

AC784xx_DFP SMU

5.1.0

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	Smu_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	Reset_Count	3
3.1.2.2	Reset_Lfs	4
3.1.2.3	Reset_Spfs	4
3.2	Smu_FaultInfoType Struct Reference	4
3.2.1	Detailed Description	4
3.2.2	Member Data Documentation	4
3.2.2.1	Count	5
3.2.2.2	Lfs	5
3.2.2.3	Spfs	5
3.2.2.4	SwLfs	5
3.2.2.5	SwSpfs	5

4	File Documentation	6
4.1	AC784xx_API_Reference_Manual_SMU.pdf File Reference	6
4.2	AC784xx_Smu_Reg.h File Reference	6
4.2.1	Detailed Description	7
4.2.2	Function Documentation	7
4.2.2.1	Smu_Reg_ClearResetCounter()	8
4.2.2.2	Smu_Reg_ClearResetCounterShadow()	8
4.2.2.3	Smu_Reg_GetLatentFaultResetEnable()	8
4.2.2.4	Smu_Reg_GetLatentFaultStatus()	9
4.2.2.5	Smu_Reg_GetLatentFaultStatusShadow()	9
4.2.2.6	Smu_Reg_GetPathCheck0()	10
4.2.2.7	Smu_Reg_GetPathCheck1()	10
4.2.2.8	Smu_Reg_GetResetCounter()	11
4.2.2.9	Smu_Reg_GetResetCounterShadow()	11
4.2.2.10	Smu_Reg_GetResetCounterThreshold()	12
4.2.2.11	Smu_Reg_GetSoftwareLatentFault()	12
4.2.2.12	Smu_Reg_GetSoftwareLatentFaultShadow()	12
4.2.2.13	Smu_Reg_GetSoftwareSPFault()	13
4.2.2.14	Smu_Reg_GetSoftwareSPFaultShadow()	13
4.2.2.15	Smu_Reg_GetSPFaultResetEnable()	14
4.2.2.16	Smu_Reg_GetSPFaultStatus()	14
4.2.2.17	Smu_Reg_GetSPFaultStatusShadow()	15
4.2.2.18	Smu_Reg_LatentFaultRegLock()	15
4.2.2.19	Smu_Reg_LatentFaultRegUnlock()	16
4.2.2.20	Smu_Reg_SetLatentFaultResetEnable()	16
4.2.2.21	Smu_Reg_SetLatentFaultStatus()	16
4.2.2.22	Smu_Reg_SetLatentFaultStatusShadow()	17
4.2.2.23	Smu_Reg_SetPathCheck0()	17
4.2.2.24	Smu_Reg_SetPathCheck1()	18
4.2.2.25	Smu_Reg_SetResetCounterThreshold()	18
4.2.2.26	Smu_Reg_SetSoftwareLatentFault()	19

4.2.2.27	Smu_Reg_SetSoftwareSPFault()	19
4.2.2.28	Smu_Reg_SetSPFaultResetEnable()	20
4.2.2.29	Smu_Reg_SetSPFaultStatus()	20
4.2.2.30	Smu_Reg_SetSPFaultStatusShadow()	20
4.2.2.31	Smu_Reg_SPFaultRegLock()	21
4.2.2.32	Smu_Reg_SPFaultRegUnlock()	21
4.3	Smu_Hal.c File Reference	22
4.3.1	Detailed Description	22
4.3.2	Function Documentation	23
4.3.2.1	Smu_Hal_AssertSwFault()	23
4.3.2.2	Smu_Hal_ClearCurrentFaultInfo()	23
4.3.2.3	Smu_Hal_ClearFault()	24
4.3.2.4	Smu_Hal_DeassertSwFault()	24
4.3.2.5	Smu_Hal_DeInit()	25
4.3.2.6	Smu_Hal_GetCurrentFaults()	25
4.3.2.7	Smu_Hal_GetLastFaults()	25
4.3.2.8	Smu_Hal_Init()	26
4.3.2.9	Smu_Hal_InjectFault()	26
4.3.2.10	Smu_Hal_Lock()	27
4.3.2.11	Smu_Hal_Unlock()	27
4.4	Smu_Hal.h File Reference	28
4.4.1	Detailed Description	28
4.4.2	Function Documentation	28
4.4.2.1	Smu_Hal_AssertSwFault()	28
4.4.2.2	Smu_Hal_ClearCurrentFaultInfo()	29
4.4.2.3	Smu_Hal_ClearFault()	29
4.4.2.4	Smu_Hal_DeassertSwFault()	30
4.4.2.5	Smu_Hal_DeInit()	30
4.4.2.6	Smu_Hal_GetCurrentFaults()	31
4.4.2.7	Smu_Hal_GetLastFaults()	31
4.4.2.8	Smu_Hal_Init()	32
4.4.2.9	Smu_Hal_InjectFault()	32
4.4.2.10	Smu_Hal_Lock()	33
4.4.2.11	Smu_Hal_Unlock()	33
4.5	Smu_Hal_Types.h File Reference	34
4.5.1	Detailed Description	34

Chapter 1

Class Index

1.1 Class List

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

Smu_ConfigType	
SMU module configuration structure	3
Smu_FaultInfoType	
SMU module information structure	4

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_SMU.pdf	6
AC784xx_Smu_Reg.h This file provides extern Low level Smu register api	6
Smu_Hal.c This file provides Hal Smu api	22
Smu_Hal.h This file provides extern Hal Smu api	28
Smu_Hal_Types.h This file provides Smu hal types header	34

Chapter 3

Class Documentation

3.1 Smu_ConfigType Struct Reference

SMU module configuration structure.

```
#include <Smu_Hal_Types.h>
```

Public Attributes

- uint8 [Reset_Count](#)
- uint32 [Reset_Spfs](#)
- uint32 [Reset_Lfs](#)

3.1.1 Detailed Description

SMU module configuration structure.

