

# AC784xx\_DFP ACMP

## 6.1.0

Generated by Doxygen 1.8.13

# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>2</b>
2.1	File List . . . . .	2
<b>3</b>	<b>Class Documentation</b>	<b>3</b>
3.1	Acmp_AnmuxType Struct Reference . . . . .	3
3.1.1	Detailed Description . . . . .	3
3.1.2	Member Data Documentation . . . . .	3
3.1.2.1	NegativeInputMux . . . . .	3
3.1.2.2	PositiveInputMux . . . . .	4
3.2	Acmp_ComparatorType Struct Reference . . . . .	4
3.2.1	Detailed Description . . . . .	4
3.2.2	Member Data Documentation . . . . .	4
3.2.2.1	Callback . . . . .	5
3.2.2.2	ClockDivide . . . . .	5
3.2.2.3	FilterEnable . . . . .	5
3.2.2.4	FilterSampleCount . . . . .	5
3.2.2.5	HysteresisLevel . . . . .	5
3.2.2.6	HysteresisMode . . . . .	6
3.2.2.7	InterruptEn . . . . .	6
3.2.2.8	InverterEnable . . . . .	6
3.2.2.9	LpfBandwidth . . . . .	6
3.2.2.10	OutputSelect . . . . .	6

3.2.2.11	OutputTrigger	7
3.2.2.12	PinState	7
3.2.2.13	PowerEnable	7
3.2.2.14	UsingLSIEnable	7
3.2.2.15	WindowModeEnable	7
3.3	Acmp_DacType Struct Reference	8
3.3.1	Detailed Description	8
3.3.2	Member Data Documentation	8
3.3.2.1	State	8
3.3.2.2	Voltage	8
3.3.2.3	VoltageReferenceSource	8
3.4	Acmp_ModuleType Struct Reference	9
3.4.1	Detailed Description	9
3.4.2	Member Data Documentation	9
3.4.2.1	Comparator	9
3.4.2.2	Dac	9
3.4.2.3	Mux	9
3.4.2.4	PollingMode	10
3.5	Acmp_PollingModeType Struct Reference	10
3.5.1	Detailed Description	10
3.5.2	Member Data Documentation	10
3.5.2.1	HallAOutputCh	10
3.5.2.2	HallBOutputCh	11
3.5.2.3	HallCOutputCh	11
3.5.2.4	HallOutputEnable	11
3.5.2.5	Mode	11
3.5.2.6	PollingClockDivide	11
3.5.2.7	PollingSequence	11

<b>4 File Documentation</b>	<b>12</b>
4.1 AC784xx_Acmp_Reg.h File Reference	12
4.1.1 Function Documentation	14
4.1.1.1 Acmp_Reg_ClearInterruptStatus()	15
4.1.1.2 Acmp_Reg_ClearOutputStatus()	15
4.1.1.3 Acmp_Reg_ClearPollingStatus()	15
4.1.1.4 Acmp_Reg_GetBase()	16
4.1.1.5 Acmp_Reg_GetDACEnableFlag()	16
4.1.1.6 Acmp_Reg_GetDACOutput()	16
4.1.1.7 Acmp_Reg_GetDACReference()	17
4.1.1.8 Acmp_Reg_GetEnableFlag()	17
4.1.1.9 Acmp_Reg_GetFilterClockPrescaler()	18
4.1.1.10 Acmp_Reg_GetFilterEnableFlag()	18
4.1.1.11 Acmp_Reg_GetFilterLength()	18
4.1.1.12 Acmp_Reg_GetHallOutputA()	19
4.1.1.13 Acmp_Reg_GetHallOutputB()	19
4.1.1.14 Acmp_Reg_GetHallOutputC()	19
4.1.1.15 Acmp_Reg_GetHallOutputEnableFlag()	20
4.1.1.16 Acmp_Reg_GetHysteresisMode()	20
4.1.1.17 Acmp_Reg_GetHysteresisVoltage()	21
4.1.1.18 Acmp_Reg_GetInterruptEnableFlag()	21
4.1.1.19 Acmp_Reg_GetInterruptStatus()	21
4.1.1.20 Acmp_Reg_GetInvertMode()	22
4.1.1.21 Acmp_Reg_GetLowPassFilter()	22
4.1.1.22 Acmp_Reg_GetNegativeInputPin()	22
4.1.1.23 Acmp_Reg_GetNegativePollingEnableFlag()	23
4.1.1.24 Acmp_Reg_GetOutputData()	23
4.1.1.25 Acmp_Reg_GetOutputPinEnableFlag()	24
4.1.1.26 Acmp_Reg_GetOutputSelection()	24
4.1.1.27 Acmp_Reg_GetOutputStatus()	24
4.1.1.28 Acmp_Reg_GetPollingData()	25

4.1.1.29	<a href="#">Acmp_Reg_GetPollingFreqDiv()</a>	25
4.1.1.30	<a href="#">Acmp_Reg_GetPollingSequence()</a>	25
4.1.1.31	<a href="#">Acmp_Reg_GetPollingStatus()</a>	26
4.1.1.32	<a href="#">Acmp_Reg_GetPositiveInputPin()</a>	26
4.1.1.33	<a href="#">Acmp_Reg_GetPositivePollingEnableFlag()</a>	27
4.1.1.34	<a href="#">Acmp_Reg_GetTriggerMode()</a>	27
4.1.1.35	<a href="#">Acmp_Reg_GetUsingLSIEnableFlag()</a>	27
4.1.1.36	<a href="#">Acmp_Reg_GetWindowMode()</a>	28
4.1.1.37	<a href="#">Acmp_Reg_SetDACEnableFlag()</a>	28
4.1.1.38	<a href="#">Acmp_Reg_SetDACOutput()</a>	28
4.1.1.39	<a href="#">Acmp_Reg_SetDACReference()</a>	30
4.1.1.40	<a href="#">Acmp_Reg_SetEnableFlag()</a>	30
4.1.1.41	<a href="#">Acmp_Reg_SetFilterClockPrescaler()</a>	31
4.1.1.42	<a href="#">Acmp_Reg_SetFilterEnableFlag()</a>	31
4.1.1.43	<a href="#">Acmp_Reg_SetFilterLength()</a>	31
4.1.1.44	<a href="#">Acmp_Reg_SetHallOutputA()</a>	32
4.1.1.45	<a href="#">Acmp_Reg_SetHallOutputB()</a>	32
4.1.1.46	<a href="#">Acmp_Reg_SetHallOutputC()</a>	33
4.1.1.47	<a href="#">Acmp_Reg_SetHallOutputEnableFlag()</a>	33
4.1.1.48	<a href="#">Acmp_Reg_SetHysteresisMode()</a>	33
4.1.1.49	<a href="#">Acmp_Reg_SetHysteresisVoltage()</a>	34
4.1.1.50	<a href="#">Acmp_Reg_SetInterruptEnableFlag()</a>	34
4.1.1.51	<a href="#">Acmp_Reg_SetInvertMode()</a>	35
4.1.1.52	<a href="#">Acmp_Reg_SetLowPassFilter()</a>	35
4.1.1.53	<a href="#">Acmp_Reg_SetNegativeInputPin()</a>	35
4.1.1.54	<a href="#">Acmp_Reg_SetNegativePollingEnableFlag()</a>	36
4.1.1.55	<a href="#">Acmp_Reg_SetOutputPinEnableFlag()</a>	36
4.1.1.56	<a href="#">Acmp_Reg_SetOutputSelection()</a>	37
4.1.1.57	<a href="#">Acmp_Reg_SetPollingFreqDiv()</a>	37
4.1.1.58	<a href="#">Acmp_Reg_SetPollingSequence()</a>	38
4.1.1.59	<a href="#">Acmp_Reg_SetPositiveInputPin()</a>	38

4.1.1.60	<a href="#">Acmp_Reg_SetPositivePollingEnableFlag()</a>	38
4.1.1.61	<a href="#">Acmp_Reg_SetTriggerMode()</a>	39
4.1.1.62	<a href="#">Acmp_Reg_SetUsingLSIEnableFlag()</a>	39
4.1.1.63	<a href="#">Acmp_Reg_SetWindowMode()</a>	40
4.2	<a href="#">AC784xx_API_Reference_Manual_ACMP.pdf File Reference</a>	40
4.3	<a href="#">Acmp_Hal.c File Reference</a>	40
4.3.1	Detailed Description	41
4.3.2	Function Documentation	41
4.3.2.1	<a href="#">Acmp_Hal_ClearInputFlags()</a>	41
4.3.2.2	<a href="#">Acmp_Hal_ClearOutputFlags()</a>	41
4.3.2.3	<a href="#">Acmp_Hal_Deinit()</a>	42
4.3.2.4	<a href="#">Acmp_Hal_GetConfigAll()</a>	42
4.3.2.5	<a href="#">Acmp_Hal_GetDefaultConfig()</a>	43
4.3.2.6	<a href="#">Acmp_Hal_GetOutputData()</a>	43
4.3.2.7	<a href="#">Acmp_Hal_GetOutputFlags()</a>	44
4.3.2.8	<a href="#">Acmp_Hal_GetPollingData()</a>	44
4.3.2.9	<a href="#">Acmp_Hal_GetPollingFlags()</a>	45
4.3.2.10	<a href="#">Acmp_Hal_Init()</a>	45
4.3.2.11	<a href="#">Acmp_Hal_Reset()</a>	45
4.3.2.12	<a href="#">ISR()</a>	46
4.4	<a href="#">Acmp_Hal.h File Reference</a>	46
4.4.1	Detailed Description	47
4.4.2	Function Documentation	47
4.4.2.1	<a href="#">Acmp_Hal_ClearInputFlags()</a>	47
4.4.2.2	<a href="#">Acmp_Hal_ClearOutputFlags()</a>	48
4.4.2.3	<a href="#">Acmp_Hal_Deinit()</a>	48
4.4.2.4	<a href="#">Acmp_Hal_GetConfigAll()</a>	49
4.4.2.5	<a href="#">Acmp_Hal_GetDefaultConfig()</a>	49
4.4.2.6	<a href="#">Acmp_Hal_GetOutputData()</a>	50
4.4.2.7	<a href="#">Acmp_Hal_GetOutputFlags()</a>	50
4.4.2.8	<a href="#">Acmp_Hal_GetPollingData()</a>	50

4.4.2.9	<a href="#">Acmp_Hal_GetPollingFlags()</a>	51
4.4.2.10	<a href="#">Acmp_Hal_Init()</a>	51
4.4.2.11	<a href="#">Acmp_Hal_Reset()</a>	52
4.5	<a href="#">Acmp_Hal_Types.h File Reference</a>	52
4.5.1	<a href="#">Detailed Description</a>	53
4.5.2	<a href="#">Typedef Documentation</a>	54
4.5.2.1	<a href="#">Acmp_CallbackType</a>	54
4.5.3	<a href="#">Enumeration Type Documentation</a>	54
4.5.3.1	<a href="#">Acmp_ChNumberType</a>	54
4.5.3.2	<a href="#">Acmp_FilterDivideType</a>	54
4.5.3.3	<a href="#">Acmp_HysteresisModeType</a>	55
4.5.3.4	<a href="#">Acmp_HysteresisType</a>	55
4.5.3.5	<a href="#">Acmp_InputPollingType</a>	56
4.5.3.6	<a href="#">Acmp_LowPassFilterType</a>	56
4.5.3.7	<a href="#">Acmp_OutputSelectType</a>	56
4.5.3.8	<a href="#">Acmp_OutputTriggerType</a>	56
4.5.3.9	<a href="#">Acmp_PollingClkDivideType</a>	57
4.5.3.10	<a href="#">Acmp_VoltageReferenceType</a>	57
<b>Index</b>		<b>58</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Acmp_AnmuxType</a>		
ACMP analog mux input configuration type	. . . . .	3
<a href="#">Acmp_ComparatorType</a>		
ACMP comparator configuration type	. . . . .	4
<a href="#">Acmp_DacType</a>		
ACMP DAC configuration type	. . . . .	8
<a href="#">Acmp_ModuleType</a>		
ACMP module configuration type	. . . . .	9
<a href="#">Acmp_PollingModeType</a>		
ACMP polling mode configuration	. . . . .	10



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">AC784xx_Acmp_Reg.h</a>	12
<a href="#">AC784xx_API_Reference_Manual_ACMP.pdf</a>	40
<a href="#">Acmp_Hal.c</a>	
This file provides analog comparator module integration functions	40
<a href="#">Acmp_Hal.h</a>	
This file provides analog comparator module integration functions interfaces	46
<a href="#">Acmp_Hal_Types.h</a>	
This file provides ACMP config	52

## Chapter 3

# Class Documentation

### 3.1 Acmp\_AnmuxType Struct Reference

ACMP analog mux input configuration type.

```
#include <Acmp_Hal_Types.h>
```

#### Public Attributes

- [Acmp\\_ChNumberType PositiveInputMux](#)
- [Acmp\\_ChNumberType NegativeInputMux](#)

#### 3.1.1 Detailed Description

ACMP analog mux input configuration type.

