag.*

 - LOCAL_INLINE uint8 [FLASH_GetFRAMECC2BitFlag](#) (void)
- Get FlexRAM ECC 2-bit error flag.*

 - LOCAL_INLINE void [FLASH_ClearFRAMECC2BitFlag](#) (void)
- Clear FlexRAM ECC 2-bit error flag.*

 - LOCAL_INLINE void [Flash_Reg_SetLVDDetection](#) (boolean IsEnable)
- Set LVD Detection.*

 - LOCAL_INLINE uint32 [Flash_Reg_ECCErrorAddr](#) (void)
- Get Ecc Error Address Info.*

 - LOCAL_INLINE void [Flash_Reg_SetEMDFIE](#) (boolean IsEnable)
- Set LVD Detection.*

- LOCAL_INLINE uint8 [Flash_Reg_GetReadProtectStatus](#) (void)
Get the flash read protection status.
- LOCAL_INLINE uint8 [Flash_Reg_GetBackdoorKeyStatus](#) (void)
Get the backdoor key verify enable status.
- LOCAL_INLINE void [Flash_Reg_ClearECCErrorAddress](#) (void)
Clear Flash ECC 2-bit error address.
- LOCAL_INLINE uint8 [Flash_Reg_GetCollIRQFlag](#) (void)
Get collision error interrupt flag.
- LOCAL_INLINE void [Flash_Reg_ClearCollIRQFlag](#) (void)
Clear collision error interrupt flag.
- LOCAL_INLINE void [Flash_Reg_ForceECCErrorDetect](#) (void)
force ECC 2-bit error detected, code should run in SRAM, usually used it for test.
- LOCAL_INLINE uint32 [Flash_Reg_GetCnfg](#) (void)
Get flash CNFG register value.

4.2.1 Detailed Description

This file provides extern Reg Flash api.

4.2.2 Macro Definition Documentation

4.2.2.1 FLASH_STAT_ERROR_BITS

```
#define FLASH_STAT_ERROR_BITS
```

Value:

```
(FLASH_STAT_DFDIF_Msk | \
    FLASH_STAT_PVIOLE_Msk | \
    FLASH_STAT_VERIFYERR_Msk | \
    FLASH_STAT_ACCERRF_Msk | \
    FLASH_STAT_COLLERRF_Msk)
```

Definition at line 57 of file AC784xx_Fls_Reg.h.

4.2.3 Function Documentation

4.2.3.1 FLASH_ClearFRAMECC1BitFlag()

```
LOCAL_INLINE void FLASH_ClearFRAMECC1BitFlag (
    void )
```

Clear FlexRAM ECC 1-bit error flag.

Returns

none

Definition at line 478 of file AC784xx_Fls_Reg.h.

4.2.3.2 FLASH_ClearFRAMECC2BitFlag()

```
LOCAL_INLINE void FLASH_ClearFRAMECC2BitFlag (  
    void )
```

Clear FlexRAM ECC 2-bit error flag.

Returns

none

Definition at line 498 of file AC784xx_Fls_Reg.h.

4.2.3.3 FLASH_GetFRAMECC1BitFlag()

```
LOCAL_INLINE uint8 FLASH_GetFRAMECC1BitFlag (  
    void )
```

Get FlexRAM ECC 1-bit error flag.

Returns

Flash FlexRAM ECC 1-bit error interrupt flag

Definition at line 468 of file AC784xx_Fls_Reg.h.

4.2.3.4 FLASH_GetFRAMECC2BitFlag()

```
LOCAL_INLINE uint8 FLASH_GetFRAMECC2BitFlag (  
    void )
```

Get FlexRAM ECC 2-bit error flag.

Returns

Flash FlexRAM ECC 2-bit error interrupt flag

Definition at line 488 of file AC784xx_Fls_Reg.h.

4.2.3.5 Flash_Reg_AbortPageEraseReg()

```
LOCAL_INLINE void Flash_Reg_AbortPageEraseReg (  
    void )
```

Flash abort page erase.

Note

Function ID:[DES_FLS_API_058]

Returns

None

Definition at line 236 of file AC784xx_Fls_Reg.h.

4.2.3.6 Flash_Reg_ClearCollIRQFlag()

```
LOCAL_INLINE void Flash_Reg_ClearCollIRQFlag (  
    void )
```

Clear collision error interrupt flag.

Returns

none

Definition at line 600 of file AC784xx_Fls_Reg.h.

4.2.3.7 Flash_Reg_ClearECCErrAddress()

```
LOCAL_INLINE void Flash_Reg_ClearECCErrAddress (  
    void )
```

Clear Flash ECC 2-bit error address.

Returns

none

Definition at line 580 of file AC784xx_Fls_Reg.h.

4.2.3.8 Flash_Reg_ClearECCIRQFlag()

```
LOCAL_INLINE void Flash_Reg_ClearECCIRQFlag (  
    void )
```

Clear Flash ECC 2-bit error interrupt flag.

Returns

none

Definition at line 189 of file AC784xx_Fls_Reg.h.

4.2.3.9 Flash_Reg_ClearStatus()

```
LOCAL_INLINE void Flash_Reg_ClearStatus (  
    void )
```

Definition at line 93 of file AC784xx_Fls_Reg.h.

4.2.3.10 Flash_Reg_ECCErrorAddr()

```
LOCAL_INLINE uint32 Flash_Reg_ECCErrorAddr (  
    void )
```

Get Ecc Error Address Info.

Note

Function ID:[DES_FLS_API_068]

Returns

None

Definition at line 522 of file AC784xx_Fls_Reg.h.

4.2.3.11 Flash_Reg_EnableCollisionIRQ()

```
LOCAL_INLINE void Flash_Reg_EnableCollisionIRQ (  
    boolean enable )
```

Set collision error interrupt register.

Parameters

in	<i>enable</i>	collision interrupt enable status
----	---------------	-----------------------------------

Returns

none

Definition at line 169 of file AC784xx_Fls_Reg.h.

4.2.3.12 Flash_Reg_EnableCompleteIRQ()

```
LOCAL_INLINE void Flash_Reg_EnableCompleteIRQ (  
    boolean enable )
```

Set command complete interrupt register.

Parameters

in	<i>enable</i>	command complete interrupt enable status
----	---------------	--

Returns

none

Definition at line 158 of file AC784xx_Fls_Reg.h.

4.2.3.13 Flash_Reg_EnableECCIRQ()

```
LOCAL_INLINE void Flash_Reg_EnableECCIRQ (  
    boolean enable )
```

Set ECC 2-bit error interrupt register.

Parameters

in	<i>enable</i>	ECC 2-bit error interrupt enable status
----	---------------	---

Returns

none

Definition at line 312 of file AC784xx_Fls_Reg.h.

4.2.3.14 Flash_Reg_ForceECCErrorDetect()

```
LOCAL_INLINE void Flash_Reg_ForceECCErrorDetect (
    void )
```

force ECC 2-bit error detected, code should run in SRAM, usually used it for test.

Returns

none

Definition at line 610 of file AC784xx_Fls_Reg.h.

4.2.3.15 Flash_Reg_GetAbortPageEraseStatusReg()

```
LOCAL_INLINE uint8 Flash_Reg_GetAbortPageEraseStatusReg (
    void )
```

Get abort page erase status.

Note

Function ID:[DES_FLS_API_057]

Returns

Abort page erase status

Definition at line 225 of file AC784xx_Fls_Reg.h.

4.2.3.16 Flash_Reg_GetBackdoorKeyStatus()

```
LOCAL_INLINE uint8 Flash_Reg_GetBackdoorKeyStatus (
    void )
```

Get the backdoor key verify enable status.

Returns

2b'11: Backdoor key can not disable read protection 2b'00/2b'01/2b'10: Backdoor key can enable read protection

Definition at line 570 of file AC784xx_Fls_Reg.h.

4.2.3.17 Flash_Reg_GetBackdoorStatusReg()

```
LOCAL_INLINE uint8 Flash_Reg_GetBackdoorStatusReg (  
    void )
```

Get backdoor verify status.

Note

Function ID:[DES_FLS_API_063]

Returns

Backdoor verify status

Definition at line 300 of file AC784xx_Fls_Reg.h.

4.2.3.18 Flash_Reg_GetCnfg()

```
LOCAL_INLINE uint32 Flash_Reg_GetCnfg (  
    void )
```

Get flash CNFG register value.

Returns

CNFG register value

Definition at line 620 of file AC784xx_Fls_Reg.h.

4.2.3.19 Flash_Reg_GetCollIRQFlag()

```
LOCAL_INLINE uint8 Flash_Reg_GetCollIRQFlag (  
    void )
```

Get collision error interrupt flag.

Returns

Collision error interrupt flag

Definition at line 590 of file AC784xx_Fls_Reg.h.

4.2.3.20 Flash_Reg_GetCommandCompleteStatusReg()

```
LOCAL_INLINE uint8 Flash_Reg_GetCommandCompleteStatusReg (  
    void )
```

Get command completed status.

Note

Function ID:[DES_FLS_API_063]

Returns

Command complete status

Definition at line 289 of file AC784xx_Fls_Reg.h.

4.2.3.21 Flash_Reg_GetCSEPartition()

```
LOCAL_INLINE uint8 Flash_Reg_GetCSEPartition (  
    void )
```

Get CSE Partition.