Definition at line 65 of file Smu_Hal_Types.h.

3.1.2 Member Data Documentation

3.1.2.1 Reset_Count

```
uint8 Smu_ConfigType::Reset_Count
```

SMU reset threshold count.

Definition at line 67 of file Smu_Hal_Types.h.

3.1.2.2 Reset_Lfs

```
uint32 Smu_ConfigType::Reset_Lfs
```

Hardware and Software latent faults enable.

Definition at line 69 of file Smu_Hal_Types.h.

3.1.2.3 Reset_Spfs

```
uint32 Smu_ConfigType::Reset_Spfs
```

Hardware and Software single point faults enable.

Definition at line 68 of file Smu_Hal_Types.h.

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

- [Smu_Hal_Types.h](#)

3.2 Smu_FaultInfoType Struct Reference

SMU module information structure.

```
#include <Smu_Hal_Types.h>
```

Public Attributes

- uint8 [Count](#)
- uint32 [Spfs](#)
- uint32 [Lfs](#)
- uint32 [SwSpfs](#)
- uint32 [SwLfs](#)

3.2.1 Detailed Description

SMU module information structure.

Definition at line 85 of file Smu_Hal_Types.h.

3.2.2 Member Data Documentation

3.2.2.1 Count

```
uint8 Smu_FaultInfoType::Count
```

SMU reset count.

Definition at line 87 of file Smu_Hal_Types.h.

3.2.2.2 Lfs

```
uint32 Smu_FaultInfoType::Lfs
```

Hardware and Software latent faults status.

Definition at line 89 of file Smu_Hal_Types.h.

3.2.2.3 Spfs

```
uint32 Smu_FaultInfoType::Spfs
```

Hardware and Software single point faults status.

Definition at line 88 of file Smu_Hal_Types.h.

3.2.2.4 SwLfs

```
uint32 Smu_FaultInfoType::SwLfs
```

Software latent faults status.

Definition at line 91 of file Smu_Hal_Types.h.

3.2.2.5 SwSpfs

```
uint32 Smu_FaultInfoType::SwSpfs
```

Software latent faults status.

Definition at line 90 of file Smu_Hal_Types.h.

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

- [Smu_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_SMU.pdf File Reference

4.2 AC784xx_Smu_Reg.h File Reference

This file provides extern Low level Smu register api.

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

Functions

- LOCAL_INLINE void [Smu_Reg_SPFaultRegLock](#) (void)
Single point fault register access permission lock.
- LOCAL_INLINE void [Smu_Reg_SPFaultRegUnlock](#) (void)
Single point fault register access permission unlock.
- LOCAL_INLINE void [Smu_Reg_LatentFaultRegLock](#) (void)
Latent fault register access permission lock.
- LOCAL_INLINE void [Smu_Reg_LatentFaultRegUnlock](#) (void)
Latent fault register access permission unlock.
- LOCAL_INLINE void [Smu_Reg_SetSPFaultResetEnable](#) (uint32 SpfChannels)
Enable which single point fault channels could trigger reset.
- LOCAL_INLINE uint32 [Smu_Reg_GetSPFaultResetEnable](#) (void)
Get which single point fault channel reset.
- LOCAL_INLINE void [Smu_Reg_SetLatentFaultResetEnable](#) (uint32 LfChannels)
Enable which latent fault channels could reset.
- LOCAL_INLINE uint32 [Smu_Reg_GetLatentFaultResetEnable](#) (void)
Get which latent fault channel reset function.
- LOCAL_INLINE void [Smu_Reg_SetSoftwareSPFault](#) (uint32 SwSpfChannels)
Set software single point fault channel function.
- LOCAL_INLINE uint32 [Smu_Reg_GetSoftwareSPFault](#) (void)
Set software single point fault channel function.
- LOCAL_INLINE uint32 [Smu_Reg_GetSoftwareSPFaultShadow](#) (void)
Set software single point fault shadow channels function.
- LOCAL_INLINE void [Smu_Reg_SetSoftwareLatentFault](#) (uint32 SwLfChannels)
Set software latent fault channel function.

- LOCAL_INLINE uint32 [Smu_Reg_GetSoftwareLatentFault](#) (void)
Get software latent fault channel function.
- LOCAL_INLINE uint32 [Smu_Reg_GetSoftwareLatentFaultShadow](#) (void)
Get software latent fault shadow channel function.
- LOCAL_INLINE void [Smu_Reg_SetResetCounterThreshold](#) (uint8 Val)
Set SMU reset counter value.
- LOCAL_INLINE uint8 [Smu_Reg_GetResetCounterThreshold](#) (void)
Set SMU reset counter value.
- LOCAL_INLINE uint32 [Smu_Reg_GetSPFaultStatus](#) (void)
Get SMU single point fault status.
- LOCAL_INLINE void [Smu_Reg_SetSPFaultStatus](#) (uint32 SpfChannels)
Set SMU single point fault status.
- LOCAL_INLINE uint32 [Smu_Reg_GetSPFaultStatusShadow](#) (void)
Get SMU single point fault status.
- LOCAL_INLINE void [Smu_Reg_SetSPFaultStatusShadow](#) (uint32 SpfChannels)
Set SMU single point fault status.
- LOCAL_INLINE uint32 [Smu_Reg_GetLatentFaultStatus](#) (void)
Get SMU latent fault status.
- LOCAL_INLINE void [Smu_Reg_SetLatentFaultStatus](#) (uint32 LfChannels)
Set SMU latent fault status.
- LOCAL_INLINE uint32 [Smu_Reg_GetLatentFaultStatusShadow](#) (void)
Get SMU latent fault status.
- LOCAL_INLINE void [Smu_Reg_SetLatentFaultStatusShadow](#) (uint32 LfChannels)
Set SMU latent fault status.
- LOCAL_INLINE uint8 [Smu_Reg_GetResetCounter](#) (void)
Get SMU reset counter status.
- LOCAL_INLINE void [Smu_Reg_ClearResetCounter](#) (void)
Clear SMU reset counter.
- LOCAL_INLINE uint8 [Smu_Reg_GetResetCounterShadow](#) (void)
Get SMU reset counter shadow status.
- LOCAL_INLINE void [Smu_Reg_ClearResetCounterShadow](#) (void)
Clear SMU reset counter shadow.
- LOCAL_INLINE void [Smu_Reg_SetPathCheck0](#) (uint32 SpfChannels)
Set SMU patch check0.
- LOCAL_INLINE uint32 [Smu_Reg_GetPathCheck0](#) (void)
Get SMU reset patch check0.
- LOCAL_INLINE void [Smu_Reg_SetPathCheck1](#) (uint32 LfChannels)
Set SMU patch check1.
- LOCAL_INLINE uint32 [Smu_Reg_GetPathCheck1](#) (void)
Get SMU reset patch check1.

