

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | *LLWEB-20133817* | | |
| Draft | | |
| 0.07M | | |
|  | *Unit Design Specification*  for IMP Driver | March 19, 2024 | | |
| Automotive Software Development Division  Automotive Software Platform Department | | |
| Approver | Checker | Author |
| <approver> |  | KSK |

**Table of Contents**

[1 Introduction 3](#_Toc135903269)

[1.1 Scope 3](#_Toc135903270)

[1.2 References 5](#_Toc135903271)

[1.3 Abbreviation 5](#_Toc135903272)

[1.4 Terminology 5](#_Toc135903273)

[2 Software Unit Design 6](#_Toc135903274)

[2.1 Functions of API Layer 6](#_Toc135903275)

[2.1.1 Public function 7](#_Toc135903276)

[*2.1.2* Private function 34](#_Toc135903277)

[2.2 Functions of General Control 41](#_Toc135903278)

[2.2.1 Public function 42](#_Toc135903279)

[2.2.2 Private function 71](#_Toc135903280)

[2.3 Functions of Common Control 82](#_Toc135903281)

[2.3.1 Public function 84](#_Toc135903282)

[*2.3.2* Private function 113](#_Toc135903283)

[2.4 Functions of Core Control 138](#_Toc135903284)

[2.4.1 Public function 140](#_Toc135903285)

[*2.4.2* Private function 181](#_Toc135903286)

[2.5 Functions of OS Dependence Layer 233](#_Toc135903287)

[*2.5.1* Public function 235](#_Toc135903288)

[2.5.2 Private function 277](#_Toc135903289)

[3 Activity Diagrams 291](#_Toc135903290)

[4 Resource Usage 292](#_Toc135903291)

[5 Data Design 293](#_Toc135903292)

[5.1 Data Type Definitions 293](#_Toc135903293)

[5.1.1 Typedef 293](#_Toc135903294)

[5.1.2 Structure 296](#_Toc135903295)

[5.2 Global Pointer Variables 326](#_Toc135903296)

[5.3 Global Variables 326](#_Toc135903297)

[5.3.1 API Layer 326](#_Toc135903298)

[5.3.2 General Control 326](#_Toc135903299)

[5.3.3 Common Control 326](#_Toc135903300)

[5.3.4 Core Control 328](#_Toc135903301)

[5.3.5 OS Dependence Layer 328](#_Toc135903302)

[5.4 Constants 329](#_Toc135903303)

[5.4.1 Definition 329](#_Toc135903304)

[5.4.2 Enum 354](#_Toc135903305)

[5.4.3 Constants Table 366](#_Toc135903306)

[6 Device Specific definition 388](#_Toc135903307)

# Introduction

## Scope

This document explains about the Component Unit Design of IMP Driver.

To enable the coding of each unit of IMP Driver, the following is defined.

- The processing flow of each unit

- etc.

The arrow connects the caller to the callee.

<<Public>> indicates a function called from another module.

<<Private>> indicates a function not called from another module.

グラフィカル ユーザー インターフェイス, PowerPoint

自動的に生成された説明

**Figure 1‑1 Target of this document**

グラフィカル ユーザー インターフェイス, アプリケーション, PowerPoint

自動的に生成された説明

Figure 1‑2 Target of this document

## References

|  |  |  |
| --- | --- | --- |
| **No.** | **Title (Document Name)** | **Version** |
|  | Software Architecture Design Specification for IMP Driver | *0.16E* |
|  | Component Architecture Design Specification for IMP Driver | *0.13E* |
|  | IMP Driver User’s Manual | *0.04E* |
|  | R-Car V3x IMP Driver Product Information | *V3M: 2.00E*  *V3H: 2.00E*  *V3H\_2: 2.00E* |
|  | Operating System Abstraction Layer API (OSAL API) User’s Manual | *3.11E* |
|  | xOS3-IMPIMRPAP\_CommonInfo.xlsx | *1.00M* |
|  | Operating System Abstraction Layer (OSAL) API Application Note | *3.12E* |
|  | IMP Framework User’s Manual | *0.20E* |

## Abbreviation

Refer to the “Abbreviation definition sheet” in [6]

## Terminology

Refer to the “Terminology definition sheet” in [6]

# Software Unit Design

The following shows the function detail of IMP Driver.

This design does not use the C standard library. However, in order to use the predefined identifier, it is necessary to include “stddef.h” and “stdint.h”.

The description of traceability ID is given to each unit.

## Functions of API Layer

Table 2‑1: Function List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Function Name** | **Access** | **ASIL** | **Source File Name** |
|  | R\_IMPDRV\_Init | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_Quit | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrInit | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrSetCoreMemInit | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_Start | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_Stop | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrSetCoreMap | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrSetCl | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrSetIrqMask | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_SetIrqGroup | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_AttrSetClBrkAddr | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_Execute | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_ResumeExecution | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_SetPMPolicy | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_GetPMPolicy | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_GetPMState | Public | ASIL D | r\_impdrv\_api.c |
|  | R\_IMPDRV\_GetVersion | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_RegGetHwInfo](#_R_IMPDRV_RegGetHwInfo) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_RegRead32](#_R_IMPDRV_RegRead32) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_RegWrite32](#_R_IMPDRV_RegWrite32) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_RegRequired](#_R_IMPDRV_RegRequired) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_SetMemProtect](#_R_IMPDRV_SetMemProtect) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_BusIfCheck](#_R_IMPDRV_BusIfCheck) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_ConfRegCheck](#_R_IMPDRV_ConfRegCheck) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_ModuleStopCheck](#_R_IMPDRV_ModuleStopCheck) | Public | ASIL D | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_AttrSetDsp](#_R_IMPDRV_AttrSetDsp) | Public | - | r\_impdrv\_api.c |
|  | [R\_IMPDRV\_AttrSetGosubCond](#_R_IMPDRV_AttrSetGosubCond) | Public | ASIL D | r\_impdrv\_api.c |
|  | impdrv\_api\_chk\_init\_data | Private | ASIL D | r\_impdrv\_api.c |
|  | impdrv\_api\_chk\_core\_info | Private | ASIL D | r\_impdrv\_api.c |
|  | impdrv\_api\_chk\_instance\_num | Private | ASIL D | r\_impdrv\_api.c |
|  | impdrv\_api\_cnv\_error\_code | Private | ASIL D | r\_impdrv\_api.c |
|  | impdrv\_api\_cnv\_ctrl\_handle | Private | ASIL D | r\_impdrv\_api.c |
|  | [impdrv\_api\_chk\_bus\_if\_check](#_i_mpdrv_api_chk_bus_if_check) | Private | ASIL D | r\_impdrv\_api.c |
|  | [impdrv\_api\_chk\_conf\_reg\_check](#_impdrv_api_chk_conf_reg_check) | Private | ASIL D | r\_impdrv\_api.c |

### Public function

#### R\_IMPDRV\_Init

Table 2‑2: R\_IMPDRV\_Init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0001  R\_IMPDRV\_Init  [Covers: AD\_PD\_CAS0105, AD\_PD\_CAS1001, AD\_PD\_CAS2001, AD\_PD\_CAS2002, AD\_PD\_CAS3001, AD\_PD\_CAS2238] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_Init (  st\_impdrv\_initdata\_t \*const p\_initdata,  impdrv\_ctrl\_handle\_t \*const p\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-Out)** | st\_impdrv\_initdata\_t \*const p\_initdata | | Initialization data for IMP Driver  The lifetime of this parameter is until this function returns.  Refer st\_impdrv\_initdata\_t for the expiration of the members of the structure. | |
| **Range** | Not NULL |
| **Parameters (Out)** | impdrv\_ctrl\_handle\_t \*const p\_handle | | Control handle for IMP Driver  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.1 R\_IMPDRV\_Init” in [3]. | | | |
| **Preconditions** | Can be executed in the “Uninitialized State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_Quit

Table 2‑3 R\_IMPDRV\_Quit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0002  R\_IMPDRV\_Quit  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS2067, AD\_PD\_CAS2068, AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_Quit (  impdrv\_ctrl\_handle\_t handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.2 R\_IMPDRV\_Quit” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State” or “Uninitialized State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrInit

Table 2‑4 R\_IMPDRV\_AttrInit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0003  R\_IMPDRV\_AttrInit  [Covers: AD\_PD\_CAS1007, AD\_PD\_CAS2115, AD\_PD\_CAS2116, AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrInit (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information  The lifetime of this parameter is until this function returns. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | const impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.3 R\_IMPDRV\_AttrInit” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrSetCoreMemInit

Table 2‑5 R\_IMPDRV\_AttrSetCoreMemInit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0004  R\_IMPDRV\_AttrSetCoreMemInit  [Covers: AD\_PD\_CAS1005, AD\_PD\_CAS1009, AD\_PD\_CAS2139, AD\_PD\_CAS2365] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetCoreMemInit (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const e\_impdrv\_param\_t enable  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const e\_impdrv\_param\_t enable | | Core memory initialization selection | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.4 R\_IMPDRV\_AttrSetCoreMemInit” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_Start

Table 2‑6 R\_IMPDRV\_Start

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0005  R\_IMPDRV\_Start  [Covers: AD\_PD\_CAS1005, AD\_PD\_CAS2231, AD\_PD\_CAS2232] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_Start (  impdrv\_ctrl\_handle\_t handle ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.5 R\_IMPDRV\_Start” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_Stop

Table 2‑7 R\_IMPDRV\_Stop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0006  R\_IMPDRV\_Stop  [Covers: AD\_PD\_CAS1006, AD\_PD\_CAS2266, AD\_PD\_CAS2267] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_Stop (  impdrv\_ctrl\_handle\_t handle ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.6 R\_IMPDRV\_Stop” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrSetCoreMap

Table 2‑8 R\_IMPDRV\_AttrSetCoreMap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0007  R\_IMPDRV\_AttrSetCoreMap  [Covers: AD\_PD\_CAS1008,AD\_PD\_CAS1009, AD\_PD\_CAS2156, AD\_PD\_CAS2370] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetCoreMap (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const st\_impdrv\_core\_info\_t core\_map [IMPDRV\_COREMAP\_MAXID]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t core\_map | | Sync core map array | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “ 5.2.1.7 R\_IMPDRV\_AttrSetCoreMap” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ” or “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrSetCl

Table 2‑9 R\_IMPDRV\_AttrSetCl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0008  R\_IMPDRV\_AttrSetCl  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2175, AD\_PD\_CAS2376] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetCl (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const uintptr\_t claddr\_phys  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const uintptr\_t claddr\_phys | | Physical memory address of CL data | |
| **Range** | 128byte alignments |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.8 R\_IMPDRV\_AttrSetCl” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ” or “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

[impdrv\_impsctl\_execute](file:///C:\\Users\\a5149970\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.MSO\\CC66D94F.xlsx" \l "'D_対象CoreControl(一覧)'!E244" \t "_parent)

#### R\_IMPDRV\_AttrSetIrqMask

Table 2‑10 R\_IMPDRV\_AttrSetIrqMask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0009  R\_IMPDRV\_AttrSetIrqMask  [Covers: AD\_PD\_CAS1008, AD\_PD\_CAS1009, AD\_PD\_CAS2192, AD\_PD\_CAS2381] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetIrqMask (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const bool irq\_mask[IMPDRV\_IRQMASK\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const bool irq\_mask[IMPDRV\_IRQMASK\_MAX] | | Array for IRQ mask setting | |
| **Range** | true or false |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.10 R\_IMPDRV\_AttrSetIrqMask” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ” or “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_SetIrqGroup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0011  R\_IMPDRV\_SetIrqGroup  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2209, AD\_PD\_CAS2386] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_SetIrqGroup (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_irq\_group\_t \*const p\_irq\_param  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_irq\_group\_t \*const p\_irq\_param | | Interrupt group to which this Control handle | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.11 R\_IMPDRV\_SetIrqGroup” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ” or “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

Table 2‑11: R\_IMPDRV\_SetIrqGroup

#### R\_IMPDRV\_AttrSetClBrkAddr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0036  R\_IMPDRV\_AttrSetClBrkAddr  [Covers: AD\_PD\_CAS2537, AD\_PD\_CAS2538] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_ AttrSetClBrkAddr (  impdrv\_ctrl\_handle\_t handle ,  const st\_impdrv\_core\_info\_t \* const p\_core\_info,  const uintptr\_t cl\_brk\_addr  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| uintptr\_t cl\_brk\_addr | | CL break point physical address | |
| **Range** | 4byte alignments |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | N/A  Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.12 R\_IMPDRV\_AttrSetClBreak” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** |  | | | |

Table 2‑12 R\_IMPDRV\_AttrSetClBrkAddr

#### R\_IMPDRV\_Execute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0012  R\_IMPDRV\_Execute  [Covers: AD\_PD\_CAS1010, AD\_PD\_CAS2292, AD\_PD\_CAS2293] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_Execute (  impdrv\_ctrl\_handle\_t handle ,  const st\_impdrv\_core\_info\_t \* const p\_core\_info  const p\_impdrv\_cbfunc\_t callback\_func,  void \*const p\_callback\_args  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const p\_impdrv\_cbfunc\_t callback\_func | | Callback function to IMP Framework | |
| **Range** | Not NULL |
| void  \*const p\_callback\_args | | Argument of callback function | |
| **Range** | None |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ATTRIBUTE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.12 R\_IMPDRV\_Execute” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

Table 2‑13: R\_IMPDRV\_Execute

#### R\_IMPDRV\_ResumeExecution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0013  R\_IMPDRV\_ResumeExecution  [Covers: AD\_PD\_CAS1014, AD\_PD\_CAS2393, AD\_PD\_CAS2394] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_ResumeExecution (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.13 R\_IMPDRV\_ResumeExecution” in [3]. | | | |
| **Preconditions** | Can be executed in the “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

Table 2‑14: R\_IMPDRV\_ResumeExecution

#### R\_IMPDRV\_SetPMPolicy

Table 2‑15 : R\_IMPDRV\_SetPMPolicy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0018  R\_IMPDRV\_SetPMPolicy  [Covers: AD\_PD\_CAS1015, AD\_PD\_CAS3001, AD\_PD\_CAS2419, AD\_PD\_CAS2420] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_SetPMPolicy (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const e\_impdrv\_pm\_policy\_t policy  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| const e\_impdrv\_pm\_policy\_t policy | | Power management policy | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.2.1 R\_IMPDRV\_SetPMPolicy” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ”, “Ready State”, “ Active State” and “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_GetPMPolicy

Table 2‑16: R\_IMPDRV\_GetPMPolicy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0019  R\_IMPDRV\_GetPMPolicy  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1016, AD\_PD\_CAS2451, AD\_PD\_CAS2452] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_GetPMPolicy (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  e\_impdrv\_pm\_policy\_t \*const policy  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (Out)** | e\_impdrv\_pm\_policy\_t \*const policy | | Power management policy | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.2.2 R\_IMPDRV\_GetPMPolicy” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State ”, “Ready State”, “ Active State” and “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | policy is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### R\_IMPDRV\_GetPMState

Table 2‑17: R\_IMPDRV\_GetPMState

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0038  R\_IMPDRV\_GetPMState  [Covers: AD\_PD\_CAS1032, AD\_PD\_CAS2097, AD\_PD\_CAS2098] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_GetPMState(  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  e\_impdrv\_pm\_state\_t \*const p\_pmstate  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | e\_impdrv\_pm\_state\_t \*const p\_pmstate | | Power management state | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.2.3 R\_IMPDRV\_GetPMState” in [3]. | | | |
| **Preconditions** | Can be executed in the “Initialized State”, “Ready State”, “Active State” and “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | Power management state is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### R\_IMPDRV\_GetVersion

Table 2‑18: R\_IMPDRV\_GetVersion

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0020  R\_IMPDRV\_GetVersion  [Covers: AD\_PD\_CAS1013, AD\_PD\_CAS2392, AD\_PD\_CAS2480] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | const st\_impdrv\_version\_t\* R\_IMPDRV\_GetVersion (  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Reentrant | | |
| **Interrupt State** | Can be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-Out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | st\_impdrv\_version\_t | | IMP driver version information, classified by major, minor, and patch versions.  Refer to 5.1.2.1.6 |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.18 R\_IMPDRV\_GetVersion” in [3]. | | |
| **Preconditions** | Can be executed in the “Uninitialized State”, “Initialized State ”, “Ready State”, “ Active State” and “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | |
| **Remarks** | N/A | | |

#### R\_IMPDRV\_RegGetHwInfo

Table 2‑19: R\_IMPDRV\_RegGetHwInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0026  R\_IMPDRV\_RegGetHwInfo  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1018, AD\_PD\_CAS2104, AD\_PD\_CAS2105] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_RegGetHwInfo (  impdrv\_ctrl\_handle\_t handle,  const char \*const p\_device\_id,  [st\_impdrv\_reg\_info\_t](#_st_impdrv_reg_info_t) \*const p\_reg\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const p\_device\_id | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | [st\_impdrv\_reg\_info\_t](#_st_impdrv_reg_info_t) \*const p\_reg\_info | | Pointer to Hardware register area specification information. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_RegGetHwInfo ” in [3]. | | | |
| **Preconditions** | Can be executed in the “Uninitialized state ”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_RegRead32

Table 2‑20: R\_IMPDRV\_RegRead32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0027  R\_IMPDRV\_RegRead32  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1019, AD\_PD\_CAS2111, AD\_PD\_CAS2112] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_RegRead32 (  impdrv\_ctrl\_handle\_t handle,  const char \*const p\_device\_id,  const uintptr\_t offset,  uint32\_t \*const p\_data  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const p\_device\_id | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| const uintptr\_t offset | | Sets the offset address from Register top. | |
| **Range** | The offset  address should be specified as 4-byte alignment. |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver. | |
| **Range** | Not NULL |
| **Parameters (Out)** | uint32\_t \*const p\_data | | Pointer to the read data to hardware registers. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_RegRead32” in [3]. | | | |
| **Preconditions** | Can be executed in the “Uninitialized state ”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | p\_data is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### R\_IMPDRV\_RegWrite32

Table 2‑21: R\_IMPDRV\_RegWrite32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0028  R\_IMPDRV\_RegWrite32  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1020, AD\_PD\_CAS2122, AD\_PD\_CAS2123] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_RegWrite32 (  impdrv\_ctrl\_handle\_t handle,  const char \*const p\_device\_id,  const uintptr\_t offset,  const uint32\_t data  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const p\_device\_id | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| const uintptr\_t offset | | Sets the offset address from Register top. | |
| **Range** | The offset  address should be specified as 4-byte alignment. |
| Uint32\_t data | | Data written to hardware registers. | |
| **Range** | None |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_RegWrite32” in [3]. | | | |
| **Preconditions** | Can be executed in the “Uninitialized state ”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_RegRequired

Table 2‑22: R\_IMPDRV\_RegRequired

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0029  R\_IMPDRV\_RegRequired  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1021, AD\_PD\_CAS2124, AD\_PD\_CAS2125] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_RegRequired (  impdrv\_ctrl\_handle\_t handle,  const char \*const p\_device\_id,  const [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t) new\_state  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const p\_device\_id | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| const  [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t)  new\_state | | Required state to the Register read/write function. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t  handle | | Control handle for IMP Driver. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_RegRequired” in [3]. | | | |
| **Preconditions** | Can be executed in the “Uninitialized State”, “Initialized State ”, “Ready State”, “ Active State” and “Int State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_SetMemProtect

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0030  R\_IMPDRV\_SetMemProtect  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1017, AD\_PD\_CAS2100, AD\_PD\_CAS2101] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_SetMemProtect (  impdrv\_ctrl\_handle\_t handle,  const [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t) protect\_mode  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Permitted | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const  [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t)  protect\_mode | | Memory protection setting. | | |
| **Range** | Within the range of ENUM type. | |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t  handle | | Control handle for IMP Driver. | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_SetMemProtect” in [3]. | | | | |
| **Preconditions** | Can be executed in the “Uninitialized State”, “Initialized State ”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | | |
| **Remarks** | N/A | | | | |

Table 2‑23: R\_IMPDRV\_SetMemProtect

#### R\_IMPDRV\_BusIfCheck

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0031  R\_IMPDRV\_BusIfCheck  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1025, AD\_PD\_CAS1026, AD\_PD\_CAS1027, AD\_PD\_CAS2301, AD\_PD\_CAS2302, AD\_PD\_CAS1030] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_BusIfCheck (  impdrv\_ctrl\_handle\_t handle,  const [st\_impdrv\_core\_info\_t](#_API_Layer_2) \*const p\_core\_info,  const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_API_Layer_2) \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t)  \*const \_chk\_resource | | Pointer to the OSAL resource for bus interface check. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Pointer to handle of the IMP Driver control | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_BusIfCheck” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

Table 2‑24: R\_IMPDRV\_BusIfCheck

#### R\_IMPDRV\_ConfRegCheck

Table 2‑25: R\_IMPDRV\_ConfRegCheck

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0032  R\_IMPDRV\_ConfRegCheck  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1028, AD\_PD\_CAS2303, AD\_PD\_CAS2304] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_BusIfCheck (  impdrv\_ctrl\_handle\_t handle,  const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param,  const uint32\_t param\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param | | Pointer to Array of check parameters. | |
| **Range** | Not NULL |
| const uint32\_t param\_num | | Number of check parameters. | |
| **Range** | Not 0 |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Pointer to handle of the IMP Driver control | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_ConfRegCheck” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_ModuleStopCheck

Table 2‑26: R\_IMPDRV\_ModuleStopCheck

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0033  R\_IMPDRV\_ModuleStopCheck  [Covers: AD\_PD\_CAS3001, AD\_PD\_CAS1029, AD\_PD\_CAS2305, AD\_PD\_CAS2306] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_ModuleStopCheck (  impdrv\_ctrl\_handle\_t handle,  const st\_impdrv\_chk\_param\_t \*const p\_chk\_param,  const uint32\_t param\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In-Out)** | const st\_impdrv\_chk\_param\_t \*const p\_chk\_param | | Pointer to Array of check parameters. | |
| **Range** | Not NULL |
| const uint32\_t param\_num | | Number of check parameters. | |
| **Range** | Not 0 |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| **Parameters (In)** | impdrv\_ctrl\_handle\_t handle | | Pointer to handle of the IMP Driver control | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2 R\_IMPDRV\_ModuleStopCheck” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrSetDsp

Table 2‑27: R\_IMPDRV\_AttrSetDsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0037  R\_IMPDRV\_AttrSetDsp  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2239, AD\_PD\_CAS2240] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetDsp(  impdrv\_ctrl\_handle\_t handle,  const [st\_impdrv\_core\_info\_t](#_st_impdrv_core_info_t) \*const p\_core\_info,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_app,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_fw,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_data,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_dtcm  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_st_impdrv_core_info_t) \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_app, | | Pointer to the application which run on the DSP. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_fw, | | Pointer to the DSP FW which run on the DSP. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_data, | | Pointer to the data which is used by application on the DSP. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_dtcm | | Pointer to the internal data which is used by application on the DSP. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Pointer to handle of the IMP Driver control | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.9 R\_IMPDRV\_AttrSetDsp” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

#### R\_IMPDRV\_AttrSetGosubCond

Table 2‑28: R\_IMPDRV\_AttrSetGosubCond

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0039  R\_IMPDRV\_AttrSetGosubCond  [Covers: AD\_PD\_CAS1031, AD\_PD\_CAS2241, AD\_PD\_CAS2242] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t R\_IMPDRV\_AttrSetGosubCond(  impdrv\_ctrl\_handle\_t handle,  const [st\_impdrv\_core\_info\_t](#_st_impdrv_core_info_t) \*const p\_core\_info,  const e\_impdrv\_gosub\_cond\_t condition  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_st_impdrv_core_info_t) \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_app, | | Pointer to the application which run on the DSP. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | impdrv\_ctrl\_handle\_t handle | | Pointer to handle of the IMP Driver control | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  For the usage, refer to the “5.2.1.XX R\_IMPDRV\_AttrSetGosubCond” in [3]. | | | |
| **Preconditions** | Can be executed in the “Ready State”.  For details, refer to the “3.1 Finite-State machine” in [3]. | | | |
| **Remarks** | N/A | | | |

### Private function

#### impdrv\_api\_chk\_init\_data

Table 2‑27: impdrv\_api\_chk\_init\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0021  impdrv\_api\_chk\_init\_data  [Covers: AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_chk\_init\_data (  const st\_impdrv\_initdata\_t \*const p\_initdata  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_initdata\_t  \*const p\_initdata | | Initialization data for IMP Driver  The lifetime of this parameter is until this function returns.  Refer st\_impdrv\_initdata\_t for the expiration of the members of the structure. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_chk\_core\_info

Table 2‑28: impdrv\_api\_chk\_core\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0022  impdrv\_api\_chk\_core\_info  [Covers: AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_chk\_core\_info(  const st\_impdrv\_core\_info\_t core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t  core\_info | | Core information for Specific core. | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_chk\_instance\_num

Table 2‑29: impdrv\_api\_chk\_instance\_num

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0023  impdrv\_api\_chk\_instance\_num  [Covers: AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_chk\_instance\_num(  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_cnv\_error\_code

Table 2‑30: impdrv\_api\_cnv\_error\_code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0024  impdrv\_api\_cnv\_error\_code  [Covers: AD\_PD\_CAS3001] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_cnv\_error\_code(  const e\_impdrv\_errorcode\_t ercd  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t ercd | | Error code | |
| **Range** | None |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | Error code translated for public interface,  All values defined in the enumeration. | |
| **Description** | Convert IMPDRV\_EC\_NG\_PARAM and IMPDRV\_EC\_NG\_ARGNULL returned from other subcomponents to IMPDRV\_EC\_NG\_SYSTEMERROR.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_cnv\_ctrl\_handle

Table 2‑31: impdrv\_api\_cnv\_ctrl\_handle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0025  impdrv\_api\_cnv\_ctrl\_handle  [Covers: AD\_PD\_CAS2001, AD\_PD\_CAS2067, AD\_PD\_CAS2115] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_cnv\_ctrl\_handle(  impdrv\_ctrl\_handle\_t handle,  st\_impdrv\_ctl\_t \*\*const p\_impdrv\_ctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | impdrv\_ctrl\_handle\_t handle | | Control handle for IMP Driver | |
| **Range** | None |
| **Parameters (In-Out)** | st\_impdrv\_ctl\_t \*\*const p\_impdrv\_ctl | | General control handle | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | Error code translated for public interface,  All values defined in the enumeration. | |
| **Description** | Cast the published structure to an internal structure.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_chk\_bus\_if\_check

Table 2‑32: impdrv\_api\_chk\_bus\_if\_check

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0034  impdrv\_api\_chk\_bus\_if\_check  [Covers: AD\_PD\_CAS2301] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_chk\_bus\_if\_check(  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info, | | Pointer to the Specified core information. | |
| **Range** | Not NULL |
| const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource | | Pointer to the OSAL resource for bus interface check. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | Error code translated for public interface,  All values defined in the enumeration. | |
| **Description** | Check the internal bus check resource data for IMP Driver.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_api\_chk\_conf\_reg\_check

Table 2‑33: impdrv\_api\_chk\_conf\_reg\_check

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_01\_0035  impdrv\_api\_chk\_conf\_reg\_check  [Covers: AD\_PD\_CAS2303] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_api\_chk\_conf\_reg\_check(  const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param,  const uint32\_t param\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param | | Pointer to Array of check parameters. | |
| **Range** | Not NULL |
| const uint32\_t param\_num | | Number of check parameters. | |
| **Range** | Not 0. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Convert the Input handle to Control structure pointer.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

### User define function

#### impdrv\_udefctl\_udef\_crc

Table 2‑: impdrv\_udefctl\_udef\_crc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0071***  impdrv\_udefctl\_udef\_crc  ***[Covers: AD\_PD\_CAS1005, AD\_PD\_CAS1010A, D\_PD\_CAS2600, AD\_PD\_CAS2603]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | uint32\_t impdrv\_udefctl\_udef\_crc(  uint32\_t st,  uint32\_t sz  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | N/A | | | |
| **Parameters (In)** | uint32\_t st | | Start Address. | |
| **Range** | N/A |
| uint32\_t sz | | Size of Memory (Long Word). | |
| **Range** | N/A |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | uint32\_t | | CRC value. | |
| **Description** | For the usage, refer to the “5.6.1 impdrv\_udefctl\_udef\_crc” in [8]. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_udefctl\_set\_ecm

Table 2‑196: impdrv\_udefctl\_set\_ecm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0072***  impdrv\_udefctl\_set\_ecm  ***[Covers: AD\_PD\_CAS1005, AD\_PD\_CAS1035, AD\_PD\_CAS1010, AD\_PD\_CAS2601, AD\_PD\_CAS2602, AD\_PD\_CAS2604]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | void impdrv\_udefctl\_set\_ecm(  int32\_t core\_num,  bool flag  ); | | | |
| **Sync/Async** | Asynchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | N/A | | | |
| **Parameters (In)** | int32\_t core\_num | | Specify VDSP core number. | |
| **Range** | 0 to 3. (only V4H)  0 to 1. (only V4M) |
| bool flag | | Specify control of ECM. | |
| **Range** | true or false. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | N/A | | N/A | |
| **Description** | For the usage, refer to the “5.6.2 impdrv\_udefctl\_set\_ecm” in [8]. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

## Functions of General Control

Table 2‑34: Function List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Function Name** | **Access** | **ASIL** | **Source File Name** |
|  | impdrv\_genctl\_init | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_quit | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_start | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_stop | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_cl\_brk\_addr | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_execute | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_resume\_exe | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_pm\_set\_policy | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_pm\_get\_policy | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_pm\_get\_state | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_attr\_init | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_core\_mem\_init | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_core\_map | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_cl | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_irq\_mask | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_set\_irq\_group | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_int\_handler | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_reg\_set\_mem\_protect](#_impdrv_genctl_reg_set_mem_protect) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_reg\_get\_hw\_info](#_impdrv_genctl_reg_get_hw_info) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_reg\_read32](#_impdrv_genctl_reg_read32) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_reg\_write32](#_impdrv_genctl_reg_write32) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_reg\_required](#_impdrv_genctl_reg_required) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_check\_intclear | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_bus\_if\_check](#_impdrv_genctl_bus_if_check) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_conf\_reg\_check](#_impdrv_genctl_conf_reg_check) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_set\_dsp](#_impdrv_genctl_set_dsp) | Public | - | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_dsp\_int\_handler](#_impdrv_genctl_dsp_int_handler) | Public | - | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_set\_gosub\_cond](#_impdrv_genctl_set_gosub_cond) | Public | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_chk\_core\_info](#_impdrv_genctl_chk_core_info) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_prologue](#_impdrv_genctl_prologue) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_epilogue](#_impdrv_genctl_epilogue) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_chk\_state\_init](#_impdrv_genctl_chk_state_init) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_chk\_state\_uninit](#_impdrv_genctl_chk_state_uninit) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_set\_state\_uninit](#_impdrv_genctl_set_state_uninit) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_core\_init](#_impdrv_genctl_core_init) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_init\_get\_func](#_impdrv_genctl_init_get_func) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_init\_chk\_state](#_impdrv_genctl_init_chk_state) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | i[mpdrv\_genctl\_init\_set\_state](#_impdrv_genctl_init_set_state) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | [impdrv\_genctl\_init\_attr\_init](#_impdrv_genctl_init_attr_init) | Private | ASIL D | r\_impdrv\_genctl.c |
|  | impdrv\_genctl\_get\_worksize | Private | ASIL D | r\_impdrv\_genctl.c |

### Public function

#### impdrv\_genctl\_init

Table 2‑35: impdrv\_genctl\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0001  impdrv\_genctl\_init  [Covers: AD\_PD\_CAS1001, AD\_PD\_CAS2003, AD\_PD\_CAS2004, AD\_PD\_CAS2005, AD\_PD\_CAS2006, AD\_PD\_CAS2007, AD\_PD\_CAS2008, AD\_PD\_CAS2009, AD\_PD\_CAS2010, AD\_PD\_CAS2011, AD\_PD\_CAS2012, AD\_PD\_CAS2117, AD\_PD\_CAS2118, AD\_PD\_CAS2119, AD\_PD\_CAS2120, AD\_PD\_CAS2121, AD\_PD\_CAS3002, AD\_PD\_CAS2227] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_init(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  st\_impdrv\_initdata\_t \*const p\_data,  const impdrv\_ctrl\_handle\_t osal\_cb\_args  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const impdrv\_ctrl\_handle\_t  osal\_cb\_args | | OSAL callback argument  The lifetime of this parameter is the period from the impdrv\_genctl\_init is executed until impdrv\_genctl\_quit is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle  The lifetime of this parameter is the period from the impdrv\_genctl\_init is executed until impdrv\_genctl\_quit is executed. | |
| **Range** | Not NULL |
| st\_impdrv\_initdata\_t \*const p\_data | | Initialization data  The lifetime of this parameter is until impdrv\_genctl\_init returns.  Refer st\_impdrv\_initdata\_t for the expiration of the members of the structure. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_INSTANCE | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_quit

Table 2‑36: impdrv\_genctl\_quit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0002  impdrv\_genctl\_quit  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS2069, AD\_PD\_CAS2070, AD\_PD\_CAS2071, AD\_PD\_CAS2072, AD\_PD\_CAS2073, AD\_PD\_CAS2074, AD\_PD\_CAS2075, AD\_PD\_CAS2076, AD\_PD\_CAS2077, AD\_PD\_CAS3002, AD\_PD\_CAS2557] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_quit(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | N/A | | N/A | | |
| **Range** | N/A | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Add arguments for call impdrv\_genctl\_chk\_state\_init * Add processing of call impdrv\_\*\*\*ctl\_quit for init state cores | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_start

Table 2‑37: impdrv\_genctl\_start

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0003  impdrv\_genctl\_start  [Covers: AD\_PD\_CAS1005, AD\_PD\_CAS2233, AD\_PD\_CAS2234, AD\_PD\_CAS2235, AD\_PD\_CAS2236, AD\_PD\_CAS2237, AD\_PD\_CAS2243, AD\_PD\_CAS2244, AD\_PD\_CAS2245, AD\_PD\_CAS2253] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_start(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | IMP Driver initialization is complete.  And the core is not execute. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_stop

Table 2‑38: impdrv\_genctl\_stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0004  impdrv\_genctl\_stop  [Covers: AD\_PD\_CAS1006, AD\_PD\_CAS2268, AD\_PD\_CAS2269, AD\_PD\_CAS2270, AD\_PD\_CAS2271, AD\_PD\_CAS2272, AD\_PD\_CAS2254] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_stop(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t  \*const p\_core\_info | | Core information | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | IMP Driver start is complete.  And the core is not execute. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_set\_cl\_brk\_addr

Table 2‑39 impdrv\_genctl\_set\_cl\_brk\_addr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0040  impdrv\_genctl\_set\_cl\_brk\_addr  [Covers: AD\_PD\_CAS2539, AD\_PD\_CAS2540, AD\_PD\_CAS2541, AD\_PD\_CAS2542, AD\_PD\_CAS2543, AD\_PD\_CAS2255, AD\_PD\_CAS2256, AD\_PD\_CAS2260, AD\_PD\_CAS2261, AD\_PD\_CAS2262] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_cl\_brk\_addr (  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \* const p\_core\_info,  uint32\_t cl\_brk\_addr  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core | |
| **Range** | Not NULL |
| uintptr\_t Cl\_brk\_addr | | CL break point physical address | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl, | | General control handle | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | |  | |
| **Preconditions** |  | |  | |
| **Remarks** |  | | | |

#### impdrv\_genctl\_execute

Table 2‑40: impdrv\_genctl\_execute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0005  impdrv\_genctl\_execute  [Covers: AD\_PD\_CAS1010, AD\_PD\_CAS2294, AD\_PD\_CAS2295, AD\_PD\_CAS2298, AD\_PD\_CAS2299, AD\_PD\_CAS2300] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_execute(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  const p\_impdrv\_cbfunc\_t callback\_func,  void \*const p\_callback\_args  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL |
| const p\_impdrv\_cbfunc\_t callback\_func | | Callback function | |
| **Range** | Not NULL |
| void  \*const p\_callback\_args | | Callback argument | |
| **Range** | None. |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver start is complete.  CL has been set. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_resume\_exe

Table 2‑41: impdrv\_genctl\_resume\_exe

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0006  impdrv\_genctl\_resume\_exe  [Covers: AD\_PD\_CAS1014, AD\_PD\_CAS2395, AD\_PD\_CAS2396, AD\_PD\_CAS2397, AD\_PD\_CAS2398, AD\_PD\_CAS2399] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_resume\_exe(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t  const \*p\_core\_info | | Core information | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_DSP\_HALT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | INT interrupt is coming after CL is running. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_pm\_set\_policy

Table 2‑42: impdrv\_genctl\_pm\_set\_policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0007  impdrv\_genctl\_pm\_set\_policy  [Covers: AD\_PD\_CAS1015, AD\_PD\_CAS2421, AD\_PD\_CAS2422, AD\_PD\_CAS2423, AD\_PD\_CAS2424, AD\_PD\_CAS2425] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_pm\_set\_policy(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info ,  const e\_impdrv\_pm\_policy\_t policy  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL |
| const e\_impdrv\_pm\_policy\_t policy | | Policy you want to set | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver initialization is complete. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_pm\_get\_policy

Table 2‑43: impdrv\_genctl\_pm\_get\_policy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0008  impdrv\_genctl\_pm\_get\_policy  [Covers: AD\_PD\_CAS1016, AD\_PD\_CAS2453, AD\_PD\_CAS2454, AD\_PD\_CAS2455, AD\_PD\_CAS2456, AD\_PD\_CAS2457] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_pm\_get\_policy(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  e\_impdrv\_pm\_policy\_t \*const p\_policy  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t  \*const p\_impdrv\_ctl | | General control handle | | | |
| **Range** | | Not NULL | |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | | | |
| **Range** | | | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | | | |
| **Parameters (Out)** | e\_impdrv\_pm\_policy\_t \*const p\_policy | | Current PM Policy | | | |
| **Range** | Not NULL | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | IMP Driver initialization is complete. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_genctl\_pm\_get\_state

Table 2‑44: impdrv\_genctl\_pm\_get\_state

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0015  impdrv\_genctl\_pm\_get\_state  [Covers: AD\_PD\_CAS1032, AD\_PD\_CAS2539, AD\_PD\_CAS2540, AD\_PD\_CAS2541, AD\_PD\_CAS2542, AD\_PD\_CAS2275] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_pm\_get\_state(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  e\_impdrv\_pm\_state\_t \*const p\_pmstate  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t  \*const p\_impdrv\_ctl | | General control handle | | | |
| **Range** | | Not NULL | |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | | | |
| **Range** | | | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | | | |
| **Parameters (Out)** | e\_impdrv\_pm\_state\_t \*const p\_pmstate | | Current PM State | | | |
| **Range** | Not NULL | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | IMP Driver initialization is complete. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_genctl\_attr\_init

Table 2‑45: impdrv\_genctl\_attr\_init

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0009  impdrv\_genctl\_attr\_init  [Covers: AD\_PD\_CAS1007, AD\_PD\_CAS3002] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_attr\_init(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t  \*const p\_core\_info | | Core information  The lifetime of this parameter is until this function returns. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle  The lifetime of this parameter is the period from the impdrv\_genctl\_init is executed until impdrv\_genctl\_quit is executed. | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_set\_core\_mem\_init

Table 2‑46: impdrv\_genctl\_set\_core\_mem\_init

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0010  impdrv\_genctl\_set\_core\_mem\_init  [Covers: AD\_PD\_CAS1005, AD\_PD\_CAS2140, AD\_PD\_CAS2366, AD\_PD\_CAS2367, AD\_PD\_CAS2368, AD\_PD\_CAS2369] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_core\_mem\_init(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info ,  const e\_impdrv\_param\_t enable  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | | | |
| **Range** | | Not NULL | |
| const e\_impdrv\_param\_t enable | | Whether to initialize memory | | | |
| **Range** | Within the range of ENUM type. | | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | | |
| **Range** | | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_genctl\_set\_core\_map

Table 2‑47: impdrv\_genctl\_set\_core\_map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0011  impdrv\_genctl\_set\_core\_map  [Covers: AD\_PD\_CAS1008, AD\_PD\_CAS2157, AD\_PD\_CAS2371, AD\_PD\_CAS2372, AD\_PD\_CAS2373, AD\_PD\_CAS2374, AD\_PD\_CAS2375] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_core\_map(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info ,  const st\_impdrv\_core\_info\_t core\_map [IMPDRV\_COREMAP\_MAXID]  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | | |
| **Range** | Not NULL | |
| const st\_impdrv\_core\_info\_t core\_map  [IMPDRV\_COREMAP\_MAXID] | | Setting Core map | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | IMP Driver start is complete.  And the core is not execute. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_set\_cl

Table 2‑48: impdrv\_genctl\_set\_cl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0012  impdrv\_genctl\_set\_cl  [Covers: AD\_PD\_CAS2176, AD\_PD\_CAS2377, AD\_PD\_CAS2378, AD\_PD\_CAS2380, AD\_PD\_CAS2379] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_cl(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info ,  const uint32\_t claddr\_phys  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL |
| const uint32\_t claddr\_phys | | Setting CL address | |
| **Range** | 4byte alignments |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | - | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver start is complete.  And the core is not execute. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_set\_irq\_mask

Table 2‑49: impdrv\_genctl\_set\_irq\_mask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0013  impdrv\_genctl\_set\_irq\_mask  [Covers: AD\_PD\_CAS1008, AD\_PD\_CAS2193, AD\_PD\_CAS2382, AD\_PD\_CAS2383, AD\_PD\_CAS2384, AD\_PD\_CAS2385] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_irq\_mask(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_core\_info\_t \*const p\_core\_info ,  const bool irq\_mask[IMPDRV\_IRQMASK\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL |
| const bool irq\_mask[IMPDRV\_IRQMASK\_MAX] | | Setting IRQ mask | |
| **Range** | true or false |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver start is complete.  And the core is not execute. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_set\_irq\_group

Table 2‑50: impdrv\_genctl\_set\_irq\_group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0014  impdrv\_genctl\_set\_irq\_group  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2210, AD\_PD\_CAS2387, AD\_PD\_CAS2388, AD\_PD\_CAS2389, AD\_PD\_CAS2390, AD\_PD\_CAS2391] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_irq\_group(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl ,  const st\_impdrv\_irq\_group\_t \*const p\_irq\_param  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_irq\_group\_t \*const p\_irq\_param | | Setting irq group | |
| **Range** | Not NULL |
| const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver start is complete.  And the core is not execute. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_int\_handler

Table 2‑51: impdrv\_genctl\_int\_handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0016  impdrv\_genctl\_int\_handler  [Covers: AD\_PD\_CAS1011, AD\_PD\_CAS2338, AD\_PD\_CAS2339, AD\_PD\_CAS2340, AD\_PD\_CAS2341] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | void impdrv\_genctl\_int\_handler(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const uint64\_t irq\_channel ,  const e\_impdrv\_errorcode\_t ercd  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can be called from OSAL interrupt. | | | |
| **Parameters (In)** | const uint64\_t irq\_channel | | IRQ channel associated with OSAL interrupt manager. | |
| **Range** | None. |
| Const e\_impdrv\_errorcode\_t ercd | | IMPDRV error code | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | N/A | | N/A | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver execute is complete. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_reg\_set\_mem\_protect

Table 2‑52: impdrv\_genctl\_reg\_set\_mem\_protect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0024  impdrv\_genctl\_reg\_set\_mem\_protect  [Covers: AD\_PD\_CAS1017, AD\_PD\_CAS2100, AD\_PD\_CAS2200, AD\_PD\_CAS2201, AD\_PD\_CAS2205 AD\_PD\_CAS2206] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_reg\_set\_mem\_protect(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t) protect\_mode  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t) protect\_mode | | Memory protection setting. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Add arguments for call impdrv\_genctl\_chk\_state\_init | | | |
| **Preconditions** | The core is not start. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_reg\_get\_hw\_info

Table 2‑53: impdrv\_genctl\_reg\_get\_hw\_info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0025  impdrv\_genctl\_reg\_get\_hw\_info  [Covers: AD\_PD\_CAS1018, AD\_PD\_CAS2104, AD\_PD\_CAS2207, AD\_PD\_CAS2208, AD\_PD\_CAS2211, AD\_PD\_CAS2212] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_reg\_get\_hw\_info(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  [st\_impdrv\_reg\_info\_t](#_st_impdrv_reg_info_t) \*const p\_reg\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | Not NULL | |
| const char \*const p\_device\_id | | Pointer to Target OSAL device ID. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | [st\_impdrv\_reg\_info\_t](#_st_impdrv_reg_info_t) \*const p\_reg\_info | | Pointer to Hardware register area specification information. | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | The core is uninitialized. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_genctl\_reg\_read32

Table 2‑54: impdrv\_genctl\_reg\_read32

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0026  impdrv\_genctl\_reg\_read32  [Covers: AD\_PD\_CAS1019, AD\_PD\_CAS2111, AD\_PD\_CAS2213, AD\_PD\_CAS2214, AD\_PD\_CAS2215, AD\_PD\_CAS2218] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_reg\_read32(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  const uintptr\_t offset,  uint32\_t \*const p\_data  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | | |
| **Range** | Not NULL | |
| const char \*const p\_device\_id | | Pointer to the Target OSAL device ID. | | |
| **Range** | Not NULL | |
| const uintptr\_t offset | | Sets the offset address from Register top. | | |
| **Range** | The offset  address should be specified as 4-byte alignment. | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | uint32\_t \*const p\_data | | Pointer to the read data to hardware registers. | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | The core is uninitialized. | | | | |
| **Remarks** | p\_data is pointer of output parameter, so the use of this pointer variable is acceptable. | | | | |

#### impdrv\_genctl\_reg\_write32

Table 2‑55: impdrv\_genctl\_reg\_write32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0027  impdrv\_genctl\_reg\_write32  [Covers: AD\_PD\_CAS1020, AD\_PD\_CAS2122, AD\_PD\_CAS2219, AD\_PD\_CAS2220, AD\_PD\_CAS2221, AD\_PD\_CAS2222] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_reg\_write32(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  const uintptr\_t offset,  const uint32\_t data  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const char \*const p\_device\_id | | Pointer to the Target OSAL device ID. | |
| **Range** | Not NULL |
| const uintptr\_t offset | | Sets the offset address from Register top. | |
| **Range** | The offset  address should be specified as 4-byte alignment. |
| const uint32\_t data | | Data written to hardware registers. | |
| **Range** | None |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | The core is uninitialized. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_reg\_required

Table 2‑56: impdrv\_genctl\_reg\_required

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0028  impdrv\_genctl\_reg\_required  [Covers: AD\_PD\_CAS1021, AD\_PD\_CAS2124, AD\_PD\_CAS2223, AD\_PD\_CAS2224, AD\_PD\_CAS2225, AD\_PD\_CAS2226] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_reg\_required(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  const [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t) new\_state  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const char \*const p\_device\_id | | Pointer to the Target OSAL device ID. | |
| **Range** | Not NULL |
| const [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t) new\_state | | Required state to the Register read/write function. | |
| **Range** | Within the range of ENUM type |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | The core is uninitialized. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_check\_intclear

Table 2‑57: impdrv\_genctl\_check\_intclear

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0029  impdrv\_genctl\_check\_intclear  [Covers: AD\_PD\_CAS2360] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_check\_intclear(  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Specified core information. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_bus\_if\_check

Table 2‑58: impdrv\_genctl\_bus\_if\_check

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0034  impdrv\_genctl\_bus\_if\_check  [Covers: AD\_PD\_CAS2301, AD\_PD\_CAS2307, AD\_PD\_CAS2308, AD\_PD\_CAS2309, AD\_PD\_CAS2400, AD\_PD\_CAS2401, AD\_PD\_CAS2402, AD\_PD\_CAS2403, AD\_PD\_CAS2404, AD\_PD\_CAS2416] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_bus\_if\_check(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info, | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource | | Pointer to the OSAL resource for bus interface check. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_conf\_reg\_check

Table 2‑59: impdrv\_genctl\_conf\_reg\_check

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0035  impdrv\_genctl\_conf\_reg\_check  [Covers: AD\_PD\_CAS2303, AD\_PD\_CAS2304, AD\_PD\_CAS2305, AD\_PD\_CAS2306, AD\_PD\_CAS2405, AD\_PD\_CAS2406, AD\_PD\_CAS2407, AD\_PD\_CAS2412, AD\_PD\_CAS2414, AD\_PD\_CAS2415] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_conf\_reg\_check(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const st\_impdrv\_chk\_param\_t \*const p\_chk\_param,  const uint32\_t param\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl, | | General control handle | |
| **Range** | Not NULL |
| const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param | | Pointer to Array of check parameters. | |
| **Range** | Not NULL |
| const uint32\_t param\_num, | | Number of check parameters | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Pointer to the Core information data. | |
|  | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | The core is ready. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_set\_dsp

able 2‑62: impdrv\_genctl\_set\_dsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD0X\_XX\_XXXX  impdrv\_genctl\_set\_dsp  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2276, AD\_PD\_CAS2277, AD\_PD\_CAS2278, AD\_PD\_CAS2279, AD\_PD\_CAS2579] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_dsp(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_app,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_fw,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_data,  const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_dtcm  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info, | | Specified core information. | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_app, | | Information of DSP data | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_fw, | | Information of DSP data | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_data | | Information of DSP data | |
| **Range** | Not NULL |
| const [st\_impdrv\_dsp\_data\_t](#_st_impdrv_dsp_data_t) \*const p\_dsp\_dtcm | | Information of DSP data | |
| **Range** | Not NULL |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | The core is ready. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_dsp\_int\_handler

Table 2‑62 impdrv\_genctl\_dsp\_int\_handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0017  impdrv\_genctl\_dsp\_int\_handler  [Covers: AD\_PD\_CAS1011, AD\_PD\_CAS2338, AD\_PD\_CAS2339, AD\_PD\_CAS2340, AD\_PD\_CAS2341, AD\_PD\_CAS2280, AD\_PD\_CAS2287, AD\_PD\_CAS2288] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | void impdrv\_genctl\_int\_handler(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const uint64\_t irq\_channel ,  const e\_impdrv\_errorcode\_t ercd,  const st\_impdrv\_core\_info\_t const core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can be called from OSAL interrupt. | | | |
| **Parameters (In)** | const uint64\_t irq\_channel | | IRQ channel associated with OSAL interrupt manager. | |
| **Range** | None. |
| Const e\_impdrv\_errorcode\_t ercd | | IMPDRV error code | |
| **Range** | Within the range of ENUM type. |
| const st\_impdrv\_core\_info\_t const core\_info | | Core information data | |
| **Range** |  |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | N/A | | N/A | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver execute is complete. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_set\_gosub\_cond

Table 2‑63: impdrv\_genctl\_set\_gosub\_cond

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0042  impdrv\_genctl\_set\_gosub\_cond  [Covers: AD\_PD\_CAS1031, AD\_PD\_CAS2263, AD\_PD\_CAS2264, AD\_PD\_CAS2265, AD\_PD\_CAS2273, AD\_PD\_CAS2274] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_gosub\_cond(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const e\_impdrv\_gosub\_cond\_t condition  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Pointer to the Core information data. | |
| **Range** | Not NULL |
| const e\_impdrv\_gosub\_cond\_t condition | | Conditional GOSUB instruction attribute. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl, | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | The core is ready. | | | |
| **Remarks** | N/A | | | |

### Private function

#### impdrv\_genctl\_chk\_core\_info

Table 2‑60: impdrv\_genctl\_chk\_core\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0021  impdrv\_genctl\_chk\_core\_info  [Covers: AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_chk\_core\_info(  const st\_impdrv\_core\_info\_t core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_info | | Core information for Specific core. | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_prologue

Table 2‑61: impdrv\_genctl\_prologue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0022  impdrv\_genctl\_prologue  [Covers: AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_prologue(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information for Specific core. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_epilogue

Table 2‑62: impdrv\_genctl\_epilogue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0023  impdrv\_genctl\_epilogue  [Covers: AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_epilogue(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const e\_impdrv\_errorcode\_t ercd  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| const e\_impdrv\_errorcode\_t  ercd | | Error code | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | Calling impdrv\_genctl\_prologue in advance. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_chk\_state\_init

Table 2‑63: impdrv\_genctl\_chk\_state\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0030  impdrv\_genctl\_chk\_state\_init  [Covers: AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_chk\_state\_init (  const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_corectl\_func\_t \*p\_funcs[IMPDRV\_INNER\_FIXED\_VALUE],  st\_impdrv\_core\_info\_t init\_core\_info[IMPDRV\_INNER\_FIXED\_VALUE],  uint32\_t \*const p\_init\_core\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | const st\_impdrv\_corectl\_func\_t \*p\_funcs[IMPDRV\_INNER\_FIXED\_VALUE] | | Array of specified control function | |
| st\_impdrv\_core\_info\_t init\_core\_info[IMPDRV\_INNER\_FIXED\_VALUE | | Array of the core of the Init state | |
| uint32\_t \*const p\_init\_core\_num | | Pointer to number of cores in Init state | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  [Update Information]   * Add Parameters (p\_funcs, init\_core\_info, p\_init\_core\_num) * Remove local variable (p\_funcs) * Replace local variable p\_funcs with Parameters p\_funcs * Update processing of Input parameter confirmation process for added Parameters * If impdrv\_\*\*\*ctl\_check\_state(INIT) return IMPDRV\_EC\_OK, store core information in Parameters init\_core\_info | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_chk\_state\_uninit

Table 2‑64: impdrv\_genctl\_chk\_state\_uninit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0031  impdrv\_genctl\_chk\_state\_uninit  [Covers: AD\_PD\_CAS2072, AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_chk\_state\_uninit (  const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_set\_state\_uninit

Table 2‑65: impdrv\_genctl\_set\_state\_uninit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0032  impdrv\_genctl\_set\_state\_uninit  [Covers: AD\_PD\_CAS2074, AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_set\_state\_uninit (  const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | const st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_core\_init

Table 2‑66: impdrv\_genctl\_core\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0033  impdrv\_genctl\_core\_init  [Covers: AD\_PD\_CAS2008, AD\_PD\_CAS2009, AD\_PD\_CAS2010, AD\_PD\_CAS2227, AD\_PD\_CAS2229, AD\_PD\_CAS3002] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_core\_init(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const st\_impdrv\_core\_info\_t core\_info[IMPDRV\_INNER\_FIXED\_VALUE],  const st\_impdrv\_corectl\_func\_t \*const p\_funcs[IMPDRV\_INNER\_FIXED\_VALUE]  const uint32\_t use\_core\_num,  const bool is\_dmac\_mb\_init  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_info  [IMPDRV\_INNER\_FIXED\_VALUE] | | Core information | |
| **Range** | Not NULL |
| st\_impdrv\_corectl\_func\_t \*const p\_funcs  [IMPDRV\_INNER\_FIXED\_VALUE] | | Core function information | |
| **Range** | Not NULL |
| const uint32\_t use\_core\_num | | Use Core num | |
| **Range** | One or more. IMPDRV\_INNER\_FIXED\_VALUE or less. |
| Const bool is\_dmac\_mb\_init | | DMAC multi-bank initialization | |
| **Range** | true or false |
| **Parameters (In-Out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_init\_get\_func

Table 2‑67: impdrv\_genctl\_init\_get\_func

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0036  impdrv\_genctl\_init\_get\_func  [Covers: AD\_PD\_CAS2005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_init\_get\_func (  const st\_impdrv\_initdata\_t \*const p\_data,  const uint32\_t use\_core\_num,  st\_impdrv\_core\_info\_t \*const p\_core\_array,  const st\_impdrv\_corectl\_func\_t \*\*const p\_func\_array  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_initdata\_t \*const p\_data | | Initialization data for IMP Driver. | |
| **Range** | Not NULL |
| const uint32\_t use\_core\_num | | Number of core information | |
| **Range** | Not 0.  Value less than IMPDRV\_INNER\_FIXED\_VALUE. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_core\_info\_t \*const p\_core\_array | | Array of specified core information | |
| **Range** | Not NULL |
| const st\_impdrv\_corectl\_func\_t \*\*const p\_func\_array | | Array of specified control function | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_init\_chk\_state

Table 2‑68: impdrv\_genctl\_init\_chk\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0037  impdrv\_genctl\_init\_chk\_state  [Covers: AD\_PD\_CAS2006] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_init\_chk\_state (  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const uint32\_t use\_core\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle. | |
| **Range** | Not NULL |
| const uint32\_t use\_core\_num | | Number of core information | |
| **Range** | Not 0.  Value less than IMPDRV\_INNER\_FIXED\_VALUE. |
| Const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Array of specified core information | |
| **Range** | Not NULL |
| const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs | | Array of specified control function | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_init\_set\_state

Table 2‑69: impdrv\_genctl\_init\_set\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0038  impdrv\_genctl\_init\_set\_state  [Covers: AD\_PD\_CAS2010] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_init\_set\_state (  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const uint32\_t use\_core\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const uint32\_t use\_core\_num | | Number of core information | |
| **Range** | Not 0.  Value less than IMPDRV\_INNER\_FIXED\_VALUE. |
| Const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Array of specified core information | |
| **Range** | Not NULL |
| const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs | | Array of specified control function | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_init\_attr\_init

Table 2‑70: impdrv\_genctl\_init\_attr\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0039  impdrv\_genctl\_init\_attr\_init  [Covers: AD\_PD\_CAS2227] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_genctl\_init\_attr\_init(  st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl,  const uint32\_t use\_core\_num,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const uint32\_t use\_core\_num | | Number of core information | |
| **Range** | Not 0.  Value less than IMPDRV\_INNER\_FIXED\_VALUE. |
| Const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Array of specified core information | |
| **Range** | Not NULL |
| const st\_impdrv\_corectl\_func\_t \*\*const p\_funcs | | Array of specified control function | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | General control handle. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_genctl\_get\_worksize

Table 2‑70: impdrv\_genctl\_get\_worksize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_02\_0043  impdrv\_genctl\_get\_worksize  [Covers: AD\_PD\_CAS2289] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_genctl\_get\_worksize(  uint32\_t work\_type,  uint32\_t \*p\_work\_size  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | uint32\_t work\_type | | type of work size | |
| **Range** | Not 0. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_ctl\_t \*const p\_impdrv\_ctl | | size of work area | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

## Functions of Common Control

Table 2‑71: Function List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Function Name** | **Access** | **ASIL** | **Source File Name** |
|  | impdrv\_cmnctl\_init | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_quit | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_judge\_int | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_mutex\_create | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_mutex\_destroy | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_mutex\_lock | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_mutex\_unlock | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_set\_irq\_group | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_get\_corectl\_func | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_get\_syncc\_val | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_fatal\_cb](#_impdrv_cmnctl_fatal_cb_1) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_chk\_instance\_num](#_impdrv_cmnctl_chk_instance_num) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_get\_io\_imptop](#_impdrv_cmnctl_get_io_imptop) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_get\_io\_dta](#_impdrv_cmnctl_get_io_dta) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_reg\_init](#_impdrv_cmnctl_reg_init) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_quit](#_impdrv_cmnctl_reg_quit_1) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_set\_mem\_protect](#_impdrv_cmnctl_reg_set_mem_protect) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_mem\_init](#_impdrv_cmnctl_reg_mem_init_1) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_get\_hw\_info](#_impdrv_cmnctl_reg_get_hw_info) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_read32](#_impdrv_cmnctl_reg_read32_1) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_write32](#_impdrv_cmnctl_reg_write32_1) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_required](#_impdrv_cmnctl_reg_required) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_save\_irq\_mask](#_impdrv_cmnctl_save_irq_mask) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_force\_irq\_mask](#_impdrv_cmnctl_force_irq_mask) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_restore\_irq\_mask](#_impdrv_cmnctl_restore_irq_mask) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_conf\_reg\_check](#_impdrv_cmnctl_conf_reg_check) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_reg\_chk\_mem\_protect](#_impdrv_cmnctl_reg_chk_mem_protect) | Public | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_get\_io\_dsp](#_impdrv_cmnctl_get_io_dsp) | Public | - | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_get\_init\_param](#_impdrv_cmnctl_get_init_param) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_get\_dsp\_func](#_impdrv_cmnctl_get_dsp_func) | Public | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_chk\_core\_info | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_cnv\_int\_core\_bit | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_cnv\_int\_bit\_core](#_impdrv_cmnctl_cnv_int_bit_core) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_execute\_no\_group](#_impdrv_cmnctl_execute_no_group_1) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_execute\_group](#_impdrv_cmnctl_execute_group_1) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_chk\_init\_data](#_impdrv_cmnctl_chk_init_data) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_chk\_group\_data](#_impdrv_cmnctl_chk_group_data) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_get\_group](#_impdrv_cmnctl_get_group) | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_calc\_checksum | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_set\_checksum | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | impdrv\_cmnctl\_test\_checksum | Private | ASIL D | r\_impdrv\_cmnctl.c |
|  | [impdrv\_cmnctl\_reg\_hwrsc\_open](#_impdrv_cmnctl_reg_hwrsc_open) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_hwrsc\_close](#_impdrv_cmnctl_reg_hwrsc_close) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_pow\_on\_hwrsc](#_impdrv_cmnctl_reg_pow_on_hwrsc_1) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_pow\_off\_hwrsc](#_impdrv_cmnctl_reg_pow_off_hwrsc) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_find\_resource](#_impdrv_cmnctl_reg_find_resource) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_chk\_rsc\_name](#_impdrv_cmnctl_reg_chk_rsc_name) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_mb\_initialize](#_impdrv_cmnctl_reg_mb_initialize) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | impdrv\_cmnctl\_reg\_set\_checksum | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | impdrv\_cmnctl\_reg\_test\_checksum | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_spm\_preset](#_impdrv_cmnctl_reg_spm_preset) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_spm\_initialize](#_impimpdrv_cmnctl_reg_spm_initialize) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_spm\_ena\_access](#_impdrv_cmnctl_reg_spm_ena_access) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_spm\_wait\_init](#_impdrv_cmnctl_reg_spm_wait_init) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |
|  | [impdrv\_cmnctl\_reg\_spm\_chk\_init](#_impdrv_cmnctl_reg_spm_chk_init) | Private | ASIL D | r\_impdrv\_cmnctl\_reg.c |

### Public function

#### impdrv\_cmnctl\_init

Table 2‑72: impdrv\_cmnctl\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0001  impdrv\_cmnctl\_init  [Covers: AD\_PD\_CAS1001, AD\_PD\_CAS1002, AD\_PD\_CAS2021, AD\_PD\_CAS2022, AD\_PD\_CAS2023, AD\_PD\_CAS2024, AD\_PD\_CAS2025, AD\_PD\_CAS2026, AD\_PD\_CAS2027, AD\_PD\_CAS2028, AD\_PD\_CAS2079, AD\_PD\_CAS2080, AD\_PD\_CAS3003, AD\_PD\_CAS2046] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_init(  st\_impdrv\_commonctl\_t \*const p\_commonctl,  const [st\_impdrv\_cmn\_init\_data\_t](#_st_impdrv_cmn_init_data_t) \*const p\_cmn\_init\_data,  bool \*const p\_is\_dmac\_mb\_init  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_cmn\_init\_data\_t](#_st_impdrv_cmn_init_data_t) \*const p\_cmn\_init\_data | | Common control initialization data. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_commonctl\_t \*const p\_commonctl, | | Common Control handle. | |
| **Range** | Not NULL |
| bool \*const p\_is\_dmac\_mb\_init | | DMAC multi-bank initialization | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_INSTANCE  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | Must be in “Uninitialized State”. | | | |
| **Remarks** | p\_is\_dmac\_mb\_init is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_quit

Table 2‑73: impdrv\_cmnctl\_quit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0003  impdrv\_cmnctl\_quit  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS2083, AD\_PD\_CAS2084, AD\_PD\_CAS2086, AD\_PD\_CAS2087, AD\_PD\_CAS2088, AD\_PD\_CAS2089, AD\_PD\_CAS3003, AD\_PD\_CAS2085, AD\_PD\_CAS2047] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_quit(  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | IMP Driver initialization is complete. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_judge\_int