Note

Function ID:[DES_FLS_API_067]

Returns

CSE PARTITION SIZE

Definition at line 457 of file AC784xx_Fls_Reg.h.

4.2.3.22 Flash_Reg_GetDFlashProtection()

```
LOCAL_INLINE uint32 Flash_Reg_GetDFlashProtection (  
    void )
```

Get the error status.

Note

Function ID:[DES_FLS_API_065]

Returns

DFlash protection

Definition at line 347 of file AC784xx_Fls_Reg.h.

4.2.3.23 Flash_Reg_GetDPartition()

```
LOCAL_INLINE uint8 Flash_Reg_GetDPartition (
    void )
```

Get DFLASH Partition.

Note

Function ID:[DES_FLS_API_067]

Returns

PFLASH PARTITION SIZE

Definition at line 435 of file AC784xx_Fls_Reg.h.

4.2.3.24 Flash_Reg_GetECCIRQFlag()

```
LOCAL_INLINE uint8 Flash_Reg_GetECCIRQFlag (
    void )
```

Get Flash ECC 2-bit error interrupt flag.

Returns

Flash ECC 2-bit error interrupt flag

Definition at line 179 of file AC784xx_Fls_Reg.h.

4.2.3.25 Flash_Reg_GetEPartition()

```
LOCAL_INLINE uint8 Flash_Reg_GetEPartition (
    void )
```

Get FlexRAM Partition.

Note

Function ID:[DES_FLS_API_067]

Returns

DFLASH PARTITION SIZE

Definition at line 446 of file AC784xx_Fls_Reg.h.

4.2.3.26 Flash_Reg_GetHighDataReg()

```
LOCAL_INLINE uint32 Flash_Reg_GetHighDataReg (
    void )
```

Get high 32-bit data register value.

Returns

High 32-bit data register value

Definition at line 378 of file AC784xx_Fls_Reg.h.

4.2.3.27 Flash_Reg_GetLockStatusReg()

```
LOCAL_INLINE uint8 Flash_Reg_GetLockStatusReg (
    void )
```

Get the can eflash controler lock status.

Note

Function ID:[DES_FLS_API_050]

Returns

Lock Status.

Definition at line 103 of file AC784xx_Fls_Reg.h.

4.2.3.28 Flash_Reg_GetLowDataReg()

```
LOCAL_INLINE uint32 Flash_Reg_GetLowDataReg (
    void )
```

get low 32-bit data register value.

Returns

Low 32-bit data register value

Definition at line 368 of file AC784xx_Fls_Reg.h.

4.2.3.29 Flash_Reg_GetPFlashProtection()

```
LOCAL_INLINE uint32 Flash_Reg_GetPFlashProtection (
    uint8 ProtectPos )
```

Get P-Flash write protection status.

Note

Function ID:[DES_FLS_API_064]

Parameters

in	<i>ProtectPos</i>	Get Pflash Protection status
----	-------------------	------------------------------

Returns

PFlash protection

Definition at line 323 of file AC784xx_Fls_Reg.h.

4.2.3.30 Flash_Reg_GetReadProtectStatus()

```
LOCAL_INLINE uint8 Flash_Reg_GetReadProtectStatus (  
    void )
```

Get the flash read protection status.

Returns

2b'11: Read protection is disabled 2b'00/2b'01/2b'10: Read protection is enabled

Definition at line 558 of file AC784xx_Fls_Reg.h.

4.2.3.31 Flash_Reg_GetStatusReg()

```
LOCAL_INLINE uint32 Flash_Reg_GetStatusReg (  
    void )
```

Get the error status.

Note

Function ID:[DES_FLS_API_059]

Returns

Status

Definition at line 246 of file AC784xx_Fls_Reg.h.

4.2.3.32 Flash_Reg_GetTrigCtrlCmdRegStatus()

```
LOCAL_INLINE uint8 Flash_Reg_GetTrigCtrlCmdRegStatus (
    void )
```

Get Trigger the FLASH operation command excute status.

Note

Function ID:[DES_FLS_API_062]

Returns

Trig control cmd reg status

Definition at line 277 of file AC784xx_Fls_Reg.h.

4.2.3.33 Flash_Reg_Lock()

```
LOCAL_INLINE void Flash_Reg_Lock (
    void )
```

Set eflash lock.

Note

Function ID:[DES_FLS_API_049]

Returns

None

Definition at line 76 of file AC784xx_Fls_Reg.h.

4.2.3.34 Flash_Reg_SetAddressReg()

```
LOCAL_INLINE void Flash_Reg_SetAddressReg (
    uint32 Addr )
```

Set program or erase start address.

Note

Function ID:[DES_FLS_API_055]

Parameters

in	<i>Addr</i>	address value
----	-------------	---------------

Returns

None

Definition at line 200 of file AC784xx_Fls_Reg.h.

4.2.3.35 Flash_Reg_SetCommandReg()

```
LOCAL_INLINE void Flash_Reg_SetCommandReg (
    uint32 Cmd )
```

Set command.

Note

Function ID:[DES_FLS_API_060]

Parameters

in	<i>Cmd</i>	Command value
----	------------	---------------

Returns

None

Definition at line 257 of file AC784xx_Fls_Reg.h.

4.2.3.36 Flash_Reg_SetDFlashProtection()

```
LOCAL_INLINE void Flash_Reg_SetDFlashProtection (
    uint16 ProtectStatus )
```

Set D-Flash write protection status.

Note

Function ID:[DES_FLS_API_054]

Parameters

in	<i>ProtectStatus</i>	protect status
----	----------------------	----------------

Returns

None

Definition at line 147 of file AC784xx_Fls_Reg.h.

4.2.3.37 Flash_Reg_SetEMDFIE()

```
LOCAL_INLINE void Flash_Reg_SetEMDFIE (
    boolean IsEnable )
```

Set LVD Detection.

Note

Function ID:[DES_FLS_API_068]

Parameters

<i>in</i>	<i>IsEnable</i>	True enable LVD Detection, false disable rtc module
-----------	-----------------	---

Returns

None

Definition at line 533 of file AC784xx_Fls_Reg.h.

4.2.3.38 Flash_Reg_SetHighDataReg()

```
LOCAL_INLINE void Flash_Reg_SetHighDataReg (
    uint32 Data )
```

Set high 32-bit data.

Note

Function ID:[DES_FLS_API_067]

Parameters

<i>in</i>	<i>Data</i>	Data write to regsiter
-----------	-------------	------------------------

Returns

None

Definition at line 389 of file AC784xx_Fls_Reg.h.

4.2.3.39 Flash_Reg_SetLengthReg()

```
LOCAL_INLINE void Flash_Reg_SetLengthReg (
    uint32 Len )
```

Set program or erase length.

Note

Function ID:[DES_FLS_API_056]

Parameters

in	<i>Len</i>	len value
----	------------	-----------

Returns

None

Definition at line 211 of file AC784xx_Fls_Reg.h.

4.2.3.40 Flash_Reg_SetLowDataReg()

```
LOCAL_INLINE void Flash_Reg_SetLowDataReg (
    uint32 Data )
```

Set low 32-bit data.

Note

Function ID:[DES_FLS_API_066]

Parameters

in	<i>Data</i>	Data write to regisiter
----	-------------	-------------------------

Returns

None

Definition at line 358 of file AC784xx_Fls_Reg.h.

4.2.3.41 Flash_Reg_SetLVDDetection()

```
LOCAL_INLINE void Flash_Reg_SetLVDDetection (
    boolean IsEnable )
```

Set LVD Detection.

Note

Function ID:[DES_FLS_API_068]

Parameters

in	<i>IsEnable</i>	True enable LVD Detection, false disable rtc module
----	-----------------	---

Returns

None

Definition at line 510 of file AC784xx_Fls_Reg.h.

4.2.3.42 Flash_Reg_SetPFlashProtection()

```
LOCAL_INLINE void Flash_Reg_SetPFlashProtection (
    uint8 ProtectPos,
    uint32 ProtectStatus )
```

Set P-Flash write protection status.

Note

Function ID:[DES_FLS_API_053]

Parameters

in	<i>ProtectPos</i>	protect bits Postion
in	<i>ProtectStatus</i>	protect status

Returns

None

Definition at line 116 of file AC784xx_Fls_Reg.h.

4.2.3.43 Flash_Reg_TrigCtrlCmdReg()

```
LOCAL_INLINE void Flash_Reg_TrigCtrlCmdReg (
    void )
```

Trigger the command to start.

Note

Function ID:[DES_FLS_API_061]

Returns

None

Definition at line 267 of file AC784xx_Fls_Reg.h.

4.2.3.44 Flash_Reg_UnLock()

```
LOCAL_INLINE void Flash_Reg_UnLock (  
    void )
```

Set eflash unlock.

Note

Function ID:[DES_FLS_API_049]

Returns

None

Definition at line 87 of file AC784xx_Fls_Reg.h.