4.2.1 Detailed Description

This file provides extern Low level Smu register api.

4.2.2 Function Documentation

4.2.2.1 Smu_Reg_ClearResetCounter()

```
LOCAL_INLINE void Smu_Reg_ClearResetCounter (
    void )
```

Clear SMU reset counter.

Note

Function ID: DES_SMU_API_125

Parameters

in	void	
----	------	--

Returns

void

Definition at line 348 of file AC784xx_Smu_Reg.h.

4.2.2.2 Smu_Reg_ClearResetCounterShadow()

```
LOCAL_INLINE void Smu_Reg_ClearResetCounterShadow (
    void )
```

Clear SMU reset counter shadow.

Note

Function ID: DES_SMU_API_127

Parameters

in	void	
----	------	--

Returns

void

Definition at line 371 of file AC784xx_Smu_Reg.h.

4.2.2.3 Smu_Reg_GetLatentFaultResetEnable()

```
LOCAL_INLINE uint32 Smu_Reg_GetLatentFaultResetEnable (
    void )
```

Get which latent fault channel reset function.

Note

Function ID: DES_SMU_API_107

Parameters

in	void	
----	------	--

Returns

latent fault channels.

Definition at line 148 of file AC784xx_Smu_Reg.h.

4.2.2.4 Smu_Reg_GetLatentFaultStatus()

```
LOCAL_INLINE uint32 Smu_Reg_GetLatentFaultStatus (  
    void )
```

Get SMU latent fault status.

Note

Function ID: DES_SMU_API_120

Parameters

in	void	
----	------	--

Returns

SMU latent fault status.

Definition at line 292 of file AC784xx_Smu_Reg.h.

4.2.2.5 Smu_Reg_GetLatentFaultStatusShadow()

```
LOCAL_INLINE uint32 Smu_Reg_GetLatentFaultStatusShadow (  
    void )
```

Get SMU latent fault status.

Note

Function ID: DES_SMU_API_122

Parameters

in	void	
----	------	--

Returns

SMU latent fault status.

Definition at line 315 of file AC784xx_Smu_Reg.h.

4.2.2.6 Smu_Reg_GetPathCheck0()

```
LOCAL_INLINE uint32 Smu_Reg_GetPathCheck0 (  
    void )
```

Get SMU reset patch check0.

Note

Function ID: DES_SMU_API_129

Parameters

in	void	
----	------	--

Returns

Hardware and software Single point fault channels.

Definition at line 394 of file AC784xx_Smu_Reg.h.

4.2.2.7 Smu_Reg_GetPathCheck1()

```
LOCAL_INLINE uint32 Smu_Reg_GetPathCheck1 (  
    void )
```

Get SMU reset patch check1.

Note

Function ID: DES_SMU_API_131

Parameters

in	void	
----	------	--

Returns

Hardware and software Single point fault channels.

Definition at line 416 of file AC784xx_Smu_Reg.h.

4.2.2.8 Smu_Reg_GetResetCounter()

```
LOCAL_INLINE uint8 Smu_Reg_GetResetCounter (  
    void )
```

Get SMU reset counter status.

Note

Function ID: DES_SMU_API_124

Parameters

in	void	
----	------	--

Returns

SMU reset counter.

Definition at line 337 of file AC784xx_Smu_Reg.h.

4.2.2.9 Smu_Reg_GetResetCounterShadow()

```
LOCAL_INLINE uint8 Smu_Reg_GetResetCounterShadow (  
    void )
```

Get SMU reset counter shadow status.

Note

Function ID: DES_SMU_API_126

Parameters

in	void	
----	------	--

Returns

SMU reset counter shadow

Definition at line 360 of file AC784xx_Smu_Reg.h.

4.2.2.10 Smu_Reg_GetResetCounterThreshold()

```
LOCAL_INLINE uint8 Smu_Reg_GetResetCounterThreshold (
    void )
```

Set SMU reset counter value.

Note

Function ID: DES_SMU_API_115

Parameters

in	void	
----	------	--

Returns

reset counter value which SHALL be 0~15.

Definition at line 237 of file AC784xx_Smu_Reg.h.

4.2.2.11 Smu_Reg_GetSoftwareLatentFault()

```
LOCAL_INLINE uint32 Smu_Reg_GetSoftwareLatentFault (
    void )
```

Get software latent fault channel function.

Note

Function ID: DES_SMU_API_112

Parameters

in	void	
----	------	--

Returns

software latent fault channel.

Definition at line 204 of file AC784xx_Smu_Reg.h.

4.2.2.12 Smu_Reg_GetSoftwareLatentFaultShadow()

```
LOCAL_INLINE uint32 Smu_Reg_GetSoftwareLatentFaultShadow (
    void )
```

Get software latent fault shadow channel function.

Note

Function ID: DES_SMU_API_113

Parameters

in	void	
----	------	--

Returns

software latent fault shadow channel.