Table 2‑74: impdrv\_cmnctl\_judge\_int

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0004  impdrv\_cmnctl\_judge\_int  [Covers: AD\_PD\_CAS1011, AD\_PD\_CAS2342, AD\_PD\_CAS2343, AD\_PD\_CAS2344, AD\_PD\_CAS2345, AD\_PD\_CAS2346, AD\_PD\_CAS2347] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_judge\_int(  const st\_impdrv\_commonctl\_t \*const p\_commonctl ,  const uint64\_t irq\_channel,  st\_impdrv\_core\_info\_t core\_info[IMPDRV\_INTSTS\_BITNUM] ,  uint32\_t \*const p\_use\_core\_num,  bool is\_fb\_callback  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL interrupt. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| const uint64\_t irq\_channel | | IRQ channel associated with OSAL interrupt manager. | |
| **Range** | None |
| bool is\_fb\_callback | | Whether to callback | |
| **Range** | true or false |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_core\_info\_t core\_info[IMPDRV\_INTSTS\_BITNUM] | | Core lists information | |
| **Range** | Not NULL |
| uint32\_t \*const p\_use\_core\_num | | Number of using core. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | p\_use\_core\_num is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_mutex\_create

Table 2‑75: impdrv\_cmnctl\_mutex\_create

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0005  impdrv\_cmnctl\_mutex\_create  [Covers: AD\_PD\_CAS1001, AD\_PD\_CAS2013, AD\_PD\_CAS2014, AD\_PD\_CAS2015, AD\_PD\_CAS2060, AD\_PD\_CAS3003] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_mutex\_create(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const osal\_mutex\_id\_t mutex\_id  const osal\_milli\_sec\_t time\_period  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const osal\_mutex\_id\_t  mutex\_id | | To set value of mutex id. | | | |
| **Range** | - | | |
| const osal\_milli\_sec\_t  time\_period | | Time period. | | | |
| **Range** | | 0 or more | |
| **Parameters (In-out)** | N/A | | N/A | | | |
| **Range** | | | N/A |
| **Parameters (Out)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle  The lifetime of this parameter is the period from the impdrv\_cmnctl\_mutex\_create is executed until impdrv\_cmnctl\_mutex\_destroy is executed. | | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_cmnctl\_mutex\_destroy

Table 2‑76: impdrv\_cmnctl\_mutex\_destroy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0006  impdrv\_cmnctl\_mutex\_destroy  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS2061, AD\_PD\_CAS2093, AD\_PD\_CAS2094, AD\_PD\_CAS2095, AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_mutex\_destroy(  const st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl, | | The Common Control handle  The lifetime of this parameter is the period from the impdrv\_cmnctl\_mutex\_create is executed until impdrv\_cmnctl\_mutex\_destroy is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_mutex\_lock

Table 2‑77: impdrv\_cmnctl\_mutex\_lock

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0007  impdrv\_cmnctl\_mutex\_lock  [Covers: AD\_PD\_CAS2016, AD\_PD\_CAS2017, AD\_PD\_CAS2018, AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_mutex\_lock(  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle  The lifetime of this parameter is the period from the impdrv\_cmnctl\_mutex\_create is executed until impdrv\_cmnctl\_mutex\_destroy is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Range** | N/A |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_mutex\_unlock

Table 2‑78: impdrv\_cmnctl\_mutex\_unlock

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0008  impdrv\_cmnctl\_mutex\_unlock  [Covers: AD\_PD\_CAS2029, AD\_PD\_CAS2030, AD\_PD\_CAS2031, AD\_PD\_CAS3003] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_mutex\_unlock(  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle  The lifetime of this parameter is the period from the impdrv\_cmnctl\_mutex\_create is executed until impdrv\_cmnctl\_mutex\_destroy is executed. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_cmnctl\_set\_irq\_group

Table 2‑79: impdrv\_cmnctl\_set\_irq\_group

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0009  impdrv\_cmnctl\_set\_irq\_group  [Covers: AD\_PD\_CAS2078, AD\_PD\_CAS2216, AD\_PD\_CAS2217, AD\_PD\_CAS2357, AD\_PD\_CAS2358, AD\_PD\_CAS2359] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_set\_irq\_group(  const st\_impdrv\_commonctl\_t \*const p\_commonctl ,  const st\_impdrv\_irq\_group\_t \*const p\_irq\_param  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle | | |
| **Range** | | Not NULL |
| const st\_impdrv\_irq\_group\_t \*const p\_irq\_param | | Setting IRQ group | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | N/A | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ATTRIBUTE | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_cmnctl\_get\_corectl\_func

Table 2‑80: impdrv\_cmnctl\_get\_corectl\_func

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0010  impdrv\_cmnctl\_get\_corectl\_func  [Covers: AD\_PD\_CAS1001, AD\_PD\_CAS1004, AD\_PD\_CAS1003, AD\_PD\_CAS1007, AD\_PD\_CAS2019, AD\_PD\_CAS2020, AD\_PD\_CAS3003, AD\_PD\_CAS1005, AD\_PD\_CAS1006, AD\_PD\_CAS1008, AD\_PD\_CAS1009, AD\_PD\_CAS1010, AD\_PD\_CAS1011, AD\_PD\_CAS1014, AD\_PD\_CAS1015] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | const st\_impdrv\_corectl\_func\_t \* impdrv\_cmnctl\_get\_corectl\_func(  const e\_impdrv\_core\_type\_t core\_type  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_core\_type\_t core\_type | | Specified core type.  The lifetime of this parameter is until this function returns. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | st\_impdrv\_corectl\_func\_t\* | | Address of Core control function table.  NULL If the argument is out of range | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_get\_syncc\_val

Table 2‑81: impdrv\_cmnctl\_get\_syncc\_val

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0011  impdrv\_cmnctl\_get\_syncc\_val  [Covers: AD\_PD\_CAS1008, AD\_PD\_CAS2081, AD\_PD\_CAS2161, AD\_PD\_CAS2162] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_get\_syncc\_val(  uint8\_t syncc\_val[IMPDRV\_COREMAP\_MAXID],  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const st\_impdrv\_core\_info\_t core\_map[IMPDRV\_COREMAP\_MAXID]); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_map[IMPDRV\_COREMAP\_MAXID] | | Setting Core map | |
| **Range** | Not NULL |
| const st\_impdrv\_commonctl\_t \*const p\_commonctl | | Common Control handle | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | uint8\_t syncc\_val[IMPDRV\_COREMAP\_MAXID] | | Core sync value | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_ATTRIBUTE | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | syncc\_val is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_fatal\_cb

Table 2‑82: impdrv\_cmnctl\_fatal\_cb

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0016  impdrv\_cmnctl\_fatal\_cb  [Covers: AD\_PD\_CAS2349, AD\_PD\_CAS2350] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_fatal\_cb(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const e\_impdrv\_fatalcode\_t fatal\_code,  const e\_impdrv\_errorcode\_t error\_code  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core lists information. | |
| **Range** | Not NULL |
| const e\_impdrv\_fatalcode\_t fatal\_code | | Fatal callback result code. | |
| **Range** | None |
| const e\_impdrv\_errorcode\_t error\_code | | IMP Driver error code. | |
| **Range** | None |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | [e\_impdrv\_errorcode\_t](#_API_Layer) | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Fatal callback to the framework.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_chk\_instance\_num

Table 2‑83: impdrv\_cmnctl\_chk\_instance\_num

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0013  impdrv\_cmnctl\_chk\_instance\_num  [Covers: AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_chk\_instance\_num(  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_get\_io\_imptop

Table 2‑84: impdrv\_cmnctl\_get\_io\_imptop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0020  impdrv\_cmnctl\_get\_io\_imptop  [Covers: AD\_PD\_CAS2129, AD\_PD\_CAS2291, AD\_PD\_CAS2296] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_get\_io\_imptop (  st\_impdrv\_device\_handle\_t \*const p\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t  \*const p\_handle | | Device handle of ‘imp\_top’ resource. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_get\_io\_dta

Table 2‑85: impdrv\_cmnctl\_get\_io\_dta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0021  impdrv\_cmnctl\_get\_io\_dta  [Covers: AD\_PD\_CAS1019] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_get\_io\_dta (  st\_impdrv\_device\_handle\_t \*const p\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t  \*const p\_handle | | Device handle of ‘imp\_dta’ resource. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_init

Table 2‑86: impdrv\_cmnctl\_reg\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0025  impdrv\_cmnctl\_reg\_init  [Covers: AD\_PD\_CAS1022, AD\_PD\_CAS2228, AD\_PD\_CAS2126, AD\_PD\_CAS2127, AD\_PD\_CAS2128, AD\_PD\_CAS2027, AD\_PD\_CAS2028, AD\_PD\_CAS2029, AD\_PD\_CAS2133, AD\_PD\_CAS2134, AD\_PD\_CAS2135] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_init (  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_quit

Table 2‑87: impdrv\_cmnctl\_reg\_quit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0026  impdrv\_cmnctl\_reg\_quit  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS2230, AD\_PD\_CAS2145, AD\_PD\_CAS2146, AD\_PD\_CAS2147, AD\_PD\_CAS2148, AD\_PD\_CAS2152, AD\_PD\_CAS2153, AD\_PD\_CAS2154] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_quit (  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_set\_mem\_protect

Table 2‑93: impdrv\_cmnctl\_reg\_set\_mem\_protect

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0039  impdrv\_cmnctl\_reg\_set\_mem\_protect  [Covers: AD\_PD\_CAS2201, AD\_PD\_CAS2155, AD\_PD\_CAS2158, AD\_PD\_CAS2159] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_set\_mem\_protect (  st\_impdrv\_commonctl\_t \*const p\_commonctl,  const [***e\_impdrv\_protect\_mode\_t***](#_e_impdrv_protect_mode_t)  protect\_mode  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const [***e\_impdrv\_protect\_mode\_t***](#_e_impdrv_protect_mode_t)  protect\_mode | | Memory protection setting. | | |
| **Range** | | Within the range of ENUM type. |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | The Common Control handle | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | ECC: V3M and V3H are not supported. | | | | |

#### impdrv\_cmnctl\_reg\_mem\_init

Table 2‑89: impdrv\_cmnctl\_reg\_mem\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0027  impdrv\_cmnctl\_reg\_mem\_init  [Covers: AD\_PD\_CAS1023, AD\_PD\_CAS2082, AD\_PD\_CAS2136, AD\_PD\_CAS2137, AD\_PD\_CAS2138, AD\_PD\_CAS2141, AD\_PD\_CAS2142, AD\_PD\_CAS2143, AD\_PD\_CAS2144] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_mem\_init (  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_get\_hw\_info

Table 2‑90: impdrv\_cmnctl\_reg\_get\_hw\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0028  impdrv\_cmnctl\_reg\_get\_hw\_info  [Covers: AD\_PD\_CAS1018, AD\_PD\_CAS2208, AD\_PD\_CAS2160, AD\_PD\_CAS2163, AD\_PD\_CAS2164] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_get\_hw\_info (  st\_impdrv\_commonctl\_t \*const p\_commonctl,  const char \*const p\_device\_id, [***st\_impdrv\_reg\_info\_t***](#_API_Layer) \*const info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Permitted | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | The Common Control handle | |
| **Range** | Not NULL |
| const char \*const p\_device\_id, | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | [***st\_impdrv\_reg\_info\_t***](#_API_Layer) \*const info | | Pointer to Hardware register area specification information. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_read32

Table 2‑91: impdrv\_cmnctl\_reg\_read32

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0029  impdrv\_cmnctl\_reg\_read32  [Covers: AD\_PD\_CAS1019, AD\_PD\_CAS2214, AD\_PD\_CAS2165, AD\_PD\_CAS2166, AD\_PD\_CAS2167, AD\_PD\_CAS2171, AD\_PD\_CAS2172] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_read32(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const char \*const p\_device\_id,  const uintptr\_t offset,  uint32\_t \*const p\_data  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | | |
| **Range** | Not NULL | |
| const char \*const  p\_device\_id, | | Pointer to Target OSAL device ID. | | |
| **Range** | Not NULL | |
| const uintptr\_t offset | | Sets the offset address from Register top. | | |
| **Range** | The offset  address should be specified as 4-byte alignment. | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | uint32\_t \*const p\_data | | Pointer to the read data to hardware registers. | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_cmnctl\_reg\_write32

Table 2‑92: impdrv\_cmnctl\_reg\_write32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0030  impdrv\_cmnctl\_reg\_read32  [Covers: AD\_PD\_CAS1020, AD\_PD\_CAS2220, AD\_PD\_CAS2173, AD\_PD\_CAS2174, AD\_PD\_CAS2177, AD\_PD\_CAS2178, AD\_PD\_CAS2179] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_write32(  const st\_impdrv\_commonctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  const uintptr\_t offset,  const uint32\_t data  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_impdrv\_ctl | | Common control handle | |
| **Range** | Not NULL |
| const char \*const  p\_device\_id, | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| const uintptr\_t offset | | Sets the offset address from Register top. | |
| **Range** | The offset  address should be specified as 4-byte alignment. |
| const uint32\_t data | | Data written to hardware registers. | |
| **Range** | None |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_required

Table 2‑98: impdrv\_cmnctl\_reg\_required

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0031  impdrv\_cmnctl\_reg\_required  [Covers: AD\_PD\_CAS1021, AD\_PD\_CAS2224, AD\_PD\_CAS2180, AD\_PD\_CAS2181, AD\_PD\_CAS2182, AD\_PD\_CAS2183, AD\_PD\_CAS2184] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_required(  const st\_impdrv\_commonctl\_t \*const p\_impdrv\_ctl,  const char \*const p\_device\_id,  const [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t) new\_state  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_impdrv\_ctl | | Common control handle | |
| **Range** | Not NULL |
| const char \*const  p\_device\_id, | | Pointer to Target OSAL device ID. | |
| **Range** | Not NULL |
| const [e\_impdrv\_reg\_req\_state\_t](#_e_impdrv_reg_req_state_t) new\_state | | Required state to the Register read/write function. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_save\_irq\_mask

Table 2‑94: impdrv\_cmnctl\_save\_irq\_mask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0040  impdrv\_cmnctl\_save\_irq\_mask  [Covers: AD\_PD\_CAS2308, AD\_PD\_CAS2513, AD\_PD\_CAS2481] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_save\_irq\_mask (  st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | This function is save the current IRQ mask settings .  To do so, do the following.  ・Check the Instance number value.  ・Read the IRQ mask register value.  ・Save the IRQ mask register value.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_force\_irq\_mask

Table 2‑95: impdrv\_cmnctl\_force\_irq\_mask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0041  impdrv\_cmnctl\_force\_irq\_mask  [Covers: AD\_PD\_CAS2309, AD\_PD\_CAS2482, AD\_PD\_CAS2483] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_force\_irq\_mask (  const st\_impdrv\_commonctl\_t \*const p\_commonctl  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | This function is force setting the IRQ mask with no group.  To do so, do the following.  ・Check the Instance number value.  ・Write the IRQ mask register  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_restore\_irq\_mask

Table 2‑96: impdrv\_cmnctl\_restore\_irq\_mask

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0049  impdrv\_cmnctl\_restore\_irq\_mask  [Covers: AD\_PD\_CAS2402, AD\_PD\_CAS2484, AD\_PD\_CAS2485] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_restore\_irq\_mask (  const st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | This function is restore the IRQ mask settings form save data.  To do so, do the following.  ・Check the Instance number value.  ・Restore of IRQ mask register.  ・Save the IRQ mask register value.  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### pdrv\_cmnctl\_conf\_reg\_check

Table 2‑97: impdrv\_cmnctl\_conf\_reg\_check

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0042  impdrv\_cmnctl\_conf\_reg\_check  [Covers: AD\_PD\_CAS2412, AD\_PD\_CAS2486, AD\_PD\_CAS2487] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_conf\_reg\_check(  const st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | |
| **Range** | Not NULL |
| **Parameters(In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | This function is configuration register check in Common control .  To do so, do the following.  ・Read the configuration register  ・Check the configuration register  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_reg\_chk\_mem\_protect

Table 2‑98: impdrv\_cmnctl\_reg\_chk\_mem\_protect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0043  impdrv\_cmnctl\_reg\_chk\_mem\_protect  [Covers: AD\_PD\_CAS2412, AD\_PD\_CAS2533, AD\_PD\_CAS2534, AD\_PD\_CAS2535] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_chk\_mem\_protect(  const st\_impdrv\_commonctl\_t \*const p\_commonctl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | This function is check the Memory protection mode.  To do so, do the following.  ・Check the Memory protection setting  Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_get\_io\_dsp

Not Use (Out of scope)

#### impdrv\_cmnctl\_get\_init\_param

Table 2‑101: impdrv\_cmnctl\_get\_init\_param

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0056  impdrv\_cmnctl\_get\_init\_param  [Covers: AD\_PD\_CAS1034, AD\_PD\_CAS2297, AD\_PD\_CAS2310] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_get\_init\_param(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  e\_osal\_interrupt\_priority\_t \*const p\_irq\_priority,  [impdrv\_ctrl\_handle\_t](#_impdrv_ctrl_handle_t)  \*const p\_osal\_cb\_args  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t  \*const p\_commonctl | | Common control handle | | |
| **Range** | Not NULL | |
| **Parameters(In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | e\_osal\_interrupt\_priority\_t \*const p\_irq\_priority | | The value of interrupt priority | | |
| **Range** | | Within the range of ENUM type |
| [impdrv\_ctrl\_handle\_t](#_impdrv_ctrl_handle_t)  \*const p\_osal\_cb\_args | | Interrupt handler function's argument. | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_cmnctl\_get\_dsp\_func

Table 2‑98: impdrv\_cmnctl\_get\_dsp\_func

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0045  impdrv\_cmnctl\_get\_dsp\_func  [Covers: AD\_PD\_CAS2324, AD\_PD\_CAS2325, AD\_PD\_CAS2327, AD\_PD\_CAS2328, AD\_PD\_CAS2348, AD\_PD\_CAS2362, AD\_PD\_CAS2290] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | p\_impdrv\_dspctl\_dsp\_start\_t impdrv\_cmnctl\_get\_dsp\_func(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Reentrant | | |
| **Interrupt State** | . | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters(In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | p\_impdrv\_dspctl\_dsp\_start\_t | | Function pointer of the Control start of DSP core pre-preparation. |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | |
| **Preconditions** | N/A | | |
| **Remarks** | N/A | | |

### Private function

#### impdrv\_cmnctl\_chk\_core\_info

Table 2‑99: impdrv\_cmnctl\_chk\_core\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0012  impdrv\_cmnctl\_chk\_core\_info  [Covers: AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_chk\_core\_info(  const st\_impdrv\_core\_info\_t core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_info | | Core information for Specific core. | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_cnv\_int\_core\_bit

Table 2‑100: impdrv\_cmnctl\_cnv\_int\_core\_bit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0014  impdrv\_cmnctl\_cnv\_int\_core\_bit  [Covers: AD\_PD\_CAS2021] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_cnv\_int\_core\_bit(  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const uint32\_t use\_core\_num,  const st\_impdrv\_inten\_tbl\_t \*const p\_s\_inten\_cnv\_table,  const uint32\_t cnv\_table\_num,  uint32\_t \*const p\_val  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Array of core information. | |
| **Range** | Not NULL |
| const uint32\_t use\_core\_num | | Maximum number of cores | |
| **Range** | One or more.  IMPDRV\_INNER\_FIXED\_VALUE or less. |
| Const st\_impdrv\_inten\_tbl\_t \*const p\_s\_inten\_cnv\_table | | Array of convert information | |
| **Range** | Not NULL |
| const uint32\_t cnv\_table\_num | | Number of elements in array p\_s\_inten\_cnv\_table[] | |
| **Range** | One or more. IMPDRV\_INNER\_FIXED\_VALUE or less. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Range** | N/A |
| **Parameters (Out)** | uint32\_t \*const p\_val | | Address of area to output value. | |
| **Return Value** | [e\_impdrv\_errorcode\_t](#_API_Layer) | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | p\_val is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_cnv\_int\_bit\_core

Table 2‑101: impdrv\_cmnctl\_cnv\_int\_bit\_core

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0017  impdrv\_cmnctl\_cnv\_int\_bit\_core  [Covers: AD\_PD\_CAS2331] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_cnv\_int\_bit\_core(  const st\_impdrv\_inten\_tbl\_t \*const p\_s\_inten\_cnv\_table,  const uint32\_t cnv\_table\_num,  const uint32\_t stat,  st\_impdrv\_core\_info\_t core\_info[IMPDRV\_INTSTS\_BITNUM],  uint32\_t \*const p\_use\_core\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_inten\_tbl\_t \*const p\_s\_inten\_cnv\_table | | Array of convert information | | |
| **Range** | Not NULL | |
| const uint32\_t cnv\_table\_num | | Number of elements in array p\_s\_inten\_cnv\_table[] | | |
| **Range** | Maximum number of cores for the Interrupt group. | |
| Const uint32\_t stat | | Register value. | | |
| **Range** | None | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | [st\_impdrv\_core\_info\_t](#_API_Layer_2)  core\_info[IMPDRV\_INTSTS\_BITNUM] | | Array of core information. | | |
| **Range** | | Not NULL |
| uint32\_t  \*const p\_use\_core\_num | | Number of using core. | | |
| **Range** | | Not NULL |
| **Return Value** | [e\_impdrv\_errorcode\_t](#_API_Layer) | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | p\_use\_core\_num is pointer of output parameter, so the use of this pointer variable is acceptable. | | | | |

#### impdrv\_cmnctl\_execute\_no\_group

Table 2‑102: impdrv\_cmnctl\_execute\_no\_group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0018  impdrv\_cmnctl\_execute\_no\_group  [Covers: AD\_PD\_CAS2357, AD\_PD\_CAS2358, AD\_PD\_CAS2359] | | | |
| **ASIL Level** | ASIL D | **ASIL Level** | ASIL D | |
| **Syntax** | IMPDRV\_STATIC  [e\_impdrv\_errorcode\_t](#_API_Layer) impdrv\_cmnctl\_execute\_no\_group (  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const st\_impdrv\_group\_tbl\_t \*const p\_inten\_group\_table,  const e\_impdrv\_instance\_t instance\_num,  const uint32\_t write\_val  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| const st\_impdrv\_group\_tbl\_t \*const p\_inten\_group\_table | | Interrupt group register. | |
| **Range** | Not NULL |
| const e\_impdrv\_instance\_t instance\_num | | Instance number. | |
| **Range** | Within the range of ENUM type. |
| Const uint32\_t write\_val | | Interrupt register value. | |
| **Range** | None |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | [e\_impdrv\_errorcode\_t](#_API_Layer) | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | Can be executed in the “Ready State”. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_execute\_group

Table 2‑103: impdrv\_cmnctl\_execute\_group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0019  impdrv\_cmnctl\_execute\_group  [Covers: AD\_PD\_CAS2357, AD\_PD\_CAS2358, AD\_PD\_CAS2359] | | | |
| **ASIL Level** | ASIL D | **ASIL Level** | ASIL D | |
| **Syntax** | IMPDRV\_STATIC  [e\_impdrv\_errorcode\_t](#_API_Layer) impdrv\_cmnctl\_execute\_group (  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const st\_impdrv\_group\_tbl\_t \*const p\_inten\_group\_table,  const e\_impdrv\_instance\_t instance\_num,  const uint32\_t group\_reg\_write\_val,  const e\_impdrv\_irq\_group\_t irq\_group;  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| const st\_impdrv\_group\_tbl\_t \*const p\_inten\_group\_table | | Array of interrupt group register. | |
| **Range** | Not NULL |
| const e\_impdrv\_instance\_t instance\_num | | Instance number. | |
| **Range** | Within the range of ENUM type. |
| Const uint32\_t group\_reg\_write\_val | | Value to set to interrupt register. | |
| **Range** | None |
| const e\_impdrv\_irq\_group\_t  irq\_group | | Interrupt group number. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | [e\_impdrv\_errorcode\_t](#_API_Layer) | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | Can be executed in the “Ready State”. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_cmnctl\_chk\_init\_data

Table 2‑104: impdrv\_cmnctl\_chk\_init\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0022  impdrv\_cmnctl\_chk\_init\_data  [Covers: AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | [Other than R-Car V4H2]  IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_chk\_init\_data(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const [st\_impdrv\_cmn\_init\_data\_t](#_st_impdrv_cmn_init_data_t) \*const p\_cmn\_init\_data,  const bool \*const p\_is\_dmac\_mb\_init,  uint32\_t \*const p\_use\_core\_info  );  [R-Car V4H2]  IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_chk\_init\_data(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const [st\_impdrv\_cmn\_init\_data\_t](#_st_impdrv_cmn_init_data_t) \*const p\_cmn\_init\_data,  const bool \*const p\_is\_dmac\_mb\_init,  uint32\_t \*const p\_use\_core\_info,  uint32\_t \*const p\_use\_dsp\_core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl, | | Common Control handle. | |
| **Range** | Not NULL |
| const [st\_impdrv\_cmn\_init\_data\_t](#_st_impdrv_cmn_init_data_t) \*const p\_cmn\_init\_data | | Common control initialization data. | |
| **Range** | Not NULL |
| const bool \*const p\_is\_dmac\_mb\_init | | MAC multi-bank initialization | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | uint32\_t \*const p\_use\_core\_info | | Use core information | |
| **Range** | Not NULL |
| uint32\_t \*const p\_use\_dsp\_core\_info | | [Use R-Car V4h2 Only]  Use DSP core information. | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_INSTANCE  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | p\_use\_core\_info is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_chk\_group\_data

Table 2‑105: impdrv\_cmnctl\_chk\_group\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0023  impdrv\_cmnctl\_chk\_group\_data  [Covers: AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_chk\_group\_data(  const st\_impdrv\_commonctl\_t \*const p\_commonctl,  const st\_impdrv\_irq\_group\_t, \*const p\_irq\_param  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_commonctl\_t \*const p\_commonctl, | | Common Control handle. | |
| **Range** | Not NULL |
| const st\_impdrv\_irq\_group\_t, \*const p\_irq\_param | | Setting IRQ group. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_get\_group

Table 2‑106: impdrv\_cmnctl\_get\_group

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0024  impdrv\_cmnctl\_get\_group  [Covers: AD\_PD\_CAS3003, AD\_PD\_CAS2085, AD\_PD\_CAS2343] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_get\_group(  e\_impdrv\_irq\_group\_t \*const p\_irq\_group,  const e\_impdrv\_instance\_t instance\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Interrupt register value. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | | |
| **Range** | | Within the range of ENUM type. |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | None | | | | |
| **Remarks** | None | | | | |

#### impdrv\_cmnctl\_calc\_checksum

Table 2‑107: impdrv\_cmnctl\_calc\_checksum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0044  impdrv\_cmnctl\_calc\_checksum  [Covers: AD\_PD\_CAS2522, AD\_PD\_CAS2523, AD\_PD\_CAS2524] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_calc\_checksum(  const void\* \*const p\_data,  const size\_t size,  uint32\_t \*const p\_checksum  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const void\*  \*const p\_data | | Start address | |
| **Range** | Not NULL |
| const size\_t  size | | Data size | |
| **Range** | N/A |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | uint32\_t  \*const p\_checksum | | Checksum value | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | p\_checksum is pointer of output parameter, so the use of this pointer variable is acceptable. | | | |

#### impdrv\_cmnctl\_set\_checksum

Table 2‑108: impdrv\_cmnctl\_set\_checksum

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0047  impdrv\_cmnctl\_set\_checksum  [Covers: AD\_PD\_CAS2522, AD\_PD\_CAS2524] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_set\_checksum(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Non-Reentrant | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL |
| **Description** | Refer to “3 Activity Diagrams”. | | |
| **Preconditions** | None | | |
| **Remarks** | None | | |

#### impdrv\_cmnctl\_test\_checksum

Table 2‑109: impdrv\_cmnctl\_test\_checksum

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0048  impdrv\_cmnctl\_test\_checksum  [Covers: AD\_PD\_CAS2523] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_test\_checksum(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Non-Reentrant | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR |
| **Description** | Refer to “3 Activity Diagrams”. | | |
| **Preconditions** | None | | |
| **Remarks** | None | | |

#### impdrv\_cmnctl\_reg\_hwrsc\_open

Table 2‑110: impdrv\_cmnctl\_reg\_hwrsc\_open

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0032  impdrv\_cmnctl\_reg\_hwrsc\_open  [Covers: AD\_PD\_CAS2129] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_hwrsc\_open(  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_hwrsc\_close

Table 2‑111: impdrv\_cmnctl\_reg\_hwrsc\_close

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0033  impdrv\_cmnctl\_reg\_hwrsc\_close  [Covers: AD\_PD\_CAS2148] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_hwrsc\_close (  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams” | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_pow\_on\_hwrsc

Table 2‑112: impdrv\_cmnctl\_reg\_pow\_on\_hwrsc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0034  impdrv\_cmnctl\_reg\_pow\_on\_hwrsc  [Covers: AD\_PD\_CAS2129] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_pow\_on\_hwrsc(  const e\_impdrv\_instance\_t instance\_num,  const uint32\_t index  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| Const uint32\_t index | | Index number to Management table | |
| **Range** | Value less than IMPDRV\_HWRSC\_TABLE\_NUM |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_pow\_off\_hwrsc

Table 2‑113: impdrv\_cmnctl\_reg\_pow\_off\_hwrsc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0035  impdrv\_cmnctl\_reg\_pow\_off\_hwrsc  [Covers: AD\_PD\_CAS2148] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_pow\_off\_hwrsc(  const e\_impdrv\_instance\_t instance\_num,  const uint32\_t index  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| Const uint32\_t index | | Index number to Management table | |
| **Range** | Value less than IMPDRV\_HWRSC\_TABLE\_NUM |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_find\_resource

Table 2‑114: impdrv\_cmnctl\_reg\_find\_resource

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0036  impdrv\_cmnctl\_reg\_find\_resource  [Covers: AD\_PD\_CAS2126, AD\_PD\_CAS2160, AD\_PD\_CAS2174, AD\_PD\_CAS2181] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_find\_resource(  const char \*const p\_device\_id,  uint32\_t \*const p\_index,  [st\_impdrv\_hwrsc\_def\_t](#_st_impdrv_hwrsc_def_t) \*const p\_resource  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | | | |
| **Range** | Within the range of ENUM type. | | |
| **Parameters (In-out)** | N/A | | N/A | | | |
| **Parameters (Out)** | uint32\_t \*const p\_index | | Pointer to Resource index value. | | | |
| **Range** | | Not NULL | |
| [st\_impdrv\_hwrsc\_def\_t](#_st_impdrv_hwrsc_def_t) \*const p\_resource | | Pointer to Resource definition data | | | |
| **Range** | | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | None | | | | | |
| **Remarks** | None | | | | | |

#### impdrv\_cmnctl\_reg\_chk\_rsc\_name

Table 2‑115: impdrv\_cmnctl\_reg\_chk\_rsc\_name

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0037  impdrv\_cmnctl\_reg\_chk\_rsc\_name  [Covers: AD\_PD\_CAS3003] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_chk\_rsc\_name(  const char \*const p\_rsc\_name,  const char \*const p\_device\_id  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const p\_rsc\_name | | Pointer to Resource name string. | |
| **Range** | Not NULL |
| const char \*const p\_device\_id | | Pointer to the Target OSAL device ID. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_mb\_initialize

Table 2‑116: impdrv\_cmnctl\_reg\_mb\_initialize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0038  impdrv\_cmnctl\_reg\_mb\_initialize  [Covers: AD\_PD\_CAS2141, AD\_PD\_CAS2142] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_mb\_initialize(  const [e\_impdrv\_mb\_mem\_init\_t](#_e_impdrv_mb_mem_init_t) mb\_mem\_init\_kind  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [e\_impdrv\_mb\_mem\_init\_t](#_e_impdrv_mb_mem_init_t) mb\_mem\_init\_kind | | Multi-Bank memory initialization kind | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_set\_checksum

Table 2‑117: impdrv\_cmnctl\_reg\_set\_checksum

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0055  impdrv\_cmnctl\_reg\_set\_checksum  [Covers: AD\_PD\_CAS2522, AD\_PD\_CAS2524] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_set\_checksum(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Non-Reentrant | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL |
| **Description** | Refer to “3 Activity Diagrams”. | | |
| **Preconditions** | None | | |
| **Remarks** | None | | |

#### impdrv\_cmnctl\_reg\_test\_checksum

Table 2‑118: impdrv\_cmnctl\_reg\_test\_checksum

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0046  impdrv\_cmnctl\_reg\_test\_checksum  [Covers: AD\_PD\_CAS2523] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_test\_checksum(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Non-Reentrant | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR |
| **Description** | Refer to “3 Activity Diagrams”. | | |
| **Preconditions** | None | | |
| **Remarks** | None | | |

#### impdrv\_cmnctl\_reg\_spm\_preset

Table 2‑126: impdrv\_cmnctl\_reg\_spm\_preset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0050  impdrv\_cmnctl\_reg\_spm\_preset  [Covers: AD\_PD\_CAS2141] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_spm\_preset (  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams” | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impimpdrv\_cmnctl\_reg\_spm\_initialize

Table 2‑127:impdrv\_cmnctl\_reg\_spm\_initialize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0054  impdrv\_cmnctl\_reg\_spm\_initialize  [Covers: AD\_PD\_CAS2141, AD\_PD\_CAS2142] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_spm\_initialize (  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams” | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_spm\_ena\_access

Table 2‑128: impdrv\_cmnctl\_reg\_spm\_ena\_access

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0051  impdrv\_cmnctl\_reg\_spm\_ena\_access  [Covers: AD\_PD\_CAS2141] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_spm\_ena\_access (  const uint32\_t index,  const bool enable  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const uint32\_t index | | Index number to Management table. | |
| **Range** | Value less than IMPDRV\_HWRSC\_TABLE\_NUM |
| const bool enable | | Enable flag of SPM register access. | |
| **Range** | true or false |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | MPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_spm\_wait\_init

Table 2‑129:impdrv\_cmnctl\_reg\_spm\_wait\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0053  impdrv\_cmnctl\_reg\_spm\_wait\_init  [Covers: AD\_PD\_CAS2136] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_spm\_wait\_init (  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams” | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

#### impdrv\_cmnctl\_reg\_spm\_chk\_init

Table 2‑129:impdrv\_cmnctl\_reg\_spm\_chk\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_03\_0052  impdrv\_cmnctl\_reg\_spm\_chk\_init  [Covers: AD\_PD\_CAS2136] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_cmnctl\_reg\_spm\_chk\_init (  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| const uint32\_t num\_devices | | Number of Target OSAL device ID. | |
| **Range** | Not 0. |
| const char \*const\*const p\_device\_id | | Pointer to the Target OSAL device ID. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams” | | | |
| **Preconditions** | None | | | |
| **Remarks** | None | | | |

## Functions of Core Control

Table 2‑119: Function List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Function Name** | **Access** | **ASIL** | **Source File Name** |
|  | impdrv\_\*\*\*ctl\_is\_valid\_core | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_check\_state | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_state | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_init\_start | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_init\_end | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_start | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_stop | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_quit | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_attr\_init | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_mem\_init | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_core\_map | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_cl | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_irq\_mask | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr](#_impdrv_***ctl_set_cl_brk_addr) | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_execute](#_impdrv_***ctl_execute) | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_int\_handler | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_resume\_exe | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_pm\_set\_policy | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_pm\_get\_policy | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_pm\_get\_state | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_dmactl\_mb\_init](#_impdrv_dmactl_mb_init) | Public | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_\*\*\*ctl\_bus\_if\_check](#_impdrv_***ctl_bus_if_check) | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_conf\_reg\_check](#_impdrv_***ctl_conf_reg_check) | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_dspctl\_dsp\_start\_pre | Public | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_set\_dsp](#_impdrv_dspctl_set_dsp) | Public | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_\*\*\*ctl\_set\_cond\_gosub](#_impdrv_***ctl_set_cond_gosub) | Public | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_init\_core | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_init\_core (\*\*\*:dma/dmas) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_get\_inten\_val | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_is\_sub\_thread | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_osal\_core\_num | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_read\_reg | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_write\_reg | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_dmactl\_tbl\_write\_reg](#_impdrv_dmactl_tbl_write_reg) | Private | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_dmactl\_mb\_init\_pre](#_impdrv_dmactl_mb_init_pre) | Private | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_dmactl\_mb\_init\_main](#_impdrv_dmactl_mb_init_main) | Private | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_dmactl\_cl\_pre](#_impdrv_dmactl_cl_pre) | Private | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_dmactl\_int\_safety\_func](#_impdrv_***ctl_int_main_func) | Private | ASIL D | r\_impdrv\_dmactl.c |
|  | [impdrv\_\*\*\*ctl\_int\_main\_func](#_impdrv_***ctl_int_main_func) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_int\_safety\_func](#_impdrv_dmactl_int_safety_func) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_dma\_transfer](#_impdrv_dmactl_dma_transfer) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_tbl\_write\_reg | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_check\_core\_busy | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_check\_core\_busy (\*\*\*:dma/dmas) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_soft\_reset](#_impdrv_***ctl_soft_reset) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_soft\_reset (\*\*\*:dma/dmas) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_set\_syncc\_config](#_impdrv_***ctl_set_syncc_config) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_set\_syncc\_config (\*\*\*:dma/dmas) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_int\_cb](#_impdrv_***ctl_int_cb) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | impdrv\_\*\*\*ctl\_int\_cb (\*\*\*:imp/imps) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_check\_inten](#_impdrv_***ctl_check_inten) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_ocvctl\_check\_inten](#_drv_ocvctl_check_inten) | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | [impdrv\_ocvctl\_irq\_status\_clear](#_impdrv_ocvctl_irq_status_clear) | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | [impdrv\_ocvctl\_check\_inten\_1st](#_impdrv_ocvctl_check_inten_1st) | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | [impdrv\_ocvctl\_check\_inten\_2nd](#_impdrv_ocvctl_check_inten_2st) | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | impdrv\_ocvctl\_init\_lwm | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | [impdrv\_ocvctl\_init\_core](#_impdrv_ocvctl_init_core) | Private | ASIL D | r\_impdrv\_ocvctl.c |
|  | impdrv\_dspctl\_dsp\_update\_app | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | impdrv\_dspctl\_load\_ptcm | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | impdrv\_dspctl\_stop\_reg | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_\*\*\*ctl\_set\_extend\_config](#_impdrv_***ctl_set_extend_config) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_\*\*\*ctl\_set\_extend\_config (\*\*\*:dma/dmas)](#_impdrv_***ctl_set_extend_config_(**) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_dspctl\_init\_dsp](#_impdrv_dspctl_init_dsp) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_dspctl\_get\_device\_io](#_impdrv_dspctl_get_device_io) | Private | ASIL D | r\_impdrv\_\*\*\*ctl.c |
|  | [impdrv\_dspctl\_init\_core](#_impdrv_dspctl_init_core) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_chk\_execute\_data](#_impdrv_dspctl_chk_execute_data) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_dsp\_execute\_pre](#_impdrv_dspctl_dsp_execute_pre) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_tcm\_config\_d](#_impdrv_dspctl_tcm_config_d) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_tcm\_config\_p](#_impdrv_dspctl_tcm_config_p) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_load\_dtcm](#_impdrv_dspctl__load_dtcm) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_dsp\_standby](#_impdrv_dspctl_dsp_standby) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_dsp\_foece\_standby](#_impdrv_dspctl_dsp_foece_standby) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | [impdrv\_dspctl\_chk\_int\_data](#_impdrv_dspctl_chk_int_data) | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | impdrv\_dspctl\_crc\_sub | Private | ASIL D | r\_impdrv\_dspctl.c |
|  | impdrv\_dspctl\_dsp\_start | Public | ASIL D | r\_impdrv\_dspctl.c |
|  | impdrv\_dspctl\_dsp\_execute | Public | ASIL D | r\_impdrv\_dspctl.c |

\*\*\* is present for each core.Specifically, there are the following

* imp
* psc
* dma
* ocv
* cnn
* imps
* dmas
* dsp

### Public function

#### impdrv\_\*\*\*ctl\_is\_valid\_core

Table 2‑120: impdrv\_\*\*\*ctl\_is\_valid\_core

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0001  impdrv\_\*\*\*ctl\_is\_valid\_core  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS3004] | | | | | |
| **ASIL Level** | ASIL D | | | **Status** | New | |
| **Syntax** | bool impdrv\_\*\*\*ctl\_is\_valid\_core(  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number.  The lifetime of this parameter is until this function returns. | |
| **Range** | Less than num of device |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | bool | | | | true  false | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | Check the validity of the core number. | | | |
| Imp | core\_num is less than IMPDRV\_IMP\_CORENUM\_VAL | | | |
| cnn | core\_num is less than IMPDRV\_CNN\_CORENUM\_VAL | | | |
| dma | core\_num is less than IMPDRV\_DMA\_CORENUM\_VAL | | | |
| ocv | core\_num is less than IMPDRV\_OCV\_CORENUM\_VAL | | | |
| psc | core\_num is less than IMPDRV\_PSC\_CORENUM\_VAL | | | |
| imps | core\_num is less than IMPDRV\_PSC\_CORENUM\_VAL | | | |
| dmas | core\_num is less than IMPDRV\_PSC\_CORENUM\_VAL | | | |
| dsp | core\_num is less than IMPDRV\_DSP\_CORENUM\_VAL | | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_check\_state

Table 2‑121: impdrv\_\*\*\*ctl\_check\_state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0002  impdrv\_\*\*\*ctl\_check\_state  [Covers: AD\_PD\_CAS1001, AD\_PD\_CAS1004, AD\_PD\_CAS1007, AD\_PD\_CAS2032, AD\_PD\_CAS2033, AD\_PD\_CAS2096, AD\_PD\_CAS3004, AD\_PD\_CAS1005, AD\_PD\_CAS1006, AD\_PD\_CAS1008, AD\_PD\_CAS1009, AD\_PD\_CAS1010, AD\_PD\_CAS1011, AD\_PD\_CAS2313] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_check\_state(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_state\_t state  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. | | |
| **Range** | Not NULL | |
| const uint32\_t core\_num | | | | Specified core number  The lifetime of this parameter is until this function returns. | | |
| **Range** | Less than num of device | |
| const e\_impdrv\_state\_t state | | | | IMP Driver state  The lifetime of this parameter is until this function returns. | | |
| **Range** | | Refer to 5.4.2.4.1. |
| **Parameters (In-out)** | N/A | | | | N/A | | |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | no difference between the cores. | | | |
| Cnn | | no difference between the cores. | | | |
| Dma | | no difference between the cores. | | | |
| Ocv | | no difference between the cores. | | | |
| Psc | | no difference between the cores. | | | |
| Imps | | no difference between the cores. | | | |
| Dmas | | no difference between the cores. | | | |
| Dsp | | no difference between the cores. | | | |
| **Preconditions** | N/A | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_set\_state

Table 2‑122: impdrv\_\*\*\*ctl\_set\_state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0003  impdrv\_\*\*\*ctl\_set\_state  [Covers: AD\_PD\_CAS1004, AD\_PD\_CAS1005, AD\_PD\_CAS1006, AD\_PD\_CAS1011, AD\_PD\_CAS2041, AD\_PD\_CAS3004] | | | | | | |
| **ASIL Level** | ASIL D | | | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_state(  st\_impdrv\_corectl\_t \*const p\_corectl,  const e\_impdrv\_state\_t state  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const e\_impdrv\_state\_t state | | | | IMP Driver state  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed until impdrv\_\*\*\*ctl\_stop is executed. | | |
| **Range** | Refer to 5.4.2.4.1. | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core.  The lifetime of this parameter is until this function returns. | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* |  | | | | |
| imp | no difference between the cores. | | | | |
| Cnn | no difference between the cores. | | | | |
| Dma | no difference between the cores. | | | | |
| Ocv | no difference between the cores. | | | | |
| Psc | no difference between the cores. | | | | |
| Imps | no difference between the cores. | | | | |
| Dmas | no difference between the cores. | | | | |
| **Preconditions** | N/A | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_init\_start

Table 2‑123: impdrv\_\*\*\*ctl\_init\_start

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0004  impdrv\_\*\*\*ctl\_init\_start  [Covers: AD\_PD\_CAS1003, AD\_PD\_CAS1005, AD\_PD\_CAS2034, AD\_PD\_CAS2035, AD\_PD\_CAS2036, AD\_PD\_CAS2037, AD\_PD\_CAS3004, AD\_PD\_CAS1033, AD\_PD\_CAS2321, AD\_PD\_CAS2322] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_init\_start(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t  core\_num | | | | Specified core number  The lifetime of this parameter is until impdrv\_\*\*\*ctl\_init\_start returns. | |
| **Range** | Less than num of device |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for each core type.  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed until impdrv\_\*\*\*ctl\_stop is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| About Request osdep to power on core.  <Only DSP core>  Call impdrv\_osdep\_pow\_on\_imp\_dsp and impdrv\_osdep\_reset\_core\_dsp  <Other than DSP core>  Call impdrv\_osdep\_pow\_on\_imp | | | | | |
| Differences between cores. | \*\*\* | | Check the IMP Core version | | |
| imp | | Version Control Register: IMPDRV\_REG\_IMP\_VCR  Check value: upper 16bit is IMPDRV\_IMP\_VCR | | |
| cnn | | Version Control Register: IMPDRV\_REG\_CNN\_VCR  Check value: IMPDRV\_CNN\_VCR | | |
| dma | | Version Control Register: IMPDRV\_REG\_DMA\_VCR  Check value: IMPDRV\_DMA\_VCR | | |
| ocv | | Version Control Register: IMPDRV\_REG\_OCV\_VCR0  Check value: IMPDRV\_OCV\_VCR | | |
| psc | | Version Control Register: IMPDRV\_REG\_PSC\_VCR  Check value: IMPDRV\_PSC\_VCR | | |
| imps | | Version Control Register: IMPDRV\_IMPS\_VCR  Check value: upper 16bit is IMPDRV\_IMPS\_VCR | | |
| dmas | | Version Control Register: IMPDRV\_DMAS\_VCR  Check value: IMPDRV\_DMAS\_VCR | | |
| dsp | | Version Control Register: IMPDRV\_REGOFS\_VDSP\_VERSION1  Check value: IMPDRV\_REGVAL\_VDSP\_VERSION1 | | |
| **Preconditions** | Must be in “Uninitialized State”. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_init\_end

Table 2‑124: impdrv\_\*\*\*ctl\_init\_end

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0005  impdrv\_\*\*\*ctl\_init\_end  [Covers: AD\_PD\_CAS1003, AD\_PD\_CAS2038, AD\_PD\_CAS2039, AD\_PD\_CAS2040, AD\_PD\_CAS3004, AD\_PD\_CAS1033, AD\_PD\_CAS2323] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_init\_end(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number  The lifetime of this parameter is until impdrv\_\*\*\*ctl\_init\_start returns. | | |
| **Range** | Less than num of device | |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for each core type  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed impdrv\_\*\*\*ctl\_stop is executed. | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | Difference is to describe the activity diagram. | | | |
| Cnn | | Difference is to describe the activity diagram. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | Difference is to describe the activity diagram. | | | |
| Psc | | Difference is to describe the activity diagram. | | | |
| Imps | | Difference is to describe the activity diagram. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| Dsp | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | Must be in “Uninitialized State”. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_start

Table 2‑125: impdrv\_\*\*\*ctl\_start

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0006  impdrv\_\*\*\*ctl\_start  [Covers: AD\_PD\_CAS2246, AD\_PD\_CAS2247, AD\_PD\_CAS2248, AD\_PD\_CAS2249, AD\_PD\_CAS2250, AD\_PD\_CAS2251] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_start(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | |
| **Range** | Less than num of device |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check thread, Init core and get core num | | |
| imp | | Initialization of specified core. | | |
| Cnn | | no difference between the cores. | | |
| Dma | | Check thread: call impdrv\_dmactl\_is\_sub\_thread() to check if main thread or sub thread .  Get core num: call impdrv\_dmactl\_osal\_core\_num() to get osal core number from core number.  Init core: call impdrv\_dmactl\_init\_core(). | | |
| Ocv | | Initialization of specified core. | | |
| Psc | | no difference between the cores. | | |
| Imps | | Initialization of specified core. | | |
| Dmas | | Check thread: call impdrv\_dmasctl\_is\_sub\_thread() to check if main thread or sub thread .  Get core num: call impdrv\_dmasctl\_osal\_core\_num() to get osal core number from core number.  Init core: call impdrv\_dmasctl\_init\_core(). | | |
| Dsp | | N/A | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_stop

Table 2‑126: impdrv\_\*\*\*ctl\_stop

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0007  impdrv\_\*\*\*ctl\_stop  [Covers: AD\_PD\_CAS1006, AD\_PD\_CAS2281, AD\_PD\_CAS2282, AD\_PD\_CAS2283, AD\_PD\_CAS2284, AD\_PD\_CAS2285, AD\_PD\_CAS2286, AD\_PD\_CAS2602, AD\_PD\_CAS2413, AD\_PD\_CAS2417] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_stop(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | Less than num of device | |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Add expression of whether ercd is IMPDRV\_EC\_OK or not when call impdrv\_dspctl\_stop\_reg | | | | | | |
| Differences between cores. | \*\*\* | | Check thread | | | |
| imp | | no difference between the cores. | | | |
| Cnn | | no difference between the cores. | | | |
| Dma | | Refer to 2.4.1.8 | | | |
| Ocv | | no difference between the cores. | | | |
| Psc | | no difference between the cores.. | | | |
| imps | | no difference between the cores. | | | |
| Dmas | | Refer to 2.4.1.8 | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_stop(\*\*\*:dma/dmas)

Table 2‑126: impdrv\_\*\*\*ctl\_stop(\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0080  impdrv\_\*\*\*ctl\_stop  [Covers: AD\_PD\_CAS1006, AD\_PD\_CAS2281, AD\_PD\_CAS2282, AD\_PD\_CAS2283, AD\_PD\_CAS2284] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_stop(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | Less than num of device | |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Add expression of whether ercd is IMPDRV\_EC\_OK or not when call impdrv\_dspctl\_stop\_reg | | | | | | |
| Differences between cores. | \*\*\* | | Check thread | | | |
| Dma | | no difference between the cores. | | | |
| Dmas | | no difference between the cores. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_quit

Table 2‑127: impdrv\_\*\*\*ctl\_quit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0024  impdrv\_\*\*\*ctl\_quit  [Covers: AD\_PD\_CAS2561, AD\_PD\_CAS2437, AD\_PD\_CAS2562, AD\_PD\_CAS2563] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_quit(  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | |
| **Range** | Less than num of device |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check thread | | |
| imp | | no difference between the cores. | | |
| Cnn | | When PM state is READY or RESET, call impdrv\_cnnctl\_force\_reset. | | |
| Dma | | When getting osal core number, call impdrv\_dmactl\_osal\_core\_num. | | |
| Ocv | | no difference between the cores. | | |
| Psc | | no difference between the cores. | | |
| Cnns | | no difference between the cores. | | |
| Dmas | | When getting osal core number, call impdrv\_dmactl\_osal\_core\_num. | | |
| Dsp | | When device open, call impdrv\_osdep\_dev\_open\_dsp instead of impdrv\_osdep\_dev\_open\_imp. | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_attr\_init

Table 2‑128: impdrv\_\*\*\*ctl\_attr\_init

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0008  impdrv\_\*\*\*ctl\_attr\_init  [Covers: AD\_PD\_CAS1007, AD\_PD\_CAS2130, AD\_PD\_CAS2131, AD\_PD\_CAS2132, AD\_PD\_CAS3004] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_attr\_init(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number  The lifetime of this parameter is until this function returns. | | |
| **Range** | Less than num of device | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed until impdrv\_\*\*\*ctl\_init\_start is executed. | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | | Check state | | | |
| imp | | call impdrv\_impctl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV3M / R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_HPINT] = true  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_APIPINT] = true  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true | | | |
| cnn | | call impdrv\_cnnctl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV3M / R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_MSCO] = true  (R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true | | | |
| dma | | call impdrv\_dmactl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true | | | |
| ocv | | call impdrv\_ocvctl\_check\_state() to check ‘INIT’.  (R-CarV3M / R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_USIER] = false;  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_USINT] = false;  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_PBCOVF] = true;  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_SBO0ME] = true;  (R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true; | | | |
| psc | | call impdrv\_pscctl\_check\_state() to check ‘INIT’.  (R-CarV3M / R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_END] = true  (R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true | | | |
| imps | | call impdrv\_impctl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV3H / R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_HPINT] = true  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_APIPINT] = true  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true | | | |
| dmas | | call impdrv\_dmasctl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV3H\_2)  p\_corectl->irq\_mask[IMPDRV\_IRQMASK\_WUPCOVF] = true . | | | |
| dsp | | call impdrv\_dspctl\_check\_state() to check ‘INIT’.  Irqmask setting  (R-CarV4H\_2)  p\_corectl->irq\_mask[i] = true .  \* i =0～IMPDRV\_IRQMASK\_MAX-1  Reset dsp\_info information  p\_corectl->dsp\_info.dsp\_app.addr\_phys = IMPDRV\_INIT\_DSPMEM\_ADDR  p\_corectl->dsp\_info.dsp\_app.size = 0U  p\_corectl->dsp\_info.dsp\_fw.addr\_phys = IMPDRV\_INIT\_DSPMEM\_ADDR  p\_corectl->dsp\_info.dsp\_fw.size = 0U  p\_corectl->dsp\_info.dsp\_data.addr\_phys = IMPDRV\_INIT\_DSPMEM\_ADDR  p\_corectl->dsp\_info.dsp\_data.size = 0U  p\_corectl->dsp\_info.dsp\_dtcm.addr\_phys = IMPDRV\_INIT\_DSPMEM\_ADDR  p\_corectl->dsp\_info.dsp\_dtcm.size = 0U  p\_corectl->dsp\_info.is\_update = false  p\_corectl->dsp\_info.is\_loaded = false  p\_corectl->dsp\_info.is\_executed = false | | | |
| **Preconditions** | Must be in “Initialization State”. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_set\_mem\_init

Table 2‑129: impdrv\_\*\*\*ctl\_set\_mem\_init

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0009  impdrv\_\*\*\*ctl\_set\_mem\_init  [Covers: AD\_PD\_CAS1005, AD\_PD\_CAS2149, AD\_PD\_CAS2150, AD\_PD\_CAS2151] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_mem\_init(  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num,  const e\_impdrv\_param\_t enable  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | |
| **Range** | Less than num of device |
| const e\_impdrv\_param\_t enable | | | | memory initialization possible flag | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check state | | |
| imp | | call impdrv\_impctl\_check\_state() to check ‘INIT’. | | |
| Cnn | | call impdrv\_cnnctl\_check\_state() to check ‘INIT’. | | |
| Dma | | call impdrv\_dmactl\_check\_state() to check ‘INIT’. | | |
| Ocv | | call impdrv\_ocvctl\_check\_state() to check ‘INIT’. | | |
| Psc | | call impdrv\_pscctl\_check\_state() to check ‘INIT’. | | |
| Imps | | call impdrv\_impsctl\_check\_state() to check ‘INIT’. | | |
| Dmas | | call impdrv\_dmasctl\_check\_state() to check ‘INIT’. | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_set\_core\_map

Table 2‑130: impdrv\_\*\*\*ctl\_set\_core\_map

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0010  impdrv\_\*\*\*ctl\_set\_core\_map  [Covers: AD\_PD\_CAS2168, AD\_PD\_CAS2169, AD\_PD\_CAS2170] | | | | | | |
| **ASIL Level** | ASIL D | | | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_core\_map(  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num,  const uint8\_t syncc\_val[IMPDRV\_COREMAP\_MAXID]  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Number of specified cores | | |
| **Range** | Less than num of device. | |
| Const uint8\_t syncc\_val | | | | Core sync value | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | Check state | | | | |
| imp | call impdrv\_impctl\_check\_state() to check ‘READY’. | | | | |
| Cnn | call impdrv\_cnnctl\_check\_state() to check ‘READY’. | | | | |
| Dma | call impdrv\_dmactl\_check\_state() to check ‘READY’. | | | | |
| Ocv | call impdrv\_ocvctl\_check\_state() to check ‘READY’. | | | | |
| Psc | call impdrv\_pscctl\_check\_state() to check ‘READY’. | | | | |
| Imps | call impdrv\_impsctl\_check\_state() to check ‘READY’. | | | | |
| Dmas | call impdrv\_dmasctl\_check\_state() to check ‘READY’. | | | | |
| Dsp | call impdrv\_dspctl\_check\_state () to check ‘READY’ | | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_set\_cl

Table 2‑131: impdrv\_\*\*\*ctl\_set\_cl

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0011  impdrv\_\*\*\*ctl\_set\_cl  [Covers: AD\_PD\_CAS1009, AD\_PD\_CAS2185 AD\_PD\_CAS2186, AD\_PD\_CAS2187] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_cl(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const uint32\_t claddr\_phys  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Number of specified cores | | |
| **Range** | Less than num of device. | |
| Const uint32\_t claddr\_phys | | | | CL address | | |
| **Range** | | 4byte alignments |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | | Check state | | | |
| imp | | call impdrv\_impctl\_check\_state() to check ‘READY’. | | | |
| Cnn | | call impdrv\_cnnctl\_check\_state() to check ‘READY’. | | | |
| Dma | | call impdrv\_dmactl\_check\_state() to check ‘READY’. | | | |
| Ocv | | call impdrv\_ocvctl\_check\_state() to check ‘READY’. | | | |
| Psc | | call impdrv\_pscctl\_check\_state() to check ‘READY’. | | | |
| Imps | | call impdrv\_impsctl\_check\_state() to check ‘READY’. | | | |
| Dmas | | call impdrv\_dmasctl\_check\_state() to check ‘READY’. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_set\_irq\_mask

Table 2‑132: impdrv\_\*\*\*ctl\_set\_irq\_mask

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0012  impdrv\_\*\*\*ctl\_set\_irq\_mask  [Covers: AD\_PD\_CAS1008, AD\_PD\_CAS2202, AD\_PD\_CAS2203, AD\_PD\_CAS2204] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_irq\_mask (  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num,  const bool irq\_mask[IMPDRV\_IRQMASK\_MAX]  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Number of specified cores | |
| **Range** | Less than num of device. |
| Const bool  irq\_mask[IMPDRV\_IRQMASK\_MAX] | | | | Register bits to be masked  (different for each core, internally converted) | |
| **Range** | true or false |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check state | | |
| imp | | call impdrv\_impctl\_check\_state() to check ‘READY’. | | |
| Cnn | | call impdrv\_cnnctl\_check\_state() to check ‘READY’. | | |
| Dma | | call impdrv\_dmactl\_check\_state() to check ‘READY’. | | |
| Ocv | | call impdrv\_ocvctl\_check\_state() to check ‘READY’. | | |
| Psc | | call impdrv\_pscctl\_check\_state() to check ‘READY’. | | |
| Imps | | call impdrv\_impsctl\_check\_state() to check ‘READY’. | | |
| Dmas | | call impdrv\_dmasctl\_check\_state() to check ‘READY’. | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr

Table 2‑133 impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0053  impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr  [Covers: AD\_PD\_CAS2544, AD\_PD\_CAS2545, AD\_PD\_CAS2546] | | | | |
| **ASIL Level** | ASIL D | | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const uint32\_t cl\_brk\_addr  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | The state of the IMP driver can be executed at the ‘Ready’ or ‘Int’ state.  Refer to e\_impdrv\_state\_t for interrupt status. | | | | |
| **Parameters (In)** | const uint32\_t  core\_num | | | Number of specified cores | |
| **Range** | Less than num of device. |
| const uint32\_t  cl\_brk\_addr | | | CL beak address | |
| **Range** | 4byte alignments.  However, exclude the invalid value (0xFFFFFFFFU). |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t  \*const p\_corectl | | | The Core control handles | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | N/A | |
| **Return Value** | [***e\_impdrv\_errorcode\_t***](#_API_Layer) | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Setting CL break address.  To do so, do the following.  ・Gets the following from work memory.  - Status  ・Setting CL break address in work memory. | | | | |
| Refer to “3 Activity Diagrams”. | | | | |
| Differences between cores. | \*\*\* |  | | |
| imp | no difference between the cores. | | |
| Cnn | no difference between the cores. | | |
| Dma | no difference between the cores. | | |
| Ocv | no difference between the cores. | | |
| Psc | no difference between the cores. | | |
| Imps | N/A | | |
| Dmas | no difference between the cores. | | |
| Dsp | N/A | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_\*\*\*ctl\_execute

Table 2‑134: impdrv\_\*\*\*ctl\_execute

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0014  impdrv\_\*\*\*ctl\_execute  [Covers: AD\_PD\_CAS1010, AD\_PD\_CAS2313, AD\_PD\_CAS2315, AD\_PD\_CAS2316, AD\_PD\_CAS2317, AD\_PD\_CAS2318, AD\_PD\_CAS2319, AD\_PD\_CAS2320, AD\_PD\_CAS2314] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_execute(  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num,  const p\_impdrv\_cbfunc\_t callback\_func ,  void \*const p\_callback\_args  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const p\_impdrv\_cbfunc\_t callback\_func | | | | Callback function | | |
| **Range** | Not NULL | |
| void  \*const p\_callback\_args | | | | Callback argument | | |
| **Range** | None | |
| const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | Less than num of device. | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_ATTRIBUTE | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Remove expression other than whether ercd is IMPDRV\_EC\_OK or not when call impdrv\_dspctl\_dsp\_update\_app * Cast void of return value of callback\_func | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | Difference is to describe the activity diagram. | | | |
| Cnn | | Difference is to describe the activity diagram. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | Difference is to describe the activity diagram. | | | |
| Psc | | Difference is to describe the activity diagram. | | | |
| Imps | | Difference is to describe the activity diagram. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_int\_handler

Table 2‑135: impdrv\_\*\*\*ctl\_int\_handler

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0015  impdrv\_\*\*\*ctl\_int\_handler  [Covers: AD\_PD\_CAS1011, AD\_PD\_CAS2351, AD\_PD\_CAS2352, AD\_PD\_CAS2353, AD\_PD\_CAS2354, AD\_PD\_CAS2355, AD\_PD\_CAS2356, AD\_PD\_CAS2311, AD\_PD\_CAS2361, AD\_PD\_CAS1037] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_int\_handler(  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | Less than num of device. | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | Difference is to describe the activity diagram. | | | |
| Cnn | | Difference is to describe the activity diagram. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | Difference is to describe the activity diagram. | | | |
| Psc | | Difference is to describe the activity diagram. | | | |
| Imps | | Difference is to describe the activity diagram. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| Dsp | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_resume\_exe

Table 2‑136: impdrv\_\*\*\*ctl\_resume\_exe

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0016  impdrv\_\*\*\*ctl\_resume\_exe  ***[Covers: AD\_PD\_CAS1014, AD\_PD\_CAS2408, AD\_PD\_CAS2409, AD\_PD\_CAS2410, AD\_PD\_CAS2411,*** AD\_PD\_CAS2312, AD\_PD\_CAS1038, AD\_PD\_CAS2433] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_resume\_exe(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback.  To enable the interrupt, set the following registers.  - Hardware interrupt | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | Less than num of device. | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t  \* const p\_corectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | Difference is to describe the activity diagram. | | | |
| Cnn | | Difference is to describe the activity diagram. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | Difference is to describe the activity diagram. | | | |
| Psc | | Difference is to describe the activity diagram. | | | |
| Imps | | Difference is to describe the activity diagram. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| Dsp | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_pm\_set\_policy

Table 2‑137: impdrv\_\*\*\*ctl\_pm\_set\_policy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0017  impdrv\_\*\*\*ctl\_pm\_set\_policy  [Covers: AD\_PD\_CAS1015, AD\_PD\_CAS2434, AD\_PD\_CAS2435, AD\_PD\_CAS2436] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_pm\_set\_policy(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const e\_impdrv\_pm\_policy\_t policy  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| const uint32\_t core\_num | | | | Specified core number | |
| **Range** | Less than num of device. |
| Const e\_impdrv\_pm\_policy\_t policy | | | | Policy you want to set | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Remove call impdrv\_osdep\_pow\_on\_imp\_dsp and impdrv\_osdep\_pow\_off\_imp | | | | | |
| Differences between cores. | \*\*\* | |  | | |
| imp | | no difference between the cores. | | |
| Cnn | | no difference between the cores. | | |
| Dma | | Difference is to describe the activity diagram. | | |
| Ocv | | no difference between the cores. | | |
| Psc | | no difference between the cores. | | |
| Imps | | no difference between the cores. | | |
| Dmas | | Difference is to describe the activity diagram. | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_pm\_get\_policy

Table 2‑138: impdrv\_\*\*\*ctl\_pm\_get\_policy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0018  impdrv\_\*\*\*ctl\_pm\_get\_policy  [Covers: AD\_PD\_CAS1016, AD\_PD\_CAS2466, AD\_PD\_CAS2467, AD\_PD\_CAS2468] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_pm\_get\_policy(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  e\_impdrv\_pm\_policy\_t \*const p\_policy  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | | Less than num of device. |
| **Parameters (In-out)** | N/A | | | | N/A | | |
| **Parameters (Out)** | e\_impdrv\_pm\_policy\_t \*const p\_policy | | | | Current PM Policy | | |
| **Range** | Not NULL | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Remove call impdrv\_osdep\_pow\_on\_imp\_dsp and impdrv\_osdep\_pow\_off\_imp | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | no difference between the cores. | | | |
| Cnn | | no difference between the cores. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | no difference between the cores. | | | |
| Psc | | no difference between the cores. | | | |
| Imps | | no difference between the cores. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_pm\_get\_state

Table 2‑139: impdrv\_\*\*\*ctl\_pm\_get\_state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0013  impdrv\_\*\*\*ctl\_pm\_get\_state  [Covers: AD\_PD\_CAS2558, AD\_PD\_CAS2559, AD\_PD\_CAS2560] | | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_pm\_get\_state(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  e\_impdrv\_pm\_state\_t \*const p\_pmstate  ); | | | | | | |
| **Sync/Async** | Synchronous | | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Specified core number | | |
| **Range** | | Less than num of device. |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for specific core | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | e\_impdrv\_pm\_state\_t \*const p\_pmstate | | | | Current PM State | | |
| **Range** | Not NULL | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | \*\*\* | |  | | | |
| imp | | no difference between the cores. | | | |
| Cnn | | no difference between the cores. | | | |
| Dma | | Difference is to describe the activity diagram. | | | |
| Ocv | | no difference between the cores. | | | |
| Psc | | no difference between the cores. | | | |
| Imps | | no difference between the cores. | | | |
| Dmas | | Difference is to describe the activity diagram. | | | |
| Dsp | | Difference is to describe the activity diagram. | | | |
| **Preconditions** | N/A. | | | | | | |
| **Remarks** | N/A | | | | | | |

#### impdrv\_dmactl\_mb\_init

Table 2‑140: impdrv\_dmactl\_mb\_init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0026  impdrv\_dmactl\_mb\_init  [Covers: AD\_PD\_CAS1024, AD\_PD\_CAS2229, AD\_PD\_CAS2469, AD\_PD\_CAS2470, AD\_PD\_CAS2472, AD\_PD\_CAS2473] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_dmactl\_mb\_init(  st\_impdrv\_corectl\_t \*const p\_corectl ,  const uint32\_t core\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | |
| **Range** | Not NULL |
| const uint32\_t core\_num | | Specified core number | |
| **Range** | Less than num of device. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Refer to “3 Activity Diagrams”.  Retrurn values IMPDRV\_EC\_NG\_CHECKFAIL, IMPDRV\_EC\_NG\_SEQSTATE and IMPDRV\_EC\_NG\_SYSTEMERROR are used only V3\*\*. | | | |
| **Preconditions** | N/A. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_\*\*\*ctl\_bus\_if\_check

Table 2‑141: impdrv\_\*\*\*ctl\_bus\_if\_check

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0079  impdrv\_\*\*\*ctl\_bus\_if\_check  [Covers: AD\_PD\_CAS2401, AD\_PD\_CAS2489, AD\_PD\_CAS2490, AD\_PD\_CAS2491, AD\_PD\_CAS2492 AD\_PD\_CAS2493 AD\_PD\_CAS2494 AD\_PD\_CAS2495 AD\_PD\_CAS2496 AD\_PD\_CAS2497 AD\_PD\_CAS2498 AD\_PD\_CAS2499, AD\_PD\_CAS2525, AD\_PD\_CAS2528, AD\_PD\_CAS2527] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_bus\_if\_check(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t  core\_num | | | | Specified core number  The lifetime of this parameter is until impdrv\_\*\*\*ctl\_init\_start returns. | |
| **Range** | Less than num of device |
| const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource | | | | Pointer to the OSAL resource for bus interface check. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for each core type.  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed until impdrv\_\*\*\*ctl\_stop is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check the IMP Core version | | |
| imp | | Difference is to describe the activity diagram. | | |
| Cnn | | N/A. | | |
| dma | | Difference is to describe the activity diagram. | | |
| Ocv | | N/A. | | |
| psc | | N/A. | | |
| imps | | Difference is to describe the activity diagram. | | |
| Dmas | | N/A. | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_conf\_reg\_check

Table 2‑142: impdrv\_\*\*\*ctl\_conf\_reg\_check

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0054  impdrv\_\*\*\*ctl\_conf\_reg\_check  [Covers: AD\_PD\_CAS2407, AD\_PD\_CAS2526, AD\_PD\_CAS2500, AD\_PD\_CAS2501] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_conf\_reg\_check(  const st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param,  const uint32\_t param\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | constst\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | | Control data structure for each core type. | |
| **Range** | Not NULL |
| const uint32\_t  core\_num | | | | Specified core number. | |
| **Range** | Less than num of device |
| const [st\_impdrv\_chk\_param\_t](#_st_impdrv_chkparam_t(T.B.D.)) \*const p\_chk\_param | | | | Pointer to Array of check parameters. | |
| **Range** | Not NULL |
| const uint32\_t param\_num | | | | Number of check parameters. | |
| **Range** | Not 0 |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check the IMP Core version | | |
| imp | | Difference is to describe the activity diagram. | | |
| Cnn | | Difference is to describe the activity diagram. | | |
| Dma | | Difference is to describe the activity diagram. | | |
| Ocv | | Difference is to describe the activity diagram. | | |
| Psc | | Difference is to describe the activity diagram. | | |
| Imps | | Difference is to describe the activity diagram. | | |
| Dmas | | Difference is to describe the activity diagram. | | |
| Dsp | | Difference is to describe the activity diagram. | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_dspctl\_dsp\_start\_pre

Table 2‑143: impdrv\_dspctl\_dsp\_start\_pre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0059  impdrv\_dspctl\_dsp\_start\_pre  [Covers: AD\_PD\_CAS2324, AD\_PD\_CAS2325, AD\_PD\_CAS2327, AD\_PD\_CAS2328, AD\_PD\_CAS2348, AD\_PD\_CAS2362] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_dspctl\_dsp\_start\_pre(  st\_impdrv\_corectl\_t \* p\_corectl,  const uint32\_t core\_num,  const e\_osal\_interrupt\_priority\_t irq\_priority,  const impdrv\_ctrl\_handle\_t osal\_cb\_args  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | Specified core number | | |
| **Range** | Less than num of device | |
| const e\_osal\_interrupt\_priority\_t irq\_priority | | The value of interrupt priority | | |
| **Range** | Within the range of ENUM type | |
| const impdrv\_ctrl\_handle\_t osal\_cb\_args | | OSAL callback argument  The lifetime of this parameter is the period from the impdrv\_genctl\_init is executed until impdrv\_genctl\_quit is executed. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \* p\_corectl | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Remove expression of processing Application update requested and not loaded | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

#### impdrv\_dspctl\_set\_dsp

Table 2‑144: impdrv\_dspctl\_set\_dsp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0060  impdrv\_dspctl\_set\_dsp  [Covers: AD\_PD\_CAS2418, AD\_PD\_CAS2426, AD\_PD\_CAS2427] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_dspctl\_set\_dsp(  st\_impdrv\_corectl\_t \* p\_corectl,  const uint32\_t core\_num,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_app,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_data,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_dtcm  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | Specified core number | | |
| **Range** | Less than num of device | |
| const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_app | | Information of DSP data | | |
| **Range** | Not NULL | |
| const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_data | | Information of DSP data | | |
| **Range** | Not NULL | |
| const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_dtcm | | Information of DSP data | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \* p\_corectl | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

#### impdrv\_\*\*\*ctl\_set\_cond\_gosub

Table 2‑149: impdrv\_\*\*\*ctl\_set\_cond\_gosub

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0058  impdrv\_\*\*\*ctl\_set\_cond\_gosub  [Covers: AD\_PD\_CAS2428, AD\_PD\_CAS2429, AD\_PD\_CAS2430] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | e\_impdrv\_errorcode\_t i impdrv\_\*\*\*ctl\_set\_cond\_gosub(  st\_impdrv\_corectl\_t \* const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_gosub\_cond\_t condition  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** |  | | | | | |
| **Parameters (In)** | const uint32\_t  core\_num | | | | Specified core number. | |
| **Range** | Less than num of device |
| const e\_impdrv\_gosub\_cond\_t condition | | | | Conditional GOSUB instruction attribute. | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \* const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | Check the IMP Core version | | |
| imp | | Difference is to describe the activity diagram. | | |
| Cnn | | Difference is to describe the activity diagram. | | |
| Dma | | Difference is to describe the activity diagram. | | |
| Ocv | | Difference is to describe the activity diagram. | | |
| Psc | | Difference is to describe the activity diagram. | | |
| Imps | | N/A | | |
| Dmas | | Difference is to describe the activity diagram. | | |
| Dsp | | N/A | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_dspctl\_dsp\_start

Table 2‑196: impdrv\_dspctl\_dsp\_start

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0073***  impdrv\_dspctl\_dsp\_start  ***[Covers: AD\_PD\_CAS2246, AD\_PD\_CAS2247, AD\_PD\_CAS2248, AD\_PD\_CAS2363, AD\_PD\_CAS2249, AD\_PD\_CAS2250, AD\_PD\_CAS2600, AD\_PD\_CAS2601, AD\_PD\_CAS2251]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_dsp\_start(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | Control data structure for each core type. | |
| **Range** | Not NULL |
| const uint32\_t  core\_num | | Specified core number. | |
| **Range** | 0 to 3 (only V4H)  0 to 1 (only V4M) |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_ATTRIBUTE  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_dsp\_execute

Table 2‑196: impdrv\_dspctl\_dsp\_execute

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0074***  impdrv\_dspctl\_dsp\_execute  ***[Covers: AD\_PD\_CAS2313, AD\_PD\_CAS2315, AD\_PD\_CAS2316, AD\_PD\_CAS2317, AD\_PD\_CAS2432, AD\_PD\_CAS2314, AD\_PD\_CAS2318, AD\_PD\_CAS2319, AD\_PD\_CAS2603, AD\_PD\_CAS2604, AD\_PD\_CAS2320, AD\_PD\_CAS1036]*** | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_dsp\_execute(  ***st\_impdrv\_corectl\_t*** \*p\_corectl,  const uint32\_t core\_num,  const ***p\_impdrv\_cbfunc\_t*** callback\_func,  void \*const p\_callback\_args  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | const uint32\_t  core\_num | | Core number. | | |
| **Range** | | 0 to 3 (only V4H)  0 to 1 (only V4M) |
| ***p\_impdrv\_cbfunc\_t***  callback\_func | | Callback function of execute request | | |
| **Range** | | Not NULL |
| void  \*const p\_callback\_args | | Callback arguments | | |
| **Range** | | N/A |
| **Parameters (In-out)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core. | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ATTRIBUTE  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

### Private function

#### impdrv\_\*\*\*ctl\_init\_core

Table 2‑144: impdrv\_\*\*\*ctl\_init\_core

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0076  impdrv\_\*\*\*ctl\_init\_core  [Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2250] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_init\_core(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | |  | | |
| imp | | Difference is to describe the activity diagram. | | |
| Cnn | | N/A | | |
| dma | | Refer to 2.4.2.2 | | |
| ocv | | Refer to 2.4.2.31 | | |
| Psc | | N/A | | |
| imps | | Difference is to describe the activity diagram. (refer to imp) | | |
| dmas | | Refer to 2.4.2.2 | | |
| Dsp | | Refer to 2.4.2.2 | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_init\_core (\*\*\*:dma/dmas)

Table 2‑145: impdrv\_\*\*\*ctl\_init\_core (\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0020  impdrv\_\*\*\*ctl\_init\_core  [Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2250] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_init\_core(  st\_impdrv\_corectl\_t \* const p\_corectl  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | st\_impdrv\_corectl\_t  \* const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| N/A | | | | N/A | |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
|  | Differences between cores. | \*\*\* | |  | | |
|  |  | dma | | no difference between the cores. | | |
|  |  | Dmas | | no difference between the cores. | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_get\_inten\_val

Table 2‑146: impdrv\_\*\*\*ctl\_get\_inten\_val

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0025  impdrv\_\*\*\*ctl\_get\_inten\_val  [Covers: AD\_PD\_CAS2315] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | IMPDRV\_STATIC  uint32\_t impdrv\_\*\*\*ctl\_get\_inten\_val(  const bool irq\_mask[IMPDRV\_IRQMASK\_MAX]  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const bool irq\_mask[ IMPDRV\_IRQMASK\_MAX] | | | | IRQ mask setting value | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | uint32\_t | | | | Interrupt enable register value | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | | interrupt enable value | | |
| imp | | (R-CarV3M)  inten\_val |= (IMPDRV\_IMP\_INTSTS\_TRAP | IMPDRV\_IMP\_INTSTS\_IER | IMPDRV\_IMP\_INTSTS\_INT);  (R-CarV3H / R-CarV3H\_2)  inten\_val |= (IMPDRV\_IMP\_INTSTS\_TRAP | IMPDRV\_IMP\_INTSTS\_IER | IMPDRV\_IMP\_INTSTS\_INT | IMPDRV\_IMP\_INTSTS\_WUP); | | |
| cnn | | (R-CarV3M)  inten\_val = (IMPDRV\_CNN\_SR\_TRAP | IMPDRV\_CNN\_SR\_IER | IMPDRV\_CNN\_SR\_INT);  (R-CarV3H / R-CarV3H\_2)  inten\_val = (IMPDRV\_CNN\_SR\_TRAP | IMPDRV\_CNN\_SR\_IER | IMPDRV\_CNN\_SR\_INT | IMPDRV\_CNN\_SR\_WUPCOVF); | | |
| dma | | (R-CarV3M)  inten\_val = (IMPDRV\_DMA\_SR\_TRAP | IMPDRV\_DMA\_SR\_IER | IMPDRV\_DMA\_SR\_INT);  (R-CarV3H / R-CarV3H\_2)  inten\_val = (IMPDRV\_DMA\_SR\_TRAP | IMPDRV\_DMA\_SR\_IER | IMPDRV\_DMA\_SR\_INT | IMPDRV\_DMA\_SR\_WUP); | | |
| ocv | | (R-CarV3M)  inten\_val = (IMPDRV\_OCV\_SR1\_TRAP | IMPDRV\_OCV\_SR1\_IER | IMPDRV\_OCV\_SR1\_INT);  (R-CarV3H / R-CarV3H\_2)  inten\_val = (IMPDRV\_OCV\_SR1\_TRAP | IMPDRV\_OCV\_SR1\_IER | IMPDRV\_OCV\_SR1\_INT | IMPDRV\_OCV\_SR1\_WUPCOVF); | | |
| psc | | (R-CarV3M / R-CarV3H)  inten\_val = (IMPDRV\_PSC\_SR\_TRAP | IMPDRV\_PSC\_SR\_IER | IMPDRV\_PSC\_SR\_INT);  (R-CarV3H\_2)  inten\_val = (IMPDRV\_PSC\_SR\_TRAP | IMPDRV\_PSC\_SR\_IER | IMPDRV\_PSC\_SR\_INT | IMPDRV\_PSC\_SR\_WUP); | | |
| imps | | ((R-CarV3H /R-CarV3H\_2)  inten\_val = IMPDRV\_IMPS\_INTSTS\_MASKCPU;  inten\_val |= (IMPDRV\_IMPS\_INTSTS\_TRAP | IMPDRV\_IMPS\_INTSTS\_IER | IMPDRV\_IMPS\_INTSTS\_INT | IMPDRV\_IMPS\_INTSTS\_WUP); | | |
| dmas | | (R-CarV3H\_2)  inten\_val = (IMPDRV\_DMAS\_SR\_TRAP | IMPDRV\_DMAS\_SR\_IER | IMPDRV\_DMAS\_SR\_INT | IMPDRV\_DMAS\_SR\_WUP); | | |
| **Preconditions** | N/A. | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_is\_sub\_thread

Table 2‑147: impdrv\_\*\*\*ctl\_is\_sub\_thread

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0021  impdrv\_\*\*\*ctl\_is\_sub\_thread  [Covers: AD\_PD\_CAS2034, AD\_PD\_CAS2039] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | IMPDRV\_STATIC  bool impdrv\_\*\*\*ctl\_is\_sub\_thread(  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Core number in the core type  The lifetime of this parameter is until this function returns. | |
| **Range** | DMAC core number (0 to 3) |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | bool | | | | true : sub thread  false : main thread | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | |  | | |
| imp | | N/A | | |
| cnn | | N/A | | |
| dma | | no difference between the cores. | | |
| Ocv | | N/A | | |
| psc | | N/A | | |
| imps | | N/A | | |
| dmas | | no difference between the cores. | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_osal\_core\_num

Table 2‑148: impdrv\_\*\*\*ctl\_osal\_core\_num

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0022  impdrv\_\*\*\*ctl\_osal\_core\_num  [Covers: AD\_PD\_CAS2034] | | | | | |
| **ASIL Level** | ASIL D | | **Status** | | New | |
| **Syntax** | IMPDRV\_STATIC  uint32\_t impdrv\_\*\*\*ctl\_osal\_core\_num(  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Reentrant | | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const uint32\_t core\_num | | | | Core number in the core type  The lifetime of this parameter is until this function returns. | |
| **Range** | DMAC core number (0 to 3) |
| **Parameters (In-out)** | N/A | | | | N/A | |
| **Parameters (Out)** | N/A | | | | N/A | |
| **Return Value** | uint32\_t | | | | OSAL resource number (0 or 1 or (uint32\_t)-1)) | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| Differences between cores. | \*\*\* | |  | | |
| imp | | N/A | | |
| cnn | | N/A | | |
| dma | | Different number of cores. | | |
| Ocv | | N/A | | |
| psc | | N/A | | |
| imps | | N/A | | |
| dmas | | Different number of cores. | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_read\_reg

Table 2‑149: impdrv\_\*\*\*ctl\_read\_reg

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0023  impdrv\_\*\*\*ctl\_read\_reg  [Covers: AD\_PD\_CAS2036] | | | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_read\_reg(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const uint32\_t offset,  uint32\_t \*p\_val,  const bool is\_sub\_theard  ); | | | | | | | |
| **Sync/Async** | | Synchronous | | | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | | | |
| **Parameters (In)** | | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_\*\*\*ctl\_init\_start is executed until impdrv\_\*\*\*ctl\_init\_stop is executed. | | | |
| **Range** | | Not NULL | |
| const uint32\_t offset | | | | Offset to get to register  The lifetime of this parameter is until this function returns. | | | |
| **Range** | | | None. |
| Const bool is\_sub\_theard | | | | Select for DMAC Internal thread  The lifetime of this parameter is until this function returns. | | | |
| **Range** | | | true or false |
| **Parameters (In-out)** | | N/A | | | | N/A | | | |
| **Parameters (Out)** | | uint32\_t \*p\_val | | | | Storing values from registers | | | |
| **Range** | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | | | |
| Differences between cores. | | \*\*\* | |  | | | | |
| imp | | N/A | | | | |
| cnn | | N/A | | | | |
| dma | | no difference between the cores. | | | | |
| Ocv | | N/A | | | | |
| psc | | N/A | | | | |
| imps | | N/A | | | | |
| dmas | | no difference between the cores. | | | | |
| **Preconditions** | | N/A | | | | | | | |
| **Remarks** | | p\_val is pointer of output parameter, so the use of this pointer variable is acceptable | | | | | | | |

#### impdrv\_\*\*\*ctl\_write\_reg

Table 2‑150: impdrv\_\*\*\*ctl\_write\_reg

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0027  impdrv\_\*\*\*ctl\_write\_reg  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_write\_reg(  const [st\_impdrv\_dma\_write\_info\_t](#_st_impdrv_dma_write_info_t) \*const p\_write\_info ,  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const [***st\_impdrv\_dma\_write\_info\_t***](#_st_impdrv_dma_write_info_t) \*const p\_write\_info | | | | Write register information. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | | N/A | | | | N/A | |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | |  | | |
| imp | | N/A | | |
| cnn | | N/A | | |
| dma | | no difference between the cores. | | |
| Ocv | | N/A | | |
| psc | | N/A | | |
| imps | | N/A | | |
| dmas | | no difference between the cores. | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_dmactl\_tbl\_write\_reg

Table 2‑151: impdrv\_dmactl\_tbl\_write\_reg

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0028  impdrv\_dmactl\_tbl\_write\_reg  [Covers: AD\_PD\_CAS2470] | | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dmactl\_tbl\_write\_reg(  st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread,  const [st\_impdrv\_dma\_write\_info\_tbl\_t](#_st_impdrv_dma_write_info_tbl_t) write\_info\_tbl[],  const uint32\_t write\_info\_tbl\_num  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core, | | | |
| **Range** | Not NULL | | |
| const bool is\_sub\_thread | | Thread selection, | | | |
| **Range** | | true or false | |
| const [st\_impdrv\_dma\_write\_info\_tbl\_t](#_st_impdrv_dma_write_info_tbl_t) write\_info\_tbl[] | | Write info table, | | | |
| **Range** | | | None. |
| Const uint32\_t  write\_info\_tbl\_num | | write\_info\_tbl\_num Number of writing info tables | | | |
| **Range** | | | None. |
| **Parameters (In-out)** | | N/A | | N/A | | | |
| **Parameters (Out)** | | N/A | | N/A | | | |
| **Return Value** | | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_dmactl\_mb\_init\_pre

Table 2‑152: impdrv\_dmactl\_mb\_init\_pre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0029  impdrv\_dmactl\_mb\_init\_pre  [Covers: AD\_PD\_CAS2470] | | | |
| **ASIL Level** | | ASIL D | **Status** | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dmactl\_mb\_init\_pre(  st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread,  ); | | | |
| **Sync/Async** | | Synchronous | | | |
| **Reentrancy** | | Reentrant | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core, | |
| **Range** | Not NULL |
| const bool is\_sub\_thread, | | Thread selection, | |
| **Range** | true or false |
| **Parameters (In-out)** | | N/A | | N/A | |
| **Parameters (Out)** | | N/A | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | | N/A | | | |
| **Remarks** | | N/A | | | |

#### impdrv\_dmactl\_mb\_init\_main

Table 2‑153: impdrv\_dmactl\_mb\_init\_main

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0030  impdrv\_dmactl\_mb\_init\_main  [Covers: AD\_PD\_CAS2470, AD\_PD\_CAS2472] | | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dmactl\_mb\_init\_main(  st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread,  const uint32\_t start\_addr,  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core, | | | |
| **Range** | Not NULL | | |
| const bool is\_sub\_thread, | | Thread selection, | | | |
| **Range** | | true or false | |
| const uint32\_t start\_addr, | | Set the start address(odd bank / even bank) | | | |
| **Range** | | | None. |
| **Parameters (In-out)** | | N/A | | N/A | | | |
| **Parameters (Out)** | | N/A | | N/A | | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | | | |
| **Description** | | Refer to “3 Activity Diagrams”. | |  | | | |
| **Preconditions** | N/A | | | | | | |
| **Remarks** | | N/A | | | | | |
|  | |  | | | | | |

#### impdrv\_dmactl\_cl\_pre

Table 2‑154: impdrv\_dmactl\_cl\_pre

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0031  impdrv\_dmactl\_cl\_pre  [Covers: AD\_PD\_CAS2316, AD\_PD\_CAS2317] | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dmactl\_cl\_pre(  st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread  ); | | | | |
| **Sync/Async** | | Synchronous | | | | |
| **Reentrancy** | | Reentrant | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core, | | |
| **Range** | Not NULL | |
| const bool is\_sub\_thread, | | Thread selection, | | |
| **Range** | | true or false |
| **Parameters (In-out)** | | N/A | | N/A | | |
| **Parameters (Out)** | | N/A | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | | N/A | | | | |
| **Remarks** | | N/A | | | | |

#### impdrv\_dmactl\_int\_safety\_func

Table 2‑156: impdrv\_dmactl\_int\_safety\_func

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0037  impdrv\_dmactl\_int\_safety\_func  [Covers: AD\_PD\_CAS2352, AD\_PD\_CAS2353] | | | |
| **ASIL Level** | | ASIL D | **Status** | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dmactl\_int\_safety\_func (  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  ); | | | |
| **Sync/Async** | | Synchronous | | | |
| **Reentrancy** | | Reentrant | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core, | |
| **Range** | Not NULL |
| const uint32\_t core\_num | | Core number. | |
| **Range** | Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-out)** | | N/A | | N/A | |
| **Parameters (Out)** | | N/A | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | | N/A | | | |
| **Remarks** | | N/A | | | |

#### impdrv\_\*\*\*ctl\_int\_main\_func

Table 2‑155: impdrv\_\*\*\*ctl\_int\_main\_func

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0036  impdrv\_\*\*\*ctl\_int\_main\_func  [Covers: AD\_PD\_CAS2351, AD\_PD\_CAS2352, AD\_PD\_CAS2353, AD\_PD\_CAS2354, AD\_PD\_CAS2311, AD\_PD\_CAS2361, AD\_PD\_CAS2355, AD\_PD\_CAS2356] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_int\_main\_func (  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const uint32\_t core\_num | | | | Core number. | |
| **Range** | Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-out)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core, | |
| **Range** | Not NULL |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | |  | | |
| imp | | no difference between the cores. | | |
| Cnn | | N/A | | |
| dma | | no difference between the cores. | | |
| Ocv | | N/A | | |
| psc | | N/A | | |
| imps | | no difference between the cores. | | |
| Dmas | | N/A | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_int\_safety\_func

Table 2‑156: impdrv\_\*\*\*ctl\_int\_safety\_func

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0078  impdrv\_\*\*\*ctl\_int\_safety\_func  [Covers: AD\_PD\_CAS2352, AD\_PD\_CAS2353] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_int\_safety\_func (  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core, | |
| **Range** | Not NULL |
| const uint32\_t core\_num | | | | Core number. | |
| **Range** | Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-out)** | | N/A | | | | N/A | |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | |  | | |
| imp | | no difference between the cores. | | |
| Cnn | | N/A | | |
| dma | | Refer to 2.4.2.12 | | |
| Ocv | | N/A | | |
| psc | | N/A | | |
| imps | | no difference between the cores. | | |
| Dmas | | N/A | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_dma\_transfer

Table 2‑157: impdrv\_\*\*\*ctl\_dma\_transfer

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0038  impdrv\_\*\*\*ctl\_dma\_transfer  [Covers: AD\_PD\_CAS2527] | | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_dma\_transfer  (  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource,  const uintptr\_t physical\_adrs,  const uint32\_t transfer\_size  ); | | | | | | |
| **Sync/Async** | | Synchronous | | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | | const uint32\_t core\_num | | | Specified core number. | | | |
| **Range** | | Value less than IMPDRV\_CORE\_NUM\_MAX | |
| const [st\_impdrv\_chk\_resource\_t](#_st_impdrv_chk_resource_t) \*const p\_chk\_resource | | | Pointer to the OSAL resource for bus interface check. | | | |
| **Range** | | Not NULL | |
| const uintptr\_t physical\_adrs | | | Physical address of the forwarding source | | | |
| **Range** | | None | |
| const uint32\_t transfer\_size | | | Transfer size | | | |
| **Range** | | None | |
| **Parameters (In-out)** | | st\_impdrv\_coretypectl\_t \*const p\_coretypectl | | | Control data structure for each core type | | | |
| **Range** | | | Not NULL |
| **Parameters (Out)** | | N/A | | | N/A | | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | | |
| Differences between cores. | | \*\*\* | | | N/A | | |
| imp | | | no difference between the cores. | | |
| Cnn | | | N/A | | |
| dma | | | no difference between the cores. | | |
| Ocv | | | N/A | | |
| psc | | | N/A | | |
| imps | | | no difference between the cores. | | |
| Dmas | | | N/A | | |
| **Preconditions** | | N/A | | | | | | |

#### impdrv\_\*\*\*ctl\_tbl\_write\_reg

Table 2‑158: impdrv\_\*\*\*ctl\_tbl\_write\_reg

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0033  impdrv\_\*\*\*ctl\_tbl\_write\_reg  [Covers: AD\_PD\_CAS2527] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_tbl\_write\_reg(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const st\_impdrv\_imp\_write\_info\_tbl\_t write\_info\_tbl[],  const uint32\_t write\_info\_tbl\_num  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| const st\_impdrv\_imp\_write\_info\_tbl\_t write\_info\_tbl[] | | | | Write info table | |
| **Range** | Not NULL |
| const uint32\_t write\_info\_tbl\_num | | | | Number of writing info tables | |
| **Range** | N/A |
| **Parameters (In-out)** | | N/A | | | | N/A | |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | |  | | |
| imp | | no difference between the cores. | | |
| Cnn | | N/A | | |
| dma | | Refer to 2.4.2.8 | | |
| ocv | | N/A | | |
| psc | | N/A | | |
| imps | | no difference between the cores. | | |
| Dmas | | N/A | | |
| **Preconditions** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_check\_core\_busy

Table 2‑159: impdrv\_\*\*\*ctl\_check\_core\_busy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0039  impdrv\_\*\*\*ctl\_check\_core\_busy  [Covers: AD\_PD\_CAS2316] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_check\_core\_busy(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (In-out)** | | N/A | | | | N/A | |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | |  | | |
| imp | | no difference between the cores. | | |
| Cnn | | no difference between the cores. | | |
| Dma | | Refer to 2.4.2.17 | | |
| ocv | | no difference between the cores. | | |
| Psc | | no difference between the cores. | | |
| Imps | | no difference between the cores. | | |
| Dmas | | Refer to 2.4.2.17 | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_check\_core\_busy (\*\*\*:dma/dmas)

Table 2‑160: impdrv\_\*\*\*ctl\_check\_core\_busy (\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0040  impdrv\_\*\*\*ctl\_check\_core\_busy  [Covers: AD\_PD\_CAS2316] | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_check\_core\_busy(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread  ); | | | | |
| **Sync/Async** | | Synchronous | | | | |
| **Reentrancy** | | Non-Reentrant | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core | |
| **Range** | Not NULL |
| const bool is\_sub\_thread | | | Thread selection | |
| **Range** | true or false |
| **Parameters (In-out)** | | N/A | | | N/A | |
| **Parameters (Out)** | | N/A | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
|  | Differences between cores. | | \*\*\* |  | | |
|  |  | | dma | no difference between the cores. | | |
|  |  | | Dmas | no difference between the cores. | | |
| **Preconditions** | | N/A | | | | |
| **Remarks** | | N/A | | | | |

#### impdrv\_\*\*\*ctl\_soft\_reset

Table 2‑161: impdrv\_\*\*\*ctl\_soft\_reset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0041  impdrv\_\*\*\*ctl\_soft\_reset  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_soft\_reset(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | | |  | |
| imp | | | Difference is to describe the activity diagram. | |
| Cnn | | | Difference is to describe the activity diagram. | |
| Dma | | | Refer to 2.4.1.22 | |
| ocv | | | Difference is to describe the activity diagram. | |
| Psc | | | Difference is to describe the activity diagram. | |
| Imps | | | Difference is to describe the activity diagram. | |
| Dmas | | | Refer to 2.4.1.22 | |
| Dsp | | | N/A | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_soft\_reset (\*\*\*:dma/dmas)

Table 2‑162: impdrv\_\*\*\*ctl\_soft\_reset (\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0042  impdrv\_\*\*\*ctl\_soft\_reset  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | **Status** | | | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_soft\_reset(  st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | st\_impdrv\_corectl\_t \*const p\_corectl | | | | Control data structure for specific core | |
| **Range** | Not NULL |
| const bool is\_sub\_thread | | | | Thread selection | |
| **Range** | true or false |
| **Parameters (In-out)** | | N/A | | | | N/A | |
| **Parameters (Out)** | | N/A | | | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
|  | Differences between cores. | | | \*\*\* |  | | |
|  |  | | | dma | no difference between the cores. | | |
|  |  | | | Dmas | no difference between the cores. | | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_set\_syncc\_config

Table 2‑163: impdrv\_\*\*\*ctl\_set\_syncc\_config

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0043  impdrv\_\*\*\*ctl\_set\_syncc\_config  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_syncc\_config(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | | |  | |
| imp | | | no difference between the cores. | |
| Cnn | | | no difference between the cores. | |
| Dma | | | Refer to 2.4.2.21 | |
| ocv | | | no difference between the cores. | |
| Psc | | | no difference between the cores. | |
| Imps | | | no difference between the cores. | |
| Dmas | | | Refer to 2.4.2.21 | |
| Dsp | | | N/A | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_set\_syncc\_config (\*\*\*:dma/dmas)

Table 2‑164: impdrv\_\*\*\*ctl\_set\_syncc\_config (\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0044  impdrv\_\*\*\*ctl\_set\_syncc\_config  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | **Status** | | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_syncc\_config(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const bool is\_sub\_thread  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core | | |
| **Range** | | Not NULL |
| const bool is\_sub\_thread | | | Thread selection | | |
| **Range** | | true or false |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
|  | Differences between cores. | | | \*\*\* | |  | |
|  |  | | | dma | | no difference between the cores. | |
|  |  | | | Dmas | | no difference between the cores. | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_int\_cb

Table 2‑165: impdrv\_\*\*\*ctl\_int\_cb

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0045  impdrv\_\*\*\*ctl\_int\_cb  [Covers: AD\_PD\_CAS2351] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_int\_cb(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  const uint32\_t cb\_ercd\_num,  const bool is\_check\_intclear,  const uint32\_t dtl\_stat\_val  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core. | | |
| **Range** | | Not NULL |
| const uint32\_t core\_num | | | Specified core number. | | |
| **Range** | | Value less than IMPDRV\_\*\*\*\_CORENUM\_VAL |
| const e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | | Callback error code. | | |
| **Range** | | Not NULL |
| const uint32\_t cb\_ercd\_num | | | Number of error codes in callback. | | |
| **Range** | | Value less than IMPDRV\_CB\_RET\_MAX |
| const bool is\_check\_intclear | | | Interrupt clear wait execution flag. | | |
| **Range** | | true or false |
| const uint32\_t dtl\_stat\_val | | | Detail status value. | | |
| **Range** | | None |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | | | N/A | |
| imp | | | Refer to 2.4.2.23 | |
| cnn | | | no difference between the cores. | |
| Dma | | | no difference between the cores. | |
| Ocv | | | no difference between the cores. | |
| Psc | | | no difference between the cores. | |
| Imps | | | Refer to 2.4.2.23 | |
| dmas | | | no difference between the cores. | |
| Dsp | | | N/A | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_int\_cb (\*\*\*:imp/imps)

Table 2‑166: impdrv\_\*\*\*ctl\_int\_cb (\*\*\*:imp/imps)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0046  impdrv\_\*\*\*ctl\_int\_cb  [Covers: AD\_PD\_CAS2351] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_int\_cb(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  const uint32\_t cb\_ercd\_num,  const bool is\_check\_intclear,  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core. | | |
| **Range** | | Not NULL |
| const uint32\_t core\_num | | | Specified core number. | | |
| **Range** | | Value less than IMPDRV\_\*\*\*\_CORENUM\_VAL |
| const e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | | Callback error code. | | |
| **Range** | | Not NULL |
| const uint32\_t cb\_ercd\_num | | | Number of error codes in callback. | | |
| **Range** | | Value less than IMPDRV\_CB\_RET\_MAX |
| const bool is\_check\_intclear | | | Interrupt clear wait execution flag. | | |
| **Range** | | true or false |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_NOTSUPPORT | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
|  | Differences between cores. | | \*\*\* | | |  | |
|  |  | | imp | | | no difference between the cores. | |
|  |  | | Imps | | | no difference between the cores. | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_check\_inten

Table 2‑167: impdrv\_\*\*\*ctl\_check\_inten

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0047  impdrv\_\*\*\*ctl\_check\_inten  [Covers: AD\_PD\_CAS2351] | | | | | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | | | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_check\_inten(  const uint32\_t stat\_val,  uint32\_t \*const p\_mask\_val,  e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  uint32\_t \*const p\_cb\_ercd\_num,  bool \*const p\_is\_check\_intclear,  e\_impdrv\_state\_t \*const p\_state  ); | | | | | | | | | |
| **Sync/Async** | | Synchronous | | | | | | | | | |
| **Reentrancy** | | Reentrant | | | | | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | | | | | |
| **Parameters (In)** | | const uint32\_t stat\_val | | | Status register value. | | | | | | |
| **Range** | None | | | | | |
| **Parameters (In-out)** | | N/A | | | N/A | | | | | | |
| **Parameters (Out)** | | uint32\_t \*const p\_mask\_val | | | Interrupt mask value. | | | | | | |
| **Range** | | | Not NULL | | | |
| e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | | Callback error code. | | | | | | |
| **Range** | | Not NULL | | | | |
| uint32\_t \*const p\_cb\_ercd\_num | | | Number of error codes in callback. | | | | | | |
| **Range** | | | | Not NULL | | |
| bool \*const p\_is\_check\_intclear | | | Interrupt clear wait execution flag. | | | | | | |
| **Range** | | | | | | Not NULL |
| e\_impdrv\_state\_t \*const p\_state | | | IMP Driver state to be transitioned. | | | | | | |
| **Range** | | | | | Not NULL | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | | | | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | | | | | |
| Differences between cores. | | \*\*\* | |  | | | | | | |
| imp | | Difference is to describe the activity diagram. | | | | | | |
| Cnn | | Difference is to describe the activity diagram. | | | | | | |
| Dma | | Difference is to describe the activity diagram. | | | | | | |
| Ocv | | Refer to 2.4.2.25 | | | | | | |
| psc | | Difference is to describe the activity diagram. | | | | | | |
| Imps | | Difference is to describe the activity diagram.. | | | | | | |
| Dmas | | Difference is to describe the activity diagram. | | | | | | |
| **Preconditions** | | N/A | | | | | | | | | |
| **Remarks** | | p\_mask\_val, cb\_ercd, p\_cb\_ercd\_num, p\_is\_check\_intclear and p\_state arre pointer of output parameter, so the use of those pointer variable is acceptable. | | | | | | | | | |

#### impdrv\_ocvctl\_check\_inten

Table 2‑168: impdrv\_ocvctl\_check\_inten

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0048  impdrv\_ocvctl\_check\_inten  [Covers: AD\_PD\_CAS2351] | | | | | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | | | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_check\_inten(  const uint32\_t stat\_sr1,  const uint32\_t stat\_sr2,  uint32\_t \*const p\_mask\_val,  e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  uint32\_t \*const p\_cb\_ercd\_num,  bool \*const p\_is\_check\_intclear,  e\_impdrv\_state\_t \*const p\_state  ); | | | | | | | | |
| **Sync/Async** | | Synchronous | | | | | | | | |
| **Reentrancy** | | Non-Reentrant | | | | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | | | | |
| **Parameters (In)** | | const uint32\_t stat\_sr1 | | SR1 register value. | | | | | | |
| **Range** | None | | | | | |
| const uint32\_t stat\_sr2 | | SR2 register value. | | | | | | |
| **Range** | None | | | | | |
| **Parameters (In-out)** | | N/A | | N/A | | | | | | |
| **Parameters (Out)** | | uint32\_t \*const p\_mask\_val | | Interrupt mask value. | | | | | | |
| **Range** | | | Not NULL | | | |
| e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | Callback error code. | | | | | | |
| **Range** | | Not NULL | | | | |
| uint32\_t \*const p\_cb\_ercd\_num | | Number of error codes in callback. | | | | | | |
| **Range** | | | | Not NULL | | |
| bool \*const p\_is\_check\_intclear | | Interrupt clear wait execution flag. | | | | | | |
| **Range** | | | | | | Not NULL |
| e\_impdrv\_state\_t \*const p\_state | | IMP Driver state to be transitioned. | | | | | | |
| **Range** | | | | | Not NULL | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | | | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | | | | |
| **Preconditions** | | N/A | | | | | | | | |
| **Remarks** | | p\_mask\_val, cb\_ercd, p\_cb\_ercd\_num, p\_is\_check\_intclear and p\_state arre pointer of output parameter, so the use of those pointer variable is acceptable. | | | | | | | | |

#### impdrv\_ocvctl\_irq\_status\_clear

Table 2‑169: impdrv\_ocvctl\_irq\_status\_clear

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0049  impdrv\_ocvctl\_irq\_status\_clear  [Covers: AD\_PD\_CAS2317] | | | |
| **ASIL Level** | | ASIL D | **Status** | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_irq\_status\_clear(  const st\_impdrv\_corectl\_t \*const p\_corectl,  uint32\_t mask\_val  ); | | | |
| **Sync/Async** | | Synchronous | | | |
| **Reentrancy** | | Reentrant | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | |
| **Range** | Not NULL |
| uint32\_t mask\_val | | IRQ status mask value | |
| **Range** | None |
| **Parameters (In-out)** | | N/A | | N/A | |
| **Parameters (Out)** | | N/A | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | | N/A | | | |
| **Remarks** | | N/A | | | |

#### impdrv\_ocvctl\_check\_inten\_1st

Table 2‑170: impdrv\_ocvctl\_check\_inten\_1st

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0050  impdrv\_ocvctl\_check\_inten\_1st  [Covers: AD\_PD\_CAS2351] | | | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_check\_inten\_1st(  const uint32\_t stat\_sr1,  const uint32\_t stat\_sr2,  uint32\_t \*const p\_mask\_val,  e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  uint32\_t \*const p\_cb\_ercd\_num,  bool \*const p\_is\_check\_intclear,  e\_impdrv\_state\_t \*const p\_state  ); | | | | | | |
| **Sync/Async** | | Synchronous | | | | | | |
| **Reentrancy** | | Reentrant | | | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | | | |
| **Parameters (In)** | | const uint32\_t stat\_sr1 | | SR1 register value. | | | | |
| **Range** | None | | | |
| const uint32\_t stat\_sr2 | | SR2 register value. | | | | |
| **Range** | None | | | |
| **Parameters (In-out)** | | uint32\_t \*const p\_mask\_val | | Interrupt mask value. | | | | |
| **Range** | Not NULL | | | |
| uint32\_t \*const p\_cb\_ercd\_num | | Number of error codes in callback. | | | | |
| **Range** | Not NULL | | | |
| **Parameters (Out)** | | e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | Callback error code. | | | | |
| **Range** | | Not NULL | | |
| bool \*const p\_is\_check\_intclear | | Interrupt clear wait execution flag. | | | | |
| **Range** | | | | Not NULL |
| e\_impdrv\_state\_t \*const p\_state | | IMP Driver state to be transitioned. | | | | |
| **Range** | | | Not NULL | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | | |
| **Preconditions** | | N/A | | | | | | |
| **Remarks** | | p\_mask\_val, cb\_ercd, p\_cb\_ercd\_num, p\_is\_check\_intclear and p\_state are pointer of output parameter, so the use of those pointer variable is acceptable. | | | | | | |

#### impdrv\_ocvctl\_check\_inten\_2nd

Table 2‑171: impdrv\_ocvctl\_check\_inten\_2nd

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0051  impdrv\_ocvctl\_check\_inten\_2nd  [Covers: AD\_PD\_CAS2351] | | | | |
| **ASIL Level** | | ASIL D | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_check\_inten\_2nd(  const uint32\_t stat\_sr1,  e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX],  uint32\_t \*const p\_cb\_ercd\_num,  e\_impdrv\_state\_t \*const p\_state  ); | | | | |
| **Sync/Async** | | Synchronous | | | | |
| **Reentrancy** | | Reentrant | | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | | const uint32\_t stat\_sr1 | | SR1 register value. | | |
| **Range** | None | |
| **Parameters (In-out)** | | uint32\_t \*const p\_cb\_ercd\_num | | Number of error codes in callback. | | |
| **Range** | Not NULL | |
| e\_impdrv\_state\_t \*const p\_state | | IMP Driver state to be transitioned. | | |
| **Range** | Not NULL | |
| **Parameters (out)** | | e\_impdrv\_cb\_ret\_t cb\_ercd[IMPDRV\_CB\_RET\_MAX] | | Callback error code. | | |
| **Range** | | Not NULL |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | | N/A | | | | |
| **Remarks** | | p\_mask\_val, cb\_ercd, p\_cb\_ercd\_num, and p\_state are pointer of output parameter, so the use of those pointer variable is acceptable. | | | | |

#### impdrv\_ocvctl\_init\_lwm

Table 2‑172: impdrv\_ocvctl\_init\_lwm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0052  impdrv\_ocvctl\_init\_lwm  [Covers: AD\_PD\_CAS2249] | | | |
| **ASIL Level** | | ASIL D | **Status** | New | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_init\_lwm(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | |
| **Sync/Async** | | Synchronous | | | |
| **Reentrancy** | | Non-Reentrant | | | |
| **Interrupt State** | | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (In-out)** | | N/A | | N/A | |
| **Parameters (Out)** | | N/A | | N/A | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | | N/A | | | |
| **Remarks** | | V3H2 is not supported. | | | |

#### impdrv\_ocvctl\_init\_core

Table 2‑144: impdrv\_ocvctl\_init\_core

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0019  impdrv\_ocvctl\_init\_core  [Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2250] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_ocvctl\_init\_core(  const st\_impdrv\_corectl\_t \*const p\_corectl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_dspctl\_dsp\_update\_app

Table 2‑173: impdrv\_dspctl\_dsp\_update\_app

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0073  impdrv\_dspctl\_dsp\_update\_app  [Covers: AD\_PD\_CAS2315, AD\_PD\_CAS2316, AD\_PD\_CAS2317, AD\_PD\_CAS2432, AD\_PD\_CAS2246, AD\_PD\_CAS2363, AD\_PD\_CAS2249, AD\_PD\_CAS2250] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_dsp\_update\_app (  st\_impdrv\_corectl\_t \*const p\_corectl,  st\_impdrv\_device\_handle\_t \*const p\_device\_io  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t \*const p\_device\_io | | Device io handle for DSP | | |
| **Range** | | Not Null |
| **Parameters (In-out)** | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | | |
| **Range** | Not Null | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]  <Only DSP core>   * Add expression that whether DSP information (is\_update, is\_loaded and is\_executed) is true or not when call impdrv\_dspctl\_dsp\_foece\_standby * Add expression that whether DSP information (is\_update, is\_loaded and is\_executed) is true or not when call impdrv\_dspctl\_load\_ptcm | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

#### impdrv\_dspctl\_load\_ptcm

Table 2‑174: impdrv\_dspctl\_load\_ptcm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0065  impdrv\_dspctl\_load\_ptcm  [Covers: AD\_PD\_CAS2246, AD\_PD\_CAS2363, AD\_PD\_CAS2249, AD\_PD\_CAS2250, AD\_PD\_CAS2315, AD\_PD\_CAS2316, AD\_PD\_CAS2317, AD\_PD\_CAS2432] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_load\_ptcm (  const st\_impdrv\_corectl\_t \*const p\_corectl,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core | |
| **Range** | Not Null |
| st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP | |
| **Range** | Not Null |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_ATTRIBUTE  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * For processing of Start from the vector address RST\_VEC\_ADDR, Include expression of whether ercd is IMPDRV\_EC\_OK or not in the previous expression * For processing of Transfer the boot program from external to internal program, remove expression of whether ercd is IMPDRV\_EC\_OK or not | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_stop\_reg

Table 2‑175: impdrv\_dspctl\_stop\_reg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0069  impdrv\_dspctl\_stop\_reg  [Covers: AD\_PD\_CAS2364] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_stop\_reg(  const bool is\_executed,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const bool is\_executed | | Is the DSP application already executed on DSP | |
| **Range** | true or false |
| st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP | |
| **Range** | Not Null |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * For processing of Clear GPIO\_IN register, remove expression of whether ercd is IMPDRV\_EC\_OK or not | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_\*\*\*ctl\_set\_extend\_config

Table 2‑177: impdrv\_\*\*\*ctl\_set\_extend\_config

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0056  impdrv\_\*\*\*ctl\_set\_extend\_config  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_extend\_config(  const st\_impdrv\_corectl\_t \*const p\_corectll  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | |  | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core. | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_SEQSTATE | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | | | N/A | |
| imp | | | no difference between the cores. | |
| cnn | | | no difference between the cores. | |
| Dma | | | Refer to 2.4.2.34 | |
| Ocv | | | no difference between the cores. | |
| Psc | | | no difference between the cores. | |
| Imps | | | N/A. | |
| dmas | | | Refer to 2.4.2.34 | |
| dsp | | | N/A | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_\*\*\*ctl\_set\_extend\_config (\*\*\*:dma/dmas)

Table 2‑178: impdrv\_\*\*\*ctl\_set\_extend\_config(\*\*\*:dma/dmas)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | | UD\_PD\_UD02\_04\_0057  impdrv\_\*\*\*ctl\_set\_extend\_config  [Covers: AD\_PD\_CAS2317] | | | | | |
| **ASIL Level** | | ASIL D | | **Status** | New | | |
| **Syntax** | | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_\*\*\*ctl\_set\_extend\_config(  const st\_impdrv\_corectl\_t \*const p\_corectll,  const bool is\_sub\_thread  ); | | | | | |
| **Sync/Async** | | Synchronous | | | | | |
| **Reentrancy** | | Reentrant | | | | | |
| **Interrupt State** | |  | | | | | |
| **Parameters (In)** | | const st\_impdrv\_corectl\_t \*const p\_corectl | | | Control data structure for specific core. | | |
| **Range** | | Not NULL |
| const bool is\_sub\_thread | | | Thread selection. | | |
| **Range** | true or false | |
| **Parameters (In-out)** | | N/A | | | N/A | | |
| **Parameters (Out)** | | N/A | | | N/A | | |
| **Return Value** | | e\_impdrv\_errorcode\_t | | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_SEQSTATE | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | | |
| Differences between cores. | | \*\*\* | | | N/A | |
| Dma | | | no difference between the cores. | |
| dmas | | | no difference between the cores. | |
| **Preconditions** | | N/A | | | | | |
| **Remarks** | | N/A | | | | | |

#### impdrv\_dspctl\_init\_dsp

Table 2‑179: impdrv\_dspctl\_init\_dsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0061  impdrv\_dspctl\_init\_dsp  [Covers: AD\_PD\_CAS2315, AD\_PD\_CAS2316, AD\_PD\_CAS2317] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_init\_dsp(  st\_impdrv\_corectl\_t \* p\_corectl  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_get\_device\_io

Table 2‑180: impdrv\_dspctl\_get\_device\_io

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0062  impdrv\_dspctl\_get\_device\_io  [Covers: AD\_PD\_CAS2315, AD\_PD\_CAS2281, AD\_PD\_CAS2246, AD\_PD\_CAS2348, AD\_PD\_CAS2038, AD\_PD\_CAS2034] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_init\_dsp(  st\_impdrv\_corectl\_t \* p\_corectl,  st\_impdrv\_device\_handle\_t \*p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const st\_impdrv\_corectl\_t \*const p\_corectl | | Control data structure for specific core. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*  p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

#### impdrv\_dspctl\_init\_core

Table 2‑194: impdrv\_dspctl\_init\_core

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_04\_0077  impdrv\_dspctl\_init\_core  [Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2250] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_dspctl\_init\_core(  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t  \*const  p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A. | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_chk\_execute\_data

Table 2‑195: impdrv\_dspctl\_chk\_execute\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0063***  impdrv\_dspctl\_chk\_execute\_data  ***[Covers: AD\_PD\_CAS2315]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_chk\_execute\_data (  const ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  const uint32\_t core\_num,  const ***p\_impdrv\_cbfunc\_t*** callback\_func  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| const uint32\_t  core\_num | | Core number. | |
| **Range** | Not NULL |
| ***p\_impdrv\_cbfunc\_t*** callback\_func | | Callback function of execute request | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_ATTRIBUTE  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_dsp\_execute\_pre

Table 2‑196: impdrv\_dspctl\_dsp\_execute\_pre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0064***  impdrv\_dspctl\_dsp\_execute\_pre  ***[Covers: AD\_PD\_CAS2315, AD\_PD\_CAS2431, AD\_PD\_CAS2317]*** | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_dsp\_execute\_pre(  ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  const uint32\_t core\_num,  st\_impdrv\_device\_handle\_t \*p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | | |
| **Range** | | Not NULL |
| const uint32\_t  core\_num | | Core number. | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | | |
| **Range** | Not NULL | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | This function is only used in DSP core | | | | |

#### impdrv\_dspctl\_tcm\_config\_d

Table 2‑197:impdrv\_dspctl\_tcm\_config\_d

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0066***  impdrv\_dspctl\_tcm\_config\_d  ***[Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2317]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_tcm\_config\_d(  const ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_tcm\_config\_p

Table 2‑197:impdrv\_dspctl\_tcm\_config\_p

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0067***  impdrv\_dspctl\_tcm\_config\_p  ***[Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2317]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_tcm\_config\_p(  const ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_ load\_dtcm

Table 2‑197:impdrv\_dspctl\_ load\_dtcm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0068***  impdrv\_dspctl\_load\_dtcm  ***[Covers: AD\_PD\_CAS2249, AD\_PD\_CAS2317]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_load\_dtcm(  const ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_dsp\_standby

Table 2‑200: impdrv\_dspctl\_dsp\_standby

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0070***  impdrv\_dspctl\_dsp\_standby  ***[Covers: AD\_PD\_CAS2432, AD\_PD\_CAS2364]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_dsp\_standby(  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_dsp\_foece\_standby

Table 2‑201: impdrv\_dspctl\_dsp\_foece\_standby

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0071***  impdrv\_dspctl\_dsp\_foece\_standby  ***[Covers: AD\_PD\_CAS2432, AD\_PD\_CAS2364]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_dsp\_foece\_standby(  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR  IMPDRV\_EC\_NG\_DSP\_HALT | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |

#### impdrv\_dspctl\_chk\_int\_data

Table 2‑196: impdrv\_dspctl\_chk\_int\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0071***  impdrv\_dspctl\_chk\_int\_data  ***[Covers: AD\_PD\_CAS2351]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_chk\_int\_data (  ***st\_impdrv\_corectl\_t*** \* const p\_corectl,  const uint32\_t core\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const ***st\_impdrv\_corectl\_t***  \*const p\_corectl | | Control data structure for specific core. | |
| **Range** | Not NULL |
| const uint32\_t  core\_num | | Core number. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  MPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_NOTSUPPORT | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |







#### impdrv\_dspctl\_crc\_sub

Table 2‑196: impdrv\_dspctl\_crc\_sub

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ***UD\_PD\_UD02\_04\_0075***  impdrv\_dspctl\_crc\_sub  ***[Covers: AD\_PD\_CAS2603, AD\_PD\_CAS2317, AD\_PD\_CAS2600, AD\_PD\_CAS2249]*** | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  ***e\_impdrv\_errorcode\_t*** impdrv\_dspctl\_crc\_sub(  const uint32\_t st,  const uint32\_t sz,  st\_impdrv\_device\_handle\_t \*const p\_device\_io[IMPDRV\_DSP\_DEV\_NUM\_MAX]  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | uint32\_t st | | Start Address. | |
| **Range** | N/A |
| uint32\_t sz | | Size of Memory (Long Word). | |
| **Range** | N/A |
| st\_impdrv\_device\_handle\_t  \*const p\_device\_io  [IMPDRV\_DSP\_DEV\_NUM\_MAX] | | Device io handle for DSP. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | ***e\_impdrv\_errorcode\_t*** | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “***3 Activity Diagrams***”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is only used in DSP core | | | |







## Functions of OS Dependence Layer

Table 2‑176: Function List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Function Name** | **Access** | **ASIL** | **Source File Name** |
|  | impdrv\_osdep\_pow\_on\_imp\_top | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_off\_imp\_top | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_register\_irq | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_enable\_irq | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_disable\_irq | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_unregister\_irq | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_on\_imp | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_off\_imp | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pm\_set\_policy | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pm\_get\_policy | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pm\_get\_state | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_write\_reg | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_read\_reg | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_mutex\_create | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_mutex\_destroy | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_mutex\_lock | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_mutex\_unlock | Public | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_reset\_core](#_impdrv_osdep_reset_core) | Public | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_pow\_on\_hwrsc](#_impdrv_osdep_pow_on_hwrsc) | Public | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_pow\_off\_hwrsc](#_impdrv_osdep_pow_off_hwrsc) | Public | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_memory\_barrier](#_impdrv_osdep_memory_barrier) | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_queue\_create | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_queue\_destroy | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_queue\_wait\_period | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_queue\_send\_period | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_dev\_open\_imp | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_get\_size | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_get\_logical | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_get\_physical | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_cache\_flush | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_cache\_clean | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_data\_set | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_memory\_compare | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_dev\_open\_dsp | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_on\_imp\_dsp | Public | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_register\_dsp\_irq](#_impdrv_osdep_register_dsp_irq_1) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_enable\_dsp\_irq](#_impdrv_osdep_enable_dsp_irq) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_disable\_dsp\_irq](#_impdrv_osdep_disable_dsp_irq) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_unregister\_dsp\_irq](#_impdrv_osdep_unregister_dsp_irq_1) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_pow\_on\_dsp](#_impdrv_osdep_pow_on_dsp) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_pow\_off\_dsp](#_impdrv_osdep_pow_off_dsp) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_dsp\_close](#_impdrv_osdep_dsp_close) | Public | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_reset\_core\_dsp](#_impdrv_osdep_reset_core_dsp) | Public | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_dev\_open | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_on | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_pow\_off | Private | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_int\_handler](#_impdrv_osdep_int_handler) | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_chk\_core\_info | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_chk\_instance\_num | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_get\_irq\_channel | Private | ASIL D | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_dev\_open\_dsp\_sub | Private | ASIL D | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_get\_dsp\_irq\_channel](#_impdrv_osdep_get_dsp_irq_channel) | Private | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_dsp0\_int\_handler](#_impdrv_osdep_dsp0_int_handler) | Private | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_dsp1\_int\_handler](#_impdrv_osdep_dsp1_int_handler) | Private | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_dsp2\_int\_handler](#_impdrv_osdep_dsp2_int_handler) | Private | - | r\_impdrv\_osdep.c |
|  | [impdrv\_osdep\_dsp3\_int\_handler](#_impdrv_osdep_dsp3_int_handler) | Private | - | r\_impdrv\_osdep.c |
|  | impdrv\_osdep\_wait\_for\_req\_state | Private | ASIL D | r\_impdrv\_osdep.c |

### Public function

#### impdrv\_osdep\_pow\_on\_imp\_top

Table 2‑177: impdrv\_osdep\_pow\_on\_imp\_top

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0001  impdrv\_osdep\_pow\_on\_imp\_top  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1002, AD\_PD\_CAS2048, AD\_PD\_CAS2049, AD\_PD\_CAS2050, AD\_PD\_CAS1005, AD\_PD\_CAS2564, AD\_PD\_CAS2465, AD\_PD\_CAS2478] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_on\_imp\_top(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pow\_off\_imp\_top

Table 2‑178: impdrv\_osdep\_pow\_off\_imp\_top

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0002  impdrv\_osdep\_pow\_off\_imp\_top  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1004, AD\_PD\_CAS2108, AD\_PD\_CAS2109 AD\_PD\_CAS2110, AD\_PD\_CAS1006, AD\_PD\_CAS2547] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_off\_imp\_top(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_register\_irq

Table 2‑179: impdrv\_osdep\_register\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0003  impdrv\_osdep\_register\_irq  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1002, AD\_PD\_CAS2051, AD\_PD\_CAS2052] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_register\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const e\_impdrv\_instance\_t instance\_num ,  const e\_osal\_interrupt\_priority\_t irq\_priority,  const impdrv\_ctrl\_handle\_t osal\_cb\_args  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | |
| **Range** | Not NULL | |
| const e\_impdrv\_instance\_t instance\_num | | Instance number  The lifetime of this parameter is until impdrv\_osdep\_pow\_on\_imp\_top returns. | | |
| **Range** | Refer to 5.4.2.1.3. | |
| const e\_osal\_interrupt\_priority\_t irq\_priority | |  | | |
| **Range** | | Within the range of ENUM type. |
| Const impdrv\_ctrl\_handle\_t osal\_cb\_args | | OSAL callback argument  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_enable\_irq

Table 2‑180: impdrv\_osdep\_enable\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0004  impdrv\_osdep\_enable\_irq  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1002, AD\_PD\_CAS2053, AD\_PD\_CAS2054] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_enable\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const e\_impdrv\_instance\_t instance\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | |
| **Range** | Not NULL | |
| const e\_impdrv\_instance\_t instance\_num | | Num of instance  The lifetime of this parameter is until impdrv\_osdep\_pow\_on\_imp\_top returns. | | |
| **Range** | | Refer to 5.4.2.1.3. |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_disable\_irq

Table 2‑181: impdrv\_osdep\_disable\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0005  impdrv\_osdep\_disable\_irq  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1004, AD\_PD\_CAS2102, AD\_PD\_CAS2103] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_disable\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const e\_impdrv\_instance\_t instance\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| const e\_impdrv\_instance\_t instance\_num | | Num of instance | | |
| **Range** | | Refer to 5.4.2.1.3. |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_unregister\_irq

Table 2‑182: impdrv\_osdep\_unregister\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0006  impdrv\_osdep\_unregister\_irq  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1004, AD\_PD\_CAS2106, AD\_PD\_CAS2107] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_unregister\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const e\_impdrv\_instance\_t instance\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| const e\_impdrv\_instance\_t instance\_num | | Num of instance | | |
| **Range** | | Refer to 5.4.2.1.3. |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_pow\_on\_imp

Table 2‑183: impdrv\_osdep\_pow\_on\_imp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0007  impdrv\_osdep\_pow\_on\_imp  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1003, AD\_PD\_CAS2059, AD\_PD\_CAS2257, AD\_PD\_CAS2258, AD\_PD\_CAS2259, AD\_PD\_CAS2548, AD\_PD\_CAS2438, AD\_PD\_CAS2440] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_on\_imp(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_API_Layer_2)  core\_info | | Core information  The lifetime of this parameter is until impdrv\_osdep\_pow\_on\_imp\_top returns. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Change from impdrv\_osdep\_dev\_open to impdrv\_osdep\_dev\_open\_imp | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pow\_off\_imp

Table 2‑184: impdrv\_osdep\_pow\_off\_imp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0008  impdrv\_osdep\_pow\_off\_imp  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1003, AD\_PD\_CAS2062, AD\_PD\_CAS2063, AD\_PD\_CAS2064, AD\_PD\_CAS2549] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_off\_imp(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pm\_set\_policy

Table 2‑185: impdrv\_osdep\_pm\_set\_policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0009  impdrv\_osdep\_pm\_set\_policy  [Covers: AD\_PD\_CAS2439, AD\_PD\_CAS2442, AD\_PD\_CAS2443, AD\_PD\_CAS2444, AD\_PD\_CAS2445, AD\_PD\_CAS2447, AD\_PD\_CAS2448] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pm\_set\_policy(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const st\_impdrv\_core\_info\_t core\_info,  const e\_impdrv\_pm\_policy\_t pm\_policy  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t  const \*p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| const st\_impdrv\_core\_info\_t core\_info, | | Core information | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| const e\_impdrv\_pm\_policy\_t pm\_policy | | Policy you want to set | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * For the process of Open the Hardware resource of core, the function to call is changed from impdrv\_osdep\_dev\_open. (To impdrv\_osdep\_dev\_open\_dsp when the core is DSP, to impdrv\_osdep\_dev\_open\_imp otherwise) * Change the third argument of R\_OSAL\_PmSetPolicy from true to false * Remove power on/off processing before calling impdrv\_osdep\_dev\_close | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pm\_get\_policy

Table 2‑186: impdrv\_osdep\_pm\_get\_policy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0010  impdrv\_osdep\_pm\_get\_policy  [Covers: AD\_PD\_CAS1016, AD\_PD\_CAS2471, AD\_PD\_CAS2474, AD\_PD\_CAS2475, AD\_PD\_CAS2476, AD\_PD\_CAS2477] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pm\_get\_policy(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const st\_impdrv\_core\_info\_t core\_info,  e\_impdrv\_pm\_policy\_t \*const p\_pm\_policy  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| const st\_impdrv\_core\_info\_t core\_info, | | Core information | | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | e\_impdrv\_pm\_policy\_t \*const pm\_policy | | Current PM Policy | | |
| **Range** | | Not NULL. |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * For the process of Open the Hardware resource of core, the function to call is changed from impdrv\_osdep\_dev\_open. (To impdrv\_osdep\_dev\_open\_dsp when the core is DSP, to impdrv\_osdep\_dev\_open\_imp otherwise) | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_pm\_get\_state

Table 2‑187: impdrv\_osdep\_pm\_get\_state

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0044  impdrv\_osdep\_pm\_get\_state  [Covers: AD\_PD\_CAS2552, AD\_PD\_CAS2553, AD\_PD\_CAS2554, AD\_PD\_CAS2555, AD\_PD\_CAS2556] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pm\_get\_state(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const st\_impdrv\_core\_info\_t core\_info,  e\_impdrv\_pm\_state\_t \*const p\_pmstate  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_info, | | Core information | | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX | |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | e\_impdrv\_pm\_state\_t \*const p\_pmstate | | Current PM State | | |
| **Range** | | Not NULL. |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_write\_reg

Table 2‑188: impdrv\_osdep\_write\_reg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0011  impdrv\_osdep\_write\_reg  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1002, AD\_PD\_CAS1003, AD\_PD\_CAS2055, AD\_PD\_CAS2056, AD\_PD\_CAS1005, AD\_PD\_CAS1010, AD\_PD\_CAS1011, AD\_PD\_CAS2326, AD\_PD\_CAS1014] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_write\_reg(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const uint32\_t offset,  const uint32\_t val,  const bool read\_back,  const uint32\_t chk\_val  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | |
| **Range** | Not NULL | |
| const uint32\_t offset | | Offset to set to register  The lifetime of this parameter is until this function returns. | | |
| **Range** | | Less than IMPDRV\_REG\_OFFSET\_MAX and 4 byte alignment. |
| Const uint32\_t val | | Value to set to register  The lifetime of this parameter is until this function returns. | | |
| **Range** | | None. |
| Const bool read\_back | | Whether to read back. | | |
| **Range** | | true or false. |
| Const uint32\_t chk\_val | | Value to read back register. | | |
| **Range** | | None. |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_read\_reg

Table 2‑189: impdrv\_osdep\_read\_reg

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0012  impdrv\_osdep\_read\_reg  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1002, AD\_PD\_CAS1003, AD\_PD\_CAS2057, AD\_PD\_CAS2058, AD\_PD\_CAS1010, AD\_PD\_CAS1011] | | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_read\_reg(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const uint32\_t offset,  uint32\_t \*p\_val  ); | | | | | |
| **Sync/Async** | Synchronous | | | | | |
| **Reentrancy** | Non-Reentrant | | | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | | |
| **Range** | | Not NULL | |
| const uint32\_t offset | | Offset to get to register | | | |
| **Range** | | | Less　than IMPDRV\_REG\_OFFSET\_MAX and 4 byte alignment. |
| **Parameters (In-out)** | N/A | | N/A | | | |
| **Parameters (Out)** | uint32\_t \*p\_val | | Storing values from registers  The lifetime of this parameter is until this function returns. | | | |
| **Range** | Not NULL | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | | |
| **Preconditions** | N/A | | | | | |
| **Remarks** | p\_val is pointer of output parameter, so the use of this pointer variable is acceptable | | | | | |

#### impdrv\_osdep\_mutex\_create

Table 2‑190: impdrv\_osdep\_mutex\_create

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0013  impdrv\_osdep\_mutex\_create  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1001, AD\_PD\_CAS2042, AD\_PD\_CAS2043] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_mutex\_create(  st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle,  const osal\_mutex\_id\_t mutex\_id  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const osal\_mutex\_id\_t  mutex\_id | | To set value of mutex id.  The lifetime of this parameter is until impdrv\_osdep\_pow\_on\_imp\_top returns.  This ID is user-defined, so the range is not stated. | |
| **Range** | None. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle | | OSAL mutex handle | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_mutex\_destroy

Table 2‑191: impdrv\_osdep\_mutex\_destroy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0014  impdrv\_osdep\_mutex\_destroy  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2113, AD\_PD\_CAS2114] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_mutex\_destroy(  st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle | | OSAL mutex handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_mutex\_lock

Table 2‑192: impdrv\_osdep\_mutex\_lock

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0015  impdrv\_osdep\_mutex\_lock  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1007, AD\_PD\_CAS2044, AD\_PD\_CAS2045, AD\_PD\_CAS1014, AD\_PD\_CAS1015, AD\_PD\_CAS1016] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_mutex\_lock(  const st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle ,  const osal\_milli\_sec\_t time\_period  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle | | OSAL mutex handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | | |
| **Range** | Not NULL | |
| const osal\_milli\_sec\_t  time\_period | | Mutex timeout value by [msec] order  The lifetime of this parameter is until this function returns. | | |
| **Range** | | 0 or more |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_mutex\_unlock

Table 2‑193: impdrv\_osdep\_mutex\_unlock

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0016  impdrv\_osdep\_mutex\_unlock  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS1007, AD\_PD\_CAS2065, AD\_PD\_CAS2066, AD\_PD\_CAS1014, AD\_PD\_CAS1015, AD\_PD\_CAS1016] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_mutex\_unlock(  const st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_mutex\_handle\_t \*const p\_mutex\_handle | | OSAL mutex handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_reset\_core

Table 2‑194: impdrv\_osdep\_reset\_core

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0025  impdrv\_osdep\_reset\_core  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2252, AD\_PD\_CAS2332, AD\_PD\_CAS2536] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_reset\_core (  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | This function is used in V3M/V3H/V3H\_2 | | | |

#### impdrv\_osdep\_pow\_on\_hwrsc

Table 2‑195: impdrv\_osdep\_pow\_on\_hwrsc

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0026  impdrv\_osdep\_pow\_on\_hwrsc  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2129, AD\_PD\_CAS2333, AD\_PD\_CAS2334, AD\_PD\_CAS2335, AD\_PD\_CAS2550] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_on\_hwrsc(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const char \*const p\_device\_id  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const char \*const p\_device\_id | | Pointer to the Target OSAL device ID. | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_pow\_off\_hwrsc

Table 2‑196: impdrv\_osdep\_pow\_off\_hwrsc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0027  impdrv\_osdep\_pow\_off\_hwrsc  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2148, AD\_PD\_CAS2330, AD\_PD\_CAS2336, AD\_PD\_CAS2337, AD\_PD\_CAS2551] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_off\_hwrsc(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** |  |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_SYSTEMERROR  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_memory\_barrier

Table 2‑197: impdrv\_osdep\_memory\_barrier

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0028  impdrv\_osdep\_memory\_barrier  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2329] | | |
| **ASIL Level** | ASIL D | **Status** | New |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_barrier(  void  ); | | |
| **Sync/Async** | Synchronous | | |
| **Reentrancy** | Non-Reentrant | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | |
| **Parameters (In)** | N/A | | N/A |
| **Parameters (In-out)** | N/A | | N/A |
| **Parameters (Out)** | N/A | | N/A |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SYSTEMERROR |
| **Description** | Refer to “3 Activity Diagrams”. | | |
| **Preconditions** | N/A | | |
| **Remarks** | N/A | | |

#### impdrv\_osdep\_queue\_create

Table 2‑198: impdrv\_osdep\_queue\_create

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0029  impdrv\_osdep\_queue\_create  [Covers: AD\_PD\_CAS2493, AD\_PD\_CAS2502, AD\_PD\_CAS2503] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_queue\_create(  st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle,  const osal\_mq\_id\_t queue\_id  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const osal\_mq\_id\_t queue\_id | | OSAL message queue ID | | |
| **Range** | More than 0 | |
| **Parameters (In-out)** | st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle | | OSAL message queue handle | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_queue\_destroy

Table 2‑199: impdrv\_osdep\_queue\_destroy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0030  impdrv\_osdep\_queue\_destroy  [Covers: AD\_PD\_CAS2496, AD\_PD\_CAS2504, AD\_PD\_CAS2505, AD\_PD\_CAS2506] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_register\_irq(  st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle | | OSAL message queue handle | |
| **Range** | Not Null |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_queue\_wait\_period

Table 2‑200: impdrv\_osdep\_queue\_wait\_period

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0031  impdrv\_osdep\_queue\_wait\_period  [Covers: AD\_PD\_CAS2495, AD\_PD\_CAS2507, AD\_PD\_CAS2508] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_queue\_wait\_period (  const st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle,  const osal\_milli\_sec\_t time\_period,  uint32\_t \*const p\_receive\_buffer  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle | | OSAL message queue handle | | |
| **Range** | Not Null | |
| const osal\_milli\_sec\_t time\_period | | Timeout period of millisecond order | | |
| **Range** | 0 or more than | |
| **Parameters (In-out)** | uint32\_t \*const p\_receive\_buffer | | Message receive buffer pointer | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_CHECKFAIL  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_queue\_send\_period

Table 2‑201: impdrv\_osdep\_queue\_send\_period

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0032  impdrv\_osdep\_queue\_send\_period  [Covers: AD\_PD\_CAS2509, AD\_PD\_CAS2510] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_queue\_send\_period (  const st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle,  const osal\_milli\_sec\_t time\_period,  const uint32\_t \*const p\_send\_buffer  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_queue\_handle\_t \*const p\_queue\_handle | | OSAL message queue handle | |
| **Range** | Not Null |
| const osal\_milli\_sec\_t time\_period | | Timeout period of millisecond order | |
| **Range** | 0 or more |
| const uint32\_t \*const p\_send\_buffer | | Message send buffer pointer | |
| **Range** | Not Null |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_dev\_open\_imp

Table 2‑202: impdrv\_osdep\_dev\_open\_imp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0046  impdrv\_osdep\_dev\_open\_imp  [Covers: AD\_PD\_CAS2565, AD\_PD\_CAS2566, AD\_PD\_CAS2567] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_dev\_open\_imp (  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info | | Core information | | |
| **Range** | | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not Null | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_dev\_close

Table 2‑203: impdrv\_osdep\_dev\_close

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0018  impdrv\_osdep\_dev\_close  [Covers: AD\_PD\_CAS3005, AD\_PD\_CAS2446] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_dev\_close(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | N/A | | N/A | |
| **Parameters (In-out)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_memory\_get\_size

Table 2‑204: impdrv\_osdep\_memory\_get\_size

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0033  impdrv\_osdep\_memory\_get\_size  [Covers: AD\_PD\_CAS2489, AD\_PD\_CAS2511, AD\_PD\_CAS2512] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_get\_size (  const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle,  size\_t \*const p\_size  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle | | OSAL memory buffer handle | | |
| **Range** | Not Null | |
| **Parameters (In-out)** | size\_t \*const p\_size | | buffer size pointer | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_memory\_get\_logical

Table 2‑205: impdrv\_osdep\_memory\_get\_logical

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0034  impdrv\_osdep\_memory\_get\_logical  [Covers: AD\_PD\_CAS2490, AD\_PD\_CAS2515, AD\_PD\_CAS2514] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_get\_logical (  const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle,  void \*\*const p\_logical\_adrs  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle | | OSAL memory buffer handle | | |
| **Range** | Not Null | |
| **Parameters (In-out)** | void \*\*const p\_logical\_adrs | | logical address pointer | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_memory\_get\_physical

Table 2‑206: impdrv\_osdep\_memory\_get\_physical

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0035  impdrv\_osdep\_memory\_get\_physical  [Covers: AD\_PD\_CAS2491, AD\_PD\_CAS2516, AD\_PD\_CAS2517] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_get\_physical (  const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle,  const osal\_axi\_bus\_id\_t axi\_id,  uintptr\_t \*const p\_physical\_adrs  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle | | OSAL memory buffer handle | | |
| **Range** | Not Null | |
| const osal\_axi\_bus\_id\_t axi\_id | | OSAL AXI bus ID | | |
| **Range** | Not OSAL\_AXI\_BUS\_ID\_INVALID | |
| **Parameters (In-out)** | uintptr\_t \*const p\_physical\_adrs | | physical address pointer | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_memory\_cache\_flush

Table 2‑207: impdrv\_osdep\_memory\_cache\_flush

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0036  impdrv\_osdep\_memory\_cache\_flush  [Covers: AD\_PD\_CAS2492, AD\_PD\_CAS2531, AD\_PD\_CAS2532] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_cache\_flush (  const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle,  const uintptr\_t offset,  const size\_t size  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle | | OSAL memory buffer handle | |
| **Range** | Not Null |
| const uintptr\_t offset | | Offset address | |
| **Range** | Not IMPDRV\_CHK\_CACHE\_ALIGN |
| const size\_t size | | Cache flush size | |
| **Range** | N/A |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_memory\_cache\_clean

Table 2‑208: impdrv\_osdep\_memory\_cache\_clean

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0037  impdrv\_osdep\_memory\_cache\_clean  [Covers: AD\_PD\_CAS2497, AD\_PD\_CAS2518, AD\_PD\_CAS2519] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_cache\_clean (  const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle,  const uintptr\_t offset,  const size\_t size  ); ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_memory\_handle\_t \*const p\_memory\_handle | | OSAL memory buffer handle | |
| **Range** | Not Null |
| const uintptr\_t offset | | Offset address | |
| **Range** | Not IMPDRV\_CHK\_CACHE\_ALIGN |
| const size\_t size | | Cache clean size | |
| **Range** | N/A |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_memory\_data\_set

Table 2‑209: impdrv\_osdep\_memory\_data\_set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0038  impdrv\_osdep\_memory\_data\_set  [Covers: AD\_PD\_CAS2528, AD\_PD\_CAS2529, AD\_PD\_CAS2530] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_data\_set (  uint32\_t \*const p\_buffer,  const uint32\_t data,  const size\_t size  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const uint32\_t data, | | Data to write | | |
| **Range** | N/A | |
| const size\_t size | | Data size | | |
| **Range** | Not IMPDRV\_CHK\_MEM\_ALIGN | |
| **Parameters (In-out)** | uint32\_t \*const p\_buffer | | Memory buffer address | | |
| **Range** | | Not Null |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_memory\_compare

Table 2‑210: impdrv\_osdep\_memory\_compare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0039  impdrv\_osdep\_memory\_compare  [Covers: AD\_PD\_CAS2498, AD\_PD\_CAS2520, AD\_PD\_CAS2521] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_memory\_compare (  const uint32\_t \*const p\_expected\_area,  const uint32\_t \*const p\_result\_area,  const size\_t size  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const uint32\_t \* const p\_expected\_area | | Data area for test expectations | |
| **Range** | Not Null |
| const uint32\_t \* const p\_result\_area, | | Data area for test result | |
| **Range** | Not Null |
| const size\_t size | | Data size | |
| **Range** | Not IMPDRV\_CHK\_MEM\_ALIGN |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_CHECKFAIL | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_dev\_open\_dsp

Table 2‑211: impdrv\_osdep\_dev\_open\_dsp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0045  impdrv\_osdep\_dev\_open\_dsp  [Covers: AD\_PD\_CAS2462, AD\_PD\_CAS2463, AD\_PD\_CAS2464] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_dev\_open\_dsp (  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info,  const uint32\_t sub\_dev\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info | | Core information | | |
| **Range** | | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| const uint32\_t sub\_dev\_num | | Sub device number | | |
| **Range** | | Less than num of sub device |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not Null | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_pow\_on\_imp\_dsp

Table 2‑212: impdrv\_osdep\_pow\_on\_imp\_dsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0051  impdrv\_osdep\_pow\_on\_imp\_dsp  [Covers: AD\_PD\_CAS2479, AD\_PD\_CAS2488, AD\_PD\_CAS2543, AD\_PD\_CAS2568, AD\_PD\_CAS2569, AD\_PD\_CAS2570, AD\_PD\_CAS2571] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_on\_imp\_dsp (  st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info,  const uint32\_t sub\_dev\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const [st\_impdrv\_core\_info\_t](#_API_Layer_2) core\_info | | Core information  The lifetime of this parameter is until impdrv\_osdep\_pow\_on\_imp\_top returns. | |
| **Range** | Not Null |
| const uint32\_t sub\_dev\_num | | Sub device number | |
| **Range** | Less than num of sub device |
| **Parameters (In-out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Change from Private function impdrv\_osdep\_dev\_open\_dsp\_sub (old name : impdrv\_osdep\_dev\_open\_dsp) to Public function impdrv\_osdep\_dev\_open\_dsp | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_register\_dsp\_irq

Table 2‑226: impdrv\_osdep\_register\_dsp\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0040  impdrv\_osdep\_register\_dsp\_irq  [Covers: AD\_PD\_CAS2458, AD\_PD\_CAS2459] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_register\_dsp\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  unit32\_t const core\_num ,  const e\_osal\_interrupt\_priority\_t irq\_priority,  const impdrv\_ctrl\_handle\_t osal\_cb\_args  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | Unit32\_t  const core\_num | | DSP core number | | |
| **Range** | Less than num of DSP device | |
| const e\_osal\_interrupt\_priority\_t irq\_priority | | Interrupt priority. | | |
| **Range** | | Within the range of ENUM type. |
| Const impdrv\_ctrl\_handle\_t osal\_cb\_args | | OSAL callback argument | | |
| **Range** | | Not NULL |
| **Parameters (In-out)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | | Not NULL |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_enable\_dsp\_irq

Not use. (Out of scope)

#### impdrv\_osdep\_disable\_dsp\_irq

Table 2‑226: impdrv\_osdep\_disable\_dsp\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0056  impdrv\_osdep\_disable\_dsp\_irq  [Covers: AD\_PD\_CAS2460, AD\_PD\_CAS2461] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_disable\_dsp\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IIMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_unregister\_dsp\_irq

Table 2‑228: impdrv\_osdep\_unregister\_dsp\_irq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0055  impdrv\_osdep\_unregister\_dsp\_irq  [Covers: AD\_PD\_CAS2572, AD\_PD\_CAS2573] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_unregister\_dsp\_irq(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const unit32\_t core\_num  ); | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | | |
| **Range** | Not NULL | |
| const unit32\_t core\_num | | DSP core number | | |
| **Range** | | Less than num of DSP device |
| **Parameters (In-out)** | N/A | | N/A | | |
| **Range** | | N/A |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM  IMPDRV\_EC\_NG\_SYSTEMERROR | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A. | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_pow\_on\_dsp

Not use. (Out of scope)

#### impdrv\_osdep\_pow\_off\_dsp

Not use. (Out of scope)

#### impdrv\_osdep\_dsp\_close

Not use. (Out of scope)

#### impdrv\_osdep\_reset\_core\_dsp

Table 2‑228: impdrv\_osdep\_reset\_core\_dsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0054  impdrv\_osdep\_reset\_core\_dsp  [Covers: AD\_PD\_CAS2574, AD\_PD\_CAS2575, AD\_PD\_CAS2576, AD\_PD\_CAS2577] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | e\_impdrv\_errorcode\_t impdrv\_osdep\_reset\_core\_dsp (  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle,  const bool reset  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| const bool reset | | Rest State | |
| **Range** | true or false |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

### Private function

#### impdrv\_osdep\_dev\_open

Table 2‑213: impdrv\_osdep\_dev\_open

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0017  impdrv\_osdep\_dev\_open  [Covers: AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_dev\_open(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const char \*const type,  const size\_t dev\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const char \*const type | | Specific device type  The lifetime of this parameter is until this function returns. | |
| **Range** | Not NULL |
| const size\_t dev\_num | | Specific device number  The lifetime of this parameter is until this function returns. | |
| **Range** | None. |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pow\_on

Table 2‑214: impdrv\_osdep\_pow\_on

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0019  impdrv\_osdep\_pow\_on  [Covers: AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_on(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \* const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Add call R\_OSAL\_PmGetRequiredState before call R\_OSAL\_PmSetRequiredState * Add a condition for call R\_OSAL\_PmSetRequiredState: State obtained in R\_OSAL\_PmGetRequiredState is not REQUIRED | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_pow\_off

Table 2‑215: impdrv\_osdep\_pow\_off

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0020  impdrv\_osdep\_pow\_off  [Covers: AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_pow\_off(  const st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_device\_handle\_t \* const p\_device\_handle | | OSAL device handle  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”.  [Update Information]   * Add call R\_OSAL\_PmGetRequiredState before call R\_OSAL\_PmSetRequiredState * Add a condition for call R\_OSAL\_PmSetRequiredState: State obtained in R\_OSAL\_PmGetRequiredState is not RELEASE | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_int\_handler

Table 2‑216: impdrv\_osdep\_int\_handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0021  impdrv\_osdep\_int\_handler  [Covers: AD\_PD\_CAS1011, AD\_PD\_CAS2331, AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  void impdrv\_osdep\_int\_handler(  osal\_device\_handle\_t device\_handle ,  const uint64\_t irq\_channel,  void\* irq\_arg  ) | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | osal\_device\_handle\_t device\_handle | | Device handle associated with OSAL interrupt manager.  The lifetime of this parameter is the period from the impdrv\_osdep\_pow\_on\_imp\_top is executed until impdrv\_osdep\_pow\_off\_imp\_top is executed. | |
| **Range** | Not NULL |
| const uint64\_t irq\_channel, | | IRQ channel associated with OSAL interrupt manager.  The lifetime of this parameter is until this function returns. | |
| **Range** | None. |
| Void\* irq\_arg | | User argument (control handle) specified when registered interrupt.  The lifetime of this parameter is until this function returns. | |
| **Range** | Not NULL |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | N/A | | N/A | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_chk\_core\_info

Table 2‑217: impdrv\_osdep\_chk\_core\_info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0022  impdrv\_osdep\_chk\_core\_info  [Covers: AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_chk\_core\_info(  const st\_impdrv\_core\_info\_t core\_info  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t core\_info | | Core information | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_chk\_instance\_num

Table 2‑218: impdrv\_osdep\_chk\_instance\_num

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0024  impdrv\_osdep\_chk\_instance\_num  [Covers: AD\_PD\_CAS3005] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_chk\_instance\_num(  const e\_impdrv\_instance\_t instance\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | N/A | | | |

#### impdrv\_osdep\_get\_irq\_channel

Table 2‑219: impdrv\_osdep\_get\_irq\_channel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0023  impdrv\_osdep\_get\_irq\_channel  [Covers: AD\_PD\_CAS2053, AD\_PD\_CAS2084, AD\_PD\_CAS2086] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_get\_irq\_channel(  const e\_impdrv\_instance\_t instance\_num,  uint64\_t \*const irq\_channel  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (In)** | const e\_impdrv\_instance\_t instance\_num | | Instance number | |
| **Range** | Within the range of ENUM type. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | uint64\_t \*const irq\_channel | | Interrupt channel  The validity period of this parameter is the period from when this function is called to when it ends. | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PARAM | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | irq\_channel is pointer of output parameter, so the use of this pointer variable is acceptable | | | |

#### impdrv\_osdep\_dev\_open\_dsp\_sub

Table 2‑220: impdrv\_osdep\_dev\_open\_dsp\_sub

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0052  impdrv\_osdep\_dev\_open\_dsp\_sub  [Covers: AD\_PD\_CAS2463] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_dev\_open\_dsp\_sub(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle ,  const char \*const type,  const size\_t dev\_num,  const uint32\_t sub\_dev\_num  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can not be called from OSAL Callback. | | | |
| **Parameters (in)** | const char \*const type | | Device type | |
| **Range** | Not NULL |
| const size\_t dev\_num | | Device number | |
| **Range** | Within the range of ENUM type.  Value less than IMPDRV\_CORE\_NUM\_MAX |
| const uint32\_t sub\_dev\_num | | Sub device number | |
| **Range** | Less than num of sub device |
| **Parameters (In-Out)** | st\_impdrv\_device\_handle\_t \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_SYSTEMERROR | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”.  [Update Information]   * Change Function Name from impdrv\_osdep\_dev\_open\_dsp | | | |
| **Preconditions** | None. | | | |
| **Remarks** | None. | | | |

#### impdrv\_osdep\_get\_dsp\_irq\_channel

Table 2‑235: impdrv\_osdep\_get\_dsp\_irq\_channel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0053  impdrv\_osdep\_get\_dsp\_irq\_channel  [Covers: AD\_PD\_CAS2449, AD\_PD\_CAS2450, AD\_PD\_CAS2458, AD\_PD\_CAS2459, AD\_PD\_CAS2572, AD\_PD\_CAS2573, AD\_PD\_CAS2460, AD\_PD\_CAS2461] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_get\_dsp\_irq\_channel(  unit64\_t \*const irq\_channel  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (in)** | N/A | | N/A | |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | unit64\_t \*const irq\_channel | | Interrupt channel | |
| **Range** | Not NULL |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_ARGNULL | |
| **Description** | Details of internal processing, refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | None. | | | |
| **Remarks** | None. | | | |

#### impdrv\_osdep\_dsp0\_int\_handler

Table 2‑233: impdrv\_osdep\_dsp0\_int\_handler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0050  impdrv\_osdep\_dsp0\_int\_handler  [Covers: AD\_PD\_CAS2578] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  void impdrv\_osdep\_dsp0\_int\_handler (  osal\_device\_handle\_t device\_handle ,  const uint64\_t irq\_channel,  void\* irq\_arg  ) | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | osal\_device\_handle\_t device\_handle | | Device handle associated with OSAL interrupt manager. | | |
| **Range** | | Not NULL |
| const uint64\_t irq\_channel, | | IRQ channel associated with OSAL interrupt manager. | | |
| **Range** | | None. |
| **Parameters (In-Out)** | Void\* irq\_arg | | User argument (control handle) specified when registered interrupt. | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | N/A | | N/A | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_dsp1\_int\_handler

Table 2‑234: impdrv\_osdep\_dsp1\_int\_handler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0049  impdrv\_osdep\_dsp1\_int\_handler  [Covers: AD\_PD\_CAS2578] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  void impdrv\_osdep\_dsp1\_int\_handler (  osal\_device\_handle\_t device\_handle ,  const uint64\_t irq\_channel,  void\* irq\_arg  ) | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | osal\_device\_handle\_t device\_handle | | Device handle associated with OSAL interrupt manager. | | |
| **Range** | | Not NULL |
| const uint64\_t irq\_channel, | | IRQ channel associated with OSAL interrupt manager.. | | |
| **Range** | | None. |
| **Parameters (In-Out)** | Void\* irq\_arg | | User argument (control handle) specified when registered interrupt. | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | N/A | | N/A | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_dsp2\_int\_handler

Table 2‑235: impdrv\_osdep\_dsp2\_int\_handler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0048  impdrv\_osdep\_dsp2\_int\_handler  [Covers: AD\_PD\_CAS2578] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  void impdrv\_osdep\_dsp2\_int\_handler (  osal\_device\_handle\_t device\_handle ,  const uint64\_t irq\_channel,  void\* irq\_arg  ) | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | osal\_device\_handle\_t device\_handle | | Device handle associated with OSAL interrupt manager. | | |
| **Range** | | Not NULL |
| const uint64\_t irq\_channel, | | IRQ channel associated with OSAL interrupt manager. | | |
| **Range** | | None. |
| **Parameters (In-Out)** | Void\* irq\_arg | | User argument (control handle) specified when registered interrupt. | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | N/A | | N/A | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_dsp3\_int\_handler

Table 2‑236: impdrv\_osdep\_dsp3\_int\_handler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0041  impdrv\_osdep\_dsp3\_int\_handler  [Covers: AD\_PD\_CAS2578] | | | | |
| **ASIL Level** | ASIL D | **Status** | New | | |
| **Syntax** | IMPDRV\_STATIC  void impdrv\_osdep\_dsp3\_int\_handler (  osal\_device\_handle\_t device\_handle ,  const uint64\_t irq\_channel,  void\* irq\_arg  ) | | | | |
| **Sync/Async** | Synchronous | | | | |
| **Reentrancy** | Non-Reentrant | | | | |
| **Interrupt State** |  | | | | |
| **Parameters (In)** | osal\_device\_handle\_t device\_handle | | Device handle associated with OSAL interrupt manager. | | |
| **Range** | | Not NULL |
| const uint64\_t irq\_channel, | | IRQ channel associated with OSAL interrupt manager. | | |
| **Range** | | None. |
| **Parameters (In-Out)** | Void\* irq\_arg | | User argument (control handle) specified when registered interrupt. | | |
| **Range** | Not NULL | |
| **Parameters (Out)** | N/A | | N/A | | |
| **Return Value** | N/A | | N/A | | |
| **Description** | Refer to “3 Activity Diagrams”. | | | | |
| **Preconditions** | N/A | | | | |
| **Remarks** | N/A | | | | |

#### impdrv\_osdep\_wait\_for\_req\_state

Table 2‑243: impdrv\_osdep\_wait\_for\_req\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | UD\_PD\_UD02\_05\_0047  impdrv\_osdep\_wait\_for\_req\_state  [Covers: AD\_PD\_CAS2108, AD\_PD\_CAS2258, AD\_PD\_CAS2334, AD\_PD\_CAS2441, AD\_PD\_CAS2444, AD\_PD\_CAS2568, AD\_PD\_CAS2569, AD\_PD\_CAS2570] | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | IMPDRV\_STATIC  e\_impdrv\_errorcode\_t impdrv\_osdep\_wait\_for\_req\_state(  st\_impdrv\_device\_handle\_t \*const p\_device\_handle  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Reentrant | | | |
| **Interrupt State** |  | | | |
| **Parameters (In)** | st\_impdrv\_device\_handle\_t  \*const p\_device\_handle | | OSAL device handle | |
| **Range** | Not NULL |
| **Parameters (In-out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | e\_impdrv\_errorcode\_t | | IMPDRV\_EC\_OK  IMPDRV\_EC\_NG\_SEQSTATE  IMPDRV\_EC\_NG\_ARGNULL  IMPDRV\_EC\_NG\_PMSYSTEMERROR | |
| **Description** | Refer to “3 Activity Diagrams”. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

# Activity Diagrams

The activity diagrams are described in this eapx file.



# Resource Usage

Refer to section “General Description” and “Usage Notes” of R-Car V3x IMP Driver Product Information .

# Data Design

## Data Type Definitions

### Typedef

#### API Layer

##### impdrv\_ctrl\_handle\_t

Table 5‑1: impdrv\_ctrl\_handle\_t

|  |  |
| --- | --- |
| **Format** | struct st\_impdrv\_ctl\_t;  typedef struct st\_impdrv\_ctl\_t\* impdrv\_ctrl\_handle\_t; |
| **Description** | Control handle definition for IMP driver.  Declare it as a pointer to the Abstract control structure. |
| **Remarks** | N/A |

##### p\_impdrv\_cbfunc\_t

Table 5‑2: p\_impdrv\_cbfunc\_t

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | p\_impdrv\_cbfunc\_t | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | typedef int32\_t (\*p\_impdrv\_cbfunc\_t)(  const st\_impdrv\_core\_info\_t \*const p\_core\_info,  const e\_impdrv\_cb\_ret\_t ercd,  const int32\_t code,  void \*const p\_callback\_args  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*p\_core\_info | | Core information | |
| **Range** | Must not be NULL. |
| Const e\_impdrv\_cb\_ret\_t ercd | | Return code of callback | |
| **Range** | Within the range of ENUM type. |
| Const int32\_t code | | TRAP or INT code (8 bit is set) | |
| **Range** | None. (Use the value read from hardware register) |
| void  \*const p\_callback\_args | | User argument | |
| **Range** | None. (Depends on the user implementation) |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | int32\_t | | 0 or non-zero | |
| **Description** | For the usage, refer to the “5.3.1 p\_impdrv\_cbfunc\_t” in [3]. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

##### p\_impdrv\_cbfunc\_fatal\_t

Table 5‑3: p\_impdrv\_cbfunc\_fatal\_t

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | p\_impdrv\_cbfunc\_fatal\_t | | | |
| **ASIL Level** | ASIL D | **Status** | New | |
| **Syntax** | typedef void (\*p\_impdrv\_cbfunc\_fatal\_t)(  const st\_impdrv\_core\_info\_t \*const p\_core\_info  const \_impdrv\_fatalcode\_t fc\_code,  const e\_impdrv\_errorcode\_t ercd,  void \*const p\_callback\_args  ); | | | |
| **Sync/Async** | Synchronous | | | |
| **Reentrancy** | Non-Reentrant | | | |
| **Interrupt State** | Can be called from OSAL Callback. | | | |
| **Parameters (In)** | const st\_impdrv\_core\_info\_t \*const p\_core\_info | | Core information | |
| **Range** | Not NULL. |
| Const e\_impdrv\_fatalcode\_t fc\_code | | Fatal callback result code | |
| **Range** | IMPDRV fatal error code.  Within the range of ENUM type. |
| Const e\_impdrv\_errorcode\_t ercd | | IMPDRV Return code | |
| **Range** | IMP Driver error code.  Within the range of ENUM type. |
| Void  \*const p\_callback\_args | | Callback function arguments | |
| **Range** | None. |
| **Parameters (In-Out)** | N/A | | N/A | |
| **Parameters (Out)** | N/A | | N/A | |
| **Return Value** | N/A | | N/A | |
| **Description** | For the usage, refer to the “ 4.4.6 st\_impdrv\_fatalcode\_t” in [3]. | | | |
| **Preconditions** | N/A | | | |
| **Remarks** | N/A | | | |

### Structure

#### API Layer

##### st\_impdrv\_initdata\_t

Table 5‑4: st\_impdrv\_initdata\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  void \*p\_work\_addr;  uint32\_t work\_size;  uint32\_t out\_work\_size;  e\_impdrv\_instance\_t instance\_num;  uint32\_t use\_core\_num;  st\_impdrv\_core\_info\_t core\_info[IMPDRV\_INNER\_FIXED\_VALUE];  p\_impdrv\_cbfunc\_fatal\_t callback\_func\_fatal;  void \*p\_cb\_args\_fatal;  st\_impdrv\_resource\_t osal\_resource;  } st\_impdrv\_initdata\_t; | | |
| **Description** | Data structure used for the R\_IMPDRV\_Init function. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| p\_work\_addr | In | Work area address  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. |
| Work\_size | In | Work area size |
| out\_work\_size | Out | Used memory size |
| instance\_num | In | Instance number |
| use\_core\_num | In | Number of core lists |
| core\_info | In | Core lists information |
| callback\_func\_fatal | In | Fatal callback function  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. |
| P\_cb\_args\_fatal | In | fatal callback args  The lifetime of this parameter is the period from the R\_IMPDRV\_Init is executed until R\_IMPDRV\_Quit is executed. |
| Osal\_resource | In | OSAL resources information |
| **Member**  **Valid value** | **Name** | **Value** | |
| p\_work\_addr | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Work\_size | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Out\_work\_size | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Instance\_num | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Use\_core\_num | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Core\_info | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Callback\_func\_fatal | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| P\_cb\_args\_fatal | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| Osal\_resource | Refer to the “4.4.1 st\_impdrv\_initdata\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_resource\_t

Table 5‑5: st\_impdrv\_resource\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_mutex\_id\_t mutex\_id;  osal\_milli\_sec\_t mutex\_time\_period;  osal\_mq\_id\_t mq\_id;  osal\_milli\_sec\_t mq\_time\_period;  e\_osal\_interrupt\_priority\_t irq\_priority;  } st\_impdrv\_resource\_t; | | |
| **Description** | Data structure used for the st\_impdrv\_initdata\_t internal structure. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| mutex\_id | In | Mutex ID used by IMP Driver |
| mutex\_time\_period | In | Millisecond order for the Timeout value of mutex |
| mq\_id | In | Message queue ID used by IMP Driver |
| mq\_time\_period | In | Millisecond order for the Timeout value of message queue |
| irq\_priority | In | Interrupt priority |
| **Member**  **Valid value** | **Name** | **Value** | |
| mutex\_id | Refer to the “ 4.4.4 st\_impdrv\_resource\_t” in [3]. | |
| Mutex\_time\_period | Refer to the “ 4.4.4 st\_impdrv\_resource\_t” in [3]. | |
| Mq\_id | Refer to the “ 4.4.4 st\_impdrv\_resource\_t” in [3]. | |
| Mq\_time\_period | Refer to the “ 4.4.4 st\_impdrv\_resource\_t” in [3]. | |
| Irq\_priority | Refer to the “ 4.4.4 st\_impdrv\_resource\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_core\_info\_t

Table 5‑6: st\_impdrv\_core\_info\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  e\_impdrv\_core\_type\_t core\_type;  uint32\_t core\_num;  } st\_impdrv\_core\_info\_t; | | |
| **Description** | IMP Driver determines the IMP core, selected by this value. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| core\_type | In | IMP Core type |
| core\_num | In | Core number in the core type |
| **Member**  **Valid value** | **Name** | **Value** | |
| core\_type | Refer to the “4.4.2 st\_impdrv\_core\_info\_t” in [3]. | |
| Core\_num | Refer to the “4.4.2 st\_impdrv\_core\_info\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_irq\_group\_t

Table 5‑7: st\_impdrv\_irq\_group\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  e\_impdrv\_irq\_group\_t irq\_group;  uint32\_t group\_core\_num;  st\_impdrv\_core\_info\_t group\_core\_info[IMPDRV\_IRQGROUP\_MAXID];  } st\_impdrv\_irq\_group\_t; | | |
| **Description** | IMP Driver determines the IMP core, selected by this value. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| irq\_group | In | Interrupt group number |
| group\_core\_num | In | Number of Group cores |
| group\_core\_info | In | Group core list |
| **Member**  **Valid value** | **Name** | **Value** | |
| irq\_group | Refer to the “ 4.4.7 st\_impdrv\_irq\_group\_t” in [3]. | |
| Group\_core\_num | Refer to the “ 4.4.7 st\_impdrv\_irq\_group\_t” in [3]. | |
| Group\_core\_info | Refer to the “ 4.4.7 st\_impdrv\_irq\_group\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_chk\_param\_t

Table 5‑8: st\_impdrv\_chk\_param\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uintptr\_t offset;  uint32\_t data;  uint32\_t bit\_mask;  } st\_impdrv\_chk\_param\_t; | | |
| **Description** | OSAL resources in used by IMPDRV for bus interface check. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| offset | In | Offset address of hardware register |
| data | In | Expected check data |
| bit\_mask | In | Bitmask value for check data |
| **Member**  **Valid value** | **Name** | **Value** | |
| offset | Offset address of hardware register. | |
| Data | Expected data | |
| bit\_mask | Bitmask value | |
| **Remarks** | N/A | | |

##### st\_impdrv\_version\_t

Table 5‑9: st\_impdrv\_version\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t major;  uint32\_t minor;  uint32\_t patch;  } st\_impdrv\_version\_t; | | |
| **Description** | This function return driver version.  The version number returned is based on the IMPDRV\_VERSION\_MAJOR, IMPDRV\_VERSION\_MINOR, and IMPDRV\_VERSION\_PATCH definition. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| major | Out | Major version of IMP driver |
| minor | Out | Minor version of IMP driver |
| patch | Out | Patch version of IMP driver |
| **Member**  **Valid value** | **Name** | **Value** | |
| major | Refer to the “4.4.5 st\_impdrv\_version\_t” in [3]. | |
| Minor | Refer to the “4.4.5 st\_impdrv\_version\_t” in [3]. | |
| Patch | Refer to the “4.4.5 st\_impdrv\_version\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_reg\_info\_t

Table 5‑10: st\_impdrv\_reg\_info\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uintptr\_t addr\_phys;  uint32\_t size;  } st\_impdrv\_reg\_info\_t; | | |
| **Description** | IMP Driver Hardware register area specification information. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| addr\_phys | Out | Top address of hardware register |
| size | Out | Area size of hardware register |
| **Member**  **Valid value** | **Name** | **Value** | |
| addr\_phys | Refer to the “ 4.4.7 st\_impdrv\_irq\_group\_t” in [3]. | |
| Size | Refer to the “ 4.4.7 st\_impdrv\_irq\_group\_t” in [3]. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_chk\_resource\_t

Table 5‑11: st\_impdrv\_chk\_resource\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_memory\_buffer\_handle\_t memory\_handle;  osal\_axi\_bus\_id\_t axi\_id;  osal\_mq\_id\_t mq\_id;  osal\_milli\_sec\_t mq\_time\_period;  } st\_impdrv\_chk\_resource\_t; | | |
| **Description** | OSAL resources in used by IMPDRV for bus interface check. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| memory\_handle | In | OSAL Memory manager handle |
| axi\_id | In | AXI bus ID |
| mq\_id | In | Queue ID used by IMP Driver |
| mq\_time\_period | In | Millisecond Order Timeout value |
| **Member**  **Valid value** | **Name** | **Value** | |
| memory\_handle | Handle type of a Buffer Object. | |
| Axi\_id | AXI Bus ID type that is assigned to a device. | |
| Mq\_id | uniquely identifies a message queue | |
| mq\_time\_period | time in milliseconds. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_dsp\_data\_t

Not use (Out of scope for xOS2.0)

Table 5‑12: st\_impdrv\_dsp\_data\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uintptr\_t addr\_phys;  uint32\_t size;  } st\_impdrv\_dsp\_data\_t; | | |
| **Description** | - | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| addr\_phys | - | - |
| size | - | - |
| **Member**  **Valid value** | **Name** | **Value** | |
| addr\_phys | - | |
| size | - | |
| **Remarks** | N/A | | |

#### General Control

##### st\_impdrv\_ctl\_t

Table 5‑13: st\_impdrv\_ctl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct st\_impdrv\_ctl  {  st\_impdrv\_coretypectl\_t coretype\_ctl[IMPDRV\_CORE\_TYPE\_MAX];  st\_impdrv\_commonctl\_t common\_ctl;  } st\_impdrv\_ctl\_t; | | |
| **Description** | Common control data structure used by general control subcomponents. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| coretype\_ctl | In/out | Core control structure data for each core type. |
| Common\_ctl | In/out | Common control structure data to all core. |
| **Member**  **Valid value** | **Name** | **Value** | |
| coretype\_ctl | Follows the st\_impdrv\_coretypectl\_t structure. | |
| Common\_ctl | Follows the st\_impdrv\_commonctl\_t structure. | |
| **Remarks** | N/A | | |

#### Common Control

##### st\_impdrv\_commonctl\_t

Table 5‑14: st\_impdrv\_commonctl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  e\_impdrv\_instance\_t instance\_num;  uint32\_t use\_core\_ins ;  p\_impdrv\_cbfunc\_fatal\_t cbfunc\_fatal;  void \*p\_cb\_args\_fatal;  bool is\_mutex\_locked;  uint32\_t irq\_mask\_value;  [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t) protect\_mode;  void \*p\_hwrsc\_def\_table;  void \*p\_hwrsc\_mng\_table;  } st\_impdrv\_commonctl\_t; | | |
| **Description** | Common resource data structure used by common control subcomponents. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| instance\_num | In/Out | Instance number. |
| Use\_core\_ins | In/Out | The current instance’s use core |
| cbfunc\_fatal | In/Out | Fatal callback function |
| p\_cb\_args\_fatal | In/Out | Fatal callback argument |
| is\_mutex\_locked | In/Out | Mutex locked state for Register access API |
| irq\_mask\_value | In/Out | Save data of IRQ mask register |
| protect\_mode | In/Out | Save data of memory protection mode |
| \*p\_hwrsc\_def\_table | In/Out | Extended information for debug interface |
| \*p\_hwrsc\_mng\_table | In/Out | Extended information for debug interface |
| **Member**  **Valid value** | **Name** | **Value** | |
| instance\_num | Follows the e\_impdrv\_instance\_t. enumeration. | |
| Use\_core\_ins | Bit position of core number | |
| cbfunc\_fatal | User-specified value of the Callback function address. | |
| P\_cb\_args\_fatal | User-specified value of the Callback function argument. | |
| Is\_mutex\_locked | true or false | |
| irq\_mask\_value | Not 0 | |
| protect\_mode | Follows the [e\_impdrv\_protect\_mode\_t](#_e_impdrv_protect_mode_t) enumeration. | |
| \*p\_hwrsc\_def\_table | Debug interface extended information address. | |
| \*p\_hwrsc\_mng\_table | Debug interface extended information address. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_cmn\_init\_data\_t

Table 5‑15: st\_impdrv\_cmn\_init\_data\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  e\_impdrv\_instance\_t instance\_num;  e\_osal\_interrupt\_priority\_t irq\_priority;  st\_impdrv\_core\_info\_t \*p\_core\_info;  uint32\_t use\_core\_num;  impdrv\_ctrl\_handle\_t osal\_cb\_args;  p\_impdrv\_cbfunc\_fatal\_t cbfunc\_fatal;  void \*p\_cb\_args\_fatal;  osal\_mutex\_id\_t mutex\_id;  osal\_milli\_sec\_t time\_period;  } st\_impdrv\_cmn\_init\_data\_t; | | |
| **Description** | IMPDRV structure of common control init data. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| instance\_num | In/Out | Instance number. |
| Irq\_priority | In/Out | Interrupt priority |
| \*p\_core\_info | In/Out | Related core number |
| use\_core\_num | In/Out | Related core information list |
| osal\_cb\_args | In/Out | OSAL callback argument |
| cbfunc\_fatal | In/Out | Fatal callback function |
| \*p\_cb\_args\_fatal | In/Out | Fatal callback argument |
| mutex\_id | In/Out | OSAL mutex id |
| time\_period | In/Out | Mutex timeout value by [msec]order |
| **Member**  **Valid value** | **Name** | **Value** | |
| instance\_num | Follows the e\_impdrv\_instance\_t. enumeration. | |
| Irq\_priority | Priority for interrupt. | |
| \*p\_core\_info | User-specified value of the Callback function address. | |
| Use\_core\_num | Core Number; | |
| osal\_cb\_args | Follows the impdrv\_ctrl\_handle\_t enumeration. | |
| Cbfunc\_fatal | Follows the p\_impdrv\_cbfunc\_fatal\_t enumeration. | |
| \*p\_cb\_args\_fatal | User-specified value of the Callback function argument. | |
| Mutex\_id | ID for mutex. | |
| Time\_period | Time in milliseconds. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_corectl\_func\_t

Table 5‑16: st\_impdrv\_corectl\_func\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  bool (\*p\_impdrvCorectlIsValidCore)(  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlCheckState)(  const st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_state\_t state  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetState)(  st\_impdrv\_corectl\_t \* const p\_corectl,  const e\_impdrv\_state\_t state  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlInitStart)(  st\_impdrv\_coretypectl\_t \* const p\_coretypectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlInitEnd)(  st\_impdrv\_coretypectl\_t \* const p\_coretypectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlStart)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl ,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlStop)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlAttInit)(  st\_impdrv\_corectl\_t \* const p\_corectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetMemInit)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_param\_t enable  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetCoreMap)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const uint8\_t syncc\_val[IMPDRV\_COREMAP\_MAXID]  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetCl)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const uint32\_t claddr\_phys  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetIrqMask)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const bool irq\_mask[IMPDRV\_IRQMASK\_MAX]  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlExecute)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const p\_impdrv\_cbfunc\_t callback\_func,  void \*const p\_callback\_args  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlIntHandler)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlResumeExe)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmSetPolicy)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const e\_impdrv\_pm\_policy\_t policy  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmGetPolicy)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  e\_impdrv\_pm\_policy\_t \*const p\_policy  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetDsp)(  st\_impdrv\_corectl\_t \*p\_corectl,  const uint32\_t core\_num,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_app,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_fw,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_data,  const st\_impdrv\_dsp\_data\_t \*const p\_dsp\_dtcm  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlBusIfCheck)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const st\_impdrv\_chk\_resource\_t \*const p\_chk\_resource  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlConfRegCheck)(  const st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  const st\_impdrv\_chk\_param\_t \*const p\_chk\_param,  const uint32\_t param\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetClBrkAddr)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const uint32\_t cl\_brk\_addr  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetGosubCond)(  st\_impdrv\_corectl\_t \*const p\_corectl,  const uint32\_t core\_num,  const e\_impdrv\_gosub\_cond\_t condition  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmGetState)(  st\_impdrv\_coretypectl\_t \*const p\_coretypectl,  const uint32\_t core\_num,  e\_impdrv\_pm\_state\_t \*const p\_pmstate  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlQuit)(  const uint32\_t core\_num  );  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSyncStart)(  st\_impdrv\_coretypectl\_t coretype\_ctl[IMPDRV\_CORE\_TYPE\_MAX],  const uint32\_t core\_num  ); /\*\*< Function table for distinguishing each core. Starts controlling the core in sync. \*/  e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSyncStop)(  st\_impdrv\_coretypectl\_t coretype\_ctl[IMPDRV\_CORE\_TYPE\_MAX],  const uint32\_t core\_num  ); /\*\*< Function table for distinguishing each core. Stop controlling the core in sync. \*/  } st\_impdrv\_corectl\_func\_t; | | |
| **Description** | Core control interface structure used by common control subcomponents.  This structure is used only with s\_corectl\_func\_tbl. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| p\_impdrvCorectlIsValidCore | In | Function table for distinguishing each core. Valid core. |
| P\_impdrvCorectlCheckState | In | Function table for distinguishing each core. Check state. |
| P\_impdrvCorectlSetState | In | Function table for distinguishing each core. Set state. |
| P\_impdrvCorectlInitStart | In | Function table for distinguishing each core. Init start. |
| P\_impdrvCorectlInitEnd | In | Function table for distinguishing each core. Init end. |
| P\_impdrvCorectlStart | In | Function table for distinguishing each core. Start core control. |
| P\_impdrvCorectlStop | In | Function table for distinguishing each core. Stop core control. |
| P\_impdrvCorectlAttInit | In | Function table for distinguishing each core. Attribute initialize. |
| P\_impdrvCorectlSetMemInit | In | Function table for distinguishing each core. Set core memory initialize. |
| P\_impdrvCorectlSetCoreMap | In | Function table for distinguishing each core. Set sync core map. |
| P\_impdrvCorectlSetCl | In | Function table for distinguishing each core. Set CL address. |
| P\_impdrvCorectlSetIrqMask | In | Function table for distinguishing each core. Set IRQ mask setting. |
| P\_impdrvCorectlExecute | In | Function table for distinguishing each core. Execute core control. |
| P\_impdrvCorectlIntHandler | In | Function table for distinguishing each core. Int handler imp control. |
| P\_impdrvCorectlResumeExe | In | Function table for distinguishing each core. Resume core control. |
| P\_impdrvCorectlPmSetPolicy | In | Function table for distinguishing each core. Pm set policy core control. |
| P\_impdrvCorectlPmGetPolicy | In | Function table for distinguishing each core. Pm get policy core control. |
| p\_impdrvCorectlSetDsp | In | Function table for distinguishing only DSP core. Attribute settings of DSP information. |
| P\_impdrvCorectlBusIfCheck | In | Function table for distinguishing each core. Interface bus check of core control. |
| P\_impdrvCorectlConfRegCheck | In | Function table for distinguishing each core. Configuration register check of core control. |
| p\_impdrvCorectlSetClBrkAddr | In | Function table for distinguishing each core. Set CL break address. |
| p\_impdrvCorectlSetGosubCond | In | Function table for distinguishing each core. Set Conditional GOSUB instruction attribute. |
| p\_impdrvCorectlPmGetState | In | Function table for distinguishing each core. Pm get state core control. |
| p\_impdrvCorectlQuit | In | Function table for distinguishing each core. Quit core control. |
| **Member**  **Valid value** | **Name** | **Value** | |
| p\_impdrvCorectlIsValidCore | Must be Not NULL, Set the address for interface function in each core control of sub component. | |
| P\_impdrvCorectlCheckState | Same as above. | |
| P\_impdrvCorectlSetState | Same as above. | |
| P\_impdrvCorectlInitStart | Same as above. | |
| P\_impdrvCorectlInitEnd | Same as above. | |
| P\_impdrvCorectlStart | Same as above. | |
| P\_impdrvCorectlStop | Same as above. | |
| P\_impdrvCorectlAttInit | Same as above. | |
| P\_impdrvCorectlSetMemInit | Same as above. | |
| P\_impdrvCorectlSetCoreMap | Same as above. | |
| P\_impdrvCorectlSetCl | Same as above. | |
| P\_impdrvCorectlSetIrqMask | Same as above. | |
| P\_impdrvCorectlExecute | Same as above. | |
| P\_impdrvCorectlIntHandler | Same as above. | |
| P\_impdrvCorectlResumeExe | Same as above. | |
| P\_impdrvCorectlPmSetPolicy | Same as above. | |
| P\_impdrvCorectlPmGetPolicy | Same as above. | |
| p\_impdrvCorectlSetDsp | Same as above. | |
| P\_impdrvCorectlBusIfCheck | Same as above. | |
| P\_impdrvCorectlConfRegCheck | Same as above. | |
| p\_impdrvCorectlSetClBrkAddr | Same as above. | |
| p\_impdrvCorectlSetGosubCond | Same as above. | |
| p\_impdrvCorectlPmGetState | Same as above. | |
| p\_impdrvCorectlQuit | Same as above. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_inten\_tbl\_t

Table 5‑17: st\_impdrv\_inten\_tbl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct st\_impdrv\_inten\_tbl  {  e\_impdrv\_core\_type\_t core\_type;  uint32\_t core\_num;  uint32\_t bit;  } st\_impdrv\_inten\_tbl\_t; | | |
| **Description** | Definition for bit allocation conversion of the Interrupt enable register | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| core\_type | In | IMP Core type |
| core\_num | In | Core number in the core type |
| bit | In | Bit value to convert |
| **Member**  **Valid value** | **Name** | **Value** | |
| core\_type | Follows the e\_impdrv\_core\_type\_t enumeration. | |
| Core\_num | Less than maximum number of the Cores for each types. | |
| Bit | Bit assignment value of the Interrupt enable register. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_group\_tbl\_t

Table 5‑18: st\_impdrv\_group\_tbl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct st\_impdrv\_group\_tbl  {  uint32\_t group\_bit;  uint32\_t group\_reg;  } st\_impdrv\_group\_tbl\_t; | | |
| **Description** | Definition for bit allocation conversion of the Interrupt group register | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| group\_bit | In | Bit value to convert |
| group\_reg | In | Offset address value to convert |
| **Member**  **Valid value** | **Name** | **Value** | |
| group\_bit | Bit assignment value of interrupt enable register. | |
| Group\_reg | Offset address of interrupt select register. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_syncc\_table\_t

Table 5‑19: st\_impdrv\_syncc\_table\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  e\_impdrv\_core\_type\_t core\_type;  uint32\_t core\_num;  uint8\_t syncc\_val;  } st\_impdrv\_syncc\_table\_t; | | |
| **Description** | Definition for conversion of the syncc value | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| core\_type | In | IMP Core type |
| core\_num | In | Core number in the core type |
| syncc\_val | In | Syncc value to convert |
| **Member**  **Valid value** | **Name** | **Value** | |
| core\_type | Follows the e\_impdrv\_core\_type\_t enumeration. | |
| Core\_num | Less than maximum number of the Cores for each types. | |
| Syncc\_val | Value of syncc register | |
| **Remarks** | N/A | | |

##### st\_impdrv\_inten\_reg\_tbl\_t

Table 5‑20: st\_impdrv\_inten\_reg\_tbl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  const st\_impdrv\_group\_tbl\_t \*p\_inten\_group\_table;  const st\_impdrv\_inten\_tbl\_t \*p\_inten\_cnv\_table;  uint32\_t cnv\_table\_num;  } st\_impdrv\_inten\_reg\_tbl\_t; | | |
| **Description** | Definition for conversion of register table | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| \*p\_inten\_group\_table | In | Convert Int group register table |
| \*p\_inten\_cnv\_table | In | Convert Int enable register table |
| cnv\_table\_num | In | Number of convert Int enable register table |
| **Member**  **Valid value** | **Name** | **Value** | |
| \*p\_inten\_group\_table | Follows the st\_impdrv\_group\_tbl\_t structure. | |
| \*p\_inten\_cnv\_table | Follows the st\_impdrv\_inten\_tbl\_t structure. | |
| Cnv\_table\_num | Maximum number of cores for the Interrupt group. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_hwrsc\_def\_t

Table 5‑21: st\_impdrv\_hwrsc\_def\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  const char \*name;  uintptr\_t addr\_phys;  uint32\_t size;  bool read\_only;  bool internal;  } st\_impdrv\_hwrsc\_def\_t;; | | |
| **Description** | Hardware resource definition table. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| \*name | In/out | Resource Name(Key) |
| addr\_phys | In/out | Physical address |
| size | In/out | Area size |
| read\_only | In/out | Read only attribute |
| internal | In/out | Internal resource attribute |
| **Member**  **Valid value** | **Name** | **Value** | |
| \*name | Hardware resource name | |
| addr\_phys | Top address of hardware register | |
| size | Area size of hardware register | |
| read\_only | true or false | |
| internal | true or false | |
| **Remarks** | N/A | | |

##### st\_impdrv\_hwrsc\_mng\_t

Table 5‑22: st\_impdrv\_hwrsc\_mng\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t resource\_used;  uint32\_t memory\_initialized;  uint32\_t in\_use[IMPDRV\_HWRSC\_TABLE\_NUM];  st\_impdrv\_device\_handle\_t handle[IMPDRV\_HWRSC\_TABLE\_NUM];  } st\_impdrv\_hwrsc\_mng\_t; | | |
| **Description** | Common control global area. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| resource\_used | In/out | Resource used state |
| memory\_initialized | In/out | Memory initialized state |
| in\_use[IMPDRV\_HWRSC\_TABLE\_NUM] | In/out | In-use status management |
| handle[IMPDRV\_HWRSC\_TABLE\_NUM] | In/out | IMPDRV device handle |
| **Member**  **Valid value** | **Name** | **Value** | |
| resource\_used | IMPDRV\_HWRSC\_UNUSED or follows the s\_impdrv\_used\_instance | |
| memory\_initialized | IMPDRV\_HWRSC\_UNUSED or follows the s\_impdrv\_used\_instance | |
| in\_use | Follows the s\_impdrv\_used\_instance | |
| handle | Follows the st\_impdrv\_device\_handle\_t structure. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_hwrsc\_def\_t

Table 5‑23: st\_impdrv\_hwrsc\_def\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  const char \*name;  uintptr\_t addr\_phys;  uint32\_t size;  bool read\_only;  bool internal;  } st\_impdrv\_hwrsc\_def\_t; | | |
| **Description** | Hardware resource definition table. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| \*name | In/out | Resource Name(Key) |
| addr\_phys | In/out | Physical address |
| size | In/out | Area size |
| read\_only | In/out | Read only attribute |
| internal | In/out | Internal resource attribute |
| **Member**  **Valid value** | **Name** | **Value** | |
| \*name | Hardware resource name | |
| addr\_phys | Top address of hardware register | |
| size | Area size of hardware register | |
| read\_only | true or false | |
| internal | true or false | |
| **Remarks** | N/A | | |

##### st\_impdrv\_commonctl\_ins\_t

Table 5‑24: st\_impdrv\_commonctl\_ins\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  st\_impdrv\_device\_handle\_t device\_io\_imptop;  st\_impdrv\_mutex\_handle\_t mutex\_handle;  osal\_mutex\_id\_t mutex\_id;  osal\_milli\_sec\_t mutex\_time\_period;  st\_impdrv\_device\_handle\_t device\_io\_dta;  st\_impdrv\_commonctl\_t \*p\_commonctl[IMPDRV\_INSTANCETABLE\_NUM];  } st\_impdrv\_commonctl\_ins\_t; | | |
| **Description** | Hardware resource definition table. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| device\_io\_imptop | In/out | OSAL device handle (IMP TOP). |
| Mutex\_handle | In/out | OSAL mutex handle |
| mutex\_id | In/out | OSAL mutex id |
| mutex\_time\_period | In/out | Millisecond Order Mutex Timeout value |
| device\_io\_dta | In/out | OSAL device handle (DTA Core). |
| \*p\_commonctl | In/out | Address of the commonctl. |
| **Member**  **Valid value** | **Name** | **Value** | |
| device\_io\_imptop | Follows the st\_impdrv\_device\_handle\_t structure. | |
| Mutex\_handle | Follows the st\_impdrv\_mutex\_handle\_t structure. | |
| Mutex\_id | ID for mutex. | |
| Mutex\_time\_period | Time in milliseconds. | |
| Device\_io\_dta | Follows the st\_impdrv\_device\_handle\_t structure. | |
| \*p\_commonctl | Follows the st\_impdrv\_commonctl\_t structure. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_commonctl\_chk\_t

Table 5‑25: st\_impdrv\_commonctl\_chk\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t checksum;  st\_impdrv\_device\_handle\_t device\_io\_imptop;  } st\_impdrv\_commonctl\_chk\_t | | |
| **Description** | Common control global check area. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| checksum | In/out | Checksum value of Global variable. |
| Device\_io\_imptop | In/out | Backup of OSAL device handle(IMP TOP). |
| **Member**  **Valid value** | **Name** | **Value** | |
| checksum | Checksum value of Global variable. | |
| Device\_io\_imptop | Follows the st\_impdrv\_device\_handle\_t structure. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_hwrsc\_chk\_t

Table 5‑26: st\_impdrv\_hwrsc\_chk\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t checksum;  } st\_impdrv\_hwrsc\_chk\_t | | |
| **Description** | Common control global check area. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| checksum | In/out | Checksum value of Global variable. |
| **Member**  **Valid value** | **Name** | **Value** | |
| checksum | Checksum value of Global variable. | |
| **Remarks** | N/A | | |

#### Core Control

##### st\_impdrv\_coretypectl\_t

Table 5‑27: st\_impdrv\_coretypectl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct{  st\_impdrv\_corectl\_t core\_ctl[IMPDRV\_CORE\_NUM\_MAX];  } st\_impdrv\_coretypectl\_t; | | |
| **Description** | Each core control data structure used by core control subcomponents. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| core\_ctl | In/Out | Array of The Core Control handles. |
| **Member**  **Valid value** | **Name** | **Value** | |
| core\_ctl | Follows the st\_impdrv\_corectl\_t structure. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_corectl\_t

Table 5‑28: st\_impdrv\_corectl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  st\_impdrv\_device\_handle\_t device\_io;  st\_impdrv\_queue\_handle\_t queue\_handle;  st\_impdrv\_cbinfo\_t cbinfo;  e\_impdrv\_state\_t core\_state;  e\_impdrv\_param\_t core\_mem\_init\_enable;  uint32\_t claddr\_phys;  uint8\_t cur\_core\_map[IMPDRV\_COREMAP\_MAXID];  bool irq\_mask[IMPDRV\_IRQMASK\_MAX];  bool is\_progress\_bus\_if\_check;  uint8\_t reserve[IMPDRV\_RESERVE\_NUM];  uint32\_t param[IMPDRV\_EXEPARAM\_MAX];  } st\_impdrv\_corectl\_t; | | |
| **Description** | Each core control data structure used by core control subcomponents. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| device\_io | In/Out | Device handle |
| queue\_handle | In/Out | Message queue handle |
| cbinfo | In/Out | Callback function information |
| core\_state | In/Out | IMP Driver state |
| core\_mem\_init\_enable | In/Out | memory initialization possible flag |
| claddr\_phys | In/Out | CL Physical address |
| cur\_core\_map | In/Out | Current core map |
| irq\_mask | In/Out | Mask of IRQ Grouping |
| is\_progress\_bus\_if\_check | In/Out | Bus interface check in progress |
| reserve | In/Out | Unused area |
| param | In/Out | Execution parameters |
| **Member**  **Valid value** | **Name** | **Value** | |
| device\_io | Follows the st\_impdrv\_device\_handle\_t structure. | |
| Queue\_handle | Follows the st\_impdrv\_queue\_handle\_t structure. | |
| Cbinfo | Follows the st\_impdrv\_cbinfo\_t structure. | |
| Core\_state | Follows the e\_impdrv\_state\_t enumeration. | |
| Core\_mem\_init\_enable | Disable (0u) or Enable (1u). | |
| claddr\_phys | 32-bit physical memory address. | |
| Cur\_core\_map | Valid values of the SYNCC register. | |
| Irq\_mask | Valid values of the Interrupt mask register for each cores. | |
| Is\_progress\_bus\_if\_check | Disable (0u) or Enable (1u). | |
| reserve | N/A | |
| param | Parameter arrays of disable (0u) or enable (1u). | |
| **Remarks** | N/A | | |

##### st\_impdrv\_cbinfo\_t

Table 5‑29: st\_impdrv\_cbinfo\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  p\_impdrv\_cbfunc\_t cbfunc;  void \*p\_cbarg;  } st\_impdrv\_cbinfo\_t; | | |
| **Description** |  | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| cbfunc | In/Out | Callback function address |
| p\_cbarg | In/Out | Address of the callback function argument. |
| **Member**  **Valid value** | **Name** | **Value** | |
| cbfunc | User-specified value of the Callback function address. | |
| P\_cbarg | User-specified value of the Callback function argument. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_dma\_write\_info\_t

Table 5‑30: st\_impdrv\_dma\_write\_info\_t

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Format** | typedef struct  {  st\_impdrv\_device\_handle\_t \*p\_device\_handle;  uint32\_t offset;  uint32\_t val;  bool is\_sub\_thread;  bool read\_back;  uint32\_t chk\_val;  } st\_impdrv\_dma\_write\_info\_t; | | | | | | | |
| **Description** | Write register information. | | | | | | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** | | | | | |
| \*p\_device\_handle | In | OSAL device handle. | | | | | |
| Offset | In | Offset to set to register. | | | | | |
| **Range** | Less than IMPDRV\_REG\_OFFSET\_MAX and 4 byte alignment. | | | | |
| Val | In | Write values to registers. | | | | | |
| **Range** | | None. | | | |
| Is\_sub\_thread | In | Select for DMAC Internal thread. | | | | | |
| **Range** | | | true or false. | | |
| Read\_back; | In | Whether to read back. | | | | | |
| **Range** | | | | true or false. | |
| Chk\_val | In | Value to read back register. | | | | | |
| **Range** | | | | | None. |
| **Member**  **Valid value** | **Name** | **Value** | | | | | | |
| \*p\_device\_handle | Follows the st\_impdrv\_device\_handle\_t structure. | | | | | | |
| Offset | Offset address of hardware register. | | | | | | |
| Val | Valid value to set to register. | | | | | | |
| Is\_sub\_thread | true or false. | | | | | | |
| Read\_back | true or false. | | | | | | |
| Chk\_val | Valid value to read back register. | | | | | | |
| **Remarks** | N/A | | | | | | | |

##### st\_impdrv\_dma\_write\_info\_tbl\_t

Table 5‑31: st\_impdrv\_dma\_write\_info\_tbl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t offset;  uint32\_t val;  bool read\_back;  uint32\_t chk\_val;  } st\_impdrv\_dma\_write\_info\_tbl\_t; | | |
| **Description** | Multi-bank memory initialization information | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| offset | In | Offset to get to register |
| val | In | Write values to registers |
| read\_back | In | Whether to read back |
| chk\_val | In | Value to read back register |
| **Member**  **Valid value** | **Name** | **Value** | |
| offset | Offset address of hardware register. | |
| Val | Valid value to set to register. | |
| Read\_back | true or false. | |
| Chk\_val | Valid value to read back register. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_imp\_write\_info\_tbl\_t

Table 5‑32: st\_impdrv\_imp\_write\_info\_tbl\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  uint32\_t offset;  uint32\_t val;  bool read\_back;  uint32\_t chk\_val;  } st\_impdrv\_imp\_write\_info\_tbl\_t; | | |
| **Description** | Multi-bank memory initialization information | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| offset | In | Offset to get to register |
| val | In | Write values to registers |
| read\_back | In | Whether to read back |
| chk\_val | In | Value to read back register |
| **Member**  **Valid value** | **Name** | **Value** | |
| offset | Offset address of hardware register. | |
| Val | Valid value to set to register. | |
| Read\_back | true or false. | |
| Chk\_val | Valid value to read back register. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_dsp\_info\_t

(Out of scope for xOS2.0)

Table 5‑33: st\_impdrv\_dsp\_info\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  st\_impdrv\_dsp\_data\_t dsp\_app;  st\_impdrv\_dsp\_data\_t dsp\_fw;  st\_impdrv\_dsp\_data\_t dsp\_data;  bool is\_updated;  bool is\_loaded;  } st\_impdrv\_dsp\_info\_t; | | |
| **Description** | - | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| st\_impdrv\_dsp\_data\_t | - | - |
| st\_impdrv\_dsp\_data\_t | - | - |
| st\_impdrv\_dsp\_data\_t | - | - |
| bool | - | - |
| bool | - | - |
| **Member**  **Valid value** | **Name** | **Value** | |
| st\_impdrv\_dsp\_data\_t | - | |
| st\_impdrv\_dsp\_data\_t | - | |
| st\_impdrv\_dsp\_data\_t | - | |
| bool | - | |
| bool | - | |
| **Remarks** | N/A | | |

#### OS Dependence Layer

##### st\_impdrv\_device\_handle\_t

Table 5‑34: st\_impdrv\_device\_handle\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_device\_handle\_t handle;  } st\_impdrv\_device\_handle\_t; | | |
| **Description** | OSAL device handle.  When NULL, the device is not open, and when it is not NULL, the device is open.  IMP Driver checks the above state and detects sequence errors. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| handle | In/Out | OSAL device handle |
| **Member**  **Valid value** | **Name** | **Value** | |
| handle | Valid value of the handle numbered by OSAL. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_mutex\_handle\_t

Table 5‑35: st\_impdrv\_mutex\_handle\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_mutex\_handle\_t handle;  } st\_impdrv\_mutex\_handle\_t; | | |
| **Description** | OSAL mutex handle.  When NULL, the mutex is not create, and when it is not NULL, the mutex is create.  IMP Driver checks the above state and detects sequence errors. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| handle | In/Out | OSAL mutex handle |
| **Member**  **Valid value** | **Name** | **Value** | |
| handle | Valid value of the handle numbered by OSAL. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_queue\_handle\_t

Table 5‑36: st\_impdrv\_queue\_handle\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_mq\_handle\_t handle;  } st\_impdrv\_queue\_handle\_t; | | |
| **Description** | OSAL message queue handle.  When NULL, the message queue is not create, and when it is not NULL, the message queue is create. IMP Driver checks the above state and detects sequence errors. | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| handle | In/Out | OSAL message queue handle |
| **Member**  **Valid value** | **Name** | **Value** | |
| handle | Valid value of the handle numbered by OSAL. | |
| **Remarks** | N/A | | |

##### st\_impdrv\_memory\_handle\_t

Table 5‑37: st\_impdrv\_memory\_handle\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef struct  {  osal\_memory\_buffer\_handle\_t handle;  } st\_impdrv\_memory\_handle\_t; | | |
| **Description** | OSAL memory buffer handle . | | |
| **Member**  **Description** | **Name** | **In/out** | **Description** |
| handle | In/Out | OSAL memory buffer handle |
| **Member**  **Valid value** | **Name** | **Value** | |
| handle | Valid value of the handle numbered by OSAL. | |
| **Remarks** | N/A | | |

## Global Pointer Variables

Not use.

## Global Variables

### API Layer

None.

### General Control

None.

### Common Control

#### g\_impdrv\_cmn\_ctl\_ins

Table 5‑38: g\_impdrv\_cmn\_ctl\_ins

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  st\_impdrv\_commonctl\_ins\_t g\_impdrv\_cmn\_ctl\_ins =  {  { NULL }, /\*\*< device\_io\_imptop \*/  { NULL }, /\*\*< mutex\_handle \*/  0U, /\*\*< mutex\_id \*/  0, /\*\*< mutex\_time\_period \*/  { NULL }, /\*\*< device\_io\_dta \*/  { NULL, NULL, NULL, NULL, NULL, NULL }  /\*\*< \*p\_commonctl[IMPDRV\_INSTANCETABLE\_NUM] \*/  }; |
| **Description** | IMP Driver determines conversion of Instance table by this table.  Follows the st\_impdrv\_commonctl\_ins\_t structure. |
| **Remarks** | This variable is used only by its own sub-component. |

#### g\_hwrsc\_mng\_table

Table 5‑39: g\_hwrsc\_mng\_table

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  [st\_impdrv\_hwrsc\_mng\_t](#_st_impdrv_hwrsc_mng_t) g\_hwrsc\_mng\_table =  {  IMPDRV\_HWRSC\_UNUSED, /\*\*< Resource used state \*/  IMPDRV\_HWRSC\_UNUSED, /\*\*< Memory initialized state \*/  {  IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED,  IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED,  IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED,  IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED, IMPDRV\_HWRSC\_UNUSED  }, /\*\*< In-use status management \*/  {  { NULL }, { NULL }, { NULL }, { NULL },  { NULL }, { NULL }, { NULL }, { NULL },  { NULL }, { NULL }, { NULL }, { NULL },  { NULL }, { NULL }, { NULL }, { NULL }  } /\*\*< IMPDRV device handle \*/  }; |
| **Description** | IMP Driver determines conversion of register table by this table.  Follows the [st\_impdrv\_hwrsc\_mng\_t](#_st_impdrv_hwrsc_mng_t) structure. |
| **Remarks** | This variable is used only by its own sub-component. |

#### g\_impdrv\_cmn\_ctl\_chk

Table 5‑40: g\_impdrv\_cmn\_ctl\_chk

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  st\_impdrv\_commonctl\_chk\_t g\_impdrv\_cmn\_ctl\_chk =  {  0U, /\*\*< Checksum data for g\_impdrv\_cmn\_ctl\_ins \*/  { NULL }, /\*\*< Backup data of device\_io\_imptop \*/  }; |
| **Description** | Common control global check area.  Follows the st\_impdrv\_commonctl\_chk\_t structure. |
| **Remarks** | This variable is used only by its own sub-component. |

#### g\_hwrsc\_chk\_table

Table 5‑41: g\_hwrsc\_chk\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  st\_impdrv\_hwrsc\_chk\_t g\_hwrsc\_chk\_table =  {  0U, /\*\*< Checksum data for g\_hwrsc\_mng\_table \*/  }; |
| **Description** | Common control global check area.  Follows the st\_impdrv\_hwrsc\_chk\_t structure. |
| **Remarks** | This variable is used only by its own sub-component. |

### Core Control

None.

### OS Dependence Layer

None.

## Constants

### Definition

#### API Layer

Table 5‑42: Definition Values (API Layer)

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_CHK\_WORK\_ADDR\_ALIGN | (7U) | The work area address alignment check definition |
| IMPDRV\_COREMAP\_MAXID | (16U) | Number of elements in the core map array |
| IMPDRV\_IRQGROUP\_MAXID | (22U) | Core specification limits for IRQ groups |
| IMPDRV\_IRQMASK\_MAX | (9U) | Number of elements in the IRQ mask array |
| IMPDRV\_INNER\_FIXED\_VALUE | (22U) | Definition values for internal management |
| IMPDRV\_CLBRK\_ADDR\_INVALID | (0xFFFFFFFFU) | Invalid value of CL break address data |
| IMPDRV\_VDSP\_INT\_PROG\_MEM\_SIZE | (0x2000U) | Maximum size of the dsp\_fw |
| IMPDRV\_VDSP\_DTCM\_MEM\_SIZE | (0x40000U) | Maximum size of the dsp\_dtcm |
| IMPDRV\_VERSION\_MAJOR | - | Major version number of IMP driver  Refer to r\_impdrv\_version.h for the major version value. |
| IMPDRV\_VERSION\_MINOR | - | Minor version number of IMP driver  Refer to r\_impdrv\_version.h for the minor version value. |
| IMPDRV\_VERSION\_PATCH | - | Patch version number of IMP driver  Refer to r\_impdrv\_version.h for the patch version value. |
| IMPDRV\_HANDLE\_INVALID | ((impdrv\_ctrl\_handle\_t)NULL) | Invalid control handle value |

#### Common definition (All sub components)

Table 5‑43: Definition Values (Common definition)

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_STATIC | static | Static declaration of IMP driver for Normal build |
| IMPDRV\_UNUSED\_ARGS(a) | ((void)(a)) | Workaround for warning fix.  This is used in the developing. Will be removed eventually. |
| IMPDRV\_CHK\_MEM\_ALIGN | (3U) | Memory alignment check definition |
| IMPDRV\_CHK\_REG\_ALIGN | (3U) | Register alignment check definition |
| IMPDRV\_CHK\_CL\_MEM\_ALIGN | (3U) | CL Memory alignment check definition |
| IMPDRV\_CHK\_CACHE\_ALIGN | (63U) | Memory cache alignment check definition |
| IMPDRV\_CORE\_TYPE\_MAX | (11U) | Maximum value of Core type definition |
| IMPDRV\_DSP\_DEV\_NUM\_MAX | (8U) | num of max sub device for DSP core |

#### General Control

None.

#### Common Control

Table 5‑44: Definition Values (Common Control)

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_REG\_INTEN00 | (0x0020U) | Offset address of hardware register for INTEN00. |
| IMPDRV\_REG\_INTEN01 | (0x0024U) | Offset address of hardware register for INTEN01. |
| IMPDRV\_REG\_INTEN02 | (0x0028U) | Offset address of hardware register for INTEN02. |
| IMPDRV\_REG\_G00INTSEL | (0x0030U) | Offset address of hardware register for G00INTSEL. |
| IMPDRV\_REG\_G01INTSEL | (0x0034U) | Offset address of hardware register for G01INTSEL. |
| IMPDRV\_REG\_G02INTSEL | (0x0038U) | Offset address of hardware register for G02INTSEL. |
| IMPDRV\_REG\_INTEN10 | (0x0040U) | Offset address of hardware register for INTEN10. |
| IMPDRV\_REG\_INTEN11 | (0x0044U) | Offset address of hardware register for INTEN11. |
| IMPDRV\_REG\_INTEN12 | (0x0048U) | Offset address of hardware register for INTEN12. |
| IMPDRV\_REG\_G10INTSEL | (0x0050U) | Offset address of hardware register for G10INTSEL. |
| IMPDRV\_REG\_G11INTSEL | (0x0054U) | Offset address of hardware register for G11INTSEL. |
| IMPDRV\_REG\_G12INTSEL | (0x0058U) | Offset address of hardware register for G12INTSEL. |
| IMPDRV\_REG\_MINTEN | (0x00B0U) | Offset address of hardware register for MINTEN. |
| IMPDRV\_REG\_MG0INTSEL | (0x00B4U) | Offset address of hardware register for MG0INTSEL. |
| IMPDRV\_REG\_MG1INTSEL | (0x00B8U) | Offset address of hardware register for MG1INTSEL. |
| IMPDRV\_REG\_MG2INTSEL | (0x00BCU) | Offset address of hardware register for MG2INTSEL. |
| IMPDRV\_REG\_INTEN\_INIT\_VAL | (0x00000000U) | Initial value of interrupt enable register. |
| IMPDRV\_INTEN0\_CORE\_MAX | (11U) | Maximum number of cores for the Interrupt group 1. |
| IMPDRV\_INTEN1\_CORE\_MAX | (10U) | Maximum number of cores for the Interrupt group 2. |
| IMPDRV\_MINTEN\_CORE\_MAX | (21U) | Maximum number of cores for the Merge interrupt group. |
| IMPDRV\_GROUP\_MAX | (3U) | Maximum number of interrupt groups. |
| IMPDRV\_INSTANCETABLE\_NUM | (8U) | Maximum number of Instance number table |
| IMPDRV\_IRQCHTABLE\_NUM | (7U) | Maximum number of IRQ Channel table. |
| IMPDRV\_SYNCCTABLE\_NUM | (23U) | Num of SYNCC table |
| IMPDRV\_INTSTS\_BITNUM | (32U) | Bit width of Interrupt status register |
| IMPDRV\_REG\_INTSTS0 | (0x0010U) | Offset address of hardware register for IMPDRV\_REG\_INTSTS0 |
| IMPDRV\_REG\_INTSTS1 | (0x0014U) | Offset address of hardware register for IMPDRV\_REG\_INTSTS1 |
| IMPDRV\_REG\_INTSTS2 | (0x0018U) | Offset address of hardware register for IMPDRV\_REG\_INTSTS2 |

Table 5‑45: Definition Values (Common Control) [R-CarV3M / V3H / V3H\_2]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_INTSTS\_BITNUM | (32U) | Bit width of Interrupt status register. |
| IMPDRV\_HWRSC\_TABLE\_NUM | (16U) | Maximum number of Resource tables. |
| IMPDRV\_REG\_ISR | (0x0100U) | Offset address of hardware register for ISR. |
| IMPDRV\_REG\_IMR | (0x010CU) | Offset address of hardware register for IMR. |
| IMPDRV\_REG\_G0INTSEL | (0x0110U) | Offset address of hardware register for G0INTSEL. |
| IMPDRV\_REG\_G1INTSEL | (0x0114U) | Offset address of hardware register for G1INTSEL. |
| IMPDRV\_REG\_G2INTSEL | (0x0118U) | Offset address of hardware register for G2INTSEL. |
| IMPDRV\_REG\_IMR1 | (0x0140U) | Offset address of hardware register for IMR1. |
| IMPDRV\_REG\_IMR2 | (0x0144U) | Offset address of hardware register for IMR2. |
| IMPDRV\_REG\_IMR3 | (0x0148U) | Offset address of hardware register for IMR3. |
| IMPDRV\_REG\_IMR4 | (0x014CU) | Offset address of hardware register for IMR4. |
| IMPDRV\_INSTANCETABLE\_NUM | (6U) | Maximum number of Instance number table. |
| IMPDRV\_IRQCHTABLE\_NUM | (5U) | Maximum number of IRQ Channel table. |
| IMPDRV\_HWRSC\_UNUSED | (0U) | Unused resources state. |
| IMPDRV\_HWRSC\_NAME\_MAX | (32U) | Maximum length of hardware resource name. |
| IMPDRV\_HWRSC\_INDEX\_IMP\_TOP | (0U) | Index number of IMP top resource. |
| IMPDRV\_REG\_WPR | (0x8000U) | Offset address of hardware register for WPR. |
| IMPDRV\_REG\_RAMTSTR | (0x8120U) | Offset address of hardware register for RAMTSTR. |
| IMPDRV\_MASK\_L16BIT | (0x0000FFFFU) | Bit mask for Lower 16 bits |
| IMPDRV\_REG\_PROTECT\_MASK | (0x80000000U) | Bitmask for Memory protection mode. |
| IMPDRV\_REG\_GROUP\_INIT\_VAL | (0x00000000U) | Initial value of group interrupt select register. |
| IMPDRV\_REG\_RAMTSTR | (0x8120U) | Offset address of hardware register for RAMTSTR. |
| IMPDRV\_REG\_WPR | (0x8000U) | Offset address of hardware register for WPR. |

Table 5‑46: Definition Values (Common Control) [R-CarV3M]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_REG\_IMR\_INIT\_VAL | (0xE00003FFU) | Initial value of interrupt mask register. |
| IMPDRV\_INTEN\_CORE\_MAX | (10U) | Maximum number of cores for the Interrupt. |
| IMPDRV\_SYNCCTABLE\_NUM | (16U) | Maximum number of SYNCC table. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_ODD | (0x0000002A) | Multi-Bank memory odd initialization value. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_EVEN | (0x00000055) | Multi-Bank memory even initialization value. |
| IMPDRV\_GROUP\_MAX | (5U) | Maximum number of interrupt groups. |

Table 5‑47: Definition Values (Common Control) [R-CarV3H]

|  |  |  |
| --- | --- | --- |
|  |  |  |
| IMPDRV\_REG\_IMR\_INIT\_VAL | (0xE003FFFFU) | Initial value of interrupt mask register. |
| IMPDRV\_INTEN\_CORE\_MAX | (18U) | Maximum number of cores for the Interrupt. |
| IMPDRV\_SYNCCTABLE\_NUM | (30U) | Maximum number of SYNCC table. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_ODD | (0x0000AAAAU) | Multi-Bank memory odd initialization value. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_EVEN | (0x00005555U) | Multi-Bank memory even initialization value. |
| IMPDRV\_GROUP\_MAX | (5U) | Maximum number of interrupt groups. |

Table 5‑48: Definition Values (Common Control) [R-CarV3H\_2]

|  |  |  |
| --- | --- | --- |
|  |  |  |
| IMPDRV\_REG\_IMR\_INIT\_VAL | (0xE007FFFFU) | Initial value of interrupt mask register. |
| IMPDRV\_INTEN\_CORE\_MAX | (19U) | Maximum number of cores for the Interrupt. |
| IMPDRV\_SYNCCTABLE\_NUM | (27U) | Maximum number of SYNCC table. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_ODD | (0x0000AAAAU) | Multi-Bank memory odd initialization value. |
| IMPDRV\_REG\_MB\_MEM\_INIT\_EVEN | (0x00005555U) | Multi-Bank memory even initialization value. |
| IMPDRV\_REG\_SRAMFMR | (0x0314U) | Offset address of hardware register for SRAMFMR. |
| IMPDRV\_REG\_PROTECT\_EDC | (0x00000000U) | EDC mode register value. |
| IMPDRV\_REG\_PROTECT\_ECC | (0x80000000U) | ECC mode register value. |
| IMPDRV\_GROUP\_MAX | (5U) | Maximum number of interrupt groups. |
| IMPDRV\_DUMMY\_UINT32 | (0xFFFFFFFFU) | Uint32\_t dummy value |

Table 5-49 Definition values (Common Control) [R-CarV4H]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_INTSTS\_BITNUM | (32U) | Bit width of Interrupt status register |
| IMPDRV\_HWRSC\_TABLE\_NUM | (16U) | Maximum number of Resource tables |
| IMPDRV\_REG\_INTEN00 | (0x0020U) | Offset address of hardware register for INTEN00 |
| IMPDRV\_REG\_INTEN01 | (0x0024U) | Offset address of hardware register for INTEN01 |
| IMPDRV\_REG\_INTEN02 | (0x0028U) | Offset address of hardware register for INTEN02 |
| IMPDRV\_REG\_G00INTSEL | (0x0030U) | Offset address of hardware register for G00INTSEL |
| IMPDRV\_REG\_G01INTSEL | (0x0034U) | Offset address of hardware register for G01INTSEL |
| IMPDRV\_REG\_G02INTSEL | (0x0038U) | Offset address of hardware register for G02INTSEL |
| IMPDRV\_REG\_INTSTS0 | (0x0010U) | Offset address of hardware register for IMPDRV\_REG\_INTSTS0 |
| IMPDRV\_REG\_INTEN\_INIT\_VAL | (0x00000000U) | Initial value of interrupt enable register |
| IMPDRV\_INTEN\_CORE\_MAX | (18U) | Maximum number of cores for the Interrupt. |
| IMPDRV\_GROUP\_MAX | (5U) | Maximum number of interrupt groups. |
| IMPDRV\_INSTANCETABLE\_NUM | (4U) | Maximum number of Instance number table |
| IMPDRV\_IRQCHTABLE\_NUM | (3U) | Maximum number of IRQ Channel table |
| IMPDRV\_SYNCCTABLE\_NUM | (29U) | Maximum number of SYNCC table |
| IMPDRV\_DUMMY\_UINT32 | (0xFFFFFFFFU) | Uint32\_t dummy value |
| IMPDRV\_REG\_DSP0CT | (0x00004010U) | Offset address of hardware register for DSP0CT |
| IMPDRV\_REG\_DSP1CT | (0x00004020U) | Offset address of hardware register for DSP1CT |
| IMPDRV\_REG\_DSP2CT | (0x00004030U) | Offset address of hardware register for DSP2CT |
| IMPDRV\_REG\_DSP3CT | (0x00004040U) | Offset address of hardware register for DSP3CT |
| IMPDRV\_DSPCORE\_NUM | (4U) | Maximum number of DSP cores |
| IMPDRV\_REG\_DSPNCT\_VAL | (0x0003FFF3U) | The setting value of DSPnCT (V4M) |
| IMPDRV\_REG\_DSPNCT\_VAL | (0x0003FFFFU) | The setting value of DSPnCT (V4H) |
| IMPDRV\_DSP\_CORE\_NUM | (4U) | Maximum number of SYNCC table |
| IMPDRV\_DSP\_SUB\_DEV\_NUM\_MAX | (8U) | Maximum number of SUB DEV table |
| IMPDRV\_HWRSC\_UNUSED | (0U) | Unused resources state |
| IMPDRV\_HWRSC\_NAME\_MAX | (32U) | Maximum length of hardware resource name |
| IMPDRV\_HWRSC\_INDEX\_IMP\_TOP | (0U) | Index number of IMP top resource |
| IMPDRV\_HWRSC\_INIT\_SPMC0\_IDX | (1U) | SPMC0 index number of the init table. |
| IMPDRV\_HWRSC\_INIT\_SPMI0\_IDX | (0U) | SPMI0 index number of the init table. |
| IMPDRV\_HWRSC\_SPM\_INIT\_TBL\_MAX | (2U) | Maximum index number of the init table. |
| IMPDRV\_HWRSC\_SPM\_PSET\_TBL\_MAX | (5U) | Maximum index number of the preset table. |
| IMPDRV\_HWRSC\_INIT\_SPMC0\_IDX | (1U) | SPMC0 index number of the init table. |

#### Core Control

Table 5‑49: Definition Values (Core Control) [R-CarV3M / V3H / V3H\_2]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_MASK\_U8BIT | (0xFF000000U) | Bit mask for Upper 8 bits |
| IMPDRV\_MASK\_U9\_16BIT | (0x00FF0000U) | Bit mask for Upper 9-16 bits |
| IMPDRV\_MASK\_U16BIT | (0xFFFF0000U) | Bit mask for Upper 16 bits |
| IMPDRV\_MASK\_L8BIT | (0x000000FFU) | Bit mask for Lower 8 bits |
| IMPDRV\_MASK\_BIT15 | (0x00008000U) | Bit mask for Only 15 bit |
| IMPDRV\_INIT\_CL\_ADDR | (0xFFFFFFFFU) | Initial value of CL address data |
| IMPDRV\_CALLBACK\_CODE\_DEF | (-1) | Callback code default value |
| IMPDRV\_EXEPARAM\_MAX | (3U) | Number of extended parameter arrays |
| IMPDRV\_CORE\_NUM\_MAX | (8U) | Maximum value of Core number definition |
| IMPDRV\_CB\_RET\_MAX | (18U) | Maximum value of Call back reason |
| IMPDRV\_OFFSET\_4BYTE | (4U) | For 4 byte offset calculation |
| IMPDRV\_SHIFT\_8BIT | (8U) | For 8 bit shift calculation |
| IMPDRV\_SHIFT\_16BIT | (16U) | For 16 bit shift calculation |
| IMPDRV\_SHIFT\_24BIT | (24U) | For 24 bit shift calculation |
| IMPDRV\_SYNCC\_REGNUM | (4U) | Array size for SYNCC register |
| IMPDRV\_IMP\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_IMP\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |

Table 5‑50: Definition Values (Core Control)

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_MASK\_U8BIT | (0xFF000000U) | Bit mask for Upper 8 bits. |
| IMPDRV\_MASK\_U9\_16BIT | (0x00FF0000U) | Bit mask for Upper 9-16 bits. |
| IMPDRV\_MASK\_U16BIT | (0xFFFF0000U) | Mask to extract the upper 16 bits. |
| IMPDRV\_MASK\_L8BIT | (0x000000FFU) | Bit mask for Lower 8 bits. |
| IMPDRV\_MASK\_BIT15 | (0x00008000U) | Bit mask for Only 15 bit. |
| IMPDRV\_INIT\_CL\_ADDR | (0xFFFFFFFFU) | Initial value of CL address data. |
| IMPDRV\_CALLBACK\_CODE\_DEF | (-1) | Callback code default value. |
| IMPDRV\_EXEPARAM\_MAX | (3U) | Maximum number of execution parameters. |
| IMPDRV\_CORE\_NUM\_MAX | (8U) | Maximum value of Core number definition |
| IMPDRV\_CB\_RET\_MAX | (16U) | Maximum value of Call back reason |
| IMPDRV\_OFFSET\_4BYTE | (4U) | For 4 byte offset calculation |
| IMPDRV\_SHIFT\_8BIT | (8U) | For 8 bit shift calculation |
| IMPDRV\_SHIFT\_16BIT | (16U) | For 16 bit shift calculation |
| IMPDRV\_SHIFT\_24BIT | (24U) | For 24 bit shift calculation |
| IMPDRV\_SYNCC\_REGNUM | (4U) | Array size for SYNCC register. |
| IMPDRV\_IMP\_CORENUM\_VAL | (4U) | Maximum number of cores for IMP. |
| IMPDRV\_REG\_IMP\_VCR | (0x0004U) | Offset address of VCR register for IMP. |
| IMPDRV\_REG\_IMP\_CNF | (0x0008U) | Offset address of CNF register for IMP. |
| IMPDRV\_REG\_IMP\_IFCFG | (0x0018U) | Offset address of CNFG register for IMP. |
| IMPDRV\_REG\_IMP\_IFCTL | (0x001CU) | Offset address of IFCTL register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for IMP. |
| IMPDRV\_REG\_IMP\_INTSTS | (0x0010U) | Offset address of INTSTS register for IMP. |
| IMPDRV\_REG\_IMP\_INTEN | (0x0014U) | Offset address of INTEN register for IMP. |
| IMPDRV\_REG\_IMP\_PSA | (0x0020U) | Offset address of PSA register for IMP. |
| IMPDRV\_REG\_IMP\_HMPTR | (0x0514U) | Offset address of HMPTR register for IMP. |
| IMPDRV\_REG\_IMP\_HMDATA | (0x0518U) | Offset address of HMDATA register for IMP. |
| IMPDRV\_REG\_IMP\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for IMP. |
| IMPDRV\_REG\_IMP\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for IMP. |
| IMPDRV\_REG\_IMP\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for IMP. |
| IMPDRV\_IMP\_VCR | (0x00700000U) | Hardware version value for IMP. |
| IMPDRV\_IMP\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for IMP. |
| IMPDRV\_IMP\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_EXE | (0x00000001U) | EXE register value for IMP. |
| IMPDRV\_IMP\_IFCFG\_VAL | (0x010000CCU) | IFCFG\_VAL register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_CLR | (IMPDRV\_IMP\_INTSTS\_TRAP | IMPDRV\_IMP\_INTSTS\_IER | IMPDRV\_IMP\_INTSTS\_INT | IMPDRV\_IMP\_INTSTS\_WUP | IMPDRV\_IMP\_INTSTS\_HPINT | IMPDRV\_IMP\_INTSTS\_APIPINT) | CLR register value for IMP. |
| IMPDRV\_IMP\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for IMP. |
| IMPDRV\_IMP\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for IMP. |
| IMPDRV\_IMP\_MEM\_FUNC\_SA\_VAL | (0x00FFU) | SA\_VAL register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_CORENUM\_VAL | (8U) | Maximum number of cores for OCV. |
| IMPDRV\_REG\_OCV\_VCR0 | (0x0000U) | Offset address of VCR register for OCV. |
| IMPDRV\_REG\_OCV\_RSTR | (0x0008U) | Offset address of RSTR register for OCV. |
| IMPDRV\_REG\_OCV\_CR | (0x000CU) | Offset address of CR register for OCV. |
| IMPDRV\_REG\_OCV\_SR0 | (0x0010U) | Offset address of SR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR1 | (0x0014U) | Offset address of SR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR1 | (0x0018U) | Offset address of SCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR1 | (0x001CU) | Offset address of ICR1 register for OCV. |
| IMPDRV\_REG\_OCV\_IMR1 | (0x0020U) | Offset address of IMR1 register for OCV. |
| IMPDRV\_REG\_OCV\_MCR0 | (0x0040U) | Offset address of MCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_DLSAR | (0x0180U) | Offset address of DLSAR register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR0 | (0x04C0U) | Offset address of SYNCCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR1 | (0x04C4U) | Offset address of SYNCCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR2 | (0x04C8U) | Offset address of SYNCCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR3 | (0x04CCU) | Offset address of SYNCCR3 register for OCV. |
| IMPDRV\_REG\_OCV\_MEMINITR | (0x00F8U) | Offset address of MEMINITR register for OCV. |
| IMPDRV\_REG\_OCV\_LWM | (0x021CU) | Offset address of LWM register for OCV. |
| IMPDRV\_OCV\_VCR | (0x00000005U) | Hardware version value for OCV. |
| IMPDRV\_OCV\_RESET\_STATUS\_VAL | (0x00F10000U) | RESET\_STATUS value for OCV. |
| IMPDRV\_OCV\_WM\_INIT\_VAL | (0x00F1U) | Working Memory init value for OCV. |
| IMPDRV\_OCV\_8KBSTART\_VAL | (0x0000U) | 8KBSTART value for OCV. |
| IMPDRV\_OCV\_8KBNEXT\_VAL | (0x2000U) | 8KBNEXT value for OCV. |
| IMPDRV\_OCV\_OFST\_START\_VAL | (0xC000U) | OFST\_START value for OCV. |
| IMPDRV\_OCV\_OFST\_END\_VAL | (0xE000U) | OFST\_END value for OCV. |
| IMPDRV\_OCV\_1KB\_VAL | (0x80064100U) | 1KB value for OCV. |
| IMPDRV\_OCV\_MCR0\_INIT\_VAL | (0x01000000U) | MCR0 value for OCV. |
| IMPDRV\_OCV\_CR\_PS | (0x00000001U) | PS register value for OCV. |
| IMPDRV\_OCV\_RSTR\_SWRST | (0x00000001U) | SWRST register value for OCV. |
| IMPDRV\_OCV\_RSTR\_CLR | (0x00000000U) | CLR register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL0 | (0x00040000U) | DCBANKSEL0 register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL1 | (0x05000000U) | DCBANKSEL1 register value for OCV. |
| IMPDRV\_OCV\_SR1\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_PBCOVF | (0x00000008U) | PBCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_SBO0ME | (0x00000010U) | SBO0ME interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USIER | (0x00001000U) | USIER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USINT | (0x00002000U) | USINT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_OCV\_OFFSET\_32B | (32U) | Offset 32byte for OCV. |
| IMPDRV\_DMA\_CORENUM\_VAL | (4U) | Maximum number of cores for DMA. |
| IMPDRV\_DMA\_CORENUM0MAIN | (0U) | DMA Core0 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM0SUB | (1U) | DMA Core0 Sub thread identification value. |
| IMPDRV\_DMA\_CORENUM1MAIN | (2U) | DMA Core1 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM1SUB | (3U) | DMA Core1 Sub thread identification value. |
| IMPDRV\_REG\_DMA\_VCR | (0x0000U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMA. |
| IMPDRV\_REG\_DMA\_SR | (0x0008U) | Offset address of SR register for DMA. |
| IMPDRV\_REG\_DMA\_SCR | (0x000CU) | Offset address of SCR register for DMA. |
| IMPDRV\_REG\_DMA\_SER | (0x0010U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_IMR | (0x0014U) | Offset address of IMR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMA. |
| IMPDRV\_REG\_DMA\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMA. |
| IMPDRV\_REG\_DMA\_SR1 | (0x0030U) | Offset address of SR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMA. |
| IMPDRV\_DMA\_VCR | (0x18071916U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_REG\_DMA\_THREAD\_OFFSET | (0x00001000U) | H/W Register offset for DMA sub-thread. |
| IMPDRV\_DMA\_SR\_TEND | (0x00000001U) | SA\_TEND interrupt enable registration. |
| IMPDRV\_DMA\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMA\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMA\_SR\_WUP | (0x00000080U) | WUP interrupt enable registration. |
| IMPDRV\_DMA\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_MASK | (0x8000U) | Bitmask of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_LOOPTIME\_VAL | (8U) | Read count for register. |
| IMPDRV\_PSC\_CORENUM\_VAL | (2U) | Maximum number of cores for PSC. |
| IMPDRV\_REG\_PSC\_VCR | (0x0000U) | Offset address of VCR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR0 | (0x0004U) | Offset address of CTLR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SR | (0x0008U) | Offset address of SR register for PSC. |
| IMPDRV\_REG\_PSC\_SCR | (0x000CU) | Offset address of SCR register for PSC. |
| IMPDRV\_REG\_PSC\_SER | (0x0010U) | Offset address of SER register for PSC. |
| IMPDRV\_REG\_PSC\_IMR | (0x0014U) | Offset address of IMR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SR1 | (0x003CU) | Offset address of SR1 register for PSC. |
| IMPDRV\_REG\_PSC\_CLSAR | (0x0038U) | Offset address of CLSAR register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR0 | (0x0058U) | Offset address of SYNCCR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR1 | (0x0060U) | Offset address of SYNCCR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR2 | (0x0064U) | Offset address of SYNCCR2 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR3 | (0x0068U) | Offset address of SYNCCR3 register for PSC. |
| IMPDRV\_PSC\_VCR | (0x18082420U) | Hardware version value for PSC. |
| IMPDRV\_PSC\_SCTLR0\_SWRST | (0x80000000U) | SWRST register value for IMP. |
| IMPDRV\_PSC\_SCTLR0\_CLR | (0x00000000U) | CLR register value for IMP. |
| IMPDRV\_PSC\_SCTLR1\_CLE | (0x00000001U) | CLE register value for IMP. |
| IMPDRV\_PSC\_SR\_TEND | (0x00000001U) | TEND interrupt enable registration. |
| IMPDRV\_PSC\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_PSC\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_PSC\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_PSC\_SR\_WUP | (0x00010000U) | WUP interrupt enable registration. |
| IMPDRV\_CNN\_CORENUM\_VAL | (3U) | Maximum number of cores for CNN. |
| IMPDRV\_REG\_CNN\_VCR | (0x0000U) | Offset address of VCR register for CNN. |
| IMPDRV\_CNN\_VCR | (0x00030200U) | Hardware version value for CNN. |
| IMPDRV\_REG\_CNN\_SWRST | (0x0008U) | Offset address of hardware register for SWRST. |
| IMPDRV\_REG\_CNN\_SR | (0x0010U) | Offset address of hardware register for SR. |
| IMPDRV\_REG\_CNN\_SRE | (0x0014U) | Offset address of hardware register for SRE. |
| IMPDRV\_REG\_CNN\_SRC | (0x0018U) | Offset address of hardware register for SRC. |
| IMPDRV\_REG\_CNN\_SRM | (0x001CU) | Offset address of hardware register for SRM. |
| IMPDRV\_REG\_CNN\_SACL | (0x0104U) | Offset address of hardware register for SACL. |
| IMPDRV\_REG\_CNN\_SCLP | (0x0108U) | Offset address of hardware register for SCLP. |
| IMPDRV\_REG\_CNN\_SYNCCR0 | (0x0900U) | Offset address of hardware register for SYNCCR0. |
| IMPDRV\_REG\_CNN\_SYNCCR1 | (0x0904U) | Offset address of hardware register for SYNCCR1. |
| IMPDRV\_REG\_CNN\_SYNCCR2 | (0x0908U) | Offset address of hardware register for S  SYNCCR2. |
| IMPDRV\_REG\_CNN\_SYNCCR3 | (0x090CU) | Offset address of hardware register for SYNCCR3. |
| IMPDRV\_CNN\_CNF\_SWRST | (0x00000001U) | SWRST register value for CNN. |
| IMPDRV\_CNN\_CNF\_CLR | (0x00000000U) | CLR register value for CNN. |
| IMPDRV\_CNN\_SR\_CLBUSY | (0x00004000U) | CLBSY register value for CNN. |
| IMPDRV\_CNN\_SR\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_CNN\_SR\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_CNN\_SR\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_CNN\_SR\_MSCO | (0x00000008U) | MSCO interrupt enable registration. |
| IMPDRV\_CNN\_SR\_WUPCOVF | (0x00000010U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_CNN\_SCLP\_START | (0x00000001U) | CNN\_SCLP\_START register value for CNN. |
| IMPDRV\_SYNCC\_REGNUM | (4U) | Array size for SYNCC register. |
| IMPDRV\_EXEPARAM\_OFF | (1U) | OFF value for Extended parameter |
| IMPDRV\_EXEPARAM\_ON | (2U) | ON value for Extended parameter |

Table 5‑51: Definition Values (Core Control) [R-CarV3M]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_IMP\_CORENUM\_VAL | (4U) | Maximum number of cores for IMP. |
| IMPDRV\_REG\_IMP\_VCR | (0x0004U) | Offset address of VCR register for IMP. |
| IMPDRV\_REG\_IMP\_CNF | (0x0008U) | Offset address of CNF register for IMP. |
| IMPDRV\_REG\_IMP\_IFCFG | (0x0018U) | Offset address of CNFG register for IMP. |
| IMPDRV\_REG\_IMP\_IFCTL | (0x001CU) | Offset address of IFCTL register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for IMP. |
| IMPDRV\_REG\_IMP\_INTSTS | (0x0010U) | Offset address of INTSTS register for IMP. |
| IMPDRV\_REG\_IMP\_INTEN | (0x0014U) | Offset address of INTEN register for IMP. |
| IMPDRV\_REG\_IMP\_PSA | (0x0020U) | Offset address of PSA register for IMP. |
| IMPDRV\_REG\_IMP\_HMPTR | (0x0514U) | Offset address of HMPTR register for IMP. |
| IMPDRV\_REG\_IMP\_HMDATA | (0x0518U) | Offset address of HMDATA register for IMP. |
| IMPDRV\_REG\_IMP\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for IMP. |
| IMPDRV\_REG\_IMP\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for IMP. |
| IMPDRV\_REG\_IMP\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for IMP. |
| IMPDRV\_REG\_IMP\_APSASP | (0x0100U) | Offset address of APSASP register for IMP |
| IMPDRV\_REG\_IMP\_APSBSP | (0x0104U) | Offset address of APSBSP register for IMP |
| IMPDRV\_REG\_IMP\_APDSP | (0x0108U) | Offset address of APSBSP register for IMP |
| IMPDRV\_REG\_IMP\_APLNG | (0x010CU) | Offset address of APLNG register for IMP |
| IMPDRV\_REG\_IMP\_APDLY | (0x0110U) | Offset address of APDLY register for IMP |
| IMPDRV\_REG\_IMP\_APMAG | (0x0114U) | Offset address of APMAG register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_SA | (0x0118U) | Offset address of APSIZE\_SA register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_SB | (0x011CU) | Offset address of APSIZE\_SB register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_DST | (0x0120U) | Offset address of APSIZE\_DST register for IMP |
| IMPDRV\_REG\_IMP\_APCMD | (0x0128U) | Offset address of APCMD register for IMP |
| IMPDRV\_REG\_IMP\_APCLPX | (0x012CU) | Offset address of APCLPX register for IMP |
| IMPDRV\_REG\_IMP\_APCFG | (0x013CU) | Offset address of APCFG register for IMP |
| IMPDRV\_REG\_IMP\_IPFUN | (0x0800U) | Offset address of IPFUN register for IMP |
| IMPDRV\_REG\_IMP\_IPFUN2 | (0x0804U) | Offset address of IPFUN2 register for IMP |
| IMPDRV\_REG\_IMP\_IPFORM | (0x0808U) | Offset address of IPFORM register for IMP |
| IMPDRV\_REG\_IMP\_CNST | (0x0814U) | Offset address of CNST register for IMP |
| IMPDRV\_REG\_IMP\_BINTHR | (0x0818U) | Offset address of BINTHR register for IMP |
| IMPDRV\_REG\_IMP\_BINTHR2 | (0x081CU) | Offset address of BINTHR2 register for IMP |
| IMPDRV\_REG\_IMP\_KNLMSK | (0x0820U) | Offset address of KNLMSK register for IMP |
| IMPDRV\_REG\_IMP\_KNLMSK2 | (0x0824U) | Offset address of KNLMSK2 register for IMP |
| IMPDRV\_REG\_IMP\_LMCTL | (0x0828U) | Offset address of LMCTL register for IMP |
| IMPDRV\_REG\_IMP\_LABCNT | (0x082CU) | Offset address of LABCNT register for IMP |
| IMPDRV\_REG\_IMP\_COEFF02 | (0x0830U) | Offset address of COEFF02 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF35 | (0x0834U) | Offset address of COEFF35 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF68 | (0x0838U) | Offset address of COEFF68 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF911 | (0x083CU) | Offset address of COEFF911 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1214 | (0x0840U) | Offset address of COEFF1214 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF15 | (0x0844U) | Offset address of COEFF15 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1820 | (0x0848U) | Offset address of COEFF1820 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF2123 | (0x084CU) | Offset address of COEFF2123 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF24 | (0x0850U) | Offset address of COEFF24 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF02H | (0x0854U) | Offset address of COEFF02H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF35H | (0x0858U) | Offset address of COEFF35H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF68H | (0x085CU) | Offset address of COEFF68H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF911H | (0x0860U) | Offset address of COEFF911H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1214H | (0x0864U) | Offset address of COEFF1214H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1517H | (0x0868U) | Offset address of COEFF1517H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1820H | (0x086CU) | Offset address of COEFF1820H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF2123H | (0x0870U) | Offset address of COEFF2123H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF24H | (0x0874U) | Offset address of COEFF24H register for IMP |
| IMPDRV\_REG\_IMP\_PFFTASEL1 | (0x0A80U) | Offset address of PFFTASEL1 register for IMP |
| IMPDRV\_REG\_IMP\_PFFTAMSK | (0x0A88U) | Offset address of PFFTAMSK register for IMP |
| IMPDRV\_REG\_IMP\_PFFTBMSK | (0x0A94U) | Offset address of PFFTBMSK register for IMP |
| IMPDRV\_IMP\_VCR | (0x00600000U) | Hardware version value for IMP. |
| IMPDRV\_IMP\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for IMP. |
| IMPDRV\_IMP\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_EXE | (0x00000001U) | EXE register value for IMP. |
| IMPDRV\_IMP\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for IMP. |
| IMPDRV\_IMP\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for IMP. |
| IMPDRV\_IMP\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for IMP. |
| IMPDRV\_IMP\_MEM\_FUNC\_SA\_VAL | (0x00FFU) | SA\_VAL register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_CLR | IMPDRV\_IMP\_INTSTS\_TRAP|IMPDRV\_IMP\_INTSTS\_IER|IMPDRV\_IMP\_INTSTS\_INT|IMPDRV\_IMP\_INTSTS\_HPINT|IMPDRV\_IMP\_INTSTS\_APIPINT) | CLR register value for IMP. |
| IMPDRV\_IMP\_APCMD\_EX | (0x80000000U) | EX bit in APCMD Register |
| IMPDRV\_IMP\_APCMD\_ACSCNT\_SA | (0x00080000U) | ACSCNT\_SA bit in APCMD Register |
| IMPDRV\_IMP\_APCMD\_ACSCNT\_DS | (0x00010000U) | ACSCNT\_DS bit in APCMD Register |
| IMPDRV\_IMP\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check |
| IMPDRV\_IMP\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check |
| IMPDRV\_IMP\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check |
| IMPDRV\_IMP\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check |
| IMPDRV\_IMP\_REG\_MAX\_ADRS\_CNFCHK | (0x0001FFFFU) | End of offset address for ConfRegCheck of IMP |
| IMPDRV\_CNN\_CORENUM\_VAL | (1U) | Maximum number of cores for CNN. |
| IMPDRV\_REG\_CNN\_VCR | (0x0000U) | Offset address of VCR register for CNN. |
| IMPDRV\_CNN\_VCR | (0x00030101U) | Hardware version value for CNN. |
| IMPDRV\_REG\_CNN\_SWRST | (0x0008U) | Offset address of hardware register for SWRST. |
| IMPDRV\_REG\_CNN\_SR | (0x0010U) | Offset address of hardware register for SR. |
| IMPDRV\_REG\_CNN\_SRE | (0x0014U) | Offset address of hardware register for SRE. |
| IMPDRV\_REG\_CNN\_SRC | (0x0018U) | Offset address of hardware register for SRC. |
| IMPDRV\_REG\_CNN\_SRM | (0x001CU) | Offset address of hardware register for SRM. |
| IMPDRV\_REG\_CNN\_CLSAR | (0x0080U) | Offset address of hardware register for CLSAR. |
| IMPDRV\_REG\_CNN\_SCLP | (0x0084U) | Offset address of hardware register for SCLP. |
| IMPDRV\_REG\_CNN\_SYNCCR0 | (0x0900U) | Offset address of hardware register for SYNCCR0. |
| IMPDRV\_REG\_CNN\_SYNCCR1 | (0x0904U) | Offset address of hardware register for SYNCCR1. |
| IMPDRV\_REG\_CNN\_SYNCCR2 | (0x0908U) | Offset address of hardware register for YNCCR2. |
| IMPDRV\_REG\_CNN\_SYNCCR3 | (0x090CU) | Offset address of hardware register for MG2ISYNCCR3NTSEL |
| IMPDRV\_CNN\_CNF\_SWRST | (0x00000001U) | SWRST register value for CNN. |
| IMPDRV\_CNN\_CNF\_CLR | (0x00000000U) | CLR register value for CNN. |
| IMPDRV\_CNN\_SR\_CLBUSY | (0x00002000U) | CLBSY register value for CNN. |
| IMPDRV\_CNN\_SR\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_CNN\_SR\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_CNN\_SR\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_CNN\_SR\_MSCO | (0x00000008U) | MSCO interrupt enable registration. |
| IMPDRV\_CNN\_SCLP\_START | (0x00000001U) | CNN\_SCLP\_START register value for CNN. |
| IMPDRV\_CNN\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of CNN. |
| IMPDRV\_DMA\_CORENUM\_VAL | (2U) | Maximum number of cores for DMA. |
| IMPDRV\_DMA\_CORENUM0MAIN | (0U) | DMA Core 0 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM0SUB | (1U) | DMA Core 0 Sub thread identification value. |
| IMPDRV\_REG\_DMA\_VCR | (0x0000U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMA. |
| IMPDRV\_REG\_DMA\_SR | (0x0008U) | Offset address of SR register for DMA. |
| IMPDRV\_REG\_DMA\_SCR | (0x000CU) | Offset address of SCR register for DMA. |
| IMPDRV\_REG\_DMA\_SER | (0x0010U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_IMR | (0x0014U) | Offset address of IMR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMA. |
| IMPDRV\_REG\_DMA\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMA. |
| IMPDRV\_REG\_DMA\_SR1 | (0x0030U) | Offset address of SR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMA. |
| IMPDRV\_DMA\_VCR | (0x17021420U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_REG\_DMA\_THREAD\_OFFSET | (0x00000800U) | H/W Register offset for DMA sub-thread. |
| IMPDRV\_DMA\_SR\_TEND | (0x00000001U) | /\*\*<SA\_TEND interrupt enable registration. |
| IMPDRV\_DMA\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMA\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMA\_SER\_FIX | (0x0000F380U) | Status Enable Registers that are always bit 1. |
| IMPDRV\_DMA\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_MASK | (0x8000U) | Bit mask of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_LOOPTIME\_VAL | (8U) | read count for register. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE1 | (3U) | Mult-Bank memory init register info1. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE2 | (7U) | Mult-Bank memory init register info2. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE3 | (5U) | Mult-Bank memory init register info3. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_ODD | (3U) | Odd mult-Bank memory init register write number of times. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_EVEN | (4U) | Even mult-Bank memory init register write number of times. |
| IMPDRV\_REG\_DMA\_IMGSIZER | (0x0094U) | Offset address of IMGSIZER register for DMA. |
| IMPDRV\_DMA\_IMGSIZER\_VAL | (0x00800001U) | IMGSIZER register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR0 | (0x00A0U) | Offset address of FCR0 register for DMA. |
| IMPDRV\_DMA\_FCR0\_VAL | (0x000000CCU) | FCR0 register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR1 | (0x00ACU) | Offset address of FCR1 register for DMA. |
| IMPDRV\_DMA\_FCR1\_VAL | (0x90000000U) | FCR1 register value for DMA. |
| IMPDRV\_REG\_DMA\_S0SAR | (0x0040U) | Offset address of S0SAR register for DMA. |
| IMPDRV\_DMA\_S0SAR\_ODD\_BASE\_ADDR | (0xED020000U) | S0SAR register value for DMA odd bank. |
| IMPDRV\_DMA\_S0SAR\_EVEN\_BASE\_ADDR | (0xED000000U) | S0SAR register value for DMA even bank. |
| IMPDRV\_DMA\_S0SAR\_OFFSET | (0x00040000U) | S0SAR register offset for DMA even bank. |
| IMPDRV\_REG\_DMA\_S0STR | (0x0044U) | Offset address of S0STR register for DMA. |
| IMPDRV\_DMA\_S0STR\_VAL | (0x00000020U) | S0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0DATAR | (0x004CU) | Offset address of S0DATAR register for DMA. |
| IMPDRV\_DMA\_S0DATAR\_VAL | (0x00000000U) | S0DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0CR | (0x0050U) | Offset address of S0CR register for DMA. |
| IMPDRV\_DMA\_S0CR\_VAL | (0x80000003U) | S0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1SAR | (0x0060U) | Offset address of S1SAR register for DMA. |
| IMPDRV\_DMA\_S1SAR\_VAL | (0x00000000U) | S1SAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1STR | (0x0064U) | Offset address of S1STR register for DMA. |
| IMPDRV\_DMA\_S1STR\_VAL | (0x00000020U) | S1STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1DATAR | (0x006CU) | Offset address of S1DATAR register for DMA. |
| IMPDRV\_DMA\_S1DATAR\_VAL | (0x00000000U) | S1DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1CR | (0x0070U) | Offset address of S1CR register for DMA. |
| IMPDRV\_DMA\_S1CR\_VAL | (0x00000003U) | S1CR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0SAR | (0x0080U) | Offset address of D0SAR register for DMA. |
| IMPDRV\_REG\_DMA\_D0STR | (0x0084U) | Offset address of D0STR register for DMA. |
| IMPDRV\_DMA\_D0STR\_VAL | (0x00000020U) | D0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0CR | (0x0090U) | Offset address of D0CR register for DMA. |
| IMPDRV\_DMA\_D0CR\_VAL | (0x00000003U) | D0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_DSWPR | (0x0038U) | Offset address of DSWPR register for DMA. |
| IMPDRV\_DMA\_DSWPR\_VAL | (0x00000000U) | DSWPR register value for DMA. |
| IMPDRV\_REG\_DMA\_TSCR | (0x0020U) | Offset address of TSCR register for DMA. |
| IMPDRV\_DMA\_TSCR\_VAL | (0x00330003U) | TSCR register value for DMA. |
| IMPDRV\_DMA\_SCTLR0\_MB\_VAL | (0x00000001U) | SCTLR0 multi-bank register value for DMA. |
| IMPDRV\_DMA\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check. |
| IMPDRV\_DMA\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_DMA\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check. |
| IMPDRV\_DMA\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check. |
| IMPDRV\_DMA\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of DMA. |
| IMPDRV\_OCV\_CORENUM\_VAL | (2U) | Maximum number of cores for OCV. |
| IMPDRV\_OCV\_OFFSET\_32B | (32U) | Offset 32byte for OCV. |
| IMPDRV\_REG\_OCV\_VCR0 | (0x0000U) | Offset address of VCR register for OCV. |
| IMPDRV\_REG\_OCV\_RSTR | (0x0008U) | Offset address of RSTR register for OCV. |
| IMPDRV\_REG\_OCV\_CR | (0x000CU) | Offset address of CR register for OCV. |
| IMPDRV\_REG\_OCV\_SR0 | (0x0010U) | Offset address of SR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR1 | (0x0014U) | Offset address of SR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR1 | (0x0018U) | Offset address of SCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR1 | (0x001CU) | Offset address of ICR1 register for OCV. |
| IMPDRV\_REG\_OCV\_IMR1 | (0x0020U) | Offset address of IMR1 register for OCV. |
| IMPDRV\_REG\_OCV\_MCR0 | (0x0040U) | Offset address of MCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_DLSAR | (0x0180U) | Offset address of DLSAR register for OCV |
| IMPDRV\_REG\_OCV\_SYNCCR0 | (0x04C0U) | Offset address of SYNCCR0 register for O |
| IMPDRV\_REG\_OCV\_SYNCCR1 | (0x04C4U) | Offset address of SYNCCR1 register for O |
| IMPDRV\_REG\_OCV\_SYNCCR2 | (0x04C8U) | Offset address of SYNCCR2 register for O |
| IMPDRV\_REG\_OCV\_SYNCCR3 | (0x04CCU) | Offset address of SYNCCR3 register for O |
| IMPDRV\_REG\_OCV\_MEMINITR | (0x00F8U) | Offset address of MEMINITR register for |
| IMPDRV\_REG\_OCV\_LWM | (0x021CU) | Offset address of LWM register for OCV. |
| IMPDRV\_OCV\_VCR | (0x00000004U) | Hardware version value for OCV. |
| IMPDRV\_OCV\_RESET\_STATUS\_VAL | (0x10000U) | RESET\_STATUS value for OCV. |
| IMPDRV\_OCV\_0X001\_VAL | (0x0001U) | 0X001 value for OCV. |
| IMPDRV\_OCV\_8KBSTART\_VAL | (0x0000U) | 8KBSTART value for OCV. |
| IMPDRV\_OCV\_8KBNEXT\_VAL | (0x2000U) | 8KBNEXT value for OCV. |
| IMPDRV\_OCV\_OFST\_START\_VAL | (0xC000U) | OFST\_START value for OCV. |
| IMPDRV\_OCV\_OFST\_END\_VAL | (0xE000U) | OFST\_END value for OCV. |
| IMPDRV\_OCV\_1KB\_VAL | (0x80064100U) | 1KB value for OCV. |
| IMPDRV\_OCV\_MCR0\_INIT\_VAL | (0x01000000U) | MCR0 value for OCV. |
| IMPDRV\_OCV\_CR\_PS | (0x00000001U) | PS register value for OCV. |
| IMPDRV\_OCV\_RSTR\_SWRST | (0x00000001U) | SWRST register value for OCV. |
| IMPDRV\_OCV\_RSTR\_CLR | (0x00000000U) | CLR register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL0 | (0x00040000U) | DCBANKSEL0register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL1 | (0x05000000U) | DCBANKSEL1 register value for OCV. |
| IMPDRV\_OCV\_SR1\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_PBCOVF | (0x00000008U) | PBCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_SBO0ME | (0x00000010U) | SBO0ME interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USIER | (0x00001000U) | USIER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USINT | (0x00002000U) | USINT interrupt enable registration. |
| IMPDRV\_PSC\_CORENUM\_VAL | (1U) | Maximum number of cores for PSC. |
| IMPDRV\_REG\_OCV\_SR2 | (0x0024U) | Offset address of SR2 register for OCV |
| IMPDRV\_REG\_OCV\_SCR2 | (0x0028U) | Offset address of SCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR2 | (0x002CU) | Offset address of ICR2 register for OCV |
| IMPDRV\_OCV\_SR1\_UDIPSBRK | (0x10000000U) | UDIPSBRK interrupt enable registration |
| IMPDRV\_OCV\_SR1\_UDIVSBRK | (0x20000000U) | UDIVSBRK interrupt enable registration. |
| IMPDRV\_OCV\_SR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SR2 register. |
| IMPDRV\_OCV\_SCR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SCR2 register |
| IMPDRV\_OCV\_ICR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in ICR2 register |
| IMPDRV\_OCV\_REG\_MAX\_ADRS\_CNFCHK | 0x0000FFFFU | End of offset address for ConfRegCheck of OCV. |
| IMPDRV\_REG\_PSC\_VCR | (0x0000U) | Offset address of VCR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR0 | (0x0004U) | Offset address of CTLR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SR | (0x0008U) | Offset address of SR register for PSC. |
| IMPDRV\_REG\_PSC\_SCR | (0x000CU) | Offset address of SCR register for PSC. |
| IMPDRV\_REG\_PSC\_SER | (0x0010U) | Offset address of SER register for PSC. |
| IMPDRV\_REG\_PSC\_IMR | (0x0014U) | Offset address of IMR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SR1 | (0x003CU) | Offset address of SR1 register for PSC. |
| IMPDRV\_REG\_PSC\_CLSAR | (0x0038U) | Offset address of CLSAR register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR0 | (0x0058U) | Offset address of SYNCCR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR1 | (0x0060U) | Offset address of SYNCCR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR2 | (0x0064U) | Offset address of SYNCCR2 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR3 | (0x0068U) | Offset address of SYNCCR3 register for PSC. |
| IMPDRV\_PSC\_VCR | (0x17020121U) | Hardware version value for PSC. |
| IMPDRV\_PSC\_SCTLR0\_SWRST | (0x80000000U) | SWRST register value for IMP. |
| IMPDRV\_PSC\_SCTLR0\_CLR | (0x00000000U) | CLR register value for IMP. |
| IMPDRV\_PSC\_SCTLR1\_CLE | (0x00000001U) | CLE register value for IMP. |
| IMPDRV\_PSC\_SR\_TEND | (0x00000001U) | TEND interrupt enable registration. |
| IMPDRV\_PSC\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_PSC\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_PSC\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_PSC\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of PSC. |
| IMPDRV\_EXEPARAM\_INVALID | (0U) | Invalid value for Extended parameter |