4.3 Fls_Hal.c File Reference

This file provides sdk flash api function.

```
#include "Fls_Hal.h"  
#include "AC784xx_Fls_Reg.h"  
#include "OsIf_Critical.h"  
#include "OsIf_Irq.h"  
#include "Spm_Hal.h"  
#include "Core_Hal.h"  
#include "System_AC784xx.h"
```

Macros

- `#define FLASH_INVALID_ADDR (0xFFFFFFFFU)`
- `#define FLASH_NOT_USE_ADDR (0x40000U)`

Functions

- void [Flash_Hal_LockCtrl](#) (void)
lock flash register write
- Hal_StatusType [Flash_Hal_UnLockCtrl](#) (void)
unlock flash register write
- Hal_StatusType [Flash_Hal_SetPartition](#) (const [Flash_CseType](#) *CseConfig)
This API set PARTITION SIZE.
- void [Flash_Hal_Init](#) (const [Flash_Config](#) *Config)
Initialize flash hardware.
- Hal_StatusType [Flash_Hal_Erase](#) (uint8 CommandType, uint32 Address)
erase flash area
- Hal_StatusType [Flash_Hal_WPInfoErase](#) (void)
erase write protect information flash area
- Hal_StatusType [Flash_Hal_AllErase](#) (void)
erase all flash area
- Hal_StatusType [Flash_Hal_Write](#) (uint8 CommandType, uint32 Address, uint32 Length, const uint8 *DataPtr)
Write one or more unit complete flash pages into given flash sector.
- Hal_StatusType [Flash_Hal_WriteOtp](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write flash OTP area
- uint32 [Flash_Hal_ErrorStatusGetAndClear](#) (uint32 *Addr)
get flash error status and clear
- Hal_StatusType [Flash_Hal_Verify](#) (uint8 CommandType, uint32 Address, uint32 Length)
verify input address
- Hal_StatusType [Flash_Hal_AllVerify](#) (void)
verify all flash area
- Hal_StatusType [Flash_Hal_ReadProtectSet](#) (void)
enable read protect function
- Hal_StatusType [Flash_Hal_WriteProtectSet](#) (boolean En, const uint8 *Data)
disable/enable write protect
- Hal_StatusType [Flash_Hal_BackdoorSet](#) (const uint8 *Data)
set backdoor key
- Hal_StatusType [Flash_Hal_BackdoorVerify](#) (const uint8 *Data)
verify backdoor key
- Hal_StatusType [Flash_Hal_EraseCseInfo](#) (void)
erase cse info area
- Hal_StatusType [Flash_Hal_ProgramCseInfo](#) (uint32 Addr, uint32 Length, const uint8 *Data)
program cse info
- Hal_StatusType [Flash_Hal_Swap](#) (boolean En)
swap flash program bank
- void [Flash_Hal_InfoRead](#) (uint32 Offset, uint32 Length, uint8 *DataPtr)
Read one or more bytes from given flash info.
- Hal_StatusType [Flash_Hal_InfoWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
Write one or more unit from given flash info.
- Hal_StatusType [Flash_Hal_InfoErase](#) (uint32 Offset)
Erase one page from given flash info.

4.3.1 Detailed Description

This file provides sdk flash api function.

4.3.2 Macro Definition Documentation

4.3.2.1 FLASH_INVALID_ADDR

```
#define FLASH_INVALID_ADDR (0xFFFFFFFFU)
```

Definition at line 53 of file Fls_Hal.c.

4.3.2.2 FLASH_NOT_USE_ADDR

```
#define FLASH_NOT_USE_ADDR (0x40000U)
```

Definition at line 54 of file Fls_Hal.c.

4.3.3 Function Documentation

4.3.3.1 Flash_Hal_AllErase()

```
Hal_StatusType Flash_Hal_AllErase (  
    void )
```

erase all flash area

Note

Function ID:[DES_FLS_API_213]

Returns

Hal_StatusType

Definition at line 464 of file Fls_Hal.c.

4.3.3.2 Flash_Hal_AllVerify()

```
Hal_StatusType Flash_Hal_AllVerify (  
    void )
```

verify all flash area

Note

Function ID:[DES_FLS_API_216]

Returns

Hal_StatusType

Definition at line 662 of file Fls_Hal.c.

4.3.3.3 Flash_Hal_BackdoorSet()

```
Hal_StatusType Flash_Hal_BackdoorSet (  
    const uint8 * Data )
```

set backdoor key

Note

Function ID:[DES_FLS_API_233]

Parameters

in	Data	backdoor key
----	------	--------------

Returns

Hal_StatusType

Definition at line 743 of file Fls_Hal.c.

4.3.3.4 Flash_Hal_BackdoorVerify()

```
Hal_StatusType Flash_Hal_BackdoorVerify (  
    const uint8 * Data )
```

verify backdoor key

verify backdoor

Note

Function ID:[DES_FLS_API_234]

Parameters

in	Data	backdoor key
----	------	--------------

Returns

Hal_StatusType

Definition at line 769 of file Fls_Hal.c.

4.3.3.5 Flash_Hal_Erase()

```
Hal_StatusType Flash_Hal_Erase (
    uint8 CommandType,
    uint32 Address )
```

erase flash area

Note

Function ID: [DES_FLS_API_202]

Parameters

in	<i>CommandType</i>	Flash command type
in	<i>Address</i>	Flash erase address

Returns

Hal_StatusType

Definition at line 412 of file Fls_Hal.c.

4.3.3.6 Flash_Hal_EraseCseInfo()

```
Hal_StatusType Flash_Hal_EraseCseInfo (
    void )
```

erase cse info area

Flash erase the UID,secret key and lock areas.

Note

Function ID:[DES_FLS_API_235]

Returns

Hal_StatusType

Definition at line 794 of file Fls_Hal.c.

4.3.3.7 Flash_Hal_ErrorStatusGetAndClear()

```
uint32 Flash_Hal_ErrorStatusGetAndClear (
    uint32 * Addr )
```

get flash error status and clear

get error status and clear

Note

Function ID:[DES_FLS_API_223]

Parameters

out	<i>Addr</i>	ECC error address
-----	-------------	-------------------

Returns

error status

Definition at line 600 of file Fls_Hal.c.

4.3.3.8 Flash_Hal_InfoErase()

```
Hal_StatusType Flash_Hal_InfoErase (
    uint32 Offset )
```

Erase one page from given flash info.

Note

Function ID:[DES_FLS_API_204]

Parameters

in	<i>Offset</i>	info address offset
----	---------------	---------------------

Returns

Hal_StatusType

Definition at line 991 of file Fls_Hal.c.

4.3.3.9 Flash_Hal_InfoRead()

```
void Flash_Hal_InfoRead (
    uint32 Offset,
    uint32 Length,
    uint8 * DataPtr )
```

Read one or more bytes from given flash info.

Note

Function ID:[DES_FLS_API_210]

Parameters

in	<i>Offset</i>	info address offset
in	<i>Length</i>	Flash read length
in	<i>DataPtr</i>	source data buffer address

Returns

None

Definition at line 927 of file Fls_Hal.c.

4.3.3.10 Flash_Hal_InfoWrite()

```
Hal_StatusType Flash_Hal_InfoWrite (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

Write one or more unit from given flash info.

Write one or more bytes from given flash info.

Note

Function ID:[DES_FLS_API_211]

Parameters

in	<i>Offset</i>	info address offset
in	<i>Length</i>	Flash read length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

Definition at line 962 of file Fls_Hal.c.

4.3.3.11 Flash_Hal_Init()

```
void Flash_Hal_Init (
    const Flash_Config * Config )
```

Initialize flash hardware.

Note

Function ID: [DES_FLS_API_201]

Parameters

in	<i>Config</i>	flash configure information
----	---------------	-----------------------------

Returns

void

Definition at line 391 of file Fls_Hal.c.

4.3.3.12 Flash_Hal_LockCtrl()

```
void Flash_Hal_LockCtrl (
    void )
```

lock flash register write

Note

Function ID:[DES_FLS_API_205]

Returns

None

Definition at line 303 of file Fls_Hal.c.

4.3.3.13 Flash_Hal_ProgramCseInfo()

```
Hal_StatusType Flash_Hal_ProgramCseInfo (
    uint32 Addr,
    uint32 Length,
    const uint8 * Data )
```

program cse info

FLASH program uid,secret key,lock.

Note

Function ID:[DES_FLS_API_236]

Parameters

in	<i>Addr</i>	info address
in	<i>Length</i>	data length
in	<i>Data</i>	cse info

Returns

Hal_StatusType

Definition at line 814 of file Fls_Hal.c.

4.3.3.14 Flash_Hal_ReadProtectSet()

```
Hal_StatusType Flash_Hal_ReadProtectSet (  
    void )
```

enable read protect function

Note

Function ID:[DES_FLS_API_217]

Returns

Hal_StatusType

Definition at line 691 of file Fls_Hal.c.

4.3.3.15 Flash_Hal_SetPartition()

```
Hal_StatusType Flash_Hal_SetPartition (  
    const Flash_CseType * CseConfig )
```

This API set PARTITION SIZE.

Note

Function ID:[DES_FLS_API_206]

Parameters

in	<i>CseConfig</i>	Cse configuration info about sfe and cse key size
----	------------------	---

Returns

Hal_StatusType

Definition at line 343 of file Fls_Hal.c.

4.3.3.16 Flash_Hal_Swap()

```
Hal_StatusType Flash_Hal_Swap (  
    boolean En )
```

swap flash program bank

FLASH program swap.

Note

Function ID:[DES_FLS_API_248]

Parameters

in	En	0 -> swapA 1 -> swapB
----	----	-----------------------

Returns

Hal_StatusType

Definition at line 910 of file Fls_Hal.c.

4.3.3.17 Flash_Hal_UnLockCtrl()