Definition at line 215 of file AC784xx_Smu_Reg.h.

4.2.2.13 Smu_Reg_GetSoftwareSPFault()

```
LOCAL_INLINE uint32 Smu_Reg_GetSoftwareSPFault (  
    void )
```

Set software single point fault channel function.

Note

Function ID: DES_SMU_API_109

Parameters

in	void	
----	------	--

Returns

software single point fault channels.

Definition at line 171 of file AC784xx_Smu_Reg.h.

4.2.2.14 Smu_Reg_GetSoftwareSPFaultShadow()

```
LOCAL_INLINE uint32 Smu_Reg_GetSoftwareSPFaultShadow (  
    void )
```

Set software single point fault shadow channels function.

Note

Function ID: DES_SMU_API_110

Parameters

in	void	
----	------	--

Returns

software single point fault shadow channels.

Definition at line 182 of file AC784xx_Smu_Reg.h.

4.2.2.15 Smu_Reg_GetSPFaultResetEnable()

```
LOCAL_INLINE uint32 Smu_Reg_GetSPFaultResetEnable (  
    void )
```

Get which single point fault channel reset.

Note

Function ID: DES_SMU_API_105

Parameters

in	void	
----	------	--

Returns

single point fault channels.

Definition at line 126 of file AC784xx_Smu_Reg.h.

4.2.2.16 Smu_Reg_GetSPFaultStatus()

```
LOCAL_INLINE uint32 Smu_Reg_GetSPFaultStatus (  
    void )
```

Get SMU single point fault status.

Note

Function ID: DES_SMU_API_116

Parameters

in	void	
----	------	--

Returns

SMU single point fault status.

Definition at line 248 of file AC784xx_Smu_Reg.h.

4.2.2.17 Smu_Reg_GetSPFaultStatusShadow()

```
LOCAL_INLINE uint32 Smu_Reg_GetSPFaultStatusShadow (  
    void )
```

Get SMU single point fault status.

Note

Function ID: DES_SMU_API_118

Parameters

in	void	
----	------	--

Returns

SMU single point fault status.

Definition at line 270 of file AC784xx_Smu_Reg.h.

4.2.2.18 Smu_Reg_LatentFaultRegLock()

```
LOCAL_INLINE void Smu_Reg_LatentFaultRegLock (  
    void )
```

Latent fault register access permission lock.

Note

Function ID: DES_SMU_API_102

Parameters

in	void	
----	------	--

Returns

void

Definition at line 91 of file AC784xx_Smu_Reg.h.

4.2.2.19 Smu_Reg_LatentFaultRegUnlock()

```
LOCAL_INLINE void Smu_Reg_LatentFaultRegUnlock (
    void )
```

Latent fault register access permission unlock.

Note

Function ID: DES_SMU_API_103

Parameters

in	<i>void</i>	
----	-------------	--

Returns

void

Definition at line 102 of file AC784xx_Smu_Reg.h.

4.2.2.20 Smu_Reg_SetLatentFaultResetEnable()

```
LOCAL_INLINE void Smu_Reg_SetLatentFaultResetEnable (
    uint32 LfChannels )
```

Enable which latent fault channels could reset.

Note

Function ID: DES_SMU_API_106

Parameters

in	<i>LfChannels</i>	latent fault channels.
----	-------------------	------------------------

Returns

void

Definition at line 137 of file AC784xx_Smu_Reg.h.

4.2.2.21 Smu_Reg_SetLatentFaultStatus()

```
LOCAL_INLINE void Smu_Reg_SetLatentFaultStatus (
    uint32 LfChannels )
```

Set SMU latent fault status.

Note

Function ID: DES_SMU_API_121

Parameters

in	<i>LfChannels</i>	latent fault channels.
----	-------------------	------------------------

Returns

void

Definition at line 303 of file AC784xx_Smu_Reg.h.

4.2.2.22 Smu_Reg_SetLatentFaultStatusShadow()

```
LOCAL_INLINE void Smu_Reg_SetLatentFaultStatusShadow (
    uint32 LfChannels )
```

Set SMU latent fault status.

Note

Function ID: DES_SMU_API_123

Parameters

in	<i>LfChannels</i>	latent fault channels.
----	-------------------	------------------------

Returns

void

Definition at line 326 of file AC784xx_Smu_Reg.h.

4.2.2.23 Smu_Reg_SetPathCheck0()

```
LOCAL_INLINE void Smu_Reg_SetPathCheck0 (
    uint32 SpfChannels )
```

Set SMU patch check0.

Note

Function ID: DES_SMU_API_128

Parameters

<i>in</i>	<i>SpfChannels</i>	Hardware and software Single point fault channels.
-----------	--------------------	--

Returns

void

Definition at line 383 of file AC784xx_Smu_Reg.h.

4.2.2.24 Smu_Reg_SetPathCheck1()

```
LOCAL_INLINE void Smu_Reg_SetPathCheck1 (
    uint32 LfChannels )
```

Set SMU patch check1.

Note

Function ID: DES_SMU_API_130

Parameters

<i>in</i>	<i>LfChannels</i>	Hardware and software Single point fault channels.
-----------	-------------------	--

Returns

void

Definition at line 405 of file AC784xx_Smu_Reg.h.

4.2.2.25 Smu_Reg_SetResetCounterThreshold()

```
LOCAL_INLINE void Smu_Reg_SetResetCounterThreshold (
    uint8 Val )
```

Set SMU reset counter value.

Note

Function ID: DES_SMU_API_114

Parameters

<i>in</i>	<i>Val</i>	reset counter value which SHALL be 0~15.
-----------	------------	--

Returns

void

Definition at line 226 of file AC784xx_Smu_Reg.h.

4.2.2.26 Smu_Reg_SetSoftwareLatentFault()

```
LOCAL_INLINE void Smu_Reg_SetSoftwareLatentFault (
    uint32 SwLfChannels )
```

Set software latent fault channel function.