Table 5‑52: Definition Values (Core Control) [R-CarV3H]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_IMP\_CORENUM\_VAL | (5U) | Maximum number of cores for IMP. |
| IMPDRV\_REG\_IMP\_VCR | (0x0004U) | Offset address of VCR register for IMP. |
| IMPDRV\_REG\_IMP\_CNF | (0x0008U) | Offset address of CNF register for IMP. |
| IMPDRV\_REG\_IMP\_IFCFG | (0x0018U) | Offset address of CNFG register for IMP. |
| IMPDRV\_REG\_IMP\_IFCTL | (0x001CU) | Offset address of IFCTL register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for IMP. |
| IMPDRV\_REG\_IMP\_INTSTS | (0x0010U) | Offset address of INTSTS register for IMP. |
| IMPDRV\_REG\_IMP\_INTEN | (0x0014U) | Offset address of INTEN register for IMP. |
| IMPDRV\_REG\_IMP\_PSA | (0x0020U) | Offset address of PSA register for IMP. |
| IMPDRV\_REG\_IMP\_HMPTR | (0x0514U) | Offset address of HMPTR register for IMP. |
| IMPDRV\_REG\_IMP\_HMDATA | (0x0518U) | Offset address of HMDATA register for IMP. |
| IMPDRV\_REG\_IMP\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for IMP. |
| IMPDRV\_REG\_IMP\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for IMP. |
| IMPDRV\_REG\_IMP\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for IMP. |
| IMPDRV\_REG\_IMP\_APSASP | (0x0100U) | Offset address of APSASP register for IMP |
| IMPDRV\_REG\_IMP\_APSBSP | (0x0104U) | Offset address of APSBSP register for IMP |
| IMPDRV\_REG\_IMP\_APDSP | (0x0108U) | Offset address of APSBSP register for IMP |
| IMPDRV\_REG\_IMP\_APLNG | (0x010CU) | Offset address of APLNG register for IMP |
| IMPDRV\_REG\_IMP\_APDLY | (0x0110U) | Offset address of APDLY register for IMP |
| IMPDRV\_REG\_IMP\_APMAG | (0x0114U) | Offset address of APMAG register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_SA | (0x0118U) | Offset address of APSIZE\_SA register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_SB | (0x011CU) | Offset address of APSIZE\_SB register for IMP |
| IMPDRV\_REG\_IMP\_APSIZE\_DST | (0x0120U) | Offset address of APSIZE\_DST register for IMP |
| IMPDRV\_REG\_IMP\_APCMD | (0x0128U) | Offset address of APCMD register for IMP |
| IMPDRV\_REG\_IMP\_APCLPX | (0x012CU) | Offset address of APCLPX register for IMP |
| IMPDRV\_REG\_IMP\_APCFG | (0x013CU) | Offset address of APCFG register for IMP |
| IMPDRV\_REG\_IMP\_IPFUN | (0x0800U) | Offset address of IPFUN register for IMP |
| IMPDRV\_REG\_IMP\_IPFUN2 | (0x0804U) | Offset address of IPFUN2 register for IMP |
| IMPDRV\_REG\_IMP\_IPFORM | (0x0808U) | Offset address of IPFORM register for IMP |
| IMPDRV\_REG\_IMP\_CNST | (0x0814U) | Offset address of CNST register for IMP |
| IMPDRV\_REG\_IMP\_BINTHR | (0x0818U) | Offset address of BINTHR register for IMP |
| IMPDRV\_REG\_IMP\_BINTHR2 | (0x081CU) | Offset address of BINTHR2 register for IMP |
| IMPDRV\_REG\_IMP\_KNLMSK | (0x0820U) | Offset address of KNLMSK register for IMP |
| IMPDRV\_REG\_IMP\_KNLMSK2 | (0x0824U) | Offset address of KNLMSK2 register for IMP |
| IMPDRV\_REG\_IMP\_LMCTL | (0x0828U) | Offset address of LMCTL register for IMP |
| IMPDRV\_REG\_IMP\_LABCNT | (0x082CU) | Offset address of LABCNT register for IMP |
| IMPDRV\_REG\_IMP\_COEFF02 | (0x0830U) | Offset address of COEFF02 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF35 | (0x0834U) | Offset address of COEFF35 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF68 | (0x0838U) | Offset address of COEFF68 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF911 | (0x083CU) | Offset address of COEFF911 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1214 | (0x0840U) | Offset address of COEFF1214 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF15 | (0x0844U) | Offset address of COEFF15 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1820 | (0x0848U) | Offset address of COEFF1820 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF2123 | (0x084CU) | Offset address of COEFF2123 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF24 | (0x0850U) | Offset address of COEFF24 register for IMP |
| IMPDRV\_REG\_IMP\_COEFF02H | (0x0854U) | Offset address of COEFF02H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF35H | (0x0858U) | Offset address of COEFF35H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF68H | (0x085CU) | Offset address of COEFF68H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF911H | (0x0860U) | Offset address of COEFF911H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1214H | (0x0864U) | Offset address of COEFF1214H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1517H | (0x0868U) | Offset address of COEFF1517H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF1820H | (0x086CU) | Offset address of COEFF1820H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF2123H | (0x0870U) | Offset address of COEFF2123H register for IMP |
| IMPDRV\_REG\_IMP\_COEFF24H | (0x0874U) | Offset address of COEFF24H register for IMP |
| IMPDRV\_REG\_IMP\_PFFTASEL1 | (0x0A80U) | Offset address of PFFTASEL1 register for IMP |
| IMPDRV\_REG\_IMP\_PFFTAMSK | (0x0A88U) | Offset address of PFFTAMSK register for IMP |
| IMPDRV\_REG\_IMP\_PFFTBMSK | (0x0A94U) | Offset address of PFFTBMSK register for IMP |
| IMPDRV\_IMP\_VCR | (0x00600000U) | Hardware version value for IMP. |
| IMPDRV\_IMP\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for IMP. |
| IMPDRV\_IMP\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_EXE | (0x00000001U) | EXE register value for IMP. |
| IMPDRV\_IMP\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for IMP. |
| IMPDRV\_IMP\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for IMP |
| IMPDRV\_IMP\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for IMP. |
| IMPDRV\_IMP\_MEM\_FUNC\_SA\_VAL | (0x00FFU) | SA\_VAL register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_CLR | (IMPDRV\_IMP\_INTSTS\_TRAP | IMPDRV\_IMP\_INTSTS\_IER | IMPDRV\_IMP\_INTSTS\_INT | IMPDRV\_IMP\_INTSTS\_WUP | IMPDRV\_IMP\_INTSTS\_HPINT | IMPDRV\_IMP\_INTSTS\_APIPINT) | CLR register value for IMP. |
| IMPDRV\_IMP\_APCMD\_EX | (0x80000000U) | EX bit in APCMD Register |
| IMPDRV\_IMP\_APCMD\_ACSCNT\_SA | (0x00080000U) | ACSCNT\_SA bit in APCMD Register |
| IMPDRV\_IMP\_APCMD\_ACSCNT\_DS | (0x00010000U) | ACSCNT\_DS bit in APCMD Register |
| IMPDRV\_IMP\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check |
| IMPDRV\_IMP\_MEM\_ALIGN\_BUSCHK (127U) | Memory alignment for Bus interface check |  |
| IMPDRV\_IMP\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check |
| IMPDRV\_IMP\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check |
| IMPDRV\_IMP\_REG\_MAX\_ADRS\_CNFCHK | (0x0001FFFFU) | End of offset address for ConfRegCheck of IMP |
| IMPDRV\_CNN\_CORENUM\_VAL | (1U) | Maximum number of cores for CNN. |
| IMPDRV\_REG\_CNN\_VCR | (0x0000U) | Offset address of VCR register for CNN. |
| IMPDRV\_CNN\_VCR | (0x00030000U) | Hardware version value for CNN. |
| IMPDRV\_REG\_CNN\_SWRST | (0x0008U) | Offset address of hardware register for SWRST. |
| IMPDRV\_REG\_CNN\_SR | (0x0010U) | Offset address of hardware register for SR. |
| IMPDRV\_REG\_CNN\_SRE | (0x0014U) | Offset address of hardware register for SRE. |
| IMPDRV\_REG\_CNN\_SRC | (0x0018U) | Offset address of hardware register for SRC. |
| IMPDRV\_REG\_CNN\_SRM | (0x001CU) | Offset address of hardware register for SRM. |
| IMPDRV\_REG\_CNN\_CLSAR | (0x0080U) | Offset address of hardware register for CLSAR. |
| IMPDRV\_REG\_CNN\_SCLP | (0x0084U) | Offset address of hardware register for SCLP. |
| IMPDRV\_REG\_CNN\_SYNCCR0 | (0x0900U) | Offset address of hardware register for SYNCCR0. |
| IMPDRV\_REG\_CNN\_SYNCCR1 | (0x0904U) | Offset address of hardware register for SYNCCR1. |
| IMPDRV\_REG\_CNN\_SYNCCR2 | (0x0908U) | Offset address of hardware register for YNCCR2. |
| IMPDRV\_REG\_CNN\_SYNCCR3 | (0x090CU) | Offset address of hardware register for MG2ISYNCCR3NTSEL. |
| IMPDRV\_CNN\_CNF\_SWRST | (0x00000001U) | SWRST register value for CNN. |
| IMPDRV\_CNN\_CNF\_CLR | (0x00000000U) | CLR register value for CNN. |
| IMPDRV\_CNN\_SR\_CLBUSY | (0x00002000U) | CLBSY register value for CNN. |
| IMPDRV\_CNN\_SR\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_CNN\_SR\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_CNN\_SR\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_CNN\_SR\_MSCO | (0x00000008U) | MSCO interrupt enable registration. |
| IMPDRV\_CNN\_SR\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_CNN\_SCLP\_START | (0x00000001U) | CNN\_SCLP\_START register value for CNN. |
| IMPDRV\_CNN\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of CNN. |
| IMPDRV\_DMA\_CORENUM\_VAL | (4U) | Maximum number of cores for DMA. |
| IMPDRV\_DMA\_CORENUM0MAIN | (0U) | DMA Core 0 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM0SUB | (1U) | DMA Core 0 Sub thread identification value. |
| IMPDRV\_DMA\_CORENUM1MAIN | (2U) | Offset address of VCR register for DMA. |
| IMPDRV\_DMA\_CORENUM1SUB | (3U) | DMA Core 1 Sub thread identification value. |
| IMPDRV\_REG\_DMA\_VCR | (0x0000U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMA. |
| IMPDRV\_REG\_DMA\_SR | (0x0008U) | Offset address of SR register for DMA. |
| IMPDRV\_REG\_DMA\_SCR | (0x000CU) | Offset address of SCR register for DMA. |
| IMPDRV\_REG\_DMA\_SER | (0x0010U) | Offset address of SER register for DMA. |
| IMPDRV\_REG\_DMA\_IMR | (0x0014U) | Offset address of IMR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMA. |
| IMPDRV\_REG\_DMA\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMA. |
| IMPDRV\_REG\_DMA\_SR1 | (0x0030U) | Offset address of SR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMA. |
| IMPDRV\_DMA\_VCR | (0x17021420U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_REG\_DMA\_THREAD\_OFFSET | (0x00000800U) | H/W Register offset for DMA sub-thread. |
| IMPDRV\_DMA\_SR\_TEND | (0x00000001U) | SA\_TEND interrupt enable registration. |
| IMPDRV\_DMA\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMA\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMA\_SR\_WUP | (0x00000080U) | WUP interrupt enable registration. |
| IMPDRV\_DMA\_SER\_FIX | (0x0000F300U) | Status Enable Registers that are always bit 1. |
| IMPDRV\_DMA\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_MASK | (0x8000U) | Bit mask of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_LOOPTIME\_VAL | (8U) | read count for register. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE1 | (3U) | Mult-Bank memory init register info1. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE2 | (7U) | Mult-Bank memory init register info2. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE3 | (5U) | Mult-Bank memory init register info3. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_ODD | (8U) | Odd mult-Bank memory init register write number of times. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_EVEN | (8U) | Even mult-Bank memory init register write number of times. |
| IMPDRV\_REG\_DMA\_IMGSIZER | (0x0094U) | Offset address of IMGSIZER register for DMA. |
| IMPDRV\_DMA\_IMGSIZER\_VAL | (0x00800001U) | IMGSIZER register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR0 | (0x00A0U) | Offset address of FCR0 register for DMA. |
| IMPDRV\_DMA\_FCR0\_VAL | (0x000000CCU) | FCR0 register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR1 | (0x00ACU) | Offset address of FCR1 register for DMA. |
| IMPDRV\_DMA\_FCR1\_VAL | (0x90000000U) | FCR1 register value for DMA. |
| IMPDRV\_REG\_DMA\_S0SAR | (0x0040U) | Offset address of S0SAR register for DMA. |
| IMPDRV\_DMA\_S0SAR\_ODD\_BASE\_ADDR | (0xED020000U) | S0SAR register value for DMA odd bank. |
| IMPDRV\_DMA\_S0SAR\_EVEN\_BASE\_ADDR | (0xED000000U) | S0SAR register value for DMA even bank. |
| IMPDRV\_DMA\_S0SAR\_OFFSET | (0x00040000U) | S0SAR register offset for DMA even bank. |
| IMPDRV\_REG\_DMA\_S0STR | (0x0044U) | Offset address of S0STR register for DMA. |
| IMPDRV\_DMA\_S0STR\_VAL | (0x00000020U) | S0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0DATAR | (0x004CU) | Offset address of S0DATAR register for DMA. |
| IMPDRV\_DMA\_S0DATAR\_VAL | (0x00000000U) | S0DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0CR | (0x0050U) | Offset address of S0CR register for DMA. |
| IMPDRV\_DMA\_S0CR\_VAL | (0x80000003U) | S0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1SAR | (0x0060U) | Offset address of S1SAR register for DMA. |
| IMPDRV\_DMA\_S1SAR\_VAL | (0x00000000U) | S1SAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1STR | (0x0064U) | Offset address of S1STR register for DMA. |
| IMPDRV\_DMA\_S1STR\_VAL | (0x00000020U) | S1STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1DATAR | (0x006CU) | Offset address of S1DATAR register for DMA. |
| IMPDRV\_DMA\_S1DATAR\_VAL | (0x00000000U) | S1DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1CR | (0x0070U) | Offset address of S1CR register for DMA. |
| IMPDRV\_DMA\_S1CR\_VAL | (0x00000003U) | S1CR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0SAR | (0x0080U) | Offset address of D0SAR register for DMA. |
| IMPDRV\_REG\_DMA\_D0STR | (0x0084U) | Offset address of D0STR register for DMA. |
| IMPDRV\_DMA\_D0STR\_VAL | (0x00000020U) | D0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0CR | (0x0090U) | Offset address of D0CR register for DMA. |
| IMPDRV\_DMA\_D0CR\_VAL | (0x00000003U) | D0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_DSWPR | (0x0038U) | Offset address of DSWPR register for DMA. |
| IMPDRV\_DMA\_DSWPR\_VAL | (0x00000000U) | DSWPR register value for DMA. |
| IMPDRV\_REG\_DMA\_TSCR | (0x0020U) | Offset address of TSCR register for DMA. |
| IMPDRV\_DMA\_TSCR\_VAL | (0x00330003U) | TSCR register value for DMA. |
| IMPDRV\_DMA\_SCTLR0\_MB\_VAL | (0x00000001U) | SCTLR0 multi-bank register value for DMA. |
| IMPDRV\_DMA\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check. |
| IMPDRV\_DMA\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_DMA\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check. |
| IMPDRV\_DMA\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check. |
| IMPDRV\_DMA\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of DMA. |
| IMPDRV\_OCV\_CORENUM\_VAL | (5U) | Maximum number of cores for OCV. |
| IMPDRV\_OCV\_OFFSET\_32B | (32U) | Offset 32byte for OCV. |
| IMPDRV\_REG\_OCV\_VCR0 | (0x0000U) | Offset address of VCR register for OCV. |
| IMPDRV\_REG\_OCV\_RSTR | (0x0008U) | Offset address of RSTR register for OCV. |
| IMPDRV\_REG\_OCV\_CR | (0x000CU) | Offset address of CR register for OCV. |
| IMPDRV\_REG\_OCV\_SR0 | (0x0010U) | Offset address of SR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR1 | (0x0014U) | Offset address of SR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR1 | (0x0018U) | Offset address of SCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR1 | (0x001CU) | Offset address of ICR1 register for OCV. |
| IMPDRV\_REG\_OCV\_IMR1 | (0x0020U) | Offset address of IMR1 register for OCV. |
| IMPDRV\_REG\_OCV\_MCR0 | (0x0040U) | Offset address of MCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_DLSAR | (0x0180U) | Offset address of DLSAR register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR0 | (0x04C0U) | Offset address of SYNCCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR1 | (0x04C4U) | Offset address of SYNCCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR2 | (0x04C8U) | Offset address of SYNCCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR3 | (0x04CCU) | Offset address of SYNCCR3 register for OCV. |
| IMPDRV\_REG\_OCV\_MEMINITR | (0x00F8U) | Offset address of MEMINITR register for OCV. |
| IMPDRV\_REG\_OCV\_LWM | (0x021CU) | Offset address of LWM register for OCV. |
| IMPDRV\_OCV\_VCR | (0x00000004U) | Hardware version value for OCV. |
| IMPDRV\_OCV\_RESET\_STATUS\_VAL | (0x10000U) | RESET\_STATUS value for OCV. |
| IMPDRV\_OCV\_0X001\_VAL | (0x0001U) | 0X001 value for OCV. |
| IMPDRV\_OCV\_8KBSTART\_VAL | (0x0000U) | 8KBSTART value for OCV. |
| IMPDRV\_OCV\_8KBNEXT\_VAL | (0x2000U) | 8KBNEXT value for OCV. |
| IMPDRV\_OCV\_OFST\_START\_VAL | (0xC000U) | OFST\_START value for OCV. |
| IMPDRV\_OCV\_OFST\_END\_VAL | (0xE000U) | OFST\_END value for OCV. |
| IMPDRV\_OCV\_1KB\_VAL | (0x80064100U) | 1KB value for OCV. |
| IMPDRV\_OCV\_MCR0\_INIT\_VAL | (0x01000000U) | MCR0 value for OCV. |
| IMPDRV\_OCV\_CR\_PS | (0x00000001U) | PS register value for OCV. |
| IMPDRV\_OCV\_RSTR\_SWRST | (0x00000001U) | SWRST register value for OCV. |
| IMPDRV\_OCV\_RSTR\_CLR | (0x00000000U) | CLR register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL0 | (0x00040000U) | DCBANKSEL0register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL1 | (0x05000000U) | DCBANKSEL1 register value for OCV. |
| IMPDRV\_OCV\_SR1\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_PBCOVF | (0x00000008U) | PBCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_SBO0ME | (0x00000010U) | SBO0ME interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USIER | (0x00001000U) | USIER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USINT | (0x00002000U) | USINT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_REG\_OCV\_SR2 | (0x0024U) | Offset address of SR2 register for OCV |
| IMPDRV\_REG\_OCV\_SCR2 | (0x0028U) | Offset address of SCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR2 | (0x002CU) | Offset address of ICR2 register for OCV |
| IMPDRV\_OCV\_REG\_MAX\_ADRS\_CNFCHK | 0x0000FFFFU | End of offset address for ConfRegCheck of OCV. |
| IMPDRV\_OCV\_SR1\_UDIPSBRK | (0x10000000U) | UDIPSBRK interrupt enable registration |
| IMPDRV\_OCV\_SR1\_UDIVSBRK | (0x20000000U) | UDIVSBRK interrupt enable registration. |
| IMPDRV\_OCV\_SR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SR2 register. |
| IMPDRV\_OCV\_SCR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SCR2 register |
| IMPDRV\_OCV\_ICR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in ICR2 register |
| IMPDRV\_PSC\_CORENUM\_VAL | (2U) | Maximum number of cores for PSC. |
| IMPDRV\_REG\_PSC\_VCR | (0x0000U) | Offset address of VCR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR0 | (0x0004U) | Offset address of CTLR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SR | (0x0008U) | Offset address of SR register for PSC. |
| IMPDRV\_REG\_PSC\_SCR | (0x000CU) | Offset address of SCR register for PSC. |
| IMPDRV\_REG\_PSC\_SER | (0x0010U) | Offset address of SER register for PSC. |
| IMPDRV\_REG\_PSC\_IMR | (0x0014U) | Offset address of IMR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SR1 | (0x003CU) | Offset address of SR1 register for PSC. |
| IMPDRV\_REG\_PSC\_CLSAR | (0x0038U) | Offset address of CLSAR register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR0 | (0x0058U) | Offset address of SYNCCR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR1 | (0x0060U) | Offset address of SYNCCR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR2 | (0x0064U) | Offset address of SYNCCR2 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR3 | (0x0068U) | Offset address of SYNCCR3 register for PSC. |
| IMPDRV\_PSC\_VCR | (0x17020121U) | Hardware version value for PSC. |
| IMPDRV\_PSC\_SCTLR0\_SWRST | (0x80000000U) | SWRST register value for IMP. |
| IMPDRV\_PSC\_SCTLR0\_CLR | (0x00000000U) | CLR register value for IMP. |
| IMPDRV\_PSC\_SCTLR1\_CLE | (0x00000001U) | CLE register value for IMP. |
| IMPDRV\_PSC\_SR\_TEND | (0x00000001U) | TEND interrupt enable registration. |
| IMPDRV\_PSC\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_PSC\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_PSC\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_PSC\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of PSC. |
| IMPDRV\_IMPS\_CORENUM\_VAL | (1U) | Maximum number of cores for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_VCR | (0x0004U) | Offset address of VCR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_CNF | (0x0008U) | Offset address of CNF register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IFCFG | (0x0018U) | Offset address of CNFG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IFCTL | (0x001CU) | Offset address of IFCTL register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for Slim-IMP |
| IMPDRV\_REG\_IMPS\_INTSTS | (0x0010U) | Offset address of INTSTS register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_INTEN | (0x0014U) | Offset address of INTEN register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PSA | (0x0020U) | Offset address of PSA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for Slim-IMP. |
| IMPDRV\_IMPS\_VCR | (0x00600000U) | Hardware version value for Slim-IMP. |
| IMPDRV\_IMPS\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCTL\_EXE | (0x00000001U) | EXE register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for Slim-IMP. |
| IMPDRV\_IMPS\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_MEM\_FUNC\_SA\_VAL | (0x0030U) | SA\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_CLR | (IMPDRV\_IMPS\_INTSTS\_TRAP | IMPDRV\_IMPS\_INTSTS\_IER | IMPDRV\_IMPS\_INTSTS\_INT | IMPDRV\_IMPS\_INTSTS\_WUP |IMPDRV\_IMPS\_INTSTS\_HPINT | IMPDRV\_IMPS\_INTSTS\_APIPINT) | CLR register value for Slim-IMP. |
| IMPDRV\_IMPS\_APCMD\_ACSCNT\_DS | (0x00010000U) | ACSCNT\_DS bit in APCMD Register. |
| IMPDRV\_IMPS\_APCMD\_ACSCNT\_SA | (0x00080000U) | ACSCNT\_SA bit in APCMD Register. |
| IMPDRV\_IMPS\_APCMD\_EX | (0x80000000U) | EX bit in APCMD Register. |
| IMPDRV\_IMPS\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_IMPS\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check. |
| IMPDRV\_IMPS\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check. |
| IMPDRV\_IMPS\_REG\_MAX\_ADRS\_CNFCHK | (0x0001FFFFU) | End of offset address for ConfRegCheck of Slim-IMP. |
| IMPDRV\_IMPS\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check. |
| IMPDRV\_REG\_IMPS\_APCFG | (0x013CU) | Offset address of APCFG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APCLPX | (0x012CU) | Offset address of APCLPX register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APCMD | (0x0128U) | Offset address of APCMD register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APDLY | (0x0110U) | Offset address of APDLY register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APDSP | (0x0108U) | Offset address of APSBSP register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APLNG | (0x010CU) | Offset address of APLNG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APMAG | (0x0114U) | Offset address of APMAG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSASP | (0x0100U) | Offset address of APSASP register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSBSP | (0x0104U) | Offset address of APSBSP register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_DST | (0x0120U) | Offset address of APSIZE\_DST register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_SA | (0x0118U) | Offset address of APSIZE\_SA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_SB | (0x011CU) | Offset address of APSIZE\_SB register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_BINTHR | (0x0818U) | Offset address of BINTHR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_BINTHR2 | (0x081CU) | Offset address of BINTHR2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_CNST | (0x0814U) | Offset address of CNST register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF02 | (0x0830U) | Offset address of COEFF02 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF02H | (0x0854U) | Offset address of COEFF02H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1214 | (0x0840U) | Offset address of COEFF1214 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1214H | (0x0864U) | Offset address of COEFF1214H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF15 | (0x0844U) | Offset address of COEFF15 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1517H | (0x0868U) | Offset address of COEFF1517H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1820 | (0x0848U) | Offset address of COEFF1820 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1820H | (0x086CU) | Offset address of COEFF1820H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF2123 | (0x084CU) | Offset address of COEFF2123 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF2123H | (0x0870U) | Offset address of COEFF2123H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF24 | (0x0850U) | Offset address of COEFF24 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF24H | (0x0874U) | Offset address of COEFF24H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF35 | (0x0834U) | Offset address of COEFF35 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF35H | (0x0858U) | Offset address of COEFF35H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF68 | (0x0838U) | Offset address of COEFF68 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF68H | (0x085CU) | Offset address of COEFF68H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF911 | (0x083CU) | Offset address of COEFF911 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF911H | (0x0860U) | Offset address of COEFF911H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFORM | (0x0808U) | Offset address of IPFORM register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFUN | (0x0800U) | Offset address of IPFUN register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFUN2 | (0x0804U) | Offset address of IPFUN2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_KNLMSK | (0x0820U) | Offset address of KNLMSK register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_KNLMSK2 | (0x0824U) | Offset address of KNLMSK2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_LABCNT | (0x082CU) | Offset address of LABCNT register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_LMCTL | (0x0828U) | Offset address of LMCTL register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTAMSK | (0x0A88U) | Offset address of PFFTAMSK register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTASEL1 | (0x0A80U) | Offset address of PFFTASEL1 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTBMSK | (0x0A94U) | Offset address of PFFTBMSK register for Slim-IMP. |

Table 5‑53: Definition Values (Core Control) [R-CarV3H\_2]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_IMP\_CORENUM\_VAL | (5U) | Maximum number of cores for IMP. |
| IMPDRV\_REG\_IMP\_VCR | (0x0004U) | Offset address of VCR register for IMP. |
| IMPDRV\_REG\_IMP\_CNF | (0x0008U) | Offset address of CNF register for IMP. |
| IMPDRV\_REG\_IMP\_IFCFG | (0x0018U) | Offset address of CNFG register for IMP. |
| IMPDRV\_REG\_IMP\_IFCTL | (0x001CU) | Offset address of IFCTL register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for IMP. |
| IMPDRV\_REG\_IMP\_INTSTS | (0x0010U) | Offset address of INTSTS register for IMP. |
| IMPDRV\_REG\_IMP\_INTEN | (0x0014U) | Offset address of INTEN register for IMP. |
| IMPDRV\_REG\_IMP\_PSA | (0x0020U) | Offset address of PSA register for IMP. |
| IMPDRV\_REG\_IMP\_HMPTR | (0x0514U) | Offset address of HMPTR register for IMP. |
| IMPDRV\_REG\_IMP\_HMDATA | (0x0518U) | Offset address of HMDATA register for IMP. |
| IMPDRV\_REG\_IMP\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for IMP. |
| IMPDRV\_REG\_IMP\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for IMP. |
| IMPDRV\_REG\_IMP\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for IMP. |
| IMPDRV\_IMP\_VCR | (0x00610000U) | Hardware version value for IMP. |
| IMPDRV\_IMP\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for IMP. |
| IMPDRV\_IMP\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_EXE | (0x00000001U) | EXE register value for IMP. |
| IMPDRV\_IMP\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for IMP. |
| IMPDRV\_IMP\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for IMP. |
| IMPDRV\_IMP\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for IMP. |
| IMPDRV\_IMP\_MEM\_FUNC\_SA\_VAL | (0x00FFU) | SA\_VAL register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_CLR | (IMPDRV\_IMP\_INTSTS\_TRAP|IMPDRV\_IMP\_INTSTS\_IER|IMPDRV\_IMP\_INTSTS\_INT|IMPDRV\_IMP\_INTSTS\_WUP|IMPDRV\_IMP\_INTSTS\_HPINT|IMPDRV\_IMP\_INTSTS\_APIPINT) | CLR register value for IMP. |
| IMPDRV\_CNN\_CORENUM\_VAL | (1U) | Maximum number of cores for CNN. |
| IMPDRV\_REG\_CNN\_VCR | (0x0000U) | Offset address of VCR register for CNN. |
| IMPDRV\_CNN\_VCR | (0x00030001U) | Hardware version value for CNN. |
| IMPDRV\_REG\_CNN\_SWRST | (0x0008U) | Offset address of hardware register for SWRST. |
| IMPDRV\_REG\_CNN\_SR | (0x0010U) | Offset address of hardware register for SR. |
| IMPDRV\_REG\_CNN\_SRE | (0x0014U) | Offset address of hardware register for SRE. |
| IMPDRV\_REG\_CNN\_SRC | (0x0018U) | Offset address of hardware register for SRC. |
| IMPDRV\_REG\_CNN\_SRM | (0x001CU) | Offset address of hardware register for SRM. |
| IMPDRV\_REG\_CNN\_SACL | (0x0104U) | Offset address of hardware register for SACL |
| IMPDRV\_REG\_CNN\_SCLP | (0x0108U) | Offset address of hardware register for SCLP. |
| IMPDRV\_REG\_CNN\_SYNCCR0 | (0x0900U) | Offset address of hardware register for SYNCCR0. |
| IMPDRV\_REG\_CNN\_SYNCCR1 | (0x0904U) | Offset address of hardware register for SYNCCR1. |
| IMPDRV\_REG\_CNN\_SYNCCR2 | (0x0908U) | Offset address of hardware register for YNCCR2. |
| IMPDRV\_REG\_CNN\_SYNCCR3 | (0x090CU) | Offset address of hardware register for MG2ISYNCCR3NTSEL |
| IMPDRV\_CNN\_CNF\_SWRST | (0x00000001U) | SWRST register value for CNN. |
| IMPDRV\_CNN\_CNF\_CLR | (0x00000000U) | CLR register value for CNN. |
| IMPDRV\_CNN\_SR\_CLBUSY | (0x00004000U) | CLBSY register value for CNN. |
| IMPDRV\_CNN\_SR\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_CNN\_SR\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_CNN\_SR\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_CNN\_SR\_MSCO | (0x00000008U) | MSCO interrupt enable registration. |
| IMPDRV\_CNN\_SR\_WUPCOVF | (0x00000010U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_CNN\_SCLP\_START | (0x00000001U) | CNN\_SCLP\_START register value for CNN. |
| IMPDRV\_CNN\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of CNN. |
| IMPDRV\_DMA\_CORENUM\_VAL | (4U) | Maximum number of cores for DMA. |
| IMPDRV\_DMA\_CORENUM0MAIN | (0U) | DMA Core 0 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM0SUB | (1U) | DMA Core 0 Sub thread identification value. |
| IMPDRV\_DMA\_CORENUM1MAIN | (2U) | DMA Core 1 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM1SUB | (3U) | DMA Core 1 Sub thread identification value. |
| IMPDRV\_REG\_DMA\_VCR | (0x0000U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMA. |
| IMPDRV\_REG\_DMA\_SR | (0x0008U) | Offset address of SR register for DMA. |
| IMPDRV\_REG\_DMA\_SCR | (0x000CU) | Offset address of SCR register for DMA. |
| IMPDRV\_REG\_DMA\_SER | (0x0010U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_IMR | (0x0014U) | Offset address of IMR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMA. |
| IMPDRV\_REG\_DMA\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMA. |
| IMPDRV\_REG\_DMA\_SR1 | (0x0030U) | Offset address of SR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMA. |
| IMPDRV\_DMA\_VCR | (0x19102516U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_REG\_DMA\_THREAD\_OFFSET | (0x00000800U) | H/W Register offset for DMA sub-thread. |
| IMPDRV\_DMA\_SR\_TEND | (0x00000001U) | /\*\*<SA\_TEND interrupt enable registration. |
| IMPDRV\_DMA\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMA\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMA\_SR\_WUP | (0x00000080U) | WUP interrupt enable registration. |
| IMPDRV\_DMA\_SER\_FIX | (0x0000F300U) | Status Enable Registers that are always bit 1. |
| IMPDRV\_DMA\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_MASK | (0x8000U) | Bit mask of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_LOOPTIME\_VAL | (8U) | read count for register. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE1 | (3U) | Mult-Bank memory init register info1. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE2 | (7U) | Mult-Bank memory init register info2. |
| IMPDRV\_DMA\_MB\_INFO\_SIZE3 | (5U) | Mult-Bank memory init register info3. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_ODD | (8U) | Odd mult-Bank memory init register write number of times. |
| IMPDRV\_DMA\_MB\_WRITE\_NUM\_EVEN | (8U) | Even mult-Bank memory init register write number of times. |
| IMPDRV\_REG\_DMA\_IMGSIZER | (0x0094U) | Offset address of IMGSIZER register for DMA. |
| IMPDRV\_DMA\_IMGSIZER\_VAL | (0x00800001U) | IMGSIZER register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR0 | (0x00A0U) | Offset address of FCR0 register for DMA. |
| IMPDRV\_DMA\_FCR0\_VAL | (0x000000CCU) | FCR0 register value for DMA. |
| IMPDRV\_REG\_DMA\_FCR1 | (0x00ACU) | Offset address of FCR1 register for DMA. |
| IMPDRV\_DMA\_FCR1\_VAL | (0x90000000U) | FCR1 register value for DMA. |
| IMPDRV\_REG\_DMA\_S0SAR | (0x0040U) | Offset address of S0SAR register for DMA. |
| IMPDRV\_DMA\_S0SAR\_ODD\_BASE\_ADDR | (0xED020000U) | S0SAR register value for DMA odd bank. |
| IMPDRV\_DMA\_S0SAR\_EVEN\_BASE\_ADDR | (0xED000000U) | S0SAR register value for DMA even bank. |
| IMPDRV\_DMA\_S0SAR\_OFFSET | (0x00040000U) | S0SAR register offset for DMA even bank. |
| IMPDRV\_REG\_DMA\_S0STR | (0x0044U) | Offset address of S0STR register for DMA. |
| IMPDRV\_DMA\_S0STR\_VAL | (0x00000020U) | S0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0DATAR | (0x004CU) | Offset address of S0DATAR register for DMA. |
| IMPDRV\_DMA\_S0DATAR\_VAL | (0x00000000U) | S0DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S0CR | (0x0050U) | Offset address of S0CR register for DMA. |
| IMPDRV\_DMA\_S0CR\_VAL | (0x80000003U) | S0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1SAR | (0x0060U) | Offset address of S1SAR register for DMA. |
| IMPDRV\_DMA\_S1SAR\_VAL | (0x00000000U) | S1SAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1STR | (0x0064U) | Offset address of S1STR register for DMA. |
| IMPDRV\_DMA\_S1STR\_VAL | (0x00000020U) | S1STR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1DATAR | (0x006CU) | Offset address of S1DATAR register for DMA. |
| IMPDRV\_DMA\_S1DATAR\_VAL | (0x00000000U) | S1DATAR register value for DMA. |
| IMPDRV\_REG\_DMA\_S1CR | (0x0070U) | Offset address of S1CR register for DMA. |
| IMPDRV\_DMA\_S1CR\_VAL | (0x00000003U) | S1CR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0SAR | (0x0080U) | Offset address of D0SAR register for DMA. |
| IMPDRV\_REG\_DMA\_D0STR | (0x0084U) | Offset address of D0STR register for DMA. |
| IMPDRV\_DMA\_D0STR\_VAL | (0x00000020U) | D0STR register value for DMA. |
| IMPDRV\_REG\_DMA\_D0CR | (0x0090U) | Offset address of D0CR register for DMA. |
| IMPDRV\_DMA\_D0CR\_VAL | (0x00000003U) | D0CR register value for DMA. |
| IMPDRV\_REG\_DMA\_DSWPR | (0x0038U) | Offset address of DSWPR register for DMA. |
| IMPDRV\_DMA\_DSWPR\_VAL | (0x00000000U) | DSWPR register value for DMA. |
| IMPDRV\_REG\_DMA\_TSCR | (0x0020U) | Offset address of TSCR register for DMA. |
| IMPDRV\_DMA\_TSCR\_VAL | (0x00330003U) | TSCR register value for DMA. |
| IMPDRV\_DMA\_SCTLR0\_MB\_VAL | (0x00000001U) | SCTLR0 multi-bank register value for DMA. |
| IMPDRV\_DMA\_SCTLR0\_MB\_VAL | (0x00000001U) | SCTLR0 multi-bank register value for DMA. |
| IMPDRV\_DMA\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check. |
| IMPDRV\_DMA\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_DMA\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check. |
| IMPDRV\_DMA\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check. |
| IMPDRV\_OCV\_CORENUM\_VAL | (5U) | Maximum number of cores for OCV. |
| IMPDRV\_OCV\_OFFSET\_32B | (32U) | Offset 32byte for OCV. |
| IMPDRV\_REG\_OCV\_VCR0 | (0x0000U) | Offset address of VCR register for OCV. |
| IMPDRV\_REG\_OCV\_RSTR | (0x0008U) | Offset address of RSTR register for OCV. |
| IMPDRV\_REG\_OCV\_CR | (0x000CU) | Offset address of CR register for OCV. |
| IMPDRV\_REG\_OCV\_SR0 | (0x0010U) | Offset address of SR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR1 | (0x0014U) | Offset address of SR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR1 | (0x0018U) | Offset address of SCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR1 | (0x001CU) | Offset address of ICR1 register for OCV. |
| IMPDRV\_REG\_OCV\_IMR1 | (0x0020U) | Offset address of IMR1 register for OCV. |
| IMPDRV\_REG\_OCV\_MCR0 | (0x0040U) | Offset address of MCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_DLSAR | (0x0180U) | Offset address of DLSAR register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR0 | (0x04C0U) | Offset address of SYNCCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR1 | (0x04C4U) | Offset address of SYNCCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR2 | (0x04C8U) | Offset address of SYNCCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR3 | (0x04CCU) | Offset address of SYNCCR3 register for OCV. |
| IMPDRV\_REG\_OCV\_MEMINITR | (0x00F8U) | Offset address of MEMINITR register for OCV. |
| IMPDRV\_OCV\_VCR | (0x00000014U) | Hardware version value for OCV. |
| IMPDRV\_OCV\_RESET\_STATUS\_VAL | (0x00F10000U) | RESET\_STATUS value for OCV. |
| IMPDRV\_OCV\_WM\_INIT\_VAL | (0x00F1U) | Working Memory init value for OCV. |
| IMPDRV\_OCV\_OFST\_START\_VAL | (0xC000U) | OFST\_START value for OCV. |
| IMPDRV\_OCV\_OFST\_END\_VAL | (0xE000U) | OFST\_END value for OCV. |
| IMPDRV\_OCV\_1KB\_VAL | (0x80064100U) | 1KB value for OCV. |
| IMPDRV\_OCV\_MCR0\_INIT\_VAL | (0x01000000U) | MCR0 value for OCV. |
| IMPDRV\_OCV\_CR\_PS | (0x00000001U) | PS register value for OCV. |
| IMPDRV\_OCV\_RSTR\_SWRST | (0x00000001U) | SWRST register value for OCV. |
| IMPDRV\_OCV\_RSTR\_CLR | (0x00000000U) | CLR register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL0 | (0x00040000U) | DCBANKSEL0register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL1 | (0x05000000U) | DCBANKSEL1 register value for OCV. |
| IMPDRV\_OCV\_SR1\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_PBCOVF | (0x00000008U) | PBCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_SBO0ME | (0x00000010U) | SBO0ME interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USIER | (0x00001000U) | USIER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USINT | (0x00002000U) | USINT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_REG\_OCV\_SR2 | (0x0024U) | Offset address of SR2 register for OCV |
| IMPDRV\_REG\_OCV\_SCR2 | (0x0028U) | Offset address of SCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR2 | (0x002CU) | Offset address of ICR2 register for OCV |
| IMPDRV\_OCV\_SR1\_UDIPSBRK | (0x10000000U) | UDIPSBRK interrupt enable registration |
| IMPDRV\_OCV\_SR1\_UDIVSBRK | (0x20000000U) | UDIVSBRK interrupt enable registration. |
| IMPDRV\_OCV\_SR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SR2 register. |
| IMPDRV\_OCV\_SCR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SCR2 register |
| IMPDRV\_OCV\_ICR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in ICR2 register |
| IMPDRV\_PSC\_CORENUM\_VAL | (1U) | Maximum number of cores for PSC. |
| IMPDRV\_REG\_PSC\_VCR | (0x0000U) | Offset address of VCR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR0 | (0x0004U) | Offset address of CTLR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SR | (0x0008U) | Offset address of SR register for PSC. |
| IMPDRV\_REG\_PSC\_SCR | (0x000CU) | Offset address of SCR register for PSC. |
| IMPDRV\_REG\_PSC\_SER | (0x0010U) | Offset address of SER register for PSC. |
| IMPDRV\_REG\_PSC\_IMR | (0x0014U) | Offset address of IMR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SR1 | (0x003CU) | Offset address of SR1 register for PSC. |
| IMPDRV\_REG\_PSC\_CLSAR | (0x0038U) | Offset address of CLSAR register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR0 | (0x0058U) | Offset address of SYNCCR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR1 | (0x0060U) | Offset address of SYNCCR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR2 | (0x0064U) | Offset address of SYNCCR2 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR3 | (0x0068U) | Offset address of SYNCCR3 register for PSC. |
| IMPDRV\_PSC\_VCR | (0x17020121U) | Hardware version value for PSC. |
| IMPDRV\_PSC\_SCTLR0\_SWRST | (0x80000000U) | SWRST register value for IMP. |
| IMPDRV\_PSC\_SCTLR0\_CLR | (0x00000000U) | CLR register value for IMP. |
| IMPDRV\_PSC\_SCTLR1\_CLE | (0x00000001U) | CLE register value for IMP. |
| IMPDRV\_PSC\_SR\_TEND | (0x00000001U) | TEND interrupt enable registration. |
| IMPDRV\_PSC\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_PSC\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_PSC\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_PSC\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of PSC. |
| IMPDRV\_PSC\_SR\_WUP | (0x00010000U) | WUP interrupt enable registration. |
| IMPDRV\_IMPS\_CORENUM\_VAL | (1U) | Maximum number of cores for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_VCR | (0x0004U) | Offset address of VCR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_CNF | (0x0008U) | Offset address of CNF register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IFCFG | (0x0018U) | Offset address of CNFG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IFCTL | (0x001CU) | Offset address of IFCTL register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_INTSTS | (0x0010U) | Offset address of INTSTS register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_INTEN | (0x0014U) | Offset address of INTEN register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PSA | (0x0020U) | Offset address of PSA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_HMPTR | (0x0514U) | Offset address of HMPTR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_HMDATA | (0x0518U) | Offset address of HMDATA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for Slim-IMP. |
| IMPDRV\_IMPS\_VCR | (0x00610000U) | Hardware version value for Slim-IMP. |
| IMPDRV\_IMPS\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCTL\_EXE | (0x00000001U) | EXE register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for Slim-IMP. |
| IMPDRV\_IMPS\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_MEM\_FUNC\_SA\_VAL | (0x0030U) | SA\_VAL register value for Slim-IMP. |
| IMPDRV\_IMPS\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMPS\_INTSTS\_CLR | (IMPDRV\_IMPS\_INTSTS\_TRAP|IMPDRV\_IMPS\_INTSTS\_IER|IMPDRV\_IMPS\_INTSTS\_INT|IMPDRV\_IMPS\_INTSTS\_WUP|IMPDRV\_IMPS\_INTSTS\_HPINT|IMPDRV\_IMPS\_INTSTS\_APIPINT) | CLR register value for Slim-IMP. |
| IMPDRV\_DMAS\_CORENUM\_VAL | (2U) | Maximum number of cores for DMAS. |
| IMPDRV\_DMAS\_CORENUM0MAIN | (0U) | DMAS Core 0 Main thread identification value. |
| IMPDRV\_DMAS\_CORENUM0SUB | (1U) | DMAS Core 0 Sub thread identification value. |
| IMPDRV\_REG\_DMAS\_VCR | (0x0000U) | Offset address of VCR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SR | (0x0008U) | Offset address of SR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SCR | (0x000CU) | Offset address of SCR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SER | (0x0010U) | Offset address of VCR register for DMAS. |
| IMPDRV\_REG\_DMAS\_IMR | (0x0014U) | Offset address of IMR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMAS. |
| IMPDRV\_REG\_DMAS\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMAS. |
| IMPDRV\_REG\_DMAS\_SR1 | (0x0030U) | Offset address of SR1 register for DMAS. |
| IMPDRV\_REG\_DMAS\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMAS. |
| IMPDRV\_REG\_DMAS\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMAS. |
| IMPDRV\_REG\_DMAS\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMAS. |
| IMPDRV\_REG\_DMAS\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMAS. |
| IMPDRV\_DMAS\_VCR | (0x19082120U) | Hardware version value for DMAS. |
| IMPDRV\_DMAS\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMAS. |
| IMPDRV\_DMAS\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMAS. |
| IMPDRV\_DMAS\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMAS. |
| IMPDRV\_REG\_DMAS\_THREAD\_OFFSET | (0x00000800U) | H/W Register offset for DMAS sub-thread. |
| IMPDRV\_DMAS\_SR\_TEND | (0x00000001U) | /\*\*<SA\_TEND interrupt enable registration. |
| IMPDRV\_DMAS\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMAS\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMAS\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMAS\_SR\_WUP | (0x00000080U) | WUP interrupt enable registration. |
| IMPDRV\_DMAS\_SER\_FIX | (0x0000F300U) | Status Enable Registers that are always bit 1. |
| IMPDRV\_DMAS\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMAS. |
| IMPDRV\_DMAS\_MASK | (0x8000U) | Bit mask of RAMTSTR register for DMAS. |
| IMPDRV\_DMAS\_LOOPTIME\_VAL | (8U) | read count for register. |
| IMPDRV\_DMAS\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of Slim-DMA |
| IMPDRV\_IMPS\_APCMD\_ACSCNT\_DS | (0x00010000U) | ACSCNT\_DS bit in APCMD Register. |
| IMPDRV\_IMPS\_APCMD\_ACSCNT\_SA | (0x00080000U) | ACSCNT\_SA bit in APCMD Register. |
| IMPDRV\_IMPS\_APCMD\_EX | (0x80000000U) | EX bit in APCMD Register. |
| IMPDRV\_IMPS\_MEM\_ALIGN\_BUSCHK | (127U) | Memory alignment for Bus interface check. |
| IMPDRV\_IMPS\_MEM\_MAX\_ADRS\_BUSCHK | (0xFFFFFFFFU) | Maximum memory address for Bus interface check. |
| IMPDRV\_IMPS\_MEM\_SIZE\_BUSCHK | (512U) | Memory size for Bus interface check. |
| IMPDRV\_IMPS\_REG\_MAX\_ADRS\_CNFCHK | (0x0001FFFFU) | End of offset address for ConfRegCheck of Slim-IMP. |
| IMPDRV\_IMPS\_SEND\_TIMEOUT\_BUSCHK | (0U) | Timeout period for send queue of bus interface check. |
| IMPDRV\_REG\_IMPS\_APCFG | (0x013CU) | Offset address of APCFG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APCLPX | (0x012CU) | Offset address of APCLPX register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APCMD | (0x0128U) | Offset address of APCMD register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APDLY | (0x0110U) | Offset address of APDLY register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APDSP | (0x0108U) | Offset address of APSBSP register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APLNG | (0x010CU) | Offset address of APLNG register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APMAG | (0x0114U) | sliのため対象外 |
| IMPDRV\_REG\_IMPS\_APSASP | (0x0100U) | sliのため対象外 |
| IMPDRV\_REG\_IMPS\_APSBSP | (0x0104U) | Offset address of APSBSP register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_DST | (0x0120U) | Offset address of APSIZE\_DST register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_SA | (0x0118U) | Offset address of APSIZE\_SA register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_APSIZE\_SB | (0x011CU) | Offset address of APSIZE\_SB register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_BINTHR | (0x0818U) | Offset address of BINTHR register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_BINTHR2 | (0x081CU) | Offset address of BINTHR2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_CNST | (0x0814U) | Offset address of CNST register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF02 | (0x0830U) | Offset address of COEFF02 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF02H | (0x0854U) | Offset address of COEFF02H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1214 | (0x0840U) | Offset address of COEFF1214 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1214H | (0x0864U) | Offset address of COEFF1214H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF15 | (0x0844U) | Offset address of COEFF15 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1517H | (0x0868U) | Offset address of COEFF1517H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1820 | (0x0848U) | Offset address of COEFF1820 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF1820H | (0x086CU) | Offset address of COEFF1820H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF2123 | (0x084CU) | Offset address of COEFF2123 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF2123H | (0x0870U) | Offset address of COEFF2123H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF24 | (0x0850U) | Offset address of COEFF24 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF24H | (0x0874U) | Offset address of COEFF24H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF35 | (0x0834U) | Offset address of COEFF35 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF35H | (0x0858U) | Offset address of COEFF35H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF68 | (0x0838U) | Offset address of COEFF68 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF68H | (0x085CU) | Offset address of COEFF68H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF911 | (0x083CU) | Offset address of COEFF911 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_COEFF911H | (0x0860U) | Offset address of COEFF911H register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFORM | (0x0808U) | Offset address of IPFORM register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFUN | (0x0800U) | Offset address of IPFUN register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_IPFUN2 | (0x0804U) | Offset address of IPFUN2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_KNLMSK | (0x0820U) | Offset address of KNLMSK register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_KNLMSK2 | (0x0824U) | Offset address of KNLMSK2 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_LABCNT | (0x082CU) | Offset address of LABCNT register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_LMCTL | (0x0828U) | Offset address of LMCTL register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTAMSK | (0x0A88U) | Offset address of PFFTAMSK register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTASEL1 | (0x0A80U) | Offset address of PFFTASEL1 register for Slim-IMP. |
| IMPDRV\_REG\_IMPS\_PFFTBMSK | (0x0A94U) | Offset address of PFFTBMSK register for Slim-IMP. |

Table 5‑54 Definition Values (Core Control) 「R-CarV4H]

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_MASK\_U8BIT | (0xFF000000U) | Bit mask for Upper 8 bits |
| IMPDRV\_MASK\_U9\_16BIT | (0x00FF0000U) | Bit mask for Upper 9-16 bits |
| IMPDRV\_MASK\_U16BIT | (0xFFFF0000U) | Bit mask for Upper 16 bits |
| IMPDRV\_MASK\_L8BIT | (0x000000FFU) | Bit mask for Lower 8 bits |
| IMPDRV\_MASK\_BIT15 | (0x00008000U) | Bit mask for Only 15 bit |
| IMPDRV\_INIT\_CL\_ADDR | (0xFFFFFFFFU) | Initial value of CL address data |
| IMPDRV\_CALLBACK\_CODE\_DEF | (-1) | Callback code default value |
| IMPDRV\_RESERVE\_NUM | (2U) | Number of reserve parameter arrays |
| IMPDRV\_EXEPARAM\_MAX | (4U) | Number of extended parameter arrays |
| IMPDRV\_CORE\_NUM\_MAX | (8U) | Maximum value of Core number definition |
| IMPDRV\_CB\_RET\_MAX | (18U) | Maximum value of Call back reason |
| IMPDRV\_OFFSET\_4BYTE | (4U) | For 4 byte offset calculation |
| IMPDRV\_SHIFT\_8BIT | (8U) | For 8 bit shift calculation |
| IMPDRV\_SHIFT\_16BIT | (16U) | For 16 bit shift calculation |
| IMPDRV\_SHIFT\_24BIT | (24U) | For 24 bit shift calculation |
| IMPDRV\_SYNCC\_REGNUM | (4U) | Array size for SYNCC register |
| IMPDRV\_IMP\_CORENUM\_VAL | (4U) | Maximum number of cores for IMP. |
| IMPDRV\_IMP\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address |
| IMPDRV\_REG\_IMP\_VCR | (0x0004U) | Offset address of VCR register for IMP. |
| IMPDRV\_REG\_IMP\_CNF | (0x0008U) | Offset address of CNF register for IMP. |
| IMPDRV\_REG\_IMP\_IFCFG | (0x0018U) | Offset address of CNFG register for IMP. |
| IMPDRV\_REG\_IMP\_IFCTL | (0x001CU) | Offset address of IFCTL register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR0 | (0x0040U) | Offset address of SYNCCR0 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR1 | (0x0044U) | Offset address of SYNCCR1 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR2 | (0x0048U) | Offset address of SYNCCR2 register for IMP. |
| IMPDRV\_REG\_IMP\_SYNCCR3 | (0x004CU) | Offset address of SYNCCR03 register for IMP. |
| IMPDRV\_REG\_IMP\_INTSTS | (0x0010U) | Offset address of INTSTS register for IMP. |
| IMPDRV\_REG\_IMP\_INTEN | (0x0014U) | Offset address of INTEN register for IMP. |
| IMPDRV\_REG\_IMP\_PSA | (0x0020U) | Offset address of PSA register for IMP. |
| IMPDRV\_REG\_IMP\_CLBRKADDRR | (0x00C0U) | Offset address of CLBRKADDRR register for IMP. |
| IMPDRV\_REG\_IMP\_HMPTR | (0x0514U) | Offset address of HMPTR register for IMP. |
| IMPDRV\_REG\_IMP\_HMDATA | (0x0518U) | Offset address of HMDATA register for IMP. |
| IMPDRV\_REG\_IMP\_HM1PTR | (0x0594U) | Offset address of HM1PTR register for IMP. |
| IMPDRV\_REG\_IMP\_HM1DATA | (0x0598U) | Offset address of HM1DATA register for IMP. |
| IMPDRV\_REG\_IMP\_MEM\_FUNC\_SA | (0x0B78U) | Offset address of SA register for IMP. |
| IMPDRV\_IMP\_VCR | (0x00600000U) | Hardware version value for IMP. |
| IMPDRV\_IMP\_CNF\_VAL | (0x00000001U) | CNF\_VAL register value for IMP. |
| IMPDRV\_IMP\_CNF\_SWRST | (0x80000000U) | CNF\_SWRST register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_EXE | (0x00000001U) | EXE register value for IMP. |
| IMPDRV\_IMP\_IFCFG\_VAL | (0x010010CCU) | IFCFG\_VAL register value for IMP. |
| IMPDRV\_IMP\_IFCTL\_ENDIAN | (0x22020200U) | ENDIAN register value for IMP. |
| IMPDRV\_IMP\_HMPTR\_VAL | (0x87FFU) | HMPTR\_VAL register value for IMP |
| IMPDRV\_IMP\_HMDATA\_VAL | (0x0000U) | HMDATA\_VAL register value for IMP. |
| IMPDRV\_IMP\_MEM\_FUNC\_SA\_VAL | (0x00FFU) | SA\_VAL register value for IMP. |
| IMPDRV\_IMP\_INTSTS\_MASKCPU | (0x10000000U) | MASKCPU interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_APIPINT | (0x00000001U) | APIPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_HPINT | (0x00000002U) | HPINT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_WUP | (0x04000000U) | WUP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_INT | (0x00000020U) | INT interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_IER | (0x00000080U) | IER interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_TRAP | (0x00000040U) | TRAP interrupt enable registration. |
| IMPDRV\_IMP\_INTSTS\_CLR | (IMPDRV\_IMP\_INTSTS\_TRAP | IMPDRV\_IMP\_INTSTS\_IER | IMPDRV\_IMP\_INTSTS\_INT | IMPDRV\_IMP\_INTSTS\_WUP | IMPDRV\_IMP\_INTSTS\_HPINT | IMPDRV\_IMP\_INTSTS\_APIPINT | IMPDRV\_IMP\_INTSTS\_CLBRK) | CLR register value for IMP. |
| IMPDRV\_IMP\_REG\_MAX\_ADRS\_CNFCHK | (0x0001FFFFU) | End of offset address for ConfRegCheck of IMP |
| IMPDRV\_CNN\_CORENUM\_VAL | (1U) | Maximum number of cores for CNN. |
| IMPDRV\_CNN\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address |
| IMPDRV\_REG\_CNN\_VCR | (0x0000U) | Offset address of VCR register for CNN. |
| IMPDRV\_CNN\_VCR | (0x00030200U) | Hardware version value for CNN. |
| IMPDRV\_REG\_CNN\_SWRST | (0x0008U) | Offset address of hardware register for SWRST. |
| IMPDRV\_REG\_CNN\_SR | (0x0010U) | Offset address of hardware register for SR. |
| IMPDRV\_REG\_CNN\_SRE | (0x0014U) | Offset address of hardware register for SRE. |
| IMPDRV\_REG\_CNN\_SRC | (0x0018U) | Offset address of hardware register for SRC. |
| IMPDRV\_REG\_CNN\_SRM | (0x001CU) | Offset address of hardware register for SRM. |
| IMPDRV\_REG\_CNN\_SACL | (0x0104U) | Offset address of hardware register for SACL |
| IMPDRV\_REG\_CNN\_SCLP | (0x0084U) | Offset address of hardware register for SCLP. |
| IMPDRV\_REG\_CNN\_CLBRKADDRR | (0x0114U) | Offset address of hardware register for CLBRKADDRR |
| IMPDRV\_REG\_CNN\_SYNCCR0 | (0x0900U) | Offset address of hardware register for SYNCCR0. |
| IMPDRV\_REG\_CNN\_SYNCCR1 | (0x0904U) | Offset address of hardware register for SYNCCR1. |
| IMPDRV\_REG\_CNN\_SYNCCR2 | (0x0908U) | Offset address of hardware register for YNCCR2. |
| IMPDRV\_REG\_CNN\_SYNCCR3 | (0x090CU) | Offset address of hardware register for MG2ISYNCCR3NTSEL. |
| IMPDRV\_CNN\_CNF\_SWRST | (0x00000001U) | SWRST register value for CNN. |
| IMPDRV\_CNN\_CNF\_CLR | (0x00000000U) | CLR register value for CNN. |
| IMPDRV\_CNN\_SR\_CLBUSY | (0x00002000U) | CLBSY register value for CNN. |
| IMPDRV\_CNN\_SR\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_CNN\_SR\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_CNN\_SR\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_CNN\_SR\_MSCO | (0x00000008U) | MSCO interrupt enable registration. |
| IMPDRV\_CNN\_SR\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_CNN\_SR\_CLBRK | (0x00000020U) | CL break interrupt enable registration |
| IMPDRV\_CNN\_SCLP\_START | (0x00000001U) | CNN\_SCLP\_START register value for CNN |
| IMPDRV\_CNN\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of CNN |
| IMPDRV\_DMA\_CORENUM\_VAL | (4U) | Maximum number of cores for DMA. |
| IMPDRV\_DMA\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address |
| IMPDRV\_DMA\_CORENUM0MAIN | (0U) | DMA Core 0 Main thread identification value. |
| IMPDRV\_DMA\_CORENUM0SUB | (1U) | DMA Core 0 Sub thread identification value. |
| IMPDRV\_DMA\_CORENUM1MAIN | (2U) | Offset address of VCR register for DMA. |
| IMPDRV\_DMA\_CORENUM1SUB | (3U) | DMA Core 1 Sub thread identification value. |
| IMPDRV\_REG\_DMA\_VCR | (0x0000U) | Offset address of VCR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR | (0x0004U) | Offset address of SCTLR register for DMA. |
| IMPDRV\_REG\_DMA\_SR | (0x0008U) | Offset address of SR register for DMA. |
| IMPDRV\_REG\_DMA\_SCR | (0x000CU) | Offset address of SCR register for DMA. |
| IMPDRV\_REG\_DMA\_SER | (0x0010U) | Offset address of SER register for DMA. |
| IMPDRV\_REG\_DMA\_IMR | (0x0014U) | Offset address of IMR register for DMA. |
| IMPDRV\_REG\_DMA\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for DMA. |
| IMPDRV\_REG\_DMA\_CLSAR | (0x0028U) | Offset address of CLSAR register for DMA. |
| IMPDRV\_REG\_DMA\_SR1 | (0x0030U) | Offset address of SR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR0 | (0x00E8U) | Offset address of SYNCCR0 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR1 | (0x00ECU) | Offset address of SYNCCR1 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR2 | (0x00F0U) | Offset address of SYNCCR2 register for DMA. |
| IMPDRV\_REG\_DMA\_SYNCCR3 | (0x00F4U) | Offset address of SYNCCR3 register for DMA. |
| IMPDRV\_DMA\_VCR | (0x17021420U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_SWRST | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR\_CLR | (0x00000000U) | Hardware version value for DMA. |
| IMPDRV\_DMA\_SCTLR1\_PS | (0x80000000U) | Hardware version value for DMA. |
| IMPDRV\_REG\_DMA\_THREAD\_OFFSET | (0x00000800U) | H/W Register offset for DMA sub-thread. |
| IMPDRV\_DMA\_SR\_TEND | (0x00000001U) | SA\_TEND interrupt enable registration. |
| IMPDRV\_DMA\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_DMA\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_DMA\_SR\_WUP | (0x00000080U) | WUP interrupt enable registration. |
| IMPDRV\_DMA\_SR\_CLBRK | (0x00000400U) | CLBRK interrupt enable registration |
| IMPDRV\_DMA\_SER\_FIX | (0x0000F300U) | Status Enable Registers that are always bit 1. |
| IMPDRV\_DMA\_RAMTSTR | (0x00D0U) | Offset address of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_MASK | (0x8000U) | Bit mask of RAMTSTR register for DMA. |
| IMPDRV\_DMA\_LOOPTIME\_VAL | (8U) | read count for register. |
| IMPDRV\_DMA\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of DMA. |
| IMPDRV\_OCV\_CORENUM\_VAL | (4U) | Maximum number of cores for OCV. |
| IMPDRV\_OCV\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address |
| IMPDRV\_OCV\_OFFSET\_32B | (32U) | Offset 32byte for OCV. |
| IMPDRV\_REG\_OCV\_VCR0 | (0x0000U) | Offset address of VCR register for OCV. |
| IMPDRV\_REG\_OCV\_RSTR | (0x0008U) | Offset address of RSTR register for OCV. |
| IMPDRV\_REG\_OCV\_CR | (0x000CU) | Offset address of CR register for OCV. |
| IMPDRV\_REG\_OCV\_SR0 | (0x0010U) | Offset address of SR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR1 | (0x0014U) | Offset address of SR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR1 | (0x0018U) | Offset address of SCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR1 | (0x001CU) | Offset address of ICR1 register for OCV. |
| IMPDRV\_REG\_OCV\_IMR1 | (0x0020U) | Offset address of IMR1 register for OCV. |
| IMPDRV\_REG\_OCV\_MCR0 | (0x0040U) | Offset address of MCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SR2 | (0x0024U) | Offset address of SR2 register for OCV. |
| IMPDRV\_REG\_OCV\_SCR2 | (0x0028U) | Offset address of SCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_ICR2 | (0x002CU) | Offset address of ICR2 register for OCV. |
| IMPDRV\_REG\_OCV\_DLSAR | (0x0180U) | Offset address of DLSAR register for OCV. |
| IMPDRV\_REG\_OCV\_CLBRKADDRR | (0x03F4U) | Offset address of CLBRKADDRR register for OCV |
| IMPDRV\_REG\_OCV\_SYNCCR0 | (0x04C0U) | Offset address of SYNCCR0 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR1 | (0x04C4U) | Offset address of SYNCCR1 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR2 | (0x04C8U) | Offset address of SYNCCR2 register for OCV. |
| IMPDRV\_REG\_OCV\_SYNCCR3 | (0x04CCU) | Offset address of SYNCCR3 register for OCV. |
| IMPDRV\_REG\_OCV\_MEMINITR | (0x00F8U) | Offset address of MEMINITR register for OCV. |
| IMPDRV\_OCV\_VCR | (0x00000004U) | Hardware version value for OCV. |
| IMPDRV\_OCV\_RESET\_STATUS\_VAL | (0x00F10000U) | RESET\_STATUS value for OCV. |
| IMPDRV\_OCV\_WM\_INIT\_VAL | (0x00F1U) | Working Memory init value for OCV |
| IMPDRV\_OCV\_OFST\_START\_VAL | (0xC000U) | OFST\_START value for OCV. |
| IMPDRV\_OCV\_OFST\_END\_VAL | (0xE000U) | OFST\_END value for OCV. |
| IMPDRV\_OCV\_1KB\_VAL | (0x80064100U) | 1KB value for OCV. |
| IMPDRV\_OCV\_MCR0\_INIT\_VAL | (0x01000000U) | MCR0 value for OCV. |
| IMPDRV\_OCV\_CR\_PS | (0x00000001U) | PS register value for OCV. |
| IMPDRV\_OCV\_RSTR\_SWRST | (0x00000001U) | SWRST register value for OCV. |
| IMPDRV\_OCV\_RSTR\_CLR | (0x00000000U) | CLR register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL0 | (0x00040000U) | DCBANKSEL0register value for OCV. |
| IMPDRV\_OCV\_MCR0\_DCBANKSEL1 | (0x05000000U) | DCBANKSEL1 register value for OCV. |
| IMPDRV\_OCV\_SR1\_TRAP | (0x00000001U) | TRAP interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_IER | (0x00000002U) | IER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_INT | (0x00000004U) | INT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_PBCOVF | (0x00000008U) | PBCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_SBO0ME | (0x00000010U) | SBO0ME interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_CLBRK | (0x00000400U) | CLBRK interrupt enable registration |
| IMPDRV\_OCV\_SR1\_USIER | (0x00001000U) | USIER interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_USINT | (0x00002000U) | USINT interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_WUPCOVF | (0x00004000U) | WUPCOVF interrupt enable registration. |
| IMPDRV\_OCV\_SR1\_UDIPSBRK | (0x10000000U) | UDIPSBRK interrupt enable registration |
| IMPDRV\_OCV\_SR1\_UDIVSBRK | (0x20000000U) | UDIVSBRK interrupt enable registration |
| IMPDRV\_OCV\_SR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SR2 register. |
| IMPDRV\_OCV\_SCR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in SCR2 register |
| IMPDRV\_OCV\_ICR2\_MASK | (0xFFFFFFFFU) | Shader thread mask bits of used in ICR2 register |
| IMPDRV\_OCV\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of OCV |
| IMPDRV\_PSC\_CORENUM\_VAL | (1U) | Maximum number of cores for PSC. |
| IMPDRV\_PSC\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address |
| IMPDRV\_REG\_PSC\_VCR | (0x0000U) | Offset address of VCR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR0 | (0x0004U) | Offset address of CTLR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SR | (0x0008U) | Offset address of SR register for PSC. |
| IMPDRV\_REG\_PSC\_SCR | (0x000CU) | Offset address of SCR register for PSC. |
| IMPDRV\_REG\_PSC\_SER | (0x0010U) | Offset address of SER register for PSC. |
| IMPDRV\_REG\_PSC\_IMR | (0x0014U) | Offset address of IMR register for PSC. |
| IMPDRV\_REG\_PSC\_SCTLR1 | (0x0018U) | Offset address of SCTLR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SR1 | (0x003CU) | Offset address of SR1 register for PSC. |
| IMPDRV\_REG\_PSC\_CLSAR | (0x0038U) | Offset address of CLSAR register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR0 | (0x0058U) | Offset address of SYNCCR0 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR1 | (0x0060U) | Offset address of SYNCCR1 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR2 | (0x0064U) | Offset address of SYNCCR2 register for PSC. |
| IMPDRV\_REG\_PSC\_SYNCCR3 | (0x0068U) | Offset address of SYNCCR3 register for PSC. |
| IMPDRV\_PSC\_VCR | (0x17020121U) | Hardware version value for PSC. |
| IMPDRV\_PSC\_SCTLR0\_SWRST | (0x80000000U) | SWRST register value for IMP. |
| IMPDRV\_PSC\_SCTLR0\_CLR | (0x00000000U) | CLR register value for IMP. |
| IMPDRV\_PSC\_SCTLR1\_CLE | (0x00000001U) | CLE register value for IMP. |
| IMPDRV\_PSC\_SR\_TEND | (0x00000001U) | TEND interrupt enable registration. |
| IMPDRV\_PSC\_SR\_TRAP | (0x00000010U) | TRAP interrupt enable registration. |
| IMPDRV\_PSC\_SR\_IER | (0x00000020U) | IER interrupt enable registration. |
| IMPDRV\_PSC\_SR\_INT | (0x00000040U) | INT interrupt enable registration. |
| IMPDRV\_PSC\_SR\_WUP | (0x00010000U) | WUP interrupt enable registration |
| IMPDRV\_PSC\_REG\_MAX\_ADRS\_CNFCHK | (0x0000FFFFU) | End of offset address for ConfRegCheck of PSC |
| IMPDRV\_OSAL\_DSP\_DEV\_NUM\_MAX | (IMPDRV\_DSP\_DEV\_NUM\_MAX + 1U) | Num of max sub device for OSAL DSP core. |
| IMPDRV\_CNN\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_CNN\_EXEPARAM\_GOSUB | (1U) | Index value of extend parameter for conditional GOSUB instruction. |
| IMPDRV\_REG\_CNN\_CLCNDGSBR | (0x0118U) | Offset address of CLCNDGSBR register for CNN. |
| IMPDRV\_DMA\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_DMA\_EXEPARAM\_GOSUB | (1U) | Index value of extend parameter for conditional GOSUB instruction. |
| IMPDRV\_REG\_DMA\_CLBRKADDRR | (0x03F4U) | Offset address of CLBRKADDRR register for DMA. |
| IMPDRV\_REG\_DMA\_CLCNDGSBR | (0x03FCU) | Offset address of CLCNDGSBR register for DMA. |
| IMPDRV\_DMAS\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_DMAS\_CORENUM1MAIN | (2U) | DMAS Core 1 Main thread identification value. |
| IMPDRV\_DMAS\_CORENUM1SUB | (3U) | DMAS Core 1 Sub thread identification value. |
| IMPDRV\_DMAS\_EXEPARAM\_CLBRK | (0U) | Index value of extend parameter for CL break address. |
| IMPDRV\_DMAS\_EXEPARAM\_GOSUB | (1U) | Index value of extend param for conditional GOSUB. |
| IMPDRV\_DMAS\_SR\_CLBRK | (0x00000400U) | CLBRK interrupt enable registration. |
| IMPDRV\_DMAS\_V1\_VCR | (0x20021013U) | Hardware version value for DMAS. |
| IMPDRV\_DMAS\_V2\_VCR | (0x22032218U) | Hardware version value for DMAS. |
| IMPDRV\_REG\_DMAS\_CLBRKADDRR | (0x03F4U) | Offset address of CLBRKADDRR register for DMA. |
| IMPDRV\_REG\_DMAS\_CLCNDGSBR | (0x03FCU) | Offset address of CLCNDGSBR register for DMAS. |
| IMPDRV\_DSP\_CORENUM\_VAL | (4U) | Maximum number of cores for IMP. |
| IMPDRV\_DSP\_REG\_MAX\_ADRS\_CNFCHK | (0x0019FFFFU) | End of offset address for ConfRegCheck of IMP. |
| IMPDRV\_DSP\_VDSP\_DEV\_MAIN | (0u) | VDSP device main. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB0 | (1u) | VDSP device sub0. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB1 | (2u) | VDSP device sub1. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB2 | (3u) | VDSP device sub2. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB3 | (4u) | VDSP device sub3. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB4 | (5u) | VDSP device sub4. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB5 | (6u) | VDSP device sub5. |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB6 | (7u) | VDSP device sub6. |
| IMPDRV\_IMPREG\_ADDR\_END | (0xFFC00000U) | D\_ADD6\_START register REGION\_START value for VDSP. |
| IMPDRV\_IMPREG\_ADDR\_START | (0xF1000000U) | D\_ADD5\_START register REGION\_START value for VDSP. |
| IMPDRV\_REG\_IMPSLV\_DSP0CSTS | (0x00004014U) | Offset address of DSP ch0 Communication STaTus. |
| IMPDRV\_REG\_IMPSLV\_DSP1CSTS | (0x00004024U) | Offset address of DSP ch1 Communication STaTus. |
| IMPDRV\_REG\_IMPSLV\_DSP2CSTS | (0x00004034U) | Offset address of DSP ch2 Communication STaTus. |
| IMPDRV\_REG\_IMPSLV\_DSP3CSTS | (0x00004044U) | Offset address of DSP ch3 Communication STaTus. |
| IMPDRV\_REGOFS\_D\_ADD0\_ATT0 | (0x80924U) | Offset address of D\_ADD0\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD0\_ATT1 | (0x80928U) | Offset address of D\_ADD1\_ATT1 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD1\_ATT0 | (0x80934U) | Offset address of D\_ADD1\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD1\_ATT1 | (0x80938U) | Offset address of D\_ADD1\_ATT1 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD1\_START | (0x80930U) | Offset address of D\_ADD1\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD2\_START | (0x80940U) | Offset address of D\_ADD2\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD3\_ATT0 | (0x80954U) | Offset address of D\_ADD3\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD3\_ATT1 | (0x80958U) | Offset address of D\_ADD3\_ATT1 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD3\_START | (0x80950U) | Offset address of D\_ADD3\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD4\_START | (0x80960U) | Offset address of D\_ADD4\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD5\_ATT0 | (0x80974U) | Offset address of D\_ADD5\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD5\_ATT1 | (0x80978U) | Offset address of D\_ADD5\_ATT1 register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD5\_START | (0x80970U) | Offset address of D\_ADD5\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_ADD6\_START | (0x80980U) | Offset address of D\_ADD6\_START register for VDSP |
| IMPDRV\_REGOFS\_D\_DDCL | (0x8068CU) | Offset address of MMS\_DDCL register for VDSP |
| IMPDRV\_REGOFS\_D\_DDEA | (0x80680U) | Offset address of MMS\_DDEA register for VDSP |
| IMPDRV\_REGOFS\_D\_DDESC\_ID | (0x806A0U) | Offset address of MSS\_DDESC\_ID register for VDSP |
| IMPDRV\_REGOFS\_D\_DDIA | (0x80684U) | Offset address of MMS\_DDIA register for VDSP |
| IMPDRV\_REGOFS\_D\_DDTC | (0x80658U) | Offset address of MMS\_DDTC register for VDSP |
| IMPDRV\_REGOFS\_D\_SFT\_SCPD\_0 | (0x807E0U) | Offset address of IMPDRV\_REGOFS\_D\_SFT\_SCPD\_0 register for VDSP |
| IMPDRV\_REGOFS\_D\_SFT\_SCPD\_1 | (0x807E4U) | Offset address of IMPDRV\_REGOFS\_D\_SFT\_SCPD\_1 register for VDSP |
| IMPDRV\_REGOFS\_D\_SFT\_SCPD\_2 | (0x807E8U) | Offset address of IMPDRV\_REGOFS\_D\_SFT\_SCPD\_2 register for VDSP |
| IMPDRV\_REGOFS\_D\_SFT\_SCPD\_3 | (0x807ECU) | Offset address of IMPDRV\_REGOFS\_D\_SFT\_SCPD\_3 register for VDSP |
| IMPDRV\_REGOFS\_DMBE | (0x80604U) | Offset address of MSS\_DMBE register for VDSP |
| IMPDRV\_REGOFS\_MSS\_DMBA | (0x80600U) | Offset address of MSS\_DMBA register for VDSP |
| IMPDRV\_REGOFS\_MSS\_PCR | (0x80400U) | Offset address of MSS\_PCR register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD0\_ATT0 | (0x80424U) | Offset address of P\_ADD0\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD1\_ATT0 | (0x8042CU) | Offset address of P\_ADD1\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD1\_START | (0x80428U) | Offset address of P\_ADD1\_START register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD2\_ATT0 | (0x80434U) | Offset address of P\_ADD2\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD2\_START | (0x80430U) | Offset address of P\_ADD2\_START register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD3\_ATT0 | (0x8043CU) | Offset address of P\_ADD3\_ATT0 register for VDSP |
| IMPDRV\_REGOFS\_P\_ADD3\_START | (0x80438U) | Offset address of P\_ADD3\_START register for VDSP. |
| IMPDRV\_REGOFS\_P\_ADD4\_START | (0x80440U) | Offset address of P\_ADD4\_START register for VDSP |
| IMPDRV\_REGOFS\_P\_CCOCR | (0x804E4U) | Offset address of P\_CCOCR register for VDSP |
| IMPDRV\_REGOFS\_P\_CCOSAR | (0x804E0U) | Offset address of P\_CCOSAR register for VDSP |
| IMPDRV\_REGOFS\_P\_PDEA | (0x80410U) | Offset address of P\_PDEA register for VDSP |
| IMPDRV\_REGOFS\_P\_PDIA | (0x80414U) | Offset address of P\_PDIA register for VDSP |
| IMPDRV\_REGOFS\_P\_PDTC | (0x80418U) | Offset address of P\_PDTC register for VDSP |
| IMPDRV\_REGOFS\_PSVM | (0x80E50U) | Offset address of PSVM register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_CONTROL | (0xC0038U) | Offset address of DSP\_CONTROL register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_GPIO\_IN | (0xC0014U) | Offset address of GPIO\_IN register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_GPIO\_OUT | (0xC0080U) | Offset address of GPIO\_OUT register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_ICU\_ACU\_LOCK | (0xC0004U) | Offset address of ICU\_ACU\_LOCK register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_OUT\_INT\_CLR | (0xC0044U) | Offset address of OUT\_INT\_CLR\_I register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_OUT\_INT\_EN | (0xC0040U) | Offset address of OUT\_INT\_EN\_I register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_OUT\_INT\_MSK | (0xC0048U) | Offset address of OUT\_INT\_MSK\_I register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_OUT\_INT\_STA | (0xC004CU) | Offset address of OUT\_INT\_STATE\_I register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_PSU\_STATE | (0xC007CU) | Offset address of PSU\_STATE register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_RST\_VEC\_ADR | (0xC0010U) | Offset address of RESET\_VEC register for VDSP |
| IMPDRV\_REGOFS\_VDSP\_VERSION1 | (0xC0094U) | Offset address of VERSION1 register for VDSP. |
| IMPDRV\_REGVAL\_CLR | (0x00000000U) | register clear value for VDSP. |
| IMPDRV\_REGVAL\_D\_SFT\_SCPD\_0\_OFF | (0x00U) | table num for crc rable. |
| IMPDRV\_REGVAL\_D\_SFT\_SCPD\_0\_ON | (0x01U) | table num for crc rable. |
| IMPDRV\_REGVAL\_DMBE\_IDLE | (0x00011FFFU) | MSS\_DMBE register value for VDSP |
| IMPDRV\_REGVAL\_DMBE\_INIT | (0x00000000U) | MSS\_DMBE register value for VDSP. |
| IMPDRV\_REGVAL\_IMPSLV\_DSPNCSTS | (0x00000000U) | DSPnCSTS register value. |
| IMPDRV\_REGVAL\_OUT\_INT\_TIMER00 | (0x0001U) | I\_TIMER00 value for VDSP. |
| IMPDRV\_REGVAL\_OUT\_INT\_TIMER01 | (0x0002U) | I\_TIMER00 value for VDSP. |
| IMPDRV\_REGVAL\_PSU\_STATE\_LSP | (0x00000008U) | PSU\_STATE register LIGHT\_SLP\_MD value for VDSP. |
| IMPDRV\_REGVAL\_PSU\_STATE\_SBY | (0x00000004U) | PSU\_STATE register STANDBY\_MD value for VDSP. |
| IMPDRV\_REGVAL\_PSVM\_PMOD\_SBY | (0x00000003U) | PSVM register value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_ACU\_LOCK | (0x0U) | ICU\_ACU\_LOCK register lock value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_ACU\_UNLOCK | (0x8U) | ICU\_ACU\_LOCK register unlock value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_BSZ\_8\_INCR | (0xA0) | MSS\_DDCL register BSZ 1000: 128 transfers (INCR) value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_CRC\_ERROR | (0x80000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_CRITICAL\_INT | (0x20000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_DIV0 | (0x08000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_ERROR | (0xFF000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_EXT\_DABSZ8 | ((uint32\_t)0xAU << 9U) | D\_ADDx\_ATT1 register DABSZ value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_DADOL | ((uint32\_t)0xFU << 13U) | D\_ADDx\_ATT1 register DADOL value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_DAUOL | ((uint32\_t)0x4U << 18U) | D\_ADDx\_ATT1 register DAUOL value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_DPRAW | ((uint32\_t)0x1U << 29U) | D\_ADDx\_START register DPRAW value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_L2A\_RD | ((uint32\_t)0x2U << 5U) | D\_ADDx\_ATT1 register L2A\_RD value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_L2A\_WR | ((uint32\_t)0x2U << 1U) | D\_ADDx\_ATT1 register L2A\_WR value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_EXT\_MOM | ((uint32\_t)0x1U << 16U) | D\_ADDx\_ATT0 register MOM value for VDSP |
| IMPDRV\_REGVAL\_VDSP\_GPIO\_FSTDBY | (0x00000800U) | GPIO\_IN register F standby value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_GPIO\_RESUME | (0x00002000U) | GPIO\_OUT register resume value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_GPIO\_START | (0x00000008U) | GPIO\_IN register start value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_GPIO\_STDBY | (0x00000001U) | GPIO\_OUT register standby value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_GPIO\_STOP | (0x00000080U) | GPIO\_OUT register stop value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_ICU\_LOCK | (0x1U) | ICU\_ACU\_LOCK register lock value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_ICU\_UNLOCK | (0x0U) | ICU\_ACU\_LOCK register unlock value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_INT | (0x00020000U) | GPIO\_OUT register INT value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_MSS\_PCR | (((uint32\_t)1U << 2U) | ((uint32\_t)1U << 6U) | ((uint32\_t)1U << 7U) | ((uint32\_t)3U << 8U) | ((uint32\_t)1U << 12U)) | MSS\_PCR register value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_NORMAL\_INT | (0x40000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_P\_CCOCR\_IV | (((uint32\_t)1U << 0U) | ((uint32\_t)1U << 1U) | ((uint32\_t)4U << 2U) | ((uint32\_t)1U << 7U) | ((uint32\_t)1024U << 16U)) | P\_CCOCR register Invalidate value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_P\_CCOCR\_PF | (((uint32\_t)1U << 0U) | ((uint32\_t)1U << 1U) | ((uint32\_t)3U << 2U) | ((uint32\_t)1U << 7U) | ((uint32\_t)1024U << 16U)) | P\_CCOCR register Pre-fetched value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_REG\_ACT | ((uint32\_t)0U << 28U) | D\_ADDx\_START register INACTIVE active value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_REG\_AXI0 | ((uint32\_t)0x2U << 20U) | D\_ADDx\_START register REGION\_MID AXIm0 value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_REG\_BLANK | ((uint32\_t)0x80U << 20U) | D\_ADDx\_START register REGION\_MID Blank region value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_REG\_EDP | ((uint32\_t)0x1U << 20U) | D\_ADDx\_START register REGION\_MID EDP value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_REG\_INACT | ((uint32\_t)1U << 28U) | D\_ADDx\_START register INACTIVE not active value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_RESUME | (0x00020000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_SAFETY\_ERROR | (0x10000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_STACK\_VIOLATION | (0x02000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_TIM\_FAULT | (0x01000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_TRAP | (0x00010000U) | GPIO\_OUT register TRAP value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_VERSION1 | (0x20111119U) | Hardware version value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_WDOG\_EXPIRE | (0x04000000U) | GPIO\_OUT register error value for VDSP. |
| IMPDRV\_SPMX\_ADDR\_END | (0xED600000U) | D\_ADD4\_START register REGION\_START value for VDSP. |
| IMPDRV\_SPMX\_ADDR\_START | (0xED300000U) | D\_ADD3\_START register REGION\_START value for VDSP. |
| IMPDRV\_VDSP\_CB\_CODE\_INT | (1) | Callback code INT. |
| IMPDRV\_VDSP\_CB\_CODE\_TRAP | (0) | Callback code TRAP. |
| IMPDRV\_VDSP\_CHANNEL\_OPE\_BITS | (0x0FU) | Channel operate bits. |
| IMPDRV\_VDSP\_CONTROL\_PLOAD\_STOP | (0x0001U) | DSP\_CONTROL register PLOAD\_STOP value for VDSP. |
| IMPDRV\_VDSP\_D\_ADDR\_SHIFT | (12U) | D\_ADDx\_START register data shift for VDSP. |
| IMPDRV\_VDSP\_D\_DDCL\_DDIE\_ENABLE | (400U) | The maximum wait loop count for the D\_DDCL register. |
| IMPDRV\_VDSP\_D\_DDCL\_DDIE\_RESET | (0U) | The maximum wait loop count for the D\_DDCL register. |
| IMPDRV\_VDSP\_D\_DDCL\_WAIT\_LOOP | (100000U) | The maximum wait loop count for the D\_DDCL register. |
| IMPDRV\_VDSP\_DDMA\_BUSY\_MASK | ((uint32\_t)1U<<8U) | MSS\_DDCL register DDMA\_BUSY mask for VDSP. |
| IMPDRV\_VDSP\_DMBE\_WAIT\_LOOP | (100U) | The maximum wait loop count for the MSS\_DMBE register. |
| IMPDRV\_VDSP\_EXT\_MEM\_END | (0xC0000000U) | Valid DSP application execution end address. |
| IMPDRV\_VDSP\_EXT\_MEM\_START | (0x40000000U) | Valid DSP application execution start address. |
| IMPDRV\_VDSP\_GPIO\_WAIT\_LOOP | (0x00100000U) | The maximum wait loop count for the GPIO register. |
| IMPDRV\_VDSP\_IDM\_FIXED\_SIZE\_64KB | ((uint32\_t)0x4000U) | IDM Fixed size |
| IMPDRV\_VDSP\_IPM\_FIXED\_SIZE\_8KB | ((uint32\_t)0x2000U) | IPM Fixed size |
| IMPDRV\_VDSP\_P\_ADDX\_ATT0 | (0x1U) | P\_ADDx\_ATT0 register L1IC value for VDSP. |
| IMPDRV\_VDSP\_P\_AP\_ATT0 | ((uint32\_t)0x3U << 4U) | P\_ADDx\_ATT0 register P\_AP value for VDSP. |
| IMPDRV\_VDSP\_P\_CCOCR\_WAIT\_LOOP | (100000U) | The maximum wait loop count for the P\_CCOCR register. |
| IMPDRV\_VDSP\_P\_CCOSA\_MASK | (0xFFFFFFC0U) | P\_CCOSAR register P\_CCOSA mask for VDSP. |
| IMPDRV\_VDSP\_P\_L1ICO\_MASK | ((uint32\_t)1U << 1U) | P\_CCOCR register P\_L1ICO mask for VDSP. |
| IMPDRV\_VDSP\_P\_L2A\_ATT0 | ((uint32\_t)0x2U << 8U) | P\_ADDx\_ATT0 register P\_L2A value for VDSP. |
| IMPDRV\_VDSP\_P\_PDTC\_WAIT\_LOOP | (100000U) | The maximum wait loop count for the P\_PDTC register. |
| IMPDRV\_VDSP\_PSU\_ST\_SLP\_LOOP | (100U) | Maximum wait loop count until PSU\_STATE register is in sleep mode. |
| IMPDRV\_VDSP\_PSU\_ST\_STD\_LOOP | (10U) | Maximum wait loop count until PSU\_STATE register is in stand by mode. |
| IMPDRV\_VDSP\_REGVAL\_PDMA\_BUSY | ((uint32\_t)1U << 29U) | P\_PDTC register PDTC PDMA BUSY value for VDSP. |
| IMPDRV\_IMP\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_IMP\_EXEPARAM\_GOSUB | (1U) | Index value of extend parameter for conditional GOSUB instruction. |
| IMPDRV\_IMP\_INTSTS\_CLBRK | (0x00000400U) | CLBRK interrupt enable registration. |
| IMPDRV\_REG\_IMP\_CLCNDGSBR | (0x00C4U) | Offset address of CLCNDGSBR register for IMP. |
| IMPDRV\_OCV\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_OCV\_EXEPARAM\_GOSUB | (1U) | Index value of extend parameter for conditional GOSUB instruction. |
| IMPDRV\_REG\_OCV\_CLCNDGSBR | (0x03FCU) | Offset address of CLCNDGSBR register for OCV. |
| IMPDRV\_PSC\_CLCNDGSBR\_MASK | (0x00000001U) | Bit mask for Condition for conditional GOSUB instruction. |
| IMPDRV\_PSC\_EXEPARAM\_GOSUB | (1U) | Index value of extend parameter for conditional GOSUB instruction. |
| IMPDRV\_PSC\_SR\_CLBRK | (0x00000400U) | CLBRK interrupt enable registration. |
| IMPDRV\_REG\_PSC\_CLBRKADDRR | (0x03F4U) | Offset address of CLBRKADDRR register for PSC. |
| IMPDRV\_REG\_PSC\_CLCNDGSBR | (0x03FCU) | Offset address of CLCNDGSBR register for PSC. |
| IMPDRV\_CRC\_TABLE\_MAX | (256) | table num for crc rable. |
| IMPDRV\_DSP\_CORENUM\_VAL | (2U) | Maximum number of cores for IMP. |
| IMPDRV\_DSP\_VDSP\_DEV\_MAIN | (0u) | Interrupt |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB7 | (8u) | VDSP device sub7. |
| IMPDRV\_REGVAL\_VDSP\_TRAP | (0x00010000U) | GPIO\_OUT register INT value for VDSP. |
| IMPDRV\_REGVAL\_VDSP\_VERSION1 | (0x20211220U) | Hardware version value for VDSP. |
| IMPDRV\_SPMX\_ADDR\_START | (0xED200000U) | D\_ADD3\_START register REGION\_START value for VDSP. |

#### OS Dependence Layer

Table 5‑55: Definition Values (OS Dependence Layer)

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| IMPDRV\_IMPTOP\_DEV\_TYPE | (“imp\_top”) | Static OSAL resource definition for “imp\_top”. |
| IMPDRV\_DEV\_LIST\_SIZE | (255U) | Buffer size for R\_OSAL\_IoGetDeviceList. |
| IMPDRV\_REG\_OFFSET\_MAX | (1024U \* 256U \* 8U) | Maximum value for Register offset. |
| IMPDRV\_IRQ\_CHANNEL\_0 | (0U) | IRQ channel number (0) |
| IMPDRV\_IRQ\_CHANNEL\_1 | (1U) | IRQ channel number (1) |
| IMPDRV\_IRQ\_CHANNEL\_2 | (2U) | IRQ channel number (2) |
| IMPDRV\_IRQ\_CHANNEL\_3 | (3U) | IRQ channel number (3) |
| IMPDRV\_IRQ\_CHANNEL\_4 | (4U) | IRQ channel number (4) |
| IMPDRV\_IRQ\_CHANNEL\_5 | (5U) | IRQ channel number (5) |
| IMPDRV\_IRQ\_CHANNEL\_6 | (6U) | IRQ channel number (6) |
| IMPDRV\_IRQ\_CHANNEL\_MAX | (7U) | Maximum value for IRQ channel |
| IMPDRV\_DUMMY\_UINT64 | (0xFFFFFFFFFFFFFFFFULL) | Uint64\_t dummy value. |
| IMPDRV\_QUEUE\_MAX\_NUMBER | (2U) | Maximum number of message queues |
| IMPDRV\_QUEUE\_MAX\_MSG\_SIZE | (sizeof(uint32\_t)) | Message size from interrupt handler |
| IMPDRV\_DSP\_CORE\_0 | (0U) | DSP core number (0). |
| IMPDRV\_DSP\_CORE\_1 | (1U) | DSP core number (1). |
| IMPDRV\_DSP\_CORE\_2 | (2U) | DSP core number (2). |
| IMPDRV\_DSP\_CORE\_3 | (3U) | DSP core number (3). |
| IMPDRV\_DSP\_VDSP\_DEV\_SUB1 | (2U) | num of sub device for vsdp\*\_reg. |

### Enum

#### API Layer

##### e\_impdrv\_errorcode\_t

Table 5‑56: e\_impdrv\_errorcode\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_EC\_OK = 0,  IMPDRV\_EC\_NG\_CHECKFAIL = -200,  IMPDRV\_EC\_NG\_SEQSTATE = -201,  IMPDRV\_EC\_NG\_SYSTEMERROR = -205,  IMPDRV\_EC\_NG\_ARGNULL = -207,  IMPDRV\_EC\_NG\_PARAM = -208,  IMPDRV\_EC\_NG\_ATTRIBUTE = -209,  IMPDRV\_EC\_NG\_NOTSUPPORT = -210,  IMPDRV\_EC\_NG\_PMSYSTEMERROR = -211,  IMPDRV\_EC\_NG\_INSTANCE = -212,  IMPDRV\_EC\_NG\_DSP\_HALT = -213  } e\_impdrv\_errorcode\_t; | | |
| **Description** | IMP Driver notifies the user of an error by this result value. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_EC\_OK | 0 | Successful completion. |
| IMPDRV\_EC\_NG\_CHECKFAIL | -200 | This error is caused by the safety mechanism check fails. |
| IMPDRV\_EC\_NG\_SEQSTATE | -201 | Sequence Error.  This error occurs by illegal state transition.  In this case, modify the function order to call the function from the correct state machine. |
| IMPDRV\_EC\_NG\_SYSTEMERROR | -205 | System error.  This error occurs by hardware error in IMP-X6 or OSAL function error.  In this case, execute system reset or reconsider the OSAL settings and try again. |
| IMPDRV\_EC\_NG\_ARGNULL | -207 | Arguments is NULL.  This error occurs by setting the argument to NULL.  In this case, the user can continue by put the correct pointer to the argument. |
| IMPDRV\_EC\_NG\_PARAM | -208 | Parameter error.  This error occurs by specifying an invalid parameter.  In this case, the user can continue by put the correct value to the argument. |
| IMPDRV\_EC\_NG\_ATTRIBUTE | -209 | This error is caused by not accurate setting the attribute. In this case, the user can continue by setting the attributes required by the following functions:  ・R\_IMPDRV\_Start  ・R\_IMPDRV\_Execute |
| IMPDRV\_EC\_NG\_NOTSUPPORT | -210 | Not supported parameter error.  Occurs when you specify a core type or core number that hardware does not support.  In this case, the user can continue by put the correct value to the argument. |
| IMPDRV\_EC\_NG\_PMSYSTEMERROR | -211 | PMSystem error.  This error occurs by OSAL function (only Power Management function) error.  In this case, execute system reset or reconsider the OSAL settings and try again. |
| IMPDRV\_EC\_NG\_INSTANCE | -212 | Instance parameter error.  Checking for errors between instances. |
| IMPDRV\_EC\_NG\_DSP\_HALT | -213 | VDSP error notice |
| **Remarks** | N/A | | |

##### e\_impdrv\_core\_type\_t

Table 5‑57: e\_impdrv\_core\_type\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_CORE\_TYPE\_INVALID = 0,  IMPDRV\_CORE\_TYPE\_IMP ,  IMPDRV\_CORE\_TYPE\_IMP\_SLIM ,  IMPDRV\_CORE\_TYPE\_OCV ,  IMPDRV\_CORE\_TYPE\_DMAC ,  IMPDRV\_CORE\_TYPE\_DMAC\_SLIM ,  IMPDRV\_CORE\_TYPE\_PSCEXE ,  IMPDRV\_CORE\_TYPE\_PSCOUT ,  IMPDRV\_CORE\_TYPE\_CNN ,  IMPDRV\_CORE\_TYPE\_DSP ,  IMPDRV\_CORE\_TYPE\_DTA ,  IMPDRV\_CORE\_TYPE\_IMR = 1000,  IMPDRV\_CORE\_TYPE\_LDMAC  } e\_impdrv\_core\_type\_t; | | |
| **Description** | IMP Driver determines the core type, selected by this value. | | |
| **Enumerato**  **r** | **Name** | **Value** | **Description** |
| IMPDRV\_CORE\_TYPE\_INVALID | 0 | Core type number of Invalid value. |
| IMPDRV\_CORE\_TYPE\_IMP | 1 | Core type number of IMP core. |
| IMPDRV\_CORE\_TYPE\_IMP\_SLIM | 2 | Core type number of Slim-IMP core. |
| IMPDRV\_CORE\_TYPE\_OCV | 3 | Core type number of OCV core. |
| IMPDRV\_CORE\_TYPE\_DMAC | 4 | Core type number of IMP DMAC. |
| IMPDRV\_CORE\_TYPE\_DMAC\_SLIM | 5 | Core type number of Slim-DMAC core. |
| IMPDRV\_CORE\_TYPE\_PSCEXE | 6 | Core type number of IMP PSC. This type can be specified only in “core” parameter for APIs. |
| IMPDRV\_CORE\_TYPE\_PSCOUT | 7 | Core type number of IMP PSC output. This type can be specified only in “coremap” parameter for R\_IMPDRV\_Start. Refer the note of Core map definition for detail. |
| IMPDRV\_CORE\_TYPE\_CNN | 8 | Core type number of IMP CNN. |
| IMPDRV\_CORE\_TYPE\_DSP | 9 | Core type number of DSP. |
| IMPDRV\_CORE\_TYPE\_DTA | 10 | Core type number of DTA. |
| IMPDRV\_CORE\_TYPE\_IMR | 1000 | Core type number of IMR. (Other than IMP-Xn) |
| IMPDRV\_CORE\_TYPE\_LDMAC | 1001 | Core type number of Lock Step DMAC core |
| **Remarks** | N/A | | |

##### e\_impdrv\_instance\_t

Table 5‑58: e\_impdrv\_instance\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_INSTANCE\_INVALID = 0  IMPDRV\_INSTANCE\_0 ,  IMPDRV\_INSTANCE\_1 ,  IMPDRV\_INSTANCE\_2 ,  IMPDRV\_INSTANCE\_3 ,  IMPDRV\_INSTANCE\_4 ,  IMPDRV\_INSTANCE\_5 ,  IMPDRV\_INSTANCE\_6  } e\_impdrv\_instance\_t; | | |
| **Description** | IMP Driver determines the Instance number(with related IRQ number), selected by this value. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_INSTANCE\_INVALID | 0 | Invalid value |
| IMPDRV\_INSTANCE\_0 | 1 | Instance number (0) SPI 576 Domain0 Interrupt |
| IMPDRV\_INSTANCE\_1 | 2 | Instance number (1) SPI 577 Domain0 Interrupt |
| IMPDRV\_INSTANCE\_2 | 3 | Instance number (2) SPI 578 Domain0 Interrupt |
| IMPDRV\_INSTANCE\_3 | 4 | Instance number (3) SPI 579 Domain1 Interrupt |
| IMPDRV\_INSTANCE\_4 | 5 | Instance number (4) SPI 580 Domain1 Interrupt |
| IMPDRV\_INSTANCE\_5 | 6 | Instance number (5) SPI 581 Domain1 Interrupt |
| IMPDRV\_INSTANCE\_6 | 7 | Instance number (6) SPI 582 Merge Interrupt |
| **Remarks** | N/A | | |

##### e\_impdrv\_irq\_group\_t

Table 5‑59: e\_impdrv\_irq\_group\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_IRQ\_GROUP\_INVALID = 0,  IMPDRV\_IRQ\_GROUP\_NONE ,  IMPDRV\_IRQ\_GROUP\_0 ,  IMPDRV\_IRQ\_GROUP\_1 ,  IMPDRV\_IRQ\_GROUP\_2  } e\_impdrv\_irq\_group\_t; | | |
| **Description** | IMP Driver determines the Interrupt group of st\_impdrv\_irq\_group\_t, selected by this value. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_IRQ\_GROUP\_INVALID | 0 | Invalid value |
| IMPDRV\_IRQ\_GROUP\_NONE | 1 | No group settings. |
| IMPDRV\_IRQ\_GROUP\_0 | 2 | IRQ group 0. |
| IMPDRV\_IRQ\_GROUP\_1 | 3 | IRQ group 1. |
| IMPDRV\_IRQ\_GROUP\_2 | 4 | IRQ group 2. |
| **Remarks** | N/A | | |

##### e\_impdrv\_cb\_ret\_t

Table 5‑60: e\_impdrv\_cb\_ret\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_CB\_RET\_OK = 0,  IMPDRV\_CB\_RET\_BEFORE\_EXEC ,  IMPDRV\_CB\_RET\_RESERVED ,  IMPDRV\_CB\_RET\_ILLEGAL ,  IMPDRV\_CB\_RET\_INT ,  IMPDRV\_CB\_RET\_CLBRK ,  IMPDRV\_CB\_RET\_PBCOVF ,  IMPDRV\_CB\_RET\_INT\_PBCOVF ,  IMPDRV\_CB\_RET\_USIER ,  IMPDRV\_CB\_RET\_SBO0ME ,  IMPDRV\_CB\_RET\_INT\_SBO0ME ,  IMPDRV\_CB\_RET\_TRAP\_SBO0ME ,  IMPDRV\_CB\_RET\_WUPCOVF ,  IMPDRV\_CB\_RET\_HPINT ,  IMPDRV\_CB\_RET\_APIPINT ,  IMPDRV\_CB\_RET\_USINT ,  IMPDRV\_CB\_RET\_END ,  IMPDRV\_CB\_RET\_MSCO ,  IMPDRV\_CB\_RET\_UDIVSBRK ,  IMPDRV\_CB\_RET\_UDIPSBRK ,  IMPDRV\_CB\_RET\_DSP\_HALT  } e\_impdrv\_cb\_ret\_t; | | |
| **Description** | IMP driver notifies the user of the callback results by this result value. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_CB\_RET\_OK | 0 | Arguments of callback function. (Normal) |
| IMPDRV\_CB\_RET\_BEFORE\_EXEC | 1 | Arguments of callback function. (Before CL execute) |
| IMPDRV\_CB\_RET\_RESERVED | 2 | Reserved definition. |
| IMPDRV\_CB\_RET\_ILLEGAL | 3 | Arguments of callback function. (Illegal error) |
| IMPDRV\_CB\_RET\_INT | 4 | Arguments of callback function. (INT interrupt) |
| IMPDRV\_CB\_RET\_CLBRK | 5 | CLBRK interrupt |
| IMPDRV\_CB\_RET\_PBCOVF | 6 | Arguments of callback function. (In the case of “Performance Busy Counter Overflow” ) |
| IMPDRV\_CB\_RET\_INT\_PBCOVF | 7 | Arguments of callback function. (In the case of ”Performance Busy Counter Overflow” and “INT interrupt”.) |
| IMPDRV\_CB\_RET\_USIER | 8 | Arguments of callback function. (In the case of “USIER” error.) |
| IMPDRV\_CB\_RET\_SBO0ME | 9 | Arguments of callback function.( In the case of “SBO0ME) |
| IMPDRV\_CB\_RET\_INT\_SBO0ME | 10 | Arguments of callback function. (In the case of ” SBO0ME” and “INT interrupt”.) |
| IMPDRV\_CB\_RET\_TRAP\_SBO0ME | 11 | Arguments of callback function. (In the case of ” SBO0ME” and “ TRAP interrupt”.) |
| IMPDRV\_CB\_RET\_WUPCOVF | 12 | In the case of “WUPCOVF” error |
| IMPDRV\_CB\_RET\_HPINT | 13 | In the case of “HPINT interrupt” |
| IMPDRV\_CB\_RET\_APIPINT | 14 | In the case of “APIPINT interrupt” |
| IMPDRV\_CB\_RET\_USINT | 15 | In the case of “USINT interrupt” |
| IMPDRV\_CB\_RET\_END | 16 | In the case of “END interrupt” |
| IMPDRV\_CB\_RET\_MSCO | 17 | In the case of “MSCO interrupt” |
| IMPDRV\_CB\_RET\_UDIVSBRK | 18 | In the case of “UDIVSBRK interrupt” |
| IMPDRV\_CB\_RET\_UDIPSBRK | 19 | In the case of “UDIPSBRK interrupt” |
| IMPDRV\_CB\_RET\_DSP\_HALT | 20 | In the case of "VDSP error notice" |
| **Remarks** | N/A | | |

##### e\_impdrv\_pm\_policy\_t

Table 5‑61: e\_impdrv\_pm\_policy\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_PM\_POLICY\_INVALID = 0,  IMPDRV\_PM\_POLICY\_CG ,  IMPDRV\_PM\_POLICY\_PG ,  IMPDRV\_PM\_POLICY\_HP  } e\_impdrv\_pm\_policy\_t; | | |
| **Description** | Power management policy. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_PM\_POLICY\_INVALID | 0 | Invalid value |
| IMPDRV\_PM\_POLICY\_CG | 1 | Specify the “Power Gated Policy” of power management, “Clock domain OFF / Power domain ON” when the core is stopped. |
| IMPDRV\_PM\_POLICY\_PG | 2 | Specify the “Clock Gated Policy” of power management, “Clock domain OFF / Power domain OFF” when the core is stopped. |
| IMPDRV\_PM\_POLICY\_HP | 3 | Specify the “High Performance Policy” of power management, “Clock domain ON / Power domain ON” when the core is stopped. |
| **Remarks** | N/A | | |

##### e\_impdrv\_param\_t

Table 5‑62: e\_impdrv\_param\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_PARAM\_INVALID = 0,  IMPDRV\_PARAM\_OFF,  IMPDRV\_PARAM\_ON  } e\_impdrv\_param\_t; | | |
| **Description** | Generic enum for parameters ON and OFF. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_PARAM\_INVALID | 0 | Invalid value |
| IMPDRV\_PARAM\_OFF | 1 | Parameter OFF. |
| IMPDRV\_PARAM\_ON | 2 | Parameter ON. |
| **Remarks** | N/A | | |

##### e\_impdrv\_irqmask\_t

Table 5‑63: e\_impdrv\_irqmask\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_IRQMASK\_END = 0,  IMPDRV\_IRQMASK\_WUPCOVF,  IMPDRV\_IRQMASK\_USIER,  IMPDRV\_IRQMASK\_USINT,  IMPDRV\_IRQMASK\_PBCOVF,  IMPDRV\_IRQMASK\_SBO0ME,  IMPDRV\_IRQMASK\_HPINT,  IMPDRV\_IRQMASK\_APIPINT,  IMPDRV\_IRQMASK\_MSCO  } e\_impdrv\_irqmask\_t; | | |
| **Description** | IMPDRV interrupt mask | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_IRQMASK\_END | 0 | Disable to interrupt for END |
| IMPDRV\_IRQMASK\_WUPCOVF | 1 | Disable to interrupt for WUPCOVF |
| IMPDRV\_IRQMASK\_USIER | 2 | Disable to interrupt for USIER |
| IMPDRV\_IRQMASK\_USINT | 3 | Disable to interrupt for USINT |
| IMPDRV\_IRQMASK\_PBCOVF | 4 | Disable to interrupt for PBCOVF |
| IMPDRV\_IRQMASK\_SBO0ME | 5 | Disable to interrupt for SBO0ME |
| IMPDRV\_IRQMASK\_HPINT | 6 | Disable to interrupt for HPINT |
| IMPDRV\_IRQMASK\_APIPINT | 7 | Disable to interrupt for APIPINT |
| IMPDRV\_IRQMASK\_MSCO | 8 | Disable to interrupt for MSCO |
| **Remarks** | N/A | | |

##### e\_impdrv\_fatalcode\_t

Table 5‑64: e\_impdrv\_fatalcode\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_FC\_DRV\_ERROR = 0,  IMPDRV\_FC\_UNEXPECT\_INT  } e\_impdrv\_fatalcode\_t; | | |
| **Description** | IMPDRV fatal error code. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_FC\_DRV\_ERROR | 0 | Driver Internal Error. |
| IMPDRV\_FC\_UNEXPECT\_INT | 1 | Unexpected interrupt. |
| **Remarks** | N/A | | |

##### e\_impdrv\_protect\_mode\_t

Table 5‑65: e\_impdrv\_protect\_mode\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_PROTECT\_INVALID = 0,  IMPDRV\_PROTECT\_EDC,  IMPDRV\_PROTECT\_ECC  } e\_impdrv\_protect\_mode\_t;; | | |
| **Description** | IMPDRV Select of Memory protection mode (EDC/ECC). | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_PROTECT\_INVALID | 0 | Invalid value |
| IMPDRV\_PROTECT\_EDC | 1 | EDC mode |
| IMPDRV\_PROTECT\_ECC | 2 | ECC mode |
| **Remarks** | N/A | | |

##### e\_impdrv\_reg\_req\_state\_t

Table 5‑66: e\_impdrv\_reg\_req\_state\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_REG\_REQ\_STATE\_INVALID = 0,  IMPDRV\_REG\_REQ\_STATE\_REQUIRED,  IMPDRV\_REG\_REQ\_STATE\_RELEASED  } e\_impdrv\_reg\_req\_state\_t; | | |
| **Description** | IMPDRV Required state to the Register read/write function. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_REG\_REQ\_STATE\_INVALID | 0 | Invalid state |
| IMPDRV\_REG\_REQ\_STATE\_REQUIRED | 1 | Required state |
| IMPDRV\_REG\_REQ\_STATE\_RELEASED | 2 | Release state |
| **Remarks** | N/A | | |

##### e\_impdrv\_pm\_state\_t

Table 5‑67: e\_impdrv\_pm\_state\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum {  IMPDRV\_PM\_STATE\_INVALID = 0,  IMPDRV\_PM\_STATE\_PG = 1,  IMPDRV\_PM\_STATE\_CG = 2,  IMPDRV\_PM\_STATE\_HP = 4,  IMPDRV\_PM\_STATE\_RESET = 8,  IMPDRV\_PM\_STATE\_READY = 16  } e\_impdrv\_pm\_state\_t; | | |
| **Description** | IMPDRV Power management state. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_PM\_STATE\_INVALID | 0 | Invalid state |
| IMPDRV\_PM\_STATE\_PG | 1 | “Power Gated Policy” for power management, and “Clock domain OFF / Power domain OFF” when the core is stopped. |
| IMPDRV\_PM\_STATE\_CG | 2 | “Clock Gated Policy” for power management, and “Clock domain OFF / Power domain ON” when the core is stopped. |
| IMPDRV\_PM\_STATE\_HP | 4 | “High Performance Policy” for power management, and “Clock domain ON / Power domain ON” when the core is stopped. |
| IMPDRV\_PM\_STATE\_RESET | 8 | “Reset state” for power management, and “Clock domain ON / Power domain ON” when the core is in reset state. |
| IMPDRV\_PM\_STATE\_READY | 16 | “Ready state” for power management, and “Clock domain ON / Power domain ON” when the core is not in reset status. |
| **Remarks** | N/A | | |

##### e\_impdrv\_gosub\_cond\_t

Table 5‑67: e\_impdrv\_gosub\_cond\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum {  IMPDRV\_GOSUB\_COND\_INVALID = 0,  IMPDRV\_GOSUB\_COND\_NOTEXEC,  IMPDRV\_GOSUB\_COND\_EXEC  } e\_impdrv\_gosub\_cond\_t; | | |
| **Description** | IMPDRV Specifies the condition in case of a conditional GOSUB instruction. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_GOSUB\_COND\_INVALID | 0 | Invalid value |
| IMPDRV\_GOSUB\_COND\_NOTEXEC | 1 | GOSUB is not executed, NOP is instead executed |
| IMPDRV\_GOSUB\_COND\_EXEC | 2 | GOSUB is executed |
| **Remarks** | N/A | | |

#### General Control

None.

#### Common Control

##### e\_impdrv\_mb\_mem\_init\_t

Table 5‑68: e\_impdrv\_mb\_mem\_init\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_MB\_MBM\_INIT\_ODD,  IMPDRV\_MB\_MBM\_INIT\_EVEN,  } e\_impdrv\_mb\_mem\_init\_t; | | |
| **Description** | IMPDRV Multi-Bank memory initialization kind. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_MB\_MBM\_INIT\_ODD | 0 | I Multi-Bank memory odd initialization |
| IMPDRV\_MB\_MBM\_INIT\_EVEN | 1 | Multi-Bank memory even initialization |
| **Remarks** | N/A | | |

#### Core Control

##### e\_impdrv\_state\_t

Table 5‑69: e\_impdrv\_state\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_STATE\_UNINIT = 0,  IMPDRV\_STATE\_INIT ,  IMPDRV\_STATE\_READY ,  IMPDRV\_STATE\_EXEC ,  IMPDRV\_STATE\_INT  } e\_impdrv\_state\_t; | | |
| **Description** | IMP Driver determines the Finite-State machine by this value.  For details, refer to the “3.1 Finite-State machine” in [3]. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_STATE\_UNINIT | 0 | Uninitialized state |
| IMPDRV\_STATE\_INIT | 1 | Initialization state |
| IMPDRV\_STATE\_READY | 2 | Ready state |
| IMPDRV\_STATE\_EXEC | 3 | Execution state |
| IMPDRV\_STATE\_INT | 4 | Interrupting state |
| **Remarks** | N/A | | |

##### e\_impdrv\_dspctl\_load\_type\_t

Table 5‑69: e\_impdrv\_dspctl\_load\_type\_t

|  |  |  |  |
| --- | --- | --- | --- |
| **Format** | typedef enum  {  IMPDRV\_VDSP\_LOAD\_TYPE\_INT = (0),  IMPDRV\_VDSP\_LOAD\_TYPE\_EXT  }e\_impdrv\_dspctl\_load\_type\_t; | | |
| **Description** | VDSP load tyoe. | | |
| **Enumerator** | **Name** | **Value** | **Description** |
| IMPDRV\_VDSP\_LOAD\_TYPE\_INT | 0 | Load type INT |
| IMPDRV\_VDSP\_LOAD\_TYPE\_EXT | 1 | Load type EXT |
| **Remarks** | N/A | | |

#### OS Dependence Layer

None.

### Constants Table

#### API Layer

None.

#### General Control

##### s\_impdrv\_core\_type\_table

Table 5‑70: s\_impdrv\_core\_type\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const e\_impdrv\_core\_type\_t s\_impdrv\_core\_type\_table[IMPDRV\_CORE\_TYPE\_MAX – 1U] =  {  /\* core\_type \*/  IMPDRV\_CORE\_TYPE\_IMP, /\*\*< IMP core \*/  IMPDRV\_CORE\_TYPE\_OCV, /\*\*< OCV core \*/  IMPDRV\_CORE\_TYPE\_DMAC, /\*\*< DMAC core \*/  IMPDRV\_CORE\_TYPE\_PSCEXE, /\*\*< PSC(exe) core \*/  IMPDRV\_CORE\_TYPE\_PSCOUT, /\*\*< PSC(out) core \*/  IMPDRV\_CORE\_TYPE\_CNN, /\*\*< CNN core \*/  IMPDRV\_CORE\_TYPE\_LDMAC, /\*\*< Lock Step DMAC core \*/  IMPDRV\_CORE\_TYPE\_DTA /\*\*< Debug Trace Agent \*/  }; |
| **Description** | IMP Driver uses this table to convert sequential numbers to core type enumerated values. |
| **Remarks** | N/A |

Table 5‑71: s\_impdrv\_core\_type\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const e\_impdrv\_core\_type\_t s\_impdrv\_core\_type\_table[IMPDRV\_CORE\_TYPE\_MAX – 1U] =  {  /\*\* core\_type \*/  IMPDRV\_CORE\_TYPE\_IMP , /\*\*< IMP core\*/  IMPDRV\_CORE\_TYPE\_IMP\_SLIM , /\*\*< Slim-IMP core\*/  IMPDRV\_CORE\_TYPE\_OCV , /\*\*< OCV core\*/  IMPDRV\_CORE\_TYPE\_DMAC , /\*\*< DMAC core\*/  IMPDRV\_CORE\_TYPE\_DMAC\_SLIM , /\*\*< Slim-DMAC core\*/  IMPDRV\_CORE\_TYPE\_PSCEXE , /\*\*< PSC(exe) core\*/  IMPDRV\_CORE\_TYPE\_PSCOUT , /\*\*< PSC(out) core\*/  IMPDRV\_CORE\_TYPE\_CNN , /\*\*< CNN core\*/  IMPDRV\_CORE\_TYPE\_DSP , /\*\*< DSP core\*/  IMPDRV\_CORE\_TYPE\_DTA /\*\*< Debug Trace Agent\*/  }; |
| **Description** | IMP Driver uses this table to convert sequential numbers to core type enumerated values. |
| **Remarks** | N/A |

#### Common Control

##### s\_impdrv\_inten\_addr

Table 5‑72: s\_impdrv\_inten\_addr

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const uint32\_t s\_impdrv\_inten\_addr[IMPDRV\_INSTANCETABLE\_NUM] =  {  0xFFFFFFFFU, /\*\*< Dummy for Index adjustment \*/  IMPDRV\_REG\_INTEN00, /\*\* IMPDRV\_INSTANCE\_0 \*/  IMPDRV\_REG\_INTEN01, /\*\* IMPDRV\_INSTANCE\_1 \*/  IMPDRV\_REG\_INTEN02, /\*\* IMPDRV\_INSTANCE\_2 \*/  IMPDRV\_REG\_INTEN10, /\*\* IMPDRV\_INSTANCE\_3 \*/  IMPDRV\_REG\_INTEN11, /\*\* IMPDRV\_INSTANCE\_4 \*/  IMPDRV\_REG\_INTEN12, /\*\* IMPDRV\_INSTANCE\_5 \*/  IMPDRV\_REG\_MINTEN /\*\* IMPDRV\_INSTANCE\_6 \*/  }; |
| **Description** | IMP Driver determines the Offset address of Interrupt enable register by this table.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten0\_cnv\_table

Table 5‑73: s\_impdrv\_inten0\_cnv\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_inten0\_cnv\_table[IMPDRV\_INTEN0\_CORE\_MAX] =  {  /\*\* coreType, CoreNum, bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 1, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_OCV, 0, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_OCV, 1, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_OCV, 4, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_OCV, 6, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_CNN, 0, 0x00000200U },  { IMPDRV\_CORE\_TYPE\_CNN, 2, 0x00000400U }  }; |
| **Description** | IMP Driver determines the Bit values of Interrupt enable register by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “INTEN00”, “INTEN01” and “INTEN02” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten0\_group\_table

Table 5‑74: s\_impdrv\_inten0\_group\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_group\_tbl\_t s\_impdrv\_inten0\_group\_table[IMPDRV\_GROUP\_MAX] =  {  /\*\* group\_bit group\_reg \*/  { 0x00000800U, IMPDRV\_REG\_G00INTSEL },  { 0x00001000U, IMPDRV\_REG\_G01INTSEL },  { 0x00002000U, IMPDRV\_REG\_G02INTSEL }  }; |
| **Description** | IMP Driver determines the IRQ grouping values of Interrupt enable register by this table.  This table provides cross-conversion between group bits and group registers.  The settings in this table apply to the “INTEN00”, “INTEN01” and “INTEN02” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten1\_cnv\_table

Table 5‑75: s\_impdrv\_inten1\_cnv\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_inten1\_cnv\_table[IMPDRV\_INTEN1\_CORE\_MAX] =  {  /\*\* coreType, CoreNum bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 2, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 3, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 1, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_OCV, 2, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_OCV, 3, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_OCV, 5, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_OCV, 7, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_CNN, 1, 0x00000400U }  }; |
| **Description** | IMP Driver determines the Bit values of Interrupt enable register by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “INTEN10”, “INTEN11” and “INTEN12” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten1\_group\_table

Table 5‑76: s\_impdrv\_inten1\_group\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_group\_tbl\_t s\_impdrv\_inten1\_group\_table[IMPDRV\_GROUP\_MAX] =  {  /\*\* group\_bit group\_reg \*/  { 0x00000800U, IMPDRV\_REG\_G10INTSEL },  { 0x00001000U, IMPDRV\_REG\_G11INTSEL },  { 0x00002000U, IMPDRV\_REG\_G12INTSEL }  }; |
| **Description** | IMP Driver determines the IRQ grouping values of Interrupt enable register by this table.  This table provides cross-conversion between group bits and group registers.  The settings in this table apply to the “INTEN10”, “INTEN11” and “INTEN12” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_minten\_cnv\_table

Table 5‑77: s\_impdrv\_minten\_cnv\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_minten\_cnv\_table[IMPDRV\_MINTEN\_CORE\_MAX] =  {  /\*\* coreType, CoreNum bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 1, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_OCV, 0, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_OCV, 1, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_OCV, 4, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_OCV, 6, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_CNN, 0, 0x00000200U },  { IMPDRV\_CORE\_TYPE\_CNN, 2, 0x00000400U },  { IMPDRV\_CORE\_TYPE\_IMP, 2, 0x00000800U },  { IMPDRV\_CORE\_TYPE\_IMP, 3, 0x00001000U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 1, 0x00002000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2, 0x00004000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3, 0x00008000U },  { IMPDRV\_CORE\_TYPE\_OCV, 2, 0x00010000U },  { IMPDRV\_CORE\_TYPE\_OCV, 3, 0x00020000U },  { IMPDRV\_CORE\_TYPE\_OCV, 5, 0x00040000U },  { IMPDRV\_CORE\_TYPE\_OCV, 7, 0x00080000U },  { IMPDRV\_CORE\_TYPE\_CNN, 1, 0x00200000U }  }; |
| **Description** | IMP Driver determines the Bit values of Interrupt enable register by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “MINTEN” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_minten\_group\_table

Table 5‑78: s\_impdrv\_minten\_group\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_group\_tbl\_t s\_impdrv\_minten\_group\_table[IMPDRV\_GROUP\_MAX] =  {  /\*\* group\_bit group\_reg \*/  { 0x00400000U, IMPDRV\_REG\_MG0INTSEL },  { 0x00800000U, IMPDRV\_REG\_MG1INTSEL },  { 0x01000000U, IMPDRV\_REG\_MG2INTSEL }  }; |
| **Description** | IMP Driver determines the IRQ grouping values of Interrupt enable register by this table.  This table provides cross-conversion between group bits and group registers.  The settings in this table apply to the “MINTEN” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten\_addr

Table 5‑79: s\_impdrv\_inten\_addr

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const uint32\_t s\_impdrv\_inten\_addr[IMPDRV\_INSTANCETABLE\_NUM] =  {  IMPDRV\_DUMMY\_UINT32, /\*\*< Dummy for Index adjustment \*/  IMPDRV\_REG\_IMR, /\*\* IMPDRV\_INSTANCE\_0 \*/  IMPDRV\_REG\_IMR1, /\*\* IMPDRV\_INSTANCE\_1 \*/  IMPDRV\_REG\_IMR2, /\*\* IMPDRV\_INSTANCE\_2 \*/  IMPDRV\_REG\_IMR3, /\*\* IMPDRV\_INSTANCE\_3 \*/  IMPDRV\_REG\_IMR4, /\*\* IMPDRV\_INSTANCE\_4 \*/  }; |
| **Description** | IMP Driver determines the Offset address of Interrupt enable register by this table.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_impdrv\_inten\_cnv\_table

Table 5‑80: s\_impdrv\_inten\_cnv\_table

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M]**  IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_inten\_cnv\_table[IMPDRV\_INTEN\_CORE\_MAX] =  {  /\*\* coreType, CoreNum, bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 1, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_IMP, 2, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_IMP, 3, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_OCV, 0, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_OCV, 1, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_CNN, 0, 0x00000200U },  };  **[R-CarV3H]**  IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_inten\_cnv\_table[IMPDRV\_INTEN\_CORE\_MAX] =  {  /\*\* coreType, CoreNum, bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 1, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_IMP, 2, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_IMP, 3, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_IMP, 4, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_IMP\_SLIM, 0, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_OCV, 0, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_OCV, 1, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_OCV, 2, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_OCV, 3, 0x00000200U },  { IMPDRV\_CORE\_TYPE\_OCV, 4, 0x00000400U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0, 0x00000800U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1, 0x00001000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2, 0x00002000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3, 0x00004000U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0, 0x00008000U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 1, 0x00010000U },  { IMPDRV\_CORE\_TYPE\_CNN, 0, 0x00020000U }  };  **[R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_inten\_tbl\_t s\_impdrv\_inten\_cnv\_table[IMPDRV\_INTEN\_CORE\_MAX] =  {  /\*\* coreType, CoreNum, bit \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0, 0x00000001U },  { IMPDRV\_CORE\_TYPE\_IMP, 1, 0x00000002U },  { IMPDRV\_CORE\_TYPE\_IMP, 2, 0x00000004U },  { IMPDRV\_CORE\_TYPE\_IMP, 3, 0x00000008U },  { IMPDRV\_CORE\_TYPE\_IMP, 4, 0x00000010U },  { IMPDRV\_CORE\_TYPE\_IMP\_SLIM, 0, 0x00000020U },  { IMPDRV\_CORE\_TYPE\_OCV, 0, 0x00000040U },  { IMPDRV\_CORE\_TYPE\_OCV, 1, 0x00000080U },  { IMPDRV\_CORE\_TYPE\_OCV, 2, 0x00000100U },  { IMPDRV\_CORE\_TYPE\_OCV, 3, 0x00000200U },  { IMPDRV\_CORE\_TYPE\_OCV, 4, 0x00000400U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0, 0x00000800U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1, 0x00001000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2, 0x00002000U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3, 0x00004000U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 0, 0x00008000U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 1, 0x00010000U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0, 0x00020000U },  { IMPDRV\_CORE\_TYPE\_CNN, 0, 0x00040000U }  };  [R-CarV4H]  Not Defined |
| **Description** | IMP Driver determines the Bit values of Interrupt enable register by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “IMR”, “IMR1”, “IMR2”, “IMR3” and , “IMR4” register.  Currently, it only uses one place, so it is defined in the function (Coverity pointed correspondence). Planned to be defined outside the function in the future. |
| **Remarks** | N/A |

##### s\_corectl\_func\_tbl

Table 5‑81: s\_corectl\_func\_tbl

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_corectl\_func\_t s\_corectl\_func\_tbl[IMPDRV\_CORE\_TYPE\_MAX] =  {  /\*\* Not Assigned \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_IMP \*/  { impdrv\_impctl\_is\_valid\_core, impdrv\_impctl\_check\_state,  impdrv\_impctl\_set\_state, impdrv\_impctl\_init\_start,  impdrv\_impctl\_init\_end, impdrv\_impctl\_start,  impdrv\_impctl\_stop, impdrv\_impctl\_attr\_init,  impdrv\_impctl\_set\_mem\_init, impdrv\_impctl\_set\_core\_map,  impdrv\_impctl\_set\_cl, impdrv\_impctl\_set\_irq\_mask,  impdrv\_impctl\_execute, impdrv\_impctl\_int\_handler, impdrv\_impctl\_resume\_exe, impdrv\_impctl\_pm\_set\_policy, impdrv\_impctl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_impctl\_set\_cl\_brk\_addr, impdrv\_impctl\_set\_cond\_gosub,  impdrv\_impctl\_pm\_get\_state, impdrv\_impctl\_quit  },  /\* IMPDRV\_CORE\_TYPE\_IMP\_SLIM \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_OCV \*/  { impdrv\_ocvctl\_is\_valid\_core, impdrv\_ocvctl\_check\_state,  impdrv\_ocvctl\_set\_state, impdrv\_ocvctl\_init\_start,  impdrv\_ocvctl\_init\_end, impdrv\_ocvctl\_start,  impdrv\_ocvctl\_stop, impdrv\_ocvctl\_attr\_init,  impdrv\_ocvctl\_set\_mem\_init, impdrv\_ocvctl\_set\_core\_map,  impdrv\_ocvctl\_set\_cl, impdrv\_ocvctl\_set\_irq\_mask,  impdrv\_ocvctl\_execute, impdrv\_ocvctl\_int\_handler, impdrv\_ocvctl\_resume\_exe, impdrv\_ocvctl\_pm\_set\_policy, impdrv\_ocvctl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_ocvctl\_set\_cl\_brk\_addr, impdrv\_ocvctl\_set\_cond\_gosub,  impdrv\_ocvctl\_pm\_get\_state, impdrv\_ocvctl\_quit  },  /\*\* IMPDRV\_CORE\_TYPE\_DMAC \*/  { impdrv\_dmactl\_is\_valid\_core, impdrv\_dmactl\_check\_state,  impdrv\_dmactl\_set\_state, impdrv\_dmactl\_init\_start,  impdrv\_dmactl\_init\_end, impdrv\_dmactl\_start,  impdrv\_dmactl\_stop, impdrv\_dmactl\_attr\_init,  impdrv\_dmactl\_set\_mem\_init, impdrv\_dmactl\_set\_core\_map,  impdrv\_dmactl\_set\_cl, impdrv\_dmactl\_set\_irq\_mask,  impdrv\_dmactl\_execute, impdrv\_dmactl\_int\_handler, impdrv\_dmactl\_resume\_exe, impdrv\_dmactl\_pm\_set\_policy, impdrv\_dmactl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_dmactl\_set\_cl\_brk\_addr, impdrv\_dmactl\_set\_cond\_gosub,  impdrv\_dmactl\_pm\_get\_state, impdrv\_dmactl\_quit  },  /\* IMPDRV\_CORE\_TYPE\_DMAC\_SLIM \*/  { impdrv\_dmasctl\_is\_valid\_core, impdrv\_dmasctl\_check\_state,  impdrv\_dmasctl\_set\_state, impdrv\_dmasctl\_init\_start,  impdrv\_dmasctl\_init\_end, impdrv\_dmasctl\_start,  impdrv\_dmasctl\_stop, impdrv\_dmasctl\_attr\_init,  impdrv\_dmasctl\_set\_mem\_init, impdrv\_dmasctl\_set\_core\_map,  impdrv\_dmasctl\_set\_cl, impdrv\_dmasctl\_set\_irq\_mask,  impdrv\_dmasctl\_execute, impdrv\_dmasctl\_int\_handler, impdrv\_dmasctl\_resume\_exe, impdrv\_dmasctl\_pm\_set\_policy, impdrv\_dmasctl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_dmasctl\_set\_cl\_brk\_addr, impdrv\_dmasctl\_set\_cond\_gosub,  impdrv\_dmasctl\_pm\_get\_state, impdrv\_dmasctl\_quit  },  /\*\* IMPDRV\_CORE\_TYPE\_PSCEXE \*/  { impdrv\_pscctl\_is\_valid\_core, impdrv\_pscctl\_check\_state,  impdrv\_pscctl\_set\_state, impdrv\_pscctl\_init\_start,  impdrv\_pscctl\_init\_end, impdrv\_pscctl\_start,  impdrv\_pscctl\_stop, impdrv\_pscctl\_attr\_init,  impdrv\_pscctl\_set\_mem\_init, impdrv\_pscctl\_set\_core\_map,  impdrv\_pscctl\_set\_cl, impdrv\_pscctl\_set\_irq\_mask,  impdrv\_pscctl\_execute, impdrv\_pscctl\_int\_handler, impdrv\_pscctl\_resume\_exe, impdrv\_pscctl\_pm\_set\_policy, impdrv\_pscctl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_pscctl\_set\_cl\_brk\_addr, impdrv\_pscctl\_set\_cond\_gosub,  impdrv\_pscctl\_pm\_get\_state, impdrv\_pscctl\_quit  },  /\*\* IMPDRV\_CORE\_TYPE\_PSCOUT \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_CNN \*/  { impdrv\_cnnctl\_is\_valid\_core, impdrv\_cnnctl\_check\_state, impdrv\_cnnctl\_set\_state, impdrv\_cnnctl\_init\_start, impdrv\_cnnctl\_init\_end, impdrv\_cnnctl\_start, impdrv\_cnnctl\_stop, impdrv\_cnnctl\_attr\_init, impdrv\_cnnctl\_set\_mem\_init, impdrv\_cnnctl\_set\_core\_map, impdrv\_cnnctl\_set\_cl, impdrv\_cnnctl\_set\_irq\_mask, impdrv\_cnnctl\_execute, impdrv\_cnnctl\_int\_handler, impdrv\_cnnctl\_resume\_exe, impdrv\_cnnctl\_pm\_set\_policy, impdrv\_cnnctl\_pm\_get\_policy, NULL, NULL, NULL,  impdrv\_cnnctl\_set\_cl\_brk\_addr, impdrv\_cnnctl\_set\_cond\_gosub,  impdrv\_cnnctl\_pm\_get\_state, impdrv\_cnnctl\_quit  },  /\*\* IMPDRV\_CORE\_TYPE\_DSP \*/  { impdrv\_dspctl\_is\_valid\_core, impdrv\_dspctl\_check\_state,  impdrv\_dspctl\_set\_state, impdrv\_dspctl\_init\_start,  impdrv\_dspctl\_init\_end, impdrv\_dspctl\_dsp\_start,  impdrv\_dspctl\_stop, impdrv\_dspctl\_attr\_init,  impdrv\_dspctl\_set\_mem\_init, impdrv\_dspctl\_set\_core\_map,  NULL, NULL, impdrv\_dspctl\_dsp\_execute,  impdrv\_dspctl\_int\_handler, impdrv\_dspctl\_resume\_exe,  impdrv\_dspctl\_pm\_set\_policy, impdrv\_dspctl\_pm\_get\_policy,  impdrv\_dspctl\_set\_dsp, NULL,  impdrv\_dspctl\_conf\_reg\_check, NULL, NULL,  impdrv\_dspctl\_pm\_get\_state, impdrv\_dspctl\_quit  },  /\*\* IMPDRV\_CORE\_TYPE\_DTA \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL  },  }; |
| **Description** | IMP Driver determines the Call functions for Core control component by this table. |
| **Remarks** | N/A |

Table 5‑82: s\_corectl\_func\_tbl

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_corectl\_func\_t s\_corectl\_func\_tbl[IMPDRV\_CORE\_TYPE\_MAX] =  {  /\*\* Not Assigned \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_IMP \*/  { impdrv\_impctl\_is\_valid\_core, impdrv\_impctl\_check\_state,  impdrv\_impctl\_set\_state, impdrv\_impctl\_init\_start,  impdrv\_impctl\_init\_end, impdrv\_impctl\_start,  impdrv\_impctl\_stop, impdrv\_impctl\_attr\_init,  impdrv\_impctl\_set\_mem\_init, impdrv\_impctl\_set\_core\_map,  impdrv\_impctl\_set\_cl, impdrv\_impctl\_set\_irq\_mask,  impdrv\_impctl\_execute, impdrv\_impctl\_int\_handler, impdrv\_impctl\_resume\_exe, impdrv\_impctl\_pm\_set\_policy, impdrv\_impctl\_pm\_get\_policy, NULL,  impdrv\_impctl\_bus\_if\_check, impdrv\_impctl\_conf\_reg\_check,  NULL, NULL  },  /\* IMPDRV\_CORE\_TYPE\_IMP\_SLIM \*/ [R-CarV3H / V3H\_2]  { impdrv\_impctl\_is\_valid\_core, impdrv\_impctl\_check\_state,  impdrv\_impctl\_set\_state, impdrv\_impctl\_init\_start,  impdrv\_impctl\_init\_end, impdrv\_impctl\_start,  impdrv\_impctl\_stop, impdrv\_impctl\_attr\_init,  impdrv\_impctl\_set\_mem\_init, impdrv\_impctl\_set\_core\_map,  impdrv\_impctl\_set\_cl, impdrv\_impctl\_set\_irq\_mask,  impdrv\_impctl\_execute, impdrv\_impctl\_int\_handler, impdrv\_impctl\_resume\_exe, impdrv\_impctl\_pm\_set\_policy, impdrv\_impctl\_pm\_get\_policy, NULL,  impdrv\_impsctl\_bus\_if\_check, impdrv\_impsctl\_conf\_reg\_check,  NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_OCV \*/  { impdrv\_ocvctl\_is\_valid\_core, impdrv\_ocvctl\_check\_state,  impdrv\_ocvctl\_set\_state, impdrv\_ocvctl\_init\_start,  impdrv\_ocvctl\_init\_end, impdrv\_ocvctl\_start,  impdrv\_ocvctl\_stop, impdrv\_ocvctl\_attr\_init,  impdrv\_ocvctl\_set\_mem\_init, impdrv\_ocvctl\_set\_core\_map,  impdrv\_ocvctl\_set\_cl, impdrv\_ocvctl\_set\_irq\_mask,  impdrv\_ocvctl\_execute, impdrv\_ocvctl\_int\_handler, impdrv\_ocvctl\_resume\_exe, impdrv\_ocvctl\_pm\_set\_policy, impdrv\_ocvctl\_pm\_get\_policy, NULL,  NULL, impdrv\_ocvctl\_conf\_reg\_check,  NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_DMAC \*/  { impdrv\_dmactl\_is\_valid\_core, impdrv\_dmactl\_check\_state,  impdrv\_dmactl\_set\_state, impdrv\_dmactl\_init\_start,  impdrv\_dmactl\_init\_end, impdrv\_dmactl\_start,  impdrv\_dmactl\_stop, impdrv\_dmactl\_attr\_init,  impdrv\_dmactl\_set\_mem\_init, impdrv\_dmactl\_set\_core\_map,  impdrv\_dmactl\_set\_cl, impdrv\_dmactl\_set\_irq\_mask,  impdrv\_dmactl\_execute, impdrv\_dmactl\_int\_handler, impdrv\_dmactl\_resume\_exe, impdrv\_dmactl\_pm\_set\_policy, impdrv\_dmactl\_pm\_get\_policy, NULL,  impdrv\_dmactl\_bus\_if\_check, impdrv\_dmactl\_conf\_reg\_check,  NULL, NULL  },  /\* IMPDRV\_CORE\_TYPE\_DMAC\_SLIM \*/ [R-CarV3H\_2]  { impdrv\_impctl\_is\_valid\_core, impdrv\_impctl\_check\_state,  impdrv\_impctl\_set\_state, impdrv\_impctl\_init\_start,  impdrv\_impctl\_init\_end, impdrv\_impctl\_start,  impdrv\_impctl\_stop, impdrv\_impctl\_attr\_init,  impdrv\_impctl\_set\_mem\_init, impdrv\_impctl\_set\_core\_map,  impdrv\_impctl\_set\_cl, impdrv\_impctl\_set\_irq\_mask,  impdrv\_impctl\_execute, impdrv\_impctl\_int\_handler, impdrv\_impctl\_resume\_exe, impdrv\_impctl\_pm\_set\_policy, impdrv\_dmasctl\_pm\_get\_policy, NULL,  NULL, impdrv\_dmasctl\_conf\_reg\_check,  NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_PSCEXE \*/  { impdrv\_pscctl\_is\_valid\_core, impdrv\_pscctl\_check\_state,  impdrv\_pscctl\_set\_state, impdrv\_pscctl\_init\_start,  impdrv\_pscctl\_init\_end, impdrv\_pscctl\_start,  impdrv\_pscctl\_stop, impdrv\_pscctl\_attr\_init,  impdrv\_pscctl\_set\_mem\_init, impdrv\_pscctl\_set\_core\_map,  impdrv\_pscctl\_set\_cl, impdrv\_pscctl\_set\_irq\_mask,  impdrv\_pscctl\_execute, impdrv\_pscctl\_int\_handler, impdrv\_pscctl\_resume\_exe, impdrv\_pscctl\_pm\_set\_policy, impdrv\_pscctl\_pm\_get\_policy, NULL,  NULL, impdrv\_pscctl\_conf\_reg\_check,  NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_PSCOUT \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_CNN \*/  { impdrv\_cnnctl\_is\_valid\_core, impdrv\_cnnctl\_check\_state, impdrv\_cnnctl\_set\_state, impdrv\_cnnctl\_init\_start, , impdrv\_cnnctl\_start, impdrv\_cnnctl\_stop, impdrv\_cnnctl\_attr\_init, impdrv\_cnnctl\_set\_mem\_init, impdrv\_cnnctl\_set\_core\_map, impdrv\_cnnctl\_set\_cl, impdrv\_cnnctl\_set\_irq\_mask, impdrv\_cnnctl\_execute, impdrv\_cnnctl\_int\_handler, impdrv\_cnnctl\_resume\_exe, impdrv\_cnnctl\_pm\_set\_policy, impdrv\_cnnctl\_pm\_get\_policy, NULL,  NULL, impdrv\_cnnctl\_conf\_reg\_check,  NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_LDMAC \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL NULL, NULL, NULL, NULL, NULL  },  /\*\* IMPDRV\_CORE\_TYPE\_DTA \*/  { NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,  NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL NULL, NULL, NULL, NULL, NULL  },  }; |
| **Description** | IMP Driver determines the Call functions for Core control component by this table. |
| **Remarks** | N/A |

##### s\_impdrv\_core\_to\_syncc\_table

Table 5‑83: s\_impdrv\_core\_to\_syncc\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_syncc\_table\_t s\_impdrv\_core\_to\_syncc\_table[IMPDRV\_SYNCCTABLE\_NUM] =  {  /\* core\_type, core\_num, syncc\_val \*/  { IMPDRV\_CORE\_TYPE\_IMP, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 1U, 0x01U },  { IMPDRV\_CORE\_TYPE\_IMP, 2U, 0x02U },  { IMPDRV\_CORE\_TYPE\_IMP, 3U, 0x03U },  { IMPDRV\_CORE\_TYPE\_OCV, 0U, 0x10U },  { IMPDRV\_CORE\_TYPE\_OCV, 1U, 0x11U },  { IMPDRV\_CORE\_TYPE\_OCV, 2U, 0x12U },  { IMPDRV\_CORE\_TYPE\_OCV, 3U, 0x13U },  { IMPDRV\_CORE\_TYPE\_OCV, 4U, 0x14U },  { IMPDRV\_CORE\_TYPE\_OCV, 5U, 0x15U },  { IMPDRV\_CORE\_TYPE\_OCV, 6U, 0x16U },  { IMPDRV\_CORE\_TYPE\_OCV, 7U, 0x17U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0U, 0x20U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1U, 0x21U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2U, 0x22U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3U, 0x23U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 1U, 0x31U },  { IMPDRV\_CORE\_TYPE\_CNN, 0U, 0x40U },  { IMPDRV\_CORE\_TYPE\_CNN, 1U, 0x41U },  { IMPDRV\_CORE\_TYPE\_CNN, 2U, 0x42U },  { IMPDRV\_CORE\_TYPE\_DTA, 0U, 0x70U },  }; |
| **Description** | IMP Driver determines the Bit values of the SYNCC value by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “SYNCCR” register. |
| **Remarks** | N/A |

Table 5‑84: s\_impdrv\_core\_to\_syncc\_table

|  |
| --- |
| **[R-CarV3M]**  IMPDRV\_STATIC  const st\_impdrv\_syncc\_table\_t s\_impdrv\_core\_to\_syncc\_table[IMPDRV\_SYNCCTABLE\_NUM] =  {  /\* core\_type, core\_num, syncc\_val \*/  { IMPDRV\_CORE\_TYPE\_INVALID, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 1U, 0x01U },  { IMPDRV\_CORE\_TYPE\_IMP, 2U, 0x02U },  { IMPDRV\_CORE\_TYPE\_IMP, 3U, 0x03U },  { IMPDRV\_CORE\_TYPE\_OCV, 0U, 0x10U },  { IMPDRV\_CORE\_TYPE\_OCV, 1U, 0x11U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0U, 0x20U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1U, 0x21U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 1U, 0x31U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 2U, 0x32U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 3U, 0x33U },  { IMPDRV\_CORE\_TYPE\_CNN, 0U, 0x40U },  { IMPDRV\_CORE\_TYPE\_DTA, 0U, 0x70U }  };  **[R-CarV3H]**  IMPDRV\_STATIC  const st\_impdrv\_syncc\_table\_t s\_impdrv\_core\_to\_syncc\_table[IMPDRV\_SYNCCTABLE\_NUM] =  {  /\* core\_type, core\_num, syncc\_val \*/  { IMPDRV\_CORE\_TYPE\_INVALID, 0U, 0xFFU },  { IMPDRV\_CORE\_TYPE\_IMP, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 1U, 0x01U },  { IMPDRV\_CORE\_TYPE\_IMP, 2U, 0x02U },  { IMPDRV\_CORE\_TYPE\_IMP, 3U, 0x03U },  { IMPDRV\_CORE\_TYPE\_IMP, 4U, 0x04U },  { IMPDRV\_CORE\_TYPE\_IMP\_SLIM, 0U, 0x05U },  { IMPDRV\_CORE\_TYPE\_OCV, 0U, 0x10U },  { IMPDRV\_CORE\_TYPE\_OCV, 1U, 0x11U },  { IMPDRV\_CORE\_TYPE\_OCV, 2U, 0x12U },  { IMPDRV\_CORE\_TYPE\_OCV, 3U, 0x13U },  { IMPDRV\_CORE\_TYPE\_OCV, 4U, 0x14U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0U, 0x20U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1U, 0x21U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2U, 0x22U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3U, 0x23U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 1U, 0x31U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 1U, 0x31U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 2U, 0x32U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 3U, 0x33U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 4U, 0x34U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 5U, 0x35U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 6U, 0x36U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 7U, 0x37U },  { IMPDRV\_CORE\_TYPE\_CNN, 0U, 0x40U },  { IMPDRV\_CORE\_TYPE\_IMR, 4U, 0x54U },  { IMPDRV\_CORE\_TYPE\_IMR, 5U, 0x55U },  { IMPDRV\_CORE\_TYPE\_DTA, 0U, 0x70U }  };  **[R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_syncc\_table\_t s\_impdrv\_core\_to\_syncc\_table[IMPDRV\_SYNCCTABLE\_NUM] =  {  /\* core\_type, core\_num, syncc\_val \*/  { IMPDRV\_CORE\_TYPE\_INVALID, 0U, 0xFFU },  { IMPDRV\_CORE\_TYPE\_IMP, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 1U, 0x01U },  { IMPDRV\_CORE\_TYPE\_IMP, 2U, 0x02U },  { IMPDRV\_CORE\_TYPE\_IMP, 3U, 0x03U },  { IMPDRV\_CORE\_TYPE\_IMP, 4U, 0x04U },  { IMPDRV\_CORE\_TYPE\_IMP\_SLIM, 0U, 0x05U },  { IMPDRV\_CORE\_TYPE\_OCV, 0U, 0x10U },  { IMPDRV\_CORE\_TYPE\_OCV, 1U, 0x11U },  { IMPDRV\_CORE\_TYPE\_OCV, 2U, 0x12U },  { IMPDRV\_CORE\_TYPE\_OCV, 3U, 0x13U },  { IMPDRV\_CORE\_TYPE\_OCV, 4U, 0x14U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0U, 0x20U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1U, 0x21U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2U, 0x22U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3U, 0x23U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 0U, 0x24U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 1U, 0x25U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 1U, 0x31U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 2U, 0x32U },  { IMPDRV\_CORE\_TYPE\_PSCOUT, 3U, 0x33U },  { IMPDRV\_CORE\_TYPE\_CNN, 0U, 0x40U },  { IMPDRV\_CORE\_TYPE\_IMR, 4U, 0x54U },  { IMPDRV\_CORE\_TYPE\_IMR, 5U, 0x55U },  { IMPDRV\_CORE\_TYPE\_DTA, 0U, 0x70U }  };  [R-CarV4H]  IMPDRV\_STATIC  const st\_impdrv\_syncc\_table\_t s\_impdrv\_core\_to\_syncc\_table[IMPDRV\_SYNCCTABLE\_NUM] =  {  /\* core\_type, core\_num, syncc\_val \*/  { IMPDRV\_CORE\_TYPE\_INVALID, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 0U, 0x00U },  { IMPDRV\_CORE\_TYPE\_IMP, 1U, 0x01U },  { IMPDRV\_CORE\_TYPE\_IMP, 2U, 0x02U },  { IMPDRV\_CORE\_TYPE\_IMP, 3U, 0x03U },  { IMPDRV\_CORE\_TYPE\_OCV, 0U, 0x10U },  { IMPDRV\_CORE\_TYPE\_OCV, 1U, 0x11U },  { IMPDRV\_CORE\_TYPE\_OCV, 2U, 0x12U },  { IMPDRV\_CORE\_TYPE\_OCV, 3U, 0x13U },  { IMPDRV\_CORE\_TYPE\_DMAC, 0U, 0x20U },  { IMPDRV\_CORE\_TYPE\_DMAC, 1U, 0x21U },  { IMPDRV\_CORE\_TYPE\_DMAC, 2U, 0x22U },  { IMPDRV\_CORE\_TYPE\_DMAC, 3U, 0x23U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 0U, 0x24U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 1U, 0x25U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 2U, 0x26U },  { IMPDRV\_CORE\_TYPE\_DMAC\_SLIM, 3U, 0x27U },  { IMPDRV\_CORE\_TYPE\_PSCEXE, 0U, 0x30U },  { IMPDRV\_CORE\_TYPE\_CNN, 0U, 0x40U },  { IMPDRV\_CORE\_TYPE\_DSP, 0U, 0x50U },  { IMPDRV\_CORE\_TYPE\_DSP, 1U, 0x51U },  { IMPDRV\_CORE\_TYPE\_DSP, 2U, 0x52U },  { IMPDRV\_CORE\_TYPE\_DSP, 3U, 0x53U },  { IMPDRV\_CORE\_TYPE\_IMR, 0U, 0x60U },  { IMPDRV\_CORE\_TYPE\_IMR, 1U, 0x61U },  { IMPDRV\_CORE\_TYPE\_IMR, 2U, 0x62U },  { IMPDRV\_CORE\_TYPE\_IMR, 3U, 0x63U },  { IMPDRV\_CORE\_TYPE\_IMR, 4U, 0x64U },  { IMPDRV\_CORE\_TYPE\_DTA, 0U, 0x70U },  }; |
| IMP Driver determines the Bit values of the SYNCC value by this table.  This bit value is converted from the core type and the core number.  The settings in this table apply to the “SYNCCR” register. |
| N/A |

##### s\_impdrv\_inten\_reg\_tbl

Table 5‑85: s\_impdrv\_inten\_reg\_tbl

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const st\_impdrv\_inten\_reg\_tbl\_t s\_impdrv\_inten\_reg\_tbl[IMPDRV\_INSTANCETABLE\_NUM] =  {  /\*\* p\_inten\_group\_table p\_inten\_cnv\_table cnv\_table\_num \*/  { NULL, NULL, 0xFFFFFFFFU },  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_minten\_group\_table, s\_impdrv\_minten\_cnv\_table, IMPDRV\_MINTEN\_CORE\_MAX},  }; |
| **Description** | IMP Driver determines conversion of register table by this table.  Follows the st\_impdrv\_inten\_reg\_tbl\_t structure. |
| **Remarks** | N/A |

Table 5‑86: s\_impdrv\_inten\_reg\_tbl

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const [st\_impdrv\_inten\_reg\_tbl\_t](#_st_impdrv_inten_reg_tbl_t)s\_impdrv\_inten\_reg\_tbl[IMPDRV\_INSTANCETABLE\_NUM] =  {  /\*\* p\_inten\_group\_table p\_inten\_cnv\_table cnv\_table\_num \*/  { NULL, NULL, 　　　　　　　IMPDRV\_DUMMY\_UINT32 },  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten0\_group\_table, s\_impdrv\_inten0\_cnv\_table, IMPDRV\_INTEN0\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_inten1\_group\_table, s\_impdrv\_inten1\_cnv\_table, IMPDRV\_INTEN1\_CORE\_MAX},  { s\_impdrv\_minten\_group\_table, s\_impdrv\_minten\_cnv\_table, IMPDRV\_MINTEN\_CORE\_MAX},  }; |
| **Description** | IMP Driver determines conversion of register table by this table.  Follows the [st\_impdrv\_inten\_reg\_tbl\_t](#_st_impdrv_inten_reg_tbl_t) structure. |
| **Remarks** | N/A |

##### s\_hwrsc\_def\_table

Table 5‑87: s\_hwrsc\_def\_table

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M]**  IMPDRV\_STATIC  const [st\_impdrv\_hwrsc\_def\_t](#_st_impdrv_hwrsc_def_t) s\_hwrsc\_def\_table[IMPDRV\_HWRSC\_TABLE\_NUM] =  {  /\*\* Resource Name(Key), Physical adrs, Area size Read only internal \*/  { “imp\_top\_00” , 0xFFA00000U , 0x00010000U , true , false },  { “imp\_spmi\_00”, 0xED000000U , 0x000E0000U , false , false },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true }  };  **[R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const [st\_impdrv\_hwrsc\_def\_t](#_st_impdrv_hwrsc_def_t) s\_hwrsc\_def\_table[IMPDRV\_HWRSC\_TABLE\_NUM] =  {  /\*\* Resource Name(Key), Physical adrs, Area size Read only internal \*/  { “imp\_top\_00” , 0xFFA00000U , 0x00010000U , true , false },  { “imp\_spmi\_00”, 0xED000000U , 0x00200000U , false , false },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true }  };  [R-CarV4H]  IMPDRV\_STATIC  const st\_impdrv\_hwrsc\_def\_t s\_hwrsc\_def\_table[IMPDRV\_HWRSC\_TABLE\_NUM] =  {  /\*\* Resource Name(Key), Physical adrs, Area size Read only internal \*/  { "imp\_top\_00" , 0xFF900000U , 0x00004000U , true , false },  { "imp\_spmi\_00" , 0xED300000U , 0x00100000U , false , false },  { "imp\_spmc\_00" , 0xED400000U , 0x00200000U , false , false },  { "imp\_reg\_spmi\_00" , 0xFFA80000U , 0x00004000U , true , false },  { "imp\_reg\_spmc\_00" , 0xFFAB0000U , 0x00004000U , true , false },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true },  { NULL , 0x00000000U , 0x00000000U , true , true }  }; |
| **Description** | IMPDRV\_Private\_Static\_Variables Private file static variables  Follows the [st\_impdrv\_hwrsc\_def\_t](#_st_impdrv_hwrsc_def_t) structure. |
| **Remarks** | N/A |

##### s\_impdrv\_used\_instance

Table 5‑88: s\_impdrv\_used\_instance

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const uint32\_t s\_impdrv\_used\_instance[IMPDRV\_INSTANCETABLE\_NUM] =  {  0xFFFFFFFFU, /\*\*< Dummy for Index adjustment \*/  0x00000001U, /\*\*< IMPDRV\_INSTANCE\_0 \*/  0x00000002U, /\*\*< IMPDRV\_INSTANCE\_1 \*/  0x00000004U, /\*\*< IMPDRV\_INSTANCE\_2 \*/  0x00000008U, /\*\*< IMPDRV\_INSTANCE\_3 \*/  0x00000010U /\*\*< IMPDRV\_INSTANCE\_4 \*/  };  [R-CarV4H]  IMPDRV\_STATIC  const uint32\_t s\_impdrv\_used\_instance[IMPDRV\_INSTANCETABLE\_NUM] =  {  0xFFFFFFFFU, /\*\*< Dummy for Index adjustment \*/  0x00000001U, /\*\*< IMPDRV\_INSTANCE\_0 \*/  0x00000002U, /\*\*< IMPDRV\_INSTANCE\_1 \*/  0x00000004U /\*\*< IMPDRV\_INSTANCE\_2 \*/  }; |
| **Description** | IMP Driver determines conversion of register table by this table. |
| **Remarks** | N/A |

##### s\_impdrv\_inten\_group\_table

Table 5‑89: s\_impdrv\_inten\_group\_table

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_group\_tbl\_t s\_impdrv\_inten\_group\_table[IMPDRV\_GROUP\_MAX] =  {  /\*\* group\_bit group\_reg \*/  { IMPDRV\_DUMMY\_UINT32, IMPDRV\_DUMMY\_UINT32 }, /\*\*< Dummy for Index \*/  { IMPDRV\_DUMMY\_UINT32, IMPDRV\_DUMMY\_UINT32 }, /\*\*< Dummy for Index \*/  { 0x20000000U, IMPDRV\_REG\_G0INTSEL }, /\*\* Group0. \*/  { 0x40000000U, IMPDRV\_REG\_G1INTSEL }, /\*\* Group1. \*/  { 0x80000000U, IMPDRV\_REG\_G2INTSEL } /\*\* Group2. \*/  };  [R-CarV4H]  Not Defined |
| **Description** | IMP Driver determines conversion of group table by this table.  Follows the [st\_impdrv\_inten\_reg\_tbl\_t](#_st_impdrv_inten_reg_tbl_t) structure. |
| **Remarks** | N/A |

##### s\_impdrv\_cnv\_ins\_table

Table 5‑90: s\_impdrv\_cnv\_ins\_table

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  e\_impdrv\_instance\_t s\_impdrv\_cnv\_ins\_table =  {  /\*\* core\_type \*/  IMPDRV\_INSTANCE\_0,/\*\*< Identification value for assigning to the IRQ number (0th) \*/  IMPDRV\_INSTANCE\_1,/\*\*< Identification value for assigning to the IRQ number (1st) \*/  IMPDRV\_INSTANCE\_2,/\*\*< Identification value for assigning to the IRQ number (2nd) \*/  IMPDRV\_INSTANCE\_3,/\*\*< Identification value for assigning to the IRQ number (3rd) \*/  IMPDRV\_INSTANCE\_4,/\*\*< Identification value for assigning to the IRQ number (4th) \*/  };  [R-CarV4H]  IMPDRV\_STATIC  const e\_impdrv\_instance\_t s\_impdrv\_cnv\_ins\_table[IMPDRV\_INSTANCETABLE\_NUM - 1U] =  {  /\*\* core\_type \*/  IMPDRV\_INSTANCE\_0 , /\*\*< Identification value for assigning to the IRQ number (0th) \*/  IMPDRV\_INSTANCE\_1 , /\*\*< Identification value for assigning to the IRQ number (1st) \*/  IMPDRV\_INSTANCE\_2 , /\*\*< Identification value for assigning to the IRQ number (2nd)\*/  }; |
| **Description** | IMP Driver determines conversion of Instance table by this table.  Follows the e\_impdrv\_instance\_t structure. |
| **Remarks** | N/A |

#### Core Control

##### s\_mb\_init\_info1

Table 5‑91: s\_mb\_init\_info1

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_mb\_init\_info1[IMPDRV\_DMA\_MB\_INFO\_SIZE1] =  {  {IMPDRV\_REG\_DMA\_IMGSIZER,IMPDRV\_DMA\_IMGSIZER\_VAL,true,IMPDRV\_DMA\_IMGSIZER\_VAL},  {IMPDRV\_REG\_DMA\_FCR0, IMPDRV\_DMA\_FCR0\_VAL, true,IMPDRV\_DMA\_FCR0\_VAL},  {IMPDRV\_REG\_DMA\_FCR1, IMPDRV\_DMA\_FCR1\_VAL, true,IMPDRV\_DMA\_FCR1\_VAL}  };  [R-CarV4H]  Not Defined |
| **Description** | The IMP Driver initializes the multi-bank memory using this table.  Follows the st\_impdrv\_dma\_write\_info\_tbl\_t structure. |
| **Remarks** | N/A |

##### s\_mb\_init\_info2

Table 5‑92: s\_mb\_init\_info2

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_mb\_init\_info2[IMPDRV\_DMA\_MB\_INFO\_SIZE2] =  {  {IMPDRV\_REG\_DMA\_S0STR, IMPDRV\_DMA\_S0STR\_VAL, true, IMPDRV\_DMA\_S0STR\_VAL },  {IMPDRV\_REG\_DMA\_S0DATAR, IMPDRV\_DMA\_S0DATAR\_VAL, true, IMPDRV\_DMA\_S0DATAR\_VAL },  {IMPDRV\_REG\_DMA\_S0CR, IMPDRV\_DMA\_S0CR\_VAL, true, IMPDRV\_DMA\_S0CR\_VAL },  {IMPDRV\_REG\_DMA\_S1SAR, IMPDRV\_DMA\_S1SAR\_VAL, true, IMPDRV\_DMA\_S1SAR\_VAL },  {IMPDRV\_REG\_DMA\_S1STR, IMPDRV\_DMA\_S1STR\_VAL, true, IMPDRV\_DMA\_S1STR\_VAL },  {IMPDRV\_REG\_DMA\_S1DATAR, IMPDRV\_DMA\_S1DATAR\_VAL, true, IMPDRV\_DMA\_S1DATAR\_VAL },  {IMPDRV\_REG\_DMA\_S1CR, IMPDRV\_DMA\_S1CR\_VAL, true, IMPDRV\_DMA\_S1CR\_VAL }  };  **[R-CarV4H]**  Not Defined |
| **Description** | The IMP Driver initializes the multi-bank memory using this table.  Follows the st\_impdrv\_dma\_write\_info\_tbl\_t structure. |
| **Remarks** | N/A |

##### s\_mb\_init\_info3

Table 5‑93: s\_mb\_init\_info3

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_mb\_init\_info3[IMPDRV\_DMA\_MB\_INFO\_SIZE3] =  {  {IMPDRV\_REG\_DMA\_D0STR, IMPDRV\_DMA\_D0STR\_VAL, true, IMPDRV\_DMA\_D0STR\_VAL },  {IMPDRV\_REG\_DMA\_D0CR, IMPDRV\_DMA\_D0CR\_VAL, true, IMPDRV\_DMA\_D0CR\_VAL },  {IMPDRV\_REG\_DMA\_DSWPR, IMPDRV\_DMA\_DSWPR\_VAL, true, IMPDRV\_DMA\_DSWPR\_VAL },  {IMPDRV\_REG\_DMA\_TSCR, IMPDRV\_DMA\_TSCR\_VAL, true, IMPDRV\_DMA\_TSCR\_VAL },  {IMPDRV\_REG\_DMA\_SCTLR, IMPDRV\_DMA\_SCTLR0\_MB\_VAL, false, 0U }  };  **[R-CarV4H]**  Not Defined |
| **Description** | The IMP Driver initializes the multi-bank memory using this table.  Follows the st\_impdrv\_dma\_write\_info\_tbl\_t structure. |
| **Remarks** | N/A |

##### s\_test\_pattern\_tbl

Table 5‑94: s\_test\_pattern\_tbl

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC const struct {  uint32\_t pattern;  size\_t size;  } s\_test\_pattern\_tbl[] = {  { 0x55555555U, (sizeof(uint32\_t) \* 8U) },  { 0xAAAAAAAAU, (sizeof(uint32\_t) \* 8U) },  { 0x33333333U, (sizeof(uint32\_t) \* 8U) },  { 0xCCCCCCCCU, (sizeof(uint32\_t) \* 8U) },  { 0x0F0F0F0FU, (sizeof(uint32\_t) \* 8U) },  { 0xF0F0F0F0U, (sizeof(uint32\_t) \* 8U) },  { 0xFFFFFFFFU, (sizeof(uint32\_t) \* 8U) },  { 0x00000000U, (sizeof(uint32\_t) \* 8U) }  };  **[R-CarV4H]**  Not Defined |
| **Description** | Test pattern to the test buffer. Private static variables. |
| **Remarks** | N/A |

##### s\_soft\_reset

Table 5‑95: s\_soft\_reset

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_soft\_reset[] =  {  { IMPDRV\_REG\_IMP\_CNF, (IMPDRV\_IMP\_CNF\_VAL | IMPDRV\_IMP\_CNF\_SWRST), false, 0U },  { IMPDRV\_REG\_IMP\_CNF, (IMPDRV\_IMP\_CNF\_VAL), false, 0U }  };  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_soft\_reset[] =  {  { IMPDRV\_REG\_DMA\_SCTLR, IMPDRV\_DMA\_SCTLR\_SWRST, false, 0U },  { IMPDRV\_REG\_DMA\_SCTLR, IMPDRV\_DMA\_SCTLR\_CLR, false, 0U }  };  **[R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_soft\_reset[] =  {  { IMPDRV\_REG\_IMPS\_CNF, (IMPDRV\_IMPS\_CNF\_VAL | IMPDRV\_IMPS\_CNF\_SWRST), false, 0U },  { IMPDRV\_REG\_IMPS\_CNF, (IMPDRV\_IMPS\_CNF\_VAL), false, 0U }  };  **[R-CarV4H]**  Not Defined |
| **Description** | Write data from software reset table. Private static variables. |
| **Remarks** | N/A |

##### s\_interrupt\_enable

Table 5‑96: s\_interrupt\_enable

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_interrupt\_enable[] =  {  { IMPDRV\_REG\_IMP\_INTSTS, IMPDRV\_IMP\_INTSTS\_CLR, false, 0U },  { IMPDRV\_REG\_IMP\_INTEN, (IMPDRV\_IMP\_INTSTS\_MASKCPU | IMPDRV\_IMP\_INTSTS\_APIPINT), false, 0U }  };  **[R-CarV3M]**  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_interrupt\_enable[] =  {  { IMPDRV\_REG\_DMA\_SCR, 0x00000071U, false, 0U },  { IMPDRV\_REG\_DMA\_SER, (0x00000071U | IMPDRV\_DMA\_SER\_FIX), true, (0x00000071U | IMPDRV\_DMA\_SER\_FIX) },  { IMPDRV\_REG\_DMA\_IMR, (0x00000070U | IMPDRV\_DMA\_SER\_FIX), false, 0U }  };  **[R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_interrupt\_enable[] =  {  { IMPDRV\_REG\_IMPS\_INTSTS, IMPDRV\_IMPS\_INTSTS\_CLR, false, 0U },  { IMPDRV\_REG\_IMPS\_INTEN, (IMPDRV\_IMPS\_INTSTS\_MASKCPU | IMPDRV\_IMPS\_INTSTS\_APIPINT), false, 0U }  };  **[R-CarV4H]**  Not Defined |
| **Description** | Write data from interrupt setting table. Private static variables. |
| **Remarks** | N/A |

##### s\_transfer\_config

Table 5‑97: s\_transfer\_config

|  |  |
| --- | --- |
| **Format** | **[R-CarV3M] [R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_transfer\_config[] =  {  { IMPDRV\_REG\_IMP\_IPFUN2, 0x00100000U, true, 0x00100000U },  { IMPDRV\_REG\_IMP\_IPFORM, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_APMAG, 0x01010101U, true, 0x01010101U },  { IMPDRV\_REG\_IMP\_APCLPX, 0x00001FFFU, true, 0x00001FFFU },  { IMPDRV\_REG\_IMP\_CNST, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_BINTHR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_BINTHR2, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_KNLMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_KNLMSK2, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_LMCTL, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF02, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF35, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF68, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF911, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF1214, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF15, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF1820, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF2123, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF24, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF02H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF35H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF68H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF911H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF1214H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF1517H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF1820H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF2123H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_COEFF24H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_PFFTASEL1, 0x76543210U, true, 0x76543210U },  { IMPDRV\_REG\_IMP\_PFFTAMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_PFFTBMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMP\_APSBSP, 0U , true, 0U },  { IMPDRV\_REG\_IMP\_APSIZE\_SA, 256U , true, 256U },  { IMPDRV\_REG\_IMP\_APSIZE\_SB, 0U , true, 0U },  { IMPDRV\_REG\_IMP\_APSIZE\_DST, 256U , true, 256U },  { IMPDRV\_REG\_IMP\_APDLY, 0x00000004U, true, 0x00000004U },  { IMPDRV\_REG\_IMP\_APCFG, 0x01033310U, true, 0x01033310U },  { IMPDRV\_REG\_IMP\_IPFUN, 0xF2000000U, true, 0xF2000000U },  { IMPDRV\_REG\_IMP\_IPFUN2, 0x60104000U, true, 0x60104000U },  { IMPDRV\_REG\_IMP\_PFFTASEL1, 0x00000000U, true, 0x00000000U }  };  **[R-CarV3M]**  IMPDRV\_STATIC  const st\_impdrv\_dma\_write\_info\_tbl\_t s\_transfer\_config[] =  {  { IMPDRV\_REG\_DMA\_FCR0, 0x000000CCU, true, 0x000000CCU },  { IMPDRV\_REG\_DMA\_S0STR, 0x00000100U, true, 0x00000100U },  { IMPDRV\_REG\_DMA\_S0DATAR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_DMA\_S0CR, 0x80000003U, true, 0x80000003U },  { IMPDRV\_REG\_DMA\_S1SAR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_DMA\_S1STR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_DMA\_S1DATAR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_DMA\_S1CR, 0x00000003U, true, 0x00000003U },  { IMPDRV\_REG\_DMA\_D0STR, 0x00000100U, true, 0x00000100U },  { IMPDRV\_REG\_DMA\_D0CR, 0x00000003U, true, 0x00000003U },  { IMPDRV\_REG\_DMA\_DSWPR, 0x00000000U, true, 0x00000000U }  };  **[R-CarV3H] [R-CarV3H\_2]**  IMPDRV\_STATIC  const st\_impdrv\_imp\_write\_info\_tbl\_t s\_transfer\_config[] =  {  { IMPDRV\_REG\_IMPS\_IPFUN2, 0x00100000U, true, 0x00100000U },  { IMPDRV\_REG\_IMPS\_IPFORM, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_APMAG, 0x01010101U, true, 0x01010101U },  { IMPDRV\_REG\_IMPS\_APCLPX, 0x00001FFFU, true, 0x00001FFFU },  { IMPDRV\_REG\_IMPS\_CNST, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_BINTHR, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_BINTHR2, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_KNLMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_KNLMSK2, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_LMCTL, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF02, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF35, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF68, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF911, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF1214, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF15, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF1820, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF2123, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF24, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF02H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF35H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF68H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF911H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF1214H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF1517H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF1820H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF2123H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_COEFF24H, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_PFFTASEL1, 0x76543210U, true, 0x76543210U },  { IMPDRV\_REG\_IMPS\_PFFTAMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_PFFTBMSK, 0x00000000U, true, 0x00000000U },  { IMPDRV\_REG\_IMPS\_APSBSP, 0U , true, 0U },  { IMPDRV\_REG\_IMPS\_APSIZE\_SA, 256U , true, 256U },  { IMPDRV\_REG\_IMPS\_APSIZE\_SB, 0U , true, 0U },  { IMPDRV\_REG\_IMPS\_APSIZE\_DST, 256U , true, 256U },  { IMPDRV\_REG\_IMPS\_APDLY, 0x00000004U, true, 0x00000004U },  { IMPDRV\_REG\_IMPS\_APCFG, 0x01033310U, true, 0x01033310U },  { IMPDRV\_REG\_IMPS\_IPFUN, 0xF2000000U, true, 0xF2000000U },  { IMPDRV\_REG\_IMPS\_IPFUN2, 0x60104000U, true, 0x60104000U },  { IMPDRV\_REG\_IMPS\_PFFTASEL1, 0x00000000U, true, 0x00000000U }  };  **[R-Car V4H]**  Not Defined |
| **Description** | Write data from DMA configuration settings table. Private static variables. |
| **Remarks** | N/A |

#### OS Dependence Layer

##### sp\_type\_cnv\_table

Table 5‑98: sp\_type\_cnv\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const char \*const sp\_type\_cnv\_table[IMPDRV\_CORE\_TYPE\_MAX] =  {  “”, 　　　　/\*\*< Not Assigned \*/  “imp”, /\*\*< IMP core \*/  “imp\_ocv”, /\*\*< OCV core \*/  “imp\_dma”, /\*\*< DMAC core \*/  “imp\_psc”, /\*\*< PSC(exe) core \*/  “”, /\*\*< PSC(out) core \*/  “imp\_cnn”, /\*\*< CNN core \*/  “imp\_ldma”, /\*\*< L-DMAC core \*/  “imp\_dta” /\*\*< DTA \*/  }; |
| **Description** | IMP Driver determines the OSAL resources for each core types by this table. |
| **Remarks** | N/A |

Table 5‑99: sp\_type\_cnv\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const char \*const sp\_type\_cnv\_table[IMPDRV\_CORE\_TYPE\_MAX] =  {  “”, 　　　　/\*\*< Not Assigned \*/  “imp”, /\*\*< IMP core \*/  “imp\_slim”, /\*\*< Slim-IMP core \*/  “imp\_ocv”, /\*\*< OCV core \*/  “imp\_dma”, /\*\*< DMAC core \*/  “imp\_dma\_slim”, /\*\*< Slim-DMAC core \*/  “imp\_psc”, /\*\*< PSC(exe) core \*/  “”, /\*\*< PSC(out) core \*/  “imp\_cnn”, /\*\*< CNN core \*/  “imp\_ldma”, /\*\*< L-DMAC core \*/  “imp\_dta” /\*\*< DTA \*/  };  **[R-CarV4H]**  IMPDRV\_STATIC  const char \*const sp\_type\_cnv\_table[IMPDRV\_CORE\_TYPE\_MAX] =  {  "", /\*\*< Not Assigned \*/  "imp", /\*\*< IMP core \*/  "imp\_slim", /\*\*< Slim-IMP core \*/  "imp\_ocv", /\*\*< OCV core \*/  "imp\_dma", /\*\*< DMAC core \*/  "imp\_dma\_slim", /\*\*< Slim-DMAC core \*/  "imp\_psc", /\*\*< PSC(exe) core \*/  "", /\*\*< PSC(out) core \*/  "imp\_cnn", /\*\*< CNN core \*/  "vdsp", /\*\*< DSP \*/  "imp\_dta" /\*\*< DTA \*/  }; |
| **Description** | IMP Driver determines the OSAL resources for each core types by this table. |
| **Remarks** | N/A |

##### s\_impdrv\_irq\_channel\_table

Table 5‑100: s\_impdrv\_irq\_channel\_table

|  |  |
| --- | --- |
| **Format** | IMPDRV\_STATIC  const uint32\_t s\_impdrv\_irq\_channel\_table[IMPDRV\_IRQ\_CHANNEL\_MAX + 1U] =  {  /\*\* IRQ channel number \*/  IMPDRV\_DUMMY\_UINT64, /\*\*< Dummy for Index adjustment \*/  IMPDRV\_IRQ\_CHANNEL\_0, /\*\*< IRQ channel number (0) \*/  IMPDRV\_IRQ\_CHANNEL\_1, /\*\*< IRQ channel number (1) \*/  IMPDRV\_IRQ\_CHANNEL\_2, /\*\*< IRQ channel number (2) \*/  IMPDRV\_IRQ\_CHANNEL\_3, /\*\*< IRQ channel number (3) \*/  IMPDRV\_IRQ\_CHANNEL\_4, /\*\*< IRQ channel number (4) \*/  IMPDRV\_IRQ\_CHANNEL\_5, /\*\*< IRQ channel number (5) \*/  IMPDRV\_IRQ\_CHANNEL\_6 /\*\*< IRQ channel number (6) \*/  }; |
| **Description** | IMP Driver uses this table to convert the instance number enumeration value into an interrupt channel value. |
| **Remarks** | N/A |

# Device Specific definition

The setting parameters to the API of IMP Driver differ depending on the device (SoC).

IMP driver checks and controls the differences in the device.

Table 6‑1 shows the device differences and IMP driver check function.

Table 6‑1 Device Differences

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_Init*** | ***R\_IMPDRV\_Quit*** | ***R\_IMPDRV\_AttrInit*** |
| ***V3M*** | Initialize of the IMP Driver (IMP-Xn cores). | Uninitialize of the IMP Driver (IMP-Xn cores). | Initialize of the Attribute data. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the cores and instances that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type to be initialized is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number to be initialized is supported by the SoC.  - impdrv\_cmnctl\_chk\_instance\_num  Check if the instance to be initialized is supported by the SoC.  There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_hwrsc\_open  Open only the available hardware resources.  [Device Specific definition]  - s\_hwrsc\_def\_table  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.4 Common Control  - 5.4.1.5 Core Control | There are differences regarding the cores and instances that the device supports.  [Check functions]  None.  (Because it is checked at the timing of R\_IMPDRV\_Init)  There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_hwrsc\_close  Close only the available hardware resources.  [Device Specific definition]  - s\_hwrsc\_def\_table  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.4 Common Control | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_AttrSetCoreMemInit*** | ***R\_IMPDRV\_Start*** | ***R\_IMPDRV\_Stop*** |
| ***V3M*** | Set whether to initialize core memory. | Start of the specific core on IMP-Xn. | Stop of the specific core on IMP-Xn. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.5 Core Control | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_AttrSetCoreMap*** | ***R\_IMPDRV\_AttrSetCl*** | ***R\_IMPDRV\_AttrSetIrqMask*** |
| ***V3M*** | Set of the Core map. | Set of the CL physical address. | Set of the IRQ mask. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  There are differences regarding the cores in the core map that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_syncc\_val  Check if the cores in the core map are supported by the SoC. | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_SetIrqGroup*** | ***R\_IMPDRV\_Execute*** | ***R\_IMPDRV\_ResumeExecution*** |
| ***V3M*** | Set of the IRQ grouping. | Execute of the specific core on IMP-Xn. | Resumes execute of the specific core on IMP-Xn. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  There are differences in the supported irq group.  [Check functions]  - impdrv\_cmnctl\_chk\_group\_data  Check if the irq group is supported by the SoC and instance.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.4 Common Control | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  There is a difference between the interrupts supported by the device.  [Check functions]  - impdrv\_\*\*\*ctl\_get\_inten\_val  Ignore interrupts that are not supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.5 Core Control | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  There is a difference between the interrupts supported by the device.  [Check functions]  - impdrv\_\*\*\*ctl\_get\_inten\_val  Ignore interrupts that are not supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.5 Core Control |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_GetVersion*** | ***R\_IMPDRV\_SetPmPolicy*** | ***R\_IMPDRV\_GetPmPolicy*** |
| ***V3M*** | Get the software version of IMP Driver. | Set of the power management policy. | Get of the power management policy. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | Not use | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_RegGetHwInfo*** | ***R\_IMPDRV\_RegRequired*** | ***R\_IMPDRV\_RegRead32*** |
| ***V3M*** | Get the Register area information of Specified device ID. | Require access to the Specified device ID. | Read 32-bit data from the Specified device ID. |
| ***V3H*** | (Same as above) | (Same as above) | (Same as above) |
| ***V3H\_2*** | (Same as above) | (Same as above) | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_find\_resource  Check if the hardware resource is supported by the SoC. | There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_find\_resource  Check if the hardware resource is supported by the SoC. | There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_find\_resource  Check if the hardware resource is supported by the SoC. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_RegWrite32*** | ***R\_IMPDRV\_SetMemPotect*** | ***R\_IMPDRV\_BusIfCheck*** |
| ***V3M*** | Write 32-bit data to the Specified device ID. | Not support | Check the internal bus interface at performing DMA transfer. |
| ***V3H*** | (Same as above) | Not support | (Same as above) |
| ***V3H\_2*** | (Same as above) | Set the memory protection function. | (Same as above) |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the hardware resources that the device supports.  [Check functions]  - impdrv\_cmnctl\_reg\_find\_resource  Check if the hardware resource is supported by the SoC. | There is a difference in the supported memory protect mode (EDC/ECC).  [Check functions]  - impdrv\_cmnctl\_reg\_set\_mem\_protect  Check if the memory protect mode (ECC) is supported by the SoC. | There are differences in the core that supports Internal Bus Interface Check.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.4 Common Control  - 5.4.1.5 Core Control  - s\_transfer\_config |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Device Name*** | ***IMP Driver Function*** | | |
| ***R\_IMPDRV\_ConfRegCheck*** | ***R\_IMPDRV\_ModuleStopCheck*** | ***R\_IMPDRV\_AttrSetDsp*** |
| ***V3M*** | Check the configuration register at read back of register. | Check the Unintended module stop at read back of register. | Not support |
| ***V3H*** | (Same as above) | (Same as above) | Not support |
| ***V3H\_2*** | (Same as above) | (Same as above) | Not support |
| ***V4H*** | (Same as above) | (Same as above) | (Same as above) |
| ***Difference between devices*** | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.5 Core Control | There are differences regarding the cores that the device supports.  [Check functions]  - impdrv\_cmnctl\_get\_corectl\_func  Check if the core type is supported by the SoC.  - impdrv\_\*\*\*ctl\_is\_valid\_core  Check if the core number is supported by the SoC.  Each device has a different register address and value.  [Device Specific definition]  - 5.4.1.5 Core Control | Not support |

**Version History**

| **Date** | **Version** | **Change** | **Approver** | **Checker** | **Author** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| October XX, 2022 | 0.01M | The following is the change history from Unit Design Specification for xOS2 (LLWEB-20103556, Ver 1.00M). |  |  | HiICS |
| Table 2-1.  - Add R\_IMPDRV\_AttrSetClBrkAddr() |
| -Add 2.1.1.11 R\_IMPDRV\_AttrSetClBrkAddr |
| Table 2-33. -Add mpdrv\_genctl\_set\_cl\_brk\_addr() |
| Add 2.2.1.1 impdrv\_genctl\_set\_cl\_brk\_addr |
| Table 2-117  -Add impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr |
| Add 2.4.1.13 impdrv\_\*\*\*ctl\_set\_cl\_brk\_addrexecute |
| Add Table 5-49 Definition values (Common Control) [R-CarV4H] |
| Add Table 5-54 Definition Values (Core Control) 「R-CarV4H] |
| Add the V4H in Table 6 |
| Add V4H definition :  Table 5 83: -s\_impdrv\_core\_to\_syncc\_table  -Table 5 85: s\_hwrsc\_def\_table  -Table 5 87: s\_impdrv\_used\_instance  -Table 5 89: s\_impdrv\_cnv\_ins\_table  -Table 5 98: |
| Add comment which have no definition in V4H :  -Table 5 88: s\_impdrv\_inten\_group\_table  -Table 5 90: s\_mb\_init\_info1  -Table 5 91: s\_mb\_init\_info2  -Table 5 92: s\_mb\_init\_info3  -Table 5 93: s\_test\_pattern\_tbl  -Table 5 94: s\_soft\_reset  -Table 5 95: s\_interrupt\_enable  -Table 5 96: s\_transfer\_config |
| Chapter and Item numbers are renumbered |
| May 25, 2023 | 0.02M | The following is the change history for the 6/9 release (performance improvement) |  |  | T.Ogura |
| Table 2-1  -Add R\_IMPDRV\_GetPMState |
| Add 2.1.1.16 R\_IMPDRV\_GetPMState |
| Table 2-34  -Add impdrv\_genctl\_pm\_get\_state |
| Add 2.2.1.10 impdrv\_genctl\_pm\_get\_state |
| Table 2-119  -Add impdrv\_\*\*\*ctl\_quit  -Add impdrv\_\*\*\*ctl\_pm\_get\_state  -Add impdrv\_dspctl\_dsp\_start\_pre  -Add impdrv\_dspctl\_dsp\_update\_app  -Add impdrv\_dspctl\_load\_ptcm  -Add impdrv\_dspctl\_stop\_reg |
| Add 2.4.1.8 impdrv\_\*\*\*ctl\_quit |
| Add 2.4.1.20 impdrv\_\*\*\*ctl\_pm\_get\_state |
| Table 2-176  -Add impdrv\_osdep\_pm\_get\_state  -Add impdrv\_osdep\_dev\_open\_imp  -Add impdrv\_osdep\_dev\_open\_dsp  -Add impdrv\_osdep\_dev\_open\_dsp\_sub |
| Add 2.5.1.11 impdrv\_osdep\_pm\_get\_state |
| Add 2.5.1.26 impdrv\_osdep\_dev\_open\_imp |
| Add 2.5.1.35  impdrv\_osdep\_dev\_open\_dsp |
| Table 5-16  -Add new member (p\_impdrvCorectlPmGetState and p\_impdrvCorectlQuit)  -Add missing member (p\_impdrvCorectlSetDsp, p\_impdrvCorectlSetClBrkAddr and p\_impdrvCorectlSetGosubCond) |
| Table 5-80  -Modify due to st\_impdrv\_corectl\_func\_t updates |
| Add 5.4.2.1.12 e\_impdrv\_pm\_state\_t |
| Modify Description to add Update Information  -2.2.1.2 impdrv\_genctl\_quit  -2.2.1.18 impdrv\_genctl\_reg\_set\_mem\_protect  -2.2.2.4 impdrv\_genctl\_chk\_state\_init  -2.3.1.9 impdrv\_cmnctl\_get\_corectl\_func  -2.4.1.5 impdrv\_\*\*\*ctl\_init\_end  -2.4.1.6 impdrv\_\*\*\*ctl\_start  -2.4.1.7 impdrv\_\*\*\*ctl\_stop  -2.4.1.15 impdrv\_\*\*\*ctl\_execute  -2.4.1.18 impdrv\_\*\*\*ctl\_pm\_set\_policy  -2.4.1.19 impdrv\_dspctl\_pm\_get\_policy  -2.4.1.24 impdrv\_dspctl\_dsp\_start\_pre  -2.4.2.30 impdrv\_dspctl\_dsp\_update\_app  -2.4.2.31 impdrv\_dspctl\_load\_ptcm  -2.4.2.32 impdrv\_dspctl\_stop\_reg  -2.5.1.7 impdrv\_osdep\_pow\_on\_imp  -2.5.1.9 impdrv\_osdep\_pm\_set\_policy  -2.5.1.10 impdrv\_osdep\_pm\_get\_policy  -2.5.1.36 impdrv\_osdep\_pow\_on\_imp\_dsp  -2.5.2.2 impdrv\_osdep\_pow\_on  -2.5.2.3 impdrv\_osdep\_pow\_off  -2.5.2.8 impdrv\_osdep\_dev\_open\_dsp\_sub |
| The following is the change history to fix review findings(SP17\_1) |
| Fixed Parameters In/Out information.  -2.1.1.16  -2.5.1.26  -2.5.2.8 |
| Fixed Parameters description.  -2.2.1.10  -2.4.1.24  -2.4.2.30  -2.4.2.32  -2.5.1.11  -2.5.1.26  -2.5.1.35  -2.5.2.8 |
| Fixed Description.  -2.2.1.2  -2.2.2.4  -2.4.1.5  -2.4.1.8 |
| Fixed Remarks  -2.4.1.24  -2.4.2.30  -2.4.2.31  -2.4.2.32 |
| Minor corrections (e.g. typo, Link error)  -Table 2-176  - 5.4.3.3.10 |
| Assign UD ID to the added function |
| Aug 10, 2023 | 0.03M | Update Activity Diagrams of Chapter 3 due to modify in 0.02M. | S.Hirosawa | H.Matsunaga | T.Ogura |
| Feb 16,2024 | 0.04M | Update Activity Diagrams of Chapter 3 due to modify in xOS3.[V3H1,V3H2,V3M2,V4H2] |  |  | HiICS |
| Modify due to st\_impdrv\_corectl\_func\_t updates.  - Table 5 16  - Table 5 82 |
| Update Data Design  -Table 5-42  -Table 5-56  -Table 5-60 |
| Add Software Unit Design  - R\_IMPDRV\_AttrSetDsp  - R\_IMPDRV\_AttrSetGosubCond  - impdrv\_genctl\_set\_dsp  - impdrv\_genctl\_dsp\_int\_handler  - impdrv\_genctl\_set\_gosub\_cond  - impdrv\_cmnctl\_get\_init\_param  - impdrv\_cmnctl\_get\_dsp\_func  - impdrv\_dspctl\_set\_dsp  - impdrv\_\*\*\*ctl\_set\_cond\_gosub  - impdrv\_\*\*\*ctl\_set\_extend\_config  - impdrv\_dspctl\_init\_dsp  - impdrv\_dspctl\_get\_device\_io  - impdrv\_osdep\_register\_dsp\_irq  - impdrv\_osdep\_enable\_dsp\_irq  - impdrv\_osdep\_disable\_dsp\_irq  - impdrv\_osdep\_unregister\_dsp\_irq  - impdrv\_osdep\_reset\_core\_dsp  - impdrv\_osdep\_get\_dsp\_irq\_channel  - impdrv\_osdep\_dsp0\_int\_handler  - impdrv\_osdep\_dsp1\_int\_handler  - impdrv\_osdep\_dsp2\_int\_handler  - impdrv\_osdep\_dsp3\_int\_handler  - impdrv\_osdep\_wait\_for\_req\_state  Update Software Unit Design  - R\_IMPDRV\_Quit  - R\_IMPDRV\_AttrSetIRQMask  - R\_IMPDRV\_AttrSetClBrkAddr  - R\_IMPDRV\_RegGetHwInfo  - impdrv\_api\_chk\_init\_data  - impdrv\_api\_chk\_core\_info  - impdrv\_api\_chk\_conf\_reg\_check  - impdrv\_genctl\_init  - impdrv\_genctl\_quit  - impdrv\_genctl\_start  - impdrv\_genctl\_stop  - impdrv\_genctl\_set\_cl\_brk\_addr  - impdrv\_genctl\_resume\_exe  - impdrv\_genctl\_set\_core\_mem\_init  - impdrv\_genctl\_set\_irq\_mask  - impdrv\_genctl\_set\_irq\_group  - impdrv\_genctl\_reg\_get\_hw\_info  - impdrv\_genctl\_reg\_read32  - impdrv\_genctl\_reg\_write32  - impdrv\_genctl\_reg\_required  - impdrv\_genctl\_prologue  - impdrv\_genctl\_chk\_state\_init  - impdrv\_genctl\_chk\_state\_uninit  - impdrv\_genctl\_set\_state\_uninit  - impdrv\_genctl\_core\_init  - impdrv\_genctl\_init\_get\_func  - impdrv\_genctl\_init\_set\_state  - impdrv\_genctl\_init\_attr\_init  - impdrv\_cmnctl\_chk\_core\_info  - impdrv\_cmnctl\_chk\_instance\_num  - impdrv\_cmnctl\_cnv\_int\_bit\_core  - impdrv\_ocvctl\_irq\_status\_clear  - impdrv\_ocvctl\_check\_inten\_1st  - impdrv\_ocvctl\_check\_inten\_2nd  - impdrv\_osdep\_mutex\_unlock  - impdrv\_osdep\_pow\_on\_hwrsc  - impdrv\_osdep\_queue\_send\_period  - impdrv\_osdep\_pow\_off  Update References |
| Feb 29,2024 | 0.05M | Add Software Unit Design  - impdrv\_cmnctl\_reg\_spm\_preset  - impdrv\_cmnctl\_reg\_spm\_initialize  - impdrv\_cmnctl\_reg\_spm\_ena\_access  - impdrv\_cmnctl\_reg\_spm\_wait\_init  - impdrv\_cmnctl\_reg\_spm\_chk\_init  - impdrv\_dspctl\_init\_core  - impdrv\_dspctl\_chk\_execute\_data  - impdrv\_dspctl\_dsp\_execute\_pre  - impdrv\_dspctl\_tcm\_config\_d  - impdrv\_dspctl\_tcm\_config\_p  - impdrv\_dspctl\_load\_dtcm  - impdrv\_dspctl\_dsp\_standby  - impdrv\_dspctl\_dsp\_foece\_standby  - impdrv\_dspctl\_chk\_int\_data  Update Software Unit Design  - impdrv\_cmnctl\_init  - impdrv\_cmnctl\_quit  - impdrv\_cmnctl\_judge\_int  - impdrv\_cmnctl\_mutex\_destroy  - impdrv\_cmnctl\_mutex\_lock  - impdrv\_cmnctl\_get\_corectl\_func  - impdrv\_cmnctl\_get\_syncc\_val  - impdrv\_cmnctl\_reg\_init  - impdrv\_cmnctl\_reg\_quit  - impdrv\_cmnctl\_reg\_set\_mem\_protect  - impdrv\_cmnctl\_reg\_mem\_init  - impdrv\_cmnctl\_reg\_read32  - impdrv\_cmnctl\_reg\_write32  - impdrv\_cmnctl\_reg\_required  - impdrv\_cmnctl\_save\_irq\_mask  - impdrv\_cmnctl\_force\_irq\_mask  - impdrv\_cmnctl\_restore\_irq\_mask  - impdrv\_cmnctl\_conf\_reg\_check  - impdrv\_cmnctl\_reg\_chk\_mem\_protect  - impdrv\_cmnctl\_cnv\_int\_core\_bit  - impdrv\_cmnctl\_execute\_no\_group  - impdrv\_cmnctl\_execute\_group  - impdrv\_cmnctl\_chk\_init\_data  - impdrv\_cmnctl\_chk\_group\_data  - impdrv\_cmnctl\_calc\_checksum  - impdrv\_cmnctl\_reg\_mb\_initialize  - impdrv\_\*\*\*ctl\_init\_end  - impdrv\_\*\*\*ctl\_set\_core\_map  - impdrv\_\*\*\*ctl\_set\_cl\_brk\_addr  - impdrv\_\*\*\*ctl\_conf\_reg\_check  - impdrv\_\*\*\*ctl\_soft\_reset  - impdrv\_\*\*\*ctl\_set\_syncc\_config  - impdrv\_\*\*\*ctl\_set\_syncc\_config (\*\*\*:dma/dmas)  - impdrv\_\*\*\*ctl\_int\_cb  - impdrv\_\*\*\*ctl\_int\_cb (\*\*\*:imp/imps)  - impdrv\_\*\*\*ctl\_check\_inten  Add DSP Core portion of Software Unit Design  - impdrv\_\*\*\*ctl\_is\_valid\_core  - impdrv\_\*\*\*ctl\_check\_state  - impdrv\_\*\*\*ctl\_init\_start  - impdrv\_\*\*\*ctl\_stop  - impdrv\_\*\*\*ctl\_attr\_init  - impdrv\_\*\*\*ctl\_int\_handler  - impdrv\_\*\*\*ctl\_resume\_exe |  |  | HiICS |
| Mar, 11, 2024 | 0.06M | Added Software Unit Design  - impdrv\_ocvctl\_init\_core  - impdrv\_\*\*\*ctl\_stop(\*\*\*:dma/dmas)  Fixed Software Unit Design  Fixed description  - impdrv\_\*\*\*ctl\_stop  - impdrv\_\*\*\*ctl\_init\_core  - impdrv\_dmactl\_mb\_init  Fixed Reentrancy  - impdrv\_cnnctl\_get\_inten\_val  - impdrv\_dmactl\_get\_inten\_val  - impdrv\_dmasctl\_get\_inten\_val  - impdrv\_impctl\_get\_inten\_val  - impdrv\_impsctl\_get\_inten\_val  - impdrv\_ocvctl\_get\_inten\_val  - impdrv\_pscctl\_get\_inten\_val  - impdrv\_dmactl\_mb\_init\_main  - impdrv\_dmactl\_mb\_init\_pre  - impdrv\_dmactl\_tbl\_write\_reg  - impdrv\_\*\*\*ctl\_tbl\_write\_reg  - impdrv\_dmactl\_cl\_pre  - impdrv\_dmactl\_mb\_init  Fixed Parameters  - impdrv\_\*\*\*ctl\_int\_main\_func  - impdrv\_dmactl\_mb\_init\_main  - impdrv\_dmactl\_mb\_init\_pre  - impdrv\_dmactl\_tbl\_write\_reg  - impdrv\_dmactl\_cl\_pre  - impdrv\_dmactl\_mb\_init |  |  | KSK |
| Mar, 13, 2024 | Update Data Design  - 5.4.1 Definition  Table 5-43: Definition Values (Common definition)  Table 5-45: Definition Values (Common Control) [R-CarV3M / V3H / V3H\_2]  Table 5-49: Definition values (Common Control) [R-CarV4H]  Table 5-50: Definition Values (Core Control) [R-CarV3M / V3H / V3H\_2]  Table 5-51: Definition Values (Core Control)  Table 5-52: Definition Values (Core Control) [R-CarV3M]  Table 5-53: Definition Values (Core Control) [R-CarV3H]  Table 5-54: Definition Values (Core Control) [R-CarV3H\_2]  Table 5-55 Definition Values (Core Control) [R-CarV4H]  Table 5-56: Definition Values (OS Dependence Layer)  - 5.4.2 Enum  Table 5-69: e\_impdrv\_gosub\_cond\_t  Table 5-72: e\_impdrv\_dspctl\_load\_type\_t |
| Mar, 15, 2024 | Update References  Added to 2.Software Unit Design  - impdrv\_udefctl\_udef\_crc  - impdrv\_udefctl\_set\_ecm  - impdrv\_dspctl\_dsp\_start  - impdrv\_dspctl\_dsp\_execute  - impdrv\_dspctl\_crc\_sub  Fixed Return Value  - impdrv\_dspctl\_resume\_exe  - impdrv\_dspctl\_tcm\_config\_p  - impdrv\_dspctl\_load\_dtcm  Updated the following function’s activity diagrams  -impdrv\_\*\*\*ctl\_pm\_get\_state Note(4)  -impdrv\_dspctl\_dsp\_start Note(6)  -impdrv\_dspctl\_dsp\_execute Note(5)(8)(9)(11)(12)  -impdrv\_dspctl\_stop Note(8)  -impdrv\_dspctl\_int\_handler Note(1)(6)(8)  -impdrv\_dspctl\_resume\_exe Note(8)  -impdrv\_dspctl\_load\_ptcm Note(4)  -impdrv\_dspctl\_tcm\_config\_p Note(7)(8)(9)(10)  -impdrv\_dspctl\_load\_dtcm Note(6)(8)(9)(10)(11)(12)  -impdrv\_dspctl\_stop\_reg Note(5)  -impdrv\_dspctl\_dsp\_foece\_standby Note(3)  -impdrv\_dspctl\_crc\_sub (New)  Update Data Design  -5.4.1 Definition  Table 5-49: Definition Values (Common Control) [R-CarV4H]  Table 5-54: Definition Values (Core Control) [R-CarV4H] |
| Mar, 19, 2024 | Added Software Unit Design  - impdrv\_genctl\_get\_worksize |
| Mar 19, 2024 | 0.07M | Fixed for PR findings(LLWEB-20186048)  2.1.3.1 impdrv\_udefctl\_udef\_crc  2.1.3.2 impdrv\_udefctl\_set\_ecm  2.4.1.28 impdrv\_dspctl\_dsp\_start  - Corrected Range.  2.4.1.29 impdrv\_dspctl\_dsp\_execute  2.4.2.48 impdrv\_dspctl\_crc\_sub  - Corrected Range, Interrupt State.  3　Activity Diagrams  Remove disable ECM from impdrv\_dspctl\_dsp\_execute Note(9).  5.4.1 Definition  - Corrected value of IMPDRV\_REGVAL\_VDSP\_ERROR |  |  | KSK |