Definition at line 201 of file Acmp\_Hal\_Types.h.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 NegativeInputMux

[Acmp\\_ChNumberType](#) Acmp\_AnmuxType::NegativeInputMux

Negative input channel selection

Definition at line 204 of file Acmp\_Hal\_Types.h.

### 3.1.2.2 PositiveInputMux

[Acmp\\_ChNumberType](#) `Acmp_AnmuxType::PositiveInputMux`

Positive input channel selection

Definition at line 203 of file `Acmp_Hal_Types.h`.

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

- [Acmp\\_Hal\\_Types.h](#)

## 3.2 Acmp\_ComparatorType Struct Reference

ACMP comparator configuration type.

```
#include <Acmp_Hal_Types.h>
```

### Public Attributes

- boolean [InterruptEn](#)
- boolean [FilterEnable](#)
- boolean [InverterEnable](#)
- boolean [PinState](#)
- boolean [UsingLSIEnable](#)
- boolean [WindowModeEnable](#)
- boolean [PowerEnable](#)
- uint8 [FilterSampleCount](#)
- [Acmp\\_OutputTriggerType](#) `OutputTrigger`
- [Acmp\\_OutputSelectType](#) `OutputSelect`
- [Acmp\\_FilterDivideType](#) `ClockDivide`
- [Acmp\\_LowPassFilterType](#) `LpfBandwidth`
- [Acmp\\_HysteresisModeType](#) `HysteresisMode`
- [Acmp\\_HysteresisType](#) `HysteresisLevel`
- [Acmp\\_CallbackType](#) `Callback`

### 3.2.1 Detailed Description

ACMP comparator configuration type.

Definition at line 179 of file `Acmp_Hal_Types.h`.

### 3.2.2 Member Data Documentation

### 3.2.2.1 Callback

`Acmp_CallbackType` `Acmp_ComparatorType::Callback`

ACMP interrupt callback function pointer

Definition at line 195 of file `Acmp_Hal_Types.h`.

### 3.2.2.2 ClockDivide

`Acmp_FilterDivideType` `Acmp_ComparatorType::ClockDivide`

Digital filter clock divide

Definition at line 191 of file `Acmp_Hal_Types.h`.

### 3.2.2.3 FilterEnable

`boolean` `Acmp_ComparatorType::FilterEnable`

Digital filter enable flag

Definition at line 182 of file `Acmp_Hal_Types.h`.

### 3.2.2.4 FilterSampleCount

`uint8` `Acmp_ComparatorType::FilterSampleCount`

Digital filter sample count

Definition at line 188 of file `Acmp_Hal_Types.h`.

### 3.2.2.5 HysteresisLevel

`Acmp_HysteresisType` `Acmp_ComparatorType::HysteresisLevel`

Hysteresis level selection

Definition at line 194 of file `Acmp_Hal_Types.h`.

### 3.2.2.6 HysteresisMode

`Acmp_HysteresisModeType` `Acmp_ComparatorType::HysteresisMode`

Hysteresis mode

Definition at line 193 of file `Acmp_Hal_Types.h`.

### 3.2.2.7 InterruptEn

`boolean` `Acmp_ComparatorType::InterruptEn`

Interrupt enable flag

Definition at line 181 of file `Acmp_Hal_Types.h`.

### 3.2.2.8 InverterEnable

`boolean` `Acmp_ComparatorType::InverterEnable`

ACMP result inversion enable flag

Definition at line 183 of file `Acmp_Hal_Types.h`.

### 3.2.2.9 LpfBandwidth

`Acmp_LowPassFilterType` `Acmp_ComparatorType::LpfBandwidth`

Low pass filter bandwidth

Definition at line 192 of file `Acmp_Hal_Types.h`.

### 3.2.2.10 OutputSelect

`Acmp_OutputSelectType` `Acmp_ComparatorType::OutputSelect`

Output signal selection

Definition at line 190 of file `Acmp_Hal_Types.h`.

### 3.2.2.11 OutputTrigger

`Acmp_OutputTriggerType` `Acmp_ComparatorType::OutputTrigger`

Trigger mode

Definition at line 189 of file `Acmp_Hal_Types.h`.

### 3.2.2.12 PinState

`boolean` `Acmp_ComparatorType::PinState`

ACMP output pin enable flag

Definition at line 184 of file `Acmp_Hal_Types.h`.

### 3.2.2.13 PowerEnable

`boolean` `Acmp_ComparatorType::PowerEnable`

ACMP analog module power enable flag

Definition at line 187 of file `Acmp_Hal_Types.h`.

### 3.2.2.14 UsingLSIEnable

`boolean` `Acmp_ComparatorType::UsingLSIEnable`

Using LSI as function clock enable flag

Definition at line 185 of file `Acmp_Hal_Types.h`.

### 3.2.2.15 WindowModeEnable

`boolean` `Acmp_ComparatorType::WindowModeEnable`

Window mode enable flag

Definition at line 186 of file `Acmp_Hal_Types.h`.

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

- [Acmp\\_Hal\\_Types.h](#)

## 3.3 Acmp\_DacType Struct Reference

ACMP DAC configuration type.

```
#include <Acmp_Hal_Types.h>
```

### Public Attributes

- [Acmp\\_VoltageReferenceType](#) VoltageReferenceSource
- uint8 [Voltage](#)
- boolean [State](#)

#### 3.3.1 Detailed Description

ACMP DAC configuration type.

Definition at line 210 of file Acmp\_Hal\_Types.h.

#### 3.3.2 Member Data Documentation

##### 3.3.2.1 State

```
boolean Acmp_DacType::State
```

DAC power enable flag

Definition at line 214 of file Acmp\_Hal\_Types.h.

##### 3.3.2.2 Voltage

```
uint8 Acmp_DacType::Voltage
```

DAC output code

Definition at line 213 of file Acmp\_Hal\_Types.h.

##### 3.3.2.3 VoltageReferenceSource

```
Acmp\_VoltageReferenceType Acmp_DacType::VoltageReferenceSource
```

DAC voltage reference source selection

Definition at line 212 of file Acmp\_Hal\_Types.h.

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

- [Acmp\\_Hal\\_Types.h](#)

## 3.4 Acmp\_ModuleType Struct Reference

ACMP module configuration type.

```
#include <Acmp_Hal_Types.h>
```

### Public Attributes

- [Acmp\\_ComparatorType](#) Comparator
- [Acmp\\_AnmuxType](#) Mux
- [Acmp\\_DacType](#) Dac
- [Acmp\\_PollingModeType](#) PollingMode

### 3.4.1 Detailed Description

ACMP module configuration type.

Definition at line 240 of file `Acmp_Hal_Types.h`.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 Comparator

[Acmp\\_ComparatorType](#) Acmp\_ModuleType::Comparator

comparator basic configuration

Definition at line 242 of file `Acmp_Hal_Types.h`.

#### 3.4.2.2 Dac

[Acmp\\_DacType](#) Acmp\_ModuleType::Dac

DAC configuration

Definition at line 244 of file `Acmp_Hal_Types.h`.

#### 3.4.2.3 Mux

[Acmp\\_AnmuxType](#) Acmp\_ModuleType::Mux

Input mux configuration

Definition at line 243 of file `Acmp_Hal_Types.h`.



#### 3.4.2.4 PollingMode

[Acmp\\_PollingModeType](#) Acmp\_ModuleType::PollingMode

Polling mode configuration

Definition at line 245 of file Acmp\_Hal\_Types.h.

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

- [Acmp\\_Hal\\_Types.h](#)

## 3.5 Acmp\_PollingModeType Struct Reference

ACMP polling mode configuration.

```
#include <Acmp_Hal_Types.h>
```

### Public Attributes

- [Acmp\\_PollingClkDivideType](#) PollingClockDivide
- [Acmp\\_InputPollingType](#) Mode
- [uint16](#) PollingSequence
- [boolean](#) HallOutputEnable
- [Acmp\\_ChNumberType](#) HallAOutputCh
- [Acmp\\_ChNumberType](#) HallBOutputCh
- [Acmp\\_ChNumberType](#) HallCOutputCh

### 3.5.1 Detailed Description

ACMP polling mode configuration.

Definition at line 224 of file Acmp\_Hal\_Types.h.

### 3.5.2 Member Data Documentation

#### 3.5.2.1 HallAOutputCh

[Acmp\\_ChNumberType](#) Acmp\_PollingModeType::HallAOutputCh

Hall A output channel selection

Definition at line 231 of file Acmp\_Hal\_Types.h.

### 3.5.2.2 HallBOutputCh

[Acmp\\_ChNumberType](#) Acmp\_PollingModeType::HallBOutputCh

Hall B output channel selection

Definition at line 232 of file Acmp\_Hal\_Types.h.

### 3.5.2.3 HallCOutputCh

[Acmp\\_ChNumberType](#) Acmp\_PollingModeType::HallCOutputCh

Hall C output channel selection

Definition at line 233 of file Acmp\_Hal\_Types.h.

### 3.5.2.4 HallOutputEnable

boolean Acmp\_PollingModeType::HallOutputEnable

Hall output enable flag

Definition at line 230 of file Acmp\_Hal\_Types.h.

### 3.5.2.5 Mode

[Acmp\\_InputPollingType](#) Acmp\_PollingModeType::Mode

Polling mode selection

Definition at line 227 of file Acmp\_Hal\_Types.h.

### 3.5.2.6 PollingClockDivide

[Acmp\\_PollingClkDivideType](#) Acmp\_PollingModeType::PollingClockDivide

Polling clock divide

Definition at line 226 of file Acmp\_Hal\_Types.h.

### 3.5.2.7 PollingSequence

uint16 Acmp\_PollingModeType::PollingSequence

Polling sequence, using bit0:8, each bit for each channel, bit 0 represent channel 0, and other bits likewise

Definition at line 228 of file Acmp\_Hal\_Types.h.