Note

Function ID: DES_SMU_API_111

Parameters

in	<i>SwLfChannels</i>	software latent fault channel.
----	---------------------	--------------------------------

Returns

void

Definition at line 193 of file AC784xx_Smu_Reg.h.

4.2.2.27 Smu_Reg_SetSoftwareSPFault()

```
LOCAL_INLINE void Smu_Reg_SetSoftwareSPFault (
    uint32 SwSpfChannels )
```

Set software single point fault channel function.

Note

Function ID: DES_SMU_API_108

Parameters

in	<i>SwSpfChannels</i>	software single point fault channels.
----	----------------------	---------------------------------------

Returns

void

Definition at line 159 of file AC784xx_Smu_Reg.h.

4.2.2.28 Smu_Reg_SetSPFaultResetEnable()

```
LOCAL_INLINE void Smu_Reg_SetSPFaultResetEnable (
    uint32 SpfChannels )
```

Enable which single point fault channels could trigger reset.

Note

Function ID: DES_SMU_API_104

Parameters

in	<i>SpfChannels</i>	single fault channels.
----	--------------------	------------------------

Returns

void

Definition at line 115 of file AC784xx_Smu_Reg.h.

4.2.2.29 Smu_Reg_SetSPFaultStatus()

```
LOCAL_INLINE void Smu_Reg_SetSPFaultStatus (
    uint32 SpfChannels )
```

Set SMU single point fault status.

Note

Function ID: DES_SMU_API_117

Parameters

in	<i>SpfChannels</i>	single fault channels.
----	--------------------	------------------------

Returns

void

Definition at line 259 of file AC784xx_Smu_Reg.h.

4.2.2.30 Smu_Reg_SetSPFaultStatusShadow()

```
LOCAL_INLINE void Smu_Reg_SetSPFaultStatusShadow (
    uint32 SpfChannels )
```

Set SMU single point fault status.

Note

Function ID: DES_SMU_API_119

Parameters

in	<i>SpfChannels</i>	single fault channels.
----	--------------------	------------------------

Returns

void

Definition at line 281 of file AC784xx_Smu_Reg.h.

4.2.2.31 Smu_Reg_SPFaultRegLock()

```
LOCAL_INLINE void Smu_Reg_SPFaultRegLock (  
    void )
```

Single point fault register access permission lock.

Note

Function ID: DES_SMU_API_100

Parameters

in	<i>void</i>	
----	-------------	--

Returns

void

Definition at line 68 of file AC784xx_Smu_Reg.h.

4.2.2.32 Smu_Reg_SPFaultRegUnlock()

```
LOCAL_INLINE void Smu_Reg_SPFaultRegUnlock (  
    void )
```

Single point fault register access permission unlock.

Note

Function ID: DES_SMU_API_101

Parameters

in	void	
----	------	--

Returns

void

Definition at line 79 of file AC784xx_Smu_Reg.h.

4.3 Smu_Hal.c File Reference

This file provides Hal Smu api.

```
#include "AC784xx_Smu_Reg.h"
#include "Smu_Hal.h"
#include "Ckgen_Hal.h"
```

Functions

- void [Smu_Hal_Lock](#) (void)
Lock SMU registers write permission.
- void [Smu_Hal_Unlock](#) (void)
Unlock SMU registers write permission.
- Hal_StatusType [Smu_Hal_Init](#) (const [Smu_ConfigType](#) *ConfigPtr)
Init SMU driver.
- void [Smu_Hal_DeInit](#) (void)
De-Initialize SMU module.
- Hal_StatusType [Smu_Hal_GetCurrentFaults](#) ([Smu_FaultInfoType](#) *FaultPtr)
Get current SMU count, single point fault, latent fault values.
- Hal_StatusType [Smu_Hal_GetLastFaults](#) ([Smu_FaultInfoType](#) *FaultPtr)
Get last SMU count, single point fault, latent fault values.
- void [Smu_Hal_ClearCurrentFaultInfo](#) (boolean Reset)
Clear current SMU count, single point fault, latent fault values.
- Hal_StatusType [Smu_Hal_AssertSwFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Assert a software single point or latent fault to SMU.
- Hal_StatusType [Smu_Hal_DeassertSwFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Deassert a software single point or latent fault to SMU.
- Hal_StatusType [Smu_Hal_InjectFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Inject a hardware fault to SMU.
- Hal_StatusType [Smu_Hal_ClearFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Clear a hardware fault to SMU.

4.3.1 Detailed Description

This file provides Hal Smu api.

4.3.2 Function Documentation

4.3.2.1 Smu_Hal_AssertSwFault()

```
Hal_StatusType Smu_Hal_AssertSwFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Assert a software single point or latent fault to SMU.

Note

Function ID: DES_SMU_API_008

Parameters

in	<i>FaultType</i>	Fault type, software single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 320 of file Smu_Hal.c.

4.3.2.2 Smu_Hal_ClearCurrentFaultInfo()

```
void Smu_Hal_ClearCurrentFaultInfo (
    boolean Reset )
```

Clear current SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_007

Parameters

in	<i>Reset</i>	Reset SMU count to zero.
----	--------------	--------------------------

Returns

void

Definition at line 290 of file Smu_Hal.c.

4.3.2.3 Smu_Hal_ClearFault()

```
Hal_StatusType Smu_Hal_ClearFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Clear a hardware fault to SMU.

Note

Function ID: DES_SMU_API_011

Parameters

in	<i>FaultType</i>	Fault type, Hardware single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 473 of file Smu_Hal.c.

