

AC784xx_DFP DMA

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	Dma_ChannelAddrType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	DestEndAddr	3
3.1.2.2	DestOffset	4
3.1.2.3	DestStartAddr	4
3.1.2.4	Length	4
3.1.2.5	SrcEndAddr	4
3.1.2.6	SrcOffset	4
3.1.2.7	SrcStartAddr	5
3.2	Dma_ChannelCBInfoType Struct Reference	5
3.2.1	Detailed Description	5
3.2.2	Member Data Documentation	5
3.2.2.1	ChannelId	5
3.2.2.2	DmaEvent	6
3.2.2.3	UserArgs	6
3.3	Dma_ChannelConfigType Struct Reference	6
3.3.1	Detailed Description	6

3.3.2	Member Data Documentation	6
3.3.2.1	HwChannelId	7
3.3.2.2	Priority	7
3.3.2.3	ReqSrc	7
3.3.2.4	VirtualChannelId	7
3.4	Dma_ChannelInfoType Struct Reference	7
3.4.1	Detailed Description	8
3.4.2	Member Data Documentation	8
3.4.2.1	Callback	8
3.4.2.2	ReqSrc	8
3.4.2.3	State	8
3.4.2.4	UserArgs	8
3.4.2.5	VirtualChannelId	8
3.5	Dma_ConfigType Struct Reference	9
3.5.1	Detailed Description	9
3.5.2	Member Data Documentation	9
3.5.2.1	ChannelCfg	9
3.5.2.2	ChannelCnt	9
3.6	Dma_TransferConfigType Struct Reference	9
3.6.1	Detailed Description	10
3.6.2	Member Data Documentation	10
3.6.2.1	Callback	10
3.6.2.2	CircularMode	10
3.6.2.3	DestEndAddr	11
3.6.2.4	DestOffset	11
3.6.2.5	DestStartAddr	11
3.6.2.6	DestUnit	11
3.6.2.7	IrqSrc	11
3.6.2.8	Length	12
3.6.2.9	SrcEndAddr	12
3.6.2.10	SrcOffset	12
3.6.2.11	SrcStartAddr	12
3.6.2.12	SrcUnit	12
3.6.2.13	TriggerMode	13
3.6.2.14	Type	13
3.6.2.15	UserArgs	13

4	File Documentation	14
4.1	AC784xx_API_Reference_Manual_DMA.pdf File Reference	14
4.2	AC784xx_Dma_Reg.h File Reference	14
4.2.1	Detailed Description	15
4.2.2	Function Documentation	15
4.2.2.1	Dma_Reg_ChannelHardRst()	16
4.2.2.2	Dma_Reg_ChannelWarmRst()	16
4.2.2.3	Dma_Reg_ClearDmaStatus()	16
4.2.2.4	Dma_Reg_FlushChannelFIFO()	17
4.2.2.5	Dma_Reg_GetChannelSource()	17
4.2.2.6	Dma_Reg_GetCircularMode()	18
4.2.2.7	Dma_Reg_GetDmaStatus()	18
4.2.2.8	Dma_Reg_GetDstTransferSize()	19
4.2.2.9	Dma_Reg_GetInterFIFODataLeftBytes()	19
4.2.2.10	Dma_Reg_GetSrcTransferSize()	20
4.2.2.11	Dma_Reg_GetTransferredBytes()	20
4.2.2.12	Dma_Reg_GetTransferLength()	21
4.2.2.13	Dma_Reg_SetChannel()	21
4.2.2.14	Dma_Reg_SetChannelDebug()	22
4.2.2.15	Dma_Reg_SetChannelPause()	22
4.2.2.16	Dma_Reg_SetChannelPriority()	23
4.2.2.17	Dma_Reg_SetChannelSource()	23
4.2.2.18	Dma_Reg_SetChannelTrigger()	24
4.2.2.19	Dma_Reg_SetCircularMode()	24
4.2.2.20	Dma_Reg_SetDestAddrOffSet()	25
4.2.2.21	Dma_Reg_SetDestEndAddr()	25
4.2.2.22	Dma_Reg_SetDestStartAddr()	26
4.2.2.23	Dma_Reg_SetDstTransferSize()	26
4.2.2.24	Dma_Reg_SetErrorInterrupt()	27
4.2.2.25	Dma_Reg_SetFIFOFastFunction()	27
4.2.2.26	Dma_Reg_SetFinishInterrupt()	28

4.2.2.27	Dma_Reg_SetInterrupt()	28
4.2.2.28	Dma_Reg_SetSrcAddrOffSet()	29
4.2.2.29	Dma_Reg_SetSrcEndAddr()	29
4.2.2.30	Dma_Reg_SetSrcStartAddr()	30
4.2.2.31	Dma_Reg_SetSrcTransferSize()	30
4.2.2.32	Dma_Reg_SetTransferLength()	31
4.2.2.33	Dma_Reg_TopHardRst()	31
4.2.2.34	Dma_Reg_TopWarmRst()	32
4.3	Dma_Hal.c File Reference	32
4.3.1	Detailed Description	34
4.3.2	Enumeration Type Documentation	34
4.3.2.1	Dma_StateType	34
4.3.3	Function Documentation	34
4.3.3.1	Dma_Hal_ChReset()	34
4.3.3.2	Dma_Hal_ConfigCh()	35
4.3.3.3	Dma_Hal_Deinit()	35
4.3.3.4	Dma_Hal_EnableChIrq()	36
4.3.3.5	Dma_Hal_GetChIdByReqSrc()	36
4.3.3.6	Dma_Hal_GetChStatus()	37
4.3.3.7	Dma_Hal_GetTransBytes()	37
4.3.3.8	Dma_Hal_Init()	37
4.3.3.9	Dma_Hal_StartCh()	38
4.3.3.10	Dma_Hal_StopCh()	38
4.3.3.11	Dma_Hal_UpdateChAddr()	39
4.3.3.12	ISR() [1/16]	39
4.3.3.13	ISR() [2/16]	40
4.3.3.14	ISR() [3/16]	40
4.3.3.15	ISR() [4/16]	40
4.3.3.16	ISR() [5/16]	41
4.3.3.17	ISR() [6/16]	41
4.3.3.18	ISR() [7/16]	41

4.3.3.19	ISR() [8/16]	42
4.3.3.20	ISR() [9/16]	42
4.3.3.21	ISR() [10/16]	42
4.3.3.22	ISR() [11/16]	43
4.3.3.23	ISR() [12/16]	43
4.3.3.24	ISR() [13/16]	43
4.3.3.25	ISR() [14/16]	44
4.3.3.26	ISR() [15/16]	44
4.3.3.27	ISR() [16/16]	44
4.4	Dma_Hal.h File Reference	45
4.4.1	Detailed Description	46
4.4.2	Function Documentation	46
4.4.2.1	Dma_Hal_ChReset()	46
4.4.2.2	Dma_Hal_ConfigCh()	47
4.4.2.3	Dma_Hal_Deinit()	47
4.4.2.4	Dma_Hal_EnableChIrq()	48
4.4.2.5	Dma_Hal_GetChIdByReqSrc()	48
4.4.2.6	Dma_Hal_GetChStatus()	49
4.4.2.7	Dma_Hal_GetTransBytes()	49
4.4.2.8	Dma_Hal_Init()	50
4.4.2.9	Dma_Hal_StartCh()	50
4.4.2.10	Dma_Hal_StopCh()	51
4.4.2.11	Dma_Hal_UpdateChAddr()	51
4.4.2.12	ISR() [1/16]	52
4.4.2.13	ISR() [2/16]	52
4.4.2.14	ISR() [3/16]	53
4.4.2.15	ISR() [4/16]	53
4.4.2.16	ISR() [5/16]	53
4.4.2.17	ISR() [6/16]	54
4.4.2.18	ISR() [7/16]	54
4.4.2.19	ISR() [8/16]	54
4.4.2.20	ISR() [9/16]	55
4.4.2.21	ISR() [10/16]	55
4.4.2.22	ISR() [11/16]	55
4.4.2.23	ISR() [12/16]	56
4.4.2.24	ISR() [13/16]	56
4.4.2.25	ISR() [14/16]	56
4.4.2.26	ISR() [15/16]	57
4.4.2.27	ISR() [16/16]	57

Chapter 1

Class Index

1.1 Class List

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

Dma_ChannelAddrType	
Dma channel address type	3
Dma_ChannelCBInfoType	
Dma channel interrupt callback type	5
Dma_ChannelConfigType	
The user configuration structure for the an DMA driver channel	6
Dma_ChannelInfoType	
Dma channel information type	7
Dma_ConfigType	
The user configuration structure for the an DMA driver channel	9
Dma_TransferConfigType	
The user configuration structure for the an DMA channel	9

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_DMA.pdf	14
AC784xx_Dma_Reg.h	
Dma register interface	14
Dma_Hal.c	
This file provides extern Dma Hal API implement	32
Dma_Hal.h	
Dma hal interface for SDK	45

Chapter 3

Class Documentation

3.1 Dma_ChannelAddrType Struct Reference

Dma channel address type.

```
#include <Dma_Hal.h>
```

Public Attributes

- uint16 [SrcOffset](#)
- uint16 [DestOffset](#)
- uint32 [SrcStartAddr](#)
- uint32 [DestStartAddr](#)
- uint32 [SrcEndAddr](#)
- uint32 [DestEndAddr](#)
- uint16 [Length](#)

3.1.1 Detailed Description

Dma channel address type.