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

- [Acmp\\_Hal\\_Types.h](#)

## Chapter 4

# File Documentation

### 4.1 AC784xx\_Acmp\_Reg.h File Reference

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

#### Functions

- LOCAL\_INLINE ACMP\_Type \* [Acmp\\_Reg\\_GetBase](#) (uint8 Instance)  
*Get ACMP base.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetOutputPinEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP output pin enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetOutputPinEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP output pin enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetOutputSelection](#) (ACMP\_Type \*Base, [Acmp\\_OutputSelectType](#) Select)  
*Set ACMP output signal selection.*
- LOCAL\_INLINE [Acmp\\_OutputSelectType](#) [Acmp\\_Reg\\_GetOutputSelection](#) (const ACMP\_Type \*Base)  
*Get ACMP output signal selection.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP analog module power enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP analog module power enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetInterruptEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP interrupt enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetInterruptEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP interrupt enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetTriggerMode](#) (ACMP\_Type \*Base, [Acmp\\_OutputTriggerType](#) Mode)  
*Set ACMP trigger mode.*
- LOCAL\_INLINE [Acmp\\_OutputTriggerType](#) [Acmp\\_Reg\\_GetTriggerMode](#) (const ACMP\_Type \*Base)  
*Get ACMP trigger mode.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetUsingLSIEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP using LSI clock as function clock enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetUsingLSIEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP using LSI clock as function clock enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetPositiveInputPin](#) (ACMP\_Type \*Base, [Acmp\\_ChNumberType](#) Channel)  
*Set ACMP positive input channel selection.*

- LOCAL\_INLINE [Acmp\\_ChNumberType](#) [Acmp\\_Reg\\_GetPositiveInputPin](#) (const ACMP\_Type \*Base)  
*Get ACMP positive input channel selection.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetNegativeInputPin](#) (ACMP\_Type \*Base, [Acmp\\_ChNumberType](#) channel)  
*Set ACMP negative input channel selection.*
- LOCAL\_INLINE [Acmp\\_ChNumberType](#) [Acmp\\_Reg\\_GetNegativeInputPin](#) (const ACMP\_Type \*Base)  
*Get ACMP negative input channel selection.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHysteresisMode](#) (ACMP\_Type \*Base, [Acmp\\_HysteresisModeType](#) mode)  
*Set ACMP hysteresis mode.*
- LOCAL\_INLINE [Acmp\\_HysteresisModeType](#) [Acmp\\_Reg\\_GetHysteresisMode](#) (const ACMP\_Type \*Base)  
*Get ACMP hysteresis mode.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHysteresisVoltage](#) (ACMP\_Type \*Base, [Acmp\\_HysteresisType](#) Vol)  
*Set ACMP hysteresis voltage type.*
- LOCAL\_INLINE [Acmp\\_HysteresisType](#) [Acmp\\_Reg\\_GetHysteresisVoltage](#) (const ACMP\_Type \*Base)  
*Get ACMP hysteresis voltage type.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetLowPassFilter](#) (ACMP\_Type \*Base, [Acmp\\_LowPassFilterType](#) Freq)  
*Set low pass filter bandwidth type.*
- LOCAL\_INLINE [Acmp\\_LowPassFilterType](#) [Acmp\\_Reg\\_GetLowPassFilter](#) (const ACMP\_Type \*Base)  
*Get low pass filter bandwidth type.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetFilterEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP digital filter enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetFilterEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP digital filter enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetFilterClockPrescaler](#) (ACMP\_Type \*Base, [Acmp\\_FilterDivideType](#) Psc)  
*Set digital filter clock prescaler.*
- LOCAL\_INLINE [Acmp\\_FilterDivideType](#) [Acmp\\_Reg\\_GetFilterClockPrescaler](#) (const ACMP\_Type \*Base)  
*Get digital filter clock prescaler.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetFilterLength](#) (ACMP\_Type \*Base, uint8 FilterCnt)  
*Set digital filter length.*
- LOCAL\_INLINE uint8 [Acmp\\_Reg\\_GetFilterLength](#) (const ACMP\_Type \*Base)  
*Get digital filter length.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetWindowMode](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP window mode enable setting.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetWindowMode](#) (const ACMP\_Type \*Base)  
*Get ACMP window mode enable setting.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetInvertMode](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP result inversion.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetInvertMode](#) (const ACMP\_Type \*Base)  
*Get ACMP result inversion.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHallOutputEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP hall output enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetHallOutputEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP hall output enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHallOutputA](#) (ACMP\_Type \*Base, [Acmp\\_ChNumberType](#) HallCh)  
*Set ACMP hall output A channel.*
- LOCAL\_INLINE [Acmp\\_ChNumberType](#) [Acmp\\_Reg\\_GetHallOutputA](#) (const ACMP\_Type \*Base)  
*Get ACMP hall output A channel.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHallOutputB](#) (ACMP\_Type \*Base, [Acmp\\_ChNumberType](#) HallCh)  
*Set ACMP hall output B channel.*
- LOCAL\_INLINE [Acmp\\_ChNumberType](#) [Acmp\\_Reg\\_GetHallOutputB](#) (const ACMP\_Type \*Base)  
*Get ACMP hall output B channel.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetHallOutputC](#) (ACMP\_Type \*Base, [Acmp\\_ChNumberType](#) HallCh)  
*Set ACMP hall output C channel.*
- LOCAL\_INLINE [Acmp\\_ChNumberType](#) [Acmp\\_Reg\\_GetHallOutputC](#) (const ACMP\_Type \*Base)

- Get ACMP hall output C channel.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetDACEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP DAC module enable flag.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetDACEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP DAC module enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetDACOutput](#) (ACMP\_Type \*Base, uint8 Value)  
*Set ACMP DAC output code value.*
- LOCAL\_INLINE uint8 [Acmp\\_Reg\\_GetDACOutput](#) (const ACMP\_Type \*Base)  
*Get ACMP DAC output code value.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetDACReference](#) (ACMP\_Type \*Base, [Acmp\\_VoltageReferenceType](#) Ref)  
*Set ACMP DAC reference source.*
- LOCAL\_INLINE [Acmp\\_VoltageReferenceType](#) [Acmp\\_Reg\\_GetDACReference](#) (const ACMP\_Type \*Base)  
*Get ACMP DAC reference source.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetPositivePollingEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP positive polling enable flag. Should not enable positive polling and negative polling.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetPositivePollingEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP positive polling enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetNegativePollingEnableFlag](#) (ACMP\_Type \*Base, boolean State)  
*Set ACMP negative polling enable flag. Should not enable positive polling and negative polling.*
- LOCAL\_INLINE boolean [Acmp\\_Reg\\_GetNegativePollingEnableFlag](#) (const ACMP\_Type \*Base)  
*Get ACMP negative polling enable flag.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetPollingSequence](#) (ACMP\_Type \*Base, uint16 Seq)  
*Set ACMP polling sequence.*
- LOCAL\_INLINE uint16 [Acmp\\_Reg\\_GetPollingSequence](#) (const ACMP\_Type \*Base)  
*Get ACMP polling sequence.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_SetPollingFreqDiv](#) (ACMP\_Type \*Base, [Acmp\\_PollingClkDivideType](#) Div)  
*Set ACMP polling clock divide.*
- LOCAL\_INLINE [Acmp\\_PollingClkDivideType](#) [Acmp\\_Reg\\_GetPollingFreqDiv](#) (const ACMP\_Type \*Base)  
*Get ACMP polling clock divide.*
- LOCAL\_INLINE uint8 [Acmp\\_Reg\\_GetOutputData](#) (const ACMP\_Type \*Base)  
*Get ACMP compare result in normal compare mode.*
- LOCAL\_INLINE uint16 [Acmp\\_Reg\\_GetPollingData](#) (const ACMP\_Type \*Base)  
*Get ACMP polling result.*
- LOCAL\_INLINE uint8 [Acmp\\_Reg\\_GetOutputStatus](#) (const ACMP\_Type \*Base)  
*Get ACMP normal comparator mode interrupt status.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_ClearOutputStatus](#) (ACMP\_Type \*Base)  
*Clear ACMP normal comparator mode interrupt status.*
- LOCAL\_INLINE uint16 [Acmp\\_Reg\\_GetPollingStatus](#) (const ACMP\_Type \*Base)  
*Get ACMP polling mode interrupt status.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_ClearPollingStatus](#) (ACMP\_Type \*Base)  
*Clear ACMP polling mode interrupt status.*
- LOCAL\_INLINE uint32 [Acmp\\_Reg\\_GetInterruptStatus](#) (const ACMP\_Type \*Base)  
*Get ACMP interrupt status.*
- LOCAL\_INLINE void [Acmp\\_Reg\\_ClearInterruptStatus](#) (ACMP\_Type \*Base, uint32 Status)  
*Clear ACMP interrupt status.*

#### 4.1.1 Function Documentation

#### 4.1.1.1 Acmp\_Reg\_ClearInterruptStatus()

```
LOCAL_INLINE void Acmp_Reg_ClearInterruptStatus (
    ACOMP_Type * Base,
    uint32 Status )
```

Clear ACMP interrupt status.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Status</i>	interrupt status, write 1 to clear interrupt status

##### Returns

void

Definition at line 902 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.2 Acmp\_Reg\_ClearOutputStatus()

```
LOCAL_INLINE void Acmp_Reg_ClearOutputStatus (
    ACOMP_Type * Base )
```

Clear ACMP normal comparator mode interrupt status.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

void

Definition at line 857 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.3 Acmp\_Reg\_ClearPollingStatus()

```
LOCAL_INLINE void Acmp_Reg_ClearPollingStatus (
    ACOMP_Type * Base )
```

Clear ACMP polling mode interrupt status.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

void

Definition at line 879 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.4 Acmp\_Reg\_GetBase()**

```
LOCAL_INLINE ACMP_Type* Acmp_Reg_GetBase (  
    uint8 Instance )
```

Get ACMP base.

**Parameters**

in	<i>Instance</i>	: ACMP hardware channel ID.
----	-----------------	-----------------------------

**Returns**

ACMP\_Type\*: the ACMP base addr.

Definition at line 67 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.5 Acmp\_Reg\_GetDACEnableFlag()**

```
LOCAL_INLINE boolean Acmp_Reg_GetDACEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP DAC module enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether DAC module is enabled

Definition at line 612 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.6 Acmp\_Reg\_GetDACOutput()**

```
LOCAL_INLINE uint8 Acmp_Reg_GetDACOutput (  
    const ACMP_Type * Base )
```

Get ACMP DAC output code value.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current DAC output code value

Definition at line 636 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.7 Acmp\_Reg\_GetDACReference()**

```
LOCAL_INLINE Acmp_VoltageReferenceType Acmp_Reg_GetDACReference (  
    const ACMP_Type * Base )
```

Get ACMP DAC reference source.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

DAC reference source

Definition at line 659 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.8 Acmp\_Reg\_GetEnableFlag()**

```
LOCAL_INLINE boolean Acmp_Reg_GetEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP analog module power enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether acmp analog module is enabled

Definition at line 153 of file AC784xx\_Acmp\_Reg.h.