4.3.2.4 Smu_Hal_DeassertSwFault()

```
Hal_StatusType Smu_Hal_DeassertSwFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Deassert a software single point or latent fault to SMU.

Note

Function ID: DES_SMU_API_009

Parameters

in	<i>FaultType</i>	Fault type, software single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 371 of file Smu_Hal.c.

4.3.2.5 Smu_Hal_DeInit()

```
void Smu_Hal_DeInit (
    void )
```

De-Initialize SMU module.

Note

Function ID: DES_SMU_API_004

Parameters

in	void.	
----	-------	--

Returns

void

Definition at line 164 of file Smu_Hal.c.

4.3.2.6 Smu_Hal_GetCurrentFaults()

```
Hal_StatusType Smu_Hal_GetCurrentFaults (
    Smu_FaultInfoType * FaultPtr )
```

Get current SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_005

Parameters

out	<i>FaultPtr</i>	pointer of current SMU status.
-----	-----------------	--------------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR↵ : command has not been accepted e.g. due to parameter error. -STATUS_SMU_FAULT_OCCURRED: Fault has been detected in current.

Definition at line 199 of file Smu_Hal.c.

4.3.2.7 Smu_Hal_GetLastFaults()

```
Hal_StatusType Smu_Hal_GetLastFaults (
    Smu_FaultInfoType * FaultPtr )
```

Get last SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_006

Parameters

out	<i>FaultPtr</i>	pointer of last SMU status.
-----	-----------------	-----------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR↵ : command has not been accepted e.g. due to parameter error. -STATUS_SMU_FAULT_OCCURRED: Fault has been detected at last boot.

Definition at line 242 of file Smu_Hal.c.

4.3.2.8 Smu_Hal_Init()

```
Hal_StatusType Smu_Hal_Init (
    const Smu_ConfigType * ConfigPtr )
```

Init SMU driver.

Note

Function ID: DES_SMU_API_003

Parameters

in	<i>ConfigPtr</i>	point of SMU driver configuration.
----	------------------	------------------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 108 of file Smu_Hal.c.

4.3.2.9 Smu_Hal_InjectFault()

```
Hal_StatusType Smu_Hal_InjectFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Inject a hardware fault to SMU.

Note

Function ID: DES_SMU_API_010

Parameters

in	<i>FaultType</i>	Fault type, Hardware single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 434 of file Smu_Hal.c.

4.3.2.10 Smu_Hal_Lock()

```
void Smu_Hal_Lock (  
    void )
```

Lock SMU registers write permission.

Note

Function ID: DES_SMU_API_001

Parameters

in	<i>void</i>	
----	-------------	--

Returns

void

Definition at line 76 of file Smu_Hal.c.

4.3.2.11 Smu_Hal_Unlock()

```
void Smu_Hal_Unlock (  
    void )
```

Unlock SMU registers write permission.

Note

Function ID: DES_SMU_API_002

Parameters

in	<i>void</i>	
----	-------------	--

Returns

void

Definition at line 91 of file Smu_Hal.c.

4.4 Smu_Hal.h File Reference

This file provides extern Hal Smu api.

```
#include "Smu_Hal_Types.h"
```

Functions

- void [Smu_Hal_Lock](#) (void)
Lock SMU registers write permission.
- void [Smu_Hal_Unlock](#) (void)
Unlock SMU registers write permission.
- Hal_StatusType [Smu_Hal_Init](#) (const [Smu_ConfigType](#) *ConfigPtr)
Init SMU driver.
- void [Smu_Hal_DeInit](#) (void)
De-Initialize SMU module.
- Hal_StatusType [Smu_Hal_GetCurrentFaults](#) ([Smu_FaultInfoType](#) *FaultPtr)
Get current SMU count, single point fault, latent fault values.
- Hal_StatusType [Smu_Hal_GetLastFaults](#) ([Smu_FaultInfoType](#) *FaultPtr)
Get last SMU count, single point fault, latent fault values.
- void [Smu_Hal_ClearCurrentFaultInfo](#) (boolean Reset)
Clear current SMU count, single point fault, latent fault values.
- Hal_StatusType [Smu_Hal_AssertSwFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Assert a software single point or latent fault to SMU.
- Hal_StatusType [Smu_Hal_DeassertSwFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Deassert a software single point or latent fault to SMU.
- Hal_StatusType [Smu_Hal_InjectFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Inject a hardware fault to SMU.
- Hal_StatusType [Smu_Hal_ClearFault](#) ([Smu_FaultType](#) FaultType, uint8 FaultId)
Clear a hardware fault to SMU.

4.4.1 Detailed Description

This file provides extern Hal Smu api.

4.4.2 Function Documentation

4.4.2.1 Smu_Hal_AssertSwFault()

```
Hal_StatusType Smu_Hal_AssertSwFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Assert a software single point or latent fault to SMU.

Note

Function ID: DES_SMU_API_008

Parameters

in	<i>FaultType</i>	Fault type, software single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 320 of file Smu_Hal.c.

4.4.2.2 Smu_Hal_ClearCurrentFaultInfo()

```
void Smu_Hal_ClearCurrentFaultInfo (
    boolean Reset )
```

Clear current SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_007

Parameters

in	<i>Reset</i>	Reset SMU count to zero.
----	--------------	--------------------------

Returns

void

Definition at line 290 of file Smu_Hal.c.

4.4.2.3 Smu_Hal_ClearFault()

```
Hal_StatusType Smu_Hal_ClearFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Clear a hardware fault to SMU.

Note

Function ID: DES_SMU_API_011

Parameters

in	<i>FaultType</i>	Fault type, Hardware single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 473 of file Smu_Hal.c.

