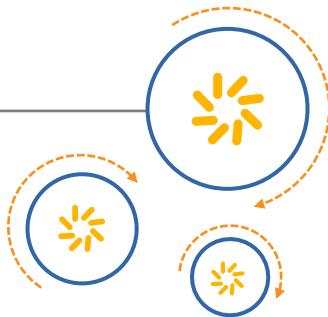




Qualcomm Technologies, Inc.



# SDM636

## Hardware Register Description

80-PD860-2X Rev. A

November 20, 2017

Qualcomm  
2019-09-18 18:26:30 PDT  
xiaoyongjie15@huajin.com

For additional information or to submit technical questions, go to <https://createpoint.qti.qualcomm.com>

**Confidential and Proprietary – Qualcomm Technologies, Inc.**

**NO PUBLIC DISCLOSURE PERMITTED:** Please report postings of this document on public servers or websites to:  
[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com).

**Restricted Distribution:** Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

MSM, Qualcomm Snapdragon, Qualcomm Snapdragon Display Engine, and Qualcomm Snapdragon Universal Bandwidth Compression are products of Qualcomm Technologies, Inc. Other Qualcomm products referenced herein are products of Qualcomm Technologies, Inc. or its subsidiaries.

Qualcomm, MSM, and Snapdragon are trademarks of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies, Inc.  
5775 Morehouse Drive  
San Diego, CA 92121  
U.S.A.

© 2017 Qualcomm Technologies, Inc. All rights reserved.

## Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at <https://createpoint.qti.qualcomm.com>. If you do not have access to the Qualcomm® CreatePoint website, register for access or send email to [support.cdmatech@qualcomm.com](mailto:support.cdmatech@qualcomm.com).

## Revision history

Bars appearing in the left margin (as shown here) indicate where technical changes have occurred for this revision. The following tables list the technical content changes for all revisions.

Revision	Date	Description
A	November 2017	Initial release

# Contents

---

<b>1 Introduction</b>	<b>4</b>
1.1 Overview .....	4
1.2 Definitions/guidelines .....	4
1.3 Register categories .....	5
1.4 Global memory map summary .....	6
1.5 Register definitions.....	6
<b>2 Security control core software interface</b>	<b>7</b>
<b>3 Top level mode multiplexing registers</b>	<b>35</b>
3.1 Top-level multiplexing .....	35
3.2 Top level mode multiplexing register .....	47
<b>4 Clock controller registers</b>	<b>325</b>
<b>5 Mobile display subsystem registers</b>	<b>353</b>
<b>6 Serial peripheral master interface registers</b>	<b>490</b>
<b>7 Universal serial bus registers</b>	<b>501</b>
<b>8 Secure digital card registers</b>	<b>621</b>
<b>9 BAM low-speed peripheral registers</b>	<b>733</b>
<b>10 Camera subsystem registers</b>	<b>1127</b>
<b>11 Low power audio subsystem registers</b>	<b>1232</b>
<b>12 LPASS digital codec core registers</b>	<b>1271</b>
<b>Index of registers</b> .....	<b>1305</b>

# 1 Introduction

---

## 1.1 Overview

This book details the SDM636 ARM-accessible registers. This chapter provides a register overview, along with the definitions and assumptions used. The remaining chapters are organized by core and provide detailed register information.

Each core's chapter includes one or two major sections listing the core's ARM registers, as well as the address for each register, its bit assignments and bit settings. The chapter also provides an overview and detailed information on the functionality of each register.

**NOTE** Addresses in this document are offsets from zero for each block.

## 1.2 Definitions/guidelines

Table 1-1 offers guidance in interpreting the register definitions and covers some of the issues and assumptions made throughout this book.

**Table 1-1 Register definitions**

Guidelines	
<b>Endian byte order</b>	Endian is a term that defines the byte order to store a sequence of bytes in memory, either MSB first (big-endian) or LSB first (little endian). Hardware coding defines MSM™ registers; in each 32-bit word, MSB = 31, LSB = 0. The MSM chips support only little endian addressing. In little endian mode, the first byte is bits 7:0, and the fourth byte is bits 31:24.
<b>Word addressing</b>	All registers are word-accessible. Assume the ARM convention: a byte, 8 bits; a half-word, 16 bits; a word, 32 bits. All accesses are expected to be word size. The ARM can do byte and half-word reads, since hardware treats a read as a word access. Individual byte accesses to a read register that triggers a hardware event triggers an event for each read performed.
<b>Active bit state</b>	Unless stated otherwise, all Boolean flags are active-high in register descriptions. Use a value of "1" to enable a function or indicate an event and a value of "0" do the opposite. Active-low flags are explicitly specified in the register descriptions. For example, if the BOOGA_ENA bit is set '1,' it enables BOOGA mode; if cleared '0,' it disables the mode.

**Table 1-1 Register definitions (cont.)**

Guidelines	
<b>Unused bits/words</b>	In the register map, many bits are marked "RESERVED" or unused. Ignore undefined bits in readable registers, software cannot assume a '0' upon a read. Write undefined bits with '0's' in writable registers. Bits marked "IGNORED" can be written with any desired value; in some cases this is convenient to software. MSM memory and register addressing simplifies implementation and allows expansion. Many holes exist in the address space. Avoid accesses to undefined memory locations, they can cause unpredictable behavior.
<b>Reset state</b>	A register's reset state is 'unknown' unless stated otherwise. It is better not to use a reset in hardware. It conserves area and simplifies coding and testing. Bits with a reset will be noted in the register descriptions. Assume that when a core comes out of reset, all its programmed register bits are either in their reset state (where defined) or unknown. Do not expect a previously programmed value to be maintained.

## 1.3 Register categories

Most registers can be categorized using one of the types listed in [Table 1-2](#). These categories are defined here to simplify the software interface.

**Table 1-2 Register categories**

Guidelines	
<b>Write command</b>	Command registers are always write only, typically there is no meaningful read-back value. Use command registers to trigger events in the hardware. Often these registers serve a single purpose. You write a bit pattern to specify the desired event. Multiple events can be combined; a register may have multiple 1-bit fields, each field triggering a hardware action, such as in an interrupt clear register. A command register can also contain encoded command fields and other data fields. The bit pattern written to it is transient; that is, there is no need to first write a value to trigger the event and another value to stop the event. If not, it will be so-stated in the register definition.
<b>Read command</b>	These registers return the desired data, but may also trigger a hardware event every time they are read. Often they are used to read out memory locations where each read triggers a read of the next memory location.
<b>Control</b>	Control registers configure the hardware to operate in a specific mode. These registers are either static or take effect at a specific time boundary (such as the next 256 chip boundary). Control bits are registered in hardware and may be readable.
<b>Status</b>	Status registers are read-only and reflect the current state of the hardware. There may be restrictions on when the status words can be reliably read, this ensures that a self-consistent value is available. Any restrictions will be indicated in the register descriptions.
<b>State</b>	State registers are read/writable by the processor but also change state due to hardware events. For software to reliably read or update a state register, special "rules of engagement" may be needed to read out a self-consistent value or avoid collisions for updates by software and hardware. Each state register has these rules described in detail.
Abbreviations	
<b>W</b>	write register
<b>R</b>	read register
<b>C</b>	command register
<b>X</b>	unknown after power-on

## 1.4 Global memory map summary

Table 1-3 shows the global memory address map.

**Table 1-3 System memory map**

End address	Start address	Size	Remarks
1FFFFFFF	80000000	6 GB	DDR
7FFFFFFF	20000000	1.5 GB	RESERVED
1FFFFFFF	1C000000	64 MB	pIMEM
1BFFFFFF	19000000	48 MB	RESERVED
18FFFFFF	18800000	8 MB	WCSS
187FFFFFF	18000000	8 MB	RESERVED
17FFFFFF	17800000	8 MB	APSS (AHB)
177FFFFFF	17000000	8 MB	RESERVED
16FFFFFF	16000000	16 MB	QDSS_STM
15FFFFFF	15C00000	4 MB	LPI (AXI TCM)
15BFFFFFF	15000000	12 MB	LPI (AHB peripherals)
14FFFFFF	14800000	8MB	RESERVED
147FFFFFF	14780000	512 KB	IPA
1477FFFFFF	146C0000	768 KB	RESERVED
146BFFFFFF	14680000	256 KB	OCIMEM
1467FFFFFF	14100000	5.5 MB	RESERVED
140FFFFFF	14000000	1 MB	L2 TCM
13FFFFFF	10000000	64 MB	Modem AXI
FFFFFFFFFF	0	256 MB	RESERVED

## 1.5 Register definitions

In the register descriptions, the addresses shown are from the perspective of the ARM.

All non command-writable registers may have a reset condition as well as a time when the write takes effect. By default, writable registers do not have a reset condition, and a value written takes effect immediately. Registers not following the default behavior use the notation below:

**Reset State:** The value this register contains following a reset.

Each register also specifies the clock regime it is on:

**Clock:** <main regime>\_<local regime>.

This information can be used by software to determine the main and/or local regime that a register is part of. Reads and writes to registers where the clock regime is disabled are not supported. The chip will not hangup, but there is no guarantee that the access takes place. Software can find where this may be an issue by enabling aborts for these types of accesses.

## 2 Security control core software interface

---

### 0x00782050 QFPROM0\_MATCH\_STATUS

**Type:** R

**Clock:** System FPB clock

**Reset State:** Undefined

This register is used to indicate the location where the fuse bit was not blown properly in QFPROM0 while blowing properly in QFPROM1. This is applicable only for the shadow regions i.e. lower 64 addresses in the 16 K wrapper (translating to lower 32 rows of each QFPROM instance).

#### QFPROM0\_MATCH\_STATUS

Bits	Name	Description
31:0	FLAG	READ ONLY. This is the QFPROM ROM #0 Match flag status. If a bit in any of the 32 locations are set to 1, it indicates an error in fuse blow for those locations.

### 0x00782054 QFPROM1\_MATCH\_STATUS

**Type:** R

**Clock:** System FPB clock

**Reset State:** Undefined

This register is used to indicate the location where the fuse bit was not blown properly in QFPROM1 while blowing properly in QFPROM0. This is applicable only for the shadow regions i.e. lower 64 addresses in the 16 K wrapper (translating to lower 32 rows of each QFPROM instance).

#### QFPROM1\_MATCH\_STATUS

Bits	Name	Description
31:0	FLAG	READ ONLY. This is the QFPROM ROM #1 Match flag status. If a bit in any of the 32 locations are set to 1, it indicates an error in fuse blow for those locations.

**0x00782058 FEC\_ESR****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

This error status register stores a notice of any fec errors or fec corrections that occur. For software accessing fuse information that has FEC enabled, this register should be checked to ensure that the fuse data is valid. To clear error or correction bits, '1' should be written to the bits that are to be cleared. The check for fec errors and corrections occurs for access to both RAW and CORR addresses.

**FEC\_ESR**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
13	CORR_SW_ACC	A '1' in this bit indicates that a FEC correction occurred on an access from the Software interface
12	RESERVED_1	
10	CORR_BOOT_ROM	A '1' in this bit indicates that a FEC correction occurred on an access from the Boot ROM Patch interface
9	CORR_FUSE_SENSE	A '1' in this bit indicates that a FEC correction occurred on an access from the Fuse Sense Controller interface
8	CORR_MULT	A '1' in this bit indicates that multiple FEC corrections have occurred since the last time the CORR_SEEN value was cleared
7	CORR_SEEN	This bit is high when a FEC correction has been detected. 0x0: NO_CORRECTION (No Correction) 0x1: CORRECTION (Correction)
6	ERR_SW_ACC	A '1' in this bit indicates that a FEC error was detected on an access from the Software interface
5	RESERVED_2	
3	ERR_BOOT_ROM	A '1' in this bit indicates that a FEC error was detected on an access from the Boot ROM Patch interface
2	ERR_FUSE_SENSE	A '1' in this bit indicates that a FEC error was detected on an access from the Fuse Sense Controller interface
1	ERR_MULT	A '1' in this bit indicates that multiple FEC errors have occurred since the last time the ERR_SEEN value was cleared
0	ERR_SEEN	This bit is high when a FEC error has been detected. 0x0: NO_ERROR (No Error) 0x1: ERROR (Error)

**0x0078205C FEC\_EAR****Type:** R**Clock:** System FPB clock**Reset State:** 0x00000000

This error address register stores the address at which the first error and correction occur. These are read only bits. Clearing the related bit in the FEC\_ESR register also clears the captured address in this register. The address refers to the row of the QFPROM, which was accessed to cause the error/correction. The check for fec errors and corrections occurs for access to both RAW and CORR addresses.

**FEC\_EAR**

Bits	Name	Description
31:16	CORR_ADDR	This is the address of the QFPROM row that first resulted in a FEC data correction. This is cleared when FEC_CORR is cleared in the FEC_ESR register.
15:0	ERR_ADDR	This is the address of the QFPROM row which first results in a FEC data error. This address is cleared when FEC_ERR is cleared in the FEC_ESR register.

**0x0078208C HW\_KEY\_STATUS****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register is used to check if the SFS Key derivation keys have been provisioned into the chips as well as whether they are currently being blocked by the settings of the chip and replaced with the dummy SFS Hardware key value. It has been extended to include various status bits upon chip bootup.

**HW\_KEY\_STATUS**

Bits	Name	Description
10	MSA_SECURE	This bit indicates whether MSA Secure status is set or not. 0x0: MSA_NOT_SECURE 0x1: MSA_SECURE
9	APPS_SECURE	This bit indicates whether Apps Secure status is set or not. 0x0: APPS_NOT_SECURE (Apps not secure) 0x1: APPS_SECURE (Apps is secure)

**HW\_KEY\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
8	KDF_AND_HW_KEY_SHIFT_DONE	This bit indicates that the Key derivation operation as well as Hardware Key Shifting operation has completed. It is recommended for the boot software to poll this bit before using the derived keys for Cryptographic operations. 0x0: KDF_AND_HW_KEY_SHIFT_IN_PROGRESS (KDF and hardware Key Shift operation in progress) 0x1: KDF_AND_HW_KEY_SHIFT_HAS_COMPLETED (KDF and hardware Key Shift operation has completed)
7	HW_KEY_SHIFT_DONE	This bit indicates that the shifting of bits on hardware Keys has completed. 0x0: HW_KEY_SHIFT_IN_PROGRESS (Hardware Key Shift operation in progress) 0x1: HW_KEY_SHIFT_HAS_COMPLETED (Hardware Key Shift operation has completed)
6	FUSE_SENSE_DONE	This bit indicates that the Key derivation operation has completed. It is recommended for the boot software to poll this bit before using the derived keys for Cryptographic operations. 0x0: FUSE_SENSE_IN_PROGRESS (KDF operation in progress) 0x1: FUSE_SENSE_HAS_COMPLETED (KDF operation has completed)
4	KDF_DONE	This bit indicates that the Key derivation operation has completed. It is recommended for the boot software to poll this bit before using the derived keys for Cryptographic operations. 0x0: KDF_OPERATION_IN_PROGRESS (KDF and key shift operation in progress) 0x1: KDF_OPERATION_HAS_COMPLETED (KDF and key shift operation has completed)
3	MSA_KEYS_BLOCKED	This bit indicates if the MSA hardware keys are currently blocked due to settings. 0x0: NOT_BLOCKED (Not Blocked) 0x1: BLOCKED (Blocked)
2	APPS_KEYS_BLOCKED	This bit indicates if the Apps hardware keys are currently blocked due to settings. 0x0: NOT_BLOCKED (Not Blocked) 0x1: BLOCKED (Blocked)

**HW\_KEY\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
1	SEC_KEY_DERIVATION_KEY_BLOWN	This bit indicates if the Secondary Key Derivation Key is provisioned. This is determined as an OR reduction of all the secondary key derivation key fuse bits. If any are blow the key is considered blown. 0x0: NOT_BLOWN (Not Blown) 0x1: BLOWN (Blown)
0	PRI_KEY_DERIVATION_KEY_BLOWN	This bit indicates if the Primary Key Derivation Key is provisioned. This is determined as an OR reduction of all the primary key derivation key fuse bits. If any are blow the key is considered blown. 0x0: NOT_BLOWN (Not Blown) 0x1: BLOWN (Blown)

**0x00782090 RESET\_JDR\_STATUS****Type:** R**Clock:** System FPB clock**Reset State:** 0x00000000

This register provides the status of the bits in the JDR chain (These bits are propagated to Reset controller to choose the correct reset triggers for sec\_ctr\_sense\_ares and acc\_ares).

**RESET\_JDR\_STATUS**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
1	FORCE_RESET	This bit provides the status of 'force_reset' bit set in the JDR chain.
0	DISABLE_SYSTEM_RESET	This bit provides the status of 'disable_system_reset' bit set in the JDR chain.

**0x00786044 OEM\_CONFIG1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the OEM configuration(63:32) bits in the QFPROM.

**OEM\_CONFIG1**

Bits	Name	Description
30		
29	DAP_SPIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_DAP_SPIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
28	APPS_SPIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_APPSPIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
27	LPASS_SPNIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_LPASS_SPNIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
25		
24	DAP_SPNIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_DAP_SPNIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
23	APPS_SPNIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_APPSPNIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
22	MSS_NIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_MSS_NIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)

**OEM\_CONFIG1 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
21	LPASS_SENSOR_NIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_LPASS_SENSOR_NIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
19		
18		
17	LPASS_NIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_RPM_LPASS_WCSS_NIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
16	DAP_NIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_DAP_NIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
15	APPS_NIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_APPS_NIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
3	DAP_DEVICEEN_DISABLE	Disables the DAP Device Enable. This OEM controlled fuse can be overridden high by the QC_DAP_DEVICEEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)
2		
1	RSVD0	Reserved.

**0x00786048 OEM\_CONFIG2****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the OEM configuration(95:64) bits in the QFPROM. This specific register contains OEM spare region secure profile configuration bits.

#### **OEM\_CONFIG2**

Bits	Name	Description
31:15	RSVD1	Reserved
9:1	RSVD0	Reserved
0	LPASS_SPIDEN_DISABLE	This OEM controlled fuse can be overridden high by the QC_LPASS_SPIDEN_DISABLE fuse. 0x0: ENABLE (Enable) 0x1: DISABLE (Disable)

#### **0x0078604C OEM\_CONFIG3**

**Type:** R

**Clock:** System FPB clock

**Reset State:** Undefined

This register allows for easy reading of the OEM configuration(119:96) bits in the QFPROM. This specific register contains OEM\_PRODUCT\_ID information.

#### **OEM\_CONFIG3**

Bits	Name	Description
31:16	OEM_PRODUCT_ID	The OEM product ID. Bits 15:0
15:0	OEM_HW_ID	The OEM hardware ID. Bits 15:0

#### **0x00786050 OEM\_CONFIG4**

**Type:** R

**Clock:** System FPB clock

**Reset State:** Undefined

This register allows for easy reading of the OEM configuration(159:128) bits in the QFPROM. This specific register contains PERIPH\_PID information.

#### **OEM\_CONFIG4**

Bits	Name	Description
31:16	PERIPH_VID	Bits 15:0 of the VID

**OEM\_CONFIG4 (cont.)**

Bits	Name	Description
15:0	PERIPH_PID	Bits 15:0 of the PID #0

**0x00786054 OEM\_CONFIG5****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the OEM configuration(191:160) bits in the QFPROM. This specific register contains PERIPH\_VID and ANTI\_ROLLBACK\_FEATURE\_EN information.

**OEM\_CONFIG5**

Bits	Name	Description
31:8	Reserved	Reserved
7:0	ANTI_ROLLBACK_FEATUR E_EN	Bit 0 - BOOT_ANTI_ROLLBACK_EN Bit 1 - TZAPPS_ANTI_ROLLBACK_EN Bit 2 - PILSUBSYS_ANTI_ROLLBACK_EN Bit 3 - MSA_ANTI_ROLLBACK_EN Bit 4 - Reserved Bit 5 - Reserved Bit 6 - Reserved Bit 7 - Reserved

**0x00786070 BOOT\_CONFIG****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for reading the proper PBL-related values. They should be used by boot code to determine the proper boot flow. The boot code should not read these directly from fuses. This register takes into account the fuse values, re-enable settings and the GPIO input during captured during reset.

**BOOT\_CONFIG**

Bits	Name	Description
10	FORCE_MSA_AUTH_EN	Indicates if MSS/MBA segments are forced for code authentication or not. 0x1: FORCE_CODE_AUTHENTICATION_FOR_SECURE_BOOT (Force code authentication for Secure Boot) 0x0: DOES_NOT_FORCE_CODE_AUTHENTICATION_FOR_SECURE_BOOT (Does not force code authentication for Secure Boot)
9:8	APPS_PBL_BOOT_SPEED	These fuses will be used to configure the frequency of the KPPPLL0(KPL2PLL) respectively. The exact frequency plan varies by target. 0x0: XO (XO (XO)) 0x1: ENUM_384_MHZ (384 MHz (384 MHz)) 0x2: ENUM_614_4_MHZ (614.4 MHz (576 MHz)) 0x3: ENUM_998_4_MHZ (998.4 MHz (768 MHz))
7	MODEM_BOOT_FROM_ROM	When set (enabled), this bit indicates that Modem is forced to boot from ROM. If not set, booting from Modem Code RAM is allowed. 0x0: BOOT_FROM_MODEM_CODE_RAM_ALLOWED (Boot from Modem Code RAM allowed) 0x1: FORCE_BOOT_FROM_ROM (Force Boot from ROM)
6	APPS_BOOT_FROM_ROM	
5:1	FAST_BOOT	Indicates which boot interface, the chip should boot from. The supported boot interface, as well as the behavior of default boot option (0x0), varies by target. The mapping is as follows: 0x0: DEFAULT (Default (UFS 1-LANE HS G1->SD->USB->eDL) (eDL Path: USB Only)) 0x1: SD_UFS_EDL (SD->UFS->eDL (eDL Path: USB Only)) 0x2: SD_EDL (SD->eDL (eDL Path: USB Only)) 0x3: USB_EDL (USB->eDL (eDL Path: SD then USB))

**BOOT\_CONFIG (cont.)**

Bits	Name	Description
0	WDOG_EN	This bit indicates if the values of the WDOG fuse and gpio allow the watch dog timer to function. WDOG is only disabled if its GPIO is high and its fuse is unblown. 0x1: ENABLE_WDOG (Enable WDOG) WDOG Enabled Regardless of the GPIO value) 0x0: GPIO_CONTROLLED (GPIO Controlled If GPIO = 0 Enable WDOG If GPIO = 1 Disable WDOG)

**0x00786078+ SECURE\_BOOTn, n=[1..14]****0x4\*n****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register indicates the authentication enablement for code segment 'n'. Software can reference a specific code segment secure boot bits to determine if the authentication is enabled for that code segment. This register takes into account the fuse values, re-enable settings and the GPIO input during captured during reset. The secure boot fuses allow for authentication of 28 separate code segments. This is useful in the case that secure code is coming from multiple sources.

**SECURE\_BOOTn**

Bits	Name	Description
8	FUSE_SRC	Indicates where SECURE_BOOTn setting for code segment is being pulled from. 0x0: QUALCOMM (Qualcomm) 0x1: OEM (OEM)
6	USE_SERIAL_NUM	This is for secure boot authentication. 0x0: USE_OEM_ID (USE_OEM_ID) 0x1: USE_SERIAL_NUM (USE_SERIAL_NUM)
5	AUTH_EN	Enables authentication for code referring to this secure boot configuration.
4		
3:0		

**0x007860C0 OVERRIDE\_0****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

This is a one-time-writable register used to override the fuse values when necessary. All writes following the first write access are ignored.

**OVERRIDE\_0**

Bits	Name	Description
31:3	RESERVED	Reserved.
2	TX_DISABLE	Used to disable the modem transmit block 0x0: ENABLE_TX (Enable Tx) 0x1: DISABLE_TX (Disable Tx)
1:0	RESERVED	Reserved.

**0x007860C4 OVERRIDE\_1****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

This is a one-time-writable register used to override the fuse values when necessary. All writes following the first write access are ignored.

**OVERRIDE\_1**

Bits	Name	Description
31:7	Reserved	Reserved
6	Reserved	Reserved
5	OVRID_DAP_DEVICEEN_DISABLE	Used to override the OEM DAP_DEVICEEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
4		
3	Reserved	Reserved
2	Reserved	Reserved

**OVERRIDE\_1 (cont.)**

Bits	Name	Description
1	Reserved	Reserved
0	OVRID_APPS_APB_DFD_DISABLE	Used to override the OEM APPS_APB_DFD_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)

**0x007860C8 OVERRIDE\_2****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

This is a one-time-writable register used to override the fuse values when necessary. OVERRIDE\_2 is used to override the Apps nonsecure disable fuses. All writes following the first write access are ignored.

**OVERRIDE\_2**

Bits	Name	Description
31:17	RESERVED	Reserved
16	OVRID_LPASS_SENSOR_NIDEN_DISABLE	Used to override the OEM LPASS_SENSOR_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
14		
13	OVRID_WCSS_NIDEN_DISABLE	Used to override the OEM WCSS_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
12	OVRID_LPASS_NIDEN_DISABLE	Used to override the OEM LPASS_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)

**OVERRIDE\_2 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
11	OVRID_DAP_NIDEN_DISABLE	Used to override the OEM DAP_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
10	OVRID_APPS_NIDEN_DISABLE	Used to override the OEM APPS_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
9	RESERVED	Reserved
7	RESERVED	Reserved
6	RESERVED	Reserved
4	RESERVED	Reserved
3	RESERVED	Reserved
2	RESERVED	Reserved
1	RESERVED	Reserved
0	RESERVED	Reserved

**0x007860CC OVERRIDE\_3****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

This is a one-time-writable register used to override the fuse values when necessary.

OVERRIDE\_3 is used to override the Apps secure disable fuses. All writes following the first write access are ignored.

 **OVERRIDE\_3**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
31:11	Reserved	Reserved.

**OVERRIDE\_3 (cont.)**

Bits	Name	Description
9	OVRID_LPASS_SPIDEN_DISABLE	Used to override the OEM LPASS_SPIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
7		
6	OVRID_DAP_SPIDEN_DISABLE	Used to override the OEM DAP_SPIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
5	OVRID_APPS_SPIDEN_DISABLE	Used to override the OEM APPS_SPIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
4	OVRID_LPASS_SPNIDEN_DISABLE	Used to override the OEM LPASS_SPNIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
2		
1	OVRID_DAP_SPNIDEN_DISABLE	Used to override the OEM DAP_SPNIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)
0	OVRID_APPS_SPNIDEN_DISABLE	Used to override the OEM APPS_SPNIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)

**0x007860D0 OVERRIDE\_4**

**Type:** RW  
**Clock:** System FPB clock  
**Reset State:** 0x00000000

This is a one-time-writable register used to override the fuse values when necessary. OVERRIDE\_4 is used to override the MSS disable fuses. All writes following the first write access are ignored.

#### OVERS<sub>IDE</sub>\_4

Bits	Name	Description
31:2	RESERVED	Reserved
1	OVRID_MSS_NIDEN_DISABLE	Used to override the OEM MSS_NIDEN_DISABLE fuse when not 0. 0x0: FUSE_VALUE (Fuse value) 0x1: QC_FUSE_VALUE (Fuse value)

#### 0x00786130 JTAG\_ID

**Type:** R  
**Clock:** System FPB clock  
**Reset State:** Undefined

This register holds the JTAG ID of the device. The upper 20 bits of this register are controlled by the JTAG ID fuses. The lower 12 bits are a constant value of 0xE1 as this is Qualcomm® identifier.

#### JTAG\_ID

Bits	Name	Description
31:0	JTAG_ID	The chip JTAG identifier. Bits 31-28: Die Revision/Version Bits 27-12: Product/Device ID Bits 11-0: Manufacturer ID (Fixed for Qualcomm @ 0xE1)

#### 0x00786134 SERIAL\_NUM

**Type:** R  
**Clock:** System FPB clock  
**Reset State:** Undefined

This register holds the 32-bit Serial Number of the device.

**SERIAL\_NUM**

Bits	Name	Description
31:0	SERIAL_NUM	32-bit Serial Number value.

**0x00786138 OEM\_ID****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register holds the OEM ID and OEM Product ID values as 16-bit values to make them easier to use by software.

**OEM\_ID**

Bits	Name	Description
31:16	OEM_ID	The OEM identifier
15:0	OEM_PRODUCT_ID	The OEM product identifier.

**0x00786150+ OEM\_IMAGE\_ENCR\_KEYn, n=[0..3]****0x4\*n****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the OEM Image Encryption Key bits in the QFPROM.

**OEM\_IMAGE\_ENCR\_KEYn**

Bits	Name	Description
31:0	KEY_DATA0	32 bits of the key data in this row.

**0x00786160 IMAGE\_ENCR\_KEY1\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the Image Encryption Key 1 bits in the QFPROM.

**IMAGE\_ENCR\_KEY1\_0**

Bits	Name	Description
31:0	KEY_DATA0	32 bits of the key data in this row.

**0x00786164 IMAGE\_ENCR\_KEY1\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the Image Encryption Key 1 bits in the QFPROM.

**IMAGE\_ENCR\_KEY1\_1**

Bits	Name	Description
31:0	KEY_DATA0	32 bits of the key data in this row.

**0x00786168 IMAGE\_ENCR\_KEY1\_2****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the Image Encryption Key 1 bits in the QFPROM.

**IMAGE\_ENCR\_KEY1\_2**

Bits	Name	Description
31:0	KEY_DATA0	32 bits of the key data in this row.

**0x0078616C IMAGE\_ENCR\_KEY1\_3****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the Image Encryption Key 1 bits in the QFPROM.

**IMAGE\_ENCR\_KEY1\_3**

Bits	Name	Description
31:0	KEY_DATA0	32 bits of the key data in this row.

**0x00786190 PK\_HASH0\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_0**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x00786194 PK\_HASH0\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_1**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x00786198 PK\_HASH0\_2****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_2**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x0078619C PK\_HASH0\_3****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_3**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861A0 PK\_HASH0\_4****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_4**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861A4 PK\_HASH0\_5****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_5**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861A8 PK\_HASH0\_6****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_6**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861AC PK\_HASH0\_7****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH0 bits in the QFPROM.

**PK\_HASH0\_7**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861B0 PK\_HASH1\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_0**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861B4 PK\_HASH1\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_1**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861B8 PK\_HASH1\_2****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_2**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861BC PK\_HASH1\_3****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_3**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861C0 PK\_HASH1\_4****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_4**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861C4 PK\_HASH1\_5****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_5**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861C8 PK\_HASH1\_6****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_6**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861CC PK\_HASH1\_7****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the PK\_HASH1 bits in the QFPROM.

**PK\_HASH1\_7**

Bits	Name	Description
31:0	HASH_DATA0	32 bits of the hash data in this row.

**0x007861E0 SW\_ROT\_STICKY\_BIT****Type:** RW**Clock:** System FPB clock**Reset State:** 0x00000000

These register controls write permissions to the PK\_HASH1 QFPROM region.

**SW\_ROT\_STICKY\_BIT**

Bits	Name	Description
0	SW_ROT_STICKY_BIT_0	Prevents write access to the PK_HASH1 region. This bit is a sticky bit, meaning when it has a 1 written to it, the 1 cannot be cleared back to 0 until a fuse sense reset is issued. The PBL writes a 1 to this register to block write access to the PK_HASH1 region of the QFPROM. 0x0: ALLOW_WRITE (Allow write) 0x1: DISABLE_WRITE (Disable write)

**0x007861E4 SW\_ROT\_CONFIG****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register allows for easy reading of the current software root of trust fuse bits in the QFPROM.

**SW\_ROT\_CONFIG**

Bits	Name	Description
1	CURRENT_SW_ROT_MOD_EM	Indicates whether PK_HASH0 or PK_HASH1 is used when verifying software code images. 0x0: PK_HASH0 (Use PK_HASH0) 0x1: PK_HASH1 (Use PK_HASH1)
0	CURRENT_SW_ROT_APPS	Indicates whether PK_HASH0 or PK_HASH1 is used when verifying software code images. 0x0: PK_HASH0 (Use PK_HASH0) 0x1: PK_HASH1 (Use PK_HASH1)

**0x00786200 ANTI\_ROLLBACK\_1\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the LSB of row 0 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_1\_0**

Bits	Name	Description
31:0	XBL0	This field contains the least significant bits of the lowest version of the Apps boot loader software that can run.

**0x00786204 ANTI\_ROLLBACK\_1\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the MSB of row 0 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_1\_1**

Bits	Name	Description
31:0	XBL1	This field contains the most significant bits of the lowest version of the Apps boot loader software that can run.

**0x00786208 ANTI\_ROLLBACK\_2\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the LSB of row 1 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_2\_0**

Bits	Name	Description
31:0	PIL_SUBSYSTEM_31_0	This field determines the least significant 32 bits for the lowest version of the subsystem images (apart from Modem Subsystem) loaded by Boot.

**0x0078620C ANTI\_ROLLBACK\_2\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the MSB of row 1 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_2\_1**

Bits	Name	Description
31:25	XBL_SEC	This field determines the lowest version of the XBL_SEC software version that can run.
24:17	RPM	This field determines the lowest version of the RPM software version that can run.
16:0	TZ	This field determines the lowest version of the TrustZone software that can run.

**0x00786210 ANTI\_ROLLBACK\_3\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the LSB of row 2 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_3\_0**

Bits	Name	Description
31:30	Reserved	Reserved
29:25	TQS_HASH_ACTIVE	TQS bits for image's Hash calculation on boot time.
24	RPMB_KEY_PROVISIONED	Indicates that the RPMB Key has been already provisioned. 0x0: RPMB_KEY_NOT_PROVISIONED (RPMB Key not provisioned) 0x1: RPMB_KEY_PROVISIONED (RPMB Key provisioned)
23:8	PIL_SUBSYSTEM_47_32	This field determines the most significant 16 bits for the lowest version of the subsystem images (apart from Modem Subsystem) loaded by Boot or TZ.
7:0	SAFESWITCH	Blown in the field.

**0x00786214 ANTI\_ROLLBACK\_3\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the MSB of row 2 of the AP Anti-Rollback region of the QFPROM. It is used by the AP to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The AP should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_3\_1**

Bits	Name	Description
31:28	RESERVED	Reserved
27:17	DEVICE_CFG	This field determines the lowest version of the device config software version that can run.
16:12	RESERVED	Reserved

**ANTI\_ROLLBACK\_3\_1 (cont.)**

Bits	Name	Description
11:0	HYPERVERISOR	This field determines the lowest version of the Hypervisor software version that can run.

**0x00786218 ANTI\_ROLLBACK\_4\_0****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the LSB of the MSA Anti-Rollback region of the QFPROM. It is used by the MSA to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The MSA should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_4\_0**

Bits	Name	Description
31:16	MSS	This field determines the lowest version of the MSS software that can run.
15:0	MBA	This field determines the lowest version of the MBA software that can run.

**0x0078621C ANTI\_ROLLBACK\_4\_1****Type:** R**Clock:** System FPB clock**Reset State:** Undefined

This register maps to the MSB of the MSA Anti-Rollback region of the QFPROM. It is used by the MSA to store the current version of the software that is expected to be running on the device. This prevents end users from rolling back software updates. The MSA should blow one fuse each time it makes a major software update. This fuse region uses a shadow double redundancy scheme.

**ANTI\_ROLLBACK\_4\_1**

Bits	Name	Description
31	SIMLOCK	To determine if the device has been unlocked by an attacker from operator lock perspective.
30:0	RESERVED	Reserved.

# 3 Top level mode multiplexing registers

## 3.1 Top-level multiplexing

Table 3-1 GPIO functions

GPIO #	Function	Output enable (gnd_tie = input) vdd_tie = output)	GPIO_CFG[5:2] (FUNC_SEL)	GPIO_CFG[1:0] (GPIO_PULL)
GPIO[0]	GPIO_IN_OUT(0)	GPIO_CFG(0)	0	(DEFAULT 01 = PD)
	BLSP_SPI_MOSI(1)	BLSP_SPI_MOSI_EN(1)	1	
	BLSP_UART_TX(1)	VDD_TIE	2	
	GPIO_IN_OUT(1)	GPIO_CFG(1)	0	
GPIO[1]	BLSP_SPI_MISO(1)	BLSP_SPI_MISO_EN(1)	1	
	BLSP_UART_RX(1)	GND_TIE	2	
	GPIO_IN_OUT(2)	GPIO_CFG(2)	0	
GPIO[2]	BLSP_SPI_CS_N(1)	BLSP_SPI_CS_N_EN(1)	1	(DEFAULT 01 = PD)
	BLSP_UART_CTS_N(1)	GND_TIE	2	
	BLSP_I2C_SDA(1)	BLSP_I2C_SDA_EN(1)	3	
	GPIO_IN_OUT(3)	GPIO_CFG(3)	0	
GPIO[3]	BLSP_SPI_CLK(1)	BLSP_SPI_CLK_EN(1)	1	
	BLSP_UART_RFR_N(1)	VDD_TIE	2	
	BLSP_I2C_SCL(1)	BLSP_I2C_SCL_EN(1)	3	
	GPIO_IN_OUT(4)	GPIO_CFG(4)	0	
	BLSP_SPI_MOSI(2)	BLSP_SPI_MOSI_EN(2)	1	
GPIO[4]	BLSP_UART_TX(2)	VDD_TIE	3	(DEFAULT 01 = PD)
	GPIO_IN_OUT(5)	GPIO_CFG(5)	0	

**Table 3-1 GPIO functions**

	BLSP_SPI_MISO(2)	BLSP_SPI_MISO_EN(2)	1	
	BLSP_UART_RX(2)	GND_TIE	3	
GPIO[6]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(6)	GPIO_CFG(6)	0	
	BLSP_SPI_CS_N(2)	BLSP_SPI_CS_N_EN(2)	1	
	BLSP_I2C_SDA(2)	BLSP_I2C_SDA_EN(2)	2	
	BLSP_UART_CTS_N(2)	GND_TIE	3	
GPIO[7]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(7)	GPIO_CFG(7)	0	
	BLSP_SPI_CLK(2)	BLSP_SPI_CLK_EN(2)	1	
	BLSP_I2C_SCL(2)	BLSP_I2C_SCL_EN(2)	2	
	BLSP_UART_RFR_N(2)	VDD_TIE	3	
GPIO[8]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(8)	GPIO_CFG(8)	0	
	BLSP_SPI_MOSI(3)	BLSP_SPI_MOSI_EN(3)	1	
GPIO[9]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(9)	GPIO_CFG(9)	0	
	BLSP_SPI_MISO(3)	BLSP_SPI_MISO_EN(3)	1	
GPIO[10]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(10)	GPIO_CFG(10)	0	
	BLSP_SPI_CS_N(3)	BLSP_SPI_CS_N_EN(3)	1	
	BLSP_I2C_SDA(3)	BLSP_I2C_SDA_EN(3)	2	
GPIO[11]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(11)	GPIO_CFG(11)	0	
	BLSP_SPI_CLK(3)	BLSP_SPI_CLK_EN(3)	1	
	BLSP_I2C_SCL(3)	BLSP_I2C_SCL_EN(3)	2	
GPIO[12]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(12)	GPIO_CFG(12)	0	
	BLSP_SPI_MOSI(4)	BLSP_SPI_MOSI_EN(4)	1	
	PRI_MI2S_SCK	PRI_MI2S_SCK_EN	2	
GPIO[13]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(13)	GPIO_CFG(13)	0	
	BLSP_SPI_MISO(4)	BLSP_SPI_MISO_EN(4)	1	
	DP_HOT_PLUG_DETECT	GND_TIE	2	
	PRI_MI2S_WS	PRI_MI2S_WS_EN	3	
	GP_PDM_MIRB(1)	GP_PDM_MIRB_EN(1)	4	
GPIO[14]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(14)	GPIO_CFG(14)	0	
	BLSP_SPI_CS_N(4)	BLSP_SPI_CS_N_EN(4)	1	

**Table 3-1 GPIO functions**

	BLSP_I2C_SDA(4)	BLSP_I2C_SDA_EN(4)	2	
	PRI_MI2S_DATA0	PRI_MI2S_DATA0_EN	3	
GPIO[15]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(15)	GPIO_CFG(15)	0	
	BLSP_SPI_CLK(4)	BLSP_SPI_CLK_EN(4)	1	
	BLSP_I2C_SCL(4)	BLSP_I2C_SCL_EN(4)	2	
	PRI_MI2S_DATA1	PRI_MI2S_DATA1_EN	3	
GPIO[16]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(16)	GPIO_CFG(16)	0	
	BLSP_UART_TX(5)	VDD_TIE	1	
	BLSP_SPI_MOSI(5)	BLSP_SPI_MOSI_EN(5)	2	
GPIO[17]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(17)	GPIO_CFG(17)	0	
	BLSP_UART_RX(5)	GND_TIE	1	
	BLSP_SPI_MISO(5)	BLSP_SPI_MISO_EN(5)	2	
GPIO[18]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(18)	GPIO_CFG(18)	0	
	BLSP_UART_CTS_N(5)	GND_TIE	1	
	BLSP_SPI_CS_N(5)	BLSP_SPI_CS_N_EN(5)	2	
	BLSP_I2C_SDA(5)	BLSP_I2C_SDA_EN(5)	3	
GPIO[19]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(19)	GPIO_CFG(19)	0	
	BLSP_UART_RFR_N(5)	VDD_TIE	1	
	BLSP_SPI_CLK(5)	BLSP_SPI_CLK_EN(5)	2	
	BLSP_I2C_SCL(5)	BLSP_I2C_SCL_EN(5)	3	
GPIO[20]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(20)	GPIO_CFG(20)	0	
GPIO[21]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(21)	GPIO_CFG(21)	0	
GPIO[22]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(22)	GPIO_CFG(22)	0	
	BLSP_SPI_CS_N(6)	BLSP_SPI_CS_N_EN(6)	1	
	BLSP_I2C_SDA(6)	BLSP_I2C_SDA_EN(6)	3	
GPIO[23]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(23)	GPIO_CFG(23)	0	
	BLSP_SPI_CLK(6)	BLSP_SPI_CLK_EN(6)	1	
	BLSP_I2C_SCL(6)	BLSP_I2C_SCL_EN(6)	2	
GPIO[24]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(24)	GPIO_CFG(24)	0	

**Table 3-1 GPIO functions**

	BLSP_SPI_MOSI(7)	BLSP_SPI_MOSI_EN(7)	1	
	BLSP_UART_TX6_MIRA	VDD_TIE	2	
	SEC_MI2S_SCK	SEC_MI2S_SCK_EN	3	
	SNDWIRE_CLK	SNDWIRE_CLK_EN	4	
	GP0_CLK_MIRB	VDD_TIE	5	
GPIO[25]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(25)	GPIO_CFG(25)	0	
	BLSP_SPI_MISO(7)	BLSP_SPI_MISO_EN(7)	1	
	BLSP_UART_RX6_MIRA	GND_TIE	2	
	SEC_MI2S_WS	SEC_MI2S_WS_EN	3	
	SNDWIRE_DATA	SNDWIRE_DATA_EN	4	
	GP1_CLK_MIRB	VDD_TIE	5	
GPIO[26]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(26)	GPIO_CFG(26)	0	
	BLSP_SPI_CS_N(7)	BLSP_SPI_CS_N_EN(7)	1	
	BLSP_UART_CTS_N6_MI RA	GND_TIE	2	
	BLSP_I2C_SDA(7)	BLSP_I2C_SDA_EN(7)	3	
	SEC_MI2S_DATA0	SEC_MI2S_DATA0_EN	4	
GPIO[27]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(27)	GPIO_CFG(27)	0	
	BLSP_SPI_CLK(7)	BLSP_SPI_CLK_EN(7)	1	
	BLSP_UART_RFR_N6_MI RA	VDD_TIE	2	
	BLSP_I2C_SCL(7)	BLSP_I2C_SCL_EN(7)	3	
	SEC_MI2S_DATA1	SEC_MI2S_DATA1_EN	5	
GPIO[28]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(28)	GPIO_CFG(28)	0	
	BLSP_SPI_MOSI8_MIRA	BLSP_SPI_MOSI8_MIRA_EN	1	
	BLSP_UART_TX6_MIRB	VDD_TIE	2	
GPIO[29]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(29)	GPIO_CFG(29)	0	
	BLSP_SPI_MISO8_MIRA	BLSP_SPI_MISO8_MIRA_EN	1	
	BLSP_UART_RX6_MIRB	GND_TIE	2	
	GP_MN	GP_MN_EN	3	
GPIO[30]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(30)	GPIO_CFG(30)	0	
	BLSP_SPI_CS_N8_MIRA	BLSP_SPI_CS_N8_MIRA_EN	1	

**Table 3-1 GPIO functions**

	BLSP_UART_CTS_N6_MI RB	GND_TIE	2	
	BLSP_I2C_SDA8_MIRA N	BLSP_I2C_SDA8_MIRA_E N	3	
	BLSP_SPI_CS1_N(3)	BLSP_SPI_CS1_N_EN(3)	4	
GPIO[31]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(31)	GPIO_CFG(31)	0	
	BLSP_SPI_CLK8_MIRA	BLSP_SPI_CLK8_MIRA_E N	1	
	BLSP_UART_RFR_N6_MI RB	VDD_TIE	2	
	BLSP_I2C_SCL8_MIRA	BLSP_I2C_SCL8_MIRA_E N	3	
GPIO[32]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(32)	GPIO_CFG(32)	0	
	CAM_MCLK0	VDD_TIE	1	
GPIO[33]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(33)	GPIO_CFG(33)	0	
	CAM_MCLK1	VDD_TIE	1	
GPIO[34]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(34)	GPIO_CFG(34)	0	
	CAM_MCLK2	VDD_TIE	1	
GPIO[35]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(35)	GPIO_CFG(35)	0	
	CAM_MCLK3	VDD_TIE	1	
GPIO[36]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(36)	GPIO_CFG(36)	0	
	CCI_I2C_SDA0	CCI_I2C_SDA0_EN	1	
GPIO[37]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(37)	GPIO_CFG(37)	0	
	CCI_I2C_SCL0	CCI_I2C_SCL0_EN	1	
GPIO[38]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(38)	GPIO_CFG(38)	0	
	CCI_I2C_SDA1	CCI_I2C_SDA1_EN	1	
GPIO[39]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(39)	GPIO_CFG(39)	0	
	CCI_I2C_SCL1	CCI_I2C_SCL1_EN	1	
GPIO[40]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(40)	GPIO_CFG(40)	0	
	CCI_TIMER0	VDD_TIE	1	
	GP0_CLK_MIRA	VDD_TIE	2	

**Table 3-1 GPIO functions**

	BLSP_SPI_MOSI8_MIRB	BLSP_SPI_MOSI8_MIRB_EN	3	
GPIO[41]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(41)	GPIO_CFG(41)	0	
	CCI_TIMER1	VDD_TIE	1	
	GP1_CLK_MIRA	VDD_TIE	2	
	BLSP_SPI_MISO8_MIRB	BLSP_SPI_MISO8_MIRB_EN	3	
GPIO[42]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(42)	GPIO_CFG(42)	0	
	MDSS_VSYNC0_OUT	VDD_TIE	1	
	MDSS_VSYNC1_OUT	VDD_TIE	2	
	MDSS_VSYNC2_OUT	VDD_TIE	3	
	MDSS_VSYNC3_OUT	VDD_TIE	4	
GPIO[43]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(43)	GPIO_CFG(43)	0	
	CCI_TIMER3	VDD_TIE	1	
	CCI_ASYNC_IN1	GND_TIE	2	
GPIO[44]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(44)	GPIO_CFG(44)	0	
	CCI_TIMER4	VDD_TIE	1	
	CCI_ASYNC_IN2	GND_TIE	2	
	BLSP_SPI_CLK8_MIRB	BLSP_SPI_CLK8_MIRB_EN	3	
	BLSP_I2C_SCL8_MIRB	BLSP_I2C_SCL8_MIRB_EN	4	
GPIO[45]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(45)	GPIO_CFG(45)	0	
	CCI_ASYNC_IN0	GND_TIE	1	
GPIO[46]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(46)	GPIO_CFG(46)	0	
	BLSP_SPI_CS1_N(1)	BLSP_SPI_CS1_N_EN(1)	1	
GPIO[47]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(47)	GPIO_CFG(47)	0	
GPIO[48]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(48)	GPIO_CFG(48)	0	
GPIO[49]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(49)	GPIO_CFG(49)	0	
	BLSP_SPI_MOSI(6)	BLSP_SPI_MOSI_EN(6)	1	
GPIO[50]				(DEFAULT 01 = PD)

**Table 3-1 GPIO functions**

	GPIO_IN_OUT(50)	GPIO_CFG(50)	0	
GPIO[51]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(51)	GPIO_CFG(51)	0	
GPIO[52]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(52)	GPIO_CFG(52)	0	
	CCI_TIMER2	VDD_TIE	1	
	BLSP_SPI_CS_N8_MIRB	BLSP_SPI_CS_N8_MIRB_EN	2	
	BLSP_I2C_SDA8_MIRB	BLSP_I2C_SDA8_MIRB_E_N	3	
	BLSP_SPI_MISO(6)	BLSP_SPI_MISO_EN(6)	4	
GPIO[53]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(53)	GPIO_CFG(53)	0	
	BOOT_CONFIG(14)	GND_TIE	MSMC_GPIO_SENSE_DONE_N_DFT	
GPIO[54]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(54)	GPIO_CFG(54)	0	
	GP_PDM_MIRB(0)	GP_PDM_MIRB_EN(0)	1	
GPIO[55]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(55)	GPIO_CFG(55)	0	
GPIO[56]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(56)	GPIO_CFG(56)	0	
GPIO[57]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(57)	GPIO_CFG(57)	0	
	GCC_GP1_CLK_MIRA	VDD_TIE	1	
	FORCED_USB_BOOT	GND_TIE	MSMC_GPIO_SENSE_DONE_N_DFT	
GPIO[58]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(58)	GPIO_CFG(58)	0	
	USB_PHY_PS	GND_TIE	1	
	GCC_GP2_CLK_MIRA	VDD_TIE	2	
GPIO[59]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(59)	GPIO_CFG(59)	0	
	MDP_VSYNC_P	GND_TIE	1	
	GCC_GP3_CLK_MIRA	VDD_TIE	2	
GPIO[60]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(60)	GPIO_CFG(60)	0	
	BOOT_CONFIG(3)	GND_TIE	MSMC_GPIO_SENSE_DONE_N_DFT	
GPIO[61]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(61)	GPIO_CFG(61)	0	

**Table 3-1 GPIO functions**

	PRI_MI2S_MCLK	VDD_TIE	1	
GPIO[62]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(62)	GPIO_CFG(62)	0	
	SEC_MI2S_MCLK	VDD_TIE	1	
	AUDIO_REF_CLK	GND_TIE	2	
	MDP_VSYNC_S	GND_TIE	3	
GPIO[63]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(63)	GPIO_CFG(63)	0	
	GP_PDM_MIRB(2)	GP_PDM_MIRB_EN(2)	1	
	BOOT_CONFIG(12)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[64]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(64)	GPIO_CFG(64)	0	
	BLSP_SPI_CS1_N(8)	BLSP_SPI_CS1_N_EN(8)	1	
GPIO[65]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(65)	GPIO_CFG(65)	0	
	GRFC(7)	VDD_TIE	1	
	NAV_PPS_IN_MIRA	GND_TIE	2	
	NAV_PPS_OUT_MIRA	VDD_TIE	3	
	GPS_TX_AGGRESSOR_M IRA	GND_TIE	4	
	BLSP_SPI_CS2_N(3)	BLSP_SPI_CS2_N_EN(3)	5	
	BOOT_CONFIG(6)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[66]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(66)	GPIO_CFG(66)	0	
GPIO[67]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(67)	GPIO_CFG(67)	0	
GPIO[68]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(68)	GPIO_CFG(68)	0	
GPIO[69]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(69)	GPIO_CFG(69)	0	
GPIO[70]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(70)	GPIO_CFG(70)	0	
GPIO[71]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(71)	GPIO_CFG(71)	0	
GPIO[72]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(72)	GPIO_CFG(72)	0	
GPIO[73]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(73)	GPIO_CFG(73)	0	

**Table 3-1 GPIO functions**

GPIO[74]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(74)	GPIO_CFG(74)	0	
	MDP_VSYNC_E	GND_TIE	1	
GPIO[75]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(75)	GPIO_CFG(75)	0	
GPIO[76]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(76)	GPIO_CFG(76)	0	
	BLSP_SPI_CS2_N(8)	BLSP_SPI_CS2_N_EN(8)	1	
	GP_PDM_MIRA(1)	GP_PDM_MIRA_EN(1)	2	
GPIO[77]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(77)	GPIO_CFG(77)	0	
GPIO[78]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(78)	GPIO_CFG(78)	0	
	GCC_GP1_CLK_MIRB	VDD_TIE	1	
GPIO[79]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(79)	GPIO_CFG(79)	0	
	GP_PDM_MIRA(2)	GP_PDM_MIRA_EN(2)	1	
GPIO[80]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(80)	GPIO_CFG(80)	0	
	NAV_PPS_IN_MIRC	GND_TIE	1	
	NAV_PPS_OUT_MIRC	VDD_TIE	2	
	GPS_TX_AGGRESSOR_MIRC	GND_TIE	3	
	BOOT_CONFIG(9)	GND_TIE	MSMC_GPIO_SENSE_DONE_N_DFT	
GPIO[81]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(81)	GPIO_CFG(81)	0	
	MSS_LTE_COXM_RXD	VDD_TIE	1	
	GCC_GP2_CLK_MIRB	VDD_TIE	2	
GPIO[82]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(82)	GPIO_CFG(82)	0	
	MSS_LTE_COXM_RXD	GND_TIE	1	
	GCC_GP3_CLK_MIRB	VDD_TIE	2	
GPIO[83]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(83)	GPIO_CFG(83)	0	
	UIM2_DATA	UIM2_DATA_EN	1	
GPIO[84]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(84)	GPIO_CFG(84)	0	
	UIM2_CLK	VDD_TIE	1	
GPIO[85]				(DEFAULT 01 = PD)

**Table 3-1 GPIO functions**

	GPIO_IN_OUT(85)	GPIO_CFG(85)	0	
	UIM2_RESET	VDD_TIE	1	
GPIO[86]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(86)	GPIO_CFG(86)	0	
	UIM2_PRESENT	GND_TIE	1	
GPIO[87]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(87)	GPIO_CFG(87)	0	
	UIM1_DATA	UIM1_DATA_EN	1	
GPIO[88]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(88)	GPIO_CFG(88)	0	
	UIM1_CLK	VDD_TIE	1	
GPIO[89]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(89)	GPIO_CFG(89)	0	
	UIM1_RESET	VDD_TIE	1	
GPIO[90]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(90)	GPIO_CFG(90)	0	
	UIM1_PRESENT	GND_TIE	1	
GPIO[91]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(91)	GPIO_CFG(91)	0	
	UIM_BATT_ALARM	UIM_BATT_ALARM_EN	1	
GPIO[92]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(92)	GPIO_CFG(92)	0	
	GRFC(0)	VDD_TIE	1	
	GRFC(13)	VDD_TIE	2	
	PA_INDICATOR_1_OR_2	VDD_TIE	3	
	GRFC(14)	VDD_TIE	4	
	BOOT_CONFIG(10)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[93]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(93)	GPIO_CFG(93)	0	
	GRFC(1)	VDD_TIE	1	
	BOOT_CONFIG(11)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[94]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(94)	GPIO_CFG(94)	0	
	GRFC(2)	VDD_TIE	1	
	BOOT_CONFIG(7)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[95]				(DEFAULT 01 = PD)

**Table 3-1 GPIO functions**

	GPIO_IN_OUT(95)	GPIO_CFG(95)	0	
	GRFC(3)	VDD_TIE	1	
	GP_PDM_MIRA(0)	GP_PDM_MIRA_EN(0)	2	
	BOOT_CONFIG[8]	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[96]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(96)	GPIO_CFG(96)	0	
	GRFC(4)	VDD_TIE	1	
	BOOT_CONFIG(0)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[97]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(97)	GPIO_CFG(97)	0	
	GRFC(5)	VDD_TIE	1	
	LDO_EN	VDD_TIE	2	
	BOOT_CONFIG(1)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[98]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(98)	GPIO_CFG(98)	0	
	GRFC(6)	VDD_TIE	1	
	NAV_PPS_OUT_MIRB	VDD_TIE	2	
	NAV_PPS_IN_MIRB	GND_TIE	3	
	GPS_TX_AGGRESSOR_M IRB	GND_TIE	4	
	LDO_UPDATE	VDD_TIE	5	
	BOOT_CONFIG(2)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	
GPIO[99]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(99)	GPIO_CFG(99)	0	
	QLINK_REQUEST	GND_TIE	1	
GPIO[100]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(100)	GPIO_CFG(100)	0	
	QLINK_ENABLE	VDD_TIE	1	
GPIO[101]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(101)	GPIO_CFG(101)	0	
	RFFE1_DATA	RFFE1_DATA_EN	1	
GPIO[102]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(102)	GPIO_CFG(102)	0	
	RFFE1_CLK	RFFE1_CLK_EN	1	
GPIO[103]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(103)	GPIO_CFG(103)	0	

**Table 3-1 GPIO functions**

	RFFE2_DATA	RFFE2_DATA_EN	1	
	GRFC(9)	VDD_TIE	2	
GPIO[104]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(104)	GPIO_CFG(104)	0	
	RFFE2_CLK	RFFE2_CLK_EN	1	
	GRFC(8)	VDD_TIE	2	
GPIO[105]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(105)	GPIO_CFG(105)	0	
	RFFE3_DATA	RFFE3_DATA_EN	1	
	GRFC(11)	VDD_TIE	2	
GPIO[106]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(106)	GPIO_CFG(106)	0	
	RFFE3_CLK	RFFE3_CLK_EN	1	
	GRFC(10)	VDD_TIE	2	
GPIO[107]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(107)	GPIO_CFG(107)	0	
	RFFE4_DATA	RFFE4_DATA_EN	1	
GPIO[108]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(108)	GPIO_CFG(108)	0	
	RFFE4_CLK	RFFE4_CLK_EN	1	
	GRFC(12)	VDD_TIE	2	
GPIO[109]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(109)	GPIO_CFG(109)	0	
	RFFE5_DATA	RFFE5_DATA_EN	1	
GPIO[110]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(110)	GPIO_CFG(110)	0	
	RFFE5_CLK	RFFE5_CLK_EN	1	
GPIO[111]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(111)	GPIO_CFG(111)	0	
	RFFE6_DATA	RFFE6_DATA_EN	1	
	GRFC(19)	VDD_TIE	2	
GPIO[112]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(112)	GPIO_CFG(112)	0	
	RFFE6_CLK	RFFE6_CLK_EN	1	
	GRFC(18)	VDD_TIE	2	
GPIO[113]				(DEFAULT 01 = PD)
	GPIO_IN_OUT(113)	GPIO_CFG(113)	0	
	BOOT_CONFIG(13)	GND_TIE	MSMC_GPIO_SENSE _DONE_N_DFT	

## 3.2 Top level mode multiplexing register

### 0x03100000 TLMM\_GPIO\_CFG0

**Type:** RW

**Clock:** gcc\_tlmm\_ahb\_clk

**Reset State:** 0x00000001

**Reset Name:** gcc\_tlmm\_ahb\_ares

#### TLMM\_GPIO\_CFG0

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

### 0x03100004 TLMM\_GPIO\_IN\_OUT0

**Type:** RW

**Clock:** gcc\_tlmm\_ahb\_clk

**Reset State:** 0x00000000

**Reset Name:** gcc\_tlmm\_ahb\_ares

#### TLMM\_GPIO\_IN\_OUT0

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03100008 TLMM\_GPIO\_INTR\_CFG0****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG0**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310000C TLMM\_GPIO\_INTR\_STATUS0****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS0**

Bits	Name	Description
0	INTR_STATUS	

**0x03100010 TLMM\_GPIO\_ID\_STATUS0****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS0**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03101000 TLMM\_GPIO\_CFG1****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG1**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03101004 TLMM\_GPIO\_IN\_OUT1****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT1**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03101008 TLMM\_GPIO\_INTR\_CFG1****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG1**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310100C TLMM\_GPIO\_INTR\_STATUS1****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS1**

Bits	Name	Description
0	INTR_STATUS	

**0x03101010 TLMM\_GPIO\_ID\_STATUS1****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS1**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03102000 TLMM\_GPIO\_CFG2****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG2**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG2 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03102004 TLMM\_GPIO\_IN\_OUT2****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT2**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03102008 TLMM\_GPIO\_INTR\_CFG2****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG2**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG2 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310200C TLMM\_GPIO\_INTR\_STATUS2****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS2**

Bits	Name	Description
0	INTR_STATUS	

**0x03102010 TLMM\_GPIO\_ID\_STATUS2****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS2**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03103000 TLMM\_GPIO\_CFG3****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG3**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03103004 TLMM\_GPIO\_IN\_OUT3****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT3**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03103008 TLMM\_GPIO\_INTR\_CFG3****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG3**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310300C TLMM\_GPIO\_INTR\_STATUS3****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS3**

Bits	Name	Description
0	INTR_STATUS	

**0x03103010 TLMM\_GPIO\_ID\_STATUS3****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS3**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03105000 TLMM\_GPIO\_CFG5****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG5**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03105004 TLMM\_GPIO\_IN\_OUT5****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT5**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03105008 TLMM\_GPIO\_INTR\_CFG5****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG5**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310500C TLMM\_GPIO\_INTR\_STATUS5****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS5**

Bits	Name	Description
0	INTR_STATUS	

**0x03105010 TLMM\_GPIO\_ID\_STATUS5****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS5**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03106000 TLMM\_GPIO\_CFG6****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG6**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03106004 TLMM\_GPIO\_IN\_OUT6****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT6**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03106008 TLMM\_GPIO\_INTR\_CFG6****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG6**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310600C TLMM\_GPIO\_INTR\_STATUS6****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS6**

Bits	Name	Description
0	INTR_STATUS	

**0x03106010 TLMM\_GPIO\_ID\_STATUS6****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS6**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03107000 TLMM\_GPIO\_CFG7****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG7**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG7 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03107004 TLMM\_GPIO\_IN\_OUT7****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT7**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03107008 TLMM\_GPIO\_INTR\_CFG7****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG7**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG7 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0310700C TLMM\_GPIO\_INTR\_STATUS7****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS7**

Bits	Name	Description
0	INTR_STATUS	

**0x03107010 TLMM\_GPIO\_ID\_STATUS7****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS7**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03114000 TLMM\_GPIO\_CFG20****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG20**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03114004 TLMM\_GPIO\_IN\_OUT20****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT20**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03114008 TLMM\_GPIO\_INTR\_CFG20****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG20**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0311400C TLMM\_GPIO\_INTR\_STATUS20****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS20**

Bits	Name	Description
0	INTR_STATUS	

**0x03114010 TLMM\_GPIO\_ID\_STATUS20****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS20**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03115000 TLMM\_GPIO\_CFG21****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG21**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03115004 TLMM\_GPIO\_IN\_OUT21****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT21**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03115008 TLMM\_GPIO\_INTR\_CFG21****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG21**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0311500C TLMM\_GPIO\_INTR\_STATUS21****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS21**

Bits	Name	Description
0	INTR_STATUS	

**0x03115010 TLMM\_GPIO\_ID\_STATUS21****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS21**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03120000 TLMM\_GPIO\_CFG32****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG32**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03120004 TLMM\_GPIO\_IN\_OUT32****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT32**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03120008 TLMM\_GPIO\_INTR\_CFG32****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG32**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312000C TLMM\_GPIO\_INTR\_STATUS32****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS32**

Bits	Name	Description
0	INTR_STATUS	

**0x03120010 TLMM\_GPIO\_ID\_STATUS32****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS32**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03121000 TLMM\_GPIO\_CFG33****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG33**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG33 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03121004 TLMM\_GPIO\_IN\_OUT33****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT33**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03121008 TLMM\_GPIO\_INTR\_CFG33****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG33**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG33 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312100C TLMM\_GPIO\_INTR\_STATUS33****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS33**

Bits	Name	Description
0	INTR_STATUS	

**0x03121010 TLMM\_GPIO\_ID\_STATUS33****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS33**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03122000 TLMM\_GPIO\_CFG34****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG34**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03122004 TLMM\_GPIO\_IN\_OUT34****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT34**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03122008 TLMM\_GPIO\_INTR\_CFG34****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG34**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312200C TLMM\_GPIO\_INTR\_STATUS34****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS34**

Bits	Name	Description
0	INTR_STATUS	

**0x03122010 TLMM\_GPIO\_ID\_STATUS34****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS34**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03123000 TLMM\_GPIO\_CFG35****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG35**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03123004 TLMM\_GPIO\_IN\_OUT35****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT35**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03123008 TLMM\_GPIO\_INTR\_CFG35****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG35**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312300C TLMM\_GPIO\_INTR\_STATUS35****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS35**

Bits	Name	Description
0	INTR_STATUS	

**0x03123010 TLMM\_GPIO\_ID\_STATUS35****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS35**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03124000 TLMM\_GPIO\_CFG36****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG36**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03124004 TLMM\_GPIO\_IN\_OUT36****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT36**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03124008 TLMM\_GPIO\_INTR\_CFG36****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG36**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312400C TLMM\_GPIO\_INTR\_STATUS36****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS36**

Bits	Name	Description
0	INTR_STATUS	

**0x03124010 TLMM\_GPIO\_ID\_STATUS36****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS36**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03125000 TLMM\_GPIO\_CFG37****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG37**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG37 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03125004 TLMM\_GPIO\_IN\_OUT37****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT37**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03125008 TLMM\_GPIO\_INTR\_CFG37****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG37**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG37 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312500C TLMM\_GPIO\_INTR\_STATUS37****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS37**

Bits	Name	Description
0	INTR_STATUS	

**0x03125010 TLMM\_GPIO\_ID\_STATUS37****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS37**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03126000 TLMM\_GPIO\_CFG38****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG38**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03126004 TLMM\_GPIO\_IN\_OUT38****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT38**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03126008 TLMM\_GPIO\_INTR\_CFG38****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG38**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312600C TLMM\_GPIO\_INTR\_STATUS38****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS38**

Bits	Name	Description
0	INTR_STATUS	

**0x03126010 TLMM\_GPIO\_ID\_STATUS38****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS38**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03127000 TLMM\_GPIO\_CFG39****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG39**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03127004 TLMM\_GPIO\_IN\_OUT39****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT39**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03127008 TLMM\_GPIO\_INTR\_CFG39****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG39**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312700C TLMM\_GPIO\_INTR\_STATUS39****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS39**

Bits	Name	Description
0	INTR_STATUS	

**0x03127010 TLMM\_GPIO\_ID\_STATUS39****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS39**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03128000 TLMM\_GPIO\_CFG40****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG40**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03128004 TLMM\_GPIO\_IN\_OUT40****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT40**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03128008 TLMM\_GPIO\_INTR\_CFG40****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG40**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312800C TLMM\_GPIO\_INTR\_STATUS40****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS40**

Bits	Name	Description
0	INTR_STATUS	

**0x03128010 TLMM\_GPIO\_ID\_STATUS40****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS40**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03129000 TLMM\_GPIO\_CFG41****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG41**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG41 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03129004 TLMM\_GPIO\_IN\_OUT41****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT41**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03129008 TLMM\_GPIO\_INTR\_CFG41****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG41**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG41 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312900C TLMM\_GPIO\_INTR\_STATUS41****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS41**

Bits	Name	Description
0	INTR_STATUS	

**0x03129010 TLMM\_GPIO\_ID\_STATUS41****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS41**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312A000 TLMM\_GPIO\_CFG42****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG42**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312A004 TLMM\_GPIO\_IN\_OUT42****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT42**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312A008 TLMM\_GPIO\_INTR\_CFG42****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG42**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312A00C TLMM\_GPIO\_INTR\_STATUS42****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS42**

Bits	Name	Description
0	INTR_STATUS	

**0x0312A010 TLMM\_GPIO\_ID\_STATUS42****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS42**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312B000 TLMM\_GPIO\_CFG43****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG43**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312B004 TLMM\_GPIO\_IN\_OUT43****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT43**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312B008 TLMM\_GPIO\_INTR\_CFG43****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG43**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312B00C TLMM\_GPIO\_INTR\_STATUS43****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS43**

Bits	Name	Description
0	INTR_STATUS	

**0x0312B010 TLMM\_GPIO\_ID\_STATUS43****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS43**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312C000 TLMM\_GPIO\_CFG44****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG44**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312C004 TLMM\_GPIO\_IN\_OUT44****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT44**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312C008 TLMM\_GPIO\_INTR\_CFG44****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG44**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312C00C TLMM\_GPIO\_INTR\_STATUS44****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS44**

Bits	Name	Description
0	INTR_STATUS	

**0x0312C010 TLMM\_GPIO\_ID\_STATUS44****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS44**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312D000 TLMM\_GPIO\_CFG45****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG45**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG45 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312D004 TLMM\_GPIO\_IN\_OUT45****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT45**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312D008 TLMM\_GPIO\_INTR\_CFG45****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG45**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG45 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312D00C TLMM\_GPIO\_INTR\_STATUS45****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS45**

Bits	Name	Description
0	INTR_STATUS	

**0x0312D010 TLMM\_GPIO\_ID\_STATUS45****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS45**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312E000 TLMM\_GPIO\_CFG46****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG46**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312E004 TLMM\_GPIO\_IN\_OUT46****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT46**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312E008 TLMM\_GPIO\_INTR\_CFG46****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG46**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312E00C TLMM\_GPIO\_INTR\_STATUS46****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS46**

Bits	Name	Description
0	INTR_STATUS	

**0x0312E010 TLMM\_GPIO\_ID\_STATUS46****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS46**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0312F000 TLMM\_GPIO\_CFG47****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG47**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0312F004 TLMM\_GPIO\_IN\_OUT47****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT47**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0312F008 TLMM\_GPIO\_INTR\_CFG47****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG47**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0312F00C TLMM\_GPIO\_INTR\_STATUS47****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS47**

Bits	Name	Description
0	INTR_STATUS	

**0x0312F010 TLMM\_GPIO\_ID\_STATUS47****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS47**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03130000 TLMM\_GPIO\_CFG48****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG48**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03130004 TLMM\_GPIO\_IN\_OUT48****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT48**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03130008 TLMM\_GPIO\_INTR\_CFG48****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG48**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313000C TLMM\_GPIO\_INTR\_STATUS48****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS48**

Bits	Name	Description
0	INTR_STATUS	

**0x03130010 TLMM\_GPIO\_ID\_STATUS48****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS48**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03131000 TLMM\_GPIO\_CFG49****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG49**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG49 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03131004 TLMM\_GPIO\_IN\_OUT49****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT49**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03131008 TLMM\_GPIO\_INTR\_CFG49****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG49**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG49 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313100C TLMM\_GPIO\_INTR\_STATUS49****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS49**

Bits	Name	Description
0	INTR_STATUS	

**0x03131010 TLMM\_GPIO\_ID\_STATUS49****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS49**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03132000 TLMM\_GPIO\_CFG50****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG50**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03132004 TLMM\_GPIO\_IN\_OUT50****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT50**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03132008 TLMM\_GPIO\_INTR\_CFG50****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG50**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313200C TLMM\_GPIO\_INTR\_STATUS50****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS50**

Bits	Name	Description
0	INTR_STATUS	

**0x03132010 TLMM\_GPIO\_ID\_STATUS50****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS50**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03133000 TLMM\_GPIO\_CFG51****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG51**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03133004 TLMM\_GPIO\_IN\_OUT51****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT51**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03133008 TLMM\_GPIO\_INTR\_CFG51****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG51**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313300C TLMM\_GPIO\_INTR\_STATUS51****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS51**

Bits	Name	Description
0	INTR_STATUS	

**0x03133010 TLMM\_GPIO\_ID\_STATUS51****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS51**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03134000 TLMM\_GPIO\_CFG52****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG52**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03134004 TLMM\_GPIO\_IN\_OUT52****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT52**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03134008 TLMM\_GPIO\_INTR\_CFG52****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG52**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313400C TLMM\_GPIO\_INTR\_STATUS52****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS52**

Bits	Name	Description
0	INTR_STATUS	

**0x03134010 TLMM\_GPIO\_ID\_STATUS52****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS52**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03137000 TLMM\_GPIO\_CFG55****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG55**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG55 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03137004 TLMM\_GPIO\_IN\_OUT55****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT55**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03137008 TLMM\_GPIO\_INTR\_CFG55****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG55**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG55 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313700C TLMM\_GPIO\_INTR\_STATUS55****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS55**

Bits	Name	Description
0	INTR_STATUS	

**0x03137010 TLMM\_GPIO\_ID\_STATUS55****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS55**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03138000 TLMM\_GPIO\_CFG56****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG56**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03138004 TLMM\_GPIO\_IN\_OUT56****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT56**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03138008 TLMM\_GPIO\_INTR\_CFG56****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG56**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313800C TLMM\_GPIO\_INTR\_STATUS56****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS56**

Bits	Name	Description
0	INTR_STATUS	

**0x03138010 TLMM\_GPIO\_ID\_STATUS56****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS56**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03139000 TLMM\_GPIO\_CFG57****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG57**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03139004 TLMM\_GPIO\_IN\_OUT57****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT57**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03139008 TLMM\_GPIO\_INTR\_CFG57****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG57**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313900C TLMM\_GPIO\_INTR\_STATUS57****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS57**

Bits	Name	Description
0	INTR_STATUS	

**0x03139010 TLMM\_GPIO\_ID\_STATUS57****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS57**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0313A000 TLMM\_GPIO\_CFG58****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG58**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0313A004 TLMM\_GPIO\_IN\_OUT58****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT58**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0313A008 TLMM\_GPIO\_INTR\_CFG58****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG58**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0313A00C TLMM\_GPIO\_INTR\_STATUS58****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS58**

Bits	Name	Description
0	INTR_STATUS	

**0x0313A010 TLMM\_GPIO\_ID\_STATUS58****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS58**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03140000 TLMM\_GPIO\_CFG64****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG64**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG64 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03140004 TLMM\_GPIO\_IN\_OUT64****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT64**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03140008 TLMM\_GPIO\_INTR\_CFG64****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG64**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG64 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0314000C TLMM\_GPIO\_INTR\_STATUS64****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS64**

Bits	Name	Description
0	INTR_STATUS	

**0x03140010 TLMM\_GPIO\_ID\_STATUS64****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS64**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03141000 TLMM\_GPIO\_CFG65****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG65**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03141004 TLMM\_GPIO\_IN\_OUT65****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT65**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03141008 TLMM\_GPIO\_INTR\_CFG65****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG65**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0314100C TLMM\_GPIO\_INTR\_STATUS65****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS65**

Bits	Name	Description
0	INTR_STATUS	

**0x03141010 TLMM\_GPIO\_ID\_STATUS65****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS65**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0314F000 TLMM\_GPIO\_CFG79****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG79**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0314F004 TLMM\_GPIO\_IN\_OUT79****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT79**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0314F008 TLMM\_GPIO\_INTR\_CFG79****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG79**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0314F00C TLMM\_GPIO\_INTR\_STATUS79****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS79**

Bits	Name	Description
0	INTR_STATUS	

**0x0314F010 TLMM\_GPIO\_ID\_STATUS79****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS79**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03150000 TLMM\_GPIO\_CFG80****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG80**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03150004 TLMM\_GPIO\_IN\_OUT80****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT80**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03150008 TLMM\_GPIO\_INTR\_CFG80****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG80**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315000C TLMM\_GPIO\_INTR\_STATUS80****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS80**

Bits	Name	Description
0	INTR_STATUS	

**0x03150010 TLMM\_GPIO\_ID\_STATUS80****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS80**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03153000 TLMM\_GPIO\_CFG83****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG83**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG83 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03153004 TLMM\_GPIO\_IN\_OUT83****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT83**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03153008 TLMM\_GPIO\_INTR\_CFG83****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG83**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG83 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315300C TLMM\_GPIO\_INTR\_STATUS83****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS83**

Bits	Name	Description
0	INTR_STATUS	

**0x03153010 TLMM\_GPIO\_ID\_STATUS83****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS83**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03154000 TLMM\_GPIO\_CFG84****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG84**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03154004 TLMM\_GPIO\_IN\_OUT84****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT84**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03154008 TLMM\_GPIO\_INTR\_CFG84****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG84**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315400C TLMM\_GPIO\_INTR\_STATUS84****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS84**

Bits	Name	Description
0	INTR_STATUS	

**0x03154010 TLMM\_GPIO\_ID\_STATUS84****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS84**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03155000 TLMM\_GPIO\_CFG85****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG85**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03155004 TLMM\_GPIO\_IN\_OUT85****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT85**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03155008 TLMM\_GPIO\_INTR\_CFG85****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG85**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315500C TLMM\_GPIO\_INTR\_STATUS85****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS85**

Bits	Name	Description
0	INTR_STATUS	

**0x03155010 TLMM\_GPIO\_ID\_STATUS85****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS85**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03156000 TLMM\_GPIO\_CFG86****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG86**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03156004 TLMM\_GPIO\_IN\_OUT86****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT86**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03156008 TLMM\_GPIO\_INTR\_CFG86****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG86**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315600C TLMM\_GPIO\_INTR\_STATUS86****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS86**

Bits	Name	Description
0	INTR_STATUS	

**0x03156010 TLMM\_GPIO\_ID\_STATUS86****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS86**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03157000 TLMM\_GPIO\_CFG87****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG87**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG87 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03157004 TLMM\_GPIO\_IN\_OUT87****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT87**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03157008 TLMM\_GPIO\_INTR\_CFG87****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG87**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG87 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315700C TLMM\_GPIO\_INTR\_STATUS87****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS87**

Bits	Name	Description
0	INTR_STATUS	

**0x03157010 TLMM\_GPIO\_ID\_STATUS87****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS87**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03158000 TLMM\_GPIO\_CFG88****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG88**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03158004 TLMM\_GPIO\_IN\_OUT88****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT88**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03158008 TLMM\_GPIO\_INTR\_CFG88****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG88**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315800C TLMM\_GPIO\_INTR\_STATUS88****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS88**

Bits	Name	Description
0	INTR_STATUS	

**0x03158010 TLMM\_GPIO\_ID\_STATUS88****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS88**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03159000 TLMM\_GPIO\_CFG89****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG89**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03159004 TLMM\_GPIO\_IN\_OUT89****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT89**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03159008 TLMM\_GPIO\_INTR\_CFG89****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG89**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315900C TLMM\_GPIO\_INTR\_STATUS89****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS89**

Bits	Name	Description
0	INTR_STATUS	

**0x03159010 TLMM\_GPIO\_ID\_STATUS89****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS89**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315A000 TLMM\_GPIO\_CFG90****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG90**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315A004 TLMM\_GPIO\_IN\_OUT90****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT90**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315A008 TLMM\_GPIO\_INTR\_CFG90****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG90**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315A00C TLMM\_GPIO\_INTR\_STATUS90****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS90**

Bits	Name	Description
0	INTR_STATUS	

**0x0315A010 TLMM\_GPIO\_ID\_STATUS90****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS90**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315B000 TLMM\_GPIO\_CFG91****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG91**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG91 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315B004 TLMM\_GPIO\_IN\_OUT91****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT91**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315B008 TLMM\_GPIO\_INTR\_CFG91****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG91**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG91 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315B00C TLMM\_GPIO\_INTR\_STATUS91****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS91**

Bits	Name	Description
0	INTR_STATUS	

**0x0315B010 TLMM\_GPIO\_ID\_STATUS91****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS91**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315C000 TLMM\_GPIO\_CFG92****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG92**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315C004 TLMM\_GPIO\_IN\_OUT92****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT92**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315C008 TLMM\_GPIO\_INTR\_CFG92****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG92**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315C00C TLMM\_GPIO\_INTR\_STATUS92****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS92**

Bits	Name	Description
0	INTR_STATUS	

**0x0315C010 TLMM\_GPIO\_ID\_STATUS92****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS92**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315D000 TLMM\_GPIO\_CFG93****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG93**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315D004 TLMM\_GPIO\_IN\_OUT93****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT93**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315D008 TLMM\_GPIO\_INTR\_CFG93****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG93**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315D00C TLMM\_GPIO\_INTR\_STATUS93****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS93**

Bits	Name	Description
0	INTR_STATUS	

**0x0315D010 TLMM\_GPIO\_ID\_STATUS93****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS93**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315E000 TLMM\_GPIO\_CFG94****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG94**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315E004 TLMM\_GPIO\_IN\_OUT94****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT94**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315E008 TLMM\_GPIO\_INTR\_CFG94****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG94**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315E00C TLMM\_GPIO\_INTR\_STATUS94****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS94**

Bits	Name	Description
0	INTR_STATUS	

**0x0315E010 TLMM\_GPIO\_ID\_STATUS94****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS94**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0315F000 TLMM\_GPIO\_CFG95****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG95**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG95 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0315F004 TLMM\_GPIO\_IN\_OUT95****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT95**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0315F008 TLMM\_GPIO\_INTR\_CFG95****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG95**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG95 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0315F00C TLMM\_GPIO\_INTR\_STATUS95****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS95**

Bits	Name	Description
0	INTR_STATUS	

**0x0315F010 TLMM\_GPIO\_ID\_STATUS95****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS95**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03160000 TLMM\_GPIO\_CFG96****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG96**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03160004 TLMM\_GPIO\_IN\_OUT96****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT96**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03160008 TLMM\_GPIO\_INTR\_CFG96****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG96**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316000C TLMM\_GPIO\_INTR\_STATUS96****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS96**

Bits	Name	Description
0	INTR_STATUS	

**0x03160010 TLMM\_GPIO\_ID\_STATUS96****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS96**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03161000 TLMM\_GPIO\_CFG97****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG97**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03161004 TLMM\_GPIO\_IN\_OUT97****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT97**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03161008 TLMM\_GPIO\_INTR\_CFG97****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG97**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316100C TLMM\_GPIO\_INTR\_STATUS97****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS97**

Bits	Name	Description
0	INTR_STATUS	

**0x03161010 TLMM\_GPIO\_ID\_STATUS97****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS97**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03162000 TLMM\_GPIO\_CFG98****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG98**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03162004 TLMM\_GPIO\_IN\_OUT98****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT98**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03162008 TLMM\_GPIO\_INTR\_CFG98****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG98**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316200C TLMM\_GPIO\_INTR\_STATUS98****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS98**

Bits	Name	Description
0	INTR_STATUS	

**0x03162010 TLMM\_GPIO\_ID\_STATUS98****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS98**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03163000 TLMM\_GPIO\_CFG99****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG99**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG99 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03163004 TLMM\_GPIO\_IN\_OUT99****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT99**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03163008 TLMM\_GPIO\_INTR\_CFG99****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG99**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG99 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316300C TLMM\_GPIO\_INTR\_STATUS99****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS99**

Bits	Name	Description
0	INTR_STATUS	

**0x03163010 TLMM\_GPIO\_ID\_STATUS99****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS99**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03164000 TLMM\_GPIO\_CFG100****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG100**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03164004 TLMM\_GPIO\_IN\_OUT100****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT100**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03164008 TLMM\_GPIO\_INTR\_CFG100****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG100**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316400C TLMM\_GPIO\_INTR\_STATUS100****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS100**

Bits	Name	Description
0	INTR_STATUS	

**0x03164010 TLMM\_GPIO\_ID\_STATUS100****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS100**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03165000 TLMM\_GPIO\_CFG101****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG101**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03165004 TLMM\_GPIO\_IN\_OUT101****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT101**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03165008 TLMM\_GPIO\_INTR\_CFG101****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG101**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316500C TLMM\_GPIO\_INTR\_STATUS101****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS101**

Bits	Name	Description
0	INTR_STATUS	

**0x03165010 TLMM\_GPIO\_ID\_STATUS101****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS101**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03166000 TLMM\_GPIO\_CFG102****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG102**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03166004 TLMM\_GPIO\_IN\_OUT102****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT102**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03166008 TLMM\_GPIO\_INTR\_CFG102****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG102**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316600C TLMM\_GPIO\_INTR\_STATUS102****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS102**

Bits	Name	Description
0	INTR_STATUS	

**0x03166010 TLMM\_GPIO\_ID\_STATUS102****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS102**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03167000 TLMM\_GPIO\_CFG103****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG103**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG103 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03167004 TLMM\_GPIO\_IN\_OUT103****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT103**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03167008 TLMM\_GPIO\_INTR\_CFG103****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG103**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG103 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316700C TLMM\_GPIO\_INTR\_STATUS103****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS103**

Bits	Name	Description
0	INTR_STATUS	

**0x03167010 TLMM\_GPIO\_ID\_STATUS103****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS103**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03168000 TLMM\_GPIO\_CFG104****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG104**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03168004 TLMM\_GPIO\_IN\_OUT104****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT104**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03168008 TLMM\_GPIO\_INTR\_CFG104****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG104**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316800C TLMM\_GPIO\_INTR\_STATUS104****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS104**

Bits	Name	Description
0	INTR_STATUS	

**0x03168010 TLMM\_GPIO\_ID\_STATUS104****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS104**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03169000 TLMM\_GPIO\_CFG105****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG105**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03169004 TLMM\_GPIO\_IN\_OUT105****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT105**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03169008 TLMM\_GPIO\_INTR\_CFG105****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG105**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316900C TLMM\_GPIO\_INTR\_STATUS105****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS105**

Bits	Name	Description
0	INTR_STATUS	

**0x03169010 TLMM\_GPIO\_ID\_STATUS105****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS105**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316A000 TLMM\_GPIO\_CFG106****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG106**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316A004 TLMM\_GPIO\_IN\_OUT106****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT106**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316A008 TLMM\_GPIO\_INTR\_CFG106****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG106**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316A00C TLMM\_GPIO\_INTR\_STATUS106****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS106**

Bits	Name	Description
0	INTR_STATUS	

**0x0316A010 TLMM\_GPIO\_ID\_STATUS106****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS106**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316B000 TLMM\_GPIO\_CFG107****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG107**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG107 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316B004 TLMM\_GPIO\_IN\_OUT107****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT107**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316B008 TLMM\_GPIO\_INTR\_CFG107****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG107**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG107 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316B00C TLMM\_GPIO\_INTR\_STATUS107****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS107**

Bits	Name	Description
0	INTR_STATUS	

**0x0316B010 TLMM\_GPIO\_ID\_STATUS107****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS107**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316C000 TLMM\_GPIO\_CFG108****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG108**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316C004 TLMM\_GPIO\_IN\_OUT108****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT108**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316C008 TLMM\_GPIO\_INTR\_CFG108****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG108**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316C00C TLMM\_GPIO\_INTR\_STATUS108****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS108**

Bits	Name	Description
0	INTR_STATUS	

**0x0316C010 TLMM\_GPIO\_ID\_STATUS108****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS108**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316D000 TLMM\_GPIO\_CFG109****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG109**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316D004 TLMM\_GPIO\_IN\_OUT109****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT109**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316D008 TLMM\_GPIO\_INTR\_CFG109****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG109**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316D00C TLMM\_GPIO\_INTR\_STATUS109****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS109**

Bits	Name	Description
0	INTR_STATUS	

**0x0316D010 TLMM\_GPIO\_ID\_STATUS109****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS109**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316E000 TLMM\_GPIO\_CFG110****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG110**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316E004 TLMM\_GPIO\_IN\_OUT110****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT110**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316E008 TLMM\_GPIO\_INTR\_CFG110****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG110**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316E00C TLMM\_GPIO\_INTR\_STATUS110****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS110**

Bits	Name	Description
0	INTR_STATUS	

**0x0316E010 TLMM\_GPIO\_ID\_STATUS110****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS110**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0316F000 TLMM\_GPIO\_CFG111****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG111**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG111 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0316F004 TLMM\_GPIO\_IN\_OUT111****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT111**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0316F008 TLMM\_GPIO\_INTR\_CFG111****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG111**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG111 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0316F00C TLMM\_GPIO\_INTR\_STATUS111****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS111**

Bits	Name	Description
0	INTR_STATUS	

**0x0316F010 TLMM\_GPIO\_ID\_STATUS111****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS111**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03170000 TLMM\_GPIO\_CFG112****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG112**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03170004 TLMM\_GPIO\_IN\_OUT112****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT112**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03170008 TLMM\_GPIO\_INTR\_CFG112****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG112**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0317000C TLMM\_GPIO\_INTR\_STATUS112****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS112**

Bits	Name	Description
0	INTR_STATUS	

**0x03170010 TLMM\_GPIO\_ID\_STATUS112****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS112**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03171000 TLMM\_GPIO\_CFG113****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG113**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03171004 TLMM\_GPIO\_IN\_OUT113****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT113**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03171008 TLMM\_GPIO\_INTR\_CFG113****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG113**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0317100C TLMM\_GPIO\_INTR\_STATUS113****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS113**

Bits	Name	Description
0	INTR_STATUS	

**0x03171010 TLMM\_GPIO\_ID\_STATUS113****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS113**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03196000 TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_0****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_0**

Bits	Name	Description
31	GPIO_101	0x0: DISABLE 0x1: ENABLE
30	GPIO_98	0x0: DISABLE 0x1: ENABLE
29	GPIO_91	0x0: DISABLE 0x1: ENABLE
28	GPIO_90	0x0: DISABLE 0x1: ENABLE
27	GPIO_87	0x0: DISABLE 0x1: ENABLE
26	GPIO_86	0x0: DISABLE 0x1: ENABLE
25	GPIO_85	0x0: DISABLE 0x1: ENABLE
24	GPIO_84	0x0: DISABLE 0x1: ENABLE
23	GPIO_83	0x0: DISABLE 0x1: ENABLE
22	GPIO_65	0x0: DISABLE 0x1: ENABLE

**TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_0 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
21	GPIO_64	0x0: DISABLE 0x1: ENABLE
20	GPIO_58	0x0: DISABLE 0x1: ENABLE
19	GPIO_56	0x0: DISABLE 0x1: ENABLE
18	GPIO_55	0x0: DISABLE 0x1: ENABLE
17	GPIO_52	0x0: DISABLE 0x1: ENABLE
16	GPIO_51	0x0: DISABLE 0x1: ENABLE
15	GPIO_50	0x0: DISABLE 0x1: ENABLE
14	GPIO_49	0x0: DISABLE 0x1: ENABLE
13	GPIO_48	0x0: DISABLE 0x1: ENABLE
12	GPIO_47	0x0: DISABLE 0x1: ENABLE
11	GPIO_46	0x0: DISABLE 0x1: ENABLE
10	GPIO_45	0x0: DISABLE 0x1: ENABLE
9	GPIO_44	0x0: DISABLE 0x1: ENABLE
8	GPIO_43	0x0: DISABLE 0x1: ENABLE
7	GPIO_42	0x0: DISABLE 0x1: ENABLE
6	GPIO_41	0x0: DISABLE 0x1: ENABLE
5	GPIO_40	0x0: DISABLE 0x1: ENABLE
4	GPIO_21	0x0: DISABLE 0x1: ENABLE
3	GPIO_6	0x0: DISABLE 0x1: ENABLE

**TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_0 (cont.)**

Bits	Name	Description
2	GPIO_5	0x0: DISABLE 0x1: ENABLE
1	GPIO_2	0x0: DISABLE 0x1: ENABLE
0	GPIO_1	0x0: DISABLE 0x1: ENABLE

**0x03196004 TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_1****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_MPM\_WAKEUP\_INT\_EN\_1**

Bits	Name	Description
5	GPIO_113	0x0: DISABLE 0x1: ENABLE
4	GPIO_111	0x0: DISABLE 0x1: ENABLE
3	GPIO_109	0x0: DISABLE 0x1: ENABLE
2	GPIO_107	0x0: DISABLE 0x1: ENABLE
1	GPIO_105	0x0: DISABLE 0x1: ENABLE
0	GPIO_103	0x0: DISABLE 0x1: ENABLE

**0x03197000 TLMM\_SOUTH\_CLK\_GATE\_EN****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_SOUTH\_CLK\_GATE\_EN**

Bits	Name	Description
2	AHB_HCLK_EN	0x1: ENABLE 0x0: DISABLE
1	SUMMARY_INTR_EN	0x1: ENABLE 0x0: DISABLE
0	CRIF_READ_EN	0x1: ENABLE 0x0: DISABLE

**0x03197004 TLMM\_SOUTH\_IE\_CTRL\_DISABLE****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_IE\_CTRL\_DISABLE**

Bits	Name	Description
0	IE_CTRL_DISABLE	0x1: DISABLE 0x0: ENABLE

**0x03199000 TLMM\_RFFE\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_RFFE\_CTL**

Bits	Name	Description
11	RFFE6_DATA_SR_CTL_EN	
10	RFFE6_CLK_SR_CTL_EN	
9	RFFE5_DATA_SR_CTL_EN	
8	RFFE5_CLK_SR_CTL_EN	
7	RFFE4_DATA_SR_CTL_EN	
6	RFFE4_CLK_SR_CTL_EN	
5	RFFE3_DATA_SR_CTL_EN	
4	RFFE3_CLK_SR_CTL_EN	

**TLMM\_RFFE\_CTL (cont.)**

Bits	Name	Description
3	RFFE2_DATA_SR_CTL_EN	
2	RFFE2_CLK_SR_CTL_EN	
1	RFFE1_DATA_SR_CTL_EN	
0	RFFE1_CLK_SR_CTL_EN	

**0x0319A000 TLMM\_PMIC\_HDRV\_PULL\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00FF581B**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_PMIC\_HDRV\_PULL\_CTL**

Bits	Name	Description
23	PMIC_SPMI_HIHYS_EN_DATA	
22	PMIC_SPMI_HIHYS_EN_CLK	
21:20	PMIC_SPMI_DATA_SR_CTL	0x0: GPIO_MODE 0x1: RFFE_MODE 0x2: SLIMBUS_MODE 0x3: SPMI_MODE
19:18	PMIC_SPMI_CLK_SR_CTL	0x0: GPIO_MODE 0x1: RFFE_MODE 0x2: SLIMBUS_MODE 0x3: SPMI_MODE
15:14	PMIC_SPMI_DATA_PULL	
13:12	PMIC_SPMI_CLK_PULL	
11:9	PSHOLD_HDRV	
8:6	CXO_EN_HDRV	
5:3	PMIC_SPMI_DATA_HDRV	
2:0	PMIC_SPMI_CLK_HDRV	

**0x0319D000 TLMM\_RESIN\_N\_HIHYS\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_RESIN\_N\_HIHYS\_CTL**

Bits	Name	Description
0	RESIN_N_HIHYS_CTL	

**0x0319F000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_SENSORS, n=[0..1]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_SENSORS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x031A0000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP, n=[0..5]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x031A1000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_RPM, n=[0..0]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_RPM**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x031A2000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_HMSS, n=[0..7]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_HMSS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x031A3000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_GSS, n=[0..1]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_GSS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x031A4000+ TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_CDSP, n=[0..1]**  
**0x4\*n**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x00000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_SOUTH\_DIR\_CONN\_INTRn\_CFG\_CDSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x03510000 TLMM\_GPIO\_CFG16**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x00000001

**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG16**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03510004 TLMM\_GPIO\_IN\_OUT16****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT16**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03510008 TLMM\_GPIO\_INTR\_CFG16****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG16**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351000C TLMM\_GPIO\_INTR\_STATUS16****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS16**

Bits	Name	Description
0	INTR_STATUS	

**0x03510010 TLMM\_GPIO\_ID\_STATUS16****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS16**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03511000 TLMM\_GPIO\_CFG17****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG17**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG17 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03511004 TLMM\_GPIO\_IN\_OUT17****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT17**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03511008 TLMM\_GPIO\_INTR\_CFG17****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG17**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG17 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351100C TLMM\_GPIO\_INTR\_STATUS17****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS17**

Bits	Name	Description
0	INTR_STATUS	

**0x03511010 TLMM\_GPIO\_ID\_STATUS17****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS17**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03512000 TLMM\_GPIO\_CFG18****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG18**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03512004 TLMM\_GPIO\_IN\_OUT18****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT18**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03512008 TLMM\_GPIO\_INTR\_CFG18****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG18**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351200C TLMM\_GPIO\_INTR\_STATUS18****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS18**

Bits	Name	Description
0	INTR_STATUS	

**0x03512010 TLMM\_GPIO\_ID\_STATUS18****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS18**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03513000 TLMM\_GPIO\_CFG19****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG19**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03513004 TLMM\_GPIO\_IN\_OUT19****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT19**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03513008 TLMM\_GPIO\_INTR\_CFG19****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG19**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351300C TLMM\_GPIO\_INTR\_STATUS19****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS19**

Bits	Name	Description
0	INTR_STATUS	

**0x03513010 TLMM\_GPIO\_ID\_STATUS19****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS19**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03516000 TLMM\_GPIO\_CFG22****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG22**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03516004 TLMM\_GPIO\_IN\_OUT22****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT22**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03516008 TLMM\_GPIO\_INTR\_CFG22****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG22**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351600C TLMM\_GPIO\_INTR\_STATUS22****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS22**

Bits	Name	Description
0	INTR_STATUS	

**0x03516010 TLMM\_GPIO\_ID\_STATUS22****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS22**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03517000 TLMM\_GPIO\_CFG23****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG23**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG23 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03517004 TLMM\_GPIO\_IN\_OUT23****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT23**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03517008 TLMM\_GPIO\_INTR\_CFG23****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG23**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG23 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351700C TLMM\_GPIO\_INTR\_STATUS23****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS23**

Bits	Name	Description
0	INTR_STATUS	

**0x03517010 TLMM\_GPIO\_ID\_STATUS23****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS23**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0351C000 TLMM\_GPIO\_CFG28****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG28**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0351C004 TLMM\_GPIO\_IN\_OUT28****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT28**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0351C008 TLMM\_GPIO\_INTR\_CFG28****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG28**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351C00C TLMM\_GPIO\_INTR\_STATUS28****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS28**

Bits	Name	Description
0	INTR_STATUS	

**0x0351C010 TLMM\_GPIO\_ID\_STATUS28****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS28**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0351D000 TLMM\_GPIO\_CFG29****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG29**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0351D004 TLMM\_GPIO\_IN\_OUT29****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT29**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0351D008 TLMM\_GPIO\_INTR\_CFG29****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG29**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351D00C TLMM\_GPIO\_INTR\_STATUS29****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS29**

Bits	Name	Description
0	INTR_STATUS	

**0x0351D010 TLMM\_GPIO\_ID\_STATUS29****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS29**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0351E000 TLMM\_GPIO\_CFG30****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG30**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0351E004 TLMM\_GPIO\_IN\_OUT30****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT30**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0351E008 TLMM\_GPIO\_INTR\_CFG30****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG30**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351E00C TLMM\_GPIO\_INTR\_STATUS30****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS30**

Bits	Name	Description
0	INTR_STATUS	

**0x0351E010 TLMM\_GPIO\_ID\_STATUS30****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS30**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0351F000 TLMM\_GPIO\_CFG31****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG31**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG31 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0351F004 TLMM\_GPIO\_IN\_OUT31****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT31**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0351F008 TLMM\_GPIO\_INTR\_CFG31****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG31**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG31 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0351F00C TLMM\_GPIO\_INTR\_STATUS31****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS31**

Bits	Name	Description
0	INTR_STATUS	

**0x0351F010 TLMM\_GPIO\_ID\_STATUS31****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS31**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03551000 TLMM\_GPIO\_CFG81****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG81**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03551004 TLMM\_GPIO\_IN\_OUT81****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT81**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03551008 TLMM\_GPIO\_INTR\_CFG81****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG81**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0355100C TLMM\_GPIO\_INTR\_STATUS81****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS81**

Bits	Name	Description
0	INTR_STATUS	

**0x03551010 TLMM\_GPIO\_ID\_STATUS81****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS81**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03552000 TLMM\_GPIO\_CFG82****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG82**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03552004 TLMM\_GPIO\_IN\_OUT82****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT82**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03552008 TLMM\_GPIO\_INTR\_CFG82****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG82**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0355200C TLMM\_GPIO\_INTR\_STATUS82****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS82**

Bits	Name	Description
0	INTR_STATUS	

**0x03552010 TLMM\_GPIO\_ID\_STATUS82****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS82**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03596000 TLMM\_CENTER\_MPM\_WAKEUP\_INT\_EN\_0****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_CENTER\_MPM\_WAKEUP\_INT\_EN\_0**

Bits	Name	Description
9	GPIO_82	0x0: DISABLE 0x1: ENABLE
8	GPIO_31	0x0: DISABLE 0x1: ENABLE
7	GPIO_30	0x0: DISABLE 0x1: ENABLE
6	GPIO_29	0x0: DISABLE 0x1: ENABLE
5	GPIO_28	0x0: DISABLE 0x1: ENABLE
4	GPIO_22	0x0: DISABLE 0x1: ENABLE
3	GPIO_17	0x0: DISABLE 0x1: ENABLE
2	SSC_31	0x0: DISABLE 0x1: ENABLE
1	SSC_22	0x0: DISABLE 0x1: ENABLE
0	SSC_20	0x0: DISABLE 0x1: ENABLE

**0x03597000 TLMM\_CENTER\_CLK\_GATE\_EN****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_CENTER\_CLK\_GATE\_EN**

Bits	Name	Description
2	AHB_HCLK_EN	0x1: ENABLE 0x0: DISABLE
1	SUMMARY_INTR_EN	0x1: ENABLE 0x0: DISABLE
0	CRIF_READ_EN	0x1: ENABLE 0x0: DISABLE

**0x03597004 TLMM\_CENTER\_IE\_CTRL\_DISABLE****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_CENTER\_IE\_CTRL\_DISABLE**

Bits	Name	Description
0	IE_CTRL_DISABLE	0x1: DISABLE 0x0: ENABLE

**0x03598024 TLMM\_PHASE\_FLAG\_STATUS****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_PHASE\_FLAG\_STATUS**

Bits	Name	Description
31:0	PHASE_FLAG_STATUS	

**0x03599000 TLMM\_GPIO\_GS\_I2C\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_GS\_I2C\_CTL**

Bits	Name	Description
1:0	MODE	0x0: GPIO_MODE 0x1: GS_I2C_MODE 0x2: FS_I2C_MODE 0x3: GS_I2C_HIGH_LOAD_MODE

**0x0359B000+ TLMM\_LPI\_GPIO\_INTR\_CFGn, n=[0..2]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_LPI\_GPIO\_INTR\_CFGn**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: APPS 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0359D000+ TLMM\_LPI\_DIR\_CONN\_INTRn\_CFG\_APPS, n=[0..2]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_LPI\_DIR\_CONN\_INTRn\_CFG\_APPS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
1:0	LPI_GPIO_SEL	

**0x0359E000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_SENSORS, n=[0..1]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_SENSORS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x0359F000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP, n=[0..5]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x035A0000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_RPM, n=[0..0]**  
**0x4\*n**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x00000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

#### TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_RPM

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x035A1000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_HMSS, n=[0..7]**  
**0x4\*n**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x00000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

#### TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_HMSS

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x035A2000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_GSS, n=[0..1]**  
**0x4\*n**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x00000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

#### TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_GSS

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x035A3000+ TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_CDSP, n=[0..1]**  
**0x4\*n**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x000000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_CDSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x03904000 TLMM\_GPIO\_CFG4**

**Type:** RW  
**Clock:** gcc\_tlmm\_ahb\_clk  
**Reset State:** 0x000000001

**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG4**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03904004 TLMM\_GPIO\_IN\_OUT4****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT4**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03904008 TLMM\_GPIO\_INTR\_CFG4****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG4**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390400C TLMM\_GPIO\_INTR\_STATUS4****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS4**

Bits	Name	Description
0	INTR_STATUS	

**0x03904010 TLMM\_GPIO\_ID\_STATUS4****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS4**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03908000 TLMM\_GPIO\_CFG8****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG8**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG8 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03908004 TLMM\_GPIO\_IN\_OUT8****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT8**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03908008 TLMM\_GPIO\_INTR\_CFG8****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG8**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG8 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390800C TLMM\_GPIO\_INTR\_STATUS8****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS8**

Bits	Name	Description
0	INTR_STATUS	

**0x03908010 TLMM\_GPIO\_ID\_STATUS8****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS8**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03909000 TLMM\_GPIO\_CFG9****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG9**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03909004 TLMM\_GPIO\_IN\_OUT9****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT9**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03909008 TLMM\_GPIO\_INTR\_CFG9****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG9**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390900C TLMM\_GPIO\_INTR\_STATUS9****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS9**

Bits	Name	Description
0	INTR_STATUS	

**0x03909010 TLMM\_GPIO\_ID\_STATUS9****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS9**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390A000 TLMM\_GPIO\_CFG10****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG10**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390A004 TLMM\_GPIO\_IN\_OUT10****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT10**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390A008 TLMM\_GPIO\_INTR\_CFG10****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG10**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390A00C TLMM\_GPIO\_INTR\_STATUS10****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS10**

Bits	Name	Description
0	INTR_STATUS	

**0x0390A010 TLMM\_GPIO\_ID\_STATUS10****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS10**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390B000 TLMM\_GPIO\_CFG11****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG11**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390B004 TLMM\_GPIO\_IN\_OUT11****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT11**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390B008 TLMM\_GPIO\_INTR\_CFG11****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG11**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390B00C TLMM\_GPIO\_INTR\_STATUS11****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS11**

Bits	Name	Description
0	INTR_STATUS	

**0x0390B010 TLMM\_GPIO\_ID\_STATUS11****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS11**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390C000 TLMM\_GPIO\_CFG12****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG12**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG12 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390C004 TLMM\_GPIO\_IN\_OUT12****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT12**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390C008 TLMM\_GPIO\_INTR\_CFG12****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG12**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG12 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390C00C TLMM\_GPIO\_INTR\_STATUS12****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS12**

Bits	Name	Description
0	INTR_STATUS	

**0x0390C010 TLMM\_GPIO\_ID\_STATUS12****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS12**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390D000 TLMM\_GPIO\_CFG13****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG13**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390D004 TLMM\_GPIO\_IN\_OUT13****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT13**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390D008 TLMM\_GPIO\_INTR\_CFG13****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG13**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390D00C TLMM\_GPIO\_INTR\_STATUS13****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS13**

Bits	Name	Description
0	INTR_STATUS	

**0x0390D010 TLMM\_GPIO\_ID\_STATUS13****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS13**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390E000 TLMM\_GPIO\_CFG14****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG14**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390E004 TLMM\_GPIO\_IN\_OUT14****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT14**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390E008 TLMM\_GPIO\_INTR\_CFG14****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG14**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390E00C TLMM\_GPIO\_INTR\_STATUS14****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS14**

Bits	Name	Description
0	INTR_STATUS	

**0x0390E010 TLMM\_GPIO\_ID\_STATUS14****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS14**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0390F000 TLMM\_GPIO\_CFG15****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG15**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0390F004 TLMM\_GPIO\_IN\_OUT15****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT15**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0390F008 TLMM\_GPIO\_INTR\_CFG15****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG15**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0390F00C TLMM\_GPIO\_INTR\_STATUS15****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS15**

Bits	Name	Description
0	INTR_STATUS	

**0x0390F010 TLMM\_GPIO\_ID\_STATUS15****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS15**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03918000 TLMM\_GPIO\_CFG24****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG24**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG24 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03918004 TLMM\_GPIO\_IN\_OUT24****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT24**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03918008 TLMM\_GPIO\_INTR\_CFG24****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG24**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG24 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0391800C TLMM\_GPIO\_INTR\_STATUS24****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS24**

Bits	Name	Description
0	INTR_STATUS	

**0x03918010 TLMM\_GPIO\_ID\_STATUS24****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS24**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03919000 TLMM\_GPIO\_CFG25****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG25**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03919004 TLMM\_GPIO\_IN\_OUT25****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT25**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03919008 TLMM\_GPIO\_INTR\_CFG25****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG25**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0391900C TLMM\_GPIO\_INTR\_STATUS25****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS25**

Bits	Name	Description
0	INTR_STATUS	

**0x03919010 TLMM\_GPIO\_ID\_STATUS25****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS25**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0391A000 TLMM\_GPIO\_CFG26****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG26**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0391A004 TLMM\_GPIO\_IN\_OUT26****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT26**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0391A008 TLMM\_GPIO\_INTR\_CFG26****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG26**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0391A00C TLMM\_GPIO\_INTR\_STATUS26****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS26**

Bits	Name	Description
0	INTR_STATUS	

**0x0391A010 TLMM\_GPIO\_ID\_STATUS26****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS26**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0391B000 TLMM\_GPIO\_CFG27****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG27**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0391B004 TLMM\_GPIO\_IN\_OUT27****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT27**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0391B008 TLMM\_GPIO\_INTR\_CFG27****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG27**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0391B00C TLMM\_GPIO\_INTR\_STATUS27****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS27**

Bits	Name	Description
0	INTR_STATUS	

**0x0391B010 TLMM\_GPIO\_ID\_STATUS27****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS27**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03935000 TLMM\_GPIO\_CFG53****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG53**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG53 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03935004 TLMM\_GPIO\_IN\_OUT53****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT53**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03935008 TLMM\_GPIO\_INTR\_CFG53****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG53**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG53 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393500C TLMM\_GPIO\_INTR\_STATUS53****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS53**

Bits	Name	Description
0	INTR_STATUS	

**0x03935010 TLMM\_GPIO\_ID\_STATUS53****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS53**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03936000 TLMM\_GPIO\_CFG54****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG54**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03936004 TLMM\_GPIO\_IN\_OUT54****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT54**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03936008 TLMM\_GPIO\_INTR\_CFG54****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG54**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393600C TLMM\_GPIO\_INTR\_STATUS54****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS54**

Bits	Name	Description
0	INTR_STATUS	

**0x03936010 TLMM\_GPIO\_ID\_STATUS54****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS54**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0393B000 TLMM\_GPIO\_CFG59****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG59**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0393B004 TLMM\_GPIO\_IN\_OUT59****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT59**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0393B008 TLMM\_GPIO\_INTR\_CFG59****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG59**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393B00C TLMM\_GPIO\_INTR\_STATUS59****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS59**

Bits	Name	Description
0	INTR_STATUS	

**0x0393B010 TLMM\_GPIO\_ID\_STATUS59****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS59**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0393C000 TLMM\_GPIO\_CFG60****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG60**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0393C004 TLMM\_GPIO\_IN\_OUT60****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT60**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0393C008 TLMM\_GPIO\_INTR\_CFG60****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG60**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393C00C TLMM\_GPIO\_INTR\_STATUS60****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS60**

Bits	Name	Description
0	INTR_STATUS	

**0x0393C010 TLMM\_GPIO\_ID\_STATUS60****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS60**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0393D000 TLMM\_GPIO\_CFG61****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG61**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG61 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0393D004 TLMM\_GPIO\_IN\_OUT61****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT61**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0393D008 TLMM\_GPIO\_INTR\_CFG61****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG61**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG61 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393D00C TLMM\_GPIO\_INTR\_STATUS61****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS61**

Bits	Name	Description
0	INTR_STATUS	

**0x0393D010 TLMM\_GPIO\_ID\_STATUS61****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS61**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0393E000 TLMM\_GPIO\_CFG62****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG62**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0393E004 TLMM\_GPIO\_IN\_OUT62****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT62**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0393E008 TLMM\_GPIO\_INTR\_CFG62****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG62**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393E00C TLMM\_GPIO\_INTR\_STATUS62****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS62**

Bits	Name	Description
0	INTR_STATUS	

**0x0393E010 TLMM\_GPIO\_ID\_STATUS62****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS62**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0393F000 TLMM\_GPIO\_CFG63****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG63**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0393F004 TLMM\_GPIO\_IN\_OUT63****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT63**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0393F008 TLMM\_GPIO\_INTR\_CFG63****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG63**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0393F00C TLMM\_GPIO\_INTR\_STATUS63****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS63**

Bits	Name	Description
0	INTR_STATUS	

**0x0393F010 TLMM\_GPIO\_ID\_STATUS63****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS63**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03942000 TLMM\_GPIO\_CFG66****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG66**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03942004 TLMM\_GPIO\_IN\_OUT66****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT66**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03942008 TLMM\_GPIO\_INTR\_CFG66****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG66**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394200C TLMM\_GPIO\_INTR\_STATUS66****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS66**

Bits	Name	Description
0	INTR_STATUS	

**0x03942010 TLMM\_GPIO\_ID\_STATUS66****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS66**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03943000 TLMM\_GPIO\_CFG67****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG67**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG67 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03943004 TLMM\_GPIO\_IN\_OUT67****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT67**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03943008 TLMM\_GPIO\_INTR\_CFG67****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG67**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG67 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394300C TLMM\_GPIