**Document Type: Detail Design**

**Document Name:**

**ADSP FRAMEWORK: PDMA DRIVER**

Renesas Electronics Corporation

Renesas Design Vietnam Co. Ltd

R-Car Software Solution 2 Group

Middleware 1 Team

Table of Contents

[1. Overview 3](#_Toc532311025)

[2. Function list 4](#_Toc532311026)

[3. Detail information 5](#_Toc532311027)

[3.1 Data type definition 5](#_Toc532311028)

[3.1.1 PDMA\_CH 5](#_Toc532311029)

[3.1.2 pdma\_error\_code 6](#_Toc532311030)

[3.1.3 PDMA\_SRCDST 6](#_Toc532311031)

[3.1.4 PDMA\_REG\_SETTING 10](#_Toc532311032)

[3.2 Macro definition 11](#_Toc532311033)

[3.3 Global variable definition 12](#_Toc532311034)

[3.4 Function definition 16](#_Toc532311035)

[3.4.1 get\_pdma\_regs 16](#_Toc532311036)

[3.4.2 set\_pdma\_regs 17](#_Toc532311037)

[3.4.3 set\_pdma\_extend 18](#_Toc532311038)

[3.4.4 enable\_pdma\_extend 20](#_Toc532311039)

[3.4.5 disable\_pdma\_extend 22](#_Toc532311040)

[3.4.6 check\_available\_pdma\_channel 23](#_Toc532311041)

[4. Revision history 25](#_Toc532311042)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc532311043)

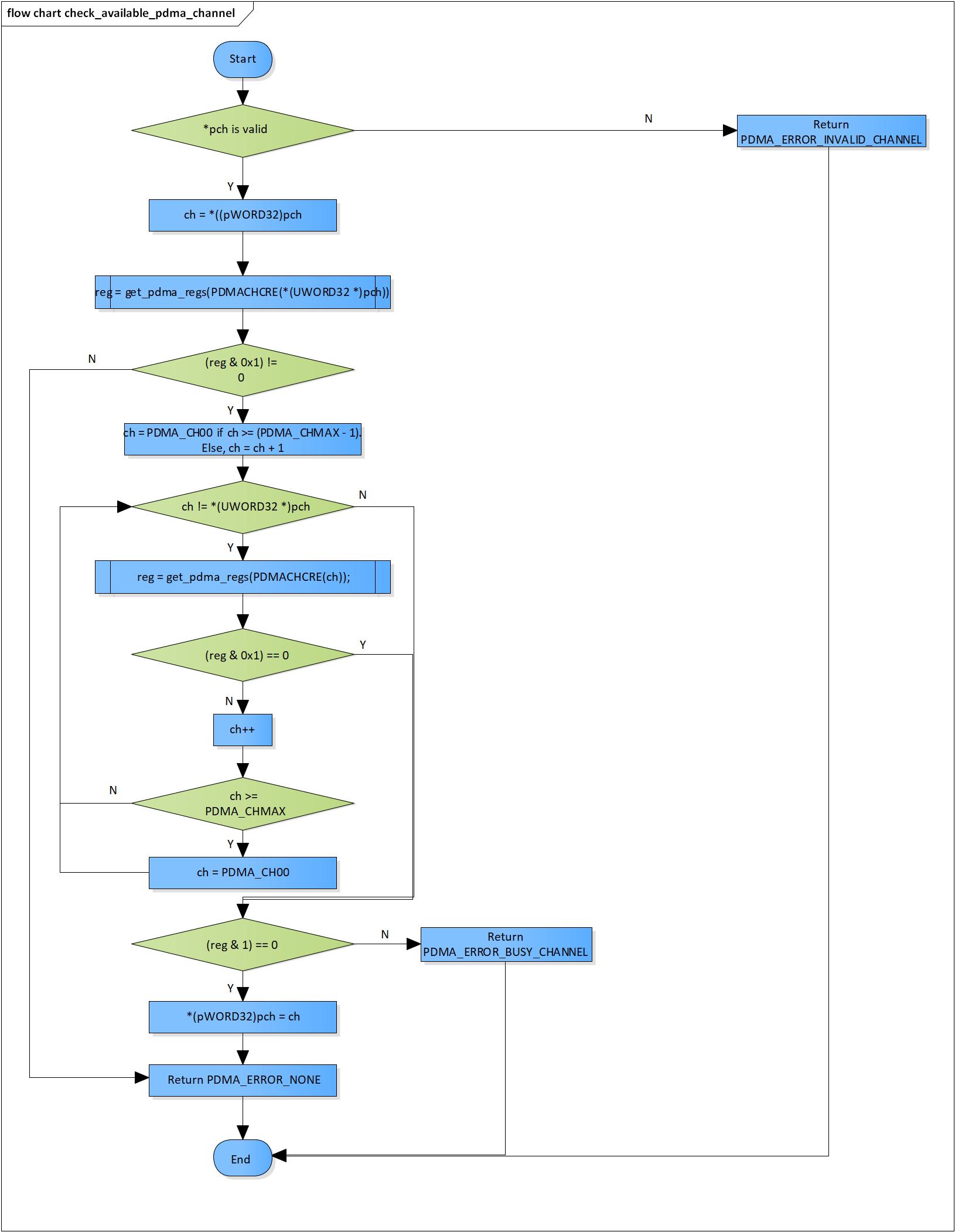
[Figure 3‑1 get\_pdma\_regs flowchart 16](#_Toc532311044)

[Figure 3‑2 set\_pdma\_regs flowchart 17](#_Toc532311045)

[Figure 3‑3 set\_pdma\_extend flowchart 19](#_Toc532311046)

[Figure 3‑5 enable\_pdma\_extend flowchart 20](#_Toc532311047)

[Figure 3‑6 disable\_pdma\_extend flowchart 22](#_Toc532311048)

[Figure 3‑7 check\_available\_pdma\_channel flowchart 24](#_Toc532311049)

List of Table

[Table 2‑1 Function list 4](#_Toc532311050)

[Table 3‑1 PDMA\_CH type information 5](#_Toc532311051)

[Table 3‑2 pdma\_error\_code type information 6](#_Toc532311052)

[Table 3‑3 PDMA\_SRCDST type information 6](#_Toc532311053)

[Table 3‑4 PDMA\_REG\_SETTING type information 10](#_Toc532311054)

[Table 2‑1 Function list 4](#_Toc532311055)

[Table 3‑1 PDMA\_CH type information 5](#_Toc532311056)

[Table 3‑2 pdma\_error\_code type information 6](#_Toc532311057)

[Table 3‑3 PDMA\_SRCDST type information 6](#_Toc532311058)

[Table 3‑4 PDMA\_REG\_SETTING type information 10](#_Toc532311059)

# Overview

This section provides an overview of software architecture.