4.4.2.4 Smu_Hal_DeassertSwFault()

```
Hal_StatusType Smu_Hal_DeassertSwFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Deassert a software single point or latent fault to SMU.

Note

Function ID: DES_SMU_API_009

Parameters

in	<i>FaultType</i>	Fault type, software single point or latent fault.
in	<i>FaultId</i>	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 371 of file Smu_Hal.c.

4.4.2.5 Smu_Hal_DeInit()

```
void Smu_Hal_DeInit (
    void )
```

De-Initialize SMU module.

Note

Function ID: DES_SMU_API_004

Parameters

in	void.	
----	-------	--

Returns

void

Definition at line 164 of file Smu_Hal.c.

4.4.2.6 Smu_Hal_GetCurrentFaults()

```
Hal_StatusType Smu_Hal_GetCurrentFaults (
    Smu_FaultInfoType * FaultPtr )
```

Get current SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_005

Parameters

out	<i>FaultPtr</i>	pointer of current SMU status.
-----	-----------------	--------------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR↵: command has not been accepted e.g. due to parameter error. -STATUS_SMU_FAULT_OCCURRED: Fault has been detected in current.

Definition at line 199 of file Smu_Hal.c.

4.4.2.7 Smu_Hal_GetLastFaults()

```
Hal_StatusType Smu_Hal_GetLastFaults (
    Smu_FaultInfoType * FaultPtr )
```

Get last SMU count, single point fault, latent fault values.

Note

Function ID: DES_SMU_API_006

Parameters

out	<i>FaultPtr</i>	pointer of last SMU status.
-----	-----------------	-----------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR↵ : command has not been accepted e.g. due to parameter error. -STATUS_SMU_FAULT_OCCURRED: Fault has been detected at last boot.

Definition at line 242 of file Smu_Hal.c.

4.4.2.8 Smu_Hal_Init()

```
Hal_StatusType Smu_Hal_Init (
    const Smu_ConfigType * ConfigPtr )
```

Init SMU driver.

Note

Function ID: DES_SMU_API_003

Parameters

in	ConfigPtr	point of SMU driver configuration.
----	-----------	------------------------------------

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 108 of file Smu_Hal.c.

4.4.2.9 Smu_Hal_InjectFault()

```
Hal_StatusType Smu_Hal_InjectFault (
    Smu_FaultType FaultType,
    uint8 FaultId )
```

Inject a hardware fault to SMU.

Note

Function ID: DES_SMU_API_010

Parameters

in	FaultType	Fault type, Hardware single point or latent fault.
in	FaultId	Fault ID.

Returns

Command has been accepted or not. -STATUS_SUCCESS: command has been accepted. -STATUS_ERROR: command has not been accepted e.g. due to parameter error.

Definition at line 434 of file Smu_Hal.c.

4.4.2.10 Smu_Hal_Lock()

```
void Smu_Hal_Lock (  
    void )
```

Lock SMU registers write permission.

Note

Function ID: DES_SMU_API_001

Parameters

in	void	
----	------	--

Returns

void

Definition at line 76 of file Smu_Hal.c.

4.4.2.11 Smu_Hal_Unlock()