#### 4.1.1.9 Acmp\_Reg\_GetFilterClockPrescaler()

```
LOCAL_INLINE Acmp_FilterDivideType Acmp_Reg_GetFilterClockPrescaler (
    const ACMP_Type * Base )
```

Get digital filter clock prescaler.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

current digital filter prescaler setting

Definition at line 413 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.10 Acmp\_Reg\_GetFilterEnableFlag()

```
LOCAL_INLINE boolean Acmp_Reg_GetFilterEnableFlag (
    const ACMP_Type * Base )
```

Get ACMP digital filter enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

whether digital filter is enabled

Definition at line 389 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.11 Acmp\_Reg\_GetFilterLength()

```
LOCAL_INLINE uint8 Acmp_Reg_GetFilterLength (
    const ACMP_Type * Base )
```

Get digital filter length.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current filter count setting

Definition at line 439 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.12 Acmp\_Reg\_GetHallOutputA()**

```
LOCAL_INLINE Acmp_ChNumberType Acmp_Reg_GetHallOutputA (  
    const ACMP_Type * Base )
```

Get ACMP hall output A channel.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

input channel of hall output

Definition at line 534 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.13 Acmp\_Reg\_GetHallOutputB()**

```
LOCAL_INLINE Acmp_ChNumberType Acmp_Reg_GetHallOutputB (  
    const ACMP_Type * Base )
```

Get ACMP hall output B channel.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

input channel of hall output

Definition at line 560 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.14 Acmp\_Reg\_GetHallOutputC()**

```
LOCAL_INLINE Acmp_ChNumberType Acmp_Reg_GetHallOutputC (  
    const ACMP_Type * Base )
```

Get ACMP hall output C channel.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

input channel of hall output

Definition at line 586 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.15 Acmp\_Reg\_GetHallOutputEnableFlag()**

```
LOCAL_INLINE boolean Acmp_Reg_GetHallOutputEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP hall output enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether hall output enable flag is enabled

Definition at line 510 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.16 Acmp\_Reg\_GetHysteresisMode()**

```
LOCAL_INLINE Acmp\_HysteresisModeType Acmp_Reg_GetHysteresisMode (  
    const ACMP_Type * Base )
```

Get ACMP hysteresis mode.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current hysteresis mode setting

Definition at line 311 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.17 Acmp\_Reg\_GetHysteresisVoltage()

```
LOCAL_INLINE Acmp_HysteresisType Acmp_Reg_GetHysteresisVoltage (  
    const ACMP_Type * Base )
```

Get ACMP hysteresis voltage type.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

current hysteresis voltage type

Definition at line 337 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.18 Acmp\_Reg\_GetInterruptEnableFlag()

```
LOCAL_INLINE boolean Acmp_Reg_GetInterruptEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP interrupt enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

whether acmp interrupt function is enabled

Definition at line 177 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.19 Acmp\_Reg\_GetInterruptStatus()

```
LOCAL_INLINE uint32 Acmp_Reg_GetInterruptStatus (  
    const ACMP_Type * Base )
```

Get ACMP interrupt status.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

interrupt status

Definition at line 890 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.20 Acmp\_Reg\_GetInvertMode()**

```
LOCAL_INLINE boolean Acmp_Reg_GetInvertMode (  
    const ACMP_Type * Base )
```

Get ACMP result inversion.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether result inversion is enabled

Definition at line 486 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.21 Acmp\_Reg\_GetLowPassFilter()**

```
LOCAL_INLINE Acmp_LowPassFilterType Acmp_Reg_GetLowPassFilter (  
    const ACMP_Type * Base )
```

Get low pass filter bandwidth type.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

low pass filter bandwidth

Definition at line 363 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.22 Acmp\_Reg\_GetNegativeInputPin()**

```
LOCAL_INLINE Acmp_ChNumberType Acmp_Reg_GetNegativeInputPin (  
    const ACMP_Type * Base )
```

Get ACMP negative input channel selection.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current channel that connect to the negative input.

Definition at line 285 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.23 Acmp\_Reg\_GetNegativePollingEnableFlag()**

```
LOCAL_INLINE boolean Acmp_Reg_GetNegativePollingEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP negative polling enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether negative polling is enabled

Definition at line 762 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.24 Acmp\_Reg\_GetOutputData()**

```
LOCAL_INLINE uint8 Acmp_Reg_GetOutputData (  
    const ACMP_Type * Base )
```

Get ACMP compare result in normal compare mode.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

compare result

Definition at line 824 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.25 Acmp\_Reg\_GetOutputPinEnableFlag()

```
LOCAL_INLINE boolean Acmp_Reg_GetOutputPinEnableFlag (  
    const ACMP_Type * Base )
```

Get ACMP output pin enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

whether output pin function is enabled

Definition at line 99 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.26 Acmp\_Reg\_GetOutputSelection()

```
LOCAL_INLINE Acmp_OutputSelectType Acmp_Reg_GetOutputSelection (  
    const ACMP_Type * Base )
```

Get ACMP output signal selection.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

which output signal is selected

- ACMP\_COUT
- ACMP\_COUTA

Definition at line 127 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.27 Acmp\_Reg\_GetOutputStatus()

```
LOCAL_INLINE uint8 Acmp_Reg_GetOutputStatus (  
    const ACMP_Type * Base )
```

Get ACMP normal comparator mode interrupt status.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

normal mode interrupt status

Definition at line 846 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.28 Acmp\_Reg\_GetPollingData()**

```
LOCAL_INLINE uint16 Acmp_Reg_GetPollingData (  
    const ACMP_Type * Base )
```

Get ACMP polling result.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

polling channel result. Bits of Channels that not polling will be 0

Definition at line 835 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.29 Acmp\_Reg\_GetPollingFreqDiv()**

```
LOCAL_INLINE Acmp_PollingClkDivideType Acmp_Reg_GetPollingFreqDiv (  
    const ACMP_Type * Base )
```

Get ACMP polling clock divide.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current polling clock divide setting

Definition at line 810 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.30 Acmp\_Reg\_GetPollingSequence()**

```
LOCAL_INLINE uint16 Acmp_Reg_GetPollingSequence (  
    const ACMP_Type * Base )
```

Get ACMP polling sequence.



**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current polling channel sequence

Definition at line 787 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.31 Acmp\_Reg\_GetPollingStatus()**

```
LOCAL_INLINE uint16 Acmp_Reg_GetPollingStatus (  
    const ACMP_Type * Base )
```

Get ACMP polling mode interrupt status.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

polling mode interrupt status

Definition at line 868 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.32 Acmp\_Reg\_GetPositiveInputPin()**

```
LOCAL_INLINE Acmp\_ChNumberType Acmp_Reg_GetPositiveInputPin (  
    const ACMP_Type * Base )
```

Get ACMP positive input channel selection.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

current channel that connect to the positive input.

Definition at line 259 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.33 Acmp\_Reg\_GetPositivePollingEnableFlag()

```
LOCAL_INLINE boolean Acmp_Reg_GetPositivePollingEnableFlag (
    const ACMP_Type * Base )
```

Get ACMP positive polling enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

whether positive polling is enabled

Definition at line 738 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.34 Acmp\_Reg\_GetTriggerMode()

```
LOCAL_INLINE Acmp_OutputTriggerType Acmp_Reg_GetTriggerMode (
    const ACMP_Type * Base )
```

Get ACMP trigger mode.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

##### Returns

acmp current trigger mode

Definition at line 201 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.35 Acmp\_Reg\_GetUsingLSIEnableFlag()

```
LOCAL_INLINE boolean Acmp_Reg_GetUsingLSIEnableFlag (
    const ACMP_Type * Base )
```

Get ACMP using LSI clock as function clock enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether ACMP using LSI clock as function clock

Definition at line 235 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.36 Acmp\_Reg\_GetWindowMode()**

```
LOCAL_INLINE boolean Acmp_Reg_GetWindowMode (  
    const ACMP_Type * Base )
```

Get ACMP window mode enable setting.

**Parameters**

in	<i>Base</i>	acmp module base
----	-------------	------------------

**Returns**

whether window mode is enabled

Definition at line 462 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.37 Acmp\_Reg\_SetDACEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetDACEnableFlag (  
    ACMP_Type * Base,  
    boolean State )
```

Set ACMP DAC module enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable DAC module

**Returns**

void

Definition at line 601 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.38 Acmp\_Reg\_SetDACOutput()**

```
LOCAL_INLINE void Acmp_Reg_SetDACOutput (  
    ACMP_Type * Base,  
    uint8 Value )
```

Set ACMP DAC output code value.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>Value</i>	DAC output code value

**Returns**

void

Definition at line 625 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.39 Acmp\_Reg\_SetDACReference()**

```
LOCAL_INLINE void Acmp_Reg_SetDACReference (
    ACOMP_Type * Base,
    Acmp_VoltageReferenceType Ref )
```

Set ACMP DAC reference source.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>Ref</i>	reference source

**Returns**

void

Definition at line 648 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.40 Acmp\_Reg\_SetEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetEnableFlag (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP analog module power enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable acmp analog module

**Returns**

void

Definition at line 142 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.41 Acmp\_Reg\_SetFilterClockPrescaler()

```
LOCAL_INLINE void Acmp_Reg_SetFilterClockPrescaler (
    ACOMP_Type * Base,
    Acmp_FilterDivideType Psc )
```

Set digital filter clock prescaler.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Psc</i>	digital filter prescaler

##### Returns

void

Definition at line 402 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.42 Acmp\_Reg\_SetFilterEnableFlag()

```
LOCAL_INLINE void Acmp_Reg_SetFilterEnableFlag (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP digital filter enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable digital filter

##### Returns

void

Definition at line 378 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.43 Acmp\_Reg\_SetFilterLength()

```
LOCAL_INLINE void Acmp_Reg_SetFilterLength (
    ACOMP_Type * Base,
    uint8 FilterCnt )
```

Set digital filter length.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>FilterCnt</i>	filter count setting

**Returns**

void

Definition at line 428 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.44 Acmp\_Reg\_SetHallOutputA()**

```
LOCAL_INLINE void Acmp_Reg_SetHallOutputA (
    ACOMP_Type * Base,
    Acmp_ChNumberType HallCh )
```

Set ACMP hall output A channel.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>HallCh</i>	hall channel

**Returns**

void

Definition at line 523 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.45 Acmp\_Reg\_SetHallOutputB()**

```
LOCAL_INLINE void Acmp_Reg_SetHallOutputB (
    ACOMP_Type * Base,
    Acmp_ChNumberType HallCh )
```

Set ACMP hall output B channel.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>HallCh</i>	hall channel

**Returns**

void

Definition at line 549 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.46 Acmp\_Reg\_SetHallOutputC()

```
LOCAL_INLINE void Acmp_Reg_SetHallOutputC (
    ACOMP_Type * Base,
    Acmp_ChNumberType HallCh )
```

Set ACMP hall output C channel.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>HallCh</i>	hall channel

##### Returns

void

Definition at line 575 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.47 Acmp\_Reg\_SetHallOutputEnableFlag()

```
LOCAL_INLINE void Acmp_Reg_SetHallOutputEnableFlag (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP hall output enable flag.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable hall output enable flag

##### Returns

void

Definition at line 499 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.48 Acmp\_Reg\_SetHysteresisMode()

```
LOCAL_INLINE void Acmp_Reg_SetHysteresisMode (
    ACOMP_Type * Base,
    Acmp_HysteresisModeType mode )
```

Set ACMP hysteresis mode.



**Parameters**

in	<i>Base</i>	acmp module base
in	<i>mode</i>	hysteresis mode

**Returns**

void

Definition at line 300 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.49 Acmp\_Reg\_SetHysteresisVoltage()**

```
LOCAL_INLINE void Acmp_Reg_SetHysteresisVoltage (
    ACMP_Type * Base,
    Acmp_HysteresisType Vol )
```

Set ACMP hysteresis voltage type.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>Vol</i>	hysteresis voltage type

**Returns**

void

Definition at line 326 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.50 Acmp\_Reg\_SetInterruptEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetInterruptEnableFlag (
    ACMP_Type * Base,
    boolean State )
```

Set ACMP interrupt enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable interrupt function

**Returns**

void

Definition at line 166 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.51 Acmp\_Reg\_SetInvertMode()

```
LOCAL_INLINE void Acmp_Reg_SetInvertMode (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP result inversion.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>state</i>	enable or disable result inversion

##### Returns

void

Definition at line 475 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.52 Acmp\_Reg\_SetLowPassFilter()

```
LOCAL_INLINE void Acmp_Reg_SetLowPassFilter (
    ACOMP_Type * Base,
    Acmp_LowPassFilterType Freq )
```

Set low pass filter bandwidth type.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Freq</i>	low pass filter bandwidth

##### Returns

void

Definition at line 352 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.53 Acmp\_Reg\_SetNegativeInputPin()

```
LOCAL_INLINE void Acmp_Reg_SetNegativeInputPin (
    ACOMP_Type * Base,
    Acmp_ChNumberType channel )
```

Set ACMP negative input channel selection.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>channel</i>	channel that need connect to the negative input

**Returns**

void

Definition at line 274 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.54 Acmp\_Reg\_SetNegativePollingEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetNegativePollingEnableFlag (
    ACMP_Type * Base,
    boolean State )
```

Set ACMP negative polling enable flag. Should not enable positive polling and negative polling.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable negative polling

**Returns**

void

Definition at line 751 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.55 Acmp\_Reg\_SetOutputPinEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetOutputPinEnableFlag (
    ACMP_Type * Base,
    boolean State )
```

Set ACMP output pin enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable output pin function

**Returns**

void

Definition at line 88 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.56 Acmp\_Reg\_SetOutputSelection()

```
LOCAL_INLINE void Acmp_Reg_SetOutputSelection (
    ACOMP_Type * Base,
    Acmp_OutputSelectType Select )
```

Set ACMP output signal selection.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>select</i>	which signal to output <ul style="list-style-type: none"><li>• ACMP_COUT</li><li>• ACMP_COUTA</li></ul>

##### Returns

void

Definition at line 114 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.57 Acmp\_Reg\_SetPollingFreqDiv()

```
LOCAL_INLINE void Acmp_Reg_SetPollingFreqDiv (
    ACOMP_Type * Base,
    Acmp_PollingClkDivideType Div )
```

Set ACMP polling clock divide.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Div</i>	polling clock divide

##### Returns

void

Definition at line 799 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.58 Acmp\_Reg\_SetPollingSequence()

```
LOCAL_INLINE void Acmp_Reg_SetPollingSequence (
    ACOMP_Type * Base,
    uint16 Seq )
```

Set ACMP polling sequence.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Seq</i>	channel sequence that need to poll, each bit corresponding to one channel bit 0 represent channel 0 and other bit is likewise.