```
Hal_StatusType Flash_Hal_UnLockCtrl (
    void )
```

unlock flash register write

Note

Function ID:[DES_FLS_API_209]

Returns

Hal_StatusType

Definition at line 313 of file Fls_Hal.c.

4.3.3.18 Flash_Hal_Verify()

```
Hal_StatusType Flash_Hal_Verify (
    uint8 CommandType,
    uint32 Address,
    uint32 Length )
```

verify input address

Note

Function ID:[DES_FLS_API_215]

Parameters

in	CommandType	flash command type
in	Address	Write Address
in	Length	Data length

Returns

Hal_StatusType

Definition at line 628 of file Fls_Hal.c.

4.3.3.19 Flash_Hal_WPInfoErase()

```
Hal_StatusType Flash_Hal_WPInfoErase (
    void )
```

erase write protect infomation flash area

Note

Function ID:[DES_FLS_API_212]

Returns

Hal_StatusType

Definition at line 448 of file Fls_Hal.c.

4.3.3.20 Flash_Hal_Write()

```
Hal_StatusType Flash_Hal_Write (
    uint8 CommandType,
    uint32 Address,
    uint32 Length,
    const uint8 * DataPtr )
```

Write one or more unit complete flash pages into given flash sector.

Write one or more complete flash pages into given flash sector.

Note

Function ID:[DES_FLS_API_203]

Parameters

in	<i>CommandType</i>	flash command type
in	<i>Address</i>	Write Address
in	<i>Length</i>	Data length
in	<i>DataPtr</i>	Data pointer

Returns

Hal_StatusType

Definition at line 497 of file Fls_Hal.c.

4.3.3.21 Flash_Hal_WriteOtp()

```
Hal_StatusType Flash_Hal_WriteOtp (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

write flash OTP area

write flash area with OTP

Note

Function ID:[DES_FLS_API_222]

Parameters

in	<i>Offset</i>	offset in one physical page
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

Definition at line 575 of file Fls_Hal.c.

4.3.3.22 Flash_Hal_WriteProtectSet()

```
Hal_StatusType Flash_Hal_WriteProtectSet (
    boolean En,
    const uint8 * Data )
```

disable/enable write protect

disable/enable write protect information (need reboot work)

Note

Function ID:[DES_FLS_API_238]

Parameters

in	<i>En</i>	disable/enable write protect configuration
in	<i>Data</i>	pflash/dflash write protect config value

Returns

Hal_StatusType

Definition at line 714 of file Fls_Hal.c.

4.4 Fls_Hal.h File Reference

This file provides sdk flash api function.

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

Classes

- struct [Flash_CseType](#)
flash cse information about size and sfe
- struct [Flash_Config](#)
flash init configuration information

Macros

- #define [DFLASH_ENABLE](#) (STD_ON)
- #define [PFLASH_ENABLE](#) (STD_ON)
- #define [FLASH_ERASE_PAGE](#) (0x01U)
- #define [FLASH_ERASE_BLOCK](#) (0x02U)
- #define [FLASH_OPTION_PAGE_ERASE](#) (0x03U)
- #define [FLASH_ERASE_ALL_BLOCK_UNSECURE](#) (0x04U)
- #define [FLASH_PROGRAM_PHRASE](#) (0x10U)
- #define [FLASH_PROGRAM_SECTION](#) (0x11U)
- #define [FLASH_PROGRAM_ONCE](#) (0x12U)
- #define [FLASH_VERIFY_SECTION](#) (0x20U)
- #define [FLASH_VERIFY_BLOCK](#) (0x21U)
- #define [FLASH_VERIFY_ALL_BLOCK](#) (0x22U)
- #define [FLASH_SECURITY_BY_PASS](#) (0x23U)
- #define [FLASH_ERASE_PARTITION](#) (0x30U)
- #define [FLASH_PROGRAM_PARTITION](#) (0x31U)
- #define [FLASH_SET_FLEXRAM](#) (0x32U)
- #define [FLASH_PROGRAM_CSE](#) (0x33U)

Enumerations

- enum [Flash_CseSize](#) { [FLASH_CSESIZE_0BYTES](#), [FLASH_CSESIZE_128BYTES](#), [FLASH_CSESIZE_256BYTES](#), [FLASH_CSESIZE_512BYTES](#) }
flash cse size

Functions

- Hal_StatusType [Flash_Hal_SetPartition](#) (const [Flash_CseType](#) *CseConfig)
This API set PARTITION SIZE.
- void [Flash_Hal_Init](#) (const [Flash_Config](#) *Config)
Initialize flash hardware.
- Hal_StatusType [Flash_Hal_Write](#) (uint8 CommandType, uint32 Address, uint32 Length, const uint8 *DataPtr)
Write one or more complete flash pages into given flash sector.
- Hal_StatusType [Flash_Hal_Erase](#) (uint8 CommandType, uint32 Address)
erase flash area
- Hal_StatusType [Flash_Hal_Verify](#) (uint8 CommandType, uint32 Address, uint32 Length)
verify input address
- Hal_StatusType [Flash_Hal_UnLockCtrl](#) (void)
unlock flash register write
- void [Flash_Hal_LockCtrl](#) (void)
lock flash register write
- Hal_StatusType [Flash_Hal_WPInfoErase](#) (void)
erase write protect information flash area
- Hal_StatusType [Flash_Hal_WriteOtp](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write flash area with OTP
- uint32 [Flash_Hal_ErrorStatusGetAndClear](#) (uint32 *Addr)
get error status and clear
- Hal_StatusType [Flash_Hal_AllVerify](#) (void)
verify all flash area
- Hal_StatusType [Flash_Hal_WriteProtectSet](#) (boolean En, const uint8 *Data)
disable/enable write protect information (need reboot work)
- Hal_StatusType [Flash_Hal_EraseCseInfo](#) (void)
Flash erase the UID,secret key and lock areas.
- Hal_StatusType [Flash_Hal_ProgramCseInfo](#) (uint32 Addr, uint32 Length, const uint8 *Data)
FLASH program uid,secret key,lock.
- Hal_StatusType [Flash_Hal_Swap](#) (boolean En)
FLASH program swap.
- Hal_StatusType [Flash_Hal_InfoErase](#) (uint32 Offset)
Erase one page from given flash info.
- Hal_StatusType [Flash_Hal_AllErase](#) (void)
erase all flash area
- Hal_StatusType [Flash_Hal_ReadProtectSet](#) (void)
enable read protect function
- Hal_StatusType [Flash_Hal_BackdoorSet](#) (const uint8 *Data)
set backdoor key
- Hal_StatusType [Flash_Hal_BackdoorVerify](#) (const uint8 *Data)
verify backdoor
- void [Flash_Hal_InfoRead](#) (uint32 Offset, uint32 Length, uint8 *DataPtr)
Read one or more bytes from given flash info.
- Hal_StatusType [Flash_Hal_InfoWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
Write one or more bytes from given flash info.
- Hal_StatusType [PFlash_Hal_PageErase](#) (uint32 Offset)
erase pflash area with page erase
- Hal_StatusType [PFlash_Hal_PageWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write pflash area with page program
- Hal_StatusType [PFlash_Hal_SectionVerify](#) (uint32 Offset, uint32 Length)
verify pflash area with section verify
- Hal_StatusType [PFlash_Hal_Read](#) (uint32 Offset, uint32 Length, uint8 *DataPtr)

Read one or more bytes from given pflash.

- Hal_StatusType [PFlash_Hal_BlockErase](#) (void)
erase pflash area with block erase
- Hal_StatusType [PFlash_Hal_SectionWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write pflash area with section program
- Hal_StatusType [PFlash_Hal_BlockVerify](#) (void)
verify pflash area with block verify
- Hal_StatusType [DFlash_Hal_PageErase](#) (uint32 Offset)
erase dflash area with page erase
- Hal_StatusType [DFlash_Hal_PageWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write dflash area with page program
- Hal_StatusType [DFlash_Hal_SectionVerify](#) (uint32 Offset, uint32 Length)
verify dflash area with section verify
- Hal_StatusType [DFlash_Hal_Read](#) (uint32 Offset, uint32 Length, uint8 *DataPtr)
Read one or more bytes from given dflash.
- Hal_StatusType [DFlash_Hal_BlockErase](#) (void)
erase dflash area with block erase
- Hal_StatusType [DFlash_Hal_SectionWrite](#) (uint32 Offset, uint32 Length, const uint8 *DataPtr)
write dflash area with section program
- Hal_StatusType [DFlash_Hal_BlockVerify](#) (void)
verify dflash area with block verify

4.4.1 Detailed Description

This file provides sdk flash api function.

4.4.2 Macro Definition Documentation

4.4.2.1 DFLASH_ENABLE

```
#define DFLASH_ENABLE (STD_ON)
```

Definition at line 74 of file Fls_Hal.h.

4.4.2.2 FLASH_ERASE_ALL_BLOCK_UNSECURE

```
#define FLASH_ERASE_ALL_BLOCK_UNSECURE (0x04U)
```

Definition at line 81 of file Fls_Hal.h.

4.4.2.3 FLASH_ERASE_BLOCK

```
#define FLASH_ERASE_BLOCK (0x02U)
```

Definition at line 79 of file Fls_Hal.h.