Definition at line 78 of file Dma_Hal.h.

3.1.2 Member Data Documentation

3.1.2.1 DestEndAddr

```
uint32 Dma_ChannelAddrType::DestEndAddr
```

Transfer destination end address

Definition at line 85 of file Dma_Hal.h.

3.1.2.2 DestOffset

```
uint16 Dma_ChannelAddrType::DestOffset
```

Destination address offset

Definition at line 81 of file Dma_Hal.h.

3.1.2.3 DestStartAddr

```
uint32 Dma_ChannelAddrType::DestStartAddr
```

Transfer destination start address

Definition at line 83 of file Dma_Hal.h.

3.1.2.4 Length

```
uint16 Dma_ChannelAddrType::Length
```

Transfer length

Definition at line 86 of file Dma_Hal.h.

3.1.2.5 SrcEndAddr

```
uint32 Dma_ChannelAddrType::SrcEndAddr
```

Transfer source end address

Definition at line 84 of file Dma_Hal.h.

3.1.2.6 SrcOffset

```
uint16 Dma_ChannelAddrType::SrcOffset
```

Source address offset

Definition at line 80 of file Dma_Hal.h.

3.1.2.7 SrcStartAddr

```
uint32 Dma_ChannelAddrType::SrcStartAddr
```

Transfer source start address

Definition at line 82 of file Dma_Hal.h.

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

- [Dma_Hal.h](#)

3.2 Dma_ChannelCBInfoType Struct Reference

Dma channel interrupt callback type.

```
#include <Dma_Hal.h>
```

Public Attributes

- uint8 [ChannelId](#)
- uint32 [DmaEvent](#)
- void * [UserArgs](#)

3.2.1 Detailed Description

Dma channel interrupt callback type.

Definition at line 67 of file Dma_Hal.h.

3.2.2 Member Data Documentation

3.2.2.1 ChannelId

```
uint8 Dma_ChannelCBInfoType::ChannelId
```

Dma channel id

Definition at line 69 of file Dma_Hal.h.

3.2.2.2 DmaEvent

```
uint32 Dma_ChannelCBInfoType::DmaEvent
```

DMA finish or error event, use DMA_FINISH_EVENT or DMA_ERROR_EVENT to parse event

Definition at line 70 of file Dma_Hal.h.

3.2.2.3 UserArgs

```
void* Dma_ChannelCBInfoType::UserArgs
```

User-defined callback parameters

Definition at line 71 of file Dma_Hal.h.

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

- [Dma_Hal.h](#)

3.3 Dma_ChannelConfigType Struct Reference

The user configuration structure for the an DMA driver channel.

```
#include <Dma_Hal.h>
```

Public Attributes

- uint8 [HwChannelId](#)
- uint8 [VirtualChannelId](#)
- Dma_ChannelPriorityType [Priority](#)
- Dma_RequestSourceType [ReqSrc](#)

3.3.1 Detailed Description

The user configuration structure for the an DMA driver channel.

Definition at line 114 of file Dma_Hal.h.

3.3.2 Member Data Documentation

3.3.2.1 HwChannelId

```
uint8 Dma_ChannelConfigType::HwChannelId
```

Dma channel ID

Definition at line 116 of file Dma_Hal.h.

3.3.2.2 Priority

```
Dma_ChannelPriorityType Dma_ChannelConfigType::Priority
```

DMA channel priority

Definition at line 118 of file Dma_Hal.h.

3.3.2.3 ReqSrc

```
Dma_RequestSourceType Dma_ChannelConfigType::ReqSrc
```

DMA channel transfer type

Definition at line 119 of file Dma_Hal.h.

3.3.2.4 VirtualChannelId

```
uint8 Dma_ChannelConfigType::VirtualChannelId
```

Dma channel ID

Definition at line 117 of file Dma_Hal.h.

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

- [Dma_Hal.h](#)

3.4 Dma_ChannelInfoType Struct Reference

Dma channel information type.

Public Attributes

- Dma_RequestSourceType [ReqSrc](#)
- [Dma_StateType](#) State
- uint8 [VirtualChannelId](#)
- Hal_CallbackType [Callback](#)
- void * [UserArgs](#)

3.4.1 Detailed Description

Dma channel information type.

Definition at line 71 of file Dma_Hal.c.

3.4.2 Member Data Documentation

3.4.2.1 Callback

`Hal_CallbackType Dma_ChannelInfoType::Callback`

Definition at line 76 of file Dma_Hal.c.

3.4.2.2 ReqSrc

`Dma_RequestSourceType Dma_ChannelInfoType::ReqSrc`

Definition at line 73 of file Dma_Hal.c.

3.4.2.3 State

`Dma_StateType Dma_ChannelInfoType::State`

Definition at line 74 of file Dma_Hal.c.

3.4.2.4 UserArgs

`void* Dma_ChannelInfoType::UserArgs`

Definition at line 77 of file Dma_Hal.c.

3.4.2.5 VirtualChannelId

`uint8 Dma_ChannelInfoType::VirtualChannelId`

Dma channel ID

Definition at line 75 of file Dma_Hal.c.

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

- [Dma_Hal.c](#)

3.5 Dma_ConfigType Struct Reference

The user configuration structure for the an DMA driver channel.

```
#include <Dma_Hal.h>
```

Public Attributes

- uint8 [ChannelCnt](#)
- const [Dma_ChannelConfigType](#) * [ChannelCfg](#)

3.5.1 Detailed Description

The user configuration structure for the an DMA driver channel.

Definition at line 125 of file Dma_Hal.h.

3.5.2 Member Data Documentation

3.5.2.1 ChannelCfg

```
const Dma\_ChannelConfigType* Dma_ConfigType::ChannelCfg
```

DMA channel cfg

Definition at line 128 of file Dma_Hal.h.

3.5.2.2 ChannelCnt

```
uint8 Dma_ConfigType::ChannelCnt
```

Dma channel counter

Definition at line 127 of file Dma_Hal.h.

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

- [Dma_Hal.h](#)

3.6 Dma_TransferConfigType Struct Reference

The user configuration structure for the an DMA channel.

```
#include <Dma_Hal.h>
```

Public Attributes

- boolean [TriggerMode](#)
- boolean [CircularMode](#)
- uint8 [IrqSrc](#)
- uint16 [SrcOffset](#)
- uint16 [DestOffset](#)
- uint16 [Length](#)
- Dma_TransferUnitType [SrcUnit](#)
- Dma_TransferUnitType [DestUnit](#)
- Dma_TransferType [Type](#)
- uint32 [SrcStartAddr](#)
- uint32 [DestStartAddr](#)
- uint32 [SrcEndAddr](#)
- uint32 [DestEndAddr](#)
- Hal_CallbackType [Callback](#)
- void * [UserArgs](#)

3.6.1 Detailed Description

The user configuration structure for the an DMA channel.

Definition at line 92 of file Dma_Hal.h.

3.6.2 Member Data Documentation

3.6.2.1 Callback

```
Hal_CallbackType Dma_TransferConfigType::Callback
```

iUser-defined callback function

Definition at line 107 of file Dma_Hal.h.

3.6.2.2 CircularMode

```
boolean Dma_TransferConfigType::CircularMode
```

Enable/Disable the circular mode

Definition at line 95 of file Dma_Hal.h.

3.6.2.3 DestEndAddr

```
uint32 Dma_TransferConfigType::DestEndAddr
```

Transfer destination end address

Definition at line 106 of file Dma_Hal.h.

3.6.2.4 DestOffset

```
uint16 Dma_TransferConfigType::DestOffset
```

Destination address offset

Definition at line 98 of file Dma_Hal.h.

3.6.2.5 DestStartAddr

```
uint32 Dma_TransferConfigType::DestStartAddr
```

Transfer destination start address

Definition at line 104 of file Dma_Hal.h.

3.6.2.6 DestUnit

```
Dma_TransferUnitType Dma_TransferConfigType::DestUnit
```

DMA channel destination transfer unit size

Definition at line 101 of file Dma_Hal.h.

3.6.2.7 IrqSrc

```
uint8 Dma_TransferConfigType::IrqSrc
```

Enable/Disable irq, half finish irq only valid for 7842x and 7843x

Definition at line 96 of file Dma_Hal.h.

3.6.2.8 Length

```
uint16 Dma_TransferConfigType::Length
```

Transfer length

Definition at line 99 of file Dma_Hal.h.

3.6.2.9 SrcEndAddr

```
uint32 Dma_TransferConfigType::SrcEndAddr
```

Transfer source end address

Definition at line 105 of file Dma_Hal.h.

3.6.2.10 SrcOffset

```
uint16 Dma_TransferConfigType::SrcOffset
```

Source address offset

Definition at line 97 of file Dma_Hal.h.

3.6.2.11 SrcStartAddr

```
uint32 Dma_TransferConfigType::SrcStartAddr
```

Transfer source start address

Definition at line 103 of file Dma_Hal.h.

3.6.2.12 SrcUnit

```
Dma_TransferUnitType Dma_TransferConfigType::SrcUnit
```

DMA channel source transfer unit size

Definition at line 100 of file Dma_Hal.h.

3.6.2.13 TriggerMode

```
boolean Dma_TransferConfigType::TriggerMode
```

Enable/Disable the trigger mode

Definition at line 94 of file Dma_Hal.h.

3.6.2.14 Type

```
Dma_TransferType Dma_TransferConfigType::Type
```

DMA channel transfer type

Definition at line 102 of file Dma_Hal.h.

3.6.2.15 UserArgs

```
void* Dma_TransferConfigType::UserArgs
```

User-defined callback parameters

Definition at line 108 of file Dma_Hal.h.

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

- [Dma_Hal.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_DMA.pdf File Reference

4.2 AC784xx_Dma_Reg.h File Reference

dma register interface

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

Functions

- LOCAL_INLINE void [Dma_Reg_SetChannel](#) (uint8 ChannelId, boolean En)
Enable/Disable DMA channel.
- LOCAL_INLINE void [Dma_Reg_TopWarmRst](#) (void)
DMA Top warm reset.
- LOCAL_INLINE void [Dma_Reg_TopHardRst](#) (void)
DMA Top Hard reset.
- LOCAL_INLINE void [Dma_Reg_ChannelWarmRst](#) (uint8 ChannelId)
DMA channel warm reset.
- LOCAL_INLINE void [Dma_Reg_ChannelHardRst](#) (uint8 ChannelId)
DMA channel hard reset.
- LOCAL_INLINE void [Dma_Reg_SetSrcStartAddr](#) (uint8 ChannelId, uint32 StartAddr)
Set DMA channel Source start address.
- LOCAL_INLINE void [Dma_Reg_SetSrcEndAddr](#) (uint8 ChannelId, uint32 EndAddr)
Set DMA channel Source end address.
- LOCAL_INLINE void [Dma_Reg_SetDestStartAddr](#) (uint8 ChannelId, uint32 StartAddr)
Set DMA channel destination start address.
- LOCAL_INLINE void [Dma_Reg_SetDestEndAddr](#) (uint8 ChannelId, uint32 EndAddr)
Set DMA channel destination end address.
- LOCAL_INLINE void [Dma_Reg_SetSrcTransferSize](#) (uint8 ChannelId, uint32 TransferSize)
Set DMA channel Source transfer size.
- LOCAL_INLINE void [Dma_Reg_SetDstTransferSize](#) (uint8 ChannelId, uint32 TransferSize)
Set DMA channel destination transfer size.
- LOCAL_INLINE void [Dma_Reg_SetSrcAddrOffset](#) (uint8 ChannelId, uint16 Offset)
Set DMA channel Source address offset.

- LOCAL_INLINE void [Dma_Reg_SetDestAddrOffSet](#) (uint8 ChannelId, uint16 Offset)
Set DMA channel destination address offset.
- LOCAL_INLINE void [Dma_Reg_SetTransferLength](#) (uint8 ChannelId, uint32 BytesLength)
Set DMA channel transfer length by bytes.
- LOCAL_INLINE uint32 [Dma_Reg_GetTransferLength](#) (uint8 ChannelId)
Get DMA channel total transfer length by bytes.
- LOCAL_INLINE uint32 [Dma_Reg_GetTransferredBytes](#) (uint8 ChannelId)
Get number of bytes transferred by DMA channel.
- LOCAL_INLINE void [Dma_Reg_SetChannelPriority](#) (uint8 ChannelId, uint32 Priority)
Set DMA channel priority.
- LOCAL_INLINE void [Dma_Reg_SetCircularMode](#) (uint8 ChannelId, boolean En)
Set DMA channel circular mode.
- LOCAL_INLINE void [Dma_Reg_SetFIFOFastFunction](#) (uint8 ChannelId, boolean En)
Set DMA channel FIFO move fast function.
- LOCAL_INLINE uint32 [Dma_Reg_GetInterFIFODataLeftBytes](#) (uint8 ChannelId)
Get the number of the data left in the DMA channel's internal FIFO.
- LOCAL_INLINE void [Dma_Reg_FlushChannelFIFO](#) (uint8 ChannelId)
Flush DMA channel data.
- LOCAL_INLINE void [Dma_Reg_SetFinishInterrupt](#) (uint8 ChannelId, boolean En)
Enable/Disable DMA channel transfer finish interrupt.
- LOCAL_INLINE void [Dma_Reg_SetErrorInterrupt](#) (uint8 ChannelId, boolean En)
Enable/Disable DMA channel Error interrupt.
- LOCAL_INLINE void [Dma_Reg_SetChannelTrigger](#) (uint8 ChannelId, boolean En)
Enable or disable DMA channel trigger.
- LOCAL_INLINE void [Dma_Reg_SetChannelSource](#) (uint8 ChannelId, uint32 Source)
Configures the DMA request for the DMA channel.
- LOCAL_INLINE void [Dma_Reg_SetChannelDebug](#) (uint8 ChannelId, boolean En)
Set DMA debug function.
- LOCAL_INLINE void [Dma_Reg_SetChannelPause](#) (uint8 ChannelId, boolean En)
Set DMA channel transfer pause/resume.
- LOCAL_INLINE uint32 [Dma_Reg_GetDmaStatus](#) (uint8 ChannelId)
get DMA channel status
- LOCAL_INLINE void [Dma_Reg_ClearDmaStatus](#) (uint8 ChannelId)
clear DMA channel status
- LOCAL_INLINE uint32 [Dma_Reg_GetCircularMode](#) (uint8 ChannelId)
Get DMA channel circular mode.
- LOCAL_INLINE uint32 [Dma_Reg_GetSrcTransferSize](#) (uint8 ChannelId)
Get DMA channel Source transfer size.
- LOCAL_INLINE uint32 [Dma_Reg_GetDstTransferSize](#) (uint8 ChannelId)
Get DMA channel destination transfer size.
- LOCAL_INLINE uint32 [Dma_Reg_GetChannelSource](#) (uint8 ChannelId)
Get the DMA request source.
- LOCAL_INLINE void [Dma_Reg_SetInterrupt](#) (uint8 ChannelId, uint32 Irq)
Enable/Disable DMA channel transfer half finish interrupt.

4.2.1 Detailed Description

dma register interface

4.2.2 Function Documentation

4.2.2.1 Dma_Reg_ChannelHardRst()

```
LOCAL_INLINE void Dma_Reg_ChannelHardRst (
    uint8 ChannelId )
```

DMA channel hard reset.

Note

Function ID : DES_MCL_API_164

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

void

Definition at line 124 of file AC784xx_Dma_Reg.h.

4.2.2.2 Dma_Reg_ChannelWarmRst()

```
LOCAL_INLINE void Dma_Reg_ChannelWarmRst (
    uint8 ChannelId )
```

DMA channel warm reset.

Note

Function ID : DES_MCL_API_163

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

void

Definition at line 113 of file AC784xx_Dma_Reg.h.

4.2.2.3 Dma_Reg_ClearDmaStatus()

```
LOCAL_INLINE void Dma_Reg_ClearDmaStatus (
    uint8 ChannelId )
```

clear DMA channel status

Note

Function ID : DES_MCL_API_188

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

void

Definition at line 420 of file AC784xx_Dma_Reg.h.

4.2.2.4 Dma_Reg_FlushChannelFIFO()

```
LOCAL_INLINE void Dma_Reg_FlushChannelFIFO (
    uint8 ChannelId )
```

Flush DMA channel data.

Note

Function ID : DES_MCL_API_180

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

void

Definition at line 318 of file AC784xx_Dma_Reg.h.

4.2.2.5 Dma_Reg_GetChannelSource()

```
LOCAL_INLINE uint32 Dma_Reg_GetChannelSource (
    uint8 ChannelId )
```

Get the DMA request source.

Note

Function ID : DES_MCL_API_191

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA request Source

Definition at line 465 of file AC784xx_Dma_Reg.h.

4.2.2.6 Dma_Reg_GetCircularMode()

```
LOCAL_INLINE uint32 Dma_Reg_GetCircularMode (  
    uint8 ChannelId )
```

Get DMA channel circular mode.

Note

Function ID : DES_MCL_API_188

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel circular mode

Definition at line 431 of file AC784xx_Dma_Reg.h.

4.2.2.7 Dma_Reg_GetDmaStatus()

```
LOCAL_INLINE uint32 Dma_Reg_GetDmaStatus (  
    uint8 ChannelId )
```

get DMA channel status

Note

Function ID : DES_MCL_API_187

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel status

Definition at line 405 of file AC784xx_Dma_Reg.h.

4.2.2.8 Dma_Reg_GetDstTransferSize()

```
LOCAL_INLINE uint32 Dma_Reg_GetDstTransferSize (
    uint8 ChannelId )
```

Get DMA channel destination transfer size.

Note

Function ID : DES_MCL_API_190

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel destination transfer size

Definition at line 454 of file AC784xx_Dma_Reg.h.

4.2.2.9 Dma_Reg_GetInterFIFODataLeftBytes()

```
LOCAL_INLINE uint32 Dma_Reg_GetInterFIFODataLeftBytes (
    uint8 ChannelId )
```

Get the number of the data left in the DMA channel's internal FIFO.