User Application

ADSP Interface

Kernel Space

User Space

ADSP Driver

ARM

Audio HW

**ADSP Framework**

TDM class

Renderer class

Equalizer class

Capture class

TDM Plugin

ADSP

Plugin

Equalizer Plugin\*

Capture Plugin

Renderer Plugin

This document’s target is in side of red square

\* not connect to hardware modules

DAC/

ADC

PDMA

FIFO

SCU

SSI

SSIU

PDMA

Figure 1‑1 ****The software architecture****

# Function list

The following is list of functions:

Table 2‑1 Function list

|  |  |  |
| --- | --- | --- |
| **Type** | **Function Name** | **Outline** |
| API | set\_pdma\_extend | This API is to set the registers for PDMA |
| clear\_pdma\_extend | This API is to clear the registers for PDMA |
| enable\_pdma\_extend | This API is to enable PDMA |
| disable\_pdma\_extend | This API is to disable PDMA |
| check\_available\_pdma\_channel | This API is to check availability of a PDMA channel and choose an available one |
| Internal function | get\_pdma\_regs | This function is to get a PDMA register’s value |
| set\_pdma\_regs | This function is to set a PDMA register |

# Detail information

This section describes detail information of data types, macro definitions, implemented APIs and internal function units, global variable.

## Data type definition

### PDMA\_CH

The data type PDMA\_CH is a type-defined enumeration that lists all supported PDMA channels.

Table 3‑1 PDMA\_CH type information

|  |  |  |
| --- | --- | --- |
| **Member name** | **Value** | **Outline** |
| PDMA\_CH00 | 0 | PDMA channel 0 |
| PDMA\_CH01 | 1 | PDMA channel 1 |
| PDMA\_CH02 | 2 | PDMA channel 2 |
| PDMA\_CH03 | 3 | PDMA channel 3 |
| PDMA\_CH04 | 4 | PDMA channel 4 |
| PDMA\_CH05 | 5 | PDMA channel 5 |
| PDMA\_CH06 | 6 | PDMA channel 6 |
| PDMA\_CH07 | 7 | PDMA channel 7 |
| PDMA\_CH08 | 8 | PDMA channel 8 |
| PDMA\_CH09 | 9 | PDMA channel 9 |
| PDMA\_CH10 | 10 | PDMA channel 10 |
| PDMA\_CH11 | 11 | PDMA channel 11 |
| PDMA\_CH12 | 12 | PDMA channel 12 |
| PDMA\_CH13 | 13 | PDMA channel 13 |
| PDMA\_CH14 | 14 | PDMA channel 14 |
| PDMA\_CH15 | 15 | PDMA channel 15 |
| PDMA\_CH16 | 16 | PDMA channel 16 |
| PDMA\_CH17 | 17 | PDMA channel 17 |
| PDMA\_CH18 | 18 | PDMA channel 18 |
| PDMA\_CH19 | 19 | PDMA channel 19 |
| PDMA\_CH20 | 20 | PDMA channel 20 |
| PDMA\_CH21 | 21 | PDMA channel 21 |
| PDMA\_CH22 | 22 | PDMA channel 22 |
| PDMA\_CH23 | 23 | PDMA channel 23 |
| PDMA\_CH24 | 24 | PDMA channel 24 |
| PDMA\_CH25 | 25 | PDMA channel 25 |
| PDMA\_CH26 | 26 | PDMA channel 26 |
| PDMA\_CH27 | 27 | PDMA channel 27 |
| PDMA\_CH28 | 28 | PDMA channel 28 |
| PDMA\_CHMAX | 29 | Total number of PDMA channels |

### pdma\_error\_code

The data type pdma\_error\_code is a type-defined enumeration that lists all error codes for PDMA.

Table 3‑2 pdma\_error\_code type information

|  |  |
| --- | --- |
| **Member name** | **Value** |
| PDMA\_ERROR\_NONE | 0 |
| PDMA\_ERROR\_INVALID\_CHANNEL | -1 |
| PDMA\_ERROR\_INVALID\_SOURCE | -2 |
| PDMA\_ERROR\_INVALID\_DESTINATION | -3 |
| PDMA\_ERROR\_BUSY\_CHANNEL | -4 |

### PDMA\_SRCDST

The data type PDMA\_SRCDST is a type-defined enumeration that shows connections between PDMA and other devices.

Table 3‑3 PDMA\_SRCDST type information

|  |  |
| --- | --- |
| **Member name** | **Value** |
| PDMA\_SSI00 | SSI\_MODULE\_MIN |
| PDMA\_SSI01 | 1 |
| PDMA\_SSI02 | 2 |
| PDMA\_SSI03 | 3 |
| PDMA\_SSI04 | 4 |
| PDMA\_SSI05 | 5 |
| PDMA\_SSI06 | 6 |
| PDMA\_SSI07 | 7 |
| PDMA\_SSI10 | 10 |
| PDMA\_SSI11 | 11 |
| PDMA\_SSI12 | 12 |
| PDMA\_SSI13 | 13 |
| PDMA\_SSI14 | 14 |
| PDMA\_SSI15 | 15 |
| PDMA\_SSI16 | 16 |
| PDMA\_SSI17 | 17 |
| PDMA\_SSI20 | 20 |
| PDMA\_SSI21 | 21 |
| PDMA\_SSI22 | 22 |
| PDMA\_SSI23 | 23 |
| PDMA\_SSI24 | 24 |
| PDMA\_SSI25 | 25 |
| PDMA\_SSI26 | 26 |
| PDMA\_SSI27 | 27 |
| PDMA\_SSI30 | 30 |
| PDMA\_SSI31 | 31 |
| PDMA\_SSI32 | 32 |
| PDMA\_SSI33 | 33 |
| PDMA\_SSI34 | 34 |
| PDMA\_SSI35 | 35 |
| PDMA\_SSI36 | 36 |
| PDMA\_SSI37 | 37 |
| PDMA\_SSI40 | 40 |
| PDMA\_SSI41 | 41 |
| PDMA\_SSI42 | 42 |
| PDMA\_SSI43 | 43 |
| PDMA\_SSI44 | 44 |
| PDMA\_SSI45 | 45 |
| PDMA\_SSI46 | 46 |
| PDMA\_SSI47 | 47 |
| PDMA\_SSI5 | 50 |
| PDMA\_SSI6 | 60 |
| PDMA\_SSI7 | 70 |
| PDMA\_SSI8 | 80 |
| PDMA\_SSI90 | 90 |
| PDMA\_SSI91 | 91 |
| PDMA\_SSI92 | 92 |
| PDMA\_SSI93 | 93 |
| PDMA\_SSI94 | 94 |
| PDMA\_SSI95 | 95 |
| PDMA\_SSI96 | 96 |
| PDMA\_SSI97 | SSI\_MODULE\_MAX |
| PDMA\_DTCPPP0 | 98 |
| PDMA\_DTCPPP1 | 99 |
| PDMA\_DTCPCP0 | 100 |
| PDMA\_DTCPCP1 | 101 |
| PDMA\_ADSPO0 | 102 |
| PDMA\_ADSPI0 | 103 |
| PDMA\_ADSPO1 | 104 |
| PDMA\_ADSPI1 | 105 |
| PDMA\_ADSPO2 | 106 |
| PDMA\_ADSPI2 | 107 |
| PDMA\_ADSPO3 | 108 |
| PDMA\_ADSPI3 | 109 |
| PDMA\_SCU\_SRCI0 | SCU\_SRC\_INPUT\_MODULE\_MIN |
| PDMA\_SCU\_SRCI1 | 111 |
| PDMA\_SCU\_SRCI2 | 112 |
| PDMA\_SCU\_SRCI3 | 113 |
| PDMA\_SCU\_SRCI4 | 114 |
| PDMA\_SCU\_SRCI5 | 115 |
| PDMA\_SCU\_SRCI6 | 116 |
| PDMA\_SCU\_SRCI7 | 117 |
| PDMA\_SCU\_SRCI8 | 118 |
| PDMA\_SCU\_SRCI9 | SCU\_SRC\_INPUT\_MODULE\_MAX |
| PDMA\_SCU\_SRCO0 | SCU\_SRC\_OUTPUT\_MODULE\_MIN |
| PDMA\_SCU\_SRCO1 | 121 |
| PDMA\_SCU\_SRCO2 | 122 |
| PDMA\_SCU\_SRCO3 | 123 |
| PDMA\_SCU\_SRCO4 | 124 |
| PDMA\_SCU\_SRCO5 | 125 |
| PDMA\_SCU\_SRCO6 | 126 |
| PDMA\_SCU\_SRCO7 | 127 |
| PDMA\_SCU\_SRCO8 | 128 |
| PDMA\_SCU\_SRCO9 | SCU\_SRC\_OUTPUT\_MODULE\_MAX |
| PDMA\_SCU\_CMD0 | SCU\_CMD\_MODULE\_MIN |
| PDMA\_SCU\_CMD1 | SCU\_CMD\_MODULE\_MAX |
| PDMA\_MLM0 | 132 |
| PDMA\_MLM1 | 133 |
| PDMA\_MLM2 | 134 |
| PDMA\_MLM3 | 135 |
| PDMA\_MLM4 | 136 |
| PDMA\_MLM5 | 137 |
| PDMA\_MLM6 | 138 |
| PDMA\_MLM7 | 139 |
| PDMA\_NONCONTROL | 140 |
| PDMA\_MAX | 141 |

### PDMA\_REG\_SETTING

The data type PDMA\_REG\_SETTING is a type-defined structure that possesses necessary parameters for PDMA module.

Table 3‑4 PDMA\_REG\_SETTING type information

|  |  |  |
| --- | --- | --- |
| **Member name** | **Data type** | **Outline** |
| SARE | UWORD32 | SARE register value |
| CHCRESRS | UWORD32 | CHCRESRS register SRS bit value |
| DARE | UWORD32 | DARE register value |
| CHCREDRS | UWORD32 | CHCREDRS register DRS bit value |

## Macro definition

|  |  |  |
| --- | --- | --- |
| **Macros** | **Value** | **Outline** |
| SSI\_MODULE\_MIN | 0 | SSI module min index |
| SSI\_MODULE\_MAX | 97 | SSI module max index |
| SCU\_SRC\_INPUT\_MODULE\_MIN | 110 | SCU input module min index |
| SCU\_SRC\_INPUT\_MODULE\_MAX | 119 | SCU input module max index |
| SCU\_SRC\_OUTPUT\_MODULE\_MIN | 120 | SCU output module min index |
| SCU\_SRC\_OUTPUT\_MODULE\_MAX | 129 | SCU output module max index |
| SCU\_CMD\_MODULE\_MIN | 130 | CMD module min index |
| SCU\_CMD\_MODULE\_MAX | 131 | CMD module max index |
| PDMASARE(n) | XF\_RCAR\_REG\_PDMASARE(n) | PDMASARE Address Calculation: XF\_RCAR\_REG\_PDMASARE(n) in xf-registers.h |
| PDMADARE(n) | XF\_RCAR\_REG\_PDMADARE(n) | PDMADARE Address Calculation: XF\_RCAR\_REG\_PDMADARE(n) in xf-registers.h |
| PPDMAHCRE(n) | XF\_RCAR\_REG\_PPDMAHCRE(n) | PPDMAHCRE Address Calculation: XF\_RCAR\_REG\_PPDMAHCRE(n) in xf-registers.h |