4.4.2.4 FLASH_ERASE_PAGE

```
#define FLASH_ERASE_PAGE (0x01U)
```

Definition at line 78 of file Fls_Hal.h.

4.4.2.5 FLASH_ERASE_PARTITION

```
#define FLASH_ERASE_PARTITION (0x30U)
```

Definition at line 89 of file Fls_Hal.h.

4.4.2.6 FLASH_OPTION_PAGE_ERASE

```
#define FLASH_OPTION_PAGE_ERASE (0x03U)
```

Definition at line 80 of file Fls_Hal.h.

4.4.2.7 FLASH_PROGRAM_CSE

```
#define FLASH_PROGRAM_CSE (0x33U)
```

Definition at line 92 of file Fls_Hal.h.

4.4.2.8 FLASH_PROGRAM_ONCE

```
#define FLASH_PROGRAM_ONCE (0x12U)
```

Definition at line 84 of file Fls_Hal.h.

4.4.2.9 FLASH_PROGRAM_PARTITION

```
#define FLASH_PROGRAM_PARTITION (0x31U)
```

Definition at line 90 of file Fls_Hal.h.

4.4.2.10 FLASH_PROGRAM_PHRASE

```
#define FLASH_PROGRAM_PHRASE (0x10U)
```

Definition at line 82 of file Fls_Hal.h.

4.4.2.11 FLASH_PROGRAM_SECTION

```
#define FLASH_PROGRAM_SECTION (0x11U)
```

Definition at line 83 of file Fls_Hal.h.

4.4.2.12 FLASH_SECURITY_BY_PASS

```
#define FLASH_SECURITY_BY_PASS (0x23U)
```

Definition at line 88 of file Fls_Hal.h.

4.4.2.13 FLASH_SET_FLEXRAM

```
#define FLASH_SET_FLEXRAM (0x32U)
```

Definition at line 91 of file Fls_Hal.h.

4.4.2.14 FLASH_VERIFY_ALL_BLOCK

```
#define FLASH_VERIFY_ALL_BLOCK (0x22U)
```

Definition at line 87 of file Fls_Hal.h.

4.4.2.15 FLASH_VERIFY_BLOCK

```
#define FLASH_VERIFY_BLOCK (0x21U)
```

Definition at line 86 of file Fls_Hal.h.

4.4.2.16 FLASH_VERIFY_SECTION

```
#define FLASH_VERIFY_SECTION (0x20U)
```

Definition at line 85 of file Fls_Hal.h.

4.4.2.17 PFLASH_ENABLE

```
#define PFLASH_ENABLE (STD_ON)
```

Definition at line 75 of file Fls_Hal.h.

4.4.3 Enumeration Type Documentation

4.4.3.1 Flash_CseSize

```
enum Flash_CseSize
```

flash cse size

Enumerator

FLASH_CSESIZE_0BYTES	
FLASH_CSESIZE_128BYTES	
FLASH_CSESIZE_256BYTES	
FLASH_CSESIZE_512BYTES	

Definition at line 56 of file Fls_Hal.h.

4.4.4 Function Documentation

4.4.4.1 DFlash_Hal_BlockErase()

```
Hal_StatusType DFlash_Hal_BlockErase (  
    void )
```

erase dflash area with block erase

Note

Function ID: [DES_FLS_API_226]

Returns

Hal_StatusType

4.4.4.2 DFlash_Hal_BlockVerify()

```
Hal_StatusType DFlash_Hal_BlockVerify (  
    void )
```

verify dflash area with block verify

Note

Function ID: [DES_FLS_API_230]

Returns

Hal_StatusType

4.4.4.3 DFlash_Hal_PageErase()

```
Hal_StatusType DFlash_Hal_PageErase (  
    uint32 Offset )
```

erase dflash area with page erase

Note

Function ID : DES_FLS_API_225

Parameters

in	<i>Offset</i>	offset in dFlash block
----	---------------	------------------------

Returns

Hal_StatusType

4.4.4.4 DFlash_Hal_PageWrite()

```
Hal_StatusType DFlash_Hal_PageWrite (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

write dflash area with page program

Note

Function ID: [DES_FLS_API_227]

Parameters

in	<i>Offset</i>	offset in dFlash block
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

4.4.4.5 DFlash_Hal_Read()

```
Hal_StatusType DFlash_Hal_Read (
    uint32 Offset,
    uint32 Length,
    uint8 * DataPtr )
```

Read one or more bytes from given dflash.

Note

Function ID: [DES_FLS_API_231]

Parameters

in	<i>Offset</i>	offset in dFlash block
in	<i>Length</i>	Flash read length
out	<i>DataPtr</i>	source data buffer address

Returns

void

4.4.4.6 DFlash_Hal_SectionVerify()

```
Hal_StatusType DFlash_Hal_SectionVerify (
    uint32 Offset,
    uint32 Length )
```

verify dflash area with section verify

Note

Function ID: [DES_FLS_API_229]

Parameters

in	<i>Offset</i>	offset in dFlash block
in	<i>Length</i>	Flash verify length

Returns

Hal_StatusType

4.4.4.7 DFlash_Hal_SectionWrite()

```
Hal_StatusType DFlash_Hal_SectionWrite (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

write dflash area with section program

Note

Function ID: [DES_FLS_API_228]

Parameters

in	<i>Offset</i>	offset in dFlash block
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

4.4.4.8 Flash_Hal_AllErase()

```
Hal_StatusType Flash_Hal_AllErase (
    void )
```

erase all flash area

Note

Function ID: [DES_FLS_API_213]

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_213]

Returns

Hal_StatusType

Definition at line 464 of file Fls_Hal.c.

4.4.4.9 Flash_Hal_AllVerify()

```
Hal_StatusType Flash_Hal_AllVerify (
    void )
```

verify all flash area

Note

Function ID: [DES_FLS_API_216]

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_216]

Returns

Hal_StatusType

Definition at line 662 of file Fls_Hal.c.

4.4.4.10 Flash_Hal_BackdoorSet()

```
Hal_StatusType Flash_Hal_BackdoorSet (
    const uint8 * Data )
```

set backdoor key

Note

Function ID: [DES_FLS_API_233]

Parameters

in	<i>Data</i>	backdoor key
----	-------------	--------------

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_233]

Parameters

in	<i>Data</i>	backdoor key
----	-------------	--------------

Returns

Hal_StatusType

Definition at line 743 of file Fls_Hal.c.

4.4.4.11 Flash_Hal_BackdoorVerify()

```
Hal_StatusType Flash_Hal_BackdoorVerify (
    const uint8 * Data )
```

verify backdoor

Note

Function ID: [DES_FLS_API_234]

Parameters

in	<i>Data</i>	backdoor key value
----	-------------	--------------------

Returns

Hal_StatusType

verify backdoor

Note

Function ID:[DES_FLS_API_234]

Parameters

in	<i>Data</i>	backdoor key
----	-------------	--------------

Returns

Hal_StatusType

Definition at line 769 of file Fls_Hal.c.

4.4.4.12 Flash_Hal_Erase()

```
Hal_StatusType Flash_Hal_Erase (
    uint8 CommandType,
    uint32 Address )
```

erase flash area

Note

Function ID: [DES_FLS_API_202]

Parameters

in	<i>CommandType</i>	Flash command type
in	<i>Address</i>	Flash erase address

Returns

Hal_StatusType

Definition at line 412 of file Fls_Hal.c.

4.4.4.13 Flash_Hal_EraseCseInfo()

```
Hal_StatusType Flash_Hal_EraseCseInfo (
    void )
```

Flash erase the UID,secret key and lock areas.

Note

Function ID: [DES_FLS_API_236]

Returns

Hal_StatusType

Flash erase the UID,secret key and lock areas.

Note

Function ID:[DES_FLS_API_235]

Returns

Hal_StatusType

Definition at line 794 of file Fls_Hal.c.

4.4.4.14 Flash_Hal_ErrorStatusGetAndClear()

```
uint32 Flash_Hal_ErrorStatusGetAndClear (
    uint32 * Addr )
```

get error status and clear

Note

Function ID: [DES_FLS_API_214]

Parameters

out	<i>Addr</i>	output the ecc error address
-----	-------------	------------------------------

Returns

error status register value

get error status and clear

Note

Function ID:[DES_FLS_API_223]

Parameters

out	<i>Addr</i>	ECC error address
-----	-------------	-------------------

Returns

error status

Definition at line 600 of file Fls_Hal.c.

4.4.4.15 Flash_Hal_InfoErase()

```
Hal_StatusType Flash_Hal_InfoErase (
    uint32 Offset )
```

Erase one page from given flash info.

Note

Function ID: [DES_FLS_API_205]

Parameters

in	<i>Offset</i>	offset in pFlash block
----	---------------	------------------------

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_204]

Parameters

in	<i>Offset</i>	info address offset
----	---------------	---------------------

Returns

Hal_StatusType

Definition at line 991 of file Fls_Hal.c.

4.4.4.16 Flash_Hal_InfoRead()

```
void Flash_Hal_InfoRead (
    uint32 Offset,
    uint32 Length,
    uint8 * DataPtr )
```

Read one or more bytes from given flash info.

Note

Function ID: [DES_FLS_API_204]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash read length
out	<i>DataPtr</i>	source data buffer address

Returns

void

Note

Function ID:[DES_FLS_API_210]

Parameters

in	<i>Offset</i>	info address offset
in	<i>Length</i>	Flash read length
in	<i>DataPtr</i>	source data buffer address

Returns

None

Definition at line 927 of file Fls_Hal.c.

4.4.4.17 Flash_Hal_InfoWrite()

```
Hal_StatusType Flash_Hal_InfoWrite (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

Write one or more bytes from given flash info.

Note

Function ID: [DES_FLS_API_205]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash read length
out	<i>DataPtr</i>	source data buffer address

Returns

void

Write one or more bytes from given flash info.

Note

Function ID:[DES_FLS_API_211]

Parameters

in	<i>Offset</i>	info address offset
in	<i>Length</i>	Flash read length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

Definition at line 962 of file Fls_Hal.c.

4.4.4.18 Flash_Hal_Init()

```
void Flash_Hal_Init (
    const Flash_Config * Config )
```

Initialize flash hardware.

Note

Function ID: [DES_FLS_API_201]

Parameters

in	<i>Config</i>	flash configure information
----	---------------	-----------------------------

Returns

void

Definition at line 391 of file Fls_Hal.c.

4.4.4.19 Flash_Hal_LockCtrl()

```
void Flash_Hal_LockCtrl (
    void )
```

lock flash register write

Note

Function ID: [DES_FLS_API_205]

Returns

void

Note

Function ID:[DES_FLS_API_205]

Returns

None

Definition at line 303 of file Fls_Hal.c.

4.4.4.20 Flash_Hal_ProgramCseInfo()

```
Hal_StatusType Flash_Hal_ProgramCseInfo (
    uint32 Addr,
    uint32 Length,
    const uint8 * Data )
```

FLASH program uid,secret key,lock.

Note

Function ID: [DES_FLS_API_237]

Parameters

in	<i>Addr</i>	Program start address
in	<i>Length</i>	Program size in byte
in	<i>Data</i>	Pointer of data source address from which will be programmed

Returns

operation status

- STATUS_SUCCESS: Operation was successful.
- STATUS_ERROR: Operation was failure.
- STATUS_BUSY: Operation was busy.

FLASH program uid,secret key,lock.

Note

Function ID:[DES_FLS_API_236]

Parameters

in	<i>Addr</i>	info address
in	<i>Length</i>	data length
in	<i>Data</i>	cse info

Returns

Hal_StatusType

Definition at line 814 of file Fls_Hal.c.

4.4.4.21 Flash_Hal_ReadProtectSet()

```
Hal_StatusType Flash_Hal_ReadProtectSet (  
    void )
```

enable read protect function

Note

Function ID: [DES_FLS_API_217]

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_217]

Returns

Hal_StatusType

Definition at line 691 of file Fls_Hal.c.

4.4.4.22 Flash_Hal_SetPartition()

```
Hal_StatusType Flash_Hal_SetPartition (  
    const Flash_CseType * CseConfig )
```

This API set PARTITION SIZE.

Note

Function ID: [DES_FLS_API_206]

Parameters

in	<i>CseConfig</i>	Cse configuration info about sfe and cse key size
----	------------------	---

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_206]

Parameters

in	<i>CseConfig</i>	Cse configuration info about sfe and cse key size
----	------------------	---

Returns

Hal_StatusType

Definition at line 343 of file Fls_Hal.c.

4.4.4.23 Flash_Hal_Swap()

```
Hal_StatusType Flash_Hal_Swap (  
    boolean En )
```

FLASH program swap.

Note

Function ID: [DES_FLS_API_238]

Parameters

in	<i>En</i>	swap config A/B
----	-----------	-----------------

Returns

operation status

- STATUS_SUCCESS: Operation was successful.
- STATUS_ERROR: Operation was failure.
- STATUS_BUSY: Operation was busy.

FLASH program swap.

Note

Function ID:[DES_FLS_API_248]

Parameters

in	En	0 -> swapA 1 -> swapB
----	----	-----------------------

Returns

Hal_StatusType

Definition at line 910 of file Fls_Hal.c.

4.4.4.24 Flash_Hal_UnLockCtrl()

```
Hal_StatusType Flash_Hal_UnLockCtrl (  
    void )
```

unlock flash register write

Note

Function ID: [DES_FLS_API_209]

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_209]

Returns

Hal_StatusType

Definition at line 313 of file Fls_Hal.c.

4.4.4.25 Flash_Hal_Verify()

```
Hal_StatusType Flash_Hal_Verify (  
    uint8 CommandType,  
    uint32 Address,  
    uint32 Length )
```

verify input address

Note

Function ID: [DES_FLS_API_215]

Parameters

in	<i>CommandType</i>	flash command type
in	<i>Address</i>	verify Address
in	<i>Length</i>	Data length to verify

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_215]

Parameters

in	<i>CommandType</i>	flash command type
in	<i>Address</i>	Write Address
in	<i>Length</i>	Data length

Returns

Hal_StatusType

Definition at line 628 of file Fls_Hal.c.

4.4.4.26 Flash_Hal_WPInfoErase()

```
Hal_StatusType Flash_Hal_WPInfoErase (  
    void )
```

erase write protect infomation flash area

Note

Function ID: [DES_FLS_API_212]

Returns

Hal_StatusType

Note

Function ID:[DES_FLS_API_212]

Returns

Hal_StatusType

Definition at line 448 of file Fls_Hal.c.

4.4.4.27 Flash_Hal_Write()

```
Hal_StatusType Flash_Hal_Write (
    uint8 CommandType,
    uint32 Address,
    uint32 Length,
    const uint8 * DataPtr )
```

Write one or more complete flash pages into given flash sector.

Note

Function ID: [DES_FLS_API_203]

Parameters

in	<i>CommandType</i>	flash command type
in	<i>Address</i>	Write Address
in	<i>Length</i>	Data length
in	<i>DataPtr</i>	source Data pointer which will write to flash

Returns

Hal_StatusType

Write one or more complete flash pages into given flash sector.

Note

Function ID:[DES_FLS_API_203]

Parameters

in	<i>CommandType</i>	flash command type
in	<i>Address</i>	Write Address
in	<i>Length</i>	Data length
in	<i>DataPtr</i>	Data pointer

Returns

Hal_StatusType

Definition at line 497 of file Fls_Hal.c.

4.4.4.28 Flash_Hal_WriteOtp()

```
Hal_StatusType Flash_Hal_WriteOtp (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

write flash area with OTP

Note

Function ID: [DES_FLS_API_222]

Parameters

in	<i>Offset</i>	offset in one physical page
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

write flash area with OTP

Note

Function ID:[DES_FLS_API_222]

Parameters

in	<i>Offset</i>	offset in one physical page
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

Definition at line 575 of file Fls_Hal.c.

4.4.4.29 Flash_Hal_WriteProtectSet()

```
Hal_StatusType Flash_Hal_WriteProtectSet (  
    boolean En,  
    const uint8 * Data )
```

disable/enable write protect information (need reboot work)

Note

Function ID: [DES_FLS_API_235]

Parameters

in	<i>En</i>	disable/enable write protect configuration
in	<i>Data</i>	pflash & dflash write protect config value

Returns

Hal_StatusType

disable/enable write protect information (need reboot work)

Note

Function ID:[DES_FLS_API_238]

Parameters

in	<i>En</i>	disable/enable write protect configuration
in	<i>Data</i>	pflash/dflash write protect config value

Returns

Hal_StatusType

Definition at line 714 of file Fls_Hal.c.

4.4.4.30 PFlash_Hal_BlockErase()

```
Hal_StatusType PFlash_Hal_BlockErase (  
    void )
```

erase pflash area with block erase

Note

Function ID: [DES_FLS_API_219]

Returns

Hal_StatusType

4.4.4.31 PFlash_Hal_BlockVerify()

```
Hal_StatusType PFlash_Hal_BlockVerify (  
    void )
```

verify pflash area with block verify

Note

Function ID: [DES_FLS_API_224]

Returns

Hal_StatusType

4.4.4.32 PFlash_Hal_PageErase()

```
Hal_StatusType PFlash_Hal_PageErase (
    uint32 Offset )
```

erase pflash area with page erase

Note

Function ID: [DES_FLS_API_218]

Parameters

in	<i>Offset</i>	offset in pFlash block
----	---------------	------------------------

Returns

Hal_StatusType

4.4.4.33 PFlash_Hal_PageWrite()

```
Hal_StatusType PFlash_Hal_PageWrite (
    uint32 Offset,
    uint32 Length,
    const uint8 * DataPtr )
```

write pflash area with page program

Note

Function ID: [DES_FLS_API_220]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

4.4.4.34 PFlash_Hal_Read()

```
Hal_StatusType PFlash_Hal_Read (
    uint32 Offset,
```

```
uint32 Length,  
uint8 * DataPtr )
```

Read one or more bytes from given pflash.

Note

Function ID: [DES_FLS_API_204]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash read length
out	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

4.4.4.35 PFlash_Hal_SectionVerify()

```
Hal_StatusType PFlash_Hal_SectionVerify (  
    uint32 Offset,  
    uint32 Length )
```

verify pflash area with section verify

Note

Function ID: [DES_FLS_API_223]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash verify length

Returns

Hal_StatusType

4.4.4.36 PFlash_Hal_SectionWrite()

```
Hal_StatusType PFlash_Hal_SectionWrite (  
    uint32 Offset,  
    uint32 Length,  
    const uint8 * DataPtr )
```

write pflash area with section program

Note

Function ID: [DES_FLS_API_221]

Parameters

in	<i>Offset</i>	offset in pFlash block
in	<i>Length</i>	Flash write length
in	<i>DataPtr</i>	source data buffer address

Returns

Hal_StatusType

Index

AC784xx_API_Reference_Manual_FLASH.pdf, [5](#)

AC784xx_Fls_Reg.h, [5](#)

- FLASH_ClearFRAMECC1BitFlag, [7](#)
- FLASH_ClearFRAMECC2BitFlag, [7](#)
- FLASH_GetFRAMECC1BitFlag, [8](#)
- FLASH_GetFRAMECC2BitFlag, [8](#)
- FLASH_STAT_ERROR_BITS, [7](#)
- Flash_Reg_AbortPageEraseReg, [8](#)
- Flash_Reg_ClearCollIRQFlag, [9](#)
- Flash_Reg_ClearECCErrorAddress, [9](#)
- Flash_Reg_ClearECCIRQFlag, [9](#)
- Flash_Reg_ClearStatus, [10](#)
- Flash_Reg_ECCErrorAddr, [10](#)
- Flash_Reg_EnableCollisionIRQ, [10](#)
- Flash_Reg_EnableCompleteIRQ, [11](#)
- Flash_Reg_EnableECCIRQ, [11](#)
- Flash_Reg_ForceECCErrorDetect, [11](#)
- Flash_Reg_GetAbortPageEraseStatusReg, [12](#)
- Flash_Reg_GetBackdoorKeyStatus, [12](#)
- Flash_Reg_GetBackdoorStatusReg, [12](#)
- Flash_Reg_GetCSEPartition, [14](#)
- Flash_Reg_GetCnfg, [13](#)
- Flash_Reg_GetCollIRQFlag, [13](#)
- Flash_Reg_GetCommandCompleteStatusReg, [13](#)
- Flash_Reg_GetDFlashProtection, [14](#)
- Flash_Reg_GetDPartition, [14](#)
- Flash_Reg_GetECCIRQFlag, [15](#)
- Flash_Reg_GetEPartition, [15](#)
- Flash_Reg_GetHighDataReg, [15](#)
- Flash_Reg_GetLockStatusReg, [16](#)
- Flash_Reg_GetLowDataReg, [16](#)
- Flash_Reg_GetPFlashProtection, [16](#)
- Flash_Reg_GetReadProtectStatus, [17](#)
- Flash_Reg_GetStatusReg, [17](#)
- Flash_Reg_GetTrigCtrlCmdRegStatus, [17](#)
- Flash_Reg_Lock, [18](#)
- Flash_Reg_SetAddressReg, [18](#)
- Flash_Reg_SetCommandReg, [19](#)
- Flash_Reg_SetDFlashProtection, [19](#)
- Flash_Reg_SetEMDFIE, [20](#)
- Flash_Reg_SetHighDataReg, [20](#)
- Flash_Reg_SetLVDDetection, [21](#)
- Flash_Reg_SetLengthReg, [20](#)
- Flash_Reg_SetLowDataReg, [21](#)
- Flash_Reg_SetPFlashProtection, [22](#)
- Flash_Reg_TrigCtrlCmdReg, [22](#)
- Flash_Reg_UnLock, [23](#)

DFLASH_ENABLE

Fls_Hal.h, [37](#)

DFlash_Hal_BlockErase

Fls_Hal.h, [40](#)

DFlash_Hal_BlockVerify

Fls_Hal.h, [41](#)

DFlash_Hal_PageErase

Fls_Hal.h, [41](#)

DFlash_Hal_PageWrite

Fls_Hal.h, [42](#)

DFlash_Hal_Read

Fls_Hal.h, [42](#)

DFlash_Hal_SectionVerify

Fls_Hal.h, [43](#)

DFlash_Hal_SectionWrite

Fls_Hal.h, [43](#)

Erase_Timeout

Flash_Config, [3](#)

FLASH_ClearFRAMECC1BitFlag

AC784xx_Fls_Reg.h, [7](#)

FLASH_ClearFRAMECC2BitFlag

AC784xx_Fls_Reg.h, [7](#)

FLASH_ERASE_ALL_BLOCK_UNSECURE

Fls_Hal.h, [37](#)

FLASH_ERASE_BLOCK

Fls_Hal.h, [37](#)

FLASH_ERASE_PAGE

Fls_Hal.h, [38](#)

FLASH_ERASE_PARTITION

Fls_Hal.h, [38](#)

FLASH_GetFRAMECC1BitFlag

AC784xx_Fls_Reg.h, [8](#)

FLASH_GetFRAMECC2BitFlag

AC784xx_Fls_Reg.h, [8](#)

FLASH_INVALID_ADDR

Fls_Hal.c, [25](#)

FLASH_NOT_USE_ADDR

Fls_Hal.c, [25](#)

FLASH_OPTION_PAGE_ERASE

Fls_Hal.h, [38](#)

FLASH_PROGRAM_CSE

Fls_Hal.h, [38](#)

FLASH_PROGRAM_ONCE

Fls_Hal.h, [38](#)

FLASH_PROGRAM_PARTITION

Fls_Hal.h, [38](#)

FLASH_PROGRAM_PHRASE

Fls_Hal.h, [39](#)

FLASH_PROGRAM_SECTION

Fls_Hal.h, [39](#)

FLASH_SECURITY_BY_PASS

Fls_Hal.h, [39](#)

FLASH_SET_FLEXRAM

Fls_Hal.h, [39](#)

FLASH_STAT_ERROR_BITS
 AC784xx_Fls_Reg.h, [7](#)
 FLASH_VERIFY_ALL_BLOCK
 Fls_Hal.h, [39](#)
 FLASH_VERIFY_BLOCK
 Fls_Hal.h, [39](#)
 FLASH_VERIFY_SECTION
 Fls_Hal.h, [40](#)
 Flash_Config, [3](#)
 Erase_Timeout, [3](#)
 LVD_Enable, [3](#)
 Write_Timeout, [4](#)
 Flash_CseSize
 Fls_Hal.h, [40](#)
 Flash_CseType, [4](#)
 KeySize, [4](#)
 Sfe, [4](#)
 Flash_Hal_AllErase
 Fls_Hal.c, [25](#)
 Fls_Hal.h, [44](#)
 Flash_Hal_AllVerify
 Fls_Hal.c, [25](#)
 Fls_Hal.h, [44](#)
 Flash_Hal_BackdoorSet
 Fls_Hal.c, [25](#)
 Fls_Hal.h, [44](#)
 Flash_Hal_BackdoorVerify
 Fls_Hal.c, [26](#)
 Fls_Hal.h, [45](#)
 Flash_Hal_Erase
 Fls_Hal.c, [26](#)
 Fls_Hal.h, [46](#)
 Flash_Hal_EraseCseInfo
 Fls_Hal.c, [27](#)
 Fls_Hal.h, [46](#)
 Flash_Hal_ErrorStatusGetAndClear
 Fls_Hal.c, [27](#)
 Fls_Hal.h, [47](#)
 Flash_Hal_InfoErase
 Fls_Hal.c, [28](#)
 Fls_Hal.h, [48](#)
 Flash_Hal_InfoRead
 Fls_Hal.c, [28](#)
 Fls_Hal.h, [48](#)
 Flash_Hal_InfoWrite
 Fls_Hal.c, [29](#)
 Fls_Hal.h, [49](#)
 Flash_Hal_Init
 Fls_Hal.c, [29](#)
 Fls_Hal.h, [50](#)
 Flash_Hal_LockCtrl
 Fls_Hal.c, [30](#)
 Fls_Hal.h, [50](#)
 Flash_Hal_ProgramCseInfo
 Fls_Hal.c, [30](#)
 Fls_Hal.h, [51](#)
 Flash_Hal_ReadProtectSet
 Fls_Hal.c, [31](#)
 Fls_Hal.h, [52](#)
 Flash_Hal_SetPartition
 Fls_Hal.c, [31](#)
 Fls_Hal.h, [52](#)
 Flash_Hal_Swap
 Fls_Hal.c, [31](#)
 Fls_Hal.h, [53](#)
 Flash_Hal_UnLockCtrl
 Fls_Hal.c, [32](#)
 Fls_Hal.h, [54](#)
 Flash_Hal_Verify
 Fls_Hal.c, [32](#)
 Fls_Hal.h, [54](#)
 Flash_Hal_WPInfoErase
 Fls_Hal.c, [33](#)
 Fls_Hal.h, [55](#)
 Flash_Hal_Write
 Fls_Hal.c, [33](#)
 Fls_Hal.h, [55](#)
 Flash_Hal_WriteOtp
 Fls_Hal.c, [34](#)
 Fls_Hal.h, [56](#)
 Flash_Hal_WriteProtectSet
 Fls_Hal.c, [34](#)
 Fls_Hal.h, [57](#)
 Flash_Reg_AbortPageEraseReg
 AC784xx_Fls_Reg.h, [8](#)
 Flash_Reg_ClearCollIRQFlag
 AC784xx_Fls_Reg.h, [9](#)
 Flash_Reg_ClearECCErrAddress
 AC784xx_Fls_Reg.h, [9](#)
 Flash_Reg_ClearECCIRQFlag
 AC784xx_Fls_Reg.h, [9](#)
 Flash_Reg_ClearStatus
 AC784xx_Fls_Reg.h, [10](#)
 Flash_Reg_ECCErrorAddr
 AC784xx_Fls_Reg.h, [10](#)
 Flash_Reg_EnableCollisionIRQ
 AC784xx_Fls_Reg.h, [10](#)
 Flash_Reg_EnableCompleteIRQ
 AC784xx_Fls_Reg.h, [11](#)
 Flash_Reg_EnableECCIRQ
 AC784xx_Fls_Reg.h, [11](#)
 Flash_Reg_ForceECCErrDetect
 AC784xx_Fls_Reg.h, [11](#)
 Flash_Reg_GetAbortPageEraseStatusReg
 AC784xx_Fls_Reg.h, [12](#)
 Flash_Reg_GetBackdoorKeyStatus
 AC784xx_Fls_Reg.h, [12](#)
 Flash_Reg_GetBackdoorStatusReg
 AC784xx_Fls_Reg.h, [12](#)
 Flash_Reg_GetCSEPartition
 AC784xx_Fls_Reg.h, [14](#)
 Flash_Reg_GetCnfg
 AC784xx_Fls_Reg.h, [13](#)
 Flash_Reg_GetCollIRQFlag
 AC784xx_Fls_Reg.h, [13](#)
 Flash_Reg_GetCommandCompleteStatusReg
 AC784xx_Fls_Reg.h, [13](#)
 Flash_Reg_GetDFlashProtection
 AC784xx_Fls_Reg.h, [14](#)
 Flash_Reg_GetDPartition
 AC784xx_Fls_Reg.h, [14](#)
 Flash_Reg_GetECCIRQFlag

- AC784xx_Fls_Reg.h, [15](#)
- Flash_Reg_GetEPartition
 - AC784xx_Fls_Reg.h, [15](#)
- Flash_Reg_GetHighDataReg
 - AC784xx_Fls_Reg.h, [15](#)
- Flash_Reg_GetLockStatusReg
 - AC784xx_Fls_Reg.h, [16](#)
- Flash_Reg_GetLowDataReg
 - AC784xx_Fls_Reg.h, [16](#)
- Flash_Reg_GetPFlashProtection
 - AC784xx_Fls_Reg.h, [16](#)
- Flash_Reg_GetReadProtectStatus
 - AC784xx_Fls_Reg.h, [17](#)
- Flash_Reg_GetStatusReg
 - AC784xx_Fls_Reg.h, [17](#)
- Flash_Reg_GetTrigCtrlCmdRegStatus
 - AC784xx_Fls_Reg.h, [17](#)
- Flash_Reg_Lock
 - AC784xx_Fls_Reg.h, [18](#)
- Flash_Reg_SetAddressReg
 - AC784xx_Fls_Reg.h, [18](#)
- Flash_Reg_SetCommandReg
 - AC784xx_Fls_Reg.h, [19](#)
- Flash_Reg_SetDFlashProtection
 - AC784xx_Fls_Reg.h, [19](#)
- Flash_Reg_SetEMDFIE
 - AC784xx_Fls_Reg.h, [20](#)
- Flash_Reg_SetHighDataReg
 - AC784xx_Fls_Reg.h, [20](#)
- Flash_Reg_SetLVDDetection
 - AC784xx_Fls_Reg.h, [21](#)
- Flash_Reg_SetLengthReg
 - AC784xx_Fls_Reg.h, [20](#)
- Flash_Reg_SetLowDataReg
 - AC784xx_Fls_Reg.h, [21](#)
- Flash_Reg_SetPFlashProtection
 - AC784xx_Fls_Reg.h, [22](#)
- Flash_Reg_TrigCtrlCmdReg
 - AC784xx_Fls_Reg.h, [22](#)
- Flash_Reg_UnLock
 - AC784xx_Fls_Reg.h, [23](#)
- Fls_Hal.c, [23](#)
 - FLASH_INVALID_ADDR, [25](#)
 - FLASH_NOT_USE_ADDR, [25](#)
 - Flash_Hal_AllErase, [25](#)
 - Flash_Hal_AllVerify, [25](#)
 - Flash_Hal_BackdoorSet, [25](#)
 - Flash_Hal_BackdoorVerify, [26](#)
 - Flash_Hal_Erase, [26](#)
 - Flash_Hal_EraseCseInfo, [27](#)
 - Flash_Hal_ErrorStatusGetAndClear, [27](#)
 - Flash_Hal_InfoErase, [28](#)
 - Flash_Hal_InfoRead, [28](#)
 - Flash_Hal_InfoWrite, [29](#)
 - Flash_Hal_Init, [29](#)
 - Flash_Hal_LockCtrl, [30](#)
 - Flash_Hal_ProgramCseInfo, [30](#)
 - Flash_Hal_ReadProtectSet, [31](#)
 - Flash_Hal_SetPartition, [31](#)
 - Flash_Hal_Swap, [31](#)
 - Flash_Hal_UnLockCtrl, [32](#)
 - Flash_Hal_Verify, [32](#)
 - Flash_Hal_WPInfoErase, [33](#)
 - Flash_Hal_Write, [33](#)
 - Flash_Hal_WriteOtp, [34](#)
 - Flash_Hal_WriteProtectSet, [34](#)
- Fls_Hal.h, [35](#)
 - DFLASH_ENABLE, [37](#)
 - DFlash_Hal_BlockErase, [40](#)
 - DFlash_Hal_BlockVerify, [41](#)
 - DFlash_Hal_PageErase, [41](#)
 - DFlash_Hal_PageWrite, [42](#)
 - DFlash_Hal_Read, [42](#)
 - DFlash_Hal_SectionVerify, [43](#)
 - DFlash_Hal_SectionWrite, [43](#)
 - FLASH_ERASE_ALL_BLOCK_UNSECURE, [37](#)
 - FLASH_ERASE_BLOCK, [37](#)
 - FLASH_ERASE_PAGE, [38](#)
 - FLASH_ERASE_PARTITION, [38](#)
 - FLASH_OPTION_PAGE_ERASE, [38](#)
 - FLASH_PROGRAM_CSE, [38](#)
 - FLASH_PROGRAM_ONCE, [38](#)
 - FLASH_PROGRAM_PARTITION, [38](#)
 - FLASH_PROGRAM_PHRASE, [39](#)
 - FLASH_PROGRAM_SECTION, [39](#)
 - FLASH_SECURITY_BY_PASS, [39](#)
 - FLASH_SET_FLEXRAM, [39](#)
 - FLASH_VERIFY_ALL_BLOCK, [39](#)
 - FLASH_VERIFY_BLOCK, [39](#)
 - FLASH_VERIFY_SECTION, [40](#)
 - Flash_CseSize, [40](#)
 - Flash_Hal_AllErase, [44](#)
 - Flash_Hal_AllVerify, [44](#)
 - Flash_Hal_BackdoorSet, [44](#)
 - Flash_Hal_BackdoorVerify, [45](#)
 - Flash_Hal_Erase, [46](#)
 - Flash_Hal_EraseCseInfo, [46](#)
 - Flash_Hal_ErrorStatusGetAndClear, [47](#)
 - Flash_Hal_InfoErase, [48](#)
 - Flash_Hal_InfoRead, [48](#)
 - Flash_Hal_InfoWrite, [49](#)
 - Flash_Hal_Init, [50](#)
 - Flash_Hal_LockCtrl, [50](#)
 - Flash_Hal_ProgramCseInfo, [51](#)
 - Flash_Hal_ReadProtectSet, [52](#)
 - Flash_Hal_SetPartition, [52](#)
 - Flash_Hal_Swap, [53](#)
 - Flash_Hal_UnLockCtrl, [54](#)
 - Flash_Hal_Verify, [54](#)
 - Flash_Hal_WPInfoErase, [55](#)
 - Flash_Hal_Write, [55](#)
 - Flash_Hal_WriteOtp, [56](#)
 - Flash_Hal_WriteProtectSet, [57](#)
 - PFLASH_ENABLE, [40](#)
 - PFlash_Hal_BlockErase, [58](#)
 - PFlash_Hal_BlockVerify, [58](#)
 - PFlash_Hal_PageErase, [58](#)
 - PFlash_Hal_PageWrite, [59](#)
 - PFlash_Hal_Read, [59](#)
 - PFlash_Hal_SectionVerify, [60](#)
 - PFlash_Hal_SectionWrite, [60](#)

KeySize
 Flash_CseType, [4](#)

LVD_Enable
 Flash_Config, [3](#)

PFLASH_ENABLE
 Fls_Hal.h, [40](#)

PFlash_Hal_BlockErase
 Fls_Hal.h, [58](#)

PFlash_Hal_BlockVerify
 Fls_Hal.h, [58](#)

PFlash_Hal_PageErase
 Fls_Hal.h, [58](#)

PFlash_Hal_PageWrite
 Fls_Hal.h, [59](#)

PFlash_Hal_Read
 Fls_Hal.h, [59](#)

PFlash_Hal_SectionVerify
 Fls_Hal.h, [60](#)

PFlash_Hal_SectionWrite
 Fls_Hal.h, [60](#)

Sfe
 Flash_CseType, [4](#)

Write_Timeout
 Flash_Config, [4](#)