##### Returns

void

Definition at line 776 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.59 Acmp\_Reg\_SetPositiveInputPin()

```
LOCAL_INLINE void Acmp_Reg_SetPositiveInputPin (
    ACOMP_Type * Base,
    Acmp_ChNumberType Channel )
```

Set ACMP positive input channel selection.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>Channel</i>	channel that need connect to the positive input

##### Returns

void

Definition at line 248 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.60 Acmp\_Reg\_SetPositivePollingEnableFlag()

```
LOCAL_INLINE void Acmp_Reg_SetPositivePollingEnableFlag (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP positive polling enable flag. Should not enable positive polling and negative polling.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable positive polling

**Returns**

void

Definition at line 727 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.61 Acmp\_Reg\_SetTriggerMode()**

```
LOCAL_INLINE void Acmp_Reg_SetTriggerMode (
    ACOMP_Type * Base,
    Acmp_OutputTriggerType Mode )
```

Set ACMP trigger mode.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>Mode</i>	trigger mode

**Returns**

void

Definition at line 190 of file AC784xx\_Acmp\_Reg.h.

**4.1.1.62 Acmp\_Reg\_SetUsingLSIEnableFlag()**

```
LOCAL_INLINE void Acmp_Reg_SetUsingLSIEnableFlag (
    ACOMP_Type * Base,
    boolean State )
```

Set ACMP using LSI clock as function clock enable flag.

**Parameters**

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable ACMP using LSI clock as function clock

**Returns**

void

Definition at line 221 of file AC784xx\_Acmp\_Reg.h.

#### 4.1.1.63 Acmp\_Reg\_SetWindowMode()

```
LOCAL_INLINE void Acmp_Reg_SetWindowMode (
    ACMP_Type * Base,
    boolean State )
```

Set ACMP window mode enable setting.

##### Parameters

in	<i>Base</i>	acmp module base
in	<i>State</i>	enable or disable window mode

##### Returns

void

Definition at line 451 of file AC784xx\_Acmp\_Reg.h.

## 4.2 AC784xx\_API\_Reference\_Manual\_ACMP.pdf File Reference

## 4.3 Acmp\_Hal.c File Reference

This file provides analog comparator module integration functions.

```
#include "AC784xx_Acmp_Reg.h"
#include "Acmp_Hal.h"
#include "Ckgen_Hal.h"
#include "Rcm_Hal.h"
#include "Core_Hal.h"
```

### Functions

- [ISR](#) (ACMP0\_IRQHandler)  
*ACMP0 interrupt handler.*
- void [Acmp\\_Hal\\_GetDefaultConfig](#) (Acmp\_ModuleType \*Config)  
*Get default configuration structure.*
- void [Acmp\\_Hal\\_Reset](#) (uint8 Instance)  
*Reset all ACMP registers. Need to be called after ACMP clock is applied and reset is deasserted.*
- void [Acmp\\_Hal\\_Init](#) (uint8 Instance, const Acmp\_ModuleType \*const Config)  
*Configure all comparator function with the given configuration structure.*
- void [Acmp\\_Hal\\_Deinit](#) (uint8 Instance)  
*Reset ACMP module and close the clock source.*
- void [Acmp\\_Hal\\_GetConfigAll](#) (uint8 Instance, Acmp\_ModuleType \*const Config)

*Get the current ACMP configuration.*

- uint8 [Acmp\\_Hal\\_GetOutputData](#) (uint8 Instance)  
*Get ACMP normal mode output data.*
- uint8 [Acmp\\_Hal\\_GetOutputFlags](#) (uint8 Instance)  
*Get ACMP output Status flags.*
- void [Acmp\\_Hal\\_ClearOutputFlags](#) (uint8 Instance)  
*Clear ACMP output flags Status.*
- uint16 [Acmp\\_Hal\\_GetPollingData](#) (uint8 Instance)  
*Get polling mode compare data.*
- uint16 [Acmp\\_Hal\\_GetPollingFlags](#) (uint8 Instance)  
*Get polling mode Status flags.*
- void [Acmp\\_Hal\\_ClearInputFlags](#) (uint8 Instance)  
*Clear polling mode Status flags.*

### 4.3.1 Detailed Description

This file provides analog comparator module integration functions.

### 4.3.2 Function Documentation

#### 4.3.2.1 Acmp\_Hal\_ClearInputFlags()

```
void Acmp_Hal_ClearInputFlags (  
    uint8 Instance )
```

Clear polling mode Status flags.

#### Note

Function ID: DES\_ACMP\_API\_007

#### Parameters

in	Instance	ACMP Instance number
----	----------	----------------------

#### Returns

void.

Definition at line 464 of file Acmp\_Hal.c.

#### 4.3.2.2 Acmp\_Hal\_ClearOutputFlags()

```
void Acmp_Hal_ClearOutputFlags (  
    uint8 Instance )
```

Clear ACMP output flags Status.



**Note**

Function ID: DES\_ACMP\_API\_005

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

void.

Definition at line 405 of file Acmp\_Hal.c.

**4.3.2.3 Acmp\_Hal\_Deinit()**

```
void Acmp_Hal_Deinit (
    uint8 Instance )
```

Reset ACMP module and close the clock source.

**Note**

Function ID: DES\_ACMP\_API\_001

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

void

< Disable ACMP interrupt

Definition at line 317 of file Acmp\_Hal.c.

**4.3.2.4 Acmp\_Hal\_GetConfigAll()**

```
void Acmp_Hal_GetConfigAll (
    uint8 Instance,
    Acmp_ModuleType *const Config )
```

Get the current ACMP configuration.

**Note**

Function ID: DES\_ACMP\_API\_001

**Parameters**

in	<i>Instance</i>	ACMP Instance number
out	<i>Config</i>	ACMP configuration structure that need to fill with current configuration

**Returns**

void.

Definition at line 343 of file Acmp\_Hal.c.

**4.3.2.5 Acmp\_Hal\_GetDefaultConfig()**

```
void Acmp_Hal_GetDefaultConfig (
    Acmp_ModuleType * Config )
```

Get default configuration structure.

**Note**

Function ID: DES\_ACMP\_API\_008

**Parameters**

out	<i>Config</i>	ACMP configuration structure
-----	---------------	------------------------------

**Returns**

void

Definition at line 218 of file Acmp\_Hal.c.

**4.3.2.6 Acmp\_Hal\_GetOutputData()**

```
uint8 Acmp_Hal_GetOutputData (
    uint8 Instance )
```

Get ACMP normal mode output data.

**Note**

Function ID: DES\_ACMP\_API\_002

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

output data.

Definition at line 364 of file Acmp\_Hal.c.

**4.3.2.7 Acmp\_Hal\_GetOutputFlags()**

```
uint8 Acmp_Hal_GetOutputFlags (  
    uint8 Instance )
```

Get ACMP output Status flags.

**Note**

Function ID: DES\_ACMP\_API\_004

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

output Status flags.

Definition at line 385 of file Acmp\_Hal.c.

**4.3.2.8 Acmp\_Hal\_GetPollingData()**

```
uint16 Acmp_Hal_GetPollingData (  
    uint8 Instance )
```

Get polling mode compare data.

**Note**

Function ID: DES\_ACMP\_API\_003

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

polling mode input channel compare data.

Definition at line 423 of file Acmp\_Hal.c.

#### 4.3.2.9 Acmp\_Hal\_GetPollingFlags()

```
uint16 Acmp_Hal_GetPollingFlags (
    uint8 Instance )
```

Get polling mode Status flags.

##### Note

Function ID: DES\_ACMP\_API\_006

##### Parameters

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

##### Returns

polling mode input channel Status.

Definition at line 444 of file Acmp\_Hal.c.

#### 4.3.2.10 Acmp\_Hal\_Init()

```
void Acmp_Hal_Init (
    uint8 Instance,
    const Acmp_ModuleType *const Config )
```

Configure all comparator function with the given configuration structure.

##### Note

Function ID: DES\_ACMP\_API\_000

##### Parameters

in	<i>Instance</i>	ACMP Instance number
in	<i>Config</i>	ACMP configuration structure that need to apply

##### Returns

void

Definition at line 291 of file Acmp\_Hal.c.

#### 4.3.2.11 Acmp\_Hal\_Reset()

```
void Acmp_Hal_Reset (
    uint8 Instance )
```

Reset all ACMP registers. Need to be called after ACMP clock is applied and reset is deasserted.

**Note**

Function ID: DES\_ACMP\_API\_009

**Parameters**

<i>in</i>	<i>Instance</i>	ACMP Instance number
-----------	-----------------	----------------------

**Returns**

void

< Disable ACMP interrupt

Definition at line 267 of file Acmp\_Hal.c.

**4.3.2.12 ISR()**

```
ISR (
    ACMP0_IRQHandler )
```

ACMP0 interrupt handler.

**Note**

Function ID: DES\_ACMP\_API\_010

**Returns**

void

Definition at line 481 of file Acmp\_Hal.c.

## 4.4 Acmp\_Hal.h File Reference

This file provides analog comparator module integration functions interfaces.

