User Manual

for S32K3 MCL Driver

Document Number: UM34MCLASR21-11 Rev0000R3.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.03.2023	NXP RTD Team	S32K3 Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 3.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR *driver* for *platform*. AUTOSAR *driver* driver configuration parameters and deviations from the specification are described in Driver chapter of this document. AUTOSAR *driver* driver requirements and APIs are described in the AUTOSAR *driver* driver software specification document.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k310_mqfp100
- s32k310_lqfp48
- s32k311_mqfp100 / MWCT2015S_mqfp100
- s32k311_lqfp48
- s32k312_mqfp100 / MWCT2016S_mqfp100
- s32k312_mqfp172 / MWCT2016S_mqfp172
- s32k314_mqfp172
- s32k314_mapbga257
- s32k322_mqfp100 / MWCT2D16S_mqfp100
- s32k322_mqfp172 / MWCT2D16S_mqfp172

- s32k324_mqfp172 / MWCT2D17S_mqfp172
- s32k324_mapbga257
- s32k341_mqfp100
- s32k341_mqfp172
- s32k342_mqfp100
- s32k342_mqfp172
- s32k344_mqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k396_mapbga289
- s32k358_mqfp172
- s32k358_mapbga289
- s32k328_mqfp172
- s32k328_mapbga289
- s32k338_mqfp172
- s32k338_mapbga289
- s32k348_mqfp172
- s32k348_mapbga289
- s32m274_lqfp64
- s32m276_lqfp64

All of the above microcontroller devices are collectively named as S32K3.

Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power.

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR:

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Signifigant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	S32K3XX Reference Manual	Rev.6, Draft B, 01/2023
2	S32K3xx Data Sheet	Rev. 6, Draft B. 01/2023
3	S32K396 Reference Manual	Rev. 2 Draft A, 11/2022
4	S32K396 Data Sheet	Rev. 1.1 — 08/2022
5	S32M27x Reference Manual	Rev.2, Draft A, — 02/2023
6	S32M2xx Data Sheet	Rev. 2 RC — 12/2022
7	S32K358_0P14E Mask Set Errata	Rev. 28, 9/2022
8	S32K396_0P40E Mask Set Errata	Rev. DEC2022, 12/2022
9	S32K311_0P98C Mask Set Errata	Rev. 6/March/2023, 3/2023
10	S32K312: Mask Set Errata for Mask 0P09C	Rev. 25/April/2022
11	S32K342: Mask Set Errata for Mask 0P97C	Rev. 10, 11/2022
12	S32K3x4: Mask Set Errata for Mask 0P55A/1P55A	Rev. 14/Oct/2022

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the Autosar Driver Software Specification document (See Table Reference List).

For CDD: MCL Driver is a Complex Device Driver (CDD), so there are no AUTOSAR requirements regarding this module.

It has vendor-specific requirements and implementation.

3.2 Driver Design Summary

The Mcl Driver controls the DMA(Direct Memory Access), TRGMUX(Trigger Mux), CACHE, LCU(Logic Control Unit), FLEXIO and EMIOS modules of the S32K3XX device. It provides the following features:

- Configuration and initialization of the DMA.
- Configuration and initialization of the TRGMUX.
- Configuration and initialization of the CACHE.
- Configuration and initialization of the LCU.
- Configuration and initialization of the EMIOS.
- Configuration and initialization of the FLEXIO.
- Handling of the DMA interrupt requests.
- DMA Normal Transfer Mode and Scatter/Gather Mode.

3.3 Hardware Resources

The Mcl Driver consists of:

1. A DMA Peripheral which has 1 Hardware Instance with 32 Hardware Channels.
2. A TRGMUX Peripheral which consists of 1 Hardware Instance.
3. A CACHE Memory included into the ARM Core Peripherals.
4. A LCU Peripheral which has 2 Hardware Instance.
5. A EMIOS Peripheral which has 3 Hardware Instance.

3.4 Deviations from Requirements

The driver deviates from the Mcl Driver Software Specification in some places.

The table Status Column Description identifies the requirements that are not fully implemented, implemented differently, or out of scope for the Mcl Driver.

The table Mcl Requirements Deviations provides the "Status" column description.

Term	Definition
N/S	Not In Scope
N/F	Not Fully Implemented
N/I	Not Implemented

3.4.0.0.1 Status Column Description

Requirement	Status	Description	Notes
CPR_RTD_00190.mcl	N/S	All modules which need to call Dem module shall provide a configuration parameter for disabling all calls of Dem_SetEventStatus. If this parameter is activated, no call of Dem_SetEventStatus must be performed. Per default, the call of Dem_SetEventStatus shall be allowed.	Only K1xx support this requirements (https://jira.sw.nxp.com/browse/AAI-767)

3.4.0.0.2 Mcl Requirements Deviations Files **Mcl_<VariantName>_PBcfg.c** and **Mcl_<VariantName>_PBcfg.h** will contain the definitions for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB).

Files **Mcl_Cfg.c** and **Mcl_Cfg.h** will contain the definitions for all parameters that are not variant aware.

3.5 Driver Limitations

The Mcl Driver has the following limitations:

- When DMA is used with CACHE enabled, the user shall Invalidate/Clean the CACHE in order to synchronize the transfered data.
- When using CACHE Invalidate and Clean functionalities, the user shall take into consideration that all variables that reside in the cache, are affected.
- The Cache Invalidate function need to be called before Cache Enable function is called (make sure that Cache wasn't enabled before).
- The Cache Clean function need to be called before Cache Disable function is called.
- The DMA Driver shall have the Source Address and Destination Address configured at runtime. These two parameters are not available in the configurator.
- When the DMA transfer has errors, user must call Mcl_SetDmaChannelCommand/Dma_Ip_SetLogic↩ ChannelCommand function to clear error state, wrong state and error status.
- In Scatter/Gather Mode, the Logic Channel will not execute the interrupt callback in a specific setup:
  - Setup:
    - TCD0 : Interrupt enabled and Mcl callback set;
    - TCD1 : Interrupt disabled.
  - Interrupt handler implementation:
    - Step 1: Check interrupt event;
    - Step 2: Clear interrupt flag;
    - Step 3: Check if interrupt is enabled;
    - Step 4: Call the callback function.

But in this case, when TCD0 is completed an interrupt occurs and the handler will execute. At the same time TCD1 is also loaded. When the handler checks if interrupt is enabled (Step 3), the value found in the register is 0 (interrupt disabled), thus the callback function is not called.

- Mcl_CacheEnable/Cache_Ip_Enable function sometimes do not work if cache contains dirty data.
- When the function Mcl_Emios_SetCounterBusPeriod() used as follow:
- -> // Function will DISABLE compartor transfer.
- -> Mcl_Emios_SetCounterBusPeriod(logicChannel, period, syncUpdate=TRUE);
- -> // Function will keep the state(now DISABLE) of the compartor transfer.
- -> Mcl_Emios_SetCounterBusPeriod(logicChannel, period, syncUpdate=FALSE);
- When using Virtual Address Mapping feature, the user shall handle the linker file to avoid the memory address is invalid.
- Clocks related limitations:

- According to the Reference Manual there are some clock ratios that must be respected, otherwise Mcl_Init call will drive the platform into a hard fault exception.
  - AIPS_SLOW_CLK - HSE_CLK ratio with respect to HSE_CLK_MODE_OPTION of DCM_GPR:DCMROF21 register:
    - 00b - Applicable for clocking option A. Ratio of 1:2 in between HSE IPS interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK), HSE_IAHB gasket enabled.
    - 01b - Applicable for clocking option C, D, E, E2, and F. Ratio of 1:2 in between HSE IPS interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK), HSE_IAHB gasket bypass.
    - 10b - 10b and 11b both are applicable for clocking option B in same way. Ratio of 1:4 in between HSE IPS interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK), HSE_IAHB gasket enabled.
    - 11b - 10b and 11b both are applicable for clocking option B in same way. Ratio of 1:4 in between HSE IPS interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK), HSE_IAHB gasket enabled.
  - AIPS_SLOW_CLK should be set to one-fourth or one-half of the frequency of CORE_CLK.

Note:

- LCU: In HLD, the application shall manage the Write Protect status of LCU. In IPL, the application can use Lcu_Ip_GetWriteProtect() to retrieve status.
- Speed optimization tips for Lcu functions (application side)
  - Use empty implementation for Exclusive Area_38.
  - In case Mcl driver is run on one core only, MclEnableMulticoreSupport can be set to STD_OFF.
  - User should select inputs/outputs that belong to the same instance.
- Recommendation: Please avoid placing the stack into a cacheable area. Rationale: If the cache policy is write-back and the stack is placed in a cacheable area, synchronization issues might appear. The application is responsible to handle such coherency issue.

3.6 Driver usage and configuration tips

3.6.1 MCAL MCL DMA migration guide to RTD MCL

3.6.1.1 Introduction

The RTD MCL DMA Driver brings a Generic Interface, to help the User in application development across multiple SoCs. The Generic Interface consists of software functions that are fully configurable using User defined configurations. The Generic Interface is structured into four function groups:

1. Set Command Functions
2. Get Status Functions
3. Set List Functions
4. Get Information (Parameter) Functions

The Set Command Functions shall trigger actions specific to the invoked Logic Entity. For example, the Logic Dma Instance shall be commanded to: “Stop execution”, “Stop execution with error signaling”, “Pause execution” or “Resume execution”.

The Get Status Functions shall read the status specific to the invoked Logic Entity. For example, the Logic Dma Instance shall return: “Hardware Errors specific to the IP”, “The Active Channel Id” and “Active Status”.

The Set List Functions shall configure a user defined list of settings for the invoked Logic Entity. For example, the Logic Dma Channel shall be configurable with the Transfer List of Parameters like: “Source Address”, “↩ Destination Address”, “Source Signed Offset”, “Destination Signed Offset”, “Major Loop Count”, etc... For the specific ScatterGather mode, the additional Logic Element parameter specifies the Software TCD that shall be loaded with the ScatterGather List of Parameters.

The Get Parameter Functions shall return specific parameter value. For example, the Logic Dma Channel shall return the “Destination Address” value, which contains the last accessed destination memory location.

3.6.1.2 TRESOS Configuration

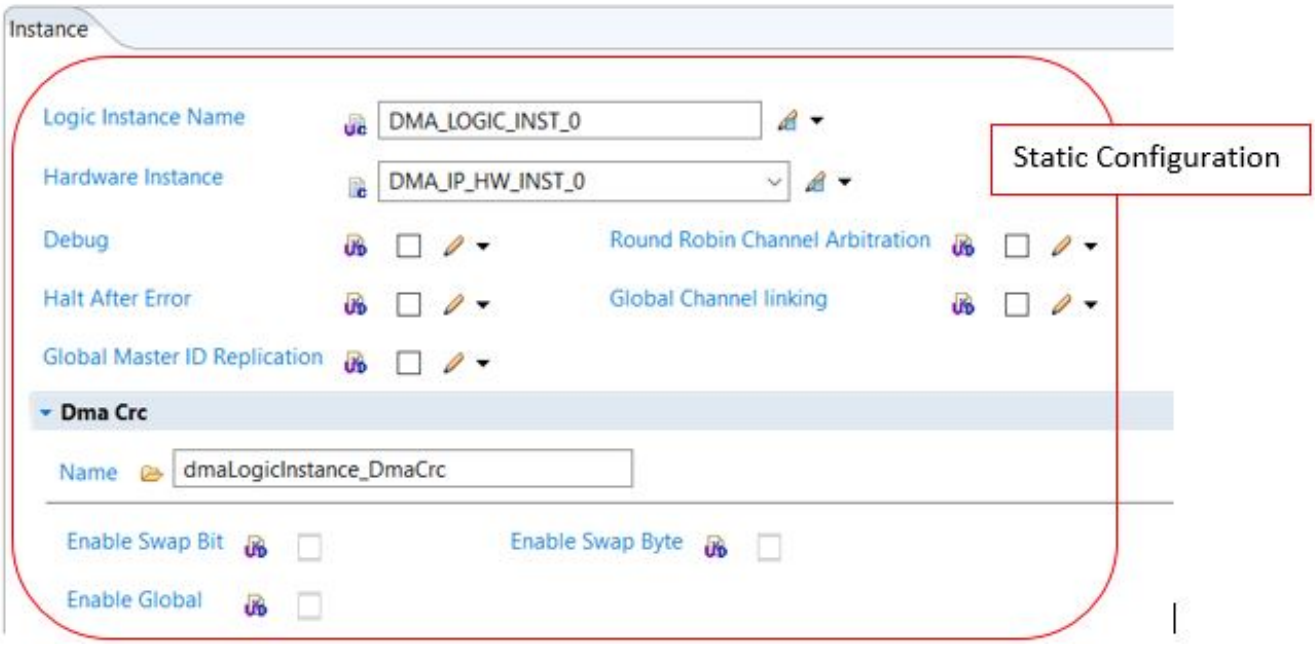

Figure 3.1 DMA Logic Instance

The DMA Logic Instance configuration presented in Figure 3.1 is static, thus it can’t be changed dynamically during runtime. The “Logic Instance Name” represents the Handler(Tag) used with the DMA API. The Handler(Tag) is set by the User for the specific Application; for example the “DMA_LOGIC_INSTANCE_0” shall be changed to “DMA_LOGIC_INST_COMMUNICATION”.

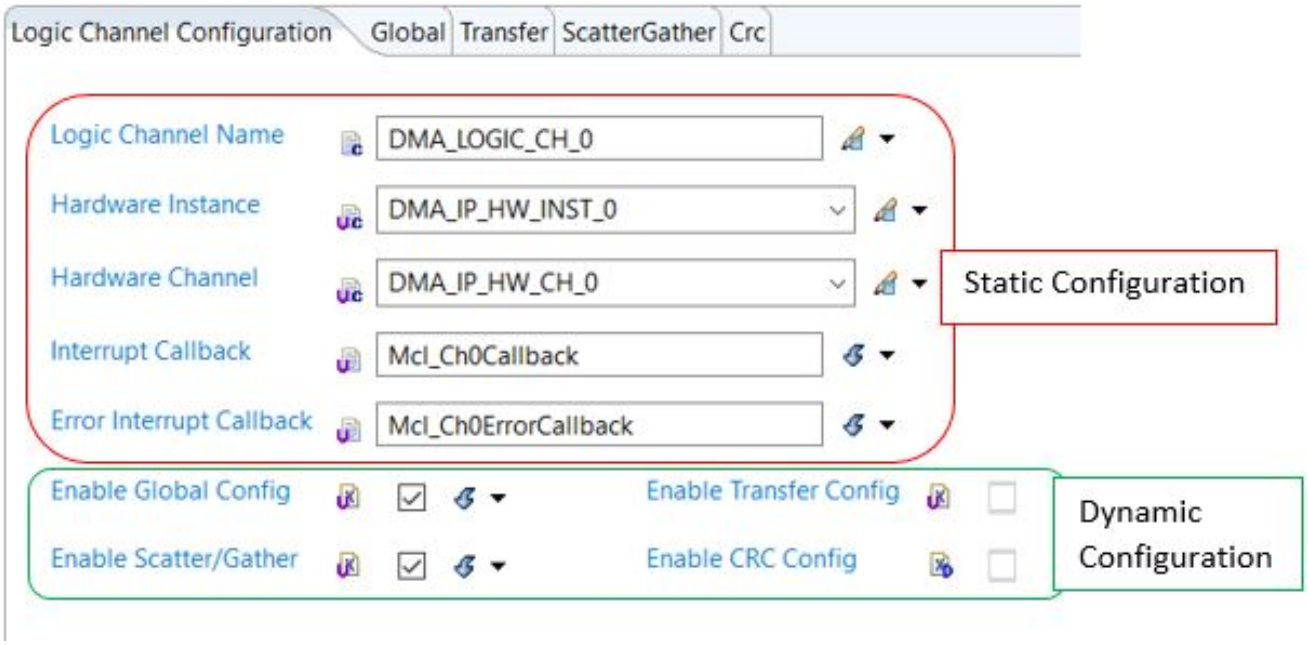

Figure 3.2 DMA Logic Channel

The DMA Logic Channel configuration in Figure 3.2 is composed of static and dynamic settings.

1. The static configuration is needed in order for the DMA Logic Channel to be created and used. The static configuration allocates resources for the DMA Logic Channel basic configuration. The static configuration can't be changed during runtime. The "Logic Channel Name" represents the Handler(Tag) used with the DMA API. The Handler(Tag) is set by the User based on the Application; For example the "DMA_LOGIC_CH_0" shall be changed to "DMA_CH_CAN0_RX" or "DMA_LOGIC_CH_SPI2_TX".
2. The dynamic configuration can be set in TRESOS and shall be automatically loaded in the DMA Logic Channel at initialization time. By enabling a checkbox, the respective configuration option is enabled and system memory is allocated during generation. The dynamic configuration can be set during runtime by using the specific API. By disabling a checkbox, the respective configuration option is disabled and system memory is not allocated during generation.

2.1. The "Enable Global Config" contains settings outside the Transfer Control Descriptor (TCD).

2.2. The "Enable Transfer Config" contains settings of the Transfer Control Descriptor (TCD).

2.3. The "Enable Scatter/Gather" contains settings that extend the "Transfer Config" by creating Software Transfer Control Descriptors (STCDs).

2.4. The "Enable CRC Config" contains settings for the DMA Logic Channel CRC computation functionality.

Note: The DMA Logic Channel can be configured in "Transfer Mode" or "Scatter/Gather Mode", thus only one of the 2 configurations can be set at any time. During runtime, the DMA Logic Channel can be configured between the 2 modes by calling the "SetTransfer" or "SetScatterGather" API.

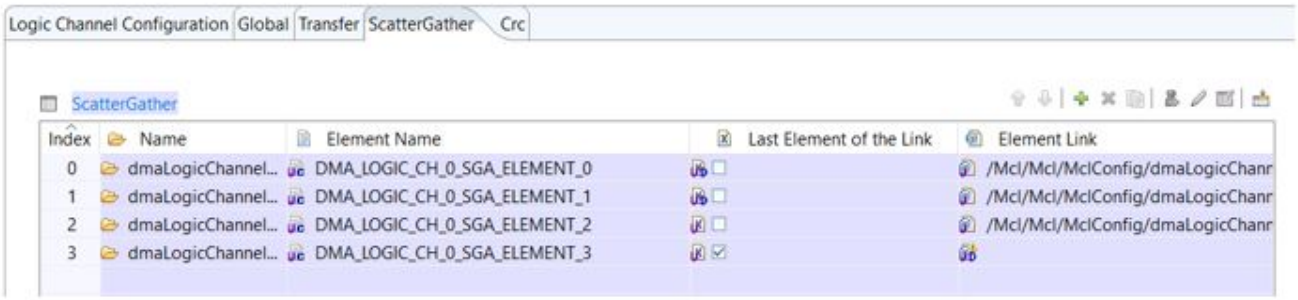

Figure 3.3 DMA Logic Channel – ScatterGather Element List

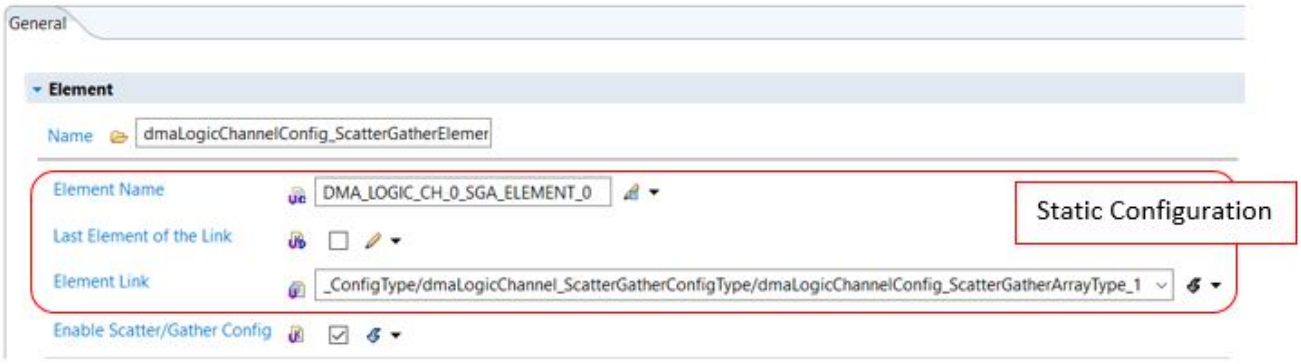

Figure 3.4 DMA Logic Channel – ScatterGather Element Configuration

The DMA Logic Channel ScatterGather Element list (Figure 3.3) shall be loaded with the required number of elements. Each element represents a STCD and is part of a linked list of elements.

The DMA Logic Channel ScatterGather Element Configuration (Figure 3.4) contains the static configuration of the element.

1. The “Element Name” represents the Handler (Tag) (static configuration).
2. The “Last Element of the Link” sets the element as the last link element (static configuration).
3. The “Element Link” points to the next element of the link (static configuration).
4. The “Enable Scatter/Gather Config” enables the element configuration.

Note1: Element can’t be added during runtime. The elements are allocated resources during generation (System memory for STCDs and configuration if set). The element linkage can’t be changed during runtime.

Note2: The DMA Logic Channel ScatterGather Element List can be configured to contain multiple independent chained lists. During runtime, the Logic Channel can be assigned a different Chained List.

Note3: Each Element is allocated 32 bytes of memory space aligned to 32 bytes, representing the STCD. Additionaly, if the configuration is enabled from TRESOS, additional 60 bytes are allocated.

3.6.1.3 Configurations using MCL EMIOS counters for Pwm, Icu and Ocu

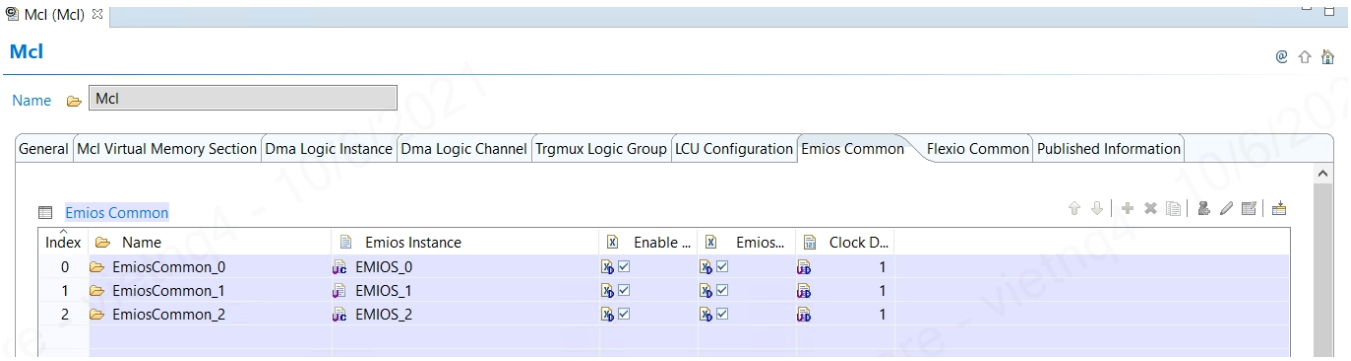

Figure 3.5 Emios Common tab – ScatterGather Emios List

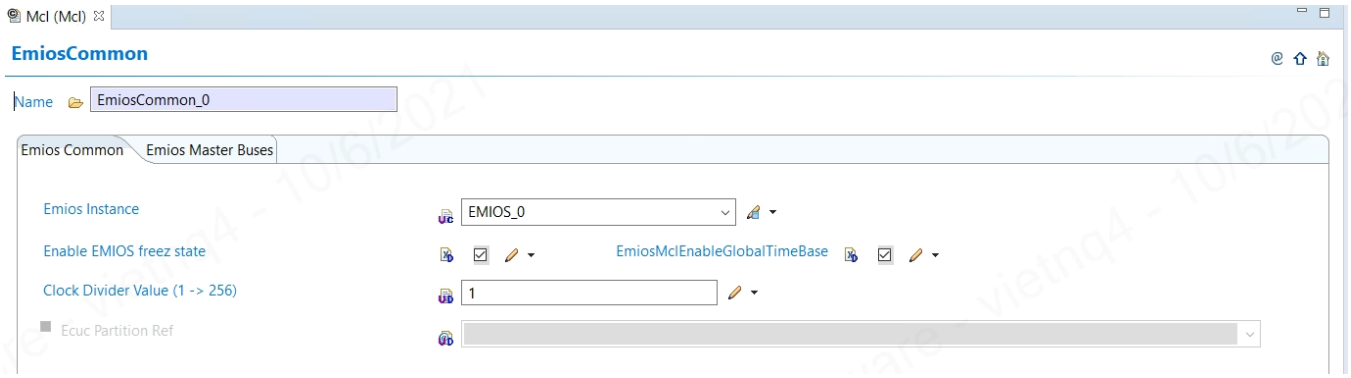

Figure 3.6 Emios Common config

1. Emios Instance: Select one of the EMIOS instance available on the platform.

2. Enable EMIOS freez state: Enable global timebase or disable on EMIOS IP.

3. EmiosMclEnableGlobalTimeBase: Enable global timebase or disable on EMIOS IP.

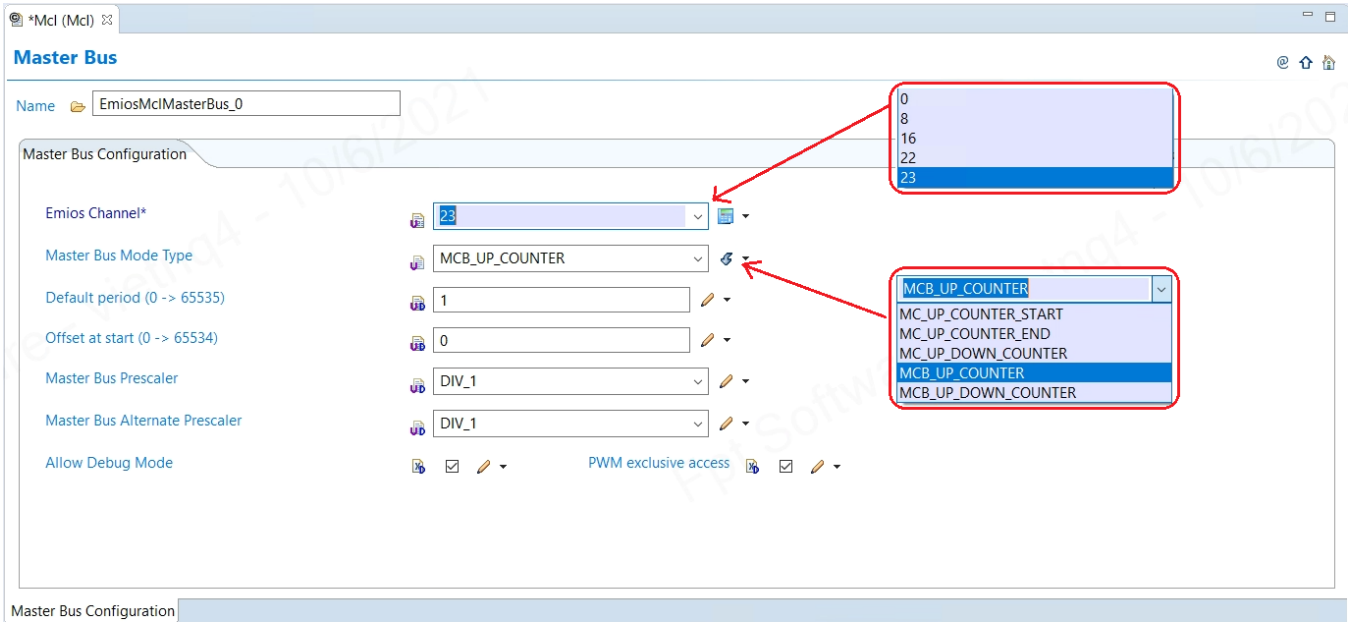

Figure 3.7 eMios Master Bus Configuration

1.Emios Channel : Selects one of the counter buses for the Unified channel

- Channel 0: counter bus B
- Channel 8: counter bus C
- Channel 16: counter bus D
- Channel 22: counter bus F
- Channel 23: counter bus A

2.Master Bus Mode Type: Configures the mode selection of counter bus.

lists mode:

- MC_UP_COUNTER_START: Modulus Counter (MC) mode with up counter with clear on match start.
- MC_UP_COUNTER_END: Modulus Counter (MC) mode with up counter with clear on match end.
- MC_UP_DOWN_COUNTER: Modulus Counter (MC) mode with up/down counter.
- MCB_UP_COUNTER: Modulus Counter Buffered (MCB) mode with up counter.
- MCB_UP_DOWN_COUNTER: Modulus Counter Buffered (MCB) mode with up or down counter.
  Note: Ocu and Icu only support with mode MCB_UP_COUNTER

3.Default period (0 -> 65535):

Note: Icu only support with Default period = 65535

4.Offset at start (0 -> 65534):

Note: When Master Bus Mode Type node is selected in a MCB mode the counter starts from 1 and counts to the period value.

5.Master Bus Prescaler: select bus Prescaler (DIV1-> DIV16)

6.Master Bus Alternate Prescaler: select bus alternate Prescaler (DIV1-> DIV16)

7.Allow Debug Mode: eMios Channel Allow DebugMode

8.PWM exclusive access: PWM exclusive access on current counter bus.

3.6.2 Channel State Machine

3.6.2.1 MCL Driver

The MCL DMA Driver runs based on the presented State Machine Diagram.

The MCL DMA State Machine applies to the MCL DMA Channels and it contains 5 States.

The State Machine supports the following 21 Transitions:

```
T0: Hardware Reset

T1: Mcl_Init() with DMA Channel no Transfer or Scatter/Gather generated configurations

T2: Mcl_DeInit()

T3: Mcl_SetDmaChannelTransferList()

T4: Mcl_Init() with DMA Channel Transfer generated configuration

T5: Mcl_SetDmaChannelScatterGatherList() + Mcl_SetDmaChannelScatterGatherConfig()

T6: Mcl_SetDmaChannelTransferList()

T7: Mcl_SetDmaChannelScatterGatherList() + Mcl_SetDmaChannelScatterGatherConfig()

T8: Mcl_Init() with DMA Channel Scatter/Gather generated configuration

T9: Detection of channel error

T10: Mcl_SetDmaInstanceCommand(CHANNEL, MCL_DMA_CH_ACK_ERROR)

T11: Detection of channel error

T12: Mcl_DeInit()

T13: Mcl_DeInit()

T14: Detection of channel error
```

```
T15: Detection of channel error

T16: Mcl_DeInit()

T17: Mcl_DeInit()

T18: Mcl_SetDmaChannelCommand(CHANNEL, MCL_DMA_CH_ACK_ERROR)

T19: Mcl_SetDmaChannelTransferList()

T20: Mcl_SetDmaChannelScatterGatherList() + Mcl_SetDmaChannelScatterGatherConfig()

T21: Detection of channel error
```

When the DMA Driver detects a software or hardware error for the Logic Channel, it shall enter the "MCL_DMA↩ _CH_ERROR_STATE".

To exit from the "MCL_DMA_CH_ERROR_STATE", the application shall use the specified transitions.

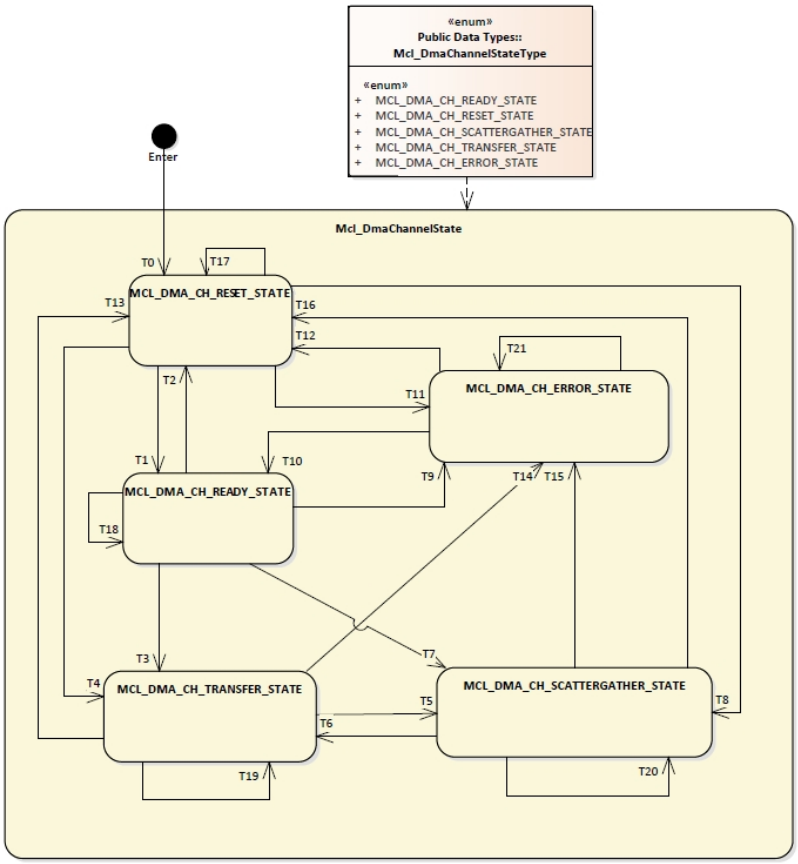

Figure 3.8 MCL DMA Channel State Machine

3.6.2.2 DMA Driver

The DMA IP Driver runs based on the presented State Machine Diagram.

The DMA State Machine applies to the DMA Logic Channels and it contains 5 States.

The State Machine supports the following 21 Transitions:

```
T0: Hardware Reset

T1: Dma_Ip_LogicChannelInit() with no Transfer or Scatter/Gather generated configurations

T2: Dma_Ip_LogicChannelDeinit()

T3: Dma_Ip_SetLogicChannelTransferList()

T4: Dma_Ip_LogicChannelInit() with Transfer generated configuration

T5: Dma_Ip_SetLogicChannelScatterGatherList() + Dma_Ip_SetLogicChannelScatterGatherConfig()

T6: Dma_Ip_SetLogicChannelTransferList()

T7: Dma_Ip_SetLogicChannelScatterGatherList() + Dma_Ip_SetLogicChannelScatterGatherConfig()

T8: Dma_Ip_LogicChannelInit() with Scatter/Gather generated configuration

T9: Detection of channel error

T10: Dma_Ip_SetLogicChannelCommand(CHANNEL, DMA_IP_CH_CLEAR_ERROR)

T11: Detection of channel error

T12: Dma_Ip_LogicChannelDeinit()

T13: Dma_Ip_LogicChannelDeinit()

T14: Detection of channel error

T15: Detection of channel error

T16: Dma_Ip_LogicChannelDeinit()

T17: Dma_Ip_LogicChannelDeinit()

T18: Dma_Ip_SetLogicChannelCommand(CHANNEL, DMA_IP_CH_CLEAR_ERROR)

T19: Dma_Ip_SetLogicChannelTransferList()

T20: Dma_Ip_SetLogicChannelScatterGatherList() + Dma_Ip_SetLogicChannelScatterGatherConfig()

T21: Detection of channel error
```

When the DMA Driver detects a software or hardware error for the Logic Channel, it shall enter the "DMA_IP_↩ CH_ERROR_STATE".

To exit from the "DMA_IP_CH_ERROR_STATE", the application shall use the specified transitions.

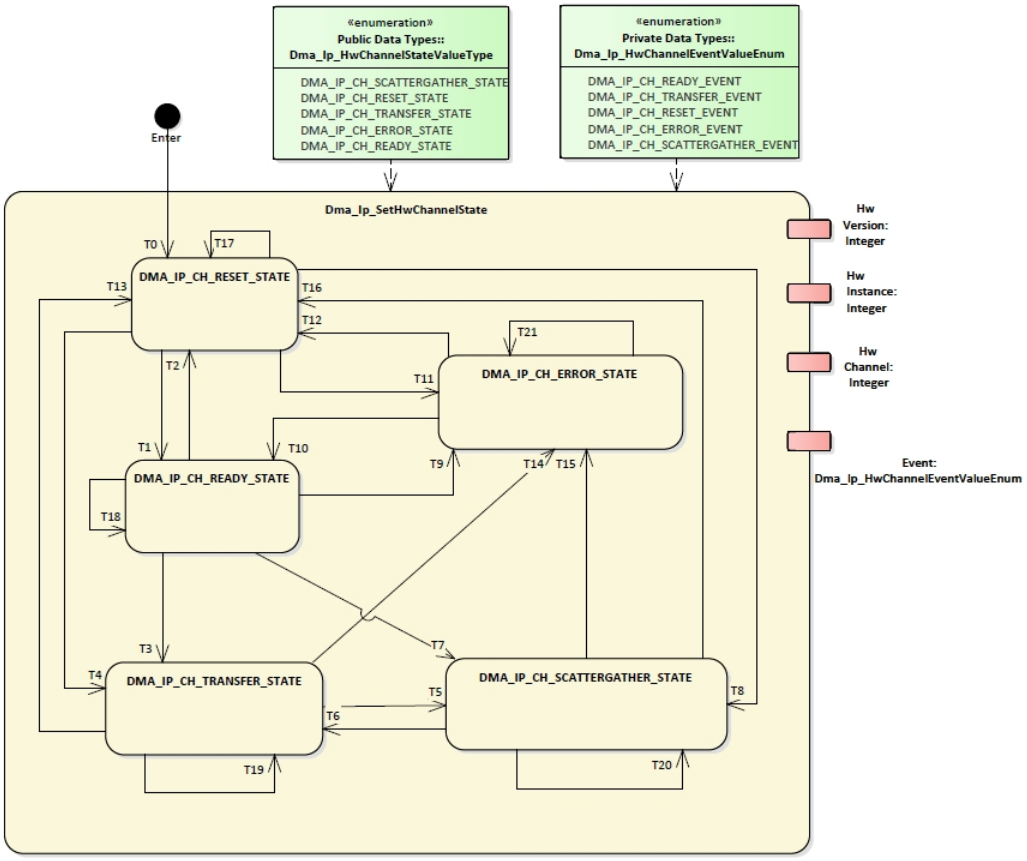

Figure 3.9 DMA IP Channel State Machine

3.7 Runtime errors

The driver generates the following DET errors at runtime.