Note

Function ID : DES_MCL_API_179

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA Channel FIFO left data bytes number

Definition at line 307 of file AC784xx_Dma_Reg.h.

4.2.2.10 Dma_Reg_GetSrcTransferSize()

```
LOCAL_INLINE uint32 Dma_Reg_GetSrcTransferSize (
    uint8 ChannelId )
```

Get DMA channel Source transfer size.

Note

Function ID : DES_MCL_API_189

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel Source transfer size.

Definition at line 443 of file AC784xx_Dma_Reg.h.

4.2.2.11 Dma_Reg_GetTransferredBytes()

```
LOCAL_INLINE uint32 Dma_Reg_GetTransferredBytes (
    uint8 ChannelId )
```

Get number of bytes transfered by DMA channel.

Note

Function ID : DES_MCL_API_175

Parameters

in	<i>Channel↵ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel transfered bytes number

Definition at line 258 of file AC784xx_Dma_Reg.h.

4.2.2.12 Dma_Reg_GetTransferLength()

```
LOCAL_INLINE uint32 Dma_Reg_GetTransferLength (
    uint8 ChannelId )
```

Get DMA channel total transfer length by bytes.

Note

Function ID : DES_MCL_API_174

Parameters

in	<i>Channel↵ Id</i>	dma channel id
----	------------------------	----------------

Returns

DMA channel transfer length.

Definition at line 247 of file AC784xx_Dma_Reg.h.

4.2.2.13 Dma_Reg_SetChannel()

```
LOCAL_INLINE void Dma_Reg_SetChannel (
    uint8 ChannelId,
    boolean En )
```

Enable/Disable DMA channel.

Note

Function ID : DES_MCL_API_160

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable/Disable DMA channel

Returns

void

Definition at line 80 of file AC784xx_Dma_Reg.h.

4.2.2.14 Dma_Reg_SetChannelDebug()

```
LOCAL_INLINE void Dma_Reg_SetChannelDebug (
    uint8 ChannelId,
    boolean En )
```

Set DMA debug function.

Note

Function ID : DES_MCL_API_185

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable or disable debug fuction

Returns

void

Definition at line 382 of file AC784xx_Dma_Reg.h.

4.2.2.15 Dma_Reg_SetChannelPause()

```
LOCAL_INLINE void Dma_Reg_SetChannelPause (
    uint8 ChannelId,
    boolean En )
```

Set DMA channel transfer pause/resume.

Note

Function ID : DES_MCL_API_186

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable/disable DMA pause

Returns

void

Definition at line 394 of file AC784xx_Dma_Reg.h.

4.2.2.16 Dma_Reg_SetChannelPriority()

```
LOCAL_INLINE void Dma_Reg_SetChannelPriority (
    uint8 ChannelId,
    uint32 Priority )
```

Set DMA channel priority.

Note

Function ID : DES_MCL_API_176

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Priority</i>	DMA chanenl priority

Returns

void

Definition at line 270 of file AC784xx_Dma_Reg.h.

4.2.2.17 Dma_Reg_SetChannelSource()

```
LOCAL_INLINE void Dma_Reg_SetChannelSource (
    uint8 ChannelId,
    uint32 Source )
```

Configures the DMA request for the DMA channel.

Note

Function ID : DES_MCL_API_184

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Source</i>	DMA request Source

Returns

void

Definition at line 369 of file AC784xx_Dma_Reg.h.

4.2.2.18 Dma_Reg_SetChannelTrigger()

```
LOCAL_INLINE void Dma_Reg_SetChannelTrigger (
    uint8 ChannelId,
    boolean En )
```

Enable or disable DMA channel trigger.

Note

Function ID : DES_MCL_API_183

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable or disable DMA channel trigger

Returns

void

Definition at line 356 of file AC784xx_Dma_Reg.h.

4.2.2.19 Dma_Reg_SetCircularMode()

```
LOCAL_INLINE void Dma_Reg_SetCircularMode (
    uint8 ChannelId,
    boolean En )
```

Set DMA channel circular mode.

Note

Function ID : DES_MCL_API_177

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable/disable DMA channel circular mode

Returns

void

Definition at line 283 of file AC784xx_Dma_Reg.h.

4.2.2.20 Dma_Reg_SetDestAddrOffset()

```
LOCAL_INLINE void Dma_Reg_SetDestAddrOffset (
    uint8 ChannelId,
    uint16 Offset )
```

Set DMA channel destination address offset.

Note

Function ID : DES_MCL_API_172

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Offset</i>	address offset

Returns

void

Definition at line 222 of file AC784xx_Dma_Reg.h.

4.2.2.21 Dma_Reg_SetDestEndAddr()

```
LOCAL_INLINE void Dma_Reg_SetDestEndAddr (
    uint8 ChannelId,
    uint32 EndAddr )
```

Set DMA channel destination end address.

Note

Function ID : DES_MCL_API_168

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>EndAddr</i>	end address

Returns

void

Definition at line 173 of file AC784xx_Dma_Reg.h.

4.2.2.22 Dma_Reg_SetDestStartAddr()

```
LOCAL_INLINE void Dma_Reg_SetDestStartAddr (
    uint8 ChannelId,
    uint32 StartAddr )
```

Set DMA channel destination start address.

Note

Function ID : DES_MCL_API_167

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>StartAddr</i>	start address

Returns

void

Definition at line 161 of file AC784xx_Dma_Reg.h.

4.2.2.23 Dma_Reg_SetDstTransferSize()

```
LOCAL_INLINE void Dma_Reg_SetDstTransferSize (
    uint8 ChannelId,
    uint32 TransferSize )
```

Set DMA channel destination transfer size.

Note

Function ID : DES_MCL_API_170

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>TransferSize</i>	transfer size

Returns

void

Definition at line 197 of file AC784xx_Dma_Reg.h.

4.2.2.24 Dma_Reg_SetErrorInterrupt()

```
LOCAL_INLINE void Dma_Reg_SetErrorInterrupt (
    uint8 ChannelId,
    boolean En )
```

Enable/Disable DMA channel Error interrupt.

Note

Function ID : DES_MCL_API_182

Parameters

in	<i>Channel↔ Id</i>	dma channel id
in	<i>En</i>	Enable/disable DMA channel Error interrupt

Returns

void

Definition at line 343 of file AC784xx_Dma_Reg.h.

4.2.2.25 Dma_Reg_SetFIFOFastFunction()

```
LOCAL_INLINE void Dma_Reg_SetFIFOFastFunction (
    uint8 ChannelId,
    boolean En )
```

Set DMA channel FIFO move fast function.

Note

Function ID : DES_MCL_API_178

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable or disable move data to FIFO early

Returns

void

Definition at line 296 of file AC784xx_Dma_Reg.h.

4.2.2.26 Dma_Reg_SetFinishInterrupt()

```
LOCAL_INLINE void Dma_Reg_SetFinishInterrupt (
    uint8 ChannelId,
    boolean En )
```

Enable/Disable DMA channel transfer finish interrupt.

Note

Function ID : DES_MCL_API_181

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>En</i>	Enable/disable DMA channel finish interrupt

Returns

void

Definition at line 330 of file AC784xx_Dma_Reg.h.

4.2.2.27 Dma_Reg_SetInterrupt()

```
LOCAL_INLINE void Dma_Reg_SetInterrupt (
    uint8 ChannelId,
    uint32 Irq )
```

Enable/Disable DMA channel transfer half finish interrupt.

Note

Function ID : DES_MCL_API_182

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Irq:interrupt</i>	

Returns

void

Definition at line 493 of file AC784xx_Dma_Reg.h.

4.2.2.28 Dma_Reg_SetSrcAddrOffSet()

```
LOCAL_INLINE void Dma_Reg_SetSrcAddrOffSet (
    uint8 ChannelId,
    uint16 Offset )
```

Set DMA channel Source address offset.

Note

Function ID : DES_MCL_API_171

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Offset</i>	address offset

Returns

void

Definition at line 209 of file AC784xx_Dma_Reg.h.

4.2.2.29 Dma_Reg_SetSrcEndAddr()

```
LOCAL_INLINE void Dma_Reg_SetSrcEndAddr (
    uint8 ChannelId,
    uint32 EndAddr )
```

Set DMA channel Source end address.

Note

Function ID : DES_MCL_API_166

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>EndAddr</i>	end address

Returns

void

Definition at line 149 of file AC784xx_Dma_Reg.h.

4.2.2.30 Dma_Reg_SetSrcStartAddr()

```
LOCAL_INLINE void Dma_Reg_SetSrcStartAddr (
    uint8 ChannelId,
    uint32 StartAddr )
```

Set DMA channel Source start address.

Note

Function ID : DES_MCL_API_165

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>StartAddr</i>	start address

Returns

void

Definition at line 137 of file AC784xx_Dma_Reg.h.

4.2.2.31 Dma_Reg_SetSrcTransferSize()

```
LOCAL_INLINE void Dma_Reg_SetSrcTransferSize (
    uint8 ChannelId,
    uint32 TransferSize )
```

Set DMA channel Source transfer size.

Note

Function ID : DES_MCL_API_169

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>TransferSize</i>	transfer size

Returns

void

Definition at line 185 of file AC784xx_Dma_Reg.h.

4.2.2.32 Dma_Reg_SetTransferLength()

```
LOCAL_INLINE void Dma_Reg_SetTransferLength (
    uint8 ChannelId,
    uint32 BytesLength )
```

Set DMA channel transfer length by bytes.

Note

Function ID : DES_MCL_API_173

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>BytesLength</i>	address offset

Returns

void

Definition at line 235 of file AC784xx_Dma_Reg.h.

4.2.2.33 Dma_Reg_TopHardRst()

```
LOCAL_INLINE void Dma_Reg_TopHardRst (
    void )
```

DMA Top Hard reset.

Note

Function ID : DES_MCL_API_162

Returns

void

Definition at line 101 of file AC784xx_Dma_Reg.h.

4.2.2.34 Dma_Reg_TopWarmRst()

```
LOCAL_INLINE void Dma_Reg_TopWarmRst (
    void )
```

DMA Top warm reset.

Note

Function ID : DES_MCL_API_161

Returns

void

Definition at line 91 of file AC784xx_Dma_Reg.h.

4.3 Dma_Hal.c File Reference

This file provides extern Dma Hal API implement.

```
#include "Ckgen_Hal.h"
#include "Core_Hal.h"
#include "Dma_Hal.h"
#include "AC784xx_Dma_Reg.h"
#include "OsIf_Critical.h"
```

Classes

- struct [Dma_ChannelInfoType](#)
Dma channel information type.

Enumerations

- enum [Dma_StateType](#) {
 [DMA_STATE_INITED](#) = 0x0U, [DMA_STATE_UNINITED](#), [DMA_STATE_CONFIGURED](#), [DMA_STATE_RUNNING](#),
 [DMA_STATE_ERROR](#) }
Dma state machine.

Functions

- void [Dma_Hal_Init](#) (const [Dma_ConfigType](#) *ConfigPtr)
Initializes the DMA module.
- void [Dma_Hal_Deinit](#) (void)
De-initializes the DMA module.
- Hal_StatusType [Dma_Hal_ConfigCh](#) (uint8 ChannelId, const [Dma_TransferConfigType](#) *ConfigPtr)
Configures data transfer with DMA.
- void [Dma_Hal_UpdateChAddr](#) (uint8 ChannelId, const [Dma_ChannelAddrType](#) *ChannelAddr)
Update Channel source/destination address and offset.
- void [Dma_Hal_EnableChIrq](#) (uint8 ChannelId, uint8 IrqSrc)
disable/enable the channel interrupt.
- Hal_StatusType [Dma_Hal_GetChStatus](#) (uint8 ChannelId)
Get DMA channel Status.
- Hal_StatusType [Dma_Hal_StartCh](#) (uint8 ChannelId)
Starts the DMA channel.
- Hal_StatusType [Dma_Hal_StopCh](#) (uint8 ChannelId)
Stops the DMA channel.
- void [Dma_Hal_ChReset](#) (uint8 ChannelId, Dma_ChannelResetType Reset)
Dma channel reset.
- uint32 [Dma_Hal_GetTransBytes](#) (uint8 ChannelId)
Get the transferred bytes by DMA.
- uint8 [Dma_Hal_GetChIdByReqSrc](#) (Dma_RequestSourceType ReqSrc)
Get dma channel id by module dma request source id.
- [ISR](#) (DMA0_Channel0_IRQHandler)
DMA0_Channel0 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel1_IRQHandler)
DMA0_Channel1 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel2_IRQHandler)
DMA0_Channel2 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel3_IRQHandler)
DMA0_Channel3 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel4_IRQHandler)
DMA0_Channel4 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel5_IRQHandler)
DMA0_Channel5 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel6_IRQHandler)
DMA0_Channel6 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel7_IRQHandler)
DMA0_Channel7 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel8_IRQHandler)
DMA0_Channel8 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel9_IRQHandler)
DMA0_Channel9 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel10_IRQHandler)
DMA0_Channel10 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel11_IRQHandler)
DMA0_Channel11 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel12_IRQHandler)
DMA0_Channel12 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel13_IRQHandler)
DMA0_Channel13 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel14_IRQHandler)
DMA0_Channel14 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel15_IRQHandler)
DMA0_Channel15 Interrupt Handler Function.

4.3.1 Detailed Description

This file provides extern Dma Hal API implement.

4.3.2 Enumeration Type Documentation

4.3.2.1 Dma_StateType

```
enum Dma_StateType
```

Dma state machine.

Enumerator

DMA_STATE_INITED	
DMA_STATE_UNINITED	
DMA_STATE_CONFIGURED	
DMA_STATE_RUNNING	
DMA_STATE_ERROR	

Definition at line 59 of file Dma_Hal.c.

4.3.3 Function Documentation

4.3.3.1 Dma_Hal_ChReset()

```
void Dma_Hal_ChReset (
    uint8 ChannelId,
    Dma_ChannelResetType Reset )
```

Dma channel reset.

Note

Function ID : DES_MCL_API_110

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Reset</i>	reset type

Returns

void

Definition at line 822 of file Dma_Hal.c.

4.3.3.2 Dma_Hal_ConfigCh()

```
Hal_StatusType Dma_Hal_ConfigCh (
    uint8 ChannelId,
    const Dma_TransferConfigType * ConfigPtr )
```

Configures data transfer with DMA.

Note

Function ID : DES_MCL_API_103

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>ConfigPtr</i>	DMA transfer configuration info of Channel

Returns

STATUS_UNSUPPORTED means fail or STATUS_SUCCESS means success.

Definition at line 497 of file Dma_Hal.c.

4.3.3.3 Dma_Hal_Deinit()

```
void Dma_Hal_Deinit (
    void )
```

De-initializes the DMA module.

Note

Function ID : DES_MCL_API_102

Returns

void

Definition at line 424 of file Dma_Hal.c.

4.3.3.4 Dma_Hal_EnableChIrq()

```
void Dma_Hal_EnableChIrq (
    uint8 ChannelId,
    uint8 IrqSrc )
```

disable/enable the channel interrupt.

Note

Function ID : DES_MCL_API_109

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>IrqSrc</i>	Enable/Disable irq, half finish irq only valid for 7842x and 7843x

Returns

void

Definition at line 635 of file Dma_Hal.c.

4.3.3.5 Dma_Hal_GetChIdByReqSrc()

```
uint8 Dma_Hal_GetChIdByReqSrc (
    Dma_RequestSourceType ReqSrc )
```

Get dma channel id by module dma request source id.

Note

Function ID : DES_MCL_API_112

Parameters

in	<i>ReqSrc</i>	dma request source id
----	---------------	-----------------------

Returns

uint8: dma channel id

Definition at line 885 of file Dma_Hal.c.

4.3.3.6 Dma_Hal_GetChStatus()

```
Hal_StatusType Dma_Hal_GetChStatus (
    uint8 ChannelId )
```

Get DMA channel Status.

Note

Function ID : DES_MCL_API_104

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

enum Dma_ChannelStatusType. STATUS_IDLE STATUS_ABORT STATUS_BUSY STATUS_CONFIG

Definition at line 681 of file Dma_Hal.c.

4.3.3.7 Dma_Hal_GetTransBytes()

```
uint32 Dma_Hal_GetTransBytes (
    uint8 ChannelId )
```

Get the transferred bytes by DMA.

Note

Function ID : DES_MCL_API_107

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

The transferred bytes to be transferred by DMA

Definition at line 858 of file Dma_Hal.c.

4.3.3.8 Dma_Hal_Init()

```
void Dma_Hal_Init (
    const Dma_ConfigType * ConfigPtr )
```

Initializes the DMA module.

Note

Function ID : DES_MCL_API_101

Parameters

in	<i>ConfigPtr</i>	The pointer to the DMA module Configed structure list.
----	------------------	--

Returns

void

Definition at line 398 of file Dma_Hal.c.

4.3.3.9 Dma_Hal_StartCh()

```
Hal_StatusType Dma_Hal_StartCh (
    uint8 ChannelId )
```

Starts the DMA channel.

Note

Function ID : DES_MCL_API_105

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

Hal_StatusType: start success or not

Definition at line 723 of file Dma_Hal.c.

4.3.3.10 Dma_Hal_StopCh()

```
Hal_StatusType Dma_Hal_StopCh (
    uint8 ChannelId )
```

Stops the DMA channel.

Note

Function ID : DES_MCL_API_106

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

Hal_StatusType: stop success or not

Definition at line 768 of file Dma_Hal.c.

4.3.3.11 Dma_Hal_UpdateChAddr()

```
void Dma_Hal_UpdateChAddr (
    uint8 ChannelId,
    const Dma_ChannelAddrType * ChannelAddr )
```

Update Channel source/destination address and offset.

Note

Function ID : DES_MCL_API_108

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>ChannelAddr</i>	the pointer to Dma_ChannelAddrType structure

Returns

void

Definition at line 595 of file Dma_Hal.c.

4.3.3.12 ISR() [1/16]

```
ISR (
    DMA0_Channel0_IRQHandler )
```

DMA0_Channel0 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_113

Returns

void

Definition at line 970 of file Dma_Hal.c.

4.3.3.13 ISR() [2/16]

```
ISR (
    DMA0_Channel1_IRQHandler )
```

DMA0_Channel1 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_114

Returns

void

Definition at line 980 of file Dma_Hal.c.

4.3.3.14 ISR() [3/16]

```
ISR (
    DMA0_Channel2_IRQHandler )
```

DMA0_Channel2 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_115

Returns

void

Definition at line 990 of file Dma_Hal.c.

4.3.3.15 ISR() [4/16]

```
ISR (
    DMA0_Channel3_IRQHandler )
```

DMA0_Channel3 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_116

Returns

void

Definition at line 1000 of file Dma_Hal.c.

4.3.3.16 ISR() [5/16]

```
ISR (
    DMA0_Channel4_IRQHandler )
```

DMA0_Channel4 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_117

Returns

void

Definition at line 1010 of file Dma_Hal.c.

4.3.3.17 ISR() [6/16]

```
ISR (
    DMA0_Channel5_IRQHandler )
```

DMA0_Channel5 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_118

Returns

void

Definition at line 1020 of file Dma_Hal.c.

4.3.3.18 ISR() [7/16]

```
ISR (
    DMA0_Channel6_IRQHandler )
```

DMA0_Channel6 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_119

Returns

void

Definition at line 1030 of file Dma_Hal.c.

4.3.3.19 ISR() [8/16]

```
ISR (
    DMA0_Channel7_IRQHandler )
```

DMA0_Channel7 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_120

Returns

void

Definition at line 1040 of file Dma_Hal.c.

4.3.3.20 ISR() [9/16]

```
ISR (
    DMA0_Channel8_IRQHandler )
```

DMA0_Channel8 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_121

Returns

void

Definition at line 1050 of file Dma_Hal.c.

4.3.3.21 ISR() [10/16]

```
ISR (
    DMA0_Channel9_IRQHandler )
```

DMA0_Channel9 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_122

Returns

void

Definition at line 1060 of file Dma_Hal.c.

4.3.3.22 ISR() [11/16]

```
ISR (
    DMA0_Channel10_IRQHandler )
```

DMA0_Channel10 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_123

Returns

void

Definition at line 1070 of file Dma_Hal.c.

4.3.3.23 ISR() [12/16]

```
ISR (
    DMA0_Channel11_IRQHandler )
```

DMA0_Channel11 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_124

Returns

void

Definition at line 1080 of file Dma_Hal.c.

4.3.3.24 ISR() [13/16]

```
ISR (
    DMA0_Channel12_IRQHandler )
```

DMA0_Channel12 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_125

Returns

void

Definition at line 1090 of file Dma_Hal.c.

4.3.3.25 ISR() [14/16]

```
ISR (
    DMA0_Channel13_IRQHandler )
```

DMA0_Channel13 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_126

Returns

void

Definition at line 1100 of file Dma_Hal.c.

4.3.3.26 ISR() [15/16]

```
ISR (
    DMA0_Channel14_IRQHandler )
```

DMA0_Channel14 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_127

Returns

void

Definition at line 1110 of file Dma_Hal.c.

4.3.3.27 ISR() [16/16]

```
ISR (
    DMA0_Channel15_IRQHandler )
```

DMA0_Channel15 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_128

Returns

void

Definition at line 1120 of file Dma_Hal.c.

4.4 Dma_Hal.h File Reference

dma hal interface for SDK

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

Classes

- struct [Dma_ChannelCBInfoType](#)
Dma channel interrupt callback type.
- struct [Dma_ChannelAddrType](#)
Dma channel address type.
- struct [Dma_TransferConfigType](#)
The user configuration structure for the an DMA channel.
- struct [Dma_ChannelConfigType](#)
The user configuration structure for the an DMA driver channel.
- struct [Dma_ConfigType](#)
The user configuration structure for the an DMA driver channel.

Functions

- void [Dma_Hal_Init](#) (const [Dma_ConfigType](#) *ConfigPtr)
Initializes the DMA module.
- void [Dma_Hal_Deinit](#) (void)
De-initializes the DMA module.
- Hal_StatusType [Dma_Hal_ConfigCh](#) (uint8 ChannelId, const [Dma_TransferConfigType](#) *ConfigPtr)
Configures data transfer with DMA.
- Hal_StatusType [Dma_Hal_GetChStatus](#) (uint8 ChannelId)
Get DMA channel Status.
- Hal_StatusType [Dma_Hal_StartCh](#) (uint8 ChannelId)
Starts the DMA channel.
- Hal_StatusType [Dma_Hal_StopCh](#) (uint8 ChannelId)
Stops the DMA channel.
- uint32 [Dma_Hal_GetTransBytes](#) (uint8 ChannelId)
Get the transferred bytes by DMA.
- void [Dma_Hal_UpdateChAddr](#) (uint8 ChannelId, const [Dma_ChannelAddrType](#) *ChannelAddr)
Update Channel source/destination address and offset.
- void [Dma_Hal_EnableChIrq](#) (uint8 ChannelId, uint8 IrqSrc)
disable/enable the channel interrupt.
- void [Dma_Hal_ChReset](#) (uint8 ChannelId, Dma_ChannelResetType Reset)
Dma channel reset.
- uint8 [Dma_Hal_GetChIdByReqSrc](#) (Dma_RequestSourceType ReqSrc)
Get dma channel id by module dma request source id.
- [ISR](#) (DMA0_Channel0_IRQHandler)
DMA0_Channel0 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel1_IRQHandler)
DMA0_Channel1 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel2_IRQHandler)
DMA0_Channel2 Interrupt Handler Function.
- [ISR](#) (DMA0_Channel3_IRQHandler)

- DMA0_Channel3 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel4_IRQHandler)
DMA0_Channel4 Interrupt Handler Function.
- DMA0_Channel5 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel5_IRQHandler)
DMA0_Channel5 Interrupt Handler Function.
- DMA0_Channel6 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel6_IRQHandler)
DMA0_Channel6 Interrupt Handler Function.
- DMA0_Channel7 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel7_IRQHandler)
DMA0_Channel7 Interrupt Handler Function.
- DMA0_Channel8 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel8_IRQHandler)
DMA0_Channel8 Interrupt Handler Function.
- DMA0_Channel9 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel9_IRQHandler)
DMA0_Channel9 Interrupt Handler Function.
- DMA0_Channel10 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel10_IRQHandler)
DMA0_Channel10 Interrupt Handler Function.
- DMA0_Channel11 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel11_IRQHandler)
DMA0_Channel11 Interrupt Handler Function.
- DMA0_Channel12 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel12_IRQHandler)
DMA0_Channel12 Interrupt Handler Function.
- DMA0_Channel13 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel13_IRQHandler)
DMA0_Channel13 Interrupt Handler Function.
- DMA0_Channel14 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel14_IRQHandler)
DMA0_Channel14 Interrupt Handler Function.
- DMA0_Channel15 Interrupt Handler Function.*
 - [ISR](#) (DMA0_Channel15_IRQHandler)
DMA0_Channel15 Interrupt Handler Function.

4.4.1 Detailed Description

dma hal interface for SDK

4.4.2 Function Documentation

4.4.2.1 Dma_Hal_ChReset()

```
void Dma_Hal_ChReset (
    uint8 ChannelId,
    Dma_ChannelResetType Reset )
```

Dma channel reset.

Note

Function ID : DES_MCL_API_110

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>Reset</i>	reset type

Returns

void

Definition at line 822 of file Dma_Hal.c.

4.4.2.2 Dma_Hal_ConfigCh()

```
Hal_StatusType Dma_Hal_ConfigCh (
    uint8 ChannelId,
    const Dma_TransferConfigType * ConfigPtr )
```

Configures data transfer with DMA.

Note

Function ID : DES_MCL_API_103

Parameters

in	<i>ChannelId</i>	Dma channel id
in	<i>ConfigPtr</i>	DMA transfer configuration info of Channel

Returns

STATUS_UNSUPPORTED means fail or STATUS_SUCCESS means success.

Note

Function ID : DES_MCL_API_103

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>ConfigPtr</i>	DMA transfer configuration info of Channel

Returns

STATUS_UNSUPPORTED means fail or STATUS_SUCCESS means success.

Definition at line 497 of file Dma_Hal.c.

4.4.2.3 Dma_Hal_Deinit()

```
void Dma_Hal_Deinit (
    void )
```

De-initializes the DMA module.

Note

Function ID : DES_MCL_API_102

Returns

void

Definition at line 424 of file Dma_Hal.c.

4.4.2.4 Dma_Hal_EnableChIrq()

```
void Dma_Hal_EnableChIrq (
    uint8 ChannelId,
    uint8 IrqSrc )
```

disable/enable the channel interrupt.

Note

Function ID : DES_MCL_API_109

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>IrqSrc</i>	Enable/Disable irq, half finish irq only valid for 7842x and 7843x

Returns

void

Definition at line 635 of file Dma_Hal.c.

4.4.2.5 Dma_Hal_GetChIdByReqSrc()

```
uint8 Dma_Hal_GetChIdByReqSrc (
    Dma_RequestSourceType ReqSrc )
```

Get dma channel id by module dma request source id.

Note

Function ID : DES_MCL_API_112

Parameters

in	<i>ReqSrc</i>	dma request source id
----	---------------	-----------------------

Returns

uint8: dma channel id

Definition at line 885 of file Dma_Hal.c.

4.4.2.6 Dma_Hal_GetChStatus()

```
Hal_StatusType Dma_Hal_GetChStatus (
    uint8 ChannelId )
```

Get DMA channel Status.

Note

Function ID : DES_MCL_API_104

Parameters

in	<i>ChannelId</i>	Dma channel id
----	------------------	----------------

Returns

enum Dma_ChannelStatusType. STATUS_IDLE STATUS_ABORT STATUS_BUSY STATUS_CONFIG

Note

Function ID : DES_MCL_API_104

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

enum Dma_ChannelStatusType. STATUS_IDLE STATUS_ABORT STATUS_BUSY STATUS_CONFIG

Definition at line 681 of file Dma_Hal.c.

4.4.2.7 Dma_Hal_GetTransBytes()

```
uint32 Dma_Hal_GetTransBytes (
    uint8 ChannelId )
```

Get the transferred bytes by DMA.

Note

Function ID : DES_MCL_API_107

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

The transferred bytes to be transferred by DMA

Definition at line 858 of file Dma_Hal.c.

4.4.2.8 Dma_Hal_Init()

```
void Dma_Hal_Init (
    const Dma_ConfigType * ConfigPtr )
```

Initializes the DMA module.

Note

Function ID : DES_MCL_API_101

Parameters

in	<i>ConfigPtr</i>	The pointer to the DMA module Configed structure list.
----	------------------	--

Returns

void

Definition at line 398 of file Dma_Hal.c.

4.4.2.9 Dma_Hal_StartCh()

```
Hal_StatusType Dma_Hal_StartCh (
    uint8 ChannelId )
```

Starts the DMA channel.

Note

Function ID : DES_MCL_API_105

Parameters

in	<i>Channel↔ Id</i>	dma channel id
----	------------------------	----------------

Returns

Hal_StatusType: start success or not

Definition at line 723 of file Dma_Hal.c.

4.4.2.10 Dma_Hal_StopCh()

```
Hal_StatusType Dma_Hal_StopCh (
    uint8 ChannelId )
```

Stops the DMA channel.

Note

Function ID : DES_MCL_API_106

Parameters

in	<i>ChannelId</i>	dma channel id
----	------------------	----------------

Returns

Hal_StatusType: stop success or not

Definition at line 768 of file Dma_Hal.c.

4.4.2.11 Dma_Hal_UpdateChAddr()

```
void Dma_Hal_UpdateChAddr (
    uint8 ChannelId,
    const Dma_ChannelAddrType * ChannelAddr )
```

Update Channel source/destination address and offset.

Note

Function ID : DES_MCL_API_108

Parameters

in	<i>ChannelId</i>	dma channel id
in	<i>ChannelAddr</i>	the pointer to Dma_ChannelAddrType structure

Returns

void

Definition at line 595 of file Dma_Hal.c.

4.4.2.12 ISR() [1/16]

```
ISR (
    DMA0_Channel0_IRQHandler )
```

DMA0_Channel0 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_113

Returns

void

Definition at line 970 of file Dma_Hal.c.

4.4.2.13 ISR() [2/16]

```
ISR (
    DMA0_Channel1_IRQHandler )
```

DMA0_Channel1 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_114

Returns

void

Definition at line 980 of file Dma_Hal.c.

4.4.2.14 ISR() [3/16]

```
ISR (
    DMA0_Channel2_IRQHandler )
```

DMA0_Channel2 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_115

Returns

void

Definition at line 990 of file Dma_Hal.c.

4.4.2.15 ISR() [4/16]

```
ISR (
    DMA0_Channel3_IRQHandler )
```

DMA0_Channel3 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_116

Returns

void

Definition at line 1000 of file Dma_Hal.c.

4.4.2.16 ISR() [5/16]

```
ISR (
    DMA0_Channel4_IRQHandler )
```

DMA0_Channel4 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_117

Returns

void

Definition at line 1010 of file Dma_Hal.c.

4.4.2.17 ISR() [6/16]

```
ISR (
    DMA0_Channel5_IRQHandler )
```

DMA0_Channel5 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_118

Returns

void

Definition at line 1020 of file Dma_Hal.c.

4.4.2.18 ISR() [7/16]

```
ISR (
    DMA0_Channel6_IRQHandler )
```

DMA0_Channel6 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_119

Returns

void

Definition at line 1030 of file Dma_Hal.c.

4.4.2.19 ISR() [8/16]

```
ISR (
    DMA0_Channel7_IRQHandler )
```

DMA0_Channel7 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_120

Returns

void

Definition at line 1040 of file Dma_Hal.c.

4.4.2.20 ISR() [9/16]

```
ISR (
    DMA0_Channel8_IRQHandler )
```

DMA0_Channel8 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_121

Returns

void

Definition at line 1050 of file Dma_Hal.c.

4.4.2.21 ISR() [10/16]

```
ISR (
    DMA0_Channel9_IRQHandler )
```

DMA0_Channel9 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_122

Returns

void

Definition at line 1060 of file Dma_Hal.c.

4.4.2.22 ISR() [11/16]

```
ISR (
    DMA0_Channel10_IRQHandler )
```

DMA0_Channel10 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_123

Returns

void

Definition at line 1070 of file Dma_Hal.c.

4.4.2.23 ISR() [12/16]

```
ISR (
    DMA0_Channel11_IRQHandler )
```

DMA0_Channel11 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_124

Returns

void

Definition at line 1080 of file Dma_Hal.c.

4.4.2.24 ISR() [13/16]

```
ISR (
    DMA0_Channel12_IRQHandler )
```

DMA0_Channel12 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_125

Returns

void

Definition at line 1090 of file Dma_Hal.c.

4.4.2.25 ISR() [14/16]

```
ISR (
    DMA0_Channel13_IRQHandler )
```

DMA0_Channel13 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_126

Returns

void

Definition at line 1100 of file Dma_Hal.c.

4.4.2.26 ISR() [15/16]

```
ISR (
    DMA0_Channel14_IRQHandler )
```

DMA0_Channel14 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_127

Returns

void

Definition at line 1110 of file Dma_Hal.c.

4.4.2.27 ISR() [16/16]

```
ISR (
    DMA0_Channel15_IRQHandler )
```

DMA0_Channel15 Interrupt Handler Function.

Note

Function ID : DES_MCL_API_128

Returns

void

Definition at line 1120 of file Dma_Hal.c.

Index

AC784xx_API_Reference_Manual_DMA.pdf, [14](#)

AC784xx_Dma_Reg.h, [14](#)

Dma_Reg_ChannelHardRst, [15](#)

Dma_Reg_ChannelWarmRst, [16](#)

Dma_Reg_ClearDmaStatus, [16](#)

Dma_Reg_FlushChannelFIFO, [17](#)

Dma_Reg_GetChannelSource, [17](#)

Dma_Reg_GetCircularMode, [18](#)

Dma_Reg_GetDmaStatus, [18](#)

Dma_Reg_GetDstTransferSize, [19](#)

Dma_Reg_GetInterFIFODataLeftBytes, [19](#)

Dma_Reg_GetSrcTransferSize, [20](#)

Dma_Reg_GetTransferLength, [21](#)

Dma_Reg_GetTransferredBytes, [20](#)

Dma_Reg_SetChannel, [21](#)

Dma_Reg_SetChannelDebug, [22](#)

Dma_Reg_SetChannelPause, [22](#)

Dma_Reg_SetChannelPriority, [23](#)

Dma_Reg_SetChannelSource, [23](#)

Dma_Reg_SetChannelTrigger, [24](#)

Dma_Reg_SetCircularMode, [24](#)

Dma_Reg_SetDestAddrOffSet, [25](#)

Dma_Reg_SetDestEndAddr, [25](#)

Dma_Reg_SetDestStartAddr, [26](#)

Dma_Reg_SetDstTransferSize, [26](#)

Dma_Reg_SetErrorInterrupt, [27](#)

Dma_Reg_SetFIFOFastFunction, [27](#)

Dma_Reg_SetFinishInterrupt, [28](#)

Dma_Reg_SetInterrupt, [28](#)

Dma_Reg_SetSrcAddrOffSet, [29](#)

Dma_Reg_SetSrcEndAddr, [29](#)

Dma_Reg_SetSrcStartAddr, [30](#)

Dma_Reg_SetSrcTransferSize, [30](#)

Dma_Reg_SetTransferLength, [31](#)

Dma_Reg_TopHardRst, [31](#)

Dma_Reg_TopWarmRst, [31](#)

Callback

Dma_ChannelInfoType, [8](#)

Dma_TransferConfigType, [10](#)

ChannelCfg

Dma_ConfigType, [9](#)

ChannelCnt

Dma_ConfigType, [9](#)

ChannelId

Dma_ChannelCBInfoType, [5](#)

CircularMode

Dma_TransferConfigType, [10](#)

DestEndAddr

Dma_ChannelAddrType, [3](#)

Dma_TransferConfigType, [10](#)

DestOffset

Dma_ChannelAddrType, [3](#)

Dma_TransferConfigType, [11](#)

DestStartAddr

Dma_ChannelAddrType, [4](#)

Dma_TransferConfigType, [11](#)

DestUnit

Dma_TransferConfigType, [11](#)

Dma_ChannelAddrType, [3](#)

DestEndAddr, [3](#)

DestOffset, [3](#)

DestStartAddr, [4](#)

Length, [4](#)

SrcEndAddr, [4](#)

SrcOffset, [4](#)

SrcStartAddr, [4](#)

Dma_ChannelCBInfoType, [5](#)

ChannelId, [5](#)

DmaEvent, [5](#)

UserArgs, [6](#)

Dma_ChannelConfigType, [6](#)

HwChannelId, [6](#)

Priority, [7](#)

ReqSrc, [7](#)

VirtualChannelId, [7](#)

Dma_ChannelInfoType, [7](#)

Callback, [8](#)

ReqSrc, [8](#)

State, [8](#)

UserArgs, [8](#)

VirtualChannelId, [8](#)

Dma_ConfigType, [9](#)

ChannelCfg, [9](#)

ChannelCnt, [9](#)

Dma_Hal.c, [32](#)

Dma_Hal_ChReset, [34](#)

Dma_Hal_ConfigCh, [35](#)

Dma_Hal_Deinit, [35](#)

Dma_Hal_EnableChIrq, [35](#)

Dma_Hal_GetChIdByReqSrc, [36](#)

Dma_Hal_GetChStatus, [36](#)

Dma_Hal_GetTransBytes, [37](#)

Dma_Hal_Init, [37](#)

Dma_Hal_StartCh, [38](#)

Dma_Hal_StopCh, [38](#)

Dma_Hal_UpdateChAddr, [39](#)

Dma_StateType, [34](#)

ISR, [39–44](#)

Dma_Hal.h, [45](#)

Dma_Hal_ChReset, [46](#)

Dma_Hal_ConfigCh, [47](#)

Dma_Hal_Deinit, [47](#)

- Dma_Hal_EnableChlrq, [48](#)
- Dma_Hal_GetChldByReqSrc, [48](#)
- Dma_Hal_GetChStatus, [49](#)
- Dma_Hal_GetTransBytes, [49](#)
- Dma_Hal_Init, [50](#)
- Dma_Hal_StartCh, [50](#)
- Dma_Hal_StopCh, [51](#)
- Dma_Hal_UpdateChAddr, [51](#)
- ISR, [52–57](#)
- Dma_Hal_ChReset
 - Dma_Hal.c, [34](#)
 - Dma_Hal.h, [46](#)
- Dma_Hal_ConfigCh
 - Dma_Hal.c, [35](#)
 - Dma_Hal.h, [47](#)
- Dma_Hal_Deinit
 - Dma_Hal.c, [35](#)
 - Dma_Hal.h, [47](#)
- Dma_Hal_EnableChlrq
 - Dma_Hal.c, [35](#)
 - Dma_Hal.h, [48](#)
- Dma_Hal_GetChldByReqSrc
 - Dma_Hal.c, [36](#)
 - Dma_Hal.h, [48](#)
- Dma_Hal_GetChStatus
 - Dma_Hal.c, [36](#)
 - Dma_Hal.h, [49](#)
- Dma_Hal_GetTransBytes
 - Dma_Hal.c, [37](#)
 - Dma_Hal.h, [49](#)
- Dma_Hal_Init
 - Dma_Hal.c, [37](#)
 - Dma_Hal.h, [50](#)
- Dma_Hal_StartCh
 - Dma_Hal.c, [38](#)
 - Dma_Hal.h, [50](#)
- Dma_Hal_StopCh
 - Dma_Hal.c, [38](#)
 - Dma_Hal.h, [51](#)
- Dma_Hal_UpdateChAddr
 - Dma_Hal.c, [39](#)
 - Dma_Hal.h, [51](#)
- Dma_Reg_ChannelHardRst
 - AC784xx_Dma_Reg.h, [15](#)
- Dma_Reg_ChannelWarmRst
 - AC784xx_Dma_Reg.h, [16](#)
- Dma_Reg_ClearDmaStatus
 - AC784xx_Dma_Reg.h, [16](#)
- Dma_Reg_FlushChannelFIFO
 - AC784xx_Dma_Reg.h, [17](#)
- Dma_Reg_GetChannelSource
 - AC784xx_Dma_Reg.h, [17](#)
- Dma_Reg_GetCircularMode
 - AC784xx_Dma_Reg.h, [18](#)
- Dma_Reg_GetDmaStatus
 - AC784xx_Dma_Reg.h, [18](#)
- Dma_Reg_GetDstTransferSize
 - AC784xx_Dma_Reg.h, [19](#)
- Dma_Reg_GetInterFIFODataLeftBytes
 - AC784xx_Dma_Reg.h, [19](#)
- Dma_Reg_GetSrcTransferSize
 - AC784xx_Dma_Reg.h, [20](#)
- Dma_Reg_GetTransferLength
 - AC784xx_Dma_Reg.h, [21](#)
- Dma_Reg_GetTransferredBytes
 - AC784xx_Dma_Reg.h, [20](#)
- Dma_Reg_SetChannel
 - AC784xx_Dma_Reg.h, [21](#)
- Dma_Reg_SetChannelDebug
 - AC784xx_Dma_Reg.h, [22](#)
- Dma_Reg_SetChannelPause
 - AC784xx_Dma_Reg.h, [22](#)
- Dma_Reg_SetChannelPriority
 - AC784xx_Dma_Reg.h, [23](#)
- Dma_Reg_SetChannelSource
 - AC784xx_Dma_Reg.h, [23](#)
- Dma_Reg_SetChannelTrigger
 - AC784xx_Dma_Reg.h, [24](#)
- Dma_Reg_SetCircularMode
 - AC784xx_Dma_Reg.h, [24](#)
- Dma_Reg_SetDestAddrOffSet
 - AC784xx_Dma_Reg.h, [25](#)
- Dma_Reg_SetDestEndAddr
 - AC784xx_Dma_Reg.h, [25](#)
- Dma_Reg_SetDestStartAddr
 - AC784xx_Dma_Reg.h, [26](#)
- Dma_Reg_SetDstTransferSize
 - AC784xx_Dma_Reg.h, [26](#)
- Dma_Reg_SetErrorInterrupt
 - AC784xx_Dma_Reg.h, [27](#)
- Dma_Reg_SetFIFOFastFunction
 - AC784xx_Dma_Reg.h, [27](#)
- Dma_Reg_SetFinishInterrupt
 - AC784xx_Dma_Reg.h, [28](#)
- Dma_Reg_SetInterrupt
 - AC784xx_Dma_Reg.h, [28](#)
- Dma_Reg_SetSrcAddrOffSet
 - AC784xx_Dma_Reg.h, [29](#)
- Dma_Reg_SetSrcEndAddr
 - AC784xx_Dma_Reg.h, [29](#)
- Dma_Reg_SetSrcStartAddr
 - AC784xx_Dma_Reg.h, [30](#)
- Dma_Reg_SetSrcTransferSize
 - AC784xx_Dma_Reg.h, [30](#)
- Dma_Reg_SetTransferLength
 - AC784xx_Dma_Reg.h, [31](#)
- Dma_Reg_TopHardRst
 - AC784xx_Dma_Reg.h, [31](#)
- Dma_Reg_TopWarmRst
 - AC784xx_Dma_Reg.h, [31](#)
- Dma_StateType
 - Dma_Hal.c, [34](#)
- Dma_TransferConfigType, [9](#)
 - Callback, [10](#)
 - CircularMode, [10](#)
 - DestEndAddr, [10](#)
 - DestOffset, [11](#)
 - DestStartAddr, [11](#)
 - DestUnit, [11](#)
 - IrqSrc, [11](#)
 - Length, [11](#)
 - SrcEndAddr, [12](#)

- SrcOffset, [12](#)
- SrcStartAddr, [12](#)
- SrcUnit, [12](#)
- TriggerMode, [12](#)
- Type, [13](#)
- UserArgs, [13](#)
- DmaEvent
 - Dma_ChannelCBInfoType, [5](#)
- HwChannelId
 - Dma_ChannelConfigType, [6](#)
- ISR
 - Dma_Hal.c, [39–44](#)
 - Dma_Hal.h, [52–57](#)
- IrqSrc
 - Dma_TransferConfigType, [11](#)
- Length
 - Dma_ChannelAddrType, [4](#)
 - Dma_TransferConfigType, [11](#)
- Priority
 - Dma_ChannelConfigType, [7](#)
- ReqSrc
 - Dma_ChannelConfigType, [7](#)
 - Dma_ChannelInfoType, [8](#)
- SrcEndAddr
 - Dma_ChannelAddrType, [4](#)
 - Dma_TransferConfigType, [12](#)
- SrcOffset
 - Dma_ChannelAddrType, [4](#)
 - Dma_TransferConfigType, [12](#)
- SrcStartAddr
 - Dma_ChannelAddrType, [4](#)
 - Dma_TransferConfigType, [12](#)
- SrcUnit
 - Dma_TransferConfigType, [12](#)
- State
 - Dma_ChannelInfoType, [8](#)
- TriggerMode
 - Dma_TransferConfigType, [12](#)
- Type
 - Dma_TransferConfigType, [13](#)
- UserArgs
 - Dma_ChannelCBInfoType, [6](#)
 - Dma_ChannelInfoType, [8](#)
 - Dma_TransferConfigType, [13](#)
- VirtualChannelId
 - Dma_ChannelConfigType, [7](#)
 - Dma_ChannelInfoType, [8](#)