```
void Smu_Hal_Unlock (  
    void )
```

Unlock SMU registers write permission.

Note

Function ID: DES_SMU_API_002

Parameters

in	void	
----	------	--

Returns

void

Definition at line 91 of file Smu_Hal.c.

4.5 Smu_Hal_Types.h File Reference

This file provides Smu hal types header.

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

Classes

- struct [Smu_ConfigType](#)
SMU module configuration structure.
- struct [Smu_FaultInfoType](#)
SMU module information structure.

4.5.1 Detailed Description

This file provides Smu hal types header.

Index

AC784xx_API_Reference_Manual_SMU.pdf, 6
AC784xx_Smu_Reg.h, 6
 Smu_Reg_ClearResetCounter, 7
 Smu_Reg_ClearResetCounterShadow, 8
 Smu_Reg_GetLatentFaultResetEnable, 8
 Smu_Reg_GetLatentFaultStatus, 9
 Smu_Reg_GetLatentFaultStatusShadow, 9
 Smu_Reg_GetPathCheck0, 10
 Smu_Reg_GetPathCheck1, 10
 Smu_Reg_GetResetCounter, 11
 Smu_Reg_GetResetCounterShadow, 11
 Smu_Reg_GetResetCounterThreshold, 11
 Smu_Reg_GetSPFaultResetEnable, 14
 Smu_Reg_GetSPFaultStatus, 14
 Smu_Reg_GetSPFaultStatusShadow, 15
 Smu_Reg_GetSoftwareLatentFault, 12
 Smu_Reg_GetSoftwareLatentFaultShadow, 12
 Smu_Reg_GetSoftwareSPFault, 13
 Smu_Reg_GetSoftwareSPFaultShadow, 13
 Smu_Reg_LatentFaultRegLock, 15
 Smu_Reg_LatentFaultRegUnlock, 15
 Smu_Reg_SPFaultRegLock, 21
 Smu_Reg_SPFaultRegUnlock, 21
 Smu_Reg_SetLatentFaultResetEnable, 16
 Smu_Reg_SetLatentFaultStatus, 16
 Smu_Reg_SetLatentFaultStatusShadow, 17
 Smu_Reg_SetPathCheck0, 17
 Smu_Reg_SetPathCheck1, 18
 Smu_Reg_SetResetCounterThreshold, 18
 Smu_Reg_SetSPFaultResetEnable, 19
 Smu_Reg_SetSPFaultStatus, 20
 Smu_Reg_SetSPFaultStatusShadow, 20
 Smu_Reg_SetSoftwareLatentFault, 19
 Smu_Reg_SetSoftwareSPFault, 19

Count
 Smu_FaultInfoType, 4

Lfs
 Smu_FaultInfoType, 5

Reset_Count
 Smu_ConfigType, 3

Reset_Lfs
 Smu_ConfigType, 3

Reset_Spfs
 Smu_ConfigType, 4

Smu_ConfigType, 3
 Reset_Count, 3
 Reset_Lfs, 3
 Reset_Spfs, 4

Smu_FaultInfoType, 4

Count, 4
Lfs, 5
Spfs, 5
SwLfs, 5
SwSpfs, 5
Smu_Hal.c, 22
 Smu_Hal_AssertSwFault, 23
 Smu_Hal_ClearCurrentFaultInfo, 23
 Smu_Hal_ClearFault, 23
 Smu_Hal_DeInit, 24
 Smu_Hal_DeassertSwFault, 24
 Smu_Hal_GetCurrentFaults, 25
 Smu_Hal_GetLastFaults, 25
 Smu_Hal_Init, 26
 Smu_Hal_InjectFault, 26
 Smu_Hal_Lock, 27
 Smu_Hal_Unlock, 27
Smu_Hal.h, 28
 Smu_Hal_AssertSwFault, 28
 Smu_Hal_ClearCurrentFaultInfo, 29
 Smu_Hal_ClearFault, 29
 Smu_Hal_DeInit, 30
 Smu_Hal_DeassertSwFault, 30
 Smu_Hal_GetCurrentFaults, 31
 Smu_Hal_GetLastFaults, 31
 Smu_Hal_Init, 32
 Smu_Hal_InjectFault, 32
 Smu_Hal_Lock, 33
 Smu_Hal_Unlock, 33
Smu_Hal_AssertSwFault
 Smu_Hal.c, 23
 Smu_Hal.h, 28
Smu_Hal_ClearCurrentFaultInfo
 Smu_Hal.c, 23
 Smu_Hal.h, 29
Smu_Hal_ClearFault
 Smu_Hal.c, 23
 Smu_Hal.h, 29
Smu_Hal_DeInit
 Smu_Hal.c, 24
 Smu_Hal.h, 30
Smu_Hal_DeassertSwFault
 Smu_Hal.c, 24
 Smu_Hal.h, 30
Smu_Hal_GetCurrentFaults
 Smu_Hal.c, 25
 Smu_Hal.h, 31
Smu_Hal_GetLastFaults
 Smu_Hal.c, 25
 Smu_Hal.h, 31
Smu_Hal_Init
 Smu_Hal.c, 26

Smu_Hal.h, [32](#)
Smu_Hal_InjectFault
 Smu_Hal.c, [26](#)
 Smu_Hal.h, [32](#)
Smu_Hal_Lock
 Smu_Hal.c, [27](#)
 Smu_Hal.h, [33](#)
Smu_Hal_Types.h, [34](#)
Smu_Hal_Unlock
 Smu_Hal.c, [27](#)
 Smu_Hal.h, [33](#)
Smu_Reg_ClearResetCounter
 AC784xx_Smu_Reg.h, [7](#)
Smu_Reg_ClearResetCounterShadow
 AC784xx_Smu_Reg.h, [8](#)
Smu_Reg_GetLatentFaultResetEnable
 AC784xx_Smu_Reg.h, [8](#)
Smu_Reg_GetLatentFaultStatus
 AC784xx_Smu_Reg.h, [9](#)
Smu_Reg_GetLatentFaultStatusShadow
 AC784xx_Smu_Reg.h, [9](#)
Smu_Reg_GetPathCheck0
 AC784xx_Smu_Reg.h, [10](#)
Smu_Reg_GetPathCheck1
 AC784xx_Smu_Reg.h, [10](#)
Smu_Reg_GetResetCounter
 AC784xx_Smu_Reg.h, [11](#)
Smu_Reg_GetResetCounterShadow
 AC784xx_Smu_Reg.h, [11](#)
Smu_Reg_GetResetCounterThreshold
 AC784xx_Smu_Reg.h, [11](#)
Smu_Reg_GetSPFaultResetEnable
 AC784xx_Smu_Reg.h, [14](#)
Smu_Reg_GetSPFaultStatus
 AC784xx_Smu_Reg.h, [14](#)
Smu_Reg_GetSPFaultStatusShadow
 AC784xx_Smu_Reg.h, [15](#)
Smu_Reg_GetSoftwareLatentFault
 AC784xx_Smu_Reg.h, [12](#)
Smu_Reg_GetSoftwareLatentFaultShadow
 AC784xx_Smu_Reg.h, [12](#)
Smu_Reg_GetSoftwareSPFault
 AC784xx_Smu_Reg.h, [13](#)
Smu_Reg_GetSoftwareSPFaultShadow
 AC784xx_Smu_Reg.h, [13](#)
Smu_Reg_LatentFaultRegLock
 AC784xx_Smu_Reg.h, [15](#)
Smu_Reg_LatentFaultRegUnlock
 AC784xx_Smu_Reg.h, [15](#)
Smu_Reg_SPFaultRegLock
 AC784xx_Smu_Reg.h, [21](#)
Smu_Reg_SPFaultRegUnlock
 AC784xx_Smu_Reg.h, [21](#)
Smu_Reg_SetLatentFaultResetEnable
 AC784xx_Smu_Reg.h, [16](#)
Smu_Reg_SetLatentFaultStatus
 AC784xx_Smu_Reg.h, [16](#)
Smu_Reg_SetLatentFaultStatusShadow
 AC784xx_Smu_Reg.h, [17](#)
Smu_Reg_SetPathCheck0
 AC784xx_Smu_Reg.h, [17](#)
Smu_Reg_SetPathCheck1
 AC784xx_Smu_Reg.h, [18](#)
Smu_Reg_SetResetCounterThreshold
 AC784xx_Smu_Reg.h, [18](#)
Smu_Reg_SetSPFaultResetEnable
 AC784xx_Smu_Reg.h, [19](#)
Smu_Reg_SetSPFaultStatus
 AC784xx_Smu_Reg.h, [20](#)
Smu_Reg_SetSPFaultStatusShadow
 AC784xx_Smu_Reg.h, [20](#)
Smu_Reg_SetSoftwareLatentFault
 AC784xx_Smu_Reg.h, [19](#)
Smu_Reg_SetSoftwareSPFault
 AC784xx_Smu_Reg.h, [19](#)
Spfs
 Smu_FaultInfoType, [5](#)
SwLfs
 Smu_FaultInfoType, [5](#)
SwSpfs
 Smu_FaultInfoType, [5](#)