Function	Error Code	Condition triggering the error
Mcl_Init()	MCL_E_UNINIT	API is called with a NULL pointer as parameter.
Mcl_DeInit()	MCL_E_PARAM_CONFIG	API is called with invalid configuration parameter.
Mcl_DeInit()	MCL_E_UNINIT	API is called with a NULL pointer as parameter.
Mcl_SetDmaInstanceCommand()	MCL_DET_DMA_INSTANCE↩ _COMMAND	API is called with invalid instance command.
Mcl_GetDmaInstanceStatus()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_SetDmaChannelCommand()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_SetDmaChannelCommand()	MCL_E_INVALID_COMMAND	API is called with invalid command.
Mcl_GetDmaChannelStatus()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_SetDmaChannelGlobalList()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.

Function	Error Code	Condition triggering the error
Mcl_SetDmaChannelGlobalList()	MCL_E_INVALID_↩ PARAMETER	API is called with invalid parameter.
Mcl_SetDmaChannelTransferList()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_SetDmaChannelTransferList()	MCL_E_INVALID_↩ PARAMETER	API is called with invalid parameter.
Mcl_SetDmaChannelScatterGatherList()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_SetDmaChannelScatterGatherList()	MCL_E_INVALID_↩ PARAMETER	API is called with invalid parameter.
Mcl_GetDmaChannelParam()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_GetDmaChannelParam()	MCL_E_INVALID_↩ PARAMETER	API is called with invalid parameter.
Mcl_SetDmaChannelScatterGatherConfig()	MCL_E_INVALID_CHANNEL	API is called with invalid channel.
Mcl_CacheEnable()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheDisable()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheInvalidate()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheClean()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheInvalidateByAddr()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheCleanByAddr()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.
Mcl_CacheCleanByAddr()	MCL_E_INVALID_INSTANCE	API is called with invalid instance.

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations or to use names that may trigger other compile issues (e.g. match existing `#ifdefs` arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the driver. All the parameters are described below.

- Module Mcl
  - Container MclGeneral
    - Parameter MclEnableDevErrorDetect
    - Parameter Mcl_VersionInfoApi
    - Parameter MclEnableUserModeSupport
    - Parameter MclEnableMulticoreSupport
    - Parameter MclEnableVirtualAddressMappingSupport
    - Reference MclEcucPartitionRef
    - Container MclDma
      - Parameter MclEnableDma
    - Container MclCache
      - Parameter MclEnableCache
    - Container MclTrgMux
      - Parameter MclEnableTrgMux
    - Container MclEmiosCommon
      - Parameter MclEnableEmiosCommon
    - Container MclFlexioCommon
      - Parameter MclEnableFlexioCommon
    - Container MclLcuConfig
      - Parameter MclEnableLcu
      - Parameter MclEnableLcuSyncFunc
      - Parameter MclEnableLcuAsyncFunc
  - Container MclConfig
    - Container MclVirtualMemorySection
      - Parameter MclVirtualAddressStart
      - Parameter MclVirtualAddressEnd
      - Parameter MclPhysicalAddressStart
      - Parameter MclPhysicalAddressEnd
    - Container dmaLogicInstance_ConfigType
      - Parameter dmaLogicInstance_IdName

- Parameter dmaLogicInstance_hwId
- Parameter dmaLogicInstance_enDebug
- Parameter dmaLogicInstance_enRoundRobin
- Parameter dmaLogicInstance_enHaltAfterError
- Parameter dmaLogicInstance_enChLinking
- Parameter dmaLogicInstance_enGlMasterIdReplication
- Reference dmaLogicInstance_EcucPartitionRef

* Container dmaLogicChannel_Type
  - Parameter dmaLogicChannel_LogicName
  - Parameter dmaLogicChannel_HwInstId
  - Parameter dmaLogicChannel_HwChId
  - Parameter dmaLogicChannel_InterruptCallback
  - Parameter dmaLogicChannel_ErrorInterruptCallback
  - Parameter dmaLogicChannel_EnableGlobalConfig
  - Parameter dmaLogicChannel_EnableTransferConfig
  - Parameter dmaLogicChannel_EnableScatterGather
  - Reference dmaLogicChannel_EcucPartitionRef
  - Container dmaLogicChannel_ConfigType
  - Container dmaLogicChannel_GlobalConfigType
  - Container dmaLogicChannelConfig_GlobalControlType
  - Parameter dmaGlobalControl_enMasterIdReplication
  - Parameter dmaGlobalControl_enBufferedWrites
  - Container dmaLogicChannelConfig_GlobalRequestType
  - Parameter dmaGlobalRequest_enDmaRequest
  - Container dmaLogicChannelConfig_GlobalInterruptType
  - Parameter dmaGlobalInterrupt_enDmaErrorInterrupt
  - Container dmaLogicChannelConfig_GlobalPriorityType
  - Parameter dmaGlobalPriority_GroupPriority
  - Parameter dmaGlobalPriority_LevelPriority
  - Parameter dmaGlobalPriority_enPreemption
  - Parameter dmaGlobalPriority_disPreempt
  - Container dmaLogicChannel_TransferConfigType
  - Container dmaLogicChannelConfig_TransferControlType
  - Parameter dmaLogicChannelConfig_enDmaMajorInterrupt
  - Parameter dmaLogicChannelConfig_enDmaHalfMajorInterrupt
  - Parameter dmaLogicChannelConfig_disDmaAutoHwReq
  - Parameter dmaLogicChannelConfig_enEndOfPacketSignal
  - Parameter dmaLogicChannelConfig_bandwidthControl
  - Parameter dmaLogicChannelConfig_ScatterGatherAddressType
  - Parameter dmaLogicChannelConfig_DestinationStoreAddressType
  - Container dmaLogicChannelConfig_TransferSourceType
  - Parameter dmaLogicChannelConfig_SourceAddressType
  - Parameter dmaLogicChannelConfig_SourceSignedOffsetType
  - Parameter dmaLogicChannelConfig_SourceLastAddressAdjustmentType
  - Parameter dmaTransferConfig_TransferSizeType
  - Parameter dmaLogicChannelConfig_SourceModuloType
  - Container dmaLogicChannelConfig_TransferDestinationType

- Parameter dmaLogicChannelConfig_DestinationAddressType
- Parameter dmaLogicChannelConfig_DestinationSignedOffsetType
- Parameter dmaLogicChannelConfig_DestinationLastAddressAdjustmentType
- Parameter dmaTransferConfig_TransferSizeType
- Parameter dmaLogicChannelConfig_DestinationModuloType
- Container dmaLogicChannelConfig_TransferMinorLoopType
- Parameter dmaLogicChannelConfig_enSourceOffset
- Parameter dmaLogicChannelConfig_enDestinationOffset
- Parameter dmaLogicChannelConfig_OffsetValueType
- Parameter dmaLogicChannelConfig_enMinorLoopLinkCh
- Parameter dmaLogicChannelConfig_MinorLoopSizeType
- Reference dynamic_dmaLogicChannelConfig_MinorLoopLinkChValueType
- Container dmaLogicChannelConfig_TransferMajorLoopType
- Parameter dmaLogicChannelConfig_enMajorLoopLinkCh
- Parameter dmaLogicChannelConfig_MajorLoopCountType
- Reference dynamic_dmaLogicChannelConfig_MajorLoopLinkChValueType
- Container dmaLogicChannel_ScatterGatherConfigType
- Container dmaLogicChannelConfig_ScatterGatherArrayType
- Container dmaLogicChannelConfig_ScatterGatherElementConfigType
- Parameter dmaLogicChannelConfig_ScatterGatherElementNameType
- Parameter dmaLogicChannelConfig_LastElementLink_ScatterGatherType
- Parameter dmaLogicChannelConfig_enScatterGatherConfig
- Reference dynamic_dmaLogicChannelConfig_BasicElementLink_ScatterGatherType
- Container dmaLogicChannelConfig_TransferControlType
- Parameter dmaLogicChannelConfig_enStart
- Parameter dmaLogicChannelConfig_enDmaMajorInterrupt
- Parameter dmaLogicChannelConfig_enDmaHalfMajorInterrupt
- Parameter dmaLogicChannelConfig_disDmaAutoHwReq
- Parameter dmaLogicChannelConfig_bandwidthControl
- Parameter dmaLogicChannelConfig_ScatterGatherAddressType
- Parameter dmaLogicChannelConfig_DestinationStoreAddressType
- Container dmaLogicChannelConfig_TransferSourceType
- Parameter dmaLogicChannelConfig_SourceAddressType
- Parameter dmaLogicChannelConfig_SourceSignedOffsetType
- Parameter dmaLogicChannelConfig_SourceLastAddressAdjustmentType
- Parameter dmaTransferConfig_TransferSizeType
- Parameter dmaLogicChannelConfig_SourceModuloType
- Container dmaLogicChannelConfig_TransferDestinationType
- Parameter dmaLogicChannelConfig_DestinationAddressType
- Parameter dmaLogicChannelConfig_DestinationSignedOffsetType
- Parameter dmaLogicChannelConfig_DestinationLastAddressAdjustmentType
- Parameter dmaTransferConfig_TransferSizeType
- Parameter dmaLogicChannelConfig_DestinationModuloType
- Container dmaLogicChannelConfig_TransferMinorLoopType
- Parameter dmaLogicChannelConfig_enSourceOffset
- Parameter dmaLogicChannelConfig_enDestinationOffset
- Parameter dmaLogicChannelConfig_OffsetValueType

- Parameter dmaLogicChannelConfig_enMinorLoopLinkCh
- Parameter dmaLogicChannelConfig_MinorLoopSizeType
- Reference dynamic_dmaLogicChannelConfig_MinorLoopLinkChValueType
- Container dmaLogicChannelConfig_TransferMajorLoopType
- Parameter dmaLogicChannelConfig_enMajorLoopLinkCh
- Parameter dmaLogicChannelConfig_MajorLoopCountType
- Reference dynamic_dmaLogicChannelConfig_MajorLoopLinkChValueType

* Container trgmuxInstaceList
  - Parameter trgmuxHardwareInstance
* Container trgmuxLogicGroup
  - Parameter trgmuxLogicGroupHardwareInstance
  - Parameter trgmuxLogicGroup_Name
  - Parameter trgmuxLogicGroup_Lock
  - Container trgmuxLogicTrigger
  - Parameter trgmuxLogicTrigger_Name
  - Parameter trgmuxLogicTrigger_Output
  - Parameter trgmuxLogicTrigger_Input
  - Reference trgmuxLogicTrigger_EcucPartitionRef
* Container lcuConfiguration
  - Container lcuInstanceCfg
  - Parameter lcuLogicInstance_LogicName
  - Parameter lcuLogicInstance_HwInstID
  - Parameter lcuLogicInstance_OperationMode
  - Parameter lcuLogicCell_UsingForceSignal
  - Parameter lcuLogicCell_UsingSyncSignal
  - Container lcuLogicCellCfg_lst
  - Parameter lcuLogicCell_HwLcID
  - Parameter lcuLogicCell_forceFilter
  - Parameter lcuLogicCell_usingCombinationalForcePath
  - Parameter lcuLogicCell_usingForcePolarity
  - Parameter lcuLogicCell_SwSyncSel
  - Reference lcuLogicCell_EcucPartitionRef
  - Container lcuLogicCell_ForceSignalCfg
  - Parameter lcuLogicCell_ForceSignalSelect
  - Parameter lcuLogicCell_combEnable
  - Parameter lcuLogicCell_forcePol
  - Container mclLcuInputConfiguration
  - Parameter lcuLogicInput_LogicName
  - Parameter lcuLogicInput_HwInstID
  - Parameter lcuLogicInput_HwLcID
  - Parameter lcuLogicInput_HwInputID
  - Parameter lcuLogicInput_MuxSelect
  - Parameter lcuLogicInput_UsingSwOverride
  - Parameter lcuLogicInput_SwOverrideMode
  - Parameter lcuLogicInput_SwOverrideValue
  - Container mclLcuOutputConfiguration
  - Parameter lcuLogicOutput_LogicName

- Parameter lcuLogicOutput_HwInstID
- Parameter lcuLogicOutput_HwLcID
- Parameter lcuLogicOutput_HwOutputID
- Parameter lcuLogicOutput_LutControl
- Parameter lcuLogicOutput_LutRiseFilter
- Parameter lcuLogicOutput_LutFallFilter
- Parameter lcuLogicOutput_InterruptCallback
- Parameter lcuLogicOutput_DebugMode
- Parameter lcuLogicOutput_InvertOutput
- Parameter lcuLogicOutput_LutDmaEnable
- Parameter lcuLogicOutput_LutIntEnable
- Parameter lcuLogicOutput_UsingForceSignal
- Container lcuLogicOutput_ForceSignalConfiguration
- Parameter lcuLogicOutput_ForceClearMode
- Parameter lcuLogicOutput_ForceSyncSelect
- Parameter lcuLogicOutput_ForceDmaEnable
- Parameter lcuLogicOutput_ForceIntEnable
- Container lcuLogicOutput_ForceSignalSelect
- Parameter lcuLogicOutput_ForceSignal
- Parameter lcuLogicOutput_ForceSignalAffect

* Container EmiosCommon
  - Parameter EmiosMclInstances
  - Parameter EmiosMclEnableFreezState
  - Parameter EmiosMclEnableGlobalTimeBase
  - Parameter EmiosMclClkDivVal
  - Reference EmiosCommonEcucPartitionRef
  - Container EmiosMclMasterBus
  - Parameter EmiosMclMasterBusNumber
  - Parameter EmiosMclMasterBusModeType
  - Parameter EmiosMclDefaultPeriod
  - Parameter EmiosMclFirstOffsetValue
  - Parameter EmiosMclMasterBusPrescaler
  - Parameter EmiosMclMasterBusAltPrescaler
  - Parameter EmiosMclChannelAllowDebugMode
  - Parameter EmiosMclPwmExclusiveAccess
* Container FlexioCommon
  - Parameter FlexioMclInstances
  - Parameter FlexioDebugEnable
  - Container FlexioMclLogicChannels
  - Parameter FlexioMclChannelId
  - Parameter FlexioMclPinId
  - Parameter FlexioMclAddPinEnable
  - Parameter FlexioMclAddPinId
  - Parameter FlexioMclAddChannelEnable
  - Parameter FlexioMclAddChannelId

– Container CommonPublishedInformation

* Parameter ArReleaseMajorVersion

* Parameter ArReleaseMinorVersion
* Parameter ArReleaseRevisionVersion
* Parameter ModuleId
* Parameter SwMajorVersion
* Parameter SwMinorVersion
* Parameter SwPatchVersion
* Parameter VendorId

4.1 Module Mcl

Vendor specific: Configuration of the Mcl (MicroController Library) module.

Included containers:

- MclGeneral
- MclConfig
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container MclGeneral

Vendor specific: Configuration of general Mcl parameters.

Included subcontainers:

- MclDma
- MclCache
- MclTrgMux
- MclEmiosCommon
- MclFlexioCommon
- MclLcuConfig

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Parameter MclEnableDevErrorDetect

Vendor specific:

Enable/Disable the Development Error Detection (DET).

true: Enabled.

false: Disabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.4 Parameter Mcl_VersionInfoApi

Vendor specific: Enables/Disables the get version info API function

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.5 Parameter MclEnableUserModeSupport

When this parameter is enabled, the MCL module will adapt to run from User Mode, with the following measures:

b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode.

for more information, please see chapter 5.7 User Mode Support in IM

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.6 Parameter MclEnableMulticoreSupport

Enable/Disable Multicore support

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.7 Parameter MclEnableVirtualAddressMappingSupport

Enable/Disable Virtual Address Mapping support

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.8 Reference MclEcucPartitionRef

The partition shall be used to initialize and de-initialize the Mcl Driver.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.9 Container MclDma

Vendor specific:

Container for the Dma related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.10 Parameter MclEnableDma

Vendor specific: Enable/Disable DMA support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.11 Container MclCache

Vendor specific:

Container for the CACHE related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.12 Parameter MclEnableCache

Vendor specific:

Enable/Disable all CACHE support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.13 Container MclTrgMux

Vendor specific:

Container for the TRGMUX related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.14 Parameter MclEnableTrgMux

Vendor specific: Enable/Disable TRGMUX support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.15 Container MclEmiosCommon

Vendor specific:

Container for the Emios Common related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.16 Parameter MclEnableEmiosCommon

Vendor specific: Enable/Disable Emios common support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.17 Container MclFlexioCommon

Vendor specific:

Container for the Flexio Common related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.18 Parameter MclEnableFlexioCommon

Vendor specific: Enable/Disable Flexio common support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.19 Container MclLcuConfig

Vendor specific:

Container for the LCU related configuration parameters.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.20 Parameter MclEnableLcu

Vendor specific: Enable/Disable LCU support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.21 Parameter MclEnableLcuSyncFunc

Vendor specific: Enable/Disable LCU SYNC Functions. SYNC functions support configure only one parameter for multiple inputs/outputs at the same time.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.22 Parameter MclEnableLcuAsyncFunc

Vendor specific: Enable/Disable LCU ASYNC Functions. ASYNC functions support configure multiple parameter for only one input/output at once.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.23 Container MclConfig

Vendor specific: This container is the base for a multiple configuration set

Included subcontainers:

- MclVirtualMemorySection
- dmaLogicInstance_ConfigType
- dmaLogicChannel_Type
- trgmuxInstaceList
- trgmuxLogicGroup
- lcuConfiguration
- EmiosCommon
- FlexioCommon

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.24 Container MclVirtualMemorySection

Vendor specific:

Data to configure Virtual address and Physical address.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.25 Parameter MclVirtualAddressStart

This parameter represents the Virtual Address Start.

The address must be aligned to 4 bytes.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	4294967295
min	0

4.26 Parameter MclVirtualAddressEnd

This parameter represents the Virtual Address End.

The address must be aligned to 4 bytes.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	255
max	4294967295
min	0

4.27 Parameter MclPhysicalAddressStart

This parameter represents the Physical Address Start.

The address must be aligned to 4 bytes.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	4294967295
min	0

4.28 Parameter MclPhysicalAddressEnd

This parameter represents the Physical Address End.

The address must be aligned to 4 bytes.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	255
max	4294967295
min	0

4.29 Container dmaLogicInstance_ConfigType

Vendor specific: Configuration of a DMA Instance.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.30 Parameter dmaLogicInstance_IdName

Vendor specific:

Logic Instance Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	DMA_LOGIC_INST_0

4.31 Parameter dmaLogicInstance_hwId

Vendor specific:

Select the Hardware DMA Instance.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_HW_INST_0
literals	['DMA_IP_HW_INST_0']

4.32 Parameter dmaLogicInstance_enDebug

Vendor specific:

DMA_CR[EDBG].

Enable Debug.

0 - The assertion of the system debug control input is ignored.

1 - The assertion of the system debug control input causes the eDMA to stall the start of a new channel.

Executing channels are allowed to complete.

Channel execution will resume when either the system debug control input is negated or the EDBG bit is cleared.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.33 Parameter dmaLogicInstance_enRoundRobin

Vendor specific:

DMA_CR[ERCA].

Enable Round Robin Channel Arbitration.

0 - Fixed-priority arbitration is used for channel selection within each group.

1 - Round-Robin arbitration is used for channel selection within each group.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.34 Parameter dmaLogicInstance_enHaltAfterError

Vendor specific:

DMA_CR[HOE] for eDMA2 instances or DMA_CSR[HAE] for eDMA3 instances.

Halt On/After Error.

0 - Normal operation.

1 - Any error will cause the HALT bit to be set.

Subsequently, all service requests will be ignored until the HALT bit is cleared.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.35 Parameter dmaLogicInstance_enChLinking

Vendor specific:

DMA_CR[GCLC].

Global Channel Linking Control.

0 - Channel linking is disabled for all channels.

1 - Channel linking is available and controlled by each channel's link settings.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.36 Parameter dmaLogicInstance_enGlMasterIdReplication

Vendor specific:

DMA_CSR[GMRC].Available only for eDMA3 instances.

Global master ID replication control

0 - Master ID replication is disabled for all channels

1 - Master ID replication is available and it is controlled by each channel's CHn_SBR[EMI] setting.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.37 Reference dmaLogicInstance_EcucPartitionRef

Maps a DMA instance to zero or multiple ECUC partitions to limit the access to this mappable element.

The ECUC partitions referenced are a subset of the ECUC partitions where the MCL driver is mapped to

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.38 Container dmaLogicChannel_Type

Vendor specific:

Logic Channel Configuration.

Included subcontainers:

- dmaLogicChannel_ConfigType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	32
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.39 Parameter dmaLogicChannel_LogicName

Vendor specific:

Logic Channel Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	DMA_LOGIC_CH_0

4.40 Parameter dmaLogicChannel_HwInstId

Vendor specific:

Select the Hardware DMA Instance.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_HW_INST_0
literals	['DMA_IP_HW_INST_0']

4.41 Parameter dmaLogicChannel_HwChId

Vendor specific:

Select the physical eDMA Channel.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_HW_CH_0
literals	['DMA_IP_HW_CH_0', 'DMA_IP_HW_CH_1', 'DMA_IP_HW_CH_↩ 2', 'DMA_IP_HW_CH_3', 'DMA_IP_HW_CH_4', 'DMA_IP_HW_CH_↩ 5', 'DMA_IP_HW_CH_6', 'DMA_IP_HW_CH_7', 'DMA_IP_HW_CH_8', 'DMA_IP_HW_CH_9', 'DMA_IP_HW_CH_10', 'DMA_IP_HW_CH_11', 'DMA_IP_HW_CH_12', 'DMA_IP_HW_CH_13', 'DMA_IP_HW_CH_14', 'DMA_IP_HW_CH_15', 'DMA_IP_HW_CH_16', 'DMA_IP_HW_CH_17', 'DMA_IP_HW_CH_18', 'DMA_IP_HW_CH_19', 'DMA_IP_HW_CH_20', 'DMA_IP_HW_CH_21', 'DMA_IP_HW_CH_22', 'DMA_IP_HW_CH_23', 'DMA_IP_HW_CH_24', 'DMA_IP_HW_CH_25', 'DMA_IP_HW_CH_26', 'DMA_IP_HW_CH_27', 'DMA_IP_HW_CH_28', 'DMA_IP_HW_CH_29', 'DMA_IP_HW_CH_30', 'DMA_IP_HW_CH_31']

4.42 Parameter dmaLogicChannel_InterruptCallback

Vendor specific:

User callback function to report that the transfer is half or complete depending on configuration.

NOTE: Use NULL_PTR w/o quotes. If the used string is different from NULL_PTR it will be used as the configured function name.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.43 Parameter dmaLogicChannel_ErrorInterruptCallback

Vendor specific:

User callback function

NOTE: Use NULL_PTR w/o quotes. If the used string is different from NULL_PTR it will be used as the configured function name.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.44 Parameter dmaLogicChannel_EnableGlobalConfig

Vendor specific: Enable and allocate memory for Global Configuration. Global Configuration can be configured during generation time or during runtime.

Note: If the check box is enabled, then memory space is allocated for the configuration.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.45 Parameter dmaLogicChannel_EnableTransferConfig

Vendor specific: Enable and allocate memory for Transfer Configuration. Transfer Configuration can be configured during generation time or during runtime.

Note: If the check box is enabled, then memory space is allocated for the configuration.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.46 Parameter dmaLogicChannel_EnableScatterGather

Vendor specific: Enable and allocate memory for ScatterGather Transfer Mode.

The ScatterGather Transfer Mode shall allocate memory for each Element, comprised of: Element Linkage and Element Software TCD.

The Element allocation can be done only in the configurator.

The Element Configuration can be further enabled for each individual element.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.47 Reference dmaLogicChannel_EcucPartitionRef

Maps a DMA channel to zero or multiple ECUC partitions to limit the access to this mappable element.

The ECUC partitions referenced are a subset of the ECUC partitions where the MCL driver is mapped to

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.48 Container dmaLogicChannel_ConfigType

Vendor specific: Logic Channel Configuration.

Included subcontainers:

- dmaLogicChannel_GlobalConfigType
- dmaLogicChannel_TransferConfigType
- dmaLogicChannel_ScatterGatherConfigType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.49 Container dmaLogicChannel_GlobalConfigType

Vendor specific:

Logic Channel Global Configuration.

Included subcontainers:

- dmaLogicChannelConfig_GlobalControlType
- dmaLogicChannelConfig_GlobalRequestType
- dmaLogicChannelConfig_GlobalInterruptType
- dmaLogicChannelConfig_GlobalPriorityType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.50 Container dmaLogicChannelConfig_GlobalControlType

Vendor specific:

TCD Global Control.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.51 Parameter dmaGlobalControl_enMasterIdReplication

Vendor specific: Set the Dma Channel to use the same protection level and system bus ID of the master programming the Dma Channel.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.52 Parameter dmaGlobalControl_enBufferedWrites

Vendor specific: Set the Dma Channel writes to be bufferable.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.53 Container dmaLogicChannelConfig_GlobalRequestType

Vendor specific:

TCD Request Control.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.54 Parameter dmaGlobalRequest_enDmaRequest

Vendor specific: Enable the Dma Channel Hardware Request.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.55 Container dmaLogicChannelConfig_GlobalInterruptType

Vendor specific:

TCD Global Interrupt.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.56 Parameter dmaGlobalInterrupt_enDmaErrorInterrupt

Vendor specific: Enable the Dma Channel Error Interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.57 Container dmaLogicChannelConfig_GlobalPriorityType

Vendor specific:

TCD Global Interrupt.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.58 Parameter dmaGlobalPriority_GroupPriority

Vendor specific: Set the Dma Channel Group Priority.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_GROUP_PRIO0

Property	Value
literals	['DMA_IP_GROUP_PRIO0', 'DMA_IP_GROUP_PRIO1', 'DMA_IP_↩ GROUP_PRIO2', 'DMA_IP_GROUP_PRIO3', 'DMA_IP_GROUP_PRIO4', 'DMA_IP_GROUP_PRIO5', 'DMA_IP_GROUP_PRIO6', 'DMA_IP_↩ GROUP_PRIO7', 'DMA_IP_GROUP_PRIO8', 'DMA_IP_GROUP_PRIO9', 'DMA_IP_GROUP_PRIO10', 'DMA_IP_GROUP_PRIO11', 'DMA_IP↩ _GROUP_PRIO12', 'DMA_IP_GROUP_PRIO13', 'DMA_IP_GROUP↩ _PRIO14', 'DMA_IP_GROUP_PRIO15', 'DMA_IP_GROUP_PRIO16', 'DMA_IP_GROUP_PRIO17', 'DMA_IP_GROUP_PRIO18', 'DMA_IP↩ _GROUP_PRIO19', 'DMA_IP_GROUP_PRIO20', 'DMA_IP_GROUP↩ _PRIO21', 'DMA_IP_GROUP_PRIO22', 'DMA_IP_GROUP_PRIO23', 'DMA_IP_GROUP_PRIO24', 'DMA_IP_GROUP_PRIO25', 'DMA_IP↩ _GROUP_PRIO26', 'DMA_IP_GROUP_PRIO27', 'DMA_IP_GROUP↩ _PRIO28', 'DMA_IP_GROUP_PRIO29', 'DMA_IP_GROUP_PRIO30', 'DMA_IP_GROUP_PRIO31']

4.59 Parameter dmaGlobalPriority_LevelPriority

Vendor specific: Set the Dma Channel Level Priority.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_LEVEL_PRIO0
literals	['DMA_IP_LEVEL_PRIO0', 'DMA_IP_LEVEL_PRIO1', 'DMA_IP_↩ LEVEL_PRIO2', 'DMA_IP_LEVEL_PRIO3', 'DMA_IP_LEVEL_PRIO4', 'DMA_IP_LEVEL_PRIO5', 'DMA_IP_LEVEL_PRIO6', 'DMA_IP_↩ LEVEL_PRIO7']

4.60 Parameter dmaGlobalPriority_enPreemption

Vendor specific: Enable the Dma Channel Preemption.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.61 Parameter dmaGlobalPriority_disPreempt

Vendor specific: Disable the Dma Channel Preempt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.62 Container dmaLogicChannel_TransferConfigType

Vendor specific:

Logic Channel Transfer Configuration.

Included subcontainers:

- dmaLogicChannelConfig_TransferControlType
- dmaLogicChannelConfig_TransferSourceType
- dmaLogicChannelConfig_TransferDestinationType
- dmaLogicChannelConfig_TransferMinorLoopType
- dmaLogicChannelConfig_TransferMajorLoopType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.63 Container dmaLogicChannelConfig_TransferControlType

Vendor specific:

TCD Global Interrupt.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.64 Parameter dmaLogicChannelConfig_enDmaMajorInterrupt

Vendor specific: Enable the Dma Channel major interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.65 Parameter dmaLogicChannelConfig_enDmaHalfMajorInterrupt

Vendor specific: Enable the Dma Channel half major interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.66 Parameter dmaLogicChannelConfig_disDmaAutoHwReq

Vendor specific: Disable the Dma Channel automatic request.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.67 Parameter dmaLogicChannelConfig_enEndOfPacketSignal

Vendor specific: Enable the Dma Channel end of packet signal.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.68 Parameter dmaLogicChannelConfig_bandwidthControl

Vendor specific: Set the Dma Channel bandwidth control.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_BWC_ENGINE_NO_STALL
literals	['DMA_IP_BWC_ENGINE_NO_STALL', 'DMA_IP_BWC_ENGINE_↩ HPE', 'DMA_IP_BWC_ENGINE_4CYCLE_STALL', 'DMA_IP_BWC_↩ ENGINE_8CYCLE_STALL']

4.69 Parameter dmaLogicChannelConfig_ScatterGatherAddressType

Vendor specific: ScatterGather Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0U

4.70 Parameter dmaLogicChannelConfig_DestinationStoreAddressType

Vendor specific: Destination Store Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0U

4.71 Container dmaLogicChannelConfig_TransferSourceType

Vendor specific: Transfer Source

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.72 Parameter dmaLogicChannelConfig_SourceAddressType

Vendor specific: Source Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0

4.73 Parameter dmaLogicChannelConfig_SourceSignedOffsetType

Vendor specific: Set the Dma Channel source signed offset value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	32767
min	-32767

4.74 Parameter dmaLogicChannelConfig_SourceLastAddressAdjustmentType

Vendor specific: Set the Dma Channel source signed last address adjustment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	2147483647
min	-2147483647

4.75 Parameter dmaTransferConfig_TransferSizeType

Vendor specific: Set the Dma Channel source transfer size.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_TRANSFER_SIZE_1_BYTE
literals	['DMA_IP_TRANSFER_SIZE_1_BYTE', 'DMA_IP_TRANSFER_↩ SIZE_2_BYTE', 'DMA_IP_TRANSFER_SIZE_4_BYTE', 'DMA_IP_↩ TRANSFER_SIZE_8_BYTE', 'DMA_IP_TRANSFER_SIZE_16_BYTE', 'DMA_IP_TRANSFER_SIZE_32_BYTE', 'DMA_IP_TRANSFER_SIZE↩ _64_BYTE']

4.76 Parameter dmaLogicChannelConfig_SourceModuloType

Vendor specific: Set the Dma Channel source modulo.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.77 Container dmaLogicChannelConfig_TransferDestinationType

Vendor specific: Transfer Destination

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.78 Parameter dmaLogicChannelConfig_DestinationAddressType

Vendor specific: Destination Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0

4.79 Parameter dmaLogicChannelConfig_DestinationSignedOffsetType

Vendor specific: Set the Dma Channel destination signed offset value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	32767
min	-32767

4.80 Parameter dmaLogicChannelConfig_DestinationLastAddressAdjustmentType

Vendor specific: Set the Dma Channel destination signed last address adjustment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	2147483647
min	-2147483647

4.81 Parameter dmaTransferConfig_TransferSizeType

Vendor specific: Set the Dma Channel destination transfer size.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_TRANSFER_SIZE_1_BYTE
literals	['DMA_IP_TRANSFER_SIZE_1_BYTE', 'DMA_IP_TRANSFER_↩ SIZE_2_BYTE', 'DMA_IP_TRANSFER_SIZE_4_BYTE', 'DMA_IP_↩ TRANSFER_SIZE_8_BYTE', 'DMA_IP_TRANSFER_SIZE_16_BYTE', 'DMA_IP_TRANSFER_SIZE_32_BYTE', 'DMA_IP_TRANSFER_SIZE↩ _64_BYTE']

4.82 Parameter dmaLogicChannelConfig_DestinationModuloType

Vendor specific: Set the Dma Channel destination modulo.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.83 Container dmaLogicChannelConfig_TransferMinorLoopType

Vendor specific: Transfer Minor Loop

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.84 Parameter dmaLogicChannelConfig_enSourceOffset

Vendor specific: Enable the Dma Channel minor loop source offset.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.85 Parameter dmaLogicChannelConfig_enDestinationOffset

Vendor specific: Enable the Dma Channel minor loop destination offset.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.86 Parameter dmaLogicChannelConfig_OffsetValueType

Vendor specific: Set the Dma Channel minor loop signed offset.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	1048575
min	-1048575

4.87 Parameter dmaLogicChannelConfig_enMinorLoopLinkCh

Vendor specific: Enable the Dma Channel minor loop logic channel linking.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.88 Parameter dmaLogicChannelConfig_MinorLoopSizeType

Vendor specific: Set the Dma Channel minor loop transfer size.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	1073741823
min	0

4.89 Reference dynamic_dmaLogicChannelConfig_MinorLoopLink↩ ChValueType

Vendor specific: Set the Dma Channel minor loop logic channel link.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP

Property	Value
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type

4.90 Container dmaLogicChannelConfig_TransferMajorLoopType

Vendor specific: Transfer Major Loop

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.91 Parameter dmaLogicChannelConfig_enMajorLoopLinkCh

Vendor specific: Enable the Dma Channel major loop logic channel linking.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.92 Parameter dmaLogicChannelConfig_MajorLoopCountType

Vendor specific: Sets the Dma Channel major loop count.

If minor loop channel linking is enabled, the major loop count limit is between 0 and 511.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	32767
min	0

4.93 Reference dynamic_dmaLogicChannelConfig_MajorLoopLink↩ ChValueType

Vendor specific: Set the Dma Channel major loop logic channel link.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false

Property	Value
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type

4.94 Container dmaLogicChannel_ScatterGatherConfigType

Vendor specific:

Logic Channel ScatterGather Configuration.

Included subcontainers:

- dmaLogicChannelConfig_ScatterGatherArrayType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.95 Container dmaLogicChannelConfig_ScatterGatherArrayType

Vendor specific: Logic Channel ScatterGather Configuration.

Included subcontainers:

- dmaLogicChannelConfig_ScatterGatherElementConfigType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	256
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.96 Container dmaLogicChannelConfig_ScatterGatherElementConfigType

Vendor specific: Element

Included subcontainers:

- dmaLogicChannelConfig_TransferControlType
- dmaLogicChannelConfig_TransferSourceType
- dmaLogicChannelConfig_TransferDestinationType
- dmaLogicChannelConfig_TransferMinorLoopType
- dmaLogicChannelConfig_TransferMajorLoopType

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.97 Parameter dmaLogicChannelConfig_ScatterGatherElementNameType

Vendor specific: Element Name

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0_SGA_ELEMENT_0

4.98 Parameter dmaLogicChannelConfig_LastElementLink_ScatterGatherType

Vendor specific: For non-circular lists, the last element shall have this checkbox set.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.99 Parameter dmaLogicChannelConfig_enScatterGatherConfig

Vendor specific: Enable Scatter/Gather Configuration

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.100 Reference dynamic_dmaLogicChannelConfig_BasicElement↩ Link_ScatterGatherType

Vendor specific: Element Link. Elements shall be part of the same Logic Channel.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type/dmaLogic↩ Channel_ConfigType/dmaLogicChannel_ScatterGatherConfigType/dma↩ LogicChannelConfig_ScatterGatherArrayType

4.101 Container dmaLogicChannelConfig_TransferControlType

Vendor specific:

TCD Global Interrupt.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.102 Parameter dmaLogicChannelConfig_enStart

Vendor specific: Enable the Dma Channel start service request (software request).

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.103 Parameter dmaLogicChannelConfig_enDmaMajorInterrupt

Vendor specific: Enable the Dma Channel major interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.104 Parameter dmaLogicChannelConfig_enDmaHalfMajorInterrupt

Vendor specific: Enable the Dma Channel half major interrupt.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.105 Parameter dmaLogicChannelConfig_disDmaAutoHwReq

Vendor specific: Disable the Dma Channel automatic request.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.106 Parameter dmaLogicChannelConfig_bandwidthControl

Vendor specific: Set the Dma Channel bandwidth control.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_BWC_ENGINE_NO_STALL

Property	Value
literals	['DMA_IP_BWC_ENGINE_NO_STALL', 'DMA_IP_BWC_ENGINE_↩ HPE', 'DMA_IP_BWC_ENGINE_4CYCLE_STALL', 'DMA_IP_BWC_↩ ENGINE_8CYCLE_STALL']

4.107 Parameter dmaLogicChannelConfig_ScatterGatherAddressType

Vendor specific: ScatterGather Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0U

4.108 Parameter dmaLogicChannelConfig_DestinationStoreAddressType

Vendor specific: Destination Store Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0U

4.109 Container dmaLogicChannelConfig_TransferSourceType

Vendor specific: Transfer Source

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.110 Parameter dmaLogicChannelConfig_SourceAddressType

Vendor specific: Source Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0

4.111 Parameter dmaLogicChannelConfig_SourceSignedOffsetType

Vendor specific: Set the Dma Channel source signed offset value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	32767
min	-32767

4.112 Parameter dmaLogicChannelConfig_SourceLastAddressAdjustmentType

Vendor specific: Set the Dma Channel source signed last address adjustment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	2147483647
min	-2147483647

4.113 Parameter dmaTransferConfig_TransferSizeType

Vendor specific: Set the Dma Channel source transfer size.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	DMA_IP_TRANSFER_SIZE_1_BYTE
literals	['DMA_IP_TRANSFER_SIZE_1_BYTE', 'DMA_IP_TRANSFER_↩ SIZE_2_BYTE', 'DMA_IP_TRANSFER_SIZE_4_BYTE', 'DMA_IP_↩ TRANSFER_SIZE_8_BYTE', 'DMA_IP_TRANSFER_SIZE_16_BYTE', 'DMA_IP_TRANSFER_SIZE_32_BYTE', 'DMA_IP_TRANSFER_SIZE↩ _64_BYTE']

4.114 Parameter dmaLogicChannelConfig_SourceModuloType

Vendor specific: Set the Dma Channel source modulo.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.115 Container dmaLogicChannelConfig_TransferDestinationType

Vendor specific: Transfer Destination

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.116 Parameter dmaLogicChannelConfig_DestinationAddressType

Vendor specific: Destination Address

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0

4.117 Parameter dmaLogicChannelConfig_DestinationSignedOffsetType

Vendor specific: Set the Dma Channel destination signed offset value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

Property	Value
defaultValue	0
max	32767
min	-32767

4.118 Parameter dmaLogicChannelConfig_DestinationLastAddressAdjustmentType

Vendor specific: Set the Dma Channel destination signed last address adjustment.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	2147483647
min	-2147483647

4.119 Parameter dmaTransferConfig_TransferSizeType

Vendor specific: Set the Dma Channel destination transfer size.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

Property	Value
defaultValue	DMA_IP_TRANSFER_SIZE_1_BYTE
literals	['DMA_IP_TRANSFER_SIZE_1_BYTE', 'DMA_IP_TRANSFER_↩ SIZE_2_BYTE', 'DMA_IP_TRANSFER_SIZE_4_BYTE', 'DMA_IP_↩ TRANSFER_SIZE_8_BYTE', 'DMA_IP_TRANSFER_SIZE_16_BYTE', 'DMA_IP_TRANSFER_SIZE_32_BYTE', 'DMA_IP_TRANSFER_SIZE↩ _64_BYTE']

4.120 Parameter dmaLogicChannelConfig_DestinationModuloType

Vendor specific: Set the Dma Channel destination modulo.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	31
min	0

4.121 Container dmaLogicChannelConfig_TransferMinorLoopType

Vendor specific: Transfer Minor Loop

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.122 Parameter dmaLogicChannelConfig_enSourceOffset

Vendor specific: Enable the Dma Channel minor loop source offset.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.123 Parameter dmaLogicChannelConfig_enDestinationOffset

Vendor specific: Enable the Dma Channel minor loop destination offset.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.124 Parameter dmaLogicChannelConfig_OffsetValueType

Vendor specific: Set the Dma Channel minor loop signed offset.

Property	Value
type	ECUC-INTEGER-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	1048575
min	-1048575

4.125 Parameter dmaLogicChannelConfig_enMinorLoopLinkCh

Vendor specific: Enable the Dma Channel minor loop logic channel linking.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.126 Parameter dmaLogicChannelConfig_MinorLoopSizeType

Vendor specific: Set the Dma Channel minor loop transfer size.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	1073741823
min	0

4.127 Reference dynamic_dmaLogicChannelConfig_MinorLoopLink↩ ChValueType

Vendor specific: Set the Dma Channel minor loop logic channel link.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type

4.128 Container dmaLogicChannelConfig_TransferMajorLoopType

Vendor specific: Transfer Major Loop

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.129 Parameter dmaLogicChannelConfig_enMajorLoopLinkCh

Vendor specific: Enable the Dma Channel major loop logic channel linking.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.130 Parameter dmaLogicChannelConfig_MajorLoopCountType

Vendor specific: Set the Dma Channel major loop count.

If minor loop channel linking is enabled, the major loop count limit is between 0 and 511.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

Property	Value
defaultValue	1
max	32767
min	0

4.131 Reference dynamic_dmaLogicChannelConfig_MajorLoopLink↩ ChValueType

Vendor specific: Set the Dma Channel major loop logic channel link.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Mcl/MclConfig/dmaLogicChannel_Type

4.132 Container trgmuxInstaceList

List of Instance.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.133 Parameter trgmuxHardwareInstance

Trigger Mux Hardware Instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_IP_HW_INST_0
literals	['TRGMUX_IP_HW_INST_0']

4.134 Container trgmuxLogicGroup

List of Logic Trigger Groups.

Included subcontainers:

- trgmuxLogicTrigger

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	40
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.135 Parameter trgmuxLogicGroupHardwareInstance

Trigger Mux Hardware Instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_IP_HW_INST_0
literals	['TRGMUX_IP_HW_INST_0']

4.136 Parameter trgmuxLogicGroup_Name

Logic Trigger Group.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_IP_ADC12_0
literals	['TRGMUX_IP_ADC12_0', 'TRGMUX_IP_ADC12_1', 'TRGMUX_IP_ADC12_2', 'TRGMUX_IP_LPCMP_0', 'TRGMUX_IP_LPCMP_1', 'TRGMUX_IP_LPCMP_2', 'TRGMUX_IP_BCTU', 'TRGMUX_IP_EMIOS012_ODIS', 'TRGMUX_IP_EMIOS0_CH1_4', 'TRGMUX_IP_EMIOS0_CH5_9', 'TRGMUX_IP_EMIOS0_CH10_13', 'TRGMUX_IP_EMIOS0_CH14_15', 'TRGMUX_IP_EMIOS1_CH1_4', 'TRGMUX_IP_EMIOS1_CH5_9', 'TRGMUX_IP_EMIOS1_CH10_13', 'TRGMUX_IP_EMIOS1_CH14_15', 'TRGMUX_IP_FLEXIO', 'TRGMUX_IP_SIUL2_0_3', 'TRGMUX_IP_SIUL2_4_7', 'TRGMUX_IP_SIUL2_8_11', 'TRGMUX_IP_SIUL2_12_15', 'TRGMUX_IP_LPI2C0', 'TRGMUX_IP_LPSPI0', 'TRGMUX_IP_LPSPI1', 'TRGMUX_IP_LPSPI2', 'TRGMUX_IP_LPUART0', 'TRGMUX_IP_LPUART1', 'TRGMUX_IP_LPUART2', 'TRGMUX_IP_LPUART3', 'TRGMUX_IP_LCU0_SYNC', 'TRGMUX_IP_LCU0_FORCE', 'TRGMUX_IP_LCU0_0', 'TRGMUX_IP_LCU0_1', 'TRGMUX_IP_LCU0_2', 'TRGMUX_IP_LCU1_SYNC', 'TRGMUX_IP_LCU1_FORCE', 'TRGMUX_IP_LCU1_0', 'TRGMUX_IP_LCU1_1', 'TRGMUX_IP_LCU1_2', 'TRGMUX_IP_CM7']

4.137 Parameter trgmuxLogicGroup_Lock

Logic Trigger Lock.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.138 Container trgmuxLogicTrigger

List of Logic Triggers.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	111
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.139 Parameter trgmuxLogicTrigger_Name

Logic Trigger Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_LOGIC_GROUP_0_TRIGGER↩ _0

4.140 Parameter trgmuxLogicTrigger_Output

Logic Trigger Output.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_IP_OUTPUT_ADC12_0_EXTRG_NORMAL_CONV

Property	Value
literals	['TRGMUX_IP_OUTPUT_ADC12_0_EXTRG_NORMAL_CONV', 'TRGMUX_IP_OUTPUT_ADC12_0_EXTRG_INJECTED_CONV', 'TRGMUX_IP_OUTPUT_ADC12_0_EXTRG_SYNC_START_↩ PULSE', 'TRGMUX_IP_OUTPUT_ADC12_1_EXTRG_NORMAL_↩ CONV', 'TRGMUX_IP_OUTPUT_ADC12_1_EXTRG_INJECTED_↩ CONV', 'TRGMUX_IP_OUTPUT_ADC12_1_EXTRG_SYNC_START↩ _PULSE', 'TRGMUX_IP_OUTPUT_ADC12_2_EXTRG_NORMAL_↩ CONV', 'TRGMUX_IP_OUTPUT_ADC12_2_EXTRG_INJECTED_↩ CONV', 'TRGMUX_IP_OUTPUT_ADC12_2_EXTRG_SYNC_START↩ _PULSE', 'TRGMUX_IP_OUTPUT_LPCMP_0_SAMPLE_WINDOW', 'TRGMUX_IP_OUTPUT_LPCMP_1_SAMPLE_WINDOW', 'TRGMUX↩ _IP_OUTPUT_LPCMP_2_SAMPLE_WINDOW', 'TRGMUX_IP_↩ OUTPUT_BCTU_TRG23', 'TRGMUX_IP_OUTPUT_BCTU_TRG47', 'TRGMUX_IP_OUTPUT_BCTU_TRG71', 'TRGMUX_IP_OUTPUT↩ _EMIOS012_ODIS0', 'TRGMUX_IP_OUTPUT_EMIOS012_ODIS1', 'TRGMUX_IP_OUTPUT_EMIOS012_ODIS2', 'TRGMUX_IP_OUTPUT↩ _EMIOS012_ODIS3', 'TRGMUX_IP_OUTPUT_EMIOS0_CH1_4_IPP↩ _IND_CH1', 'TRGMUX_IP_OUTPUT_EMIOS0_CH1_4_IPP_IND↩ _CH2', 'TRGMUX_IP_OUTPUT_EMIOS0_CH1_4_IPP_IND_CH3', 'TRGMUX_IP_OUTPUT_EMIOS0_CH1_4_IPP_IND_CH4', 'TRGMUX↩ _IP_OUTPUT_EMIOS0_CH5_9_IPP_IND_CH5', 'TRGMUX_IP_↩ OUTPUT_EMIOS0_CH5_9_IPP_IND_CH6', 'TRGMUX_IP_OUTPUT↩ _EMIOS0_CH5_9_IPP_IND_CH7', 'TRGMUX_IP_OUTPUT_EMIOS0↩ _CH5_9_IPP_IND_CH9', 'TRGMUX_IP_OUTPUT_EMIOS0_CH10↩ _13_IPP_IND_CH10', 'TRGMUX_IP_OUTPUT_EMIOS0_CH10_↩ 13_IPP_IND_CH11', 'TRGMUX_IP_OUTPUT_EMIOS0_CH10_13↩ _IPP_IND_CH12', 'TRGMUX_IP_OUTPUT_EMIOS0_CH10_13_↩ IPP_IND_CH13', 'TRGMUX_IP_OUTPUT_EMIOS0_CH14_15_IPP↩ _IND_CH14', 'TRGMUX_IP_OUTPUT_EMIOS0_CH14_15_IPP_↩ IND_CH15', 'TRGMUX_IP_OUTPUT_EMIOS1_CH1_4_IPP_IND↩ _CH1', 'TRGMUX_IP_OUTPUT_EMIOS1_CH1_4_IPP_IND_CH2', 'TRGMUX_IP_OUTPUT_EMIOS1_CH1_4_IPP_IND_CH3', 'TRGMUX↩ _IP_OUTPUT_EMIOS1_CH1_4_IPP_IND_CH4', 'TRGMUX_IP_↩ OUTPUT_EMIOS1_CH5_9_IPP_IND_CH5', 'TRGMUX_IP_OUTPUT↩ _EMIOS1_CH5_9_IPP_IND_CH6', 'TRGMUX_IP_OUTPUT_EMIOS1↩ _CH5_9_IPP_IND_CH7', 'TRGMUX_IP_OUTPUT_EMIOS1_CH5↩ _9_IPP_IND_CH9', 'TRGMUX_IP_OUTPUT_EMIOS1_CH10_13_↩ IPP_IND_CH10', 'TRGMUX_IP_OUTPUT_EMIOS1_CH10_13_IPP↩ _IND_CH11', 'TRGMUX_IP_OUTPUT_EMIOS1_CH10_13_IPP_↩ IND_CH12', 'TRGMUX_IP_OUTPUT_EMIOS1_CH10_13_IPP_IND↩ _CH13', 'TRGMUX_IP_OUTPUT_EMIOS1_CH14_15_IPP_IND_↩ CH14', 'TRGMUX_IP_OUTPUT_EMIOS1_CH14_15_IPP_IND_CH15', 'TRGMUX_IP_OUTPUT_FLEXIO_EXT_IN_TRG0', 'TRGMUX_IP↩ _OUTPUT_FLEXIO_EXT_IN_TRG1', 'TRGMUX_IP_OUTPUT_↩ FLEXIO_EXT_IN_TRG2', 'TRGMUX_IP_OUTPUT_FLEXIO_EXT_IN↩ _TRG3', 'TRGMUX_IP_OUTPUT_SIUL2_0_3_OUT0', 'TRGMUX_IP↩ _OUTPUT_SIUL2_0_3_OUT1', 'TRGMUX_IP_OUTPUT_SIUL2_0_3↩ _OUT2', 'TRGMUX_IP_OUTPUT_SIUL2_0_3_OUT3', 'TRGMUX_IP↩ _OUTPUT_SIUL2_4_7_OUT4', 'TRGMUX_IP_OUTPUT_SIUL2_4_7↩ _OUT5', 'TRGMUX_IP_OUTPUT_SIUL2_4_7_OUT6', 'TRGMUX_IP↩ _OUTPUT_SIUL2_4_7_OUT7', 'TRGMUX_IP_OUTPUT_SIUL2_8_11↩ _OUT8', 'TRGMUX_IP_OUTPUT_SIUL2_8_11_OUT9', 'TRGMUX↩ _IP_OUTPUT_SIUL2_8_11_OUT10', 'TRGMUX_IP_OUTPUT_↩ SIUL2_8_11_OUT11', 'TRGMUX_IP_OUTPUT_SIUL2_12_15_OUT12', 'TRGMUX_IP_OUTPUT_SIUL2_12_15_OUT13', 'TRGMUX_IP_↩ OUTPUT_SIUL2_12_15_OUT14', 'TRGMUX_IP_OUTPUT_SIUL2_↩ 12_15_OUT15', 'TRGMUX_IP_OUTPUT_LPI2C0', 'TRGMUX_IP_↩ OUTPUT_LPSPI0', 'TRGMUX_IP_OUTPUT_LPSPI1', 'TRGMUX_IP_↩

Property	Value

4.141 Parameter trgmuxLogicTrigger_Input

Logic Trigger Input.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	TRGMUX_IP_INPUT_LOGIC0_VSS

Property	Value
literals	['TRGMUX_IP_INPUT_LOGIC0_VSS', 'TRGMUX_IP_INPUT_LOGIC1_VDD', 'TRGMUX_IP_INPUT_ADC0_EOC', 'TRGMUX_IP_INPUT_ADC1_EOC', 'TRGMUX_IP_INPUT_ADC2_EOC', 'TRGMUX_IP_INPUT_CMP0_COUT', 'TRGMUX_IP_INPUT_CMP1_COUT', 'TRGMUX_IP_INPUT_CMP2_COUT', 'TRGMUX_IP_INPUT_EDMA_CH0', 'TRGMUX_IP_INPUT_EDMA_CH1', 'TRGMUX_IP_INPUT_EDMA_CH16', 'TRGMUX_IP_INPUT_EDMA_CH17', 'TRGMUX_IP_INPUT_EMIOS0_RELOAD_OUT_CH23', 'TRGMUX_IP_INPUT_EMIOS0_RELOAD_OUT_CH22', 'TRGMUX_IP_INPUT_EMIOS0_RELOAD_OUT_CH8', 'TRGMUX_IP_INPUT_EMIOS0_RELOAD_OUT_CH0', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH0', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH1', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH2', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH3', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH4', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH5', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH6', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH7', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH8', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH9', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH10', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH11', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH12', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH13', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH14', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH15', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH22', 'TRGMUX_IP_INPUT_EMIOS0_IPP_CH23', 'TRGMUX_IP_INPUT_EMIOS1_RELOAD_OUT_CH23', 'TRGMUX_IP_INPUT_EMIOS1_RELOAD_OUT_CH22', 'TRGMUX_IP_INPUT_EMIOS1_RELOAD_OUT_CH8', 'TRGMUX_IP_INPUT_EMIOS1_RELOAD_OUT_CH0', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH0', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH1', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH2', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH3', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH4', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH5', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH6', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH7', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH8', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH9', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH10', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH11', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH12', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH13', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH14', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH15', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH22', 'TRGMUX_IP_INPUT_EMIOS1_IPP_CH23', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_0', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_1', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_2', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_3', 'TRGMUX_IP_INPUT_SIUL2_IN0', 'TRGMUX_IP_INPUT_SIUL2_IN1', 'TRGMUX_IP_INPUT_SIUL2_IN2', 'TRGMUX_IP_INPUT_SIUL2_IN3', 'TRGMUX_IP_INPUT_SIUL2_IN4', 'TRGMUX_IP_INPUT_SIUL2_IN5', 'TRGMUX_IP_INPUT_SIUL2_IN6', 'TRGMUX_IP_INPUT_SIUL2_IN7', 'TRGMUX_IP_INPUT_SIUL2_IN8', 'TRGMUX_IP_INPUT_SIUL2_IN9', 'TRGMUX_IP_INPUT_SIUL2_IN10', 'TRGMUX_IP_INPUT_SIUL2_IN11', 'TRGMUX_IP_INPUT_SIUL2_IN12', 'TRGMUX_IP_INPUT_SIUL2_IN13', 'TRGMUX_IP_INPUT_SIUL2_IN14', 'TRGMUX_IP_INPUT_SIUL2_IN15', 'TRGMUX_IP_INPUT_LPI2C0_MASTER', 'TRGMUX_IP_INPUT_LPI2C0_SLAVE', 'TRGMUX_IP_INPUT_LPSPI0_END', 'TRGMUX_IP_INPUT_LPSPI0_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI1_END', 'TRGMUX_IP_INPUT_LPSPI1_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI2_END', 'TRGMUX_IP_INPUT_LPSPI2_RECEIVE', 'TRGMUX_IP_INPUT_LPUART0_TX', 'TRGMUX_IP_INPUT_LPUART0_RX', 'TRGMUX_IP_INPUT_LPUART0_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART1_TX', 'TRGMUX_IP_INPUT_LPUART1_RX', 'TRGMUX_IP_INPUT_LPUART1_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART2_TX', 'TRGMUX_IP_INPUT_LPUART2_RX', 'TRGMUX_IP_INPUT_LPUART2_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART3_TX', 'TRGMUX_IP_INPUT_LPUART3_RX', 'TRGMUX_IP_INPUT_LPUART3_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART4_TX', 'TRGMUX_IP_INPUT_LPUART4_RX', 'TRGMUX_IP_INPUT_LPUART4_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART5_TX', 'TRGMUX_IP_INPUT_LPUART5_RX', 'TRGMUX_IP_INPUT_LPUART5_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART6_TX', 'TRGMUX_IP_INPUT_LPUART6_RX', 'TRGMUX_IP_INPUT_LPUART6_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART7_TX', 'TRGMUX_IP_INPUT_LPUART7_RX', 'TRGMUX_IP_INPUT_LPUART7_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART8_TX', 'TRGMUX_IP_INPUT_LPUART8_RX', 'TRGMUX_IP_INPUT_LPUART8_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART9_TX', 'TRGMUX_IP_INPUT_LPUART9_RX', 'TRGMUX_IP_INPUT_LPUART9_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART10_TX', 'TRGMUX_IP_INPUT_LPUART10_RX', 'TRGMUX_IP_INPUT_LPUART10_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART11_TX', 'TRGMUX_IP_INPUT_LPUART11_RX', 'TRGMUX_IP_INPUT_LPUART11_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART12_TX', 'TRGMUX_IP_INPUT_LPUART12_RX', 'TRGMUX_IP_INPUT_LPUART12_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART13_TX', 'TRGMUX_IP_INPUT_LPUART13_RX', 'TRGMUX_IP_INPUT_LPUART13_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART14_TX', 'TRGMUX_IP_INPUT_LPUART14_RX', 'TRGMUX_IP_INPUT_LPUART14_RX_IDLE', 'TRGMUX_IP_INPUT_LPUART15_TX', 'TRGMUX_IP_INPUT_LPUART15_RX', 'TRGMUX_IP_INPUT_LPUART15_RX_IDLE', 'TRGMUX_IP_INPUT_LPI2C1_MASTER', 'TRGMUX_IP_INPUT_LPI2C1_SLAVE', 'TRGMUX_IP_INPUT_LPSPI3_END', 'TRGMUX_IP_INPUT_LPSPI3_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI4_END', 'TRGMUX_IP_INPUT_LPSPI4_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI5_END', 'TRGMUX_IP_INPUT_LPSPI5_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_4', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_5', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_6', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_7', 'TRGMUX_IP_INPUT_SIUL2_IN16', 'TRGMUX_IP_INPUT_SIUL2_IN17', 'TRGMUX_IP_INPUT_SIUL2_IN18', 'TRGMUX_IP_INPUT_SIUL2_IN19', 'TRGMUX_IP_INPUT_SIUL2_IN20', 'TRGMUX_IP_INPUT_SIUL2_IN21', 'TRGMUX_IP_INPUT_SIUL2_IN22', 'TRGMUX_IP_INPUT_SIUL2_IN23', 'TRGMUX_IP_INPUT_SIUL2_IN24', 'TRGMUX_IP_INPUT_SIUL2_IN25', 'TRGMUX_IP_INPUT_SIUL2_IN26', 'TRGMUX_IP_INPUT_SIUL2_IN27', 'TRGMUX_IP_INPUT_SIUL2_IN28', 'TRGMUX_IP_INPUT_SIUL2_IN29', 'TRGMUX_IP_INPUT_SIUL2_IN30', 'TRGMUX_IP_INPUT_SIUL2_IN31', 'TRGMUX_IP_INPUT_LPI2C2_MASTER', 'TRGMUX_IP_INPUT_LPI2C2_SLAVE', 'TRGMUX_IP_INPUT_LPSPI6_END', 'TRGMUX_IP_INPUT_LPSPI6_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI7_END', 'TRGMUX_IP_INPUT_LPSPI7_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_8', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_9', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_10', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_11', 'TRGMUX_IP_INPUT_SIUL2_IN32', 'TRGMUX_IP_INPUT_SIUL2_IN33', 'TRGMUX_IP_INPUT_SIUL2_IN34', 'TRGMUX_IP_INPUT_SIUL2_IN35', 'TRGMUX_IP_INPUT_SIUL2_IN36', 'TRGMUX_IP_INPUT_SIUL2_IN37', 'TRGMUX_IP_INPUT_SIUL2_IN38', 'TRGMUX_IP_INPUT_SIUL2_IN39', 'TRGMUX_IP_INPUT_SIUL2_IN40', 'TRGMUX_IP_INPUT_SIUL2_IN41', 'TRGMUX_IP_INPUT_SIUL2_IN42', 'TRGMUX_IP_INPUT_SIUL2_IN43', 'TRGMUX_IP_INPUT_SIUL2_IN44', 'TRGMUX_IP_INPUT_SIUL2_IN45', 'TRGMUX_IP_INPUT_SIUL2_IN46', 'TRGMUX_IP_INPUT_SIUL2_IN47', 'TRGMUX_IP_INPUT_LPI2C3_MASTER', 'TRGMUX_IP_INPUT_LPI2C3_SLAVE', 'TRGMUX_IP_INPUT_LPSPI8_END', 'TRGMUX_IP_INPUT_LPSPI8_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI9_END', 'TRGMUX_IP_INPUT_LPSPI9_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_12', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_13', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_14', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_15', 'TRGMUX_IP_INPUT_SIUL2_IN48', 'TRGMUX_IP_INPUT_SIUL2_IN49', 'TRGMUX_IP_INPUT_SIUL2_IN50', 'TRGMUX_IP_INPUT_SIUL2_IN51', 'TRGMUX_IP_INPUT_SIUL2_IN52', 'TRGMUX_IP_INPUT_SIUL2_IN53', 'TRGMUX_IP_INPUT_SIUL2_IN54', 'TRGMUX_IP_INPUT_SIUL2_IN55', 'TRGMUX_IP_INPUT_SIUL2_IN56', 'TRGMUX_IP_INPUT_SIUL2_IN57', 'TRGMUX_IP_INPUT_SIUL2_IN58', 'TRGMUX_IP_INPUT_SIUL2_IN59', 'TRGMUX_IP_INPUT_SIUL2_IN60', 'TRGMUX_IP_INPUT_SIUL2_IN61', 'TRGMUX_IP_INPUT_SIUL2_IN62', 'TRGMUX_IP_INPUT_SIUL2_IN63', 'TRGMUX_IP_INPUT_LPI2C4_MASTER', 'TRGMUX_IP_INPUT_LPI2C4_SLAVE', 'TRGMUX_IP_INPUT_LPSPI10_END', 'TRGMUX_IP_INPUT_LPSPI10_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI11_END', 'TRGMUX_IP_INPUT_LPSPI11_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_16', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_17', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_18', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_19', 'TRGMUX_IP_INPUT_SIUL2_IN64', 'TRGMUX_IP_INPUT_SIUL2_IN65', 'TRGMUX_IP_INPUT_SIUL2_IN66', 'TRGMUX_IP_INPUT_SIUL2_IN67', 'TRGMUX_IP_INPUT_SIUL2_IN68', 'TRGMUX_IP_INPUT_SIUL2_IN69', 'TRGMUX_IP_INPUT_SIUL2_IN70', 'TRGMUX_IP_INPUT_SIUL2_IN71', 'TRGMUX_IP_INPUT_SIUL2_IN72', 'TRGMUX_IP_INPUT_SIUL2_IN73', 'TRGMUX_IP_INPUT_SIUL2_IN74', 'TRGMUX_IP_INPUT_SIUL2_IN75', 'TRGMUX_IP_INPUT_SIUL2_IN76', 'TRGMUX_IP_INPUT_SIUL2_IN77', 'TRGMUX_IP_INPUT_SIUL2_IN78', 'TRGMUX_IP_INPUT_SIUL2_IN79', 'TRGMUX_IP_INPUT_LPI2C5_MASTER', 'TRGMUX_IP_INPUT_LPI2C5_SLAVE', 'TRGMUX_IP_INPUT_LPSPI12_END', 'TRGMUX_IP_INPUT_LPSPI12_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI13_END', 'TRGMUX_IP_INPUT_LPSPI13_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_20', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_21', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_22', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_23', 'TRGMUX_IP_INPUT_SIUL2_IN80', 'TRGMUX_IP_INPUT_SIUL2_IN81', 'TRGMUX_IP_INPUT_SIUL2_IN82', 'TRGMUX_IP_INPUT_SIUL2_IN83', 'TRGMUX_IP_INPUT_SIUL2_IN84', 'TRGMUX_IP_INPUT_SIUL2_IN85', 'TRGMUX_IP_INPUT_SIUL2_IN86', 'TRGMUX_IP_INPUT_SIUL2_IN87', 'TRGMUX_IP_INPUT_SIUL2_IN88', 'TRGMUX_IP_INPUT_SIUL2_IN89', 'TRGMUX_IP_INPUT_SIUL2_IN90', 'TRGMUX_IP_INPUT_SIUL2_IN91', 'TRGMUX_IP_INPUT_SIUL2_IN92', 'TRGMUX_IP_INPUT_SIUL2_IN93', 'TRGMUX_IP_INPUT_SIUL2_IN94', 'TRGMUX_IP_INPUT_SIUL2_IN95', 'TRGMUX_IP_INPUT_LPI2C6_MASTER', 'TRGMUX_IP_INPUT_LPI2C6_SLAVE', 'TRGMUX_IP_INPUT_LPSPI14_END', 'TRGMUX_IP_INPUT_LPSPI14_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI15_END', 'TRGMUX_IP_INPUT_LPSPI15_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_24', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_25', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_26', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_27', 'TRGMUX_IP_INPUT_SIUL2_IN96', 'TRGMUX_IP_INPUT_SIUL2_IN97', 'TRGMUX_IP_INPUT_SIUL2_IN98', 'TRGMUX_IP_INPUT_SIUL2_IN99', 'TRGMUX_IP_INPUT_SIUL2_IN100', 'TRGMUX_IP_INPUT_SIUL2_IN101', 'TRGMUX_IP_INPUT_SIUL2_IN102', 'TRGMUX_IP_INPUT_SIUL2_IN103', 'TRGMUX_IP_INPUT_SIUL2_IN104', 'TRGMUX_IP_INPUT_SIUL2_IN105', 'TRGMUX_IP_INPUT_SIUL2_IN106', 'TRGMUX_IP_INPUT_SIUL2_IN107', 'TRGMUX_IP_INPUT_SIUL2_IN108', 'TRGMUX_IP_INPUT_SIUL2_IN109', 'TRGMUX_IP_INPUT_SIUL2_IN110', 'TRGMUX_IP_INPUT_SIUL2_IN111', 'TRGMUX_IP_INPUT_LPI2C7_MASTER', 'TRGMUX_IP_INPUT_LPI2C7_SLAVE', 'TRGMUX_IP_INPUT_LPSPI16_END', 'TRGMUX_IP_INPUT_LPSPI16_RECEIVE', 'TRGMUX_IP_INPUT_LPSPI17_END', 'TRGMUX_IP_INPUT_LPSPI17_RECEIVE', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_28', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_29', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_30', 'TRGMUX_IP_INPUT_FLEXIO_EXT_OUTPUT_TRIG_31', 'TRGMUX_IP_INPUT_SIUL2_IN112', 'TRGMUX_IP_INPUT_SIUL2_IN113', 'TRGMUX_IP_INPUT_SIUL2_IN114', 'TRGMUX_IP_INPUT_SIUL2_IN115', 'TRGMUX_IP_INPUT_SIUL2_IN116', 'TRGMUX_IP_INPUT_SIUL2_IN117', 'TRGMUX_IP_INPUT_SIUL2_IN118', 'TRGMUX_IP_INPUT_SIUL2_IN119', 'TRGMUX_IP_INPUT_SIUL2_IN120', 'TRGMUX_IP_INPUT_SIUL2_IN121', 'TRGMUX_IP_INPUT_SIUL2_IN122', 'TRGMUX_IP_INPUT_SIUL2_IN123', 'TRGMUX_IP_INPUT_SIUL2_IN124', 'TRGMUX_IP_INPUT_SIUL2_IN125', 'TRGMUX_IP_INPUT_SIUL2_IN126', 'TRGMUX_IP_INPUT_SIUL2_IN127']

Property	Value

4.142 Reference trgmuxLogicTrigger_EcucPartitionRef

Maps a TRGMUX logic channel to zero or multiple ECUC partitions to limit the access to this mappable element.

The ECUC partitions referenced are a subset of the ECUC partitions where the MCL driver is mapped to

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.143 Container lcuConfiguration

Vendor specific: Configuration of LCU.

Included subcontainers:

- lcuInstanceCfg
- mclLcuInputConfiguration
- mclLcuOutputConfiguration

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.144 Container lcuInstanceCfg

Vendor specific:

Logic Instance Configuration.

Included subcontainers:

- lcuLogicCellCfg_lst

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.145 Parameter lcuLogicInstance_LogicName

Vendor specific:

Logic Instance Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	LCU_LOGIC_INSTANCE_0

4.146 Parameter lcuLogicInstance_HwInstID

Vendor specific:

Select the Hardware LCU Instance.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_INST_0
literals	['LCU_IP_HW_INST_0', 'LCU_IP_HW_INST_1']

4.147 Parameter lcuLogicInstance_OperationMode

Vendor specific:

Select the Operation mode for whole HW instance. This operation mode will be applied for all outputs which belong to this Hw Instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	INTERRUPT
literals	['INTERRUPT', 'POLLING']

4.148 Parameter lcuLogicCell_UsingForceSignal

Vendor specific:

Using Force Signal

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.149 Parameter lcuLogicCell_UsingSyncSignal

Vendor specific:

[SW_SYNC_SEL] Selects which sync input to use for software synced mode.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.150 Container lcuLogicCellCfg_lst

Vendor specific:

lcuLogicCellCfg_lst

NOTE: This is an Implementation Specific Parameter.

Included subcontainers:

- lcuLogicCell_ForceSignalCfg

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	3
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.151 Parameter lcuLogicCell_HwLcID

Vendor specific:

Select the Hardware Logic Cell (LC).

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_LC_0
literals	['LCU_IP_HW_LC_0', 'LCU_IP_HW_LC_1', 'LCU_IP_HW_LC_2']

4.152 Parameter lcuLogicCell_forceFilter

Vendor specific:

Force Filter

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	255
min	0

4.153 Parameter lcuLogicCell_usingCombinationalForcePath

Vendor specific:

Using Combinational Force Path (CFP)

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.154 Parameter lcuLogicCell_usingForcePolarity

Vendor specific:

Using Force Polarity feature

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.155 Parameter lcuLogicCell_SwSyncSel

Vendor specific:

[SW_SYNC_SEL] Selects which sync input to use for software synced mode of SW override feature.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_SYNC_SEL_INPUT0
literals	['LCU_IP_SYNC_SEL_INPUT0', 'LCU_IP_SYNC_SEL_INPUT1']

4.156 Reference lcuLogicCell_EcucPartitionRef

Maps a LCU logic cell to zero or multiple ECUC partitions to limit the access to this mappable element. The ECUC partitions referenced are a subset of the ECUC partitions where the MCL driver is mapped to,.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.157 Container lcuLogicCell_ForceSignalCfg

Vendor specific: Configuration of Enable Combinational Force Path (CFP) and Forece Polarity

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	3
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.158 Parameter lcuLogicCell_ForceSignalSelect

Vendor specific:

Select Force Signal to configure CFP, Force signal polarity

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_FORCE_SEL_INPUT0
literals	['LCU_IP_FORCE_SEL_INPUT0', 'LCU_IP_FORCE_SEL_INPUT1', 'LCU_IP_FORCE_SEL_INPUT2']

4.159 Parameter lcuLogicCell_combEnable

Vendor specific:

[COMB_EN] Enable/Disable Combinational Force Path (CFP)

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.160 Parameter lcuLogicCell_forcePol

Vendor specific:

[FORCE_POL] Invert Force signal

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.161 Container mclLcuInputConfiguration

Vendor specific:

Logic Input Configuration.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	24
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.162 Parameter lcuLogicInput_LogicName

Vendor specific:

Logic Input Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	LCU_LOGIC_INPUT_0

4.163 Parameter lcuLogicInput_HwInstID

Vendor specific:

Select the Hardware LCU Instance.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_INST_0
literals	['LCU_IP_HW_INST_0', 'LCU_IP_HW_INST_1']

4.164 Parameter lcuLogicInput_HwLcID

Vendor specific:

Select the Hardware Logic Cell (LC).

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_LC_0
literals	['LCU_IP_HW_LC_0', 'LCU_IP_HW_LC_1', 'LCU_IP_HW_LC_2']

4.165 Parameter lcuLogicInput_HwInputID

Vendor specific:

Select the Hardware input.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_INPUT_0
literals	['LCU_IP_HW_INPUT_0', 'LCU_IP_HW_INPUT_1', 'LCU_IP_HW_↩ INPUT_2', 'LCU_IP_HW_INPUT_3']

4.166 Parameter lcuLogicInput_MuxSelect

Vendor specific:

Mux Select

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_MUX_SEL_LOGIC_0
literals	['LCU_IP_MUX_SEL_LOGIC_0', 'LCU_IP_MUX_SEL_LU_IN_↩ 0', 'LCU_IP_MUX_SEL_LU_IN_1', 'LCU_IP_MUX_SEL_LU_IN_↩ 2', 'LCU_IP_MUX_SEL_LU_IN_3', 'LCU_IP_MUX_SEL_LU_IN_↩ 4', 'LCU_IP_MUX_SEL_LU_IN_5', 'LCU_IP_MUX_SEL_LU_IN_↩ 6', 'LCU_IP_MUX_SEL_LU_IN_7', 'LCU_IP_MUX_SEL_LU_IN_8', 'LCU_IP_MUX_SEL_LU_IN_9', 'LCU_IP_MUX_SEL_LU_IN_10', 'LCU_IP_MUX_SEL_LU_IN_11', 'LCU_IP_MUX_SEL_LU_OUT_0', 'LCU_IP_MUX_SEL_LU_OUT_1', 'LCU_IP_MUX_SEL_LU_OUT_2', 'LCU_IP_MUX_SEL_LU_OUT_3', 'LCU_IP_MUX_SEL_LU_OUT_4', 'LCU_IP_MUX_SEL_LU_OUT_5', 'LCU_IP_MUX_SEL_LU_OUT_6', 'LCU_IP_MUX_SEL_LU_OUT_7', 'LCU_IP_MUX_SEL_LU_OUT_8', 'LCU_IP_MUX_SEL_LU_OUT_9', 'LCU_IP_MUX_SEL_LU_OUT_10', 'LCU_IP_MUX_SEL_LU_OUT_11']

4.167 Parameter lcuLogicInput_UsingSwOverride

Vendor specific:

Using Software Override

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.168 Parameter lcuLogicInput_SwOverrideMode

Vendor specific:

Software Override Mode

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_SW_SYNC_IMMEDIATE
literals	['LCU_IP_SW_SYNC_IMMEDIATE', 'LCU_IP_SW_SYNC_ON_RISING↩ _EDGE']

4.169 Parameter lcuLogicInput_SwOverrideValue

Vendor specific:

Software Override Value

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_SW_OVERRIDE_LOGIC_LOW
literals	['LCU_IP_SW_OVERRIDE_LOGIC_LOW', 'LCU_IP_SW_OVERRIDE_↩ LOGIC_HIGH']

4.170 Container mclLcuOutputConfiguration

Vendor specific:

Logic Output Configuration.

Included subcontainers:

- lcuLogicOutput_ForceSignalConfiguration

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	24
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.171 Parameter lcuLogicOutput_LogicName

Vendor specific:

Logic Output Name.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	LCU_LOGIC_OUTPUT_0

4.172 Parameter lcuLogicOutput_HwInstID

Vendor specific:

Select the Hardware LCU Instance.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_INST_0
literals	['LCU_IP_HW_INST_0', 'LCU_IP_HW_INST_1']

4.173 Parameter lcuLogicOutput_HwLcID

Vendor specific:

Select the Hardware Logic Cell (LC).

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_LC_0
literals	['LCU_IP_HW_LC_0', 'LCU_IP_HW_LC_1', 'LCU_IP_HW_LC_2']

4.174 Parameter lcuLogicOutput_HwOutputID

Vendor specific:

Select the Hardware output.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_HW_OUTPUT_0
literals	['LCU_IP_HW_OUTPUT_0', 'LCU_IP_HW_OUTPUT_1', 'LCU_IP_↩ HW_OUTPUT_2', 'LCU_IP_HW_OUTPUT_3']

4.175 Parameter lcuLogicOutput_LutControl

Vendor specific:

Output LUT Control

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	65535
min	0

4.176 Parameter lcuLogicOutput_LutRiseFilter

Vendor specific:

Output LUT Rise Filter

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	65535
min	0

4.177 Parameter lcuLogicOutput_LutFallFilter

Vendor specific:

Output LUT Fall Filter

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	65535
min	0

4.178 Parameter lcuLogicOutput_InterruptCallback

Vendor specific:

The Callback is defined by the user and installed by the driver in the corresponding IRQ.

NOTE: please use NULL or NULL_PTR w/o any quotes. If the used string is different from NULL

or NULL_PTR it will be used as the configured function name.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.179 Parameter lcuLogicOutput_DebugMode

Vendor specific:

[DBGEN] Enable Debug Mode

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.180 Parameter lcuLogicOutput_InvertOutput

Vendor specific:

[OUTPOL] Set Output Polarity: invert or not

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.181 Parameter lcuLogicOutput_LutDmaEnable

Vendor specific:

[LUT_DMA_EN] Enables/Disable the generation of a DMA request when an LUT event occurs

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.182 Parameter lcuLogicOutput_LutIntEnable

Vendor specific:

[LUT_INT_EN] Enables the generation of an interrupt request when an LUT event occurs

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.183 Parameter lcuLogicOutput_UsingForceSignal

Vendor specific:

Enable this node to configure for FORCE signal

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.184 Container lcuLogicOutput_ForceSignalConfiguration

Vendor specific: Force Signal Configuration.

Included subcontainers:

- lcuLogicOutput_ForceSignalSelect

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.185 Parameter lcuLogicOutput_ForceClearMode

Vendor specific:

[FORCE_MODE] Specifies the timing for clearing force events for output.

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_CLEAR_FORCE_SIGNAL_IMMEDIATE
literals	['LCU_IP_CLEAR_FORCE_SIGNAL_IMMEDIATE', 'LCU_IP_CLEAR↩ _FORCE_SIGNAL_ON_RISING_EDGE', 'LCU_IP_CLEAR_FORCE↩ _SIGNAL_AFTER_CLEAR_STATUS', 'LCU_IP_CLEAR_FORCE_↩ SIGNAL_ON_RISING_EDGE_AFTER_CLEAR_STATUS']

4.186 Parameter lcuLogicOutput_ForceSyncSelect

Vendor specific:

[SYNC_SEL] Force Sync signal Selection.Selects which sync input to use for output

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_SYNC_SEL_INPUT0
literals	['LCU_IP_SYNC_SEL_INPUT0', 'LCU_IP_SYNC_SEL_INPUT1']

4.187 Parameter lcuLogicOutput_ForceDmaEnable

Vendor specific:

[FORCE_DMA_EN] Enables the generation of a DMA request when a force event occurs

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.188 Parameter lcuLogicOutput_ForceIntEnable

Vendor specific:

[FORCE_INT_EN] Enables the generation of an interrupt request when a force event occurs

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.189 Container lcuLogicOutput_ForceSignalSelect

Vendor specific:

Configure force signal for output

NOTE: This is an Implementation Specific Parameter.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	3
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.190 Parameter lcuLogicOutput_ForceSignal

Vendor specific:

Select force signal to specify force signal affect for output

NOTE: This is an Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LCU_IP_FORCE_SEL_INPUT0
literals	['LCU_IP_FORCE_SEL_INPUT0', 'LCU_IP_FORCE_SEL_INPUT1', 'LCU_IP_FORCE_SEL_INPUT2']

4.191 Parameter lcuLogicOutput_ForceSignalAffect

Vendor specific: [FORCE_SENSE] Selects force input affect output or not.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.192 Container EmiosCommon

List of EMIOS instances available on the platform.

Included subcontainers:

- EmiosMclMasterBus

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	3
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.193 Parameter EmiosMclInstances

Select one of the EMIOS instance available on the platform.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	EMIOS_0
literals	['EMIOS_0', 'EMIOS_1', 'EMIOS_2']

4.194 Parameter EmiosMclEnableFreezState

Allow all channel in EMIOS group can enter debug mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.195 Parameter EmiosMclEnableGlobalTimeBase

Enable global timebase or disable on EMIOS IP.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.196 Parameter EmiosMclClkDivVal

Select the clock divider value for the global prescaler in range (1-256).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	256
min	1

4.197 Reference EmiosCommonEcucPartitionRef

Maps a EMIOS instance to zero or multiple ECUC partitions to limit the access to this mappable element. The ECUC partitions referenced are a subset of the ECUC partitions where the MCL driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE

Property	Value
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.198 Container EmiosMclMasterBus

EMIOS Master Bus Configuration

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	5
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.199 Parameter EmiosMclMasterBusNumber

Select one of the EMIOS channels available on the platform for selected instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	EMIOS_CH_0
literals	['EMIOS_CH_0', 'EMIOS_CH_8', 'EMIOS_CH_16', 'EMIOS_CH_22', 'EMIOS_CH_23']

4.200 Parameter EmiosMclMasterBusModeType

Configures master bus mode of this channel

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MC_UP_COUNTER_START
literals	['MC_UP_COUNTER_START', 'MC_UP_COUNTER_END', 'MC_UP↩ _DOWN_COUNTER', 'MCB_UP_COUNTER', 'MCB_UP_DOWN_↩ COUNTER']

4.201 Parameter EmiosMclDefaultPeriod

<h>IMPORTANT!</h> When Master Bus Mode Type node is selected in a MCB mode the counter starts from 1 and counts to the period value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	65535
min	0

4.202 Parameter EmiosMclFirstOffsetValue

<h>IMPORTANT!</h> This offset applies only when channel is started.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	65534
min	0

4.203 Parameter EmiosMclMasterBusPrescaler

EmiosMclMasterBusPrescaler

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	DIV_1
literals	['DIV_1', 'DIV_2', 'DIV_3', 'DIV_4', 'DIV_5', 'DIV_6', 'DIV_7', 'DIV_↩ 8', 'DIV_9', 'DIV_10', 'DIV_11', 'DIV_12', 'DIV_13', 'DIV_14', 'DIV_15', 'DIV_16']

4.204 Parameter EmiosMclMasterBusAltPrescaler

EmiosMclMasterBusAltPrescaler

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	DIV_1
literals	['DIV_1', 'DIV_2', 'DIV_3', 'DIV_4', 'DIV_5', 'DIV_6', 'DIV_7', 'DIV_↩ 8', 'DIV_9', 'DIV_10', 'DIV_11', 'DIV_12', 'DIV_13', 'DIV_14', 'DIV_15', 'DIV_16']

4.205 Parameter EmiosMclChannelAllowDebugMode

EmiosMclChannelAllowDebugMode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.206 Parameter EmiosMclPwmExclusiveAccess

PWM exclusive access on current counter bus.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.207 Container FlexioCommon

List of Flexio instances available on the platform.

Included subcontainers:

- FlexioMclLogicChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	8
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

4.208 Parameter FlexioMclInstances

Select one of the Flexio instance available on the platform.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	FLEXIO_0
literals	['FLEXIO_0']

4.209 Parameter FlexioDebugEnable

Enable Debug Mode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.210 Container FlexioMclLogicChannels

Flexio Logic Channel Configuration

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	8
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.211 Parameter FlexioMclChannelId

Select one of the Flexio channels available on the platform for selected instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CHANNEL_0
literals	['CHANNEL_0', 'CHANNEL_1', 'CHANNEL_2', 'CHANNEL_3', 'CHANNEL_4', 'CHANNEL_5', 'CHANNEL_6', 'CHANNEL_7']

4.212 Parameter FlexioMclPinId

Select one of the Flexio pins available on the platform for selected instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	PIN_0
literals	['PIN_0', 'PIN_1', 'PIN_2', 'PIN_3', 'PIN_4', 'PIN_5', 'PIN_6', 'PIN_7', 'PIN_8', 'PIN_9', 'PIN_10', 'PIN_11', 'PIN_12', 'PIN_13', 'PIN_14', 'PIN_15', 'PIN_16', 'PIN_17', 'PIN_18', 'PIN_19', 'PIN_20', 'PIN_21', 'PIN_22', 'PIN_23', 'PIN_24', 'PIN_25', 'PIN_26', 'PIN_27', 'PIN_28', 'PIN_29', 'PIN_30', 'PIN_31']

4.213 Parameter FlexioMclAddPinEnable

Enable feature to select one more Flexio pin.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.214 Parameter FlexioMclAddPinId

Select one of the Flexio pins available on the platform for selected instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	PIN_1
literals	['PIN_0', 'PIN_1', 'PIN_2', 'PIN_3', 'PIN_4', 'PIN_5', 'PIN_6', 'PIN_7', 'PIN_8', 'PIN_9', 'PIN_10', 'PIN_11', 'PIN_12', 'PIN_13', 'PIN_14', 'PIN↩ _15', 'PIN_16', 'PIN_17', 'PIN_18', 'PIN_19', 'PIN_20', 'PIN_21', 'PIN_22', 'PIN_23', 'PIN_24', 'PIN_25', 'PIN_26', 'PIN_27', 'PIN_28', 'PIN_29', 'PIN↩ _30', 'PIN_31']

4.215 Parameter FlexioMclAddChannelEnable

Enable feature to select one more Flexio channel.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.216 Parameter FlexioMclAddChannelId

Select one of the Flexio channels available on the platform for selected instance.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	CHANNEL_0
literals	['CHANNEL_0', 'CHANNEL_1', 'CHANNEL_2', 'CHANNEL_3', 'CHANNEL_4', 'CHANNEL_5', 'CHANNEL_6', 'CHANNEL_7']

4.217 Container CommonPublishedInformation

Vendor specific:

Common container, aggregated by all modules. It contains published information about vendor and versions.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.218 Parameter ArReleaseMajorVersion

Vendor specific:

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.219 Parameter ArReleaseMinorVersion

Vendor specific:

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	7
max	7
min	7

4.220 Parameter ArReleaseRevisionVersion

Vendor specific:

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.221 Parameter ModuleId

Vendor specific:

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	255
max	255
min	255

4.222 Parameter SwMajorVersion

Vendor specific:

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	3
max	3
min	3

4.223 Parameter SwMinorVersion

Vendor specific:

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP

Property	Value
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.224 Parameter SwPatchVersion

Vendor specific:

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.225 Parameter VendorId

Vendor specific:

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

This chapter describes the Tresos configuration plug-in for the *driver* Driver. The most of the parameters are described below.

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 CACHE IP Driver

6.1.1 Detailed Description

Enum Reference

- enum Cache_Ip_Type

  This type contains the Cache Ip types.
- enum Cache_Ip_BusType

  This type contains the Cache Ip Bus types.

Function Reference

- Std_ReturnType Cache_Ip_Enable (const Cache_Ip_Type CacheType, const Cache_Ip_BusType BusType)

  This function enables the Cache Ip Driver.
- Std_ReturnType Cache_Ip_Disable (const Cache_Ip_Type CacheType, const Cache_Ip_BusType Bus↩ Type)

  This function disables the Cache Ip Driver.
- Std_ReturnType Cache_Ip_Invalidate (const Cache_Ip_Type CacheType, const Cache_Ip_BusType Bus↩ Type)

  This function Invalidates the Cache Ip Driver.
- Std_ReturnType Cache_Ip_Clean (const Cache_Ip_Type CacheType, const Cache_Ip_BusType BusType, const boolean EnInvalidate)

  This function Clean the Cache Ip Driver.
- Std_ReturnType Cache_Ip_InvalidateByAddr (const Cache_Ip_Type CacheType, const Cache_Ip_BusType BusType, const uint32 Addr, const uint32 Length)

  This function Invalidates By Address the Cache Ip Driver.
- Std_ReturnType Cache_Ip_CleanByAddr (const Cache_Ip_Type CacheType, const Cache_Ip_BusType BusType, const boolean EnInvalidate, const uint32 Addr, const uint32 Length)

  This function Clean By Address the Cache Ip Driver.

6.1.2 Enum Reference

6.1.2.1 Cache_Ip_Type

```
enum Cache_Ip_Type
```

This type contains the Cache Ip types.

The cache types specifies what type of cache shall be used when calling the interface. The CACHE_IP_LMEM select LMEM caches. The CACHE_IP_CORE select CORE caches.

Definition at line 109 of file Cache_Ip.h.

6.1.2.2 Cache_Ip_BusType

```
enum Cache_Ip_BusType
```

This type contains the Cache Ip Bus types.

The cache bus types specifies what type of cache shall be used when calling the interface. The CACHE_IP_PC↩ _BUS selects Processor Code (PC) bus (used with Cache Lmem). The CACHE_IP_PS_BUS selects Processor System (PS) bus (used with Cache Lmem). The CACHE_IP_ALL_BUS selects PC and PS bus (used with Cache Lmem). The CACHE_IP_INSTRUCTION selects instruction cache (used with Cache Core). The CACHE_IP_↩ DATA selects data cache (used with Cache Core).

Definition at line 124 of file Cache_Ip.h.

6.1.3 Function Reference

6.1.3.1 Cache_Ip_Enable()

```
Std_ReturnType Cache_Ip_Enable (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType )
```

This function enables the Cache Ip Driver.

This service is a reentrant function that shall enable the Cache Ip driver.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.

Returns

E_OK is returned if the enable action finished correctly. E_NOT_OK is returned if the enable action finished incorrectly.

6.1.3.2 Cache_Ip_Disable()

```
Std_ReturnType Cache_Ip_Disable (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType )
```

This function disables the Cache Ip Driver.

This service is a reentrant function that shall disables the Cache Ip driver.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.

Returns

E_OK is returned if the disable action finished correctly. E_NOT_OK is returned if the disable action finished incorrectly.

6.1.3.3 Cache_Ip_Invalidate()

```
Std_ReturnType Cache_Ip_Invalidate (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType )
```

This function Invalidates the Cache Ip Driver.

This service is a reentrant function that shall Invalidates the Cache Ip driver. The Invalidate operation applies for the entire selected Cache Type.

A cache invalidate operation ensures that updates made visible by observers that access memory at the point to which the invalidate is defined are made visible to an observer that controls the cache. This might result in the loss of updates to the locations affected by the invalidate operation that have been written by observers that access the cache. If the address of an entry on which the invalidate operates does not have a Normal Cacheable attribute, or if the cache is disabled, then an invalidate operation also ensures that this address is not present in the cache.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.

Returns

E_OK is returned if the invalidation finished correctly. E_NOT_OK is returned if the invalidation finished incorrectly.

6.1.3.4 Cache_Ip_Clean()

```
Std_ReturnType Cache_Ip_Clean (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType,
            const boolean EnInvalidate )
```

This function Clean the Cache Ip Driver.

This service is a reentrant function that shall Clean the Cache Ip driver. The Clean operation applies for the entire selected Cache Type.

A cache clean operation ensures that updates made by an observer that controls the cache are made visible to other observers that can access memory at the point to which the operation is performed. Once the Clean has completed, the new memory values are guaranteed to be visible to the point to which the operation is performed, for example to the point of unification. The cleaning of a cache entry from a cache can overwrite memory that has been written by another observer only if the entry contains a location that has been written to by an observer in the shareability domain of that memory location.

A cache clean and invalidate operation behaves as the execution of a clean operation followed immediately by an invalidate operation. Both operations are performed to the same location.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.
in	*enInvalidate*	Specifies to execute operation Clean&Invalidate.

Returns

E_OK is returned if the cleaning finished correctly. E_NOT_OK is returned if the cleaning finished incorrectly.

6.1.3.5 Cache_Ip_InvalidateByAddr()

```
Std_ReturnType Cache_Ip_InvalidateByAddr (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType,
            const uint32 Addr,
            const uint32 Length )
```

This function Invalidates By Address the Cache Ip Driver.

This service is a reentrant function that shall Invalidates By Address the Cache Ip driver. The Invalidate By Address operation applies for the memory segment specified by the start Address and Length. The operation Invalidates only multiple of Cache Line Size, thus the specified memory segment shall be aligned and multiple of the Cache Line Size.

A cache invalidate operation ensures that updates made visible by observers that access memory at the point to which the invalidate is defined are made visible to an observer that controls the cache. This might result in the loss of updates to the locations affected by the invalidate operation that have been written by observers that access the cache. If the address of an entry on which the invalidate operates does not have a Normal Cacheable attribute, or if the cache is disabled, then an invalidate operation also ensures that this address is not present in the cache.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.
in	*Addr*	Specifies the memory segment start address.
in	*Length*	Specifies the memory segment length.

Returns

E_OK is returned if the invalidation finished correctly. E_NOT_OK is returned if the invalidation finished incorrectly.

6.1.3.6 Cache_Ip_CleanByAddr()

```
Std_ReturnType Cache_Ip_CleanByAddr (
            const Cache_Ip_Type CacheType,
            const Cache_Ip_BusType BusType,
            const boolean EnInvalidate,
            const uint32 Addr,
            const uint32 Length )
```

This function Clean By Address the Cache Ip Driver.

This service is a reentrant function that shall Clean By Address the Cache Ip driver. The Clean By Address operation applies for the memory segment specified by the start Address and Length. The operation Cleans only multiple of Cache Line Size, thus the specified memory segment shall be aligned and multiple of the Cache Line Size.

A cache clean operation ensures that updates made by an observer that controls the cache are made visible to other observers that can access memory at the point to which the operation is performed. Once the Clean has completed, the new memory values are guaranteed to be visible to the point to which the operation is performed, for example to the point of unification. The cleaning of a cache entry from a cache can overwrite memory that has been written by another observer only if the entry contains a location that has been written to by an observer in the shareability domain of that memory location.

A cache clean and invalidate operation behaves as the execution of a clean operation followed immediately by an invalidate operation. Both operations are performed to the same location.

Parameters

in	*CacheType*	Specifies the cache type.
in	*BusType*	Specifies the bus type.
in	*enInvalidate*	Specifies to execute operation Clean&Invalidate.
in	*Addr*	Specifies the memory segment start address.
in	*Length*	Specifies the memory segment length.

Returns

E_OK is returned if the cleaning finished correctly. E_NOT_OK is returned if the cleaning finished incorrectly.

6.2 MCL Driver

6.2.1 Detailed Description

Data Structures

- struct Mcl_DmaChannelGlobalListType

  This type contains the Mcl Dma Channel Global List. More...
- struct Mcl_DmaChannelTransferListType

  This type contains the Mcl Dma Channel Transfer List. More...
- struct Mcl_DmaChannelScatterGatherListType

  This type contains the Mcl Dma Channel Scatter/Gather List. More...
- struct Mcl_DmaInstanceStatusType

  This type contains the Mcl Dma Instance Status. More...
- struct Mcl_DmaChannelStatusType

  This type contains the Mcl Dma Channel Status. More...
- struct Mcl_ConfigType

  This type contains the Mcl Configuration. More...

Macros

- #define MCL_DET_INIT

  API service ID for Mcl_Init function.
- #define MCL_DET_DMA_INSTANCE_COMMAND

  API service ID for Mcl_SetDmaInstanceCommand function.
- #define MCL_DET_DMA_INSTANCE_STATUS

  API service ID for Mcl_GetDmaInstanceStatus function.
- #define MCL_DET_DMA_CHANNEL_COMMAND

  API service ID for Mcl_SetDmaChannelCommand function.
- #define MCL_DET_DMA_CHANNEL_STATUS

  API service ID for Mcl_GetDmaChannelStatus function.
- #define MCL_DET_DMA_GLOBAL

  API service ID for Mcl_SetDmaChannelGlobalList function.
- #define MCL_DET_DMA_TRANSFER

  API service ID for Mcl_SetDmaChannelTransferList function.
- #define MCL_DET_DMA_SCATTER_GATHER_LIST

  API service ID for Mcl_SetDmaChannelScatterGatherList function.
- #define MCL_DET_DMA_INFORMATION

  API service ID for Mcl_GetDmaChannelParam function.
- #define MCL_DET_DMA_SCATTER_GATHER_CONFIG

  API service ID for Mcl_SetDmaChannelScatterGatherConfig function.
- #define MCL_DET_DEINIT

  API service ID for Mcl_DeInit function.
- #define MCL_DET_CACHE_ENABLE

  API service ID for Mcl_CacheEnable function.
- #define MCL_DET_CACHE_DISABLE

  API service ID for Mcl_CacheDisable function.
- #define MCL_DET_CACHE_INVALIDATE

  API service ID for Mcl_CacheInvalidate function.
- #define MCL_DET_CACHE_CLEAN

  API service ID for Mcl_CacheClean function.
- #define MCL_DET_CACHE_INVALIDATE_BY_ADDRESS

  API service ID for Mcl_CacheInvalidateByAddr function.
- #define MCL_DET_CACHE_CLEAN_BY_ADDRESS

  API service ID for Mcl_CacheCleanByAddr function.
- #define MCL_DET_TRGMUX_INPUT

  API service ID for Mcl_SetTrgMuxInput function.
- #define MCL_DET_TRGMUX_LOCK

  API service ID for Mcl_SetTrgMuxLock function.
- #define MCL_DET_LCU_ASYNC_SET_INPUT

  API service ID for Mcl_SetLcuAsyncInputList function.
- #define MCL_DET_LCU_ASYNC_SET_OUTPUT

  API service ID for Mcl_SetLcuAsyncOutputList function.
- #define MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_EN

  API service ID for Mcl_SetLcuSyncInputSwOverrideEnable function.
- #define MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_VALUE

API service ID for Mcl_SetLcuSyncInputSwOverrideValue function.

- #define MCL_DET_LCU_SYNC_SET_INPUT_MUX_SEL

  API service ID for Mcl_SetLcuSyncInputMuxSelect function.
- #define MCL_DET_LCU_SYNC_SET_INPUT_SW_SYNC_MODE

  API service ID for Mcl_SetLcuSyncInputSwSyncMode function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_DEBUG_MODE_EN

  API service ID for Mcl_SetLcuSyncOutputDebugMode function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_ENABLE

  API service ID for Mcl_SetLcuSyncOutputEnable function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SENSITIVITY

  API service ID for Mcl_SetLcuSyncOutputForceInputSensitivity function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_CLEAR_MODE

  API service ID for Mcl_SetLcuSyncOutputForceClearingMode function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SYNC_SEL

  API service ID for Mcl_SetLcuSyncOutputForceSyncSelect function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_POLARITY

  API service ID for Mcl_SetLcuSyncOutputPolarity function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_DMA

  API service ID for Mcl_SetLcuSyncOutputForceDma function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_INT

  API service ID for Mcl_SetLcuSyncOutputForceInt function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_DMA

  API service ID for Mcl_SetLcuSyncOutputLutDma function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_INT

  API service ID for Mcl_SetLcuSyncOutputLutInt function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_FALL_FILTER

  API service ID for Mcl_SetLcuSyncOutputFallFilter function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_RISE_FILTER

  API service ID for Mcl_SetLcuSyncOutputRiseFilter function.
- #define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_CONTROL

  API service ID for Mcl_SetLcuSyncOutputLutControl function.
- #define MCL_DET_LCU_SYNC_CLEAR_OUTPUT_FORCE_EVENT

  API service ID for Mcl_ClearLcuSyncOutputForceEvent function.
- #define MCL_DET_LCU_SYNC_GET_LOGIC_INPUT

  API service ID for Mcl_GetLcuSyncLogicInput function.
- #define MCL_DET_LCU_SYNC_GET_SW_OVERRIDE_INPUT

  API service ID for Mcl_GetLcuSyncSwOverrideInput function.
- #define MCL_DET_LCU_SYNC_GET_LOGIC_OUTPUT

  API service ID for Mcl_GetLcuSyncLogicOutput function.
- #define MCL_DET_LCU_SYNC_GET_FORCE_OUTPUT

  API service ID for Mcl_GetLcuSyncForceOutput function.
- #define MCL_DET_LCU_SYNC_GET_FORCE_STATUS

  API service ID for Mcl_GetLcuSyncForceStatus function.
- #define MCL_DET_LCU_SYNC_GET_COMBINE_FORCE_INPUT

  API service ID for Mcl_GetLcuSyncCombineForceInput function.
- #define MCL_DET_LCU_SET_WRITE_PROTECT

  API service ID for Mcl_GetLcuSyncCombineForceInput function.

- #define MCL_DET_LCU_ASYNC_GET_INPUT_INFO

  API service ID for Mcl_SetLcuAsyncInputList function.
- #define MCL_DET_LCU_ASYNC_GET_OUTPUT_INFO

  API service ID for Mcl_SetLcuAsyncOutputList function.
- #define MCL_DET_EMIOS_E_INVALID_CHANNEL

  All API's called with wrong logic channel shall return this error.
- #define MCL_DET_EMIOS_E_INVALID_SET

  All API's called with wrong logic channel shall return this error.
- #define MCL_DET_TRGMUX_SET_INPUT

  API service ID for Mcl_SetTrgMuxInput function.
- #define MCL_DET_TRGMUX_SET_LOC

  API service ID for Mcl_SetTrgMuxLock function.
- #define MCL_E_UNINIT

  All API's having pointers as parameters shall return this error if called with with a NULL value.
- #define MCL_E_PARAM_POINTER

  All API's having pointers as parameters shall return this error if called with with a NULL value.
- #define MCL_E_INVALID_INSTANCE

  All API's called with wrong instance shall return this error.
- #define MCL_E_INVALID_CHANNEL

  All API's called with wrong channel shall return this error.
- #define MCL_E_INVALID_COMMAND

  All API's called with wrong instance shall return this error.
- #define MCL_E_INVALID_PARAMETER

  All API's called with wrong read parameter shall return this error.
- #define MCL_E_INVALID_STATE

  All API's called in wrong sequence shall return this error.
- #define MCL_E_INCONSISTENCY

  All API's called while hardware has error status shall return this error.
- #define MCL_E_TIMEOUT

  All API's called with a timeout value shall return this error if execution is not finished in the allocated timeframe.
- #define MCL_E_PARAM_CONFIG

  If DET error reporting is enabled, the MCL will check upon each API call if the requested resource is configured to be available on the current core, and in case of error will return MCL_E_PARAM_CONFIG.
- #define MCL_E_PROTECTED

  If DET error reporting is enabled, the MCL will check if registers are protected.
- #define MCL_E_INIT_FAILED

  If VariantPreCompile is used, the configuration pointer shall have a NULL_PTR value. If VariantPostBuild is used, the configuration pointer shall be different from NULL_PTR. And in case of violate will return MCL_E_INIT_↩ FAILED.

Types Reference

- typedef Lcu_Ip_SyncInputValueType Mcl_LcuSyncInputValueType

  This type contains the Mcl Lcu Multiple Inputs and Multiple Value.
- typedef Lcu_Ip_SyncOutputValueType Mcl_LcuSyncOutputValueType

  This type contains the Mcl Lcu Multiple Outputs and Multiple Value.
- typedef Lcu_Ip_AsyncInputValueType Mcl_LcuAsyncInputValueType

  This type contains the Mcl Lcu Input with multiple value.
- typedef Lcu_Ip_AsyncOutputValueType Mcl_LcuAsyncOutputValueType

  This type contains the Mcl Lcu Output with multiple value.

Enum Reference

- enum Mcl_DmaInstanceCmdType

  This type contains the Mcl Dma Instance Commands.
- enum Mcl_DmaChannelCmdType

  This type contains the Mcl Dma Channel Commands.
- enum Mcl_DmaChannelGlobalParamType

  This type contains the Mcl Dma Channel Global Parameters.
- enum Mcl_DmaChannelTransferParamType

  This type contains the Mcl Dma Channel Transfer Parameters.
- enum Mcl_DmaChannelStateType

  This type contains the Mcl Dma Channel State values.
- enum Mcl_DmaChannelInfoParamType

  This type contains the Mcl Dma Channel Information Parameters.
- enum Mcl_CacheType

  This type contains the Mcl Cache Type selection.
- enum Mcl_CacheBusType

  This type contains the Mcl Cache Bus Type selection.
- enum Mcl_LcuInputParamType

  This type contains the LCU Input Param Type.
- enum Mcl_LcuOutputParamType

  This type contains the LCU Output Param Type.
- enum Mcl_LcuInputInfoParamType

  This type contains the LCU Param Type to get information for input.
- enum Mcl_LcuOutputInfoParamType

  This type contains the LCU Param Type to get information for output.
- enum Mcl_ReturnType

  This type contains the Mcl Return Type.

Function Reference

- void Mcl_Init (const Mcl_ConfigType *const ConfigPtr)

  This function initializes the Mcl Driver.
- void Mcl_DeInit (void)

  This function deinitializes the Mcl Driver.
- void Mcl_SetDmaInstanceCommand (const uint32 Instance, const Mcl_DmaInstanceCmdType Command)

  This function sets Dma Instance Command.
- void Mcl_GetDmaInstanceStatus (const uint32 Instance, Mcl_DmaInstanceStatusType *const Status)

  This function gets Dma Instance Status.
- void Mcl_SetDmaChannelCommand (const uint32 Channel, const Mcl_DmaChannelCmdType Command)

  This function sets Dma Channel Command.
- void Mcl_GetDmaChannelStatus (const uint32 Channel, Mcl_DmaChannelStatusType *const Status)

  This function gets Dma Channel Status.
- void Mcl_SetDmaChannelGlobalList (const uint32 Channel, const Mcl_DmaChannelGlobalListType List[], const uint32 ListDimension)

  This function sets Dma Channel Global List settings.

- void Mcl_SetDmaChannelTransferList (const uint32 Channel, const Mcl_DmaChannelTransferListType List[], const uint32 ListDimension)

  This function sets Dma Channel Transfer List settings.
- void Mcl_SetDmaChannelScatterGatherList (const uint32 Channel, const uint32 Element, const Mcl_DmaChannelScatterG List[], const uint32 ListDimension)

  This function sets Dma Channel Scatter/Gather List settings.
- void Mcl_GetDmaChannelParam (const uint32 Channel, const Mcl_DmaChannelInfoParamType Param, uint32 *const Value)

  This function gets the Dma Channel Parameter value.
- void Mcl_SetDmaChannelScatterGatherConfig (const uint32 Channel, const uint32 Element)

  This function configures the Dma Channel Scatter/Gather.
- void Mcl_SetTrgMuxInput (const uint32 Trigger, const uint32 Input)

  This function sets the Trgmux Trigger Input selection.
- void Mcl_SetTrgMuxLock (const uint32 Trigger)

  This function sets the Trgmux Trigger Lock.
- void Mcl_CacheEnable (Mcl_CacheType CacheType, Mcl_CacheBusType BusType)

  This function enables the Cache.
- void Mcl_CacheDisable (Mcl_CacheType CacheType, Mcl_CacheBusType BusType)

  This function disables the Cache.
- void Mcl_CacheInvalidate (Mcl_CacheType CacheType, Mcl_CacheBusType BusType)

  This function Invalidates the Cache.
- void Mcl_CacheClean (Mcl_CacheType CacheType, Mcl_CacheBusType BusType, boolean EnInvalidate)

  This function Cleans the Cache.
- void Mcl_CacheInvalidateByAddr (Mcl_CacheType CacheType, Mcl_CacheBusType BusType, uint32 Addr, uint32 Length)

  This function Invalidates the Cache by address.
- void Mcl_CacheCleanByAddr (Mcl_CacheType CacheType, Mcl_CacheBusType BusType, boolean En↩ Invalidate, uint32 Addr, uint32 Length)

  This function Cleans the Cache by address.
- void Mcl_Emios_SetReloadInterval (uint16 logicChannel, uint8 interval)

  Allow the user to specify the number of bus reload events skipped. Reload Signal Output Delay Interval.
- void Mcl_Emios_SetCounterBusPeriod (uint16 logicChannel, uint32 period, boolean syncUpdate)

  Change the period on active/intialized EMIOS counter(master) bus.
- void Mcl_EmiosConfigureGlobalTimebase (uint8 Instance, uint8 Value)

  eMios Global Timebase Enable.
- void Mcl_SetLcuWriteProtect (const uint8 LogicInstance)

  [WP]This function Enable Write Protect feature for the Logic Instance.
- void Mcl_SetLcuSyncInputSwOverrideEnable (const Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [SWEN] This function Enable/Disable software override of LC input.
- void Mcl_SetLcuSyncInputSwOverrideValue (const Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [SWVALUE] This function specifies the software override value for each LC input.
- void Mcl_SetLcuSyncInputMuxSelect (const Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [MUXSEL] This function selects the source of the LC input.
- void Mcl_SetLcuSyncInputSwSyncMode (const Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [SW_MODE] specifies the software sync mode for the inputs to this LC.

- void Mcl_SetLcuSyncOutputDebugMode (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [DBGEN] This function Enables/Disables outputs to continue operation in Debug mode.
- void Mcl_SetLcuSyncOutputEnable (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [OUTEN] This function Enables/Disables LC outputs.
- void Mcl_SetLcuSyncOutputForceInputSensitivity (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCE_SENSE] This function specifies which force inputs affect output.
- void Mcl_SetLcuSyncOutputForceClearingMode (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCE_MODE] This function specifies the timing for clearing force events for output.
- void Mcl_SetLcuSyncOutputForceSyncSelect (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [SYNC_SEL] This function specifies which sync input to use for Force signal.
- void Mcl_SetLcuSyncOutputPolarity (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [OUTPOL] This function specifies the polarity of the outputs.
- void Mcl_SetLcuSyncOutputForceDma (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCE_DMA_EN] This function Enables/Disable the generation of a DMA request when a force event occurs.
- void Mcl_SetLcuSyncOutputForceInt (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCE_INT_EN] This function Enables/Disable the generation of an interrupt request when force event occurs.
- void Mcl_SetLcuSyncOutputLutDma (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LUT_DMA_EN] This function Enables/Disable the generation of a DMA request when a LUT event occurs.
- void Mcl_SetLcuSyncOutputLutInt (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LUT_INT_EN] This function Enables/Disable the generation of an interrupt request when LUT event occurs.
- void Mcl_SetLcuSyncOutputFallFilter (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LUT_FALL_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low
- void Mcl_SetLcuSyncOutputRiseFilter (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LUT_RISE_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high
- void Mcl_SetLcuSyncOutputLutControl (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LUTCTRL] This function specifies the LUT positions, based on the combined LC input value
- void Mcl_ClearLcuSyncOutputForceEvent (const Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCESTS] This function Clear Force Event
- void Mcl_GetLcuSyncLogicInput (Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [LC_INPUTS] This function Indicates states of LC inputs
- void Mcl_GetLcuSyncSwOverrideInput (Mcl_LcuSyncInputValueType List[], const uint8 Dimension)

  [SWOUT] This function Indicates states of LC inputs or software-overridden inputs
- void Mcl_GetLcuSyncLogicOutput (Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [LCOUT] This function Indicates states of LC outputs
- void Mcl_GetLcuSyncForceOutput (Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCEOUT] This function Indicates the current state of force outputs for the logic outputs
- void Mcl_GetLcuSyncForceStatus (Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [FORCESTS] This function Indicates that a force event has occurred on the associated output
- void Mcl_GetLcuSyncCombineForceInput (Mcl_LcuSyncOutputValueType List[], const uint8 Dimension)

  [COMB_FORCE] This function Indicates the combined value of force inputs to each output

- void Mcl_SetLcuAsyncInputList (const uint8 LogicInput, const Mcl_LcuAsyncInputValueType List[], const uint8 Dimension)

  This function configure multiple configuration for one logic input.
- void Mcl_SetLcuAsyncOutputList (const uint8 LogicOutput, const Mcl_LcuAsyncOutputValueType List[], const uint8 Dimension)

  This function configure multiple configuration for one logic output.
- void Mcl_GetLcuAsyncLogicInputInfo (const uint8 LogicInput, const Mcl_LcuInputInfoParamType Param, uint8 *const Value)

  This function get information of Logic Input.
- void Mcl_GetLcuAsyncLogicOutputInfo (const uint8 LogicOutput, const Mcl_LcuOutputInfoParamType Param, uint8 *const Value)

  This function get information of Logic Output.
- void Mcl_GetVersionInfo (Std_VersionInfoType *const VersionInfo)

  Returns the version information of this module.

6.2.1.1 MISRA-C:2012 violations

6.2.2 Data Structure Documentation

6.2.2.1 struct Mcl_DmaChannelGlobalListType

This type contains the Mcl Dma Channel Global List.

The Mcl Dma Channel Global List contains a pair composed from Dma Channel Global Parameter Type and the Value of the parameter.

Definition at line 782 of file CDD_Mcl.h.

Data Fields

- Mcl_DmaChannelGlobalParamType Param

  The Mcl Dma Channel Global Parameter Type selects a parameter form the Global Parameter enum type.
- uint32 Value

  The Value stores the parameter's value.

6.2.2.1.1 Field Documentation

6.2.2.1.1.1 Param `Mcl_DmaChannelGlobalParamType Param`

The Mcl Dma Channel Global Parameter Type selects a parameter form the Global Parameter enum type.

Definition at line 783 of file CDD_Mcl.h.

6.2.2.1.1.2 Value uint32 Value

The Value stores the parameter's value.

Definition at line 784 of file CDD_Mcl.h.

6.2.2.2 struct Mcl_DmaChannelTransferListType

This type contains the Mcl Dma Channel Transfer List.

The Mcl Dma Channel Transfer List contains a pair composed from Dma Channel Transfer Parameter Type and the Value of the parameter.

Definition at line 794 of file CDD_Mcl.h.

Data Fields

- Mcl_DmaChannelTransferParamType Param
  The Mcl Dma Channel Transfer Parameter Type selects a parameter form the Transfer Parameter enum type.
- uint32 Value
  The Value stores the parameter's value.

6.2.2.2.1 Field Documentation

6.2.2.2.1.1 Param Mcl_DmaChannelTransferParamType Param

The Mcl Dma Channel Transfer Parameter Type selects a parameter form the Transfer Parameter enum type.

Definition at line 795 of file CDD_Mcl.h.

6.2.2.2.1.2 Value uint32 Value

The Value stores the parameter's value.

Definition at line 796 of file CDD_Mcl.h.

6.2.2.3 struct Mcl_DmaChannelScatterGatherListType

This type contains the Mcl Dma Channel Scatter/Gather List.

The Mcl Dma Channel Scatter/Gather List contains a pair composed from Dma Channel Scatter/Gather Parameter Type and the Value of the parameter.

Definition at line 806 of file CDD_Mcl.h.

Data Fields

- Mcl_DmaChannelTransferParamType Param
  The Mcl Dma Channel Transfer Parameter Type selects a parameter form the Transfer Parameter enum type.
- uint32 Value
  The Value stores the parameter's value.

6.2.2.3.1 Field Documentation

6.2.2.3.1.1 Param `Mcl_DmaChannelTransferParamType Param`

The Mcl Dma Channel Transfer Parameter Type selects a parameter form the Transfer Parameter enum type.

Definition at line 807 of file CDD_Mcl.h.

6.2.2.3.1.2 Value `uint32 Value`

The Value stores the parameter's value.

Definition at line 808 of file CDD_Mcl.h.

6.2.2.4 struct Mcl_DmaInstanceStatusType

This type contains the Mcl Dma Instance Status.

The Mcl Dma Instance Status contains the Hardware Errors, Active Id and Active indication for the running Dma Channel.

Definition at line 819 of file CDD_Mcl.h.

Data Fields

- uint32 Errors
  [VALUE] The Errors value is read from the DMA Instance Error Register (ES) as it is.
- uint8 ActiveId
  [VALUE] The ActiveId value is read from the DMA Instance Control Register (CR) field ACTIVE_ID.
- boolean Active
  [BOOLEAN] The Active value is read from the DMA Instance Control Register (CR) field ACTIVE.
- uint8 Version
  [VALUE] The Version value is read from the DMA Instance Control Register (CR) field VER.

6.2.2.4.1 Field Documentation

6.2.2.4.1.1 Errors `uint32 Errors`

[VALUE] The Errors value is read from the DMA Instance Error Register (ES) as it is.

Definition at line 820 of file CDD_Mcl.h.

6.2.2.4.1.2 ActiveId `uint8 ActiveId`

[VALUE] The ActiveId value is read from the DMA Instance Control Register (CR) field ACTIVE_ID.

Definition at line 821 of file CDD_Mcl.h.

6.2.2.4.1.3 Active `boolean Active`

[BOOLEAN] The Active value is read from the DMA Instance Control Register (CR) field ACTIVE.

Definition at line 822 of file CDD_Mcl.h.

6.2.2.4.1.4 Version `uint8 Version`

[VALUE] The Version value is read from the DMA Instance Control Register (CR) field VER.

Definition at line 823 of file CDD_Mcl.h.

6.2.2.5 struct Mcl_DmaChannelStatusType

This type contains the Mcl Dma Channel Status.

The Mcl Dma Channel Status contains the Hardware Errors, Active status and Done indication for the running Dma Channel.

Definition at line 833 of file CDD_Mcl.h.

Data Fields

- Mcl_DmaChannelStateType ChannelState

  [VALUE] The ChStateValue value is read from the internal DMA Driver Channel State Machine. Check UM for additional information.
- uint32 Errors

  [VALUE] The Errors value is read from the DMA Channel Error Register (CHx_ES) as it is.
- boolean Active

  [BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field ACTIVE.
- boolean Done

  [BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field DONE.

6.2.2.5.1 Field Documentation

6.2.2.5.1.1 ChannelState `Mcl_DmaChannelStateType ChannelState`

[VALUE] The ChStateValue value is read from the internal DMA Driver Channel State Machine. Check UM for additional information.

Definition at line 834 of file CDD_Mcl.h.

6.2.2.5.1.2 Errors `uint32 Errors`

[VALUE] The Errors value is read from the DMA Channel Error Register (CHx_ES) as it is.

Definition at line 835 of file CDD_Mcl.h.

6.2.2.5.1.3 Active `boolean Active`

[BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field ACTIVE.

Definition at line 836 of file CDD_Mcl.h.

6.2.2.5.1.4 Done `boolean Done`

[BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field DONE.

Definition at line 837 of file CDD_Mcl.h.

6.2.2.6 struct Mcl_ConfigType

This type contains the Mcl Configuration.

The Mcl Configuration structure contains pointers to the Ip's configuration structure. Based on the available support, specific configurations shall be stored.

Definition at line 289 of file Mcl_Types.h.

6.2.3 Macro Definition Documentation

6.2.3.1 MCL_DET_INIT

```
#define MCL_DET_INIT
```

API service ID for Mcl_Init function.

Parameters used when raising an error/exception

Definition at line 127 of file CDD_Mcl.h.

6.2.3.2 MCL_DET_DMA_INSTANCE_COMMAND

```
#define MCL_DET_DMA_INSTANCE_COMMAND
```

API service ID for Mcl_SetDmaInstanceCommand function.

Parameters used when raising an error/exception

Definition at line 135 of file CDD_Mcl.h.

6.2.3.3 MCL_DET_DMA_INSTANCE_STATUS

```
#define MCL_DET_DMA_INSTANCE_STATUS
```

API service ID for Mcl_GetDmaInstanceStatus function.

Parameters used when raising an error/exception

Definition at line 141 of file CDD_Mcl.h.

6.2.3.4 MCL_DET_DMA_CHANNEL_COMMAND

```
#define MCL_DET_DMA_CHANNEL_COMMAND
```

API service ID for Mcl_SetDmaChannelCommand function.

Parameters used when raising an error/exception

Definition at line 147 of file CDD_Mcl.h.

6.2.3.5 MCL_DET_DMA_CHANNEL_STATUS

```
#define MCL_DET_DMA_CHANNEL_STATUS
```

API service ID for Mcl_GetDmaChannelStatus function.

Parameters used when raising an error/exception

Definition at line 153 of file CDD_Mcl.h.

6.2.3.6 MCL_DET_DMA_GLOBAL

```
#define MCL_DET_DMA_GLOBAL
```

API service ID for Mcl_SetDmaChannelGlobalList function.

Parameters used when raising an error/exception

Definition at line 159 of file CDD_Mcl.h.

6.2.3.7 MCL_DET_DMA_TRANSFER

```
#define MCL_DET_DMA_TRANSFER
```

API service ID for Mcl_SetDmaChannelTransferList function.

Parameters used when raising an error/exception

Definition at line 165 of file CDD_Mcl.h.

6.2.3.8 MCL_DET_DMA_SCATTER_GATHER_LIST

```
#define MCL_DET_DMA_SCATTER_GATHER_LIST
```

API service ID for Mcl_SetDmaChannelScatterGatherList function.

Parameters used when raising an error/exception

Definition at line 171 of file CDD_Mcl.h.

6.2.3.9 MCL_DET_DMA_INFORMATION

```
#define MCL_DET_DMA_INFORMATION
```

API service ID for Mcl_GetDmaChannelParam function.

Parameters used when raising an error/exception

Definition at line 177 of file CDD_Mcl.h.

6.2.3.10 MCL_DET_DMA_SCATTER_GATHER_CONFIG

```
#define MCL_DET_DMA_SCATTER_GATHER_CONFIG
```

API service ID for Mcl_SetDmaChannelScatterGatherConfig function.

Parameters used when raising an error/exception

Definition at line 183 of file CDD_Mcl.h.

6.2.3.11 MCL_DET_DEINIT

```
#define MCL_DET_DEINIT
```

API service ID for Mcl_DeInit function.

Parameters used when raising an error/exception

Definition at line 193 of file CDD_Mcl.h.

6.2.3.12 MCL_DET_CACHE_ENABLE

```
#define MCL_DET_CACHE_ENABLE
```

API service ID for Mcl_CacheEnable function.

Parameters used when raising an error/exception

Definition at line 200 of file CDD_Mcl.h.

6.2.3.13 MCL_DET_CACHE_DISABLE

```
#define MCL_DET_CACHE_DISABLE
```

API service ID for Mcl_CacheDisable function.

Parameters used when raising an error/exception

Definition at line 206 of file CDD_Mcl.h.

6.2.3.14 MCL_DET_CACHE_INVALIDATE

```
#define MCL_DET_CACHE_INVALIDATE
```

API service ID for Mcl_CacheInvalidate function.

Parameters used when raising an error/exception

Definition at line 212 of file CDD_Mcl.h.

6.2.3.15 MCL_DET_CACHE_CLEAN

```
#define MCL_DET_CACHE_CLEAN
```

API service ID for Mcl_CacheClean function.

Parameters used when raising an error/exception

Definition at line 218 of file CDD_Mcl.h.

6.2.3.16 MCL_DET_CACHE_INVALIDATE_BY_ADDRESS

```
#define MCL_DET_CACHE_INVALIDATE_BY_ADDRESS
```

API service ID for Mcl_CacheInvalidateByAddr function.

Parameters used when raising an error/exception

Definition at line 224 of file CDD_Mcl.h.

6.2.3.17 MCL_DET_CACHE_CLEAN_BY_ADDRESS

```
#define MCL_DET_CACHE_CLEAN_BY_ADDRESS
```

API service ID for Mcl_CacheCleanByAddr function.

Parameters used when raising an error/exception

Definition at line 230 of file CDD_Mcl.h.

6.2.3.18 MCL_DET_TRGMUX_INPUT

```
#define MCL_DET_TRGMUX_INPUT
```

API service ID for Mcl_SetTrgMuxInput function.

Parameters used when raising an error/exception

Definition at line 240 of file CDD_Mcl.h.

6.2.3.19 MCL_DET_TRGMUX_LOCK

```
#define MCL_DET_TRGMUX_LOCK
```

API service ID for Mcl_SetTrgMuxLock function.

Parameters used when raising an error/exception

Definition at line 246 of file CDD_Mcl.h.

6.2.3.20 MCL_DET_LCU_ASYNC_SET_INPUT

```
#define MCL_DET_LCU_ASYNC_SET_INPUT
```

API service ID for Mcl_SetLcuAsyncInputList function.

Parameters used when raising an error/exception

Definition at line 258 of file CDD_Mcl.h.

6.2.3.21 MCL_DET_LCU_ASYNC_SET_OUTPUT

```
#define MCL_DET_LCU_ASYNC_SET_OUTPUT
```

API service ID for Mcl_SetLcuAsyncOutputList function.

Parameters used when raising an error/exception

Definition at line 264 of file CDD_Mcl.h.

6.2.3.22 MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_EN

```
#define MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_EN
```

API service ID for Mcl_SetLcuSyncInputSwOverrideEnable function.

Parameters used when raising an error/exception

Definition at line 274 of file CDD_Mcl.h.

6.2.3.23 MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_VALUE

```
#define MCL_DET_LCU_SYNC_SET_INPUT_SW_OVERRIDE_VALUE
```

API service ID for Mcl_SetLcuSyncInputSwOverrideValue function.

Parameters used when raising an error/exception

Definition at line 280 of file CDD_Mcl.h.

6.2.3.24 MCL_DET_LCU_SYNC_SET_INPUT_MUX_SEL

```
#define MCL_DET_LCU_SYNC_SET_INPUT_MUX_SEL
```

API service ID for Mcl_SetLcuSyncInputMuxSelect function.

Parameters used when raising an error/exception

Definition at line 286 of file CDD_Mcl.h.

6.2.3.25 MCL_DET_LCU_SYNC_SET_INPUT_SW_SYNC_MODE

```
#define MCL_DET_LCU_SYNC_SET_INPUT_SW_SYNC_MODE
```

API service ID for Mcl_SetLcuSyncInputSwSyncMode function.

Parameters used when raising an error/exception

Definition at line 292 of file CDD_Mcl.h.

6.2.3.26 MCL_DET_LCU_SYNC_SET_OUTPUT_DEBUG_MODE_EN

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_DEBUG_MODE_EN
```

API service ID for Mcl_SetLcuSyncOutputDebugMode function.

Parameters used when raising an error/exception

Definition at line 298 of file CDD_Mcl.h.

6.2.3.27 MCL_DET_LCU_SYNC_SET_OUTPUT_ENABLE

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_ENABLE
```

API service ID for Mcl_SetLcuSyncOutputEnable function.

Parameters used when raising an error/exception

Definition at line 304 of file CDD_Mcl.h.

6.2.3.28 MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SENSITIVITY

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SENSITIVITY
```

API service ID for Mcl_SetLcuSyncOutputForceInputSensitivity function.

Parameters used when raising an error/exception

Definition at line 310 of file CDD_Mcl.h.

6.2.3.29 MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_CLEAR_MODE

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_CLEAR_MODE
```

API service ID for Mcl_SetLcuSyncOutputForceClearingMode function.

Parameters used when raising an error/exception

Definition at line 316 of file CDD_Mcl.h.

6.2.3.30 MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SYNC_SEL

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_SYNC_SEL
```

API service ID for Mcl_SetLcuSyncOutputForceSyncSelect function.

Parameters used when raising an error/exception

Definition at line 322 of file CDD_Mcl.h.

6.2.3.31 MCL_DET_LCU_SYNC_SET_OUTPUT_POLARITY

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_POLARITY
```

API service ID for Mcl_SetLcuSyncOutputPolarity function.

Parameters used when raising an error/exception

Definition at line 328 of file CDD_Mcl.h.

6.2.3.32 MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_DMA

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_DMA
```

API service ID for Mcl_SetLcuSyncOutputForceDma function.

Parameters used when raising an error/exception

Definition at line 334 of file CDD_Mcl.h.

6.2.3.33 MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_INT

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FORCE_INT
```

API service ID for Mcl_SetLcuSyncOutputForceInt function.

Parameters used when raising an error/exception

Definition at line 340 of file CDD_Mcl.h.

6.2.3.34 MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_DMA

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_DMA
```

API service ID for Mcl_SetLcuSyncOutputLutDma function.

Parameters used when raising an error/exception

Definition at line 346 of file CDD_Mcl.h.

6.2.3.35 MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_INT

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_INT
```

API service ID for Mcl_SetLcuSyncOutputLutInt function.

Parameters used when raising an error/exception

Definition at line 352 of file CDD_Mcl.h.

6.2.3.36 MCL_DET_LCU_SYNC_SET_OUTPUT_FALL_FILTER

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_FALL_FILTER
```

API service ID for Mcl_SetLcuSyncOutputFallFilter function.

Parameters used when raising an error/exception

Definition at line 358 of file CDD_Mcl.h.

6.2.3.37 MCL_DET_LCU_SYNC_SET_OUTPUT_RISE_FILTER

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_RISE_FILTER
```

API service ID for Mcl_SetLcuSyncOutputRiseFilter function.

Parameters used when raising an error/exception

Definition at line 364 of file CDD_Mcl.h.

6.2.3.38 MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_CONTROL

```
#define MCL_DET_LCU_SYNC_SET_OUTPUT_LUT_CONTROL
```

API service ID for Mcl_SetLcuSyncOutputLutControl function.

Parameters used when raising an error/exception

Definition at line 370 of file CDD_Mcl.h.

6.2.3.39 MCL_DET_LCU_SYNC_CLEAR_OUTPUT_FORCE_EVENT

```
#define MCL_DET_LCU_SYNC_CLEAR_OUTPUT_FORCE_EVENT
```

API service ID for Mcl_ClearLcuSyncOutputForceEvent function.

Parameters used when raising an error/exception

Definition at line 376 of file CDD_Mcl.h.

6.2.3.40 MCL_DET_LCU_SYNC_GET_LOGIC_INPUT

```
#define MCL_DET_LCU_SYNC_GET_LOGIC_INPUT
```

API service ID for Mcl_GetLcuSyncLogicInput function.

Parameters used when raising an error/exception

Definition at line 382 of file CDD_Mcl.h.

6.2.3.41 MCL_DET_LCU_SYNC_GET_SW_OVERRIDE_INPUT

```
#define MCL_DET_LCU_SYNC_GET_SW_OVERRIDE_INPUT
```

API service ID for Mcl_GetLcuSyncSwOverrideInput function.

Parameters used when raising an error/exception

Definition at line 388 of file CDD_Mcl.h.

6.2.3.42 MCL_DET_LCU_SYNC_GET_LOGIC_OUTPUT

```
#define MCL_DET_LCU_SYNC_GET_LOGIC_OUTPUT
```

API service ID for Mcl_GetLcuSyncLogicOutput function.

Parameters used when raising an error/exception

Definition at line 394 of file CDD_Mcl.h.

6.2.3.43 MCL_DET_LCU_SYNC_GET_FORCE_OUTPUT

```
#define MCL_DET_LCU_SYNC_GET_FORCE_OUTPUT
```

API service ID for Mcl_GetLcuSyncForceOutput function.

Parameters used when raising an error/exception

Definition at line 400 of file CDD_Mcl.h.

6.2.3.44 MCL_DET_LCU_SYNC_GET_FORCE_STATUS

```
#define MCL_DET_LCU_SYNC_GET_FORCE_STATUS
```

API service ID for Mcl_GetLcuSyncForceStatus function.

Parameters used when raising an error/exception

Definition at line 406 of file CDD_Mcl.h.

6.2.3.45 MCL_DET_LCU_SYNC_GET_COMBINE_FORCE_INPUT

```
#define MCL_DET_LCU_SYNC_GET_COMBINE_FORCE_INPUT
```

API service ID for Mcl_GetLcuSyncCombineForceInput function.

Parameters used when raising an error/exception

Definition at line 412 of file CDD_Mcl.h.

6.2.3.46 MCL_DET_LCU_SET_WRITE_PROTECT

```
#define MCL_DET_LCU_SET_WRITE_PROTECT
```

API service ID for Mcl_GetLcuSyncCombineForceInput function.

Parameters used when raising an error/exception

Definition at line 420 of file CDD_Mcl.h.

6.2.3.47 MCL_DET_LCU_ASYNC_GET_INPUT_INFO

```
#define MCL_DET_LCU_ASYNC_GET_INPUT_INFO
```

API service ID for Mcl_SetLcuAsyncInputList function.

Parameters used when raising an error/exception

Definition at line 427 of file CDD_Mcl.h.

6.2.3.48 MCL_DET_LCU_ASYNC_GET_OUTPUT_INFO

```
#define MCL_DET_LCU_ASYNC_GET_OUTPUT_INFO
```

API service ID for Mcl_SetLcuAsyncOutputList function.

Parameters used when raising an error/exception

Definition at line 432 of file CDD_Mcl.h.

6.2.3.49 MCL_DET_EMIOS_E_INVALID_CHANNEL

```
#define MCL_DET_EMIOS_E_INVALID_CHANNEL
```

All API's called with wrong logic channel shall return this error.

Definition at line 442 of file CDD_Mcl.h.

6.2.3.50 MCL_DET_EMIOS_E_INVALID_SET

```
#define MCL_DET_EMIOS_E_INVALID_SET
```

All API's called with wrong logic channel shall return this error.

Definition at line 445 of file CDD_Mcl.h.

6.2.3.51 MCL_DET_TRGMUX_SET_INPUT

```
#define MCL_DET_TRGMUX_SET_INPUT
```

API service ID for Mcl_SetTrgMuxInput function.

Parameters used when raising an error/exception

Definition at line 455 of file CDD_Mcl.h.

6.2.3.52 MCL_DET_TRGMUX_SET_LOC

```
#define MCL_DET_TRGMUX_SET_LOC
```

API service ID for Mcl_SetTrgMuxLock function.

Parameters used when raising an error/exception

Definition at line 461 of file CDD_Mcl.h.

6.2.3.53 MCL_E_UNINIT

```
#define MCL_E_UNINIT
```

All API's having pointers as parameters shall return this error if called with with a NULL value.

Definition at line 471 of file CDD_Mcl.h.

6.2.3.54 MCL_E_PARAM_POINTER

```
#define MCL_E_PARAM_POINTER
```

All API's having pointers as parameters shall return this error if called with with a NULL value.

Definition at line 478 of file CDD_Mcl.h.

6.2.3.55 MCL_E_INVALID_INSTANCE

```
#define MCL_E_INVALID_INSTANCE
```

All API's called with wrong instance shall return this error.

Definition at line 484 of file CDD_Mcl.h.

6.2.3.56 MCL_E_INVALID_CHANNEL

```
#define MCL_E_INVALID_CHANNEL
```

All API's called with wrong channel shall return this error.

Definition at line 490 of file CDD_Mcl.h.

6.2.3.57 MCL_E_INVALID_COMMAND

```
#define MCL_E_INVALID_COMMAND
```

All API's called with wrong instance shall return this error.

Definition at line 496 of file CDD_Mcl.h.

6.2.3.58 MCL_E_INVALID_PARAMETER

```
#define MCL_E_INVALID_PARAMETER
```

All API's called with wrong read parameter shall return this error.

Definition at line 502 of file CDD_Mcl.h.

6.2.3.59 MCL_E_INVALID_STATE

```
#define MCL_E_INVALID_STATE
```

All API's called in wrong sequence shall return this error.

Definition at line 508 of file CDD_Mcl.h.

6.2.3.60 MCL_E_INCONSISTENCY

```
#define MCL_E_INCONSISTENCY
```

All API's called while hardware has error status shall return this error.

Definition at line 514 of file CDD_Mcl.h.

6.2.3.61 MCL_E_TIMEOUT

```
#define MCL_E_TIMEOUT
```

All API's called with a timeout value shall return this error if execution is not finished in the allocated timeframe.

Definition at line 521 of file CDD_Mcl.h.

6.2.3.62 MCL_E_PARAM_CONFIG

```
#define MCL_E_PARAM_CONFIG
```

If DET error reporting is enabled, the MCL will check upon each API call if the requested resource is configured to be available on the current core, and in case of error will return MCL_E_PARAM_CONFIG.

Definition at line 529 of file CDD_Mcl.h.

6.2.3.63 MCL_E_PROTECTED

```
#define MCL_E_PROTECTED
```

If DET error reporting is enabled, the MCL will check if registers are protected.

Definition at line 534 of file CDD_Mcl.h.

6.2.3.64 MCL_E_INIT_FAILED

```
#define MCL_E_INIT_FAILED
```

If VariantPreCompile is used, the configuration pointer shall have a NULL_PTR value. If VariantPostBuild is used, the configuration pointer shall be different from NULL_PTR. And in case of violate will return MCL_E_INIT_↩ FAILED.

Definition at line 541 of file CDD_Mcl.h.

6.2.4 Types Reference

6.2.4.1 Mcl_LcuSyncInputValueType

```
typedef Lcu_Ip_SyncInputValueType Mcl_LcuSyncInputValueType
```

This type contains the Mcl Lcu Multiple Inputs and Multiple Value.

The Mcl Lcu Multi Input Value contains a pair composed from Logic Input ID and the Value of configuration. The LogicInputId selects a Logic Input The Value stores the configuration's value.

Definition at line 850 of file CDD_Mcl.h.

6.2.4.2 Mcl_LcuSyncOutputValueType

```
typedef Lcu_Ip_SyncOutputValueType Mcl_LcuSyncOutputValueType
```

This type contains the Mcl Lcu Multiple Outputs and Multiple Value.

The Mcl Lcu Multi Output Value contains a pair composed from Logic Output ID and the Value of configuration. The LogicOutputId selects a Logic Output The Value stores the configuration's value.

Definition at line 860 of file CDD_Mcl.h.

6.2.4.3 Mcl_LcuAsyncInputValueType

```
typedef Lcu_Ip_AsyncInputValueType Mcl_LcuAsyncInputValueType
```

This type contains the Mcl Lcu Input with multiple value.

The Mcl Lcu Input with Multiple Value contains a pair composed from Input Parameter and Value of the Parameter. The Mcl Lcu Input Param Type selects a parameter from the Lcu_Ip_InputParamType enum The Value stores the configuration's value.

Definition at line 870 of file CDD_Mcl.h.

6.2.4.4 Mcl_LcuAsyncOutputValueType

```
typedef Lcu_Ip_AsyncOutputValueType Mcl_LcuAsyncOutputValueType
```

This type contains the Mcl Lcu Output with multiple value.

The Mcl Lcu Output with Multiple Value contains a pair composed from Output Parameter and Value of the Parameter. The Mcl Lcu Output Param Type selects a parameter from the Lcu_Ip_OutputParamType enum The Value stores the configuration's value.

Definition at line 880 of file CDD_Mcl.h.

6.2.5 Enum Reference

6.2.5.1 Mcl_DmaInstanceCmdType

```
enum Mcl_DmaInstanceCmdType
```

This type contains the Mcl Dma Instance Commands.

The Commands trigger specific actions in the Dma Instance.

Enumerator

MCL_DMA_INST_STOP	The Stop Command stops the executing channel and forces the Minor Loop to finish.
MCL_DMA_INST_STOP_ERROR	The StopError Command stops the executing channel, forces the Minor Loop to finish and generates an error interrupt.
MCL_DMA_INST_PAUSE	The Pause Command allows the ongoing transfer to finish and pauses any new transfer.
MCL_DMA_INST_RESUME	The Resume Command allows the transfer to continue.

Definition at line 553 of file CDD_Mcl.h.

6.2.5.2 Mcl_DmaChannelCmdType

enum Mcl_DmaChannelCmdType

This type contains the Mcl Dma Channel Commands.

The Commands trigger specific actions in the Dma Channel.

Enumerator

MCL_DMA_CH_START_REQUEST	The Start Request Command enables the Dma Channel to be triggered by hardware requests.
MCL_DMA_CH_STOP_REQUEST	The Stop Request Command disables the Dma Channel to be triggered by hardware requests.
MCL_DMA_CH_START_SERVICE	The Start Service Command sends a start request to the Dma Channel.
MCL_DMA_CH_ACK_DONE	The Ack Done Command resets the Dma Channel Done status.
MCL_DMA_CH_ACK_ERROR	The Ack Error Command resets the Dma Channel Error status.

Definition at line 566 of file CDD_Mcl.h.

6.2.5.3 Mcl_DmaChannelGlobalParamType

enum Mcl_DmaChannelGlobalParamType

This type contains the Mcl Dma Channel Global Parameters.

The Parameters set specific functionalities.

Enumerator

MCL_DMA_CH_SET_EN_MASTER_ID_↩ REPLICATION	[BOOLEAN] The EnMasterIdReplication Parameter sets the Dma Channel to use the same protection level and system bus ID of the master programming the Dma Channel.
MCL_DMA_CH_SET_EN_BUFFERED_WRITES	[BOOLEAN] The EnBufferedWrites Parameter sets the Dma Channel writes to be bufferable.
MCL_DMA_CH_SET_EN_HARDWARE_REQ	[BOOLEAN] The EnRequest Parameter enables the Dma Channel Request.
MCL_DMA_CH_SET_EN_ERROR_INTERRUPT	[BOOLEAN] The EnError Parameter enables the Dma Channel Error Interrupt.

Enumerator

MCL_DMA_CH_SET_GROUP_PRIORITY	[VALUE] The Group Parameter sets the Dma Channel Group Priority.
MCL_DMA_CH_SET_LEVEL_PRIORITY	[VALUE] The Level Parameter sets the Dma Channel Level Priority.
MCL_DMA_CH_SET_EN_PREEMPTION_↩ PRIORITY	[BOOLEAN] The EnPreemption Parameter enables the Dma Channel Preemption.
MCL_DMA_CH_SET_DIS_PREEMPT_↩ PRIORITY	[BOOLEAN] The DisPreempt Parameter disables the Dma Channel Preempt.

Definition at line 580 of file CDD_Mcl.h.

6.2.5.4 Mcl_DmaChannelTransferParamType

enum Mcl_DmaChannelTransferParamType

This type contains the Mcl Dma Channel Transfer Parameters.

The Parameters set specific functionalities.

Enumerator

MCL_DMA_CH_SET_SOURCE_ADDRESS	[VALUE] The Source Address Parameter sets the Dma Channel source address value.
MCL_DMA_CH_SET_SOURCE_SIGNED_↩ OFFSET	[VALUE] The Source Signed Offset Parameter sets the Dma Channel source signed offset value.
MCL_DMA_CH_SET_SOURCE_SIGNED_↩ LAST_ADDR_ADJ	[VALUE] The Source Signed Last Address Adjustment Parameter sets the Dma Channel source signed last address adjustment.
MCL_DMA_CH_SET_SOURCE_TRANSFER_↩ SIZE	[VALUE] The Source Transfer Size Parameter sets the Dma Channel source transfer size.
MCL_DMA_CH_SET_SOURCE_MODULO	[VALUE] The Source Modulo Parameter sets the Dma Channel source modulo.
MCL_DMA_CH_SET_DESTINATION_ADDRESS	[VALUE] The Destination Address Parameter sets the Dma Channel destination address value.
MCL_DMA_CH_SET_DESTINATION_↩ SIGNED_OFFSET	[VALUE] The Destination Signed Offset Parameter sets the Dma Channel destination signed offset value.
MCL_DMA_CH_SET_DESTINATION_↩ SIGNED_LAST_ADDR_ADJ	[VALUE] The Destination Signed Last Address Adjustment Parameter sets the Dma Channel destination signed last address adjustment.
MCL_DMA_CH_SET_DESTINATION_↩ TRANSFER_SIZE	[VALUE] The Destination Transfer Size Parameter sets the Dma Channel destination transfer size.
MCL_DMA_CH_SET_DESTINATION_MODULO	[VALUE] The Destination Modulo Parameter sets the Dma Channel destination modulo.

Enumerator

MCL_DMA_CH_SET_MINORLOOP_EN_SRC_OFFSET	[BOOLEAN] The Minor Loop Enable Source Offset Parameter enables the Dma Channel minor loop source offset.
MCL_DMA_CH_SET_MINORLOOP_EN_DST_OFFSET	[BOOLEAN] The Minor Loop Enable Destination Offset Parameter enables the Dma Channel minor loop destination offset.
MCL_DMA_CH_SET_MINORLOOP_SIGNED_OFFSET	[VALUE] The Minor Loop Signed Offset Parameter sets the Dma Channel minor loop signed offset.
MCL_DMA_CH_SET_MINORLOOP_EN_LINK	[BOOLEAN] The Minor Loop Enable Link Parameter enables the Dma Channel minor loop logic channel linking.
MCL_DMA_CH_SET_MINORLOOP_LOGIC_LINK_CH	[VALUE] The Minor Loop Logic Channel Link Parameter sets the Dma Channel minor loop logic channel link.
MCL_DMA_CH_SET_MINORLOOP_SIZE	[VALUE] The Minor Loop Size Parameter sets the Dma Channel minor loop transfer size.
MCL_DMA_CH_SET_MAJORLOOP_EN_LINK	[BOOLEAN] The Major Loop Enable Link Parameter enables the Dma Channel major loop logic channel linking.
MCL_DMA_CH_SET_MAJORLOOP_LOGIC_LINK_CH	[VALUE] The Major Loop Logic Channel Link Parameter sets the Dma Channel major loop logic channel link.
MCL_DMA_CH_SET_MAJORLOOP_COUNT	[VALUE] The Major Loop Count Parameter sets the Dma Channel major loop count.
MCL_DMA_CH_SET_CONTROL_STORE_DST_ADDR	[VALUE] The Store Destination Address Parameter saves the final destination address in system memory.
MCL_DMA_CH_SET_CONTROL_SOFTWARE_REQUEST	[BOOLEAN] The Enable Start Parameter enables the Dma Channel start service request.
MCL_DMA_CH_SET_CONTROL_EN_MAJOR_INTERRUPT	[BOOLEAN] The Enable Major Interrupt Parameter enables the Dma Channel major interrupt.
MCL_DMA_CH_SET_CONTROL_EN_HALF_MAJOR_INTERRUPT	[BOOLEAN] The Enable Half Interrupt Parameter enables the Dma Channel half major interrupt.
MCL_DMA_CH_SET_CONTROL_DIS_AUTO_REQUEST	[BOOLEAN] The Disable Automatic Request Parameter disables the Dma Channel automatic request.
MCL_DMA_CH_SET_CONTROL_EN_END_OF_PACKET_SIGNAL	[BOOLEAN] The Enable End Of Packet Signal Parameter enables the Dma Channel end of packet signal.
MCL_DMA_CH_SET_CONTROL_BANDWIDTH	[VALUE] The Bandwidth Control Parameter sets the Dma Channel bandwidth control.

Definition at line 605 of file CDD_Mcl.h.

6.2.5.5 Mcl_DmaChannelStateType

```
enum Mcl_DmaChannelStateType
```

This type contains the Mcl Dma Channel State values.

The states represent the Channel status during runtime.

Definition at line 644 of file CDD_Mcl.h.

6.2.5.6 Mcl_DmaChannelInfoParamType

enum Mcl_DmaChannelInfoParamType

This type contains the Mcl Dma Channel Information Parameters.

The Parameters get specific information.

Enumerator

MCL_DMA_CH_GET_SOURCE_ADDRESS	[VALUE] The Source Address Parameter gets the Dma Channel source address.
MCL_DMA_CH_GET_DESTINATION_ADDRESS	[VALUE] The Destination Address Parameter gets the Dma Channel destination address.
MCL_DMA_CH_GET_BEGIN_ITER_COUNT	[VALUE] The Begin Iteration Count Parameter gets the Dma Channel begin iteration count.
MCL_DMA_CH_GET_CURRENT_ITER_↩ COUNT	[VALUE] The Current Iteration Count Parameter gets the Dma Channel current iteration count.
MCL_DMA_CH_GET_STORE_DST_ADDR	[VALUE] The Store Destination Address Parameter gets the Dma Channel stored destination address.
MCL_DMA_CH_GET_MASTER_ID	[VALUE] The Master Id Parameter gets the Dma Channel master id.
MCL_DMA_CH_GET_MAJOR_INTERRUPT	[BOOLEAN] The Major Interrupt Parameter gets the Dma Channel major interrupt.
MCL_DMA_CH_GET_HALF_MAJOR_↩ INTERRUPT	[BOOLEAN] The Half Major Interrupt Parameter gets the Dma Channel half major interrupt.

Definition at line 659 of file CDD_Mcl.h.

6.2.5.7 Mcl_CacheType

enum Mcl_CacheType

This type contains the Mcl Cache Type selection.

The Cache Types select specific cache memory types.

Enumerator

MCL_CACHE_LMEM	The Cache Lmem Parameter selects LMEM cache types.
MCL_CACHE_CORE	The Cache Core Parameter selects CORE cache types.

Definition at line 682 of file CDD_Mcl.h.

6.2.5.8 Mcl_CacheBusType

enum Mcl_CacheBusType

This type contains the Mcl Cache Bus Type selection.

The Cache Bus Types select Code and System caches and bus.

Enumerator

MCL_CACHE_PC_BUS	The Cache PC Bus selects Processor Code (PC) bus (used with Cache Lmem).
MCL_CACHE_PS_BUS	The Cache PS Bus selects Processor System (PS) bus (used with Cache Lmem).
MCL_CACHE_ALL_BUS	The Cache All Bus selects PC and PS bus (used with Cache Lmem).
MCL_CACHE_INSTRUCTION	The Cache Instruction Parameter selects instruction cache (used with Cache Core).
MCL_CACHE_DATA	The Cache Data Parameter selects data cache (used with Cache Core).

Definition at line 693 of file CDD_Mcl.h.

6.2.5.9 Mcl_LcuInputParamType

enum Mcl_LcuInputParamType

This type contains the LCU Input Param Type.

The Parameters set specific functionalities for Input

Enumerator

MCL_LCU_IP_INPUT_SET_MUX_SEL	[MUXSEL] Input MUX Select.

Enumerator

MCL_LCU_IP_INPUT_SET_SW_SYNC_MODE	[SW_MODE] Specifies the software sync mode for the inputs to this LC.When Software Override is enabled (SWEN),these bits control whether Software Override Value (SWVALUE) changes occur immediately or on the rising edge of the selected sync pulse
MCL_LCU_IP_INPUT_SET_SW_OVERRIDE_EN	[SWEN] Software override input enable
MCL_LCU_IP_INPUT_SET_SW_VALUE	[SWVALUE] Software override input value

Definition at line 710 of file CDD_Mcl.h.

6.2.5.10 Mcl_LcuOutputParamType

enum Mcl_LcuOutputParamType

This type contains the LCU Output Param Type.

The Parameters set specific functionalities for Output

Enumerator

MCL_LCU_IP_OUTPUT_SET_EN_DEBUG_MODE	[DBGEN] Enables outputs to continue operation in Debug mode
MCL_LCU_IP_OUTPUT_SET_OUTPUT_ENABLE	[OUTEN] Enables LC outputs
MCL_LCU_IP_OUTPUT_SET_LUT_CONTROL	[LUTCTRL] LUT control
MCL_LCU_IP_OUTPUT_SET_LUT_RISE_FILTER	[LUT_RISE_FILT] LUT Rise Filter
MCL_LCU_IP_OUTPUT_SET_LUT_FALL_FILTER	[LUT_FALL_FILT] LUT Fall Filter
MCL_LCU_IP_OUTPUT_SET_EN_FORCE_DMA	[LUT_DMA_EN] Enables the generation of a DMA request when an LUT event occurs
MCL_LCU_IP_OUTPUT_SET_EN_LUT_DMA	[FORCE_DMA_EN] Enables the generation of a DMA request when a force event occurs
MCL_LCU_IP_OUTPUT_SET_EN_FORCE_INT	[LUT_INT_EN] Enables the generation of an interrupt request when an LUT event
MCL_LCU_IP_OUTPUT_SET_EN_LUT_INT	[FORCE_INT_EN] Enables the generation of an interrupt request when a force event occurs
MCL_LCU_IP_OUTPUT_SET_INVERT_OUTPUT	[OUTPOL] Set Output Polarity: invert or not.
MCL_LCU_IP_OUTPUT_SET_FORCE_SIGNAL_SEL	[FORCE_SENSE] Select Force signal
MCL_LCU_IP_OUTPUT_SET_CLEAR_FORCE_MODE	[FORCE_MODE] Force Clearing Mode
MCL_LCU_IP_OUTPUT_SET_FORCE_SYNC_SEL	[SYNC_SEL] The Force Sync Select Parameter specifies which sync input to use for this output
MCL_LCU_IP_OUTPUT_CLEAR_FORCE_STS	[FORCESTS] Clear force event in STS

Definition at line 724 of file CDD_Mcl.h.

6.2.5.11 Mcl_LcuInputInfoParamType

enum Mcl_LcuInputInfoParamType

This type contains the LCU Param Type to get information for input.

The Parameters get specific functionalities for Input

Definition at line 748 of file CDD_Mcl.h.

6.2.5.12 Mcl_LcuOutputInfoParamType

enum Mcl_LcuOutputInfoParamType

This type contains the LCU Param Type to get information for output.

The Parameters get specific functionalities for Output

Definition at line 760 of file CDD_Mcl.h.

6.2.5.13 Mcl_ReturnType

enum Mcl_ReturnType

This type contains the Mcl Return Type.

The Return Type give information for the execution of interfaces.

Definition at line 259 of file Mcl_Types.h.

6.2.6 Function Reference

6.2.6.1 Mcl_Init()

```
void Mcl_Init (
            const Mcl_ConfigType *const ConfigPtr )
```

This function initializes the Mcl Driver.

This service is a non reentrant function that shall initialize the Mcl driver. The initialization is applied for the enabled IPs, configured statically.

Parameters

in	*ConfigPtr*	Pointer to the configuration structure.

Returns

void

6.2.6.2 Mcl_DeInit()

```
void Mcl_DeInit (
            void  )
```

This function deinitializes the Mcl Driver.

This service is a non reentrant function that shall deinitialize the Mcl driver. The deinitialization is applied for the enabled IPs, configured statically.

Returns

void

6.2.6.3 Mcl_SetDmaInstanceCommand()

```
void Mcl_SetDmaInstanceCommand (
            const uint32 Instance,
            const Mcl_DmaInstanceCmdType Command )
```

This function sets Dma Instance Command.

This service is a reentrant function that shall command the Dma Instance. The command shall trigger specific functionalities of the Dma Instance.

Parameters

in	*Instance*	Selection value of the Logic Instance.
in	*Command*	The command for the Logic Instance.

Returns

void

6.2.6.4 Mcl_GetDmaInstanceStatus()

```
void Mcl_GetDmaInstanceStatus (
            const uint32 Instance,
            Mcl_DmaInstanceStatusType *const Status )
```

This function gets Dma Instance Status.

This service is a reentrant function that shall get the Dma Instance status. The command shall read specific functionalities of the Dma Instance.

Parameters

in	*Instance*	Selection value of the Logic Instance.
out	*Status*	Pointer to the Dma Instance status.

Returns

void

6.2.6.5 Mcl_SetDmaChannelCommand()

```
void Mcl_SetDmaChannelCommand (
            const uint32 Channel,
            const Mcl_DmaChannelCmdType Command )
```

This function sets Dma Channel Command.

This service is a reentrant function that shall command the Dma Channel. The command shall trigger specific functionalities of the Dma Channel.

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*Command*	The command for the Logic Channel.

Returns

void

6.2.6.6 Mcl_GetDmaChannelStatus()

```
void Mcl_GetDmaChannelStatus (
            const uint32 Channel,
            Mcl_DmaChannelStatusType *const Status )
```

This function gets Dma Channel Status.

This service is a reentrant function that shall get the Dma Channel status. The command shall read specific functionalities of the Dma Channel.

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
out	*Status*	Pointer to the Dma Channel status.

Returns

void

6.2.6.7 Mcl_SetDmaChannelGlobalList()

```
void Mcl_SetDmaChannelGlobalList (
            const uint32 Channel,
            const Mcl_DmaChannelGlobalListType List[],
            const uint32 ListDimension )
```

This function sets Dma Channel Global List settings.

This service is a reentrant function that shall set the Dma Channel global parameters list. The list is composed of an array of Dma Channel global parameters settings. The settings list(array) is defined by the user needs: it contains the desired parameters to be configured, in any desired order.

How to use this interface:

1. Use the "Mcl_DmaChannelGlobalListType" to create a list(array) with the desired paramaters to configure (see parameters: "Mcl_DmaChannelGlobalParamType") The list can declared globally or locally: Global example: Mcl_DmaChannelGlobalListType global_Mcl_DmaChannelGlobalList0[NUMBER_OF_PARAMETERS] = {...}; Local example: Mcl_DmaChannelGlobalListType Mcl_DmaChannelGlobalList[NUMBER_OF_PARAMETERS]; Mcl_DmaChannelGlobalList[PARAMETER0].Param = MCL_DMA_CH_SET_EN_BUFFERED_WRITES; Mcl_DmaChannelGlobalList[PARAMETER0].Value = TRUE; Mcl_DmaChannelGlobalList[PARAMETER1].Param = ...; Mcl_DmaChannelGlobalList[PARAMETER1].Value = ...;
2. Call the "Mcl_SetDmaChannelGlobalList()" interface: Mcl_SetDmaChannelGlobalList(LOGIC_CHANNELx, Mcl_DmaChannelGlobalList, NUMBER_OF_PARAMETERS);

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*List*	Pointer to the Global List Array.
in	*ListDimension*	Number of entries in the List.

Returns

void

6.2.6.8 Mcl_SetDmaChannelTransferList()

```
void Mcl_SetDmaChannelTransferList (
            const uint32 Channel,
            const Mcl_DmaChannelTransferListType List[],
            const uint32 ListDimension )
```

This function sets Dma Channel Transfer List settings.

This service is a reentrant function that shall set the Dma Channel transfer parameters list. The list is composed of an array of Dma Channel transfer parameters settings. The settings array is defined by the user needs: it contains entries for each desired parameter to be configured, in any suitable order.

How to use this interface:

1. Use the "Mcl_DmaChannelTransferListType" to create a list(array) with the desired paramaters to configure (see parameters: "Mcl_DmaChannelTransferParamType") The list can declared globally or locally: Global example: Mcl_DmaChannelTransferListType global_Mcl_DmaChannelTransferList0[NUMBER_OF_↩ PARAMETERS] = {...}; Local example: Mcl_DmaChannelTransferListType Mcl_DmaChannelTransfer↩ List[NUMBER_OF_PARAMETERS]; Mcl_DmaChannelTransferList[PARAMETER0].Param = MCL_↩ DMA_CH_SET_SOURCE_ADDRESS; Mcl_DmaChannelTransferList[PARAMETER0].Value = &Source↩ Buffer; Mcl_DmaChannelTransferList[PARAMETER1].Param = MCL_DMA_CH_SET_DESTINATION↩ _ADDRESS; Mcl_DmaChannelTransferList[PARAMETER1].Value = &DestinationBuffer;
2. Call the "Mcl_SetDmaChannelTransferList()" interface: Mcl_SetDmaChannelTransferList(LOGIC_↩ CHANNELx, Mcl_DmaChannelTransferList, NUMBER_OF_PARAMETERS);

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*List*	Pointer to the Transfer List Array.
in	*ListDimension*	Number of entries in the List.

Returns

void

6.2.6.9 Mcl_SetDmaChannelScatterGatherList()

```
void Mcl_SetDmaChannelScatterGatherList (
            const uint32 Channel,
            const uint32 Element,
            const Mcl_DmaChannelScatterGatherListType List[],
            const uint32 ListDimension )
```

This function sets Dma Channel Scatter/Gather List settings.

This service is a reentrant function that shall set the Dma Channel scatter/gather parameters list. The Scatter/↩ Gather List configures Logic Elements belonging to the same Dma Logic Channel. The settings array is defined by the user needs: it contains entries for each desired parameter to be configured, in any suitable order.

How to use this interface:

1. Use the "Mcl_DmaChannelScatterGatherListType" to create a list(array) with the desired paramaters to configure (see parameters: "Mcl_DmaChannelTransferParamType") The list can declared globally or locally: Global example: Mcl_DmaChannelScatterGatherListType global_Mcl_DmaChannelScatterGather↩ List0[NUMBER_OF_PARAMETERS] = {...}; Local example: Mcl_DmaChannelScatterGatherListType Mcl_DmaChannelScatterGatherList[NUMBER_OF_PARAMETERS]; Mcl_DmaChannelScatterGather↩ List[PARAMETER0].Param = MCL_DMA_CH_SET_SOURCE_ADDRESS; Mcl_DmaChannelScatter↩ GatherList[PARAMETER0].Value = &SourceBuffer; Mcl_DmaChannelScatterGatherList[PARAMETER1].Param = MCL_DMA_CH_SET_DESTINATION_ADDRESS; Mcl_DmaChannelScatterGatherList[PARAMETER1].Value = &DestinationBuffer;
2. Call the "Mcl_SetDmaChannelScatterGatherList()" interface: Mcl_SetDmaChannelScatterGatherList(↩ LOGIC_CHANNELx, LOGIC_ELEMENTy, Mcl_DmaChannelScatterGatherList, NUMBER_OF_↩ PARAMETERS);

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*Element*	Specifies the Logic Element Id.
in	*List*	Pointer to the Scatter/Gather List Array.
in	*ListDimension*	Number of entries in the List.

Returns

void

6.2.6.10 Mcl_GetDmaChannelParam()

```
void Mcl_GetDmaChannelParam (
            const uint32 Channel,
            const Mcl_DmaChannelInfoParamType Param,
            uint32 *const Value )
```

This function gets the Dma Channel Parameter value.

This service is a reentrant function that shall get the Dma Channel parameters value.

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*Param*	Selection parameter.
out	*Value*	Pointer to the parameter value.

Returns

void

6.2.6.11 Mcl_SetDmaChannelScatterGatherConfig()

```
void Mcl_SetDmaChannelScatterGatherConfig (
            const uint32 Channel,
            const uint32 Element )
```

This function configures the Dma Channel Scatter/Gather.

This service is a reentrant function that shall configure the Dma Channel scatter/gather functionality. The Scatter/↩ Gather settings, for the specified Dma Logic Channel, are loaded into the Software TCDs. Each software TCD corresponds to a Logic Element. The specified Logic Element shall be loaded into the Dma Logic Channel's Hardware TCD. The Logic Elements (describing the Software TCDs) form a simple chained list, the "Element" function parameter representing the lists's head.

Parameters

in	*Channel*	Specifies the Logic Channel Tag defined by the user.
in	*Element*	Specifies the Logic Element Id representing the list's head.

Returns

void

6.2.6.12 Mcl_SetTrgMuxInput()

```
void Mcl_SetTrgMuxInput (
            const uint32 Trigger,
            const uint32 Input )
```

This function sets the Trgmux Trigger Input selection.

This service is a reentrant function that shall configure the Trgmux Trigger functionality.

Parameters

in	*Trigger*	Selection value of the Logic Trigger.
in	*Input*	Selection value for the Logic Trigger's Input.

Returns

void

6.2.6.13 Mcl_SetTrgMuxLock()

```
void Mcl_SetTrgMuxLock (
            const uint32 Trigger )
```

This function sets the Trgmux Trigger Lock.

This service is a reentrant function that shall configure the Trgmux Trigger Lock functionality.

Parameters

in	*Trigger*	Selection value of the Logic Trigger.

Returns

void

6.2.6.14 Mcl_CacheEnable()

```
void Mcl_CacheEnable (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType )
```

This function enables the Cache.

This service is a reentrant function that shall enable the Cache functionality.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.

Returns

void

6.2.6.15 Mcl_CacheDisable()

```
void Mcl_CacheDisable (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType )
```

This function disables the Cache.

This service is a reentrant function that shall disable the Cache functionality.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.

Returns

void

6.2.6.16 Mcl_CacheInvalidate()

```
void Mcl_CacheInvalidate (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType )
```

This function Invalidates the Cache.

This service is a reentrant function that shall Invalidate the Cache functionality. The Invalidation applies to the entire Cache.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.

Returns

void

6.2.6.17 Mcl_CacheClean()

```
void Mcl_CacheClean (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType,
            boolean EnInvalidate )
```

This function Cleans the Cache.

This service is a reentrant function that shall Clean the Cache functionality. The Clean applies to the entire Cache. By enabling the Invalidation, the function shall execute specific Cache Clean&Invalidate function.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.
in	*EnInvalidate*	Enable the Invalidation specific functionality.

Returns

void

6.2.6.18 Mcl_CacheInvalidateByAddr()

```
void Mcl_CacheInvalidateByAddr (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType,
            uint32 Addr,
            uint32 Length )
```

This function Invalidates the Cache by address.

This service is a reentrant function that shall Invalidate the Cache by address. The Invalidation applies to the area in Cache specified by the address and length. The buffer shall be aligned to the Cache Line size.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.
in	*Addr*	Address value of the buffer.
in	*Length*	Length value of the buffer.

Returns

void

6.2.6.19 Mcl_CacheCleanByAddr()

```
void Mcl_CacheCleanByAddr (
            Mcl_CacheType CacheType,
            Mcl_CacheBusType BusType,
            boolean EnInvalidate,
            uint32 Addr,
            uint32 Length )
```

This function Cleans the Cache by address.

This service is a reentrant function that shall Clean the Cache by address. The Clean applies to the area in Cache specified by the address and length. By enabling the Invalidation, the function shall execute specific Cache Clean&Invalidate function.

Parameters

in	*CacheType*	Selection value of the Cache Type.
in	*BusType*	Selection value of the Bus Type.
in	*EnInvalidate*	Enable the Invalidation specific functionality.
in	*Addr*	Address value of the buffer.
in	*Length*	Length value of the buffer.

Returns

void

6.2.6.20 Mcl_Emios_SetReloadInterval()

```
void Mcl_Emios_SetReloadInterval (
            uint16 logicChannel,
            uint8 interval )
```

Allow the user to specify the number of bus reload events skipped. Reload Signal Output Delay Interval.

Parameters

logicChannel	Should be used the generated define MCL_EMIOS_LOGIC_CH[number] • [number] can be indentied by (uint16)((0U << hwNumber) + hwChannel) from define A list with this generated defines can be found on Emios_Mcl_Ip_Cfg_Defines.h
interval	00000b - Every event 00001b - Every 2nd event 00010b - Every 3rd event . . . 11111b - Every 32nd event

6.2.6.21 Mcl_Emios_SetCounterBusPeriod()

```
void Mcl_Emios_SetCounterBusPeriod (
            uint16 logicChannel,
            uint32 period,
            boolean syncUpdate )
```

Change the period on active/intialized EMIOS counter(master) bus.

Parameters

logicChannel	Should be used the generated define MCL_EMIOS_LOGIC_CH[number] • [number] can be indentied by (uint16)((0U << hwNumber) + hwChannel) from define A list with this generated defines can be found on Emios_Mcl_Ip_Cfg_Defines.h
period	New period value.
syncUpdate	Stop the channel output if syncronus start is chose. FALSE - the function will update the period and the transfer compartor will be enable TRUE - the function will disable the comparator transfer and after that update the period After the function is called with syncUpdate=TRUE, it will be needed to call a function which will enable the transfer comparator.

Running the following code will NOT DISABLE AND after ENABLE compartor transfer. Mcl_Emios_↩ SetCounterBusPeriod(logicChannel, period, syncUpdate=TRUE); Mcl_Emios_SetCounterBusPeriod(logicChannel, period, syncUpdate=FALSE);

6.2.6.22 Mcl_EmiosConfigureGlobalTimebase()

```
void Mcl_EmiosConfigureGlobalTimebase (
            uint8 Instance,
            uint8 Value )
```

eMios Global Timebase Enable.

This function enables or disables the GTBE bit of the EMIOSMCR register of an addressed eMIOS instance.

Parameters

in	*Instance*	Instance of EMIOS used
in	*Value*	STD_ON --> Global Timebase Enabled STD_OFF --> Global Timebase Disabled

Returns

void

6.2.6.23 Mcl_SetLcuWriteProtect()

```
void Mcl_SetLcuWriteProtect (
            const uint8 LogicInstance )
```

[WP]This function Enable Write Protect feature for the Logic Instance.

This service is a reentrant function that shall turns on write protection for all LCU registers except SWVALUE, LCn_STS, and FORCEST.

Parameters

in	*LogicInstance*	Specifies the Logic Instance.Using define generated by the configurator (Example : LCU_LOGIC_INSTANCE_0)

Returns

void

6.2.6.24 Mcl_SetLcuSyncInputSwOverrideEnable()

```
void Mcl_SetLcuSyncInputSwOverrideEnable (
            const Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[SWEN] This function Enable/Disable software override of LC input.

This service is a reentrant function that shall Enable/Disable software override of LC input. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_OVERRIDE_DISABLE/LCU_IP_SW_OVERRIDE↩ _ENABLE

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.25 Mcl_SetLcuSyncInputSwOverrideValue()

```
void Mcl_SetLcuSyncInputSwOverrideValue (
            const Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[SWVALUE] This function specifies the software override value for each LC input.

This service is a reentrant function that shall specifies the software override value for each LC input. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_OVERRIDE_LOGIC_LOW/LCU_IP_SW_↩ OVERRIDE_LOGIC_HIGH

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.26 Mcl_SetLcuSyncInputMuxSelect()

```
void Mcl_SetLcuSyncInputMuxSelect (
            const Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[MUXSEL] This function selects the source of the LC input.

This service is a reentrant function that shall selects the source of the LC input. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 255]

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.27 Mcl_SetLcuSyncInputSwSyncMode()

```
void Mcl_SetLcuSyncInputSwSyncMode (
            const Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[SW_MODE] specifies the software sync mode for the inputs to this LC.

This service is a reentrant function that shall specifies the software sync mode for the inputs to this LC.When Software Override is enabled (SWEN), these bits control whether Software Override Value (SWVALUE) changes occur immediately or on the rising edge of the selected sync pulse. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_SYNC_IMMEDIATE/LCU_IP_SW_SYNC_ON_↩ RISING_EDGE

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.28 Mcl_SetLcuSyncOutputDebugMode()

```
void Mcl_SetLcuSyncOutputDebugMode (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[DBGEN] This function Enables/Disables outputs to continue operation in Debug mode.

This service is a reentrant function that shall Enables/Disables outputs to continue operation in Debug mode. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DEBUG_DISABLE/LCU_IP_DEBUG_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.29 Mcl_SetLcuSyncOutputEnable()

```
void Mcl_SetLcuSyncOutputEnable (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[OUTEN] This function Enables/Disables LC outputs.

This service is a reentrant function that shall Enables/Disables LC outputs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_OUTPUT_DISABLE/LCU_IP_OUTPUT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.30 Mcl_SetLcuSyncOutputForceInputSensitivity()

```
void Mcl_SetLcuSyncOutputForceInputSensitivity (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_SENSE] This function specifies which force inputs affect output.

This service is a reentrant function that shall specifies which force inputs affect output. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 255] For each bit: 0b - Does not affect 1b - Affects .* Example: 011b specifies: force inputs 0 and 1 will affect for this logic output, .* but force inputs 2 will not affect

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.31 Mcl_SetLcuSyncOutputForceClearingMode()

```
void Mcl_SetLcuSyncOutputForceClearingMode (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_MODE] This function specifies the timing for clearing force events for output.

This service is a reentrant function that shall specifies the timing for clearing force events for output. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_CLEAR_FORCE_SIGNAL_IMMEDIATE/ LCU_IP↩_CLEAR_FORCE_SIGNAL_ON_RISING_EDGE/ LCU_IP_CLEAR_FORCE_SIGNAL_AFTER_CLEAR↩_STATUS/ LCU_IP_CLEAR_FORCE_SIGNAL_ON_RISING_EDGE_AFTER_CLEAR_STATUS

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.32 Mcl_SetLcuSyncOutputForceSyncSelect()

```
void Mcl_SetLcuSyncOutputForceSyncSelect (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[SYNC_SEL] This function specifies which sync input to use for Force signal.

This service is a reentrant function that shall specifies which sync input to use for Force signal. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 255] 00b - Sync input 0 01b - Sync input 1

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.33 Mcl_SetLcuSyncOutputPolarity()

```
void Mcl_SetLcuSyncOutputPolarity (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[OUTPOL] This function specifies the polarity of the outputs.

This service is a reentrant function that shall specifies the polarity of the outputs When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_FORCE_POL_NOT_INVERTED/LCU_IP_FORCE↩ _POL_INVERTED

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.34 Mcl_SetLcuSyncOutputForceDma()

```
void Mcl_SetLcuSyncOutputForceDma (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_DMA_EN] This function Enables/Disable the generation of a DMA request when a force event occurs.

This service is a reentrant function that shall Enables/Disable the generation of a DMA request when a force event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DMA_DISABLE/LCU_IP_DMA_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.35 Mcl_SetLcuSyncOutputForceInt()

```
void Mcl_SetLcuSyncOutputForceInt (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_INT_EN] This function Enables/Disable the generation of an interrupt request when force event occurs.

This service is a reentrant function that shall Enables/Disable the generation of an interrupt request when force event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_INT_DISABLE/LCU_IP_INT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.36 Mcl_SetLcuSyncOutputLutDma()

```
void Mcl_SetLcuSyncOutputLutDma (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_DMA_EN] This function Enables/Disable the generation of a DMA request when a LUT event occurs.

This service is a reentrant function that shall Enables/Disable the generation of a DMA request when a LUT event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DMA_DISABLE/LCU_IP_DMA_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.37 Mcl_SetLcuSyncOutputLutInt()

```
void Mcl_SetLcuSyncOutputLutInt (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_INT_EN] This function Enables/Disable the generation of an interrupt request when LUT event occurs.

This service is a reentrant function that shall Enables/Disable the generation of an interrupt request when LUT event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_INT_DISABLE/LCU_IP_INT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.38 Mcl_SetLcuSyncOutputFallFilter()

```
void Mcl_SetLcuSyncOutputFallFilter (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_FALL_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low

This service is a reentrant function that shall specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.39 Mcl_SetLcuSyncOutputRiseFilter()

```
void Mcl_SetLcuSyncOutputRiseFilter (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_RISE_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high

This service is a reentrant function that shall Specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.40 Mcl_SetLcuSyncOutputLutControl()

```
void Mcl_SetLcuSyncOutputLutControl (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LUTCTRL] This function specifies the LUT positions, based on the combined LC input value

This service is a reentrant function that shall Specifies the LUT positions, based on the combined LC input value When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.41 Mcl_ClearLcuSyncOutputForceEvent()

```
void Mcl_ClearLcuSyncOutputForceEvent (
            const Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCESTS] This function Clear Force Event

This service is a reentrant function that shall Clear Force Event When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.42 Mcl_GetLcuSyncLogicInput()

```
void Mcl_GetLcuSyncLogicInput (
            Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[LC_INPUTS] This function Indicates states of LC inputs

This service is a reentrant function that shall Indicates states of LC inputs When enable multi cores, the API shall get the values from the register only if the list contains all the Inputs related to the same partition

How to use this interface: [in] List[x].LogicInputId -> The Logic Input Id generated by the configurator [out] List[x].Value -> Store the states of LC inputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.43 Mcl_GetLcuSyncSwOverrideInput()

```
void Mcl_GetLcuSyncSwOverrideInput (
            Mcl_LcuSyncInputValueType List[],
            const uint8 Dimension )
```

[SWOUT] This function Indicates states of LC inputs or software-overridden inputs

This service is a reentrant function that shall Indicates states of LC inputs or software-overridden inputs When enable multi cores, the API shall get the values from the register only if the list contains all the Inputs related to the same partition

How to use this interface: [in] List[x].LogicInputId -> The Logic Input Id generated by the configurator [out] List[x].Value -> Store the states of LC inputs or software-overridden inputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.44 Mcl_GetLcuSyncLogicOutput()

```
void Mcl_GetLcuSyncLogicOutput (
            Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[LCOUT] This function Indicates states of LC outputs

This service is a reentrant function that shall Indicates states of LC outputs When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the states of LC outputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.45 Mcl_GetLcuSyncForceOutput()

```
void Mcl_GetLcuSyncForceOutput (
            Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCEOUT] This function Indicates the current state of force outputs for the logic outputs

This service is a reentrant function that shall the current state of force outputs for the logic outputs When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the current state of force outputs for the logic outputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.46 Mcl_GetLcuSyncForceStatus()

```
void Mcl_GetLcuSyncForceStatus (
            Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCESTS] This function Indicates that a force event has occurred on the associated output

This service is a reentrant function that shall Indicates that a force event has occurred on the associated output When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the force event value. Value = 0: force event not occurs Value = 1: force event occurs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.47 Mcl_GetLcuSyncCombineForceInput()

```
void Mcl_GetLcuSyncCombineForceInput (
            Mcl_LcuSyncOutputValueType List[],
            const uint8 Dimension )
```

[COMB_FORCE] This function Indicates the combined value of force inputs to each output

This service is a reentrant function that shall Indicates the combined value of force inputs to each output When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the combined value of force inputs to each output

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.48 Mcl_SetLcuAsyncInputList()

```
void Mcl_SetLcuAsyncInputList (
            const uint8 LogicInput,
            const Mcl_LcuAsyncInputValueType List[],
            const uint8 Dimension )
```

This function configure multiple configuration for one logic input.

This service is a reentrant function that shall configure multiple configuration for one logic input

How to use this interface: List[x].Param -> Select member of Mcl_LcuInputParamType List[x].Value -> Unsigned Integer: [0, 255] or some defines generated by configurator base on Param

Parameters

in	*LogicInput*	Logic input Id
in	*List*	List of param and value
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.49 Mcl_SetLcuAsyncOutputList()

```
void Mcl_SetLcuAsyncOutputList (
            const uint8 LogicOutput,
            const Mcl_LcuAsyncOutputValueType List[],
            const uint8 Dimension )
```

This function configure multiple configuration for one logic output.

This service is a reentrant function that shall configure multiple configuration for one logic output

How to use this interface: List[x].Param -> Select member of Mcl_LcuOutputParamType List[x].Value -> Unsigned Integer: [0, 255] or some defines generated by configurator base on Param

Parameters

in	*LogicOutput*	Logic output Id
in	*List*	List of param and value
in	*Dimension*	Number of entries in the List

Returns

void

6.2.6.50 Mcl_GetLcuAsyncLogicInputInfo()

```
void Mcl_GetLcuAsyncLogicInputInfo (
            const uint8 LogicInput,
            const Mcl_LcuInputInfoParamType Param,
            uint8 *const Value )
```

This function get information of Logic Input.

This service is a reentrant function that shall get states of Logic Inputs How to use this interface: [in] LogicInput -> The Logic Input Name generated by the configurator. By default: LCU_LOGIC_INPUT_0, LCU_LOGIC_↩ INPUT_1,... [in] Param -> Select parameter. Example: MCL_LCU_IP_INPUT_GET_LOGIC_INPUT_STATE [out] Value -> Store the states of LC inputs

Parameters

in	*LogicInput*	The Logic Input Name
in	*Param*	Selection parameter
out	*Value*	Output value

Returns

void

6.2.6.51 Mcl_GetLcuAsyncLogicOutputInfo()

```
void Mcl_GetLcuAsyncLogicOutputInfo (
            const uint8 LogicOutput,
            const Mcl_LcuOutputInfoParamType Param,
            uint8 *const Value )
```

This function get information of Logic Output.

This service is a reentrant function that shall get states of Logic Outputs How to use this interface: [in] LogicOutput -> The Logic Output Name generated by the configurator. By default: LCU_LOGIC_OUTPUT_0, LCU_↩ LOGIC_OUTPUT_1,... [in] Param -> Select parameter. Example: MCL_LCU_IP_OUTPUT_GET_LOGIC↩ _OUTPUT_STATE [out] Value -> Store the states of LC inputs

Parameters

in	*LogicOutput*	The Logic Output Name
in	*Param*	Selection parameter
out	*Value*	Output value

Returns

void

6.2.6.52 Mcl_GetVersionInfo()

```
void Mcl_GetVersionInfo (
            Std_VersionInfoType *const VersionInfo )
```

Returns the version information of this module.

Returns the version information of MCL module.

Parameters

out	*VersionInfo*	A pointer to a structure used to get version information.

Returns

void

6.3 DMA IP Driver

6.3.1 Detailed Description

Data Structures

- struct Dma_Ip_LogicChannelGlobalListType
  This type contains the Dma Ip Logic Channel Global List. More...
- struct Dma_Ip_LogicChannelTransferListType
  This type contains the Dma Ip Channel Transfer List. More...
- struct Dma_Ip_LogicInstanceStatusType
  This type contains the Dma Ip Instance Status. More...
- struct Dma_Ip_LogicChannelStatusType
  This type contains the Dma Ip Channel Status. More...
- struct Dma_Ip_LogicChannelIdType
  This type contains the Dma Ip Logic Channel Identification. More...
- struct Dma_Ip_GlobalConfigType
  This type contains the Dma Ip Global Configuration. More...
- struct Dma_Ip_TransferConfigType
  This type contains the Dma Ip Transfer Configuration. More...
- struct Dma_Ip_ScatterGatherConfigType
  This type contains the Dma Ip Scatter/Gather Configuration. More...
- struct Dma_Ip_LogicChannelConfigType
  This type contains the Dma Ip Logic Channel Configuration. More...
- struct Dma_Ip_LogicInstanceIdType
  This type contains the Dma Ip Logic Instance Identification. More...
- struct Dma_Ip_LogicInstanceConfigType
  This type contains the Dma Ip Logic Instance Configuration. More...

- struct Dma_Ip_HwChannelStateType

  This type contains the Dma Ip Hardware Channel State. More...
- struct Dma_Ip_VirtualSectionConfigType

  This type contains the Dma Ip Virtual memory section configuration Type. More...
- struct Dma_Ip_VirtualMemoryConfigType

  This type contains the Dma Ip Virtual memory configuration Type. More...
- struct Dma_Ip_InitType

  This type contains the Dma Ip Initialization. More...

Types Reference

- typedef void(* Dma_Ip_Callback) (void)

  This type contains the Dma Ip Callback interface.

Enum Reference

- enum Dma_Ip_LogicInstanceCmdType

  This type contains the Dma Ip Logic Instance Commands.
- enum Dma_Ip_LogicChannelCmdType

  This type contains the Dma Ip Logic Channel Commands.
- enum Dma_Ip_LogicChannelGlobalParamType

  This type contains the Dma Ip Logic Channel Global Parameters.
- enum Dma_Ip_LogicChannelTransferParamType

  This type contains the Dma Ip Logic Channel Transfer Parameters.
- enum Dma_Ip_LogicChannelInfoParamType

  This type contains the Dma Ip Logic Channel Information Parameters.
- enum Dma_Ip_ReturnType

  This type contains the Dma Ip Return Type.
- enum Dma_Ip_HwChannelStateValueType

  This type contains the Dma Ip Channel State Value Type.

Function Reference

- Dma_Ip_ReturnType Dma_Ip_Init (const Dma_Ip_InitType *const DmaInit)

  This function initializes the Dma Ip Driver.
- Dma_Ip_ReturnType Dma_Ip_Deinit (void)

  This function deinitializes the Dma Ip Driver.
- Dma_Ip_ReturnType Dma_Ip_SetLogicInstanceCommand (const uint32 LogicInst, const Dma_Ip_LogicInstanceCmdTy
  Command)

  This function sets Dma Ip Instance Command.
- Dma_Ip_ReturnType Dma_Ip_GetLogicInstanceStatus (const uint32 LogicInst, Dma_Ip_LogicInstanceStatusType
  *const Status)

  This function gets Dma Ip Instance Status.
- Dma_Ip_ReturnType Dma_Ip_LogicChannelInit (const uint32 LogicCh)

This function initializes the Dma Ip Logic Channel.

- Dma_Ip_ReturnType Dma_Ip_LogicChannelDeinit (const uint32 LogicCh)

  This function deinitializes the Dma Ip Logic Channel.
- Dma_Ip_ReturnType Dma_Ip_SetLogicChannelCommand (const uint32 LogicCh, const Dma_Ip_LogicChannelCmdTyp Command)

  This function sets Dma Ip Logic Channel Command.
- Dma_Ip_ReturnType Dma_Ip_GetLogicChannelStatus (const uint32 LogicCh, Dma_Ip_LogicChannelStatusType *const ChStatus)

  This function gets Dma Ip Logic Channel Status.
- Dma_Ip_ReturnType Dma_Ip_SetLogicChannelGlobalList (const uint32 LogicCh, const Dma_Ip_LogicChannelGlobalLi List[], const uint32 ListDimension)

  This function sets Dma Ip Logic Channel Global List settings.
- Dma_Ip_ReturnType Dma_Ip_SetLogicChannelTransferList (const uint32 LogicCh, const Dma_Ip_LogicChannelTransfe List[], const uint32 ListDimension)

  This function sets Dma Ip Logic Channel Transfer List settings.
- Dma_Ip_ReturnType Dma_Ip_SetLogicChannelScatterGatherList (const uint32 LogicCh, const uint32 Element, const Dma_Ip_LogicChannelScatterGatherListType List[], const uint32 ListDimension)

  This function sets Dma Ip Logic Channel Scatter/Gather List settings.
- Dma_Ip_ReturnType Dma_Ip_GetLogicChannelParam (const uint32 LogicCh, const Dma_Ip_LogicChannelInfoParamT Param, uint32 *const Value)

  This function gets the Dma Ip Logic Channel Parameter value.
- Dma_Ip_ReturnType Dma_Ip_SetLogicChannelScatterGatherConfig (const uint32 LogicCh, const uint32 Element)

  This function configures the Dma Ip Logic Channel Scatter/Gather.

6.3.2 Data Structure Documentation

6.3.2.1 struct Dma_Ip_LogicChannelGlobalListType

This type contains the Dma Ip Logic Channel Global List.

The Dma Ip Channel Global List contains a pair composed from Dma Channel Global Parameter Type and the Value of the parameter. The Dma Ip Channel Global Parameter Type selects a parameter form the Global Parameter enum type. The Value stores the parameter's value.

Definition at line 234 of file Dma_Ip.h.

6.3.2.2 struct Dma_Ip_LogicChannelTransferListType

This type contains the Dma Ip Channel Transfer List.

The Dma Ip Channel Transfer List contains a pair composed from Dma Channel Transfer Parameter Type and the Value of the parameter. The Dma Ip Channel Transfer Parameter Type selects a parameter form the Transfer Parameter enum type. The Value stores the parameter's value.

This type contains the Dma Ip Channel ScatterGather List.

The Dma Ip Channel Transfer List contains a pair composed from Dma Channel ScatterGather Parameter Type and the Value of the parameter. The Dma Ip Channel ScatterGather Parameter Type selects a parameter form the ScatterGather Parameter enum type. The Value stores the parameter's value.

Definition at line 257 of file Dma_Ip.h.

6.3.2.3 struct Dma_Ip_LogicInstanceStatusType

This type contains the Dma Ip Instance Status.

The Dma Ip Instance Status contains the Hardware Errors, Active Id and Active indication for the running Dma Channel. The Errors shall contain the Hardware Errors. The ActiveId shall contain the running Dma Channel Id. The Active shall contain the running Dma Channel Active status.

Definition at line 273 of file Dma_Ip.h.

Data Fields

- uint32 Errors

  [VALUE] The Errors value is read from the DMA Instance Error Register (ES) as it is.
- uint8 ActiveId

  [VALUE] The ActiveId value is read from the DMA Instance Control Register (CR) field ACTIVE_ID.
- boolean Active

  [BOOLEAN] The Active value is read from the DMA Instance Control Register (CR) field ACTIVE.
- uint8 Version

  [VALUE] The Version value is read from the DMA Instance Control Register (CR) field VER.

6.3.2.3.1 Field Documentation

6.3.2.3.1.1 Errors `uint32 Errors`

[VALUE] The Errors value is read from the DMA Instance Error Register (ES) as it is.

Definition at line 274 of file Dma_Ip.h.

6.3.2.3.1.2 ActiveId `uint8 ActiveId`

[VALUE] The ActiveId value is read from the DMA Instance Control Register (CR) field ACTIVE_ID.

Definition at line 275 of file Dma_Ip.h.

6.3.2.3.1.3 Active `boolean Active`

[BOOLEAN] The Active value is read from the DMA Instance Control Register (CR) field ACTIVE.

Definition at line 276 of file Dma_Ip.h.

6.3.2.3.1.4 Version uint8 Version

[VALUE] The Version value is read from the DMA Instance Control Register (CR) field VER.

Definition at line 277 of file Dma_Ip.h.

6.3.2.4 struct Dma_Ip_LogicChannelStatusType

This type contains the Dma Ip Channel Status.

The Dma Ip Channel Status contains the Hardware Errors, Active status and Done indication for the running Dma Channel. The Channel State Value shall contain the internal driver state of the Dma Channel. The Errors shall contain the Hardware Dma Channel Errors. The Active shall contain the running Dma Channel Id. The Done shall contain the running Dma Channel Active status.

Definition at line 291 of file Dma_Ip.h.

Data Fields

- Dma_Ip_HwChannelStateValueType ChStateValue

  [VALUE] The ChStateValue value is read from the internal DMA Driver Channel State Machine. Check UM for additional information.

- uint32 Errors

  [VALUE] The Errors value is read from the DMA Channel Error Register (CHx_ES) as it is.
- boolean Active

  [BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field ACTIVE.
- boolean Done

  [BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field DONE.

6.3.2.4.1 Field Documentation

6.3.2.4.1.1 ChStateValue Dma_Ip_HwChannelStateValueType ChStateValue

[VALUE] The ChStateValue value is read from the internal DMA Driver Channel State Machine. Check UM for additional information.

Definition at line 292 of file Dma_Ip.h.

6.3.2.4.1.2 Errors uint32 Errors

[VALUE] The Errors value is read from the DMA Channel Error Register (CHx_ES) as it is.

Definition at line 294 of file Dma_Ip.h.

6.3.2.4.1.3 Active `boolean Active`

[BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field ACTIVE.

Definition at line 295 of file Dma_Ip.h.

6.3.2.4.1.4 Done `boolean Done`

[BOOLEAN] The Active value is read from the DMA Channel Control and Status Register (CHx_CSR) field DONE.

Definition at line 296 of file Dma_Ip.h.

6.3.2.5 struct Dma_Ip_LogicChannelIdType

This type contains the Dma Ip Logic Channel Identification.

The Logic Channel is identified by the following structure: The Logic Channel Id contains the ID value. The Hardware Version Id contains the DMA Hardware Ip Block Version. The Hardware Instance Id contains the DMA Hardware Instance identification. The Hardware Channel Id contains the DMA Hardware Channel identification. The Hardware Crc Instance Id contains the DMA Hardware Crc Instance identification if CRC_IP is available. The Hardware Crc Channel Id contains the DMA Hardware Crc Channel identification if CRC_IP is available. The Interrupt Callback stores pointer to the user defined interrupt callback. The Error Interrupt Callback stores pointer to the user defined error interrupt callback.

Definition at line 179 of file Dma_Ip_Types.h.

Data Fields

- uint32 LogicChId

  DMA logic channel number.
- uint8 HwVersId

  DMA hardware version.
- uint8 HwInstId

  DMA hardware instance number.
- uint8 HwChId

  DMA hardware channel number.
- Dma_Ip_Callback IntCallback

  The channel callback is installed in the interrupt and is called automatically from the interrupt every time it is triggered.
- Dma_Ip_Callback ErrIntCallback

  The channel error callback is installed in the error interrupt and is called automatically from the interrupt every time it is triggered.

6.3.2.5.1 Field Documentation

6.3.2.5.1.1 LogicChId `uint32 LogicChId`

DMA logic channel number.

Definition at line 180 of file Dma_Ip_Types.h.

6.3.2.5.1.2 HwVersId `uint8 HwVersId`

DMA hardware version.

Definition at line 181 of file Dma_Ip_Types.h.

6.3.2.5.1.3 HwInstId `uint8 HwInstId`

DMA hardware instance number.

Definition at line 182 of file Dma_Ip_Types.h.

6.3.2.5.1.4 HwChId `uint8 HwChId`

DMA hardware channel number.

Definition at line 183 of file Dma_Ip_Types.h.

6.3.2.5.1.5 IntCallback `Dma_Ip_Callback IntCallback`

The channel callback is installed in the interrupt and is called automatically from the interrupt every time it is triggered.

Definition at line 184 of file Dma_Ip_Types.h.

6.3.2.5.1.6 ErrIntCallback `Dma_Ip_Callback ErrIntCallback`

The channel error callback is installed in the error interrupt and is called automatically from the interrupt every time it is triggered.

Definition at line 186 of file Dma_Ip_Types.h.

6.3.2.6 struct Dma_Ip_GlobalConfigType

This type contains the Dma Ip Global Configuration.

The Global Configuration of an Logic Channel, contains all the information describing a channel, but are not present in the DMA TCD. It contains configuration split in four main areas: Control defines functionality covering the channel's bus control. Request defines functionality covering the channel's request interface Interrupt defines functionality covering special channel interrupts. Priority defines functionality covering the channel's priority mechanism.

Definition at line 202 of file Dma_Ip_Types.h.

6.3.2.7 struct Dma_Ip_TransferConfigType

This type contains the Dma Ip Transfer Configuration.

The Transfer Configuration of an Logic Channel, contains all the information describing a channel transfer functionalities and are present in the DMA TCD. It contains configuration split in five main areas: Control defines functionality covering the channel control functions. It includes additionaly the Scatter/Gather Address and Destination Store Address.

Definition at line 235 of file Dma_Ip_Types.h.

6.3.2.8 struct Dma_Ip_ScatterGatherConfigType

This type contains the Dma Ip Scatter/Gather Configuration.

The Scatter/Gather Configuration of a Logic Channel, contains all the information describing the channel's needed resources for a Scatter/Gather element. It contains a pointer to the transfer configuration, a pointer to a Software TCD, a pointer to the next Scatter/Gather configuration and a status flag for loading the transfer configuration into the Software TCD.

Definition at line 294 of file Dma_Ip_Types.h.

Data Fields

Type	Name	Description
Dma_Ip_TransferConfigType *	TransferConfig	Configuration that shall be loaded into the Software TCD.
Dma_Ip_SwTcdRegType *	Stcd	Software TCD shall be loaded with own TCD configuration.
boolean	Loaded	
struct Dma_Ip_ScatterGatherConfigType *	NextConfig	Next Logic Channel configuration. If the address is not NULL, then the ESG flag is enabled (Scatter/Gather address) and the destination adjustment is disabled. The next configuration address is stored.

6.3.2.9 struct Dma_Ip_LogicChannelConfigType

This type contains the Dma Ip Logic Channel Configuration.

The Logic Channel Configuration consists of the Logic Channel Identifier, pointer to the Global Configuration, pointer to the Transfer Configuration and pointer to the Scatter/Gather configuration. The Logic Channel Configuration contains all data needed to initialize a Logic Channel.

Definition at line 313 of file Dma_Ip_Types.h.

Data Fields

Type	Name	Description
Dma_Ip_LogicChannelIdType	LogicChId	The Logic Channel ID contains configuration information and identification.
const Dma_Ip_GlobalConfigType *	pxGlobalConfig	The Global Configuration pointer shall contain the global data.
const Dma_Ip_TransferConfigType *	pxTransferConfig	The Transfer Configuration pointer shall contain the transfer data.
Dma_Ip_ScatterGatherConfigType **	ppxScatterGatherConfigArray	The Scatter/Gather Configuration pointer shall contain a pointer to an array containing all Scatter/Gather Logic Elements.

6.3.2.10 struct Dma_Ip_LogicInstanceIdType

This type contains the Dma Ip Logic Instance Identification.

The Logic Instance is identified by the following structure: The Logic Instance Id contains the ID value. The Hardware Version Id contains the DMA Hardware Ip Block Version. The Hardware Instance Id contains the DMA Hardware Instance identification.

Definition at line 329 of file Dma_Ip_Types.h.

Data Fields

Type	Name	Description
uint32	LogicInstId	DMA logic instance number.
uint8	HwVersId	DMA hardware version number.
uint8	HwInstId	DMA hardware instance number.

6.3.2.11 struct Dma_Ip_LogicInstanceConfigType

This type contains the Dma Ip Logic Instance Configuration.

The Logic Instance Configuration contains all the information describing an instance functionality.

Definition at line 341 of file Dma_Ip_Types.h.

Data Fields

Type	Name	Description
Dma_Ip_LogicInstanceIdType	LogicInstId	DMA logic instance number.
boolean	EnDebug	When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0.
boolean	EnRoundRobin	Enable Round Robin Channel Arbitration.
boolean	EnHaltAfterError	When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0.
boolean	EnChLinking	Global Channel Linking Control.
boolean	EnGlMasterIdReplication	If master ID replication is disabled, the privileged protection level (Supervisor mode) for DMA transfers is used.

6.3.2.12 struct Dma_Ip_HwChannelStateType

This type contains the Dma Ip Hardware Channel State.

The Hardware Channel State contains the channel's state based on runtime actions. The structure links the hardware channel state with the Logic Channel.

Definition at line 359 of file Dma_Ip_Types.h.

6.3.2.13 struct Dma_Ip_VirtualSectionConfigType

This type contains the Dma Ip Virtual memory section configuration Type.

Definition at line 383 of file Dma_Ip_Types.h.

6.3.2.14 struct Dma_Ip_VirtualMemoryConfigType

This type contains the Dma Ip Virtual memory configuration Type.

Definition at line 395 of file Dma_Ip_Types.h.

6.3.2.15 struct Dma_Ip_InitType

This type contains the Dma Ip Initialization.

The Dma Ip Initialization contains all the information required to initialize the Dma Peripheral. Each pointer shall be loaded with a specific configuration used be the Dma.

Definition at line 409 of file Dma_Ip_Types.h.

6.3.3 Types Reference

6.3.3.1 Dma_Ip_Callback

```
typedef void(* Dma_Ip_Callback) (void)
```

This type contains the Dma Ip Callback interface.

The Callback is defined by the user and installed by the driver in the corresponding IRQ.

Returns

void

Definition at line 163 of file Dma_Ip_Types.h.

6.3.4 Enum Reference

6.3.4.1 Dma_Ip_LogicInstanceCmdType

```
enum Dma_Ip_LogicInstanceCmdType
```

This type contains the Dma Ip Logic Instance Commands.

The Commands trigger specific actions in the Dma Logic Instance.

Enumerator

DMA_IP_INST_CANCEL_TRANSFER	The Cancel Transfer cancels the executing channel and forces the Minor Loop to finish.
DMA_IP_INST_CANCEL_TRANSFER_WITH↩_ERROR	The Cancel Transfer With Error Command cancels the executing channel, forces the Minor Loop to finish and generates an error interrupt.
DMA_IP_INST_HALT	The Halt Command allows the ongoing transfer to finish and halts any new transfer.
DMA_IP_INST_RESUME	The Resume Command allows the transfer to continue.

Definition at line 105 of file Dma_Ip.h.

6.3.4.2 Dma_Ip_LogicChannelCmdType

enum Dma_Ip_LogicChannelCmdType

This type contains the Dma Ip Logic Channel Commands.

The Commands trigger specific actions in the Dma Ip Logic Channel.

Enumerator

DMA_IP_CH_SET_HARDWARE_REQUEST	The Set Hardware Request Command enables the Dma Channel to be triggered by hardware requests.
DMA_IP_CH_CLEAR_HARDWARE_REQUEST	The Clear Hardware Request Command disables the Dma Channel to be triggered by hardware requests.
DMA_IP_CH_SET_SOFTWARE_REQUEST	The Set Software Request Command sends a soft start request to the Dma Channel.
DMA_IP_CH_CLEAR_DONE	The Clear Done Command resets the Dma Channel Done status.
DMA_IP_CH_CLEAR_ERROR	The Clear Error Command resets the Dma Channel Error status.

Definition at line 117 of file Dma_Ip.h.

6.3.4.3 Dma_Ip_LogicChannelGlobalParamType

enum Dma_Ip_LogicChannelGlobalParamType

This type contains the Dma Ip Logic Channel Global Parameters.

The Parameters set specific functionalities for the Dma Ip Logic Channel.

Enumerator

DMA_IP_CH_SET_EN_MASTER_ID_↩ REPLICATION	[BOOLEAN] The EnMasterIdReplication Parameter sets the Dma Channel to use the same protection level and system bus ID of the master programming the Dma Channel.
DMA_IP_CH_SET_EN_BUFFERED_WRITES	[BOOLEAN] The EnBufferedWrites Parameter sets the Dma Channel writes to be bufferable.
DMA_IP_CH_SET_EN_HARDWARE_REQ	[BOOLEAN] The EnRequest Parameter enables the Dma Channel Request.

Enumerator

DMA_IP_CH_SET_EN_ERROR_INTERRUPT	[BOOLEAN] The EnError Parameter enables the Dma Channel Error Interrupt.
DMA_IP_CH_SET_GROUP_PRIORITY	[VALUE] The Group Parameter sets the Dma Channel Group Priority.
DMA_IP_CH_SET_LEVEL_PRIORITY	[VALUE] The Level Parameter sets the Dma Channel Level Priority.
DMA_IP_CH_SET_EN_PREEMPTION_↩ PRIORITY	[BOOLEAN] The EnPreemption Parameter enables the Dma Channel Preemption.
DMA_IP_CH_SET_DIS_PREEMPT_PRIORITY	[BOOLEAN] The DisPreempt Parameter disables the Dma Channel Preempt.

Definition at line 130 of file Dma_Ip.h.

6.3.4.4 Dma_Ip_LogicChannelTransferParamType

enum Dma_Ip_LogicChannelTransferParamType

This type contains the Dma Ip Logic Channel Transfer Parameters.

The Parameters set specific functionalities.

Enumerator

DMA_IP_CH_SET_SOURCE_ADDRESS	[VALUE] The Source Address Parameter sets the Dma Channel source address value.
DMA_IP_CH_SET_SOURCE_SIGNED_OFFSET	[VALUE] The Source Signed Offset Parameter sets the Dma Channel source signed offset value.
DMA_IP_CH_SET_SOURCE_SIGNED_LAST↩ _ADDR_ADJ	[VALUE] The Source Signed Last Address Adjustment Parameter sets the Dma Channel source signed last address adjustment.
DMA_IP_CH_SET_SOURCE_TRANSFER_SIZE	[VALUE] The Source Transfer Size Parameter sets the Dma Channel source transfer size.
DMA_IP_CH_SET_SOURCE_MODULO	[VALUE] The Source Modulo Parameter sets the Dma Channel source modulo.
DMA_IP_CH_SET_DESTINATION_ADDRESS	[VALUE] The Destination Address Parameter sets the Dma Channel destination address value.
DMA_IP_CH_SET_DESTINATION_SIGNED_↩ OFFSET	[VALUE] The Destination Signed Offset Parameter sets the Dma Channel destination signed offset value.
DMA_IP_CH_SET_DESTINATION_SIGNED_↩ LAST_ADDR_ADJ	[VALUE] The Destination Signed Last Address Adjustment Parameter sets the Dma Channel destination signed last address adjustment.
DMA_IP_CH_SET_DESTINATION_↩ TRANSFER_SIZE	[VALUE] The Destination Transfer Size Parameter sets the Dma Channel destination transfer size.
DMA_IP_CH_SET_DESTINATION_MODULO	[VALUE] The Destination Modulo Parameter sets the Dma Channel destination modulo.

Enumerator

DMA_IP_CH_SET_MINORLOOP_EN_SRC_↩ OFFSET	[BOOLEAN] The Minor Loop Enable Source Offset Parameter enables the Dma Channel minor loop source offset.
DMA_IP_CH_SET_MINORLOOP_EN_DST_↩ OFFSET	[BOOLEAN] The Minor Loop Enable Destination Offset Parameter enables the Dma Channel minor loop destination offset.
DMA_IP_CH_SET_MINORLOOP_SIGNED_↩ OFFSET	[VALUE] The Minor Loop Signed Offset Parameter sets the Dma Channel minor loop signed offset.
DMA_IP_CH_SET_MINORLOOP_EN_LINK	[BOOLEAN] The Minor Loop Enable Link Parameter enables the Dma Channel minor loop logic channel linking.
DMA_IP_CH_SET_MINORLOOP_LOGIC_↩ LINK_CH	[VALUE] The Minor Loop Logic Channel Link Parameter sets the Dma Channel minor loop logic channel link.
DMA_IP_CH_SET_MINORLOOP_SIZE	[VALUE] The Minor Loop Size Parameter sets the Dma Channel minor loop transfer size.
DMA_IP_CH_SET_MAJORLOOP_EN_LINK	[BOOLEAN] The Major Loop Enable Link Parameter enables the Dma Channel major loop logic channel linking.
DMA_IP_CH_SET_MAJORLOOP_LOGIC_↩ LINK_CH	[VALUE] The Major Loop Logic Channel Link Parameter sets the Dma Channel major loop logic channel link.
DMA_IP_CH_SET_MAJORLOOP_COUNT	[VALUE] The Major Loop Count Parameter sets the Dma Channel major loop count.
DMA_IP_CH_SET_CONTROL_STORE_DST_↩ ADDR	[VALUE] The Store Destination Address Parameter saves the final destination address in system memory.
DMA_IP_CH_SET_CONTROL_SOFTWARE_↩ REQUEST	[BOOLEAN] The Enable Start Parameter enables the Dma Channel start service request. The main usage is for ScatterGather Element configuration.
DMA_IP_CH_SET_CONTROL_EN_MAJOR_↩ INTERRUPT	[BOOLEAN] The Enable Major Interrupt Parameter enables the Dma Channel major interrupt.
DMA_IP_CH_SET_CONTROL_EN_HALF_↩ MAJOR_INTERRUPT	[BOOLEAN] The Enable Half Interrupt Parameter enables the Dma Channel half major interrupt.
DMA_IP_CH_SET_CONTROL_DIS_AUTO_↩ REQUEST	[BOOLEAN] The Disable Automatic Request Parameter disables the Dma Channel automatic request.
DMA_IP_CH_SET_CONTROL_EN_END_OF↩ _PACKET_SIGNAL	[BOOLEAN] The Enable End Of Packet Signal Parameter enables the Dma Channel end of packet signal.
DMA_IP_CH_SET_CONTROL_BANDWIDTH	[VALUE] The Bandwidth Control Parameter sets the Dma Channel bandwidth control.

Definition at line 157 of file Dma_Ip.h.

6.3.4.5 Dma_Ip_LogicChannelInfoParamType

enum Dma_Ip_LogicChannelInfoParamType

This type contains the Dma Ip Logic Channel Information Parameters.

The Parameters get specific information.

Enumerator

DMA_IP_CH_GET_SOURCE_ADDRESS	[VALUE] The Source Address Parameter gets the Dma Channel source address.
DMA_IP_CH_GET_DESTINATION_ADDRESS	[VALUE] The Destination Address Parameter gets the Dma Channel destination address.
DMA_IP_CH_GET_BEGIN_ITER_COUNT	[VALUE] The Begin Iteration Count Parameter gets the Dma Channel begin iteration count.
DMA_IP_CH_GET_CURRENT_ITER_COUNT	[VALUE] The Current Iteration Count Parameter gets the Dma Channel current iteration count.
DMA_IP_CH_GET_STORE_DST_ADDR	[VALUE] The Store Destination Address Parameter gets the Dma Channel stored destination address.
DMA_IP_CH_GET_MASTER_ID	[VALUE] The Master Id Parameter gets the Dma Channel master id.
DMA_IP_CH_GET_MAJOR_INTERRUPT	[BOOLEAN] The Major Interrupt Parameter gets the Dma Channel major interrupt.
DMA_IP_CH_GET_HALF_MAJOR_↩ INTERRUPT	[BOOLEAN] The Half Major Interrupt Parameter gets the Dma Channel half major interrupt.

Definition at line 199 of file Dma_Ip.h.

6.3.4.6 Dma_Ip_ReturnType

enum Dma_Ip_ReturnType

This type contains the Dma Ip Return Type.

The Return Type give information for the execution of interfaces.

Definition at line 127 of file Dma_Ip_Types.h.

6.3.4.7 Dma_Ip_HwChannelStateValueType

enum Dma_Ip_HwChannelStateValueType

This type contains the Dma Ip Channel State Value Type.

The Channel State type provides information about the channel's general state. The Reset State is present when the Dma Channel is uninitialized. The Ready State is present when the Dma Channel is initialized and without error. The Error State is present when the Dma Channel is initialized and with error.

Definition at line 144 of file Dma_Ip_Types.h.

6.3.5 Function Reference

6.3.5.1 Dma_Ip_Init()

```
Dma_Ip_ReturnType Dma_Ip_Init (
            const Dma_Ip_InitType *const DmaInit )
```

This function initializes the Dma Ip Driver.

This service is a non reentrant function that shall initialize the Dma Ip driver.

Parameters

in	*DmaInit*	Pointer to the configuration structure.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ResetState.

6.3.5.2 Dma_Ip_Deinit()

```
Dma_Ip_ReturnType Dma_Ip_Deinit (
            void  )
```

This function deinitializes the Dma Ip Driver.

This service is a non reentrant function that shall deinitialize the Dma Ip driver.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the deinitialization finished ok

6.3.5.3 Dma_Ip_SetLogicInstanceCommand()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicInstanceCommand (
            const uint32 LogicInst,
            const Dma_Ip_LogicInstanceCmdType Command )
```

This function sets Dma Ip Instance Command.

This service is a reentrant function that shall command the Dma Instance. The command shall trigger specific functionalities of the Dma Instance.

Parameters

in	*LogicInst*	Selection value of the Logic Instance.
in	*Command*	The command for the Logic Instance.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the command finished ok.

6.3.5.4 Dma_Ip_GetLogicInstanceStatus()

```
Dma_Ip_ReturnType Dma_Ip_GetLogicInstanceStatus (
            const uint32 LogicInst,
            Dma_Ip_LogicInstanceStatusType *const Status )
```

This function gets Dma Ip Instance Status.

This service is a reentrant function that shall get the Dma Instance status. The command shall read specific functionalities of the Dma Instance.

Parameters

in	*LogicInst*	Selection value of the Logic Instance.
out	*Status*	Pointer to the Dma Instance status.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the get status finished ok.

6.3.5.5 Dma_Ip_LogicChannelInit()

```
Dma_Ip_ReturnType Dma_Ip_LogicChannelInit (
            const uint32 LogicCh )
```

This function initializes the Dma Ip Logic Channel.

This service is a non reentrant function that shall initialize the Dma Ip Logic Channel.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ResetState.

6.3.5.6 Dma_Ip_LogicChannelDeinit()

```
Dma_Ip_ReturnType Dma_Ip_LogicChannelDeinit (
            const uint32 LogicCh )
```

This function deinitializes the Dma Ip Logic Channel.

This service is a non reentrant function that shall deinitialize the Dma Ip Logic Channel.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the deinitialization finished ok.

6.3.5.7 Dma_Ip_SetLogicChannelCommand()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicChannelCommand (
            const uint32 LogicCh,
            const Dma_Ip_LogicChannelCmdType Command )
```

This function sets Dma Ip Logic Channel Command.

This service is a reentrant function that shall command the Dma Channel. The command shall trigger specific functionalities of the Dma Channel.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.
in	*Command*	The command for the Logic Channel.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ReadyState.

6.3.5.8 Dma_Ip_GetLogicChannelStatus()

```
Dma_Ip_ReturnType Dma_Ip_GetLogicChannelStatus (
            const uint32 LogicCh,
            Dma_Ip_LogicChannelStatusType *const ChStatus )
```

This function gets Dma Ip Logic Channel Status.

This service is a reentrant function that shall get the Dma Channel status. The command shall read specific functionalities of the Dma Channel.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.
out	*Status*	Pointer to the Dma Channel status.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the get status finished ok.

6.3.5.9 Dma_Ip_SetLogicChannelGlobalList()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicChannelGlobalList (
            const uint32 LogicCh,
            const Dma_Ip_LogicChannelGlobalListType List[],
            const uint32 ListDimension )
```

This function sets Dma Ip Logic Channel Global List settings.

This service is a reentrant function that shall set the Dma Ip Logic Channel global parameters list. The list is composed of an array of Dma Ip Logic Channel global parameters settings. The settings list(array) is defined by the user needs: it contains the desired parameters to be configured, in any desired order.

How to use this interface:

1. Use the "Dma_Ip_LogicChannelGlobalListType" to create a list(array) with the desired paramaters to configure (see parameters: "Dma_Ip_LogicChannelGlobalParamType") The list can declared globally or locally: A. Global example: Dma_Ip_LogicChannelGlobalListType global_Dma_Ip_ChannelGlobalList0[NUMBER_↩ OF_PARAMETERS] = {...}; B. Local example: Dma_Ip_LogicChannelGlobalListType Dma_Ip_Channel↩ GlobalList[NUMBER_OF_PARAMETERS]; Dma_Ip_ChannelGlobalList[PARAMETER0].Param = DMA_IP_CH_SET_EN_PREEMPTION_PRIORITY; Dma_Ip_ChannelGlobalList[PARAMETER0].Value = TRUE; Dma_Ip_ChannelGlobalList[PARAMETER1].Param = ...; Dma_Ip_ChannelGlobalList[PARAMETER1].Value = ...;
2. Call the "Dma_Ip_SetLogicChannelGlobalList()" interface: Dma_Ip_SetLogicChannelGlobalList(LOGIC↩ _CHANNELx, Dma_Ip_ChannelGlobalList, NUMBER_OF_PARAMETERS);

Parameters

in	*Channel*	Specifies the Logic Channel Id.
in	*List*	Pointer to the Global List Array.
in	*ListDimension*	Number of entries in the List.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ReadyState.

6.3.5.10 Dma_Ip_SetLogicChannelTransferList()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicChannelTransferList (
            const uint32 LogicCh,
            const Dma_Ip_LogicChannelTransferListType List[],
            const uint32 ListDimension )
```

This function sets Dma Ip Logic Channel Transfer List settings.

-> This service is a reentrant function that shall set the Dma Ip Logic Channel transfer parameters list. -> The "Transfer List" loads the configuration directly into the Hardware TCD and disables the ScatterGather for the Hardware TCD. -> The list is composed of an array of Dma Ip Logic Channel transfer parameters settings. -> The settings array is defined by the user needs: it contains entries for each desired parameter to be configured, in any suitable order.

-> How to use this interface: <-

1. Use the "Dma_Ip_LogicChannelTransferListType" to create a list(array) with the desired paramaters to configure (see parameters: "Dma_Ip_LogicChannelTransferParamType") The list can declared globally or locally: Global example: #define DMA_IP_TRANSFER_LIST0_DIMENSION ((uint32)2U) Dma_Ip_LogicChannelTransferListType global_Dma_Ip_ChannelTransferList0[DMA_IP↩ _TRANSFER_LIST0_DIMENSION] = {...}; Local example: #define DMA_IP_TRANSFER_LIST0_↩ DIMENSION ((uint32)2U) Dma_Ip_LogicChannelTransferListType Dma_Ip_ChannelTransferList0[DMA↩ _IP_TRANSFER_LIST0_DIMENSION]; Dma_Ip_ChannelTransferList0[PARAMETER0].Param = DMA_IP_CH_SET_VAL_SOURCE_ADDRESS; Dma_Ip_ChannelTransferList0[PARAMETER0].Value = &SourceBuffer; Dma_Ip_ChannelTransferList0[PARAMETER1].Param = DMA_IP_CH_SET_↩ DESTINATION_ADDRESS; Dma_Ip_ChannelTransferList0[PARAMETER1].Value = &DestinationBuffer;
2. Call the "Dma_Ip_SetLogicChannelTransferList()" interface: Dma_Ip_SetLogicChannelTransferList(↩ LOGIC_CHANNELx, Dma_Ip_ChannelTransferList0, DMA_IP_TRANSFER_LIST0_DIMENSION);

-> Coding Example: <- -> The user shall create the desired configuration list for his specific application. "UserDefinedFileName.h" #define DMA_IP_TRANSFER_LIST0_DIMENSION ((uint32)8U) #define DMA_↩ IP_SET_TRANSFER_TYPE0(CHANNEL, DIMENSION, SADDR, SOFF, SSIZE, DADDR, DOFF, DSIZE,

MINOR_SIZE, MAJOR_COUNT) \ Dma_Ip_LogicChannelTransferListType Dma_Ip_ChannelTransfer↩ List0[DIMENSION]; \ Dma_Ip_ChannelTransferList0[0U].Param = DMA_IP_CH_SET_VAL_SOURCE_↩ ADDRESS; \ Dma_Ip_ChannelTransferList0[0U].Value = SADDR; \ Dma_Ip_ChannelTransferList0[1U].Param = DMA_IP_CH_SET_SOURCE_SIGNED_OFFSET; \ Dma_Ip_ChannelTransferList0[1U].Value = SOFF; \ Dma_Ip_ChannelTransferList0[2U].Param = DMA_IP_CH_SET_SOURCE_TRANSFER_SIZE; \ Dma↩ _Ip_ChannelTransferList0[2U].Value = SSIZE; \ Dma_Ip_ChannelTransferList0[3U].Param = DMA_IP_↩ CH_SET_DESTINATION_ADDRESS; \ Dma_Ip_ChannelTransferList0[3U].Value = DADDR; \ Dma_Ip↩ _ChannelTransferList0[4U].Param = DMA_IP_CH_SET_DESTINATION_SIGNED_OFFSET; \ Dma_Ip↩ _ChannelTransferList0[4U].Value = DOFF; \ Dma_Ip_ChannelTransferList0[5U].Param = DMA_IP_CH↩ _SET_DESTINATION_TRANSFER_SIZE; \ Dma_Ip_ChannelTransferList0[5U].Value = DSIZE; \ Dma↩ _Ip_ChannelTransferList0[6U].Param = DMA_IP_CH_SET_MINORLOOP_SIZE; \ Dma_Ip_Channel↩ TransferList0[6U].Value = MINOR_SIZE; \ Dma_Ip_ChannelTransferList0[7U].Param = DMA_IP_CH_↩ SET_MAJORLOOP_COUNT; \ Dma_Ip_ChannelTransferList0[7U].Value = MAJOR_COUNT; \ Dma_Ip_↩ SetLogicChannelTransferList(CHANNEL, Dma_Ip_ChannelTransferList0, DIMENSION);

"ApplicationFileName.c" void ConfigureDmaChannel(ConfigType * pxConfig) { uint32 MinorLoopSize = 2U; uint32 MajorLoopCount; if(pxConfig->MajorLoopCondition == TRUE) { MajorLoopCount = 8U; } else { Major↩ LoopCount = 24U; } DMA_IP_SET_TRANSFER_TYPE0(pxConfig->LogicChannel, DMA_IP_TRANSFER↩ _LIST0_DIMENSION, pxConfig->SourceBuffer, 2U, DMA_IP_TRANSFER_SIZE_2_BYTE, &RegisterAddress, 0U, DMA_IP_TRANSFER_SIZE_2_BYTE, MinorLoopSize, MajorLoopCount); }

Parameters

in	*Channel*	Specifies the Logic Channel Id.
in	*List*	Pointer to the Transfer List Array.
in	*ListDimension*	Number of entries in the List.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ReadyState.

6.3.5.11 Dma_Ip_SetLogicChannelScatterGatherList()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicChannelScatterGatherList (
            const uint32 LogicCh,
            const uint32 Element,
            const Dma_Ip_LogicChannelScatterGatherListType List[],
            const uint32 ListDimension )
```

This function sets Dma Ip Logic Channel Scatter/Gather List settings.

-> This service is a reentrant function that shall set the Dma Ip Logic Channel scatter/gather parameters list. -> The "Scatter/Gather List" configures Logic Elements belonging to the same Dma Logic Channel. -> The "Scatter/↩ Gather List" loads the configuration into the Software TCD. The Software TCD has the Scatter/Gather Enable set (ESG bit) and the Next Software TCD Address already loaded during the configuration generation process. -> The "Scatter/Gather List" shall not be able to modify the Scatter/Gather Element linkage (reorder of elements in the chain). The linkage of the elements is set only during the configuration process. -> The settings array is defined

by the user needs: it contains entries for each desired parameter to be configured, in any suitable order. -> This service does not load the Logic Element into the Hardware TCD. This functionality is covered by Dma_Ip_Set↩ LogicChannelScatterGatherConfig.

How to use this interface:

1. Use the "Dma_Ip_LogicChannelScatterGatherListType" to create a list(array) with the desired paramaters to configure (see parameters: "Dma_Ip_LogicChannelTransferParamType") The list can declared globally or locally: Global example: Dma_Ip_LogicChannelScatterGatherListType global_Dma_Ip_ChannelScatter↩ GatherList0[NUMBER_OF_PARAMETERS] = {...}; Local example: Dma_Ip_LogicChannelScatter↩ GatherListType Dma_Ip_ChannelScatterGatherList[NUMBER_OF_PARAMETERS]; Dma_Ip_Channel↩ ScatterGatherList[PARAMETER0].Param = DMA_IP_CH_SET_VAL_SOURCE_ADDRESS; Dma_↩ Ip_ChannelScatterGatherList[PARAMETER0].Value = &SourceBuffer; Dma_Ip_ChannelScatterGather↩ List[PARAMETER1].Param = DMA_IP_CH_SET_DESTINATION_ADDRESS; Dma_Ip_Channel↩ ScatterGatherList[PARAMETER1].Value = &DestinationBuffer;
2. Call the "Dma_Ip_SetLogicChannelScatterGatherList()" interface: Dma_Ip_SetLogicChannelScatter↩ GatherList(LOGIC_CHANNELx, LOGIC_ELEMENTy, Dma_Ip_ChannelScatterGatherList, NUMBER↩ _OF_PARAMETERS);

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ReadyState.

6.3.5.12 Dma_Ip_GetLogicChannelParam()

```
Dma_Ip_ReturnType Dma_Ip_GetLogicChannelParam (
            const uint32 LogicCh,
            const Dma_Ip_LogicChannelInfoParamType Param,
            uint32 *const Value )
```

This function gets the Dma Ip Logic Channel Parameter value.

This service is a reentrant function that shall get the Dma Channel parameters value.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.
in	*Param*	Selection parameter.
out	*Value*	Pointer to the parameter value.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the get information finished ok.

6.3.5.13 Dma_Ip_SetLogicChannelScatterGatherConfig()

```
Dma_Ip_ReturnType Dma_Ip_SetLogicChannelScatterGatherConfig (
            const uint32 LogicCh,
            const uint32 Element )
```

This function configures the Dma Ip Logic Channel Scatter/Gather.

This service is a reentrant function that shall configure the Dma Channel scatter/gather functionality. The specified Logic Element (corresponding to a Software TCD) shall be loaded into the Dma Logic Channel's Hardware TCD. The Logic Elements (describing the Software TCDs) form a simple chained list and the "Element" parameter represents the lists's head.

Parameters

in	*LogicCh*	Selection value of the Logic Channel.
in	*Element*	Selection value of the Logic Element representing the list's head.

Returns

Dma_Ip_ReturnType DMA_IP_STATUS_SUCCESS is returned if the initialization finished ok. DMA_↩ IP_STATUS_WRONG_STATE is returned if the Dma Ip Channel state is not Dma_Ip_Ch_ReadyState. DMA_IP_STATUS_ERROR is returned if the Dma Ip Channel contains an error.

6.4 EMIOS IP Driver

6.4.1 Detailed Description

Data Structures

- struct Emios_Mcl_Ip_ConfigType

  EMIOS IP configuration structure. *More...*
- struct Emios_Ip_ChStateType

  Store EMIOS channel state. *More...*
- struct Emios_Ip_InstStateType

  Store EMIOS instance state. *More...*

Macros

- #define EMIOS_CHANNELMASK_MAXVAL

  Channel mask maxim value.

Function Reference

- void Emios_Mcl_Ip_EnableChannel (uint8 Instance, uint8 HwChannel)

  Emios start channel function.
- void Emios_Mcl_Ip_DisableChannel (uint8 Instance, uint8 HwChannel)

  Emios stop channel function.
- void Emios_Mcl_Ip_ComparatorTransferEnable (uint8 Instance, uint32 ChannelMask)

  The function shall enable the output update for the corresponding channel.
- void Emios_Mcl_Ip_ComparatorTransferDisable (uint8 Instance, uint32 ChannelMask)

  The function shall disable the output update for the corresponding channel.
- Emios_Ip_CommonStatusType Emios_Mcl_Ip_Deinit (uint8 Instance)

  This function shall reset all the register used to initialize a channel as a masterbus. The API should reset to POR values all registers used for configuring the common part of EMIOS IP.
- Emios_Ip_CommonStatusType Emios_Mcl_Ip_Init (uint8 Instance, const Emios_Mcl_Ip_ConfigType *const ConfigPtr)

  The API should configure the common part of EMIOS IP to be used by any driver implementation with EMIOS support (ex: ICU, PWM, GPT, OCU)
- void Emios_Mcl_Ip_SetReloadInterval (uint8 HwInstance, uint8 HwChannel, uint8 Interval)

  Allow the user to specify the number of bus reload events skipped.
- boolean Emios_Mcl_Ip_ValidateChannel (uint8 HwInstance, uint8 HwChannel)

  Change the period on active/intialized EMIOS counter(master) bus.
- Emios_Ip_CommonStatusType Emios_Mcl_Ip_SetCounterBusPeriod (uint8 HwInstance, uint8 HwChannel, uint32 Period)
- uint16 Emios_Mcl_Ip_GetCounterBusPeriod (uint8 Instance, uint8 Channel)

  Get period of counter bus channels.
- void Emios_Mcl_Ip_ConfigureGlobalTimebase (uint8 Instance, uint8 Value)

  Configure Global Timebase.

6.4.2 Data Structure Documentation

6.4.2.1 struct Emios_Mcl_Ip_ConfigType

EMIOS IP configuration structure.

Definition at line 147 of file Emios_Mcl_Ip_Types.h.

Data Fields

Type	Name	Description
const uint8	channelsNumber	Number of master buses used.
const Emios_Ip_GlobalConfigType *	emiosGlobalConfig	Pointer to EMIOS configuration.
const Emios_Ip_MasterBusConfigType(*	masterBusConfig)[]	Pointer to an array with all master buses used.

6.4.2.2 struct Emios_Ip_ChStateType

Store EMIOS channel state.

Definition at line 158 of file Emios_Mcl_Ip_Types.h.

6.4.2.3 struct Emios_Ip_InstStateType

Store EMIOS instance state.

Definition at line 166 of file Emios_Mcl_Ip_Types.h.

6.4.3 Macro Definition Documentation

6.4.3.1 EMIOS_CHANNELMASK_MAXVAL

```
#define EMIOS_CHANNELMASK_MAXVAL
```

Channel mask maxim value.

Definition at line 100 of file Emios_Mcl_Ip_Types.h.

6.4.4 Function Reference

6.4.4.1 Emios_Mcl_Ip_EnableChannel()

```
void Emios_Mcl_Ip_EnableChannel (
            uint8 Instance,
            uint8 HwChannel )
```

Emios start channel function.

This function is called separately for each EMIOS hw channel in order to enable it.

Parameters

in	*Instance*	Instance of EMIOS used.
in	*HwChannel*	EMIOS hardware channel used.

6.4.4.2 Emios_Mcl_Ip_DisableChannel()

```
void Emios_Mcl_Ip_DisableChannel (
```

```
            uint8 Instance,
            uint8 HwChannel )
```

Emios stop channel function.

This function is called separately for each EMIOS hw channel in order to disable it.

Parameters

in	*Instance*	Instance of EMIOS used.
in	*HwChannel*	EMIOS hardware channel used.

6.4.4.3 Emios_Mcl_Ip_ComparatorTransferEnable()

```
void Emios_Mcl_Ip_ComparatorTransferEnable (
            uint8 Instance,
            uint32 ChannelMask )
```

The function shall enable the output update for the corresponding channel.

Parameters

in	*Instance*	Instance of EMIOS used.
in	*ChannelMask*	EMIOS channel mask used to enable the output update. This mask should be on 32 bits. 0x00000001U <- First channel will be affected ^ ^ | | MSB LSB 0x00FFFFFFU <- All channels will be affected (EMIOS instance has only 24 channels)

6.4.4.4 Emios_Mcl_Ip_ComparatorTransferDisable()

```
void Emios_Mcl_Ip_ComparatorTransferDisable (
            uint8 Instance,
            uint32 ChannelMask )
```

The function shall disable the output update for the corresponding channel.

Parameters

in	*Instance*	Instance of EMIOS used.
in	*ChannelMask*	EMIOS channel mask used to disable the output update. This mask should be on 32 bits. 0x00000001U <- First channel will be affected ^ ^ | | MSB LSB 0x00FFFFFFU <- All channels will be affected (EMIOS instance has only 24 channels)

6.4.4.5 Emios_Mcl_Ip_Deinit()

```
Emios_Ip_CommonStatusType Emios_Mcl_Ip_Deinit (
            uint8 Instance )
```

This function shall reset all the register used to initialize a channel as a masterbus. The API should reset to POR values all registers used for configuring the common part of EMIOS IP.

Parameters

in	*Instance*	Instance of EMIOS used.

Returns

Emios_Ip_CommonStatusType

6.4.4.6 Emios_Mcl_Ip_Init()

```
Emios_Ip_CommonStatusType Emios_Mcl_Ip_Init (
            uint8 Instance,
            const Emios_Mcl_Ip_ConfigType *const ConfigPtr )
```

The API should configure the common part of EMIOS IP to be used by any driver implementation with EMIOS support (ex: ICU, PWM, GPT, OCU)

Parameters

in	*Instance*	Instance of EMIOS used.
in	*ConfigPtr*	EMIOS instance specific configuration.

Returns

Emios_Ip_CommonStatusType

6.4.4.7 Emios_Mcl_Ip_SetReloadInterval()

```
void Emios_Mcl_Ip_SetReloadInterval (
            uint8 HwInstance,
```

```
            uint8 HwChannel,
            uint8 Interval )
```

Allow the user to specify the number of bus reload events skipped.

Reload Signal Output Delay Interval Specifies the delay interval, in counter bus reload events, between each assertion of AS1-BS1 reload in MC and MCB modes. 00000b - Every event 00001b - Every 2nd event 00010b - Every 3rd event . . . 11111b - Every 32nd event

Parameters

HwInstance	Instance of EMIOS used.
HwChannel	EMIOS hardware channel used.
Interval	

6.4.4.8 Emios_Mcl_Ip_ValidateChannel()

```
boolean Emios_Mcl_Ip_ValidateChannel (
            uint8 HwInstance,
            uint8 HwChannel )
```

Change the period on active/intialized EMIOS counter(master) bus.

Parameters

HwInstance	Instance of EMIOS used.
HwChannel	EMIOS hardware channel used.

Returns

boolean Status of operation.

6.4.4.9 Emios_Mcl_Ip_SetCounterBusPeriod()

```
Emios_Ip_CommonStatusType Emios_Mcl_Ip_SetCounterBusPeriod (
            uint8 HwInstance,
            uint8 HwChannel,
            uint32 Period )
```

Parameters

HwInstance	Instance of EMIOS used.
HwChannel	EMIOS hardware channel used.
Period	Period

Returns

Emios_Ip_CommonStatusType

6.4.4.10 Emios_Mcl_Ip_GetCounterBusPeriod()

```
uint16 Emios_Mcl_Ip_GetCounterBusPeriod (
            uint8 Instance,
            uint8 Channel )
```

Get period of counter bus channels.

Parameters

Instance	Instance of EMIOS used.
Channel	EMIOS hardware channel used.

Returns

The period value of Counter Bus channels

6.4.4.11 Emios_Mcl_Ip_ConfigureGlobalTimebase()

```
void Emios_Mcl_Ip_ConfigureGlobalTimebase (
            uint8 Instance,
            uint8 Value )
```

Configure Global Timebase.

This function enables or disables the GTBE bit of the EMIOS->MCR register of an addressed eMIOS instance.

Parameters

in	*Instance*	Instance of EMIOS used.
in	*Value*	STD_ON to set bit and STD_OFF to reset bit.

Returns

void

6.5 FLEXIO IP Driver

6.5.1 Detailed Description

Enum Reference

- enum Flexio_Mcl_Ip_TimerPolarityType
- enum Flexio_Mcl_Ip_PinPolarityType
- enum Flexio_Mcl_Ip_PinConfigType
- enum Flexio_Mcl_Ip_TriggerPolarityType
- enum Flexio_Mcl_Ip_TriggerSourceType
- enum Flexio_Mcl_Ip_TimerModeType
- enum Flexio_Mcl_Ip_TimerOutputType
- enum Flexio_Mcl_Ip_TimerDecrementType
- enum Flexio_Mcl_Ip_TimerResetType
- enum Flexio_Mcl_Ip_TimerDisableType
- enum Flexio_Mcl_Ip_TimerEnableType
- enum Flexio_Mcl_Ip_TimerStopType
- enum Flexio_Mcl_Ip_TimerStartType

6.5.2 Enum Reference

6.5.2.1 Flexio_Mcl_Ip_TimerPolarityType

enum Flexio_Mcl_Ip_TimerPolarityType

Enumerator

FLEXIO_TIMER_POLARITY_POSEDGE	Shift on positive edge of Shift clock
FLEXIO_TIMER_POLARITY_NEGEDGE	Shift on negative edge of Shift clock

Definition at line 123 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.2 Flexio_Mcl_Ip_PinPolarityType

enum Flexio_Mcl_Ip_PinPolarityType

Enumerator

FLEXIO_PIN_POLARITY_HIGH	Pin is active high
FLEXIO_PIN_POLARITY_LOW	Pin is active low

Definition at line 131 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.3 Flexio_Mcl_Ip_PinConfigType

enum Flexio_Mcl_Ip_PinConfigType

Enumerator

FLEXIO_PIN_CONFIG_DISABLED	Shifter pin output disabled
FLEXIO_PIN_CONFIG_OPEN_DRAIN	Shifter pin open drain or bidirectional output enable
FLEXIO_PIN_CONFIG_BIDIR_OUTPUT	Shifter pin bidirectional output data
FLEXIO_PIN_CONFIG_OUTPUT	Shifter pin output

Definition at line 138 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.4 Flexio_Mcl_Ip_TriggerPolarityType

enum Flexio_Mcl_Ip_TriggerPolarityType

Enumerator

FLEXIO_TRIGGER_POLARITY_HIGH	Trigger is active high
FLEXIO_TRIGGER_POLARITY_LOW	Trigger is active low

Definition at line 171 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.5 Flexio_Mcl_Ip_TriggerSourceType

enum Flexio_Mcl_Ip_TriggerSourceType

Enumerator

FLEXIO_TRIGGER_SOURCE_EXTERNAL	External trigger selected
FLEXIO_TRIGGER_SOURCE_INTERNAL	Internal trigger selected

Definition at line 178 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.6 Flexio_Mcl_Ip_TimerModeType

enum Flexio_Mcl_Ip_TimerModeType

Enumerator

FLEXIO_TIMER_MODE_DISABLED	Timer Disabled.
FLEXIO_TIMER_MODE_8BIT_BAUD	Dual 8-bit counters baud/bit mode.
FLEXIO_TIMER_MODE_8BIT_PWM	Dual 8-bit counters PWM mode.
FLEXIO_TIMER_MODE_16BIT	Single 16-bit counter mode.
FLEXIO_TIMER_MODE_16BIT_DIS	Single 16-bit counter disable mode.
FLEXIO_TIMER_MODE_8BIT_DUAL	Dual 8-bit counters word mode.
FLEXIO_TIMER_MODE_8BIT_DUAL_PWM	Dual 8-bit counters PWM low mode.
FLEXIO_TIMER_16BIT_INPUT_CAPTURE_MODE	Single 16-bit input capture mode.

Definition at line 185 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.7 Flexio_Mcl_Ip_TimerOutputType

enum Flexio_Mcl_Ip_TimerOutputType

Enumerator

FLEXIO_TIMER_INITOUT_ONE	Timer output is logic one when enabled, unaffected by timer reset.
FLEXIO_TIMER_INITOUT_ZERO	Timer output is logic zero when enabled, unaffected by timer reset.
FLEXIO_TIMER_INITOUT_ONE_RESET	Timer output is logic one when enabled and on timer reset.
FLEXIO_TIMER_INITOUT_ZERO_RESET	Timer output is logic zero when enabled and on timer reset.

Definition at line 198 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.8 Flexio_Mcl_Ip_TimerDecrementType

enum Flexio_Mcl_Ip_TimerDecrementType

Enumerator

FLEXIO_TIMER_DECREMENT_CLK_SHIFT↩ _TMR	Decrement counter on FlexIO clock, Shift clock equals Timer output.
FLEXIO_TIMER_DECREMENT_TRG_SHIFT↩ _TMR	Decrement counter on Trigger input (both edges), Shift clock equals Timer output.
FLEXIO_TIMER_DECREMENT_PIN_SHIFT_↩ PIN	Decrement counter on Pin input (both edges), Shift clock equals Pin input.
FLEXIO_TIMER_DECREMENT_TRG_SHIFT↩ _TRG	Decrement counter on Trigger input (both edges), Shift clock equals Trigger input.

Definition at line 207 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.9 Flexio_Mcl_Ip_TimerResetType

enum Flexio_Mcl_Ip_TimerResetType

Enumerator

FLEXIO_TIMER_RESET_NEVER	Timer never reset.
FLEXIO_TIMER_RESET_PIN_OUT	Timer reset on Timer Pin equal to Timer Output.
FLEXIO_TIMER_RESET_TRG_OUT	Timer reset on Timer Trigger equal to Timer Output.
FLEXIO_TIMER_RESET_PIN_RISING	Timer reset on Timer Pin rising edge.
FLEXIO_TIMER_RESET_TRG_RISING	Timer reset on Trigger rising edge.
FLEXIO_TIMER_RESET_TRG_BOTH	Timer reset on Trigger rising or falling edge.

Definition at line 216 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.10 Flexio_Mcl_Ip_TimerDisableType

enum Flexio_Mcl_Ip_TimerDisableType

Enumerator

FLEXIO_TIMER_DISABLE_NEVER	Timer never disabled.
FLEXIO_TIMER_DISABLE_TIM_DISABLE	Timer disabled on Timer N-1 disable.
FLEXIO_TIMER_DISABLE_TIM_CMP	Timer disabled on Timer compare.

Enumerator

FLEXIO_TIMER_DISABLE_TIM_CMP_TRG_↩ LOW	Timer disabled on Timer compare and Trigger Low.
FLEXIO_TIMER_DISABLE_PIN	Timer disabled on Pin rising or falling edge.
FLEXIO_TIMER_DISABLE_PIN_TRG_HIGH	Timer disabled on Pin rising or falling edge provided Trigger is high.
FLEXIO_TIMER_DISABLE_TRG	Timer disabled on Trigger falling edge.

Definition at line 227 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.11 Flexio_Mcl_Ip_TimerEnableType

enum Flexio_Mcl_Ip_TimerEnableType

Enumerator

FLEXIO_TIMER_ENABLE_ALWAYS	Timer always enabled.
FLEXIO_TIMER_ENABLE_TIM_ENABLE	Timer enabled on Timer N-1 enable.
FLEXIO_TIMER_ENABLE_TRG_HIGH	Timer enabled on Trigger high.
FLEXIO_TIMER_ENABLE_TRG_PIN_HIGH	Timer enabled on Trigger high and Pin high.
FLEXIO_TIMER_ENABLE_PIN_POSEDGE	Timer enabled on Pin rising edge.
FLEXIO_TIMER_ENABLE_PIN_POSEDGE_TRG↩ _HIGH	Timer enabled on Pin rising edge and Trigger high.
FLEXIO_TIMER_ENABLE_TRG_POSEDGE	Timer enabled on Trigger rising edge.
FLEXIO_TIMER_ENABLE_TRG_EDGE	Timer enabled on Trigger rising or falling edge.

Definition at line 239 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.12 Flexio_Mcl_Ip_TimerStopType

enum Flexio_Mcl_Ip_TimerStopType

Enumerator

FLEXIO_TIMER_STOP_BIT_DISABLED	Stop bit disabled.
FLEXIO_TIMER_STOP_BIT_TIM_CMP	Stop bit is enabled on timer compare.
FLEXIO_TIMER_STOP_BIT_TIM_DIS	Stop bit is enabled on timer disable.
FLEXIO_TIMER_STOP_BIT_TIM_CMP_DIS	Stop bit is enabled on timer compare and disable.

Definition at line 252 of file Flexio_Mcl_Ip_HwAccess.h.

6.5.2.13 Flexio_Mcl_Ip_TimerStartType

enum Flexio_Mcl_Ip_TimerStartType

Enumerator

FLEXIO_TIMER_START_BIT_DISABLED	Start bit disabled.
FLEXIO_TIMER_START_BIT_ENABLED	Start bit enabled.

Definition at line 278 of file Flexio_Mcl_Ip_HwAccess.h.

6.6 LCU IP Driver

6.6.1 Detailed Description

Data Structures

- struct Lcu_Ip_SyncInputValueType

  This type contains the Lcu Ip Multiple Inputs and Multiple Value. More...
- struct Lcu_Ip_SyncOutputValueType

  This type contains the Lcu Ip Multiple Outputs and Multiple Value. More...
- struct Lcu_Ip_AsyncInputValueType

  This type contains the Lcu Ip Input with multiple value. More...
- struct Lcu_Ip_AsyncOutputValueType

  This type contains the Lcu Ip Output with multiple value. More...

Types Reference

- typedef void(* Lcu_Ip_Callback) (Lcu_Ip_Event Event)

  This type contains the LCU IP Callback interface.

Enum Reference

- enum Lcu_Ip_InputParamType

  This type contains the LCU Input Param Type.
- enum Lcu_Ip_OutputParamType

  This type contains the LCU Output Param Type.
- enum Lcu_Ip_LogicInputInfoParamType

  This type contains the LCU Param Type to get information for input.
- enum Lcu_Ip_LogicOutputInfoParamType

  This type contains the LCU Param Type to get information for output.
- enum Lcu_Ip_Event

  This type contains the LCU IP Event.
- enum Lcu_Ip_ReturnType

  This type contains the LCU IP Return Type.

Function Reference

- Lcu_Ip_ReturnType Lcu_Ip_Init (const Lcu_Ip_InitType *const pxLcuInit)

  This function initializes the Lcu Ip Driver.
- Lcu_Ip_ReturnType Lcu_Ip_Deinit (void)

  This function De-initializes the Lcu Ip Driver.
- Lcu_Ip_ReturnType Lcu_Ip_SetWriteProtect (const uint8 LogicInstance)

  [WP] This function Enable Write Protect feature for the Logic Instance.
- Lcu_Ip_ReturnType Lcu_Ip_GetWriteProtect (const uint8 LogicInstance)

  [WP] This function get Write Protect status for the Logic Instance
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwOverrideEnable (const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [SWEN] This function Enable/Disable software override of LC input.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwOverrideValue (const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [SWVALUE] This function specifies the software override value for each LC input.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputMuxSelect (const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [MUXSEL] This function selects the source of the LC input.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwSyncMode (const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [SW_MODE] specifies the software sync mode for the inputs to this LC.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputDebugMode (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [DBGEN] This function Enables/Disables outputs to continue operation in Debug mode.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputEnable (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [OUTEN] This function Enables/Disables LC outputs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceInputSensitivity (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCE_SENSE] This function specifies which force inputs affect output.

- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceClearingMode (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCE_MODE] This function specifies the timing for clearing force events for output.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceSyncSelect (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [SYNC_SEL] This function specifies which sync input to use for Force signal.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputPolarity (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [OUTPOL] This function specifies the polarity of the outputs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceDma (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCE_DMA_EN] This function Enables/Disable the generation of a DMA request when a force event occurs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceInt (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCE_INT_EN] This function Enables/Disable the generation of an interrupt request when force event occurs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutDma (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LUT_DMA_EN] This function Enables/Disable the generation of a DMA request when a LUT event occurs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutInt (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LUT_INT_EN] This function Enables/Disable the generation of an interrupt request when LUT event occurs.
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputFallFilter (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LUT_FALL_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputRiseFilter (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LUT_RISE_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high
- Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutControl (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LUTCTRL] This function specifies the LUT positions, based on the combined LC input value
- Lcu_Ip_ReturnType Lcu_Ip_ClearSyncOutputForceEvent (const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCESTS] This function Clear Force Event
- Lcu_Ip_ReturnType Lcu_Ip_GetSyncLogicInput (Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [LC_INPUTS] This function Indicates states of LC inputs
- Lcu_Ip_ReturnType Lcu_Ip_GetSyncSwOverrideInput (Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)

  [SWOUT] This function Indicates states of LC inputs or software-overridden inputs
- Lcu_Ip_ReturnType Lcu_Ip_GetSyncLogicOutput (Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [LCOUT] This function Indicates states of LC outputs
- Lcu_Ip_ReturnType Lcu_Ip_GetSyncForceOutput (Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [FORCEOUT] This function Indicates the current state of force outputs for the logic outputs
- Lcu_Ip_ReturnType Lcu_Ip_GetSyncForceStatus (Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

[FORCESTS] This function Indicates that a force event has occurred on the associated output

- Lcu_Ip_ReturnType Lcu_Ip_GetSyncCombineForceInput (Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)

  [COMB_FORCE] This function Indicates the combined value of force inputs to each output
- Lcu_Ip_ReturnType Lcu_Ip_SetAsyncInputList (const uint8 LogicInput, const Lcu_Ip_AsyncInputValueType List[], const uint8 Dimension)

  This function configure multiple configuration for one logic input.
- Lcu_Ip_ReturnType Lcu_Ip_SetAsyncOutputList (const uint8 LogicOutput, const Lcu_Ip_AsyncOutputValueType List[], const uint8 Dimension)

  This function configure multiple configuration for one logic output.
- Lcu_Ip_ReturnType Lcu_Ip_GetAsyncLogicInputInfo (const uint8 LogicInput, const Lcu_Ip_LogicInputInfoParamType Param, uint8 *const Value)

  This function get information of Logic Input.
- Lcu_Ip_ReturnType Lcu_Ip_GetAsyncLogicOutputInfo (const uint8 LogicOutput, const Lcu_Ip_LogicOutputInfoParam Param, uint8 *const Value)

  This function get information of Logic Output.

6.6.2 Data Structure Documentation

6.6.2.1 struct Lcu_Ip_SyncInputValueType

This type contains the Lcu Ip Multiple Inputs and Multiple Value.

The Lcu Ip Multi Input Value contains a pair composed from Logic Input ID and the Value of configuration. The LogicInputId selects a Logic Input The Value stores the configuration's value.

Definition at line 172 of file Lcu_Ip.h.

6.6.2.2 struct Lcu_Ip_SyncOutputValueType

This type contains the Lcu Ip Multiple Outputs and Multiple Value.

The Lcu Ip Multi Output Value contains a pair composed from Logic Output ID and the Value of configuration. The LogicOutputId selects a Logic Output The Value stores the configuration's value.

Definition at line 184 of file Lcu_Ip.h.

6.6.2.3 struct Lcu_Ip_AsyncInputValueType

This type contains the Lcu Ip Input with multiple value.

The Lcu Ip Input with Multiple Value contains a pair composed from Input Parameter and Value of the Parameter. The Lcu Ip Input Param Type selects a parameter from the Lcu_Ip_InputParamType enum The Value stores the configuration's value.

Definition at line 196 of file Lcu_Ip.h.

6.6.2.4 struct Lcu_Ip_AsyncOutputValueType

This type contains the Lcu Ip Output with multiple value.

The Lcu Ip Output with Multiple Value contains a pair composed from Output Parameter and Value of the Parameter. The Lcu Ip Output Param Type selects a parameter from the Lcu_Ip_OutputParamType enum The Value stores the configuration's value.

Definition at line 208 of file Lcu_Ip.h.

6.6.3 Types Reference

6.6.3.1 Lcu_Ip_Callback

```
typedef void(* Lcu_Ip_Callback) (Lcu_Ip_Event Event)
```

This type contains the LCU IP Callback interface.

The Callback is defined by the user and installed by the driver in the corresponding IRQ.

Returns

void

Definition at line 184 of file Lcu_Ip_Types.h.

6.6.4 Enum Reference

6.6.4.1 Lcu_Ip_InputParamType

```
enum Lcu_Ip_InputParamType
```

This type contains the LCU Input Param Type.

The Parameters set specific functionalities for Input

Enumerator

LCU_IP_INPUT_SET_MUX_SEL	[MUXSEL] Input MUX Select.
LCU_IP_INPUT_SET_SW_SYNC_MODE	[SW_MODE] Specifies the software sync mode for the inputs to this LC.When Software Override is enabled (SWEN),these bits control whether Software Override Value (SWVALUE) changes occur immediately or on the rising edge of the selected sync pulse
LCU_IP_INPUT_SET_SW_OVERRIDE_EN	[SWEN] Software override input enable
LCU_IP_INPUT_SET_SW_VALUE	[SWVALUE] Software override input value

Definition at line 106 of file Lcu_Ip.h.

6.6.4.2 Lcu_Ip_OutputParamType

enum Lcu_Ip_OutputParamType

This type contains the LCU Output Param Type.

The Parameters set specific functionalities for Output

Enumerator

LCU_IP_OUTPUT_SET_EN_DEBUG_MODE	[DBGEN] Enables outputs to continue operation in Debug mode
LCU_IP_OUTPUT_SET_OUTPUT_ENABLE	[OUTEN] Enables LC outputs
LCU_IP_OUTPUT_SET_LUT_CONTROL	[LUTCTRL] LUT control
LCU_IP_OUTPUT_SET_LUT_RISE_FILTER	[LUT_RISE_FILT] LUT Rise Filter
LCU_IP_OUTPUT_SET_LUT_FALL_FILTER	[LUT_FALL_FILT] LUT Fall Filter
LCU_IP_OUTPUT_SET_EN_FORCE_DMA	[LUT_DMA_EN] Enables the generation of a DMA request when an LUT event occurs
LCU_IP_OUTPUT_SET_EN_LUT_DMA	[FORCE_DMA_EN] Enables the generation of a DMA request when a force event occurs
LCU_IP_OUTPUT_SET_EN_FORCE_INT	[LUT_INT_EN] Enables the generation of an interrupt request when an LUT event
LCU_IP_OUTPUT_SET_EN_LUT_INT	[FORCE_INT_EN] Enables the generation of an interrupt request when a force event occurs
LCU_IP_OUTPUT_SET_INVERT_OUTPUT	[OUTPOL] Set Output Polarity: invert or not.
LCU_IP_OUTPUT_SET_FORCE_SIGNAL_SEL	[FORCE_SENSE] Select Force signal
LCU_IP_OUTPUT_SET_CLEAR_FORCE_↩ MODE	[FORCE_MODE] Force Clearing Mode
LCU_IP_OUTPUT_SET_FORCE_SYNC_SEL	[SYNC_SEL] The Force Sync Select Parameter specifies which sync input to use for this output
LCU_IP_OUTPUT_CLEAR_FORCE_STS	[FORCESTS] Clear force event in STS

Definition at line 119 of file Lcu_Ip.h.

6.6.4.3 Lcu_Ip_LogicInputInfoParamType

enum Lcu_Ip_LogicInputInfoParamType

This type contains the LCU Param Type to get information for input.

The Parameters get specific functionalities for Input

Definition at line 142 of file Lcu_Ip.h.

6.6.4.4 Lcu_Ip_LogicOutputInfoParamType

```
enum Lcu_Ip_LogicOutputInfoParamType
```

This type contains the LCU Param Type to get information for output.

The Parameters get specific functionalities for Output

Definition at line 153 of file Lcu_Ip.h.

6.6.4.5 Lcu_Ip_Event

```
enum Lcu_Ip_Event
```

This type contains the LCU IP Event.

The Event Type give information for interrupt event is LUT or FORCE.

Definition at line 155 of file Lcu_Ip_Types.h.

6.6.4.6 Lcu_Ip_ReturnType

```
enum Lcu_Ip_ReturnType
```

This type contains the LCU IP Return Type.

The Return Type give information for the execution of interfaces.

Definition at line 166 of file Lcu_Ip_Types.h.

6.6.5 Function Reference

6.6.5.1 Lcu_Ip_Init()

```
Lcu_Ip_ReturnType Lcu_Ip_Init (
            const Lcu_Ip_InitType *const pxLcuInit )
```

This function initializes the Lcu Ip Driver.

This service is a non reentrant function that shall initialize the Lcu Ip driver. When enable multi cores, the API shall write the values into the register only if the reference partition is same with executing core

Parameters

in	*pxLcuInit*	Pointer to the configuration structure.

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and the reference partition is same with executing core. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.2 Lcu_Ip_Deinit()

```
Lcu_Ip_ReturnType Lcu_Ip_Deinit (
            void  )
```

This function De-initializes the Lcu Ip Driver.

This service is a non reentrant function that shall De-initialize the Lcu Ip driver. When enable multi cores, the API shall write the values into the register only if the reference partition is same with executing core

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and the reference partition is same with executing core. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.3 Lcu_Ip_SetWriteProtect()

```
Lcu_Ip_ReturnType Lcu_Ip_SetWriteProtect (
            const uint8 LogicInstance )
```

[WP] This function Enable Write Protect feature for the Logic Instance.

This service is a reentrant function that shall turns on write protection for all LCU registers except SWVALUE, LCn_STS, and FORCEST.

Parameters

in	*LogicInstance*	Specifies the Logic Instance.Using define generated by the configurator (Example : LCU_LOGIC_INSTANCE_0)

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_SUCCESS is returned if the write protect was enabled successfully. LCU_IP_STATUS_PROTECTED is returned if the write protect was already enable before

6.6.5.4 Lcu_Ip_GetWriteProtect()

```
Lcu_Ip_ReturnType Lcu_Ip_GetWriteProtect (
            const uint8 LogicInstance )
```

[WP] This function get Write Protect status for the Logic Instance

This service is a reentrant function that shall return Write Protect Status.

Parameters

in	*LogicInstance*	Specifies the Logic Instance.Using define generated by the configurator (Example : LCU_LOGIC_INSTANCE_0)

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_PROTECTED is returned if the write protect was already enabled before. LCU_IP_STATUS_SUCCESS is returned if the write protect is disabled. In this case user can call Lcu_Ip_SetWriteProtect to enable Write Protect feature.

6.6.5.5 Lcu_Ip_SetSyncInputSwOverrideEnable()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwOverrideEnable (
            const Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[SWEN] This function Enable/Disable software override of LC input.

This service is a reentrant function that shall Enable/Disable software override of LC input. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_OVERRIDE_DISABLE/LCU_IP_SW_OVERRIDE↩ _ENABLE

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Input in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.6 Lcu_Ip_SetSyncInputSwOverrideValue()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwOverrideValue (
            const Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[SWVALUE] This function specifies the software override value for each LC input.

This service is a reentrant function that shall specifies the software override value for each LC input. When enable multi cores, the API shall write the values into the register only if the list contains all the Inputs related to the same partition

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_OVERRIDE_LOGIC_LOW/LCU_IP_SW_↩ OVERRIDE_LOGIC_HIGH

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Input in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.7 Lcu_Ip_SetSyncInputMuxSelect()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputMuxSelect (
            const Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[MUXSEL] This function selects the source of the LC input.

This service is a reentrant function that shall selects the source of the LC input

```
How to use this interface:
List[x].LogicInputId -> The Logic Input Id generated by the configurator
List[x].Value -> Unsigned Integer: [0, 255]
```

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Input in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.8 Lcu_Ip_SetSyncInputSwSyncMode()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncInputSwSyncMode (
            const Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[SW_MODE] specifies the software sync mode for the inputs to this LC.

This service is a reentrant function that shall specifies the software sync mode for the inputs to this LC.When Software Override is enabled (SWEN), these bits control whether Software Override Value (SWVALUE) changes occur immediately or on the rising edge of the selected sync pulse

How to use this interface: List[x].LogicInputId -> The Logic Input Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_SW_SYNC_IMMEDIATE/LCU_IP_SW_SYNC_ON_↩ RISING_EDGE

Parameters

in	*List*	Specifies the input configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Input in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.9 Lcu_Ip_SetSyncOutputDebugMode()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputDebugMode (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[DBGEN] This function Enables/Disables outputs to continue operation in Debug mode.

This service is a reentrant function that shall Enables/Disables outputs to continue operation in Debug mode. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DEBUG_DISABLE/LCU_IP_DEBUG_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.10 Lcu_Ip_SetSyncOutputEnable()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputEnable (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[OUTEN] This function Enables/Disables LC outputs.

This service is a reentrant function that shall Enables/Disables LC outputs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_OUTPUT_DISABLE/LCU_IP_OUTPUT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.11 Lcu_Ip_SetSyncOutputForceInputSensitivity()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceInputSensitivity (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_SENSE] This function specifies which force inputs affect output.

This service is a reentrant function that shall specifies which force inputs affect output. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 255] For each bit: 0b - Does not affect 1b - Affects .* Example: 011b specifies: force inputs 0 and 1 will affect for this logic output, .* but force inputs 2 will not affect

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.12 Lcu_Ip_SetSyncOutputForceClearingMode()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceClearingMode (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_MODE] This function specifies the timing for clearing force events for output.

This service is a reentrant function that shall specifies the timing for clearing force events for output. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_CLEAR_FORCE_SIGNAL_IMMEDIATE/ LCU_IP↩ _CLEAR_FORCE_SIGNAL_ON_RISING_EDGE/ LCU_IP_CLEAR_FORCE_SIGNAL_AFTER_CLEAR↩ _STATUS/ LCU_IP_CLEAR_FORCE_SIGNAL_ON_RISING_EDGE_AFTER_CLEAR_STATUS

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.13 Lcu_Ip_SetSyncOutputForceSyncSelect()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceSyncSelect (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[SYNC_SEL] This function specifies which sync input to use for Force signal.

This service is a reentrant function that shall specifies which sync input to use for Force signal. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 255] 00b - Sync input 0 01b - Sync input 1

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.14 Lcu_Ip_SetSyncOutputPolarity()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputPolarity (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[OUTPOL] This function specifies the polarity of the outputs.

This service is a reentrant function that shall specifies the polarity of the outputs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_FORCE_POL_NOT_INVERTED/LCU_IP_FORCE↩ _POL_INVERTED

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.15 Lcu_Ip_SetSyncOutputForceDma()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceDma (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_DMA_EN] This function Enables/Disable the generation of a DMA request when a force event occurs.

This service is a reentrant function that shall Enables/Disable the generation of a DMA request when a force event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DMA_DISABLE/LCU_IP_DMA_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.16 Lcu_Ip_SetSyncOutputForceInt()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputForceInt (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCE_INT_EN] This function Enables/Disable the generation of an interrupt request when force event occurs.

This service is a reentrant function that shall Enables/Disable the generation of an interrupt request when force event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_INT_DISABLE/LCU_IP_INT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.17 Lcu_Ip_SetSyncOutputLutDma()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutDma (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_DMA_EN] This function Enables/Disable the generation of a DMA request when a LUT event occurs.

This service is a reentrant function that shall Enables/Disable the generation of a DMA request when a LUT event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_DMA_DISABLE/LCU_IP_DMA_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.18 Lcu_Ip_SetSyncOutputLutInt()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutInt (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_INT_EN] This function Enables/Disable the generation of an interrupt request when LUT event occurs.

This service is a reentrant function that shall Enables/Disable the generation of an interrupt request when LUT event occurs. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Using define generated by the configurator : LCU_IP_INT_DISABLE/LCU_IP_INT_ENABLE

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.19 Lcu_Ip_SetSyncOutputFallFilter()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputFallFilter (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_FALL_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low

This service is a reentrant function that shall specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.20 Lcu_Ip_SetSyncOutputRiseFilter()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputRiseFilter (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LUT_RISE_FILT] This function specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high

This service is a reentrant function that shall Specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.21 Lcu_Ip_SetSyncOutputLutControl()

```
Lcu_Ip_ReturnType Lcu_Ip_SetSyncOutputLutControl (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LUTCTRL] This function specifies the LUT positions, based on the combined LC input value

This service is a reentrant function that shall Specifies the LUT positions, based on the combined LC input value. When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator List[x].Value -> Unsigned Integer: [0, 65535]

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.22 Lcu_Ip_ClearSyncOutputForceEvent()

```
Lcu_Ip_ReturnType Lcu_Ip_ClearSyncOutputForceEvent (
            const Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCESTS] This function Clear Force Event

This service is a reentrant function that shall Clear Force Event When enable multi cores, the API shall write the values into the register only if the list contains all the Outputs related to the same partition

How to use this interface: List[x].LogicOutputId -> The Logic Output Id generated by the configurator

Parameters

in	*List*	Specifies the Output configuration.
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.23 Lcu_Ip_GetSyncLogicInput()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncLogicInput (
            Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[LC_INPUTS] This function Indicates states of LC inputs

This service is a reentrant function that shall Indicates states of LC inputs When enable multi cores, the API shall get the values from the register only if the list contains all the Inputs related to the same partition

How to use this interface: [in] List[x].LogicInputId -> The Logic Input Id generated by the configurator [out] List[x].Value -> Store the states of LC inputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.24 Lcu_Ip_GetSyncSwOverrideInput()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncSwOverrideInput (
            Lcu_Ip_SyncInputValueType List[],
            const uint8 Dimension )
```

[SWOUT] This function Indicates states of LC inputs or software-overridden inputs

This service is a reentrant function that shall Indicates states of LC inputs or software-overridden inputs When enable multi cores, the API shall get the values from the register only if the list contains all the Inputs related to the same partition

How to use this interface: [in] List[x].LogicInputId -> The Logic Input Id generated by the configurator [out] List[x].Value -> Store the states of LC inputs or software-overridden inputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.25 Lcu_Ip_GetSyncLogicOutput()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncLogicOutput (
            Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[LCOUT] This function Indicates states of LC outputs

This service is a reentrant function that shall Indicates states of LC outputs When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the states of LC outputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.26 Lcu_Ip_GetSyncForceOutput()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncForceOutput (
            Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCEOUT] This function Indicates the current state of force outputs for the logic outputs

This service is a reentrant function that shall the current state of force outputs for the logic outputs When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the current state of force outputs for the logic outputs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.27 Lcu_Ip_GetSyncForceStatus()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncForceStatus (
            Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[FORCESTS] This function Indicates that a force event has occurred on the associated output

This service is a reentrant function that shall Indicates that a force event has occurred on the associated output When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the force event value. Value = 0: force event not occurs Value = 1: force event occurs

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.28 Lcu_Ip_GetSyncCombineForceInput()

```
Lcu_Ip_ReturnType Lcu_Ip_GetSyncCombineForceInput (
            Lcu_Ip_SyncOutputValueType List[],
            const uint8 Dimension )
```

[COMB_FORCE] This function Indicates the combined value of force inputs to each output

This service is a reentrant function that shall Indicates the combined value of force inputs to each output When enable multi cores, the API shall get the values from the register only if the list contains all the Outputs related to the same partition

How to use this interface: [in] List[x].LogicOutputId -> The Logic Output Id generated by the configurator [out] List[x].Value -> Store the combined value of force inputs to each output

Parameters

	[in/out]	List Pointer to the configuration structure
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and at least one Output in the List has the incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.29 Lcu_Ip_SetAsyncInputList()

```
Lcu_Ip_ReturnType Lcu_Ip_SetAsyncInputList (
            const uint8 LogicInput,
            const Lcu_Ip_AsyncInputValueType List[],
            const uint8 Dimension )
```

This function configure multiple configuration for one logic input.

This service is a reentrant function that shall configure multiple configuration for one logic input

How to use this interface: List[x].Param -> Select member of Lcu_Ip_InputParamType List[x].Value -> Unsigned Integer: [0, 255] or some defines generated by configurator base on Param

Parameters

in	*LogicInput*	Logic input Id
in	*List*	List of param and value
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and the LogicInput incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.30 Lcu_Ip_SetAsyncOutputList()

```
Lcu_Ip_ReturnType Lcu_Ip_SetAsyncOutputList (
            const uint8 LogicOutput,
```

```
            const Lcu_Ip_AsyncOutputValueType List[],
            const uint8 Dimension )
```

This function configure multiple configuration for one logic output.

This service is a reentrant function that shall configure multiple configuration for one logic output

How to use this interface: List[x].Param -> Select member of Lcu_Ip_InputParamType List[x].Value -> Unsigned Integer: [0, 255] or some defines generated by configurator base on Param

Parameters

in	*LogicOutput*	Logic output Id
in	*List*	List of param and value
in	*Dimension*	Number of entries in the List

Returns

Lcu_Ip_ReturnType LCU_IP_STATUS_WRONG_CORE is returned if enable the multi-cores feature and the LogicInput incorrect partition. LCU_IP_STATUS_SUCCESS is returned in the remaining cases

6.6.5.31 Lcu_Ip_GetAsyncLogicInputInfo()

```
Lcu_Ip_ReturnType Lcu_Ip_GetAsyncLogicInputInfo (
            const uint8 LogicInput,
            const Lcu_Ip_LogicInputInfoParamType Param,
            uint8 *const Value )
```

This function get information of Logic Input.

This service is a reentrant function that shall get states of Logic Inputs How to use this interface: [in] LogicInput -> The Logic Input Name generated by the configurator. By default: LCU_LOGIC_INPUT_0, LCU_LOGIC_↩ INPUT_1,... [in] Param -> Select parameter. Example: LCU_IP_INPUT_GET_LOGIC_INPUT_STATE [out] Value -> Store the states of LC inputs

Parameters

in	*LogicInput*	The Logic Input Name
in	*Param*	Selection parameter
out	*Value*	Output value

Returns

void

6.6.5.32 Lcu_Ip_GetAsyncLogicOutputInfo()

```
Lcu_Ip_ReturnType Lcu_Ip_GetAsyncLogicOutputInfo (
            const uint8 LogicOutput,
            const Lcu_Ip_LogicOutputInfoParamType Param,
            uint8 *const Value )
```

This function get information of Logic Output.

This service is a reentrant function that shall get states of Logic Outputs How to use this interface: [in] LogicOutput -> The Logic Output Name generated by the configurator. By default: LCU_LOGIC_OUTPUT_0, LCU_LOGIC↩ _OUTPUT_1,... [in] Param -> Select parameter. Example: LCU_IP_OUTPUT_GET_LOGIC_OUTPUT_↩ STATE [out] Value -> Store the states of LC inputs

Parameters

in	*LogicOutput*	The Logic Output Name
in	*Param*	Selection parameter
out	*Value*	Output value

Returns

void

6.7 TRGMUX IP Driver

6.7.1 Detailed Description

Function Reference

- Trgmux_Ip_StatusType Trgmux_Ip_Init (const Trgmux_Ip_InitType *const pxTrgmuxInit)
  This function initializes the Trgmux Ip Driver.
- Trgmux_Ip_StatusType Trgmux_Ip_SetInput (const uint32 LogicTrigger, const uint32 Input)
  This function sets the Trgmux Ip Input.
- Trgmux_Ip_StatusType Trgmux_Ip_SetLock (const uint32 LogicTrigger)
  This function locks the Trgmux Ip Trigger.

6.7.2 Function Reference

6.7.2.1 Trgmux_Ip_Init()

```
Trgmux_Ip_StatusType Trgmux_Ip_Init (
            const Trgmux_Ip_InitType *const pxTrgmuxInit )
```

This function initializes the Trgmux Ip Driver.

This service is a non reentrant function that shall initialize the Trgmux Ip driver.

Parameters

in	*pxTrgmuxInit*	Pointer to the initialization structure.

Returns

TRGMUX_IP_STATUS_LOCKED The trigger is locked. TRGMUX_IP_STATUS_SUCCESS The initialization was successful.

6.7.2.2 Trgmux_Ip_SetInput()

```
Trgmux_Ip_StatusType Trgmux_Ip_SetInput (
            const uint32 LogicTrigger,
            const uint32 Input )
```

This function sets the Trgmux Ip Input.

This service is a reentrant function that shall set the Trgmux Ip Input.

Parameters

in	*LogicTrigger*	The logic trigger id.
in	*Input*	The input value.

Returns

TRGMUX_IP_STATUS_LOCKED The trigger is locked. TRGMUX_IP_STATUS_SUCCESS The initialization was successful.

6.7.2.3 Trgmux_Ip_SetLock()

```
Trgmux_Ip_StatusType Trgmux_Ip_SetLock (
            const uint32 LogicTrigger )
```

This function locks the Trgmux Ip Trigger.

This service is a reentrant function that shall lock the Trgmux Ip Trigger.

Parameters

in	*LogicTrigger*	The logic trigger id.

Returns

TRGMUX_IP_STATUS_LOCKED The trigger is locked. TRGMUX_IP_STATUS_SUCCESS The initialization was successful.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support