## Global variable definition

Below is the array of struct PDMA\_REG\_SETTING-datatype pdma\_set\_value holds 141 (PDMA\_MAX) elements representing settings for PDMA data transfer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| static PDMA\_REG\_SETTING pdma\_set\_value[PDMA\_MAX] | | | | |
| Array’s index | SARE | CHCRESRS | DARE | CHCREDRS |
| PDMA\_SSI00 | 0xEC400000 | 0x00 | 0xEC400000 | 0x00 |
| PDMA\_SSI01 | 0xEC400400 | 0x01 | 0xEC400400 | 0x01 |
| PDMA\_SSI02 | 0xEC400800 | 0x02 | 0xEC400800 | 0x02 |
| PDMA\_SSI03 | 0xEC400C00 | 0x03 | 0xEC400C00 | 0x03 |
| PDMA\_SSI04 | 0xEC40A000 | 0x39 | 0xEC40A000 | 0x39 |
| PDMA\_SSI05 | 0xEC40A400 | 0x3A | 0xEC40A400 | 0x3A |
| PDMA\_SSI06 | 0xEC40A800 | 0x3B | 0xEC40A800 | 0x3B |
| PDMA\_SSI07 | 0xEC40AC00 | 0x3C | 0xEC40AC00 | 0x3C |
| PDMA\_SSI10 | 0xEC401000 | 0x04 | 0xEC401000 | 0x04 |
| PDMA\_SSI11 | 0xEC401400 | 0x05 | 0xEC401400 | 0x05 |
| PDMA\_SSI12 | 0xEC401800 | 0x06 | 0xEC401800 | 0x06 |
| PDMA\_SSI13 | 0xEC401C00 | 0x07 | 0xEC401C00 | 0x07 |
| PDMA\_SSI14 | 0xEC40B000 | 0x3D | 0xEC40B000 | 0x3D |
| PDMA\_SSI15 | 0xEC40B400 | 0x3E | 0xEC40B400 | 0x3E |
| PDMA\_SSI16 | 0xEC40B800 | 0x3F | 0xEC40B800 | 0x3F |
| PDMA\_SSI17 | 0xEC40BC00 | 0x40 | 0xEC40BC00 | 0x40 |
| PDMA\_SSI20 | 0xEC402000 | 0x08 | 0xEC402000 | 0x08 |
| PDMA\_SSI21 | 0xEC402400 | 0x09 | 0xEC402400 | 0x09 |
| PDMA\_SSI22 | 0xEC402800 | 0x0A | 0xEC402800 | 0x0A |
| PDMA\_SSI23 | 0xEC402C00 | 0x0B | 0xEC402C00 | 0x0B |
| PDMA\_SSI24 | 0xEC40C000 | 0x41 | 0xEC40C000 | 0x41 |
| PDMA\_SSI25 | 0xEC40C400 | 0x42 | 0xEC40C400 | 0x42 |
| PDMA\_SSI26 | 0xEC40C800 | 0x43 | 0xEC40C800 | 0x43 |
| PDMA\_SSI27 | 0xEC40CC00 | 0x44 | 0xEC40CC00 | 0x44 |
| PDMA\_SSI30 | 0xEC403000 | 0x0C | 0xEC403000 | 0x0C |
| PDMA\_SSI31 | 0xEC403400 | 0x45 | 0xEC403400 | 0x45 |
| PDMA\_SSI32 | 0xEC403800 | 0x46 | 0xEC403800 | 0x46 |
| PDMA\_SSI33 | 0xEC403C00 | 0x47 | 0xEC403C00 | 0x47 |
| PDMA\_SSI34 | 0xEC40D000 | 0x48 | 0xEC40D000 | 0x48 |
| PDMA\_SSI35 | 0xEC40D400 | 0x49 | 0xEC40D400 | 0x49 |
| PDMA\_SSI36 | 0xEC40D800 | 0x4A | 0xEC40D800 | 0x4A |
| PDMA\_SSI37 | 0xEC40DC00 | 0x4B | 0xEC40DC00 | 0x4B |
| PDMA\_SSI40 | 0xEC404000 | 0x0D | 0xEC404000 | 0x0D |
| PDMA\_SSI41 | 0xEC404400 | 0x4C | 0xEC404400 | 0x4C |
| PDMA\_SSI42 | 0xEC404800 | 0x4D | 0xEC404800 | 0x4D |
| PDMA\_SSI43 | 0xEC404C00 | 0x4E | 0xEC404C00 | 0x4E |
| PDMA\_SSI44 | 0xEC40E000 | 0x4F | 0xEC40E000 | 0x4F |
| PDMA\_SSI45 | 0xEC40E400 | 0x50 | 0xEC40E400 | 0x50 |
| PDMA\_SSI46 | 0xEC40E800 | 0x51 | 0xEC40E800 | 0x51 |
| PDMA\_SSI47 | 0xEC40EC00 | 0x52 | 0xEC40EC00 | 0x52 |
| PDMA\_SSI5 | 0xEC405000 | 0x0E | 0xEC405000 | 0x0E |
| PDMA\_SSI6 | 0xEC406000 | 0x0F | 0xEC406000 | 0x0F |
| PDMA\_SSI7 | 0xEC407000 | 0x10 | 0xEC407000 | 0x10 |
| PDMA\_SSI8 | 0xEC408000 | 0x11 | 0xEC408000 | 0x11 |
| PDMA\_SSI90 | 0xEC409000 | 0x12 | 0xEC409000 | 0x12 |
| PDMA\_SSI91 | 0xEC409400 | 0x13 | 0xEC409400 | 0x13 |
| PDMA\_SSI92 | 0xEC409800 | 0x14 | 0xEC409800 | 0x14 |
| PDMA\_SSI93 | 0xEC409C00 | 0x15 | 0xEC409C00 | 0x15 |
| PDMA\_SSI94 | 0xEC40F000 | 0x53 | 0xEC40F000 | 0x53 |
| PDMA\_SSI95 | 0xEC40F400 | 0x54 | 0xEC40F400 | 0x54 |
| PDMA\_SSI96 | 0xEC40F800 | 0x55 | 0xEC40F800 | 0x55 |
| PDMA\_SSI97 | 0xEC40FC00 | 0x56 | 0xEC40FC00 | 0x56 |
| PDMA\_DTCPPP0 | 0xEC420000 | 0x16 | 0xEC420000 | 0x16 |
| PDMA\_DTCPPP1 | 0xEC420400 | 0x17 | 0xEC420400 | 0x17 |
| PDMA\_DTCPCP0 | 0xEC420800 | 0x18 | 0xEC420800 | 0x18 |
| PDMA\_DTCPCP1 | 0xEC420C00 | 0x19 | 0xEC420C00 | 0x19 |
| PDMA\_ADSPO0 | 0xEC460000 | 0x23 | 0xff | 0xff |
| PDMA\_ADSPI0 | 0xff | 0xff | 0xEC460400 | 0x24 |
| PDMA\_ADSPO1 | 0xEC460800 | 0x57 | 0xff | 0xff |
| PDMA\_ADSPI1 | 0xff | 0xff | 0xEC460C00 | 0x58 |
| PDMA\_ADSPO2 | 0xEC461000 | 0x59 | 0xff | 0xff |
| PDMA\_ADSPI2 | 0xff | 0xff | 0xEC461400 | 0x5A |
| PDMA\_ADSPO3 | 0xEC461800 | 0x5B | 0xff | 0xff |
| PDMA\_ADSPI3 | 0xff | 0xff | 0xEC461C00 | 0x5C |
| PDMA\_SCU\_SRCI0 | 0xff | 0xff | 0xEC300000 | 0x2D |
| PDMA\_SCU\_SRCI1 | 0xff | 0xff | 0xEC300400 | 0x2E |
| PDMA\_SCU\_SRCI2 | 0xff | 0xff | 0xEC300800 | 0x2F |
| PDMA\_SCU\_SRCI3 | 0xff | 0xff | 0xEC300C00 | 0x30 |
| PDMA\_SCU\_SRCI4 | 0xff | 0xff | 0xEC301000 | 0x31 |
| PDMA\_SCU\_SRCI5 | 0xff | 0xff | 0xEC301400 | 0x32 |
| PDMA\_SCU\_SRCI6 | 0xff | 0xff | 0xEC301800 | 0x33 |
| PDMA\_SCU\_SRCI7 | 0xff | 0xff | 0xEC301C00 | 0x34 |
| PDMA\_SCU\_SRCI8 | 0xff | 0xff | 0xEC302000 | 0x35 |
| PDMA\_SCU\_SRCI9 | 0xff | 0xff | 0xEC302400 | 0x36 |
| PDMA\_SCU\_SRCO0 | 0xEC304000 | 0x2D | 0xff | 0xff |
| PDMA\_SCU\_SRCO1 | 0xEC304400 | 0x2E | 0xff | 0xff |
| PDMA\_SCU\_SRCO2 | 0xEC304800 | 0x2F | 0xff | 0xff |
| PDMA\_SCU\_SRCO3 | 0xEC304C00 | 0x30 | 0xff | 0xff |
| PDMA\_SCU\_SRCO4 | 0xEC305000 | 0x31 | 0xff | 0xff |
| PDMA\_SCU\_SRCO5 | 0xEC305400 | 0x32 | 0xff | 0xff |
| PDMA\_SCU\_SRCO6 | 0xEC305800 | 0x33 | 0xff | 0xff |
| PDMA\_SCU\_SRCO7 | 0xEC305C00 | 0x34 | 0xff | 0xff |
| PDMA\_SCU\_SRCO8 | 0xEC306000 | 0x35 | 0xff | 0xff |
| PDMA\_SCU\_SRCO9 | 0xEC306400 | 0x36 | 0xff | 0xff |
| PDMA\_SCU\_CMD0 | 0xEC308000 | 0x37 | 0xff | 0xff |
| PDMA\_SCU\_CMD1 | 0xEC308400 | 0x38 | 0xff | 0xff |
| PDMA\_MLM0 | 0xEC320000 | 0x25 | 0xEC320000 | 0x25 |
| PDMA\_MLM1 | 0xEC320400 | 0x26 | 0xEC320400 | 0x26 |
| PDMA\_MLM2 | 0xEC320800 | 0x27 | 0xEC320800 | 0x27 |
| PDMA\_MLM3 | 0xEC320C00 | 0x28 | 0xEC320C00 | 0x28 |
| PDMA\_MLM4 | 0xEC321000 | 0x29 | 0xEC321000 | 0x29 |
| PDMA\_MLM5 | 0xEC321400 | 0x2A | 0xEC321400 | 0x2A |
| PDMA\_MLM6 | 0xEC321800 | 0x2B | 0xEC321800 | 0x2B |
| PDMA\_MLM7 | 0xEC321C00 | 0x2C | 0xEC321C00 | 0x2C |

## Function definition

### get\_pdma\_regs

DD\_PLG\_RDR\_03\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline UWORD32 get\_pdma\_regs(UWORD32 reg\_addr) | | | |
| **Function** | This function is to get a PDMA register’s value. | | | |
| **Arguments** | Type | Name | I/O | Description |
| UWORD32 | reg\_addr | I | Address of a PDMA register whose value needs taking |
| **Return value** | The value of register whose address is reg\_addr | | | |
| **Description** | * pdma\_get\_regs command processing:   - Get a PDMA register’s value | | | |

[Covers: FD\_PLG\_RDR\_005, FD\_PLG\_RDR\_014, FD\_PLG\_RDR\_032, FD\_PLG\_RDR\_043]

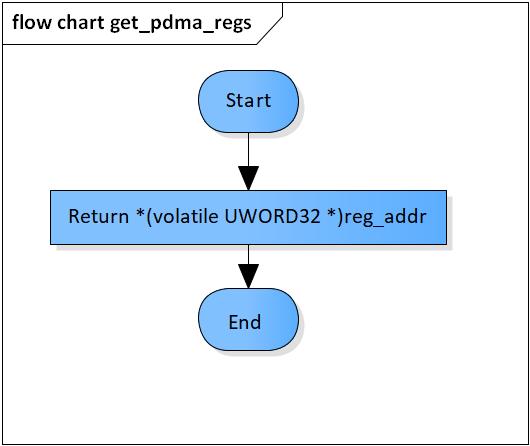


Figure 3‑1 get\_pdma\_regs flowchart

### set\_pdma\_regs

DD\_PLG\_RDR\_03\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline VOID set\_pdma\_regs(UWORD32 reg\_addr, UWORD32 value) | | | |
| **Function** | This function is to set value for register of address reg\_addr. | | | |
| **Arguments** | Type | Name | I/O | Description |
| UWORD32 | reg\_addr | I | Address of the register that needs setting |
| UWORD32 | value | I | Value to set |
| **Return value** | None | | | |
| **Description** | * pdma\_set\_stage\_number command processing:   - Set value for register whose address is reg\_addr | | | |

[Covers: FD\_PLG\_RDR\_005, FD\_PLG\_RDR\_014, FD\_PLG\_RDR\_032, FD\_PLG\_RDR\_043]

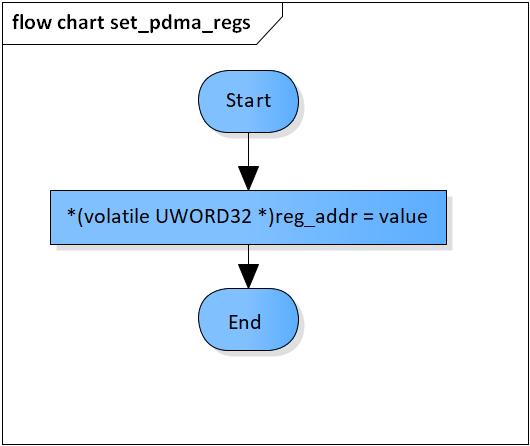


Figure 3‑2 set\_pdma\_regs flowchart

### set\_pdma\_extend

DD\_PLG\_RDR\_03\_003

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Syntax** | pdma\_error\_code set\_pdma\_extend(PDMA\_CH ch, PDMA\_SRCDST src, PDMA\_SRCDST dst) | | | | |
| **Function** | This function is to set PDMA registers. | | | | |
| **Arguments** | Type | Name | I/O | | Description |
| PDMA\_CH | ch | I | | PDMA channel  Valid range: [0: 28] |
| PDMA\_SRCDST | src | I | | Source module  Valid range: [0: 140] |
| PDMA\_SRCDST | dst | I | | Destination module  Valid range: [0: 140] |
| **Return value** | PDMA\_ERROR\_INVALID\_CHANNEL | | | Channel is invalid  Source module is invalid  Destination module is invalid | |
| PDMA\_ERROR\_INVALID\_SOURCE | | | Register SARE’s value is invalid | |
| PDMA\_ERROR\_INVALID\_DESTINATION | | | Register DARE’s value is invalid | |
| PDMA\_ERROR\_BUSY\_CHANNEL | | | Current PDMA channel is busy | |
| PDMA\_ERROR\_NONE | | | Normal end | |
| **Description** | * set\_pdma\_extend command processing:   - Set PDMA registers | | | | |

[Covers: FD\_PLG\_RDR\_005]

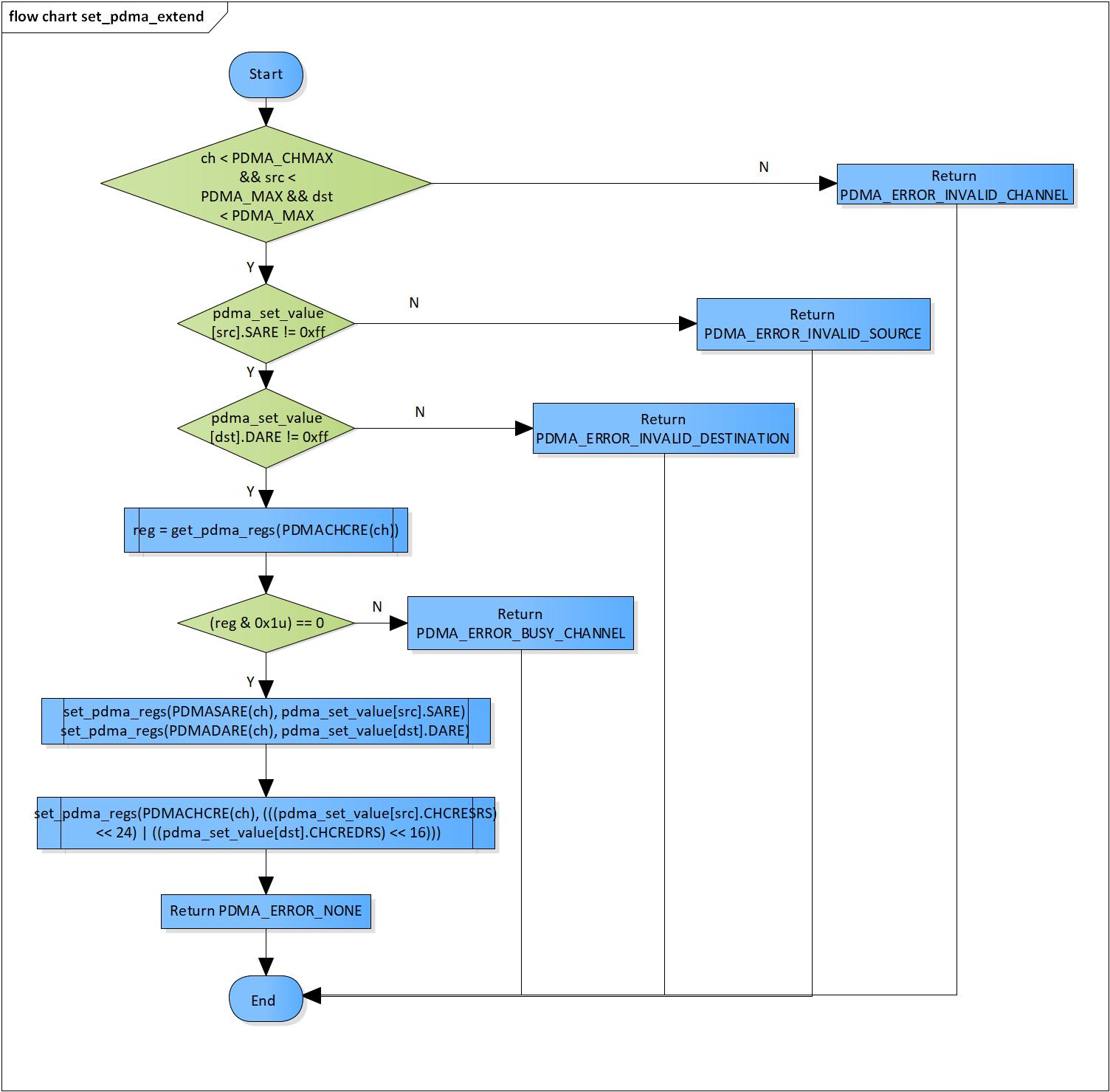


Figure 3‑3 set\_pdma\_extend flowchart

### enable\_pdma\_extend

DD\_PLG\_RDR\_03\_005

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Syntax** | pdma\_error\_code enable\_pdma\_extend(PDMA\_CH ch) | | | | |
| **Function** | This function is to enable PDMA channel ch. | | | | |
| **Arguments** | Type | Name | I/O | | Description |
| PDMA\_CH | ch | I | | PDMA channel  Valid range: [0: 28] |
| **Return value** | PDMA\_ERROR\_INVALID\_CHANNEL | | | Channel is invalid | |
| PDMA\_ERROR\_BUSY\_CHANNEL | | | Current PDMA channel is busy | |
| PDMA\_ERROR\_NONE | | | Normal end | |
| **Description** | * enable\_pdma\_extend command processing:   - Enable PDMA | | | | |

[Covers: FD\_PLG\_RDR\_005]

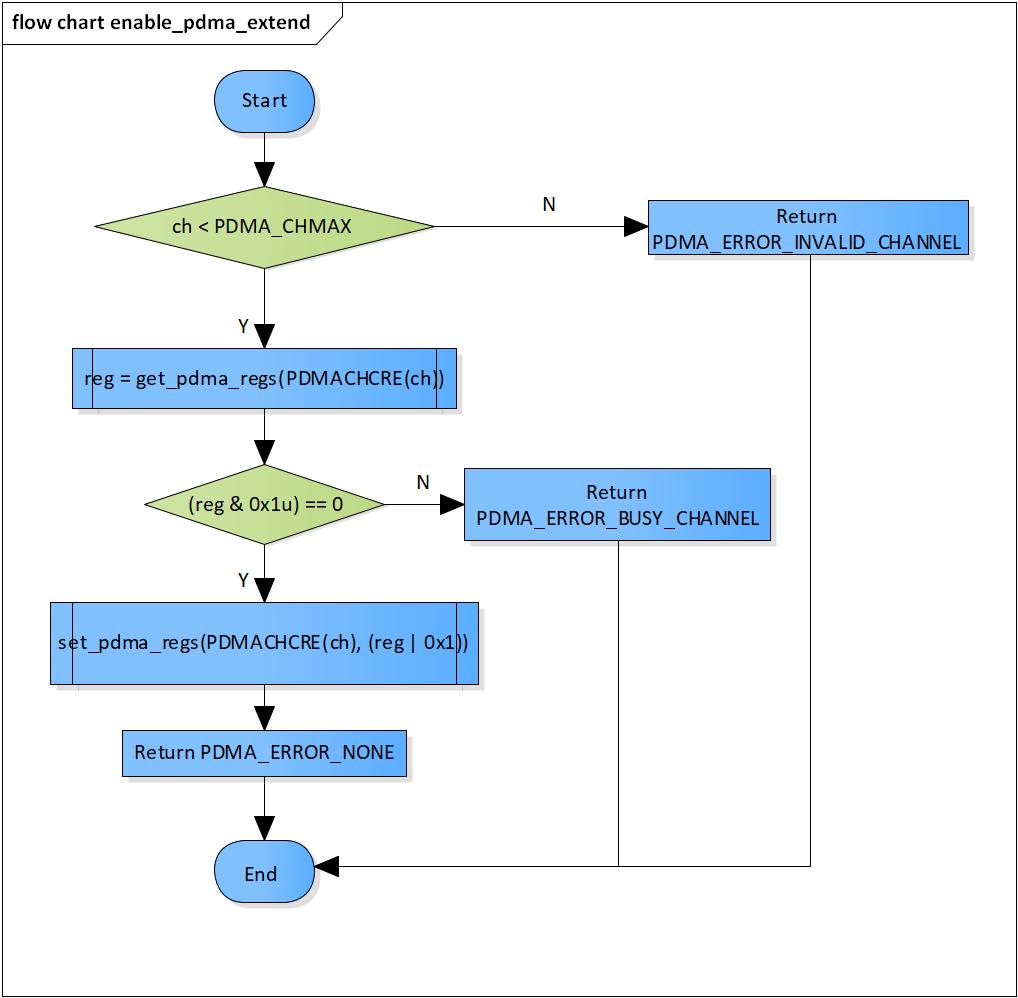


Figure 3‑4 enable\_pdma\_extend flowchart

### disable\_pdma\_extend

DD\_PLG\_RDR\_03\_006

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Syntax** | pdma\_error\_code disable\_pdma\_extend(PDMA\_CH ch) | | | | |
| **Function** | This function is to enable PDMA channel ch. | | | | |
| **Arguments** | Type | Name | I/O | | Description |
| PDMA\_CH | ch | I | | PDMA channel  Valid range: [0: 28] |
| **Return value** | PDMA\_ERROR\_INVALID\_CHANNEL | | | Channel is invalid | |
| PDMA\_ERROR\_NONE | | | Normal end | |
| **Description** | * disable\_pdma\_extend command processing:   - Disable PDMA | | | | |

[Covers: FD\_PLG\_RDR\_005, FD\_PLG\_RDR\_014, FD\_PLG\_RDR\_032, FD\_PLG\_RDR\_043]

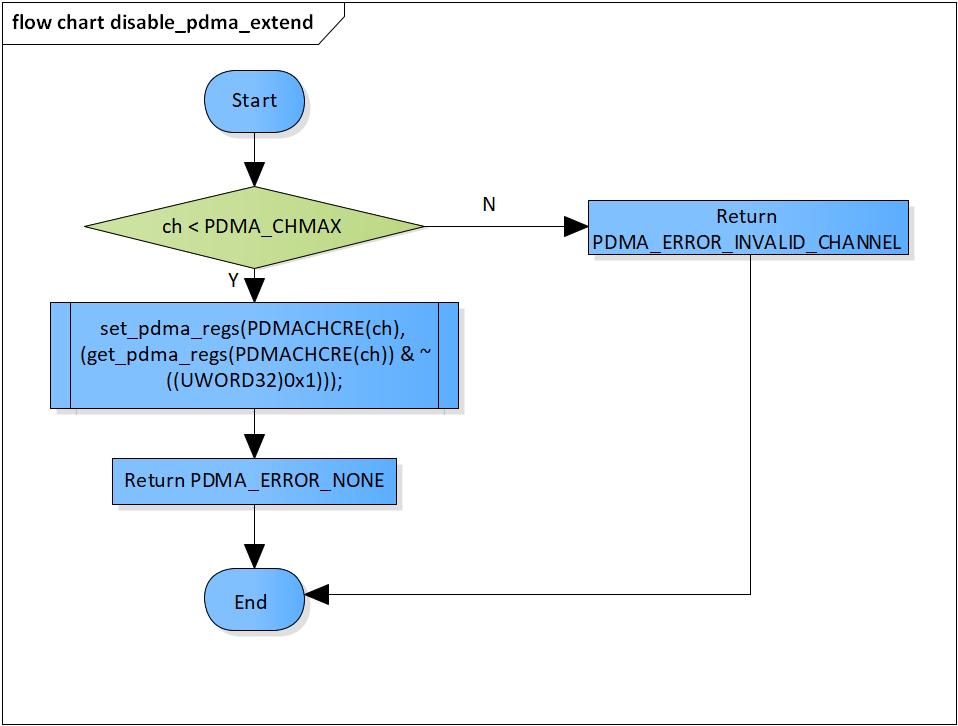


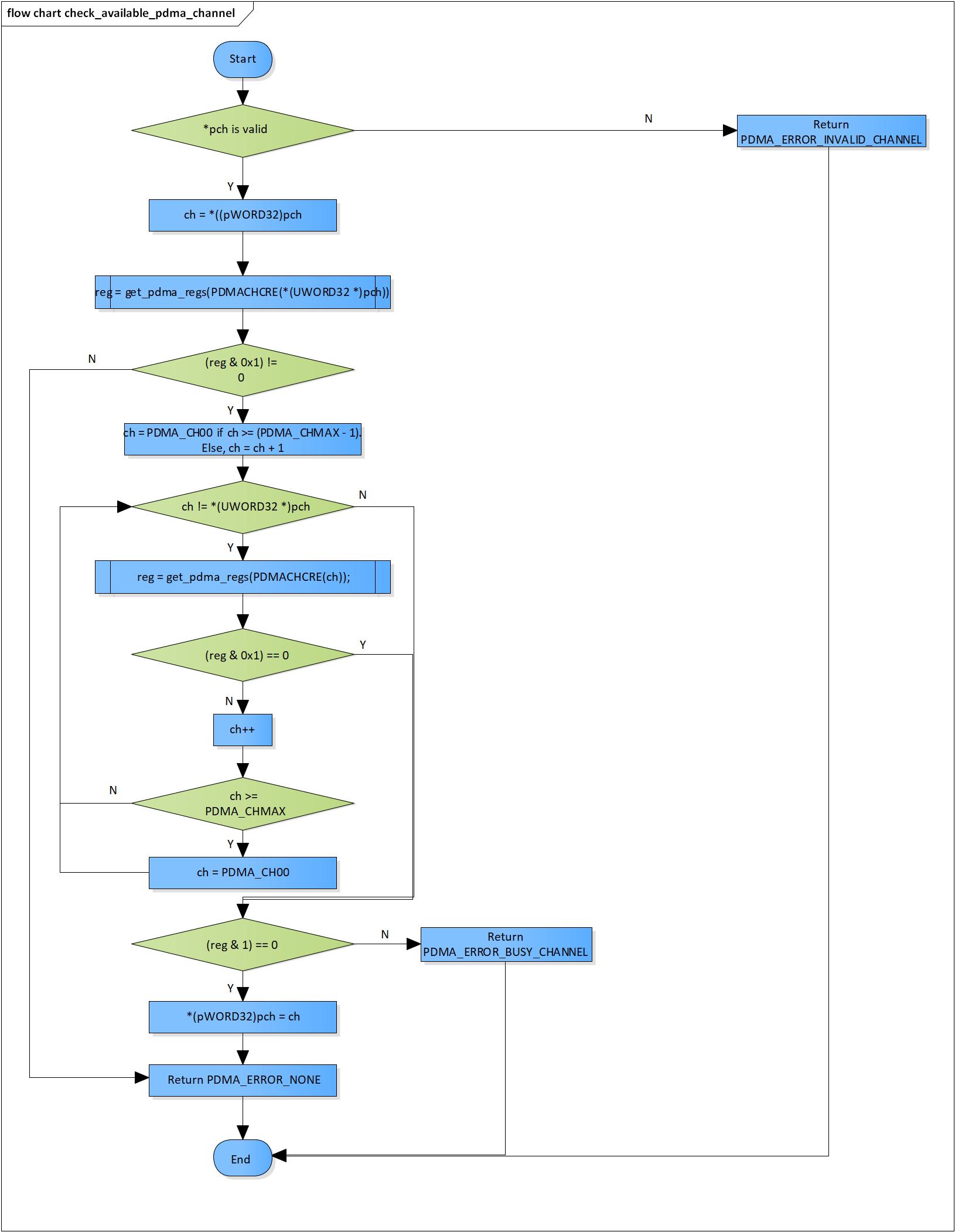
Figure 3‑5 disable\_pdma\_extend flowchart

### check\_available\_pdma\_channel

DD\_PLG\_RDR\_03\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | pdma\_error\_code check\_available\_pdma\_channel(pVOID pch) | | | |
| **Function** | This function is to check availability of PDMAs and choose an available one for operation. | | | |
| **Arguments** | Type | Name | I/O | Description |
| pVOID | pch | I/O | Pointer to channel of PDMA  Valid range: [0: 28] |
| **Return value** | PDMA\_ERROR\_INVALID\_CHANNEL | | Current PDMA channel is invalid | |
| PDMA\_ERROR\_BUSY\_CHANNEL | | No available PDMA module | |
| PDMA\_ERROR\_NONE | | Normal end | |
| **Description** | * check\_available\_pdma\_channel command processing:   - Check availability of PDMAs  - Choose an available one for operation | | | |

[Covers: FD\_PLG\_RDR\_005]

Figure 3‑6 check\_available\_pdma\_channel flowchart

# Revision history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Page** | **Content** | **Approved** | **Changed** |
| 1.0.0 | Nov 14 2018 | - | First Edition issued | Vu Phan | Vu Phan |
| 1.1.0 | Dec 10 2018 | - | Add traceability ID | Vu Phan | Ngu Pham |
| 1.2.0 | Jan 03, 2019 | - | Add range for input parameters | Vu Phan | Tien Tran |