```
#include "Acmp_Hal_Types.h"
```

## Functions

- void [Acmp\\_Hal\\_GetDefaultConfig](#) ([Acmp\\_ModuleType](#) \*Config)  
*Get default configuration structure.*
- void [Acmp\\_Hal\\_Reset](#) (uint8 Instance)  
*Reset all ACMP registers. Need to be called after ACMP clock is applied and reset is deasserted.*
- void [Acmp\\_Hal\\_Init](#) (uint8 Instance, const [Acmp\\_ModuleType](#) \*const Config)  
*Configure all comparator function with the given configuration structure.*
- void [Acmp\\_Hal\\_Deinit](#) (uint8 Instance)  
*Reset ACMP module and close the clock source.*
- void [Acmp\\_Hal\\_GetConfigAll](#) (uint8 Instance, [Acmp\\_ModuleType](#) \*const Config)  
*Get the current ACMP configuration.*
- uint8 [Acmp\\_Hal\\_GetOutputData](#) (uint8 Instance)  
*Get ACMP normal mode output data.*
- uint8 [Acmp\\_Hal\\_GetOutputFlags](#) (uint8 Instance)  
*Get ACMP output Status flags.*
- void [Acmp\\_Hal\\_ClearOutputFlags](#) (uint8 Instance)  
*Clear ACMP output flags Status.*
- uint16 [Acmp\\_Hal\\_GetPollingData](#) (uint8 Instance)  
*Get polling mode compare data.*
- uint16 [Acmp\\_Hal\\_GetPollingFlags](#) (uint8 Instance)  
*Get polling mode Status flags.*
- void [Acmp\\_Hal\\_ClearInputFlags](#) (uint8 Instance)  
*Clear polling mode Status flags.*

### 4.4.1 Detailed Description

This file provides analog comparator module integration functions interfaces.

### 4.4.2 Function Documentation

#### 4.4.2.1 Acmp\_Hal\_ClearInputFlags()

```
void Acmp_Hal_ClearInputFlags (
    uint8 Instance )
```

Clear polling mode Status flags.

#### Note

Function ID: DES\_ACMP\_API\_007

#### Parameters

in	Instance	ACMP Instance number
----	----------	----------------------

**Returns**

void.

Definition at line 464 of file Acmp\_Hal.c.

**4.4.2.2 Acmp\_Hal\_ClearOutputFlags()**

```
void Acmp_Hal_ClearOutputFlags (
    uint8 Instance )
```

Clear ACMP output flags Status.

**Note**

Function ID: DES\_ACMP\_API\_005

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

void.

Definition at line 405 of file Acmp\_Hal.c.

**4.4.2.3 Acmp\_Hal\_Deinit()**

```
void Acmp_Hal_Deinit (
    uint8 Instance )
```

Reset ACMP module and close the clock source.

**Note**

Function ID: DES\_ACMP\_API\_001

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

void

< Disable ACMP interrupt

Definition at line 317 of file Acmp\_Hal.c.

#### 4.4.2.4 Acmp\_Hal\_GetConfigAll()

```
void Acmp_Hal_GetConfigAll (
    uint8 Instance,
    Acmp_ModuleType *const Config )
```

Get the current ACMP configuration.

##### Note

Function ID: DES\_ACMP\_API\_001

##### Parameters

in	<i>Instance</i>	ACMP Instance number
out	<i>Config</i>	ACMP configuration structure that need to fill with current configuration

##### Returns

void.

Definition at line 343 of file Acmp\_Hal.c.

#### 4.4.2.5 Acmp\_Hal\_GetDefaultConfig()

```
void Acmp_Hal_GetDefaultConfig (
    Acmp_ModuleType * Config )
```

Get default configuration structure.

##### Note

Function ID: DES\_ACMP\_API\_008

##### Parameters

out	<i>Config</i>	ACMP configuration structure
-----	---------------	------------------------------

##### Returns

void

Definition at line 218 of file Acmp\_Hal.c.



#### 4.4.2.6 Acmp\_Hal\_GetOutputData()

```
uint8 Acmp_Hal_GetOutputData (
    uint8 Instance )
```

Get ACMP normal mode output data.

##### Note

Function ID: DES\_ACMP\_API\_002

##### Parameters

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

##### Returns

output data.

Definition at line 364 of file Acmp\_Hal.c.

#### 4.4.2.7 Acmp\_Hal\_GetOutputFlags()

```
uint8 Acmp_Hal_GetOutputFlags (
    uint8 Instance )
```

Get ACMP output Status flags.

##### Note

Function ID: DES\_ACMP\_API\_004

##### Parameters

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

##### Returns

output Status flags.

Definition at line 385 of file Acmp\_Hal.c.

#### 4.4.2.8 Acmp\_Hal\_GetPollingData()

```
uint16 Acmp_Hal_GetPollingData (
    uint8 Instance )
```

Get polling mode compare data.

**Note**

Function ID: DES\_ACMP\_API\_003

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

polling mode input channel compare data.

Definition at line 423 of file Acmp\_Hal.c.

**4.4.2.9 Acmp\_Hal\_GetPollingFlags()**

```
uint16 Acmp_Hal_GetPollingFlags (
    uint8 Instance )
```

Get polling mode Status flags.

**Note**

Function ID: DES\_ACMP\_API\_006

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

polling mode input channel Status.

Definition at line 444 of file Acmp\_Hal.c.

**4.4.2.10 Acmp\_Hal\_Init()**

```
void Acmp_Hal_Init (
    uint8 Instance,
    const Acmp_ModuleType *const Config )
```

Configure all comparator function with the given configuration structure.

**Note**

Function ID: DES\_ACMP\_API\_000

**Parameters**

in	<i>Instance</i>	ACMP Instance number
in	<i>Config</i>	ACMP configuration structure that need to apply

**Returns**

void

Definition at line 291 of file Acmp\_Hal.c.

**4.4.2.11 Acmp\_Hal\_Reset()**

```
void Acmp_Hal_Reset (
    uint8 Instance )
```

Reset all ACMP registers. Need to be called after ACMP clock is applied and reset is deasserted.

**Note**

Function ID: DES\_ACMP\_API\_009

**Parameters**

in	<i>Instance</i>	ACMP Instance number
----	-----------------	----------------------

**Returns**

void

< Disable ACMP interrupt

Definition at line 267 of file Acmp\_Hal.c.

## 4.5 Acmp\_Hal\_Types.h File Reference

This file provides ACMP config.

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

**Classes**

- struct [Acmp\\_ComparatorType](#)  
*ACMP comparator configuration type.*
- struct [Acmp\\_AnmuxType](#)  
*ACMP analog mux input configuration type.*

- struct [Acmp\\_DacType](#)  
*ACMP DAC configuration type.*
- struct [Acmp\\_PollingModeType](#)  
*ACMP polling mode configuration.*
- struct [Acmp\\_ModuleType](#)  
*ACMP module configuration type.*

## Typedefs

- typedef void(\* [Acmp\\_CallbackType](#)) (uint8 Instance, uint32 Status)  
*ACMP interrupt callback function type.*

## Enumerations

- enum [Acmp\\_ChNumberType](#) {  
[ACMP\\_EXTERNAL\\_CH0](#) = 0U, [ACMP\\_EXTERNAL\\_CH1](#), [ACMP\\_EXTERNAL\\_CH2](#), [ACMP\\_EXTERNAL\\_CH3](#),  
[ACMP\\_EXTERNAL\\_CH4](#), [ACMP\\_EXTERNAL\\_CH5](#), [ACMP\\_EXTERNAL\\_CH6](#), [ACMP\\_EXTERNAL\\_CH7](#),  
[ACMP\\_DAC\\_OUTPUT](#), [ACMP\\_CHANNEL\\_MAX](#) }  
*ACMP input channel number.*
- enum [Acmp\\_OutputTriggerType](#) { [ACMP\\_FALLING\\_EDGE](#) = 0U, [ACMP\\_RISING\\_EDGE](#) = 1U, [ACMP\\_BOTH\\_EDGES](#) = 3U }  
*ACMP output interrupt trigger type.*
- enum [Acmp\\_OutputSelectType](#) { [ACMP\\_COUT](#) = 0U, [ACMP\\_COUTA](#) = 1U }  
*ACMP result output signal select.*
- enum [Acmp\\_HysteresisModeType](#) { [ACMP\\_HYS\\_FALLING\\_EDGE](#) = 0U, [ACMP\\_HYS\\_BOTH\\_EDGE](#) }  
*ACMP hysteresis mode type.*
- enum [Acmp\\_HysteresisType](#) { [ACMP\\_LEVEL\\_HYS\\_0MV](#) = 0U, [ACMP\\_LEVEL\\_HYS\\_10MV](#), [ACMP\\_LEVEL\\_HYS\\_20MV](#), [ACMP\\_LEVEL\\_HYS\\_40MV](#) }  
*ACMP hysteresis level type.*
- enum [Acmp\\_FilterDivideType](#) {  
[ACMP\\_FLT\\_DIVIDE\\_1](#) = 0U, [ACMP\\_FLT\\_DIVIDE\\_2](#), [ACMP\\_FLT\\_DIVIDE\\_3](#), [ACMP\\_FLT\\_DIVIDE\\_4](#),  
[ACMP\\_FLT\\_DIVIDE\\_5](#), [ACMP\\_FLT\\_DIVIDE\\_6](#), [ACMP\\_FLT\\_DIVIDE\\_7](#), [ACMP\\_FLT\\_DIVIDE\\_8](#) }  
*ACMP digital filter clock divide type.*
- enum [Acmp\\_LowPassFilterType](#) { [ACMP\\_LPF\\_500KHZ](#) = 0U, [ACMP\\_LPF\\_1MHZ](#), [ACMP\\_LPF\\_2MHZ](#), [ACMP\\_LPF\\_BYPASS](#) }  
*ACMP low pass filter bandwidth type.*
- enum [Acmp\\_VoltageReferenceType](#) { [ACMP\\_DAC\\_BANDGAP](#) = 0U, [ACMP\\_DAC\\_VDD](#) }  
*ACMP DAC voltage reference source type.*
- enum [Acmp\\_InputPollingType](#) { [ACMP\\_NONE\\_POLLING](#) = 0U, [ACMP\\_POSITIVE\\_POLLING](#), [ACMP\\_NEGATIVE\\_POLLING](#) }  
*ACMP input channel polling mode type.*
- enum [Acmp\\_PollingClkDivideType](#) {  
[ACMP\\_CLK\\_DIVIDE\\_256](#) = 0U, [ACMP\\_CLK\\_DIVIDE\\_101](#), [ACMP\\_CLK\\_DIVIDE\\_71](#), [ACMP\\_CLK\\_DIVIDE\\_51](#),  
[ACMP\\_CLK\\_DIVIDE\\_MAX](#) }  
*ACMP polling clock divide type.*

### 4.5.1 Detailed Description

This file provides ACMP config.

## 4.5.2 Typedef Documentation

### 4.5.2.1 Acmp\_CallbackType

```
typedef void(* Acmp_CallbackType) (uint8 Instance, uint32 Status)
```

ACMP interrupt callback function type.

Definition at line 174 of file Acmp\_Hal\_Types.h.

## 4.5.3 Enumeration Type Documentation

### 4.5.3.1 Acmp\_ChNumberType

```
enum Acmp_ChNumberType
```

ACMP input channel number.

Enumerator

ACMP_EXTERNAL_CH0	External channel 0
ACMP_EXTERNAL_CH1	External channel 1
ACMP_EXTERNAL_CH2	External channel 2
ACMP_EXTERNAL_CH3	External channel 3
ACMP_EXTERNAL_CH4	External channel 4
ACMP_EXTERNAL_CH5	External channel 5
ACMP_EXTERNAL_CH6	External channel 6
ACMP_EXTERNAL_CH7	External channel 7
ACMP_DAC_OUTPUT	DAC output channel 8
ACMP_CHANNEL_MAX	Invalid channel

Definition at line 61 of file Acmp\_Hal\_Types.h.

### 4.5.3.2 Acmp\_FilterDivideType

```
enum Acmp_FilterDivideType
```

ACMP digital filter clock divide type.

Enumerator

ACMP_FLT_DIVIDE↔ _1	Input clock divided by 1.
------------------------	---------------------------

## Enumerator

ACMP_FLT_DIVIDE↔ _2	Input clock divided by 2.
ACMP_FLT_DIVIDE↔ _3	Input clock divided by 3.
ACMP_FLT_DIVIDE↔ _4	Input clock divided by 4.
ACMP_FLT_DIVIDE↔ _5	Input clock divided by 5.
ACMP_FLT_DIVIDE↔ _6	Input clock divided by 6.
ACMP_FLT_DIVIDE↔ _7	Input clock divided by 7.
ACMP_FLT_DIVIDE↔ _8	Input clock divided by 8.

Definition at line 117 of file Acmp\_Hal\_Types.h.

## 4.5.3.3 Acmp\_HysteresisModeType

enum [Acmp\\_HysteresisModeType](#)

ACMP hysteresis mode type.

## Enumerator

ACMP_HYS_FALLING_EDGE	hysteresis direction on falling edge
ACMP_HYS_BOTH_EDGE	hysteresis direction on both edge

Definition at line 97 of file Acmp\_Hal\_Types.h.

## 4.5.3.4 Acmp\_HysteresisType

enum [Acmp\\_HysteresisType](#)

ACMP hysteresis level type.

## Enumerator

ACMP_LEVEL_HYS_0MV	0mv LEVEL_HYS voltage
ACMP_LEVEL_HYS_10MV	10mv LEVEL_HYS voltage
ACMP_LEVEL_HYS_20MV	20mv LEVEL_HYS voltage
ACMP_LEVEL_HYS_40MV	40mv LEVEL_HYS voltage

Definition at line 106 of file Acmp\_Hal\_Types.h.

#### 4.5.3.5 Acmp\_InputPollingType

enum [Acmp\\_InputPollingType](#)

ACMP input channel polling mode type.

##### Enumerator

ACMP_NONE_POLLING	Not using polling mode, which is normal comparator mode
ACMP_POSITIVE_POLLING	Positive polling
ACMP_NEGATIVE_POLLING	Negative polling

Definition at line 152 of file [Acmp\\_Hal\\_Types.h](#).

#### 4.5.3.6 Acmp\_LowPassFilterType

enum [Acmp\\_LowPassFilterType](#)

ACMP low pass filter bandwidth type.

##### Enumerator

ACMP_LPF_500KHZ	500KHz filter frequency
ACMP_LPF_1MHZ	1MHz filter frequency
ACMP_LPF_2MHZ	2MHz filter frequency
ACMP_LPF_BYPASS	Bypass filter frequency

Definition at line 132 of file [Acmp\\_Hal\\_Types.h](#).

#### 4.5.3.7 Acmp\_OutputSelectType

enum [Acmp\\_OutputSelectType](#)

ACMP result output signal select.

##### Enumerator

ACMP_COUT	Select filtered result COUT as comparator output signal.
ACMP_COUTA	Select unfiltered result COUTA as comparator output signal.

Definition at line 88 of file [Acmp\\_Hal\\_Types.h](#).

#### 4.5.3.8 Acmp\_OutputTriggerType

enum [Acmp\\_OutputTriggerType](#)

ACMP output interrupt trigger type.

#### Enumerator

ACMP_FALLING_EDGE	Trigger interrupt on falling edge
ACMP_RISING_EDGE	Trigger interrupt on rising edge
ACMP_BOTH_EDGES	Trigger interrupt on falling/rising edge

Definition at line 78 of file Acmp\_Hal\_Types.h.

#### 4.5.3.9 Acmp\_PollingClkDivideType

```
enum Acmp_PollingClkDivideType
```

ACMP polling clock divide type.

#### Enumerator

ACMP_CLK_DIVIDE_256	Clock/256
ACMP_CLK_DIVIDE_101	Clock/101
ACMP_CLK_DIVIDE_71	Clock/71
ACMP_CLK_DIVIDE_51	Clock/51
ACMP_CLK_DIVIDE_MAX	Invalid divid ratio

Definition at line 162 of file Acmp\_Hal\_Types.h.

#### 4.5.3.10 Acmp\_VoltageReferenceType

```
enum Acmp_VoltageReferenceType
```

ACMP DAC voltage reference source type.

#### Enumerator

ACMP_DAC_BANDGAP	Bandgap as reference source
ACMP_DAC_VDD	Vdd as reference source

Definition at line 143 of file Acmp\_Hal\_Types.h.



# Index

AC784xx\_API\_Reference\_Manual\_ACMP.pdf, [40](#)

AC784xx\_Acmp\_Reg.h, [12](#)

[Acmp\\_Reg\\_ClearInterruptStatus, 14](#)  
[Acmp\\_Reg\\_ClearOutputStatus, 15](#)  
[Acmp\\_Reg\\_ClearPollingStatus, 15](#)  
[Acmp\\_Reg\\_GetBase, 16](#)  
[Acmp\\_Reg\\_GetDACEnableFlag, 16](#)  
[Acmp\\_Reg\\_GetDACOutput, 16](#)  
[Acmp\\_Reg\\_GetDACReference, 17](#)  
[Acmp\\_Reg\\_GetEnableFlag, 17](#)  
[Acmp\\_Reg\\_GetFilterClockPrescaler, 17](#)  
[Acmp\\_Reg\\_GetFilterEnableFlag, 18](#)  
[Acmp\\_Reg\\_GetFilterLength, 18](#)  
[Acmp\\_Reg\\_GetHallOutputEnableFlag, 20](#)  
[Acmp\\_Reg\\_GetHallOutputA, 19](#)  
[Acmp\\_Reg\\_GetHallOutputB, 19](#)  
[Acmp\\_Reg\\_GetHallOutputC, 19](#)  
[Acmp\\_Reg\\_GetHysteresisMode, 20](#)  
[Acmp\\_Reg\\_GetHysteresisVoltage, 20](#)  
[Acmp\\_Reg\\_GetInterruptEnableFlag, 21](#)  
[Acmp\\_Reg\\_GetInterruptStatus, 21](#)  
[Acmp\\_Reg\\_GetInvertMode, 22](#)  
[Acmp\\_Reg\\_GetLowPassFilter, 22](#)  
[Acmp\\_Reg\\_GetNegativeInputPin, 22](#)  
[Acmp\\_Reg\\_GetNegativePollingEnableFlag, 23](#)  
[Acmp\\_Reg\\_GetOutputData, 23](#)  
[Acmp\\_Reg\\_GetOutputPinEnableFlag, 23](#)  
[Acmp\\_Reg\\_GetOutputSelection, 24](#)  
[Acmp\\_Reg\\_GetOutputStatus, 24](#)  
[Acmp\\_Reg\\_GetPollingData, 25](#)  
[Acmp\\_Reg\\_GetPollingFreqDiv, 25](#)  
[Acmp\\_Reg\\_GetPollingSequence, 25](#)  
[Acmp\\_Reg\\_GetPollingStatus, 26](#)  
[Acmp\\_Reg\\_GetPositiveInputPin, 26](#)  
[Acmp\\_Reg\\_GetPositivePollingEnableFlag, 26](#)  
[Acmp\\_Reg\\_GetTriggerMode, 27](#)  
[Acmp\\_Reg\\_GetUsingLSIEnableFlag, 27](#)  
[Acmp\\_Reg\\_GetWindowMode, 28](#)  
[Acmp\\_Reg\\_SetDACEnableFlag, 28](#)  
[Acmp\\_Reg\\_SetDACOutput, 28](#)  
[Acmp\\_Reg\\_SetDACReference, 30](#)  
[Acmp\\_Reg\\_SetEnableFlag, 30](#)  
[Acmp\\_Reg\\_SetFilterClockPrescaler, 31](#)  
[Acmp\\_Reg\\_SetFilterEnableFlag, 31](#)  
[Acmp\\_Reg\\_SetFilterLength, 31](#)  
[Acmp\\_Reg\\_SetHallOutputEnableFlag, 33](#)  
[Acmp\\_Reg\\_SetHallOutputA, 32](#)  
[Acmp\\_Reg\\_SetHallOutputB, 32](#)  
[Acmp\\_Reg\\_SetHallOutputC, 33](#)  
[Acmp\\_Reg\\_SetHysteresisMode, 33](#)  
[Acmp\\_Reg\\_SetHysteresisVoltage, 34](#)  
[Acmp\\_Reg\\_SetInterruptEnableFlag, 34](#)

[Acmp\\_Reg\\_SetInvertMode, 35](#)  
[Acmp\\_Reg\\_SetLowPassFilter, 35](#)  
[Acmp\\_Reg\\_SetNegativeInputPin, 35](#)  
[Acmp\\_Reg\\_SetNegativePollingEnableFlag, 36](#)  
[Acmp\\_Reg\\_SetOutputPinEnableFlag, 36](#)  
[Acmp\\_Reg\\_SetOutputSelection, 37](#)  
[Acmp\\_Reg\\_SetPollingFreqDiv, 37](#)  
[Acmp\\_Reg\\_SetPollingSequence, 37](#)  
[Acmp\\_Reg\\_SetPositiveInputPin, 38](#)  
[Acmp\\_Reg\\_SetPositivePollingEnableFlag, 38](#)  
[Acmp\\_Reg\\_SetTriggerMode, 39](#)  
[Acmp\\_Reg\\_SetUsingLSIEnableFlag, 39](#)  
[Acmp\\_Reg\\_SetWindowMode, 40](#)

Acmp\_AnmuxType, [3](#)

[NegativeInputMux, 3](#)

[PositiveInputMux, 3](#)

Acmp\_CallbackType

[Acmp\\_Hal\\_Types.h, 54](#)

Acmp\_ChNumberType

[Acmp\\_Hal\\_Types.h, 54](#)

Acmp\_ComparatorType, [4](#)

[Callback, 4](#)

[ClockDivide, 5](#)

[FilterEnable, 5](#)

[FilterSampleCount, 5](#)

[HysteresisLevel, 5](#)

[HysteresisMode, 5](#)

[InterruptEn, 6](#)

[InverterEnable, 6](#)

[LpfBandwidth, 6](#)

[OutputSelect, 6](#)

[OutputTrigger, 6](#)

[PinState, 7](#)

[PowerEnable, 7](#)

[UsingLSIEnable, 7](#)

[WindowModeEnable, 7](#)

Acmp\_DacType, [8](#)

[State, 8](#)

[Voltage, 8](#)

[VoltageReferenceSource, 8](#)

Acmp\_FilterDivideType

[Acmp\\_Hal\\_Types.h, 54](#)

Acmp\_Hal.c, [40](#)

[Acmp\\_Hal\\_ClearInputFlags, 41](#)

[Acmp\\_Hal\\_ClearOutputFlags, 41](#)

[Acmp\\_Hal\\_Deinit, 42](#)

[Acmp\\_Hal\\_GetConfigAll, 42](#)

[Acmp\\_Hal\\_GetDefaultConfig, 43](#)

[Acmp\\_Hal\\_GetOutputData, 43](#)

[Acmp\\_Hal\\_GetOutputFlags, 44](#)

[Acmp\\_Hal\\_GetPollingData, 44](#)

[Acmp\\_Hal\\_GetPollingFlags, 44](#)

- Acmp\_Hal\_Init, 45
- Acmp\_Hal\_Reset, 45
- ISR, 46
- Acmp\_Hal.h, 46
  - Acmp\_Hal\_ClearInputFlags, 47
  - Acmp\_Hal\_ClearOutputFlags, 48
  - Acmp\_Hal\_Deinit, 48
  - Acmp\_Hal\_GetConfigAll, 49
  - Acmp\_Hal\_GetDefaultConfig, 49
  - Acmp\_Hal\_GetOutputData, 49
  - Acmp\_Hal\_GetOutputFlags, 50
  - Acmp\_Hal\_GetPollingData, 50
  - Acmp\_Hal\_GetPollingFlags, 51
  - Acmp\_Hal\_Init, 51
  - Acmp\_Hal\_Reset, 52
- Acmp\_Hal\_ClearInputFlags
  - Acmp\_Hal.c, 41
  - Acmp\_Hal.h, 47
- Acmp\_Hal\_ClearOutputFlags
  - Acmp\_Hal.c, 41
  - Acmp\_Hal.h, 48
- Acmp\_Hal\_Deinit
  - Acmp\_Hal.c, 42
  - Acmp\_Hal.h, 48
- Acmp\_Hal\_GetConfigAll
  - Acmp\_Hal.c, 42
  - Acmp\_Hal.h, 49
- Acmp\_Hal\_GetDefaultConfig
  - Acmp\_Hal.c, 43
  - Acmp\_Hal.h, 49
- Acmp\_Hal\_GetOutputData
  - Acmp\_Hal.c, 43
  - Acmp\_Hal.h, 49
- Acmp\_Hal\_GetOutputFlags
  - Acmp\_Hal.c, 44
  - Acmp\_Hal.h, 50
- Acmp\_Hal\_GetPollingData
  - Acmp\_Hal.c, 44
  - Acmp\_Hal.h, 50
- Acmp\_Hal\_GetPollingFlags
  - Acmp\_Hal.c, 44
  - Acmp\_Hal.h, 51
- Acmp\_Hal\_Init
  - Acmp\_Hal.c, 45
  - Acmp\_Hal.h, 51
- Acmp\_Hal\_Reset
  - Acmp\_Hal.c, 45
  - Acmp\_Hal.h, 52
- Acmp\_Hal\_Types.h, 52
  - Acmp\_CallbackType, 54
  - Acmp\_ChNumberType, 54
  - Acmp\_FilterDivideType, 54
  - Acmp\_HysteresisModeType, 55
  - Acmp\_HysteresisType, 55
  - Acmp\_InputPollingType, 55
  - Acmp\_LowPassFilterType, 56
  - Acmp\_OutputSelectType, 56
  - Acmp\_OutputTriggerType, 56
  - Acmp\_PollingClkDivideType, 57
  - Acmp\_VoltageReferenceType, 57
- Acmp\_HysteresisModeType
  - Acmp\_Hal\_Types.h, 55
- Acmp\_HysteresisType
  - Acmp\_Hal\_Types.h, 55
- Acmp\_InputPollingType
  - Acmp\_Hal\_Types.h, 55
- Acmp\_LowPassFilterType
  - Acmp\_Hal\_Types.h, 56
- Acmp\_ModuleType, 9
  - Comparator, 9
  - Dac, 9
  - Mux, 9
  - PollingMode, 9
- Acmp\_OutputSelectType
  - Acmp\_Hal\_Types.h, 56
- Acmp\_OutputTriggerType
  - Acmp\_Hal\_Types.h, 56
- Acmp\_PollingClkDivideType
  - Acmp\_Hal\_Types.h, 57
- Acmp\_PollingModeType, 10
  - HallAOutputCh, 10
  - HallBOutputCh, 10
  - HallCOutputCh, 11
  - HallOutputEnable, 11
  - Mode, 11
  - PollingClockDivide, 11
  - PollingSequence, 11
- Acmp\_Reg\_ClearInterruptStatus
  - AC784xx\_Acmp\_Reg.h, 14
- Acmp\_Reg\_ClearOutputStatus
  - AC784xx\_Acmp\_Reg.h, 15
- Acmp\_Reg\_ClearPollingStatus
  - AC784xx\_Acmp\_Reg.h, 15
- Acmp\_Reg\_GetBase
  - AC784xx\_Acmp\_Reg.h, 16
- Acmp\_Reg\_GetDACEnableFlag
  - AC784xx\_Acmp\_Reg.h, 16
- Acmp\_Reg\_GetDACOutput
  - AC784xx\_Acmp\_Reg.h, 16
- Acmp\_Reg\_GetDACReference
  - AC784xx\_Acmp\_Reg.h, 17
- Acmp\_Reg\_GetEnableFlag
  - AC784xx\_Acmp\_Reg.h, 17
- Acmp\_Reg\_GetFilterClockPrescaler
  - AC784xx\_Acmp\_Reg.h, 17
- Acmp\_Reg\_GetFilterEnableFlag
  - AC784xx\_Acmp\_Reg.h, 18
- Acmp\_Reg\_GetFilterLength
  - AC784xx\_Acmp\_Reg.h, 18
- Acmp\_Reg\_GetHallOutputEnableFlag
  - AC784xx\_Acmp\_Reg.h, 20
- Acmp\_Reg\_GetHallOutputA
  - AC784xx\_Acmp\_Reg.h, 19
- Acmp\_Reg\_GetHallOutputB
  - AC784xx\_Acmp\_Reg.h, 19
- Acmp\_Reg\_GetHallOutputC
  - AC784xx\_Acmp\_Reg.h, 19
- Acmp\_Reg\_GetHysteresisMode
  - AC784xx\_Acmp\_Reg.h, 20
- Acmp\_Reg\_GetHysteresisVoltage
  - AC784xx\_Acmp\_Reg.h, 20
- Acmp\_Reg\_GetInterruptEnableFlag

- AC784xx\_Acmp\_Reg.h, [21](#)
- Acmp\_Reg\_GetInterruptStatus
  - AC784xx\_Acmp\_Reg.h, [21](#)
- Acmp\_Reg\_GetInvertMode
  - AC784xx\_Acmp\_Reg.h, [22](#)
- Acmp\_Reg\_GetLowPassFilter
  - AC784xx\_Acmp\_Reg.h, [22](#)
- Acmp\_Reg\_GetNegativeInputPin
  - AC784xx\_Acmp\_Reg.h, [22](#)
- Acmp\_Reg\_GetNegativePollingEnableFlag
  - AC784xx\_Acmp\_Reg.h, [23](#)
- Acmp\_Reg\_GetOutputData
  - AC784xx\_Acmp\_Reg.h, [23](#)
- Acmp\_Reg\_GetOutputPinEnableFlag
  - AC784xx\_Acmp\_Reg.h, [23](#)
- Acmp\_Reg\_GetOutputSelection
  - AC784xx\_Acmp\_Reg.h, [24](#)
- Acmp\_Reg\_GetOutputStatus
  - AC784xx\_Acmp\_Reg.h, [24](#)
- Acmp\_Reg\_GetPollingData
  - AC784xx\_Acmp\_Reg.h, [25](#)
- Acmp\_Reg\_GetPollingFreqDiv
  - AC784xx\_Acmp\_Reg.h, [25](#)
- Acmp\_Reg\_GetPollingSequence
  - AC784xx\_Acmp\_Reg.h, [25](#)
- Acmp\_Reg\_GetPollingStatus
  - AC784xx\_Acmp\_Reg.h, [26](#)
- Acmp\_Reg\_GetPositiveInputPin
  - AC784xx\_Acmp\_Reg.h, [26](#)
- Acmp\_Reg\_GetPositivePollingEnableFlag
  - AC784xx\_Acmp\_Reg.h, [26](#)
- Acmp\_Reg\_GetTriggerMode
  - AC784xx\_Acmp\_Reg.h, [27](#)
- Acmp\_Reg\_GetUsingLSIEnableFlag
  - AC784xx\_Acmp\_Reg.h, [27](#)
- Acmp\_Reg\_GetWindowMode
  - AC784xx\_Acmp\_Reg.h, [28](#)
- Acmp\_Reg\_SetDACEnableFlag
  - AC784xx\_Acmp\_Reg.h, [28](#)
- Acmp\_Reg\_SetDACOutput
  - AC784xx\_Acmp\_Reg.h, [28](#)
- Acmp\_Reg\_SetDACReference
  - AC784xx\_Acmp\_Reg.h, [30](#)
- Acmp\_Reg\_SetEnableFlag
  - AC784xx\_Acmp\_Reg.h, [30](#)
- Acmp\_Reg\_SetFilterClockPrescaler
  - AC784xx\_Acmp\_Reg.h, [31](#)
- Acmp\_Reg\_SetFilterEnableFlag
  - AC784xx\_Acmp\_Reg.h, [31](#)
- Acmp\_Reg\_SetFilterLength
  - AC784xx\_Acmp\_Reg.h, [31](#)
- Acmp\_Reg\_SetHallOutputEnableFlag
  - AC784xx\_Acmp\_Reg.h, [33](#)
- Acmp\_Reg\_SetHallOutputA
  - AC784xx\_Acmp\_Reg.h, [32](#)
- Acmp\_Reg\_SetHallOutputB
  - AC784xx\_Acmp\_Reg.h, [32](#)
- Acmp\_Reg\_SetHallOutputC
  - AC784xx\_Acmp\_Reg.h, [33](#)
- Acmp\_Reg\_SetHysteresisMode
  - AC784xx\_Acmp\_Reg.h, [33](#)
- Acmp\_Reg\_SetHysteresisVoltage
  - AC784xx\_Acmp\_Reg.h, [34](#)
- Acmp\_Reg\_SetInterruptEnableFlag
  - AC784xx\_Acmp\_Reg.h, [34](#)
- Acmp\_Reg\_SetInvertMode
  - AC784xx\_Acmp\_Reg.h, [35](#)
- Acmp\_Reg\_SetLowPassFilter
  - AC784xx\_Acmp\_Reg.h, [35](#)
- Acmp\_Reg\_SetNegativeInputPin
  - AC784xx\_Acmp\_Reg.h, [35](#)
- Acmp\_Reg\_SetNegativePollingEnableFlag
  - AC784xx\_Acmp\_Reg.h, [36](#)
- Acmp\_Reg\_SetOutputPinEnableFlag
  - AC784xx\_Acmp\_Reg.h, [36](#)
- Acmp\_Reg\_SetOutputSelection
  - AC784xx\_Acmp\_Reg.h, [37](#)
- Acmp\_Reg\_SetPollingFreqDiv
  - AC784xx\_Acmp\_Reg.h, [37](#)
- Acmp\_Reg\_SetPollingSequence
  - AC784xx\_Acmp\_Reg.h, [37](#)
- Acmp\_Reg\_SetPositiveInputPin
  - AC784xx\_Acmp\_Reg.h, [38](#)
- Acmp\_Reg\_SetPositivePollingEnableFlag
  - AC784xx\_Acmp\_Reg.h, [38](#)
- Acmp\_Reg\_SetTriggerMode
  - AC784xx\_Acmp\_Reg.h, [39](#)
- Acmp\_Reg\_SetUsingLSIEnableFlag
  - AC784xx\_Acmp\_Reg.h, [39](#)
- Acmp\_Reg\_SetWindowMode
  - AC784xx\_Acmp\_Reg.h, [40](#)
- Acmp\_VoltageReferenceType
  - Acmp\_Hal\_Types.h, [57](#)
- Callback
  - Acmp\_ComparatorType, [4](#)
- ClockDivide
  - Acmp\_ComparatorType, [5](#)
- Comparator
  - Acmp\_ModuleType, [9](#)
- Dac
  - Acmp\_ModuleType, [9](#)
- FilterEnable
  - Acmp\_ComparatorType, [5](#)
- FilterSampleCount
  - Acmp\_ComparatorType, [5](#)
- HallAOutputCh
  - Acmp\_PollingModeType, [10](#)
- HallBOutputCh
  - Acmp\_PollingModeType, [10](#)
- HallCOutputCh
  - Acmp\_PollingModeType, [11](#)
- HallOutputEnable
  - Acmp\_PollingModeType, [11](#)
- HysteresisLevel
  - Acmp\_ComparatorType, [5](#)
- HysteresisMode
  - Acmp\_ComparatorType, [5](#)
- ISR

- Acmp\_Hal.c, [46](#)
- InterruptEn
  - Acmp\_ComparatorType, [6](#)
- InverterEnable
  - Acmp\_ComparatorType, [6](#)
- LpfBandwidth
  - Acmp\_ComparatorType, [6](#)
- Mode
  - Acmp\_PollingModeType, [11](#)
- Mux
  - Acmp\_ModuleType, [9](#)
- NegativeInputMux
  - Acmp\_AnmuxType, [3](#)
- OutputSelect
  - Acmp\_ComparatorType, [6](#)
- OutputTrigger
  - Acmp\_ComparatorType, [6](#)
- PinState
  - Acmp\_ComparatorType, [7](#)
- PollingClockDivide
  - Acmp\_PollingModeType, [11](#)
- PollingMode
  - Acmp\_ModuleType, [9](#)
- PollingSequence
  - Acmp\_PollingModeType, [11](#)
- PositiveInputMux
  - Acmp\_AnmuxType, [3](#)
- PowerEnable
  - Acmp\_ComparatorType, [7](#)
- State
  - Acmp\_DacType, [8](#)
- UsingLSIEnable
  - Acmp\_ComparatorType, [7](#)
- Voltage
  - Acmp\_DacType, [8](#)
- VoltageReferenceSource
  - Acmp\_DacType, [8](#)
- WindowModeEnable
  - Acmp\_ComparatorType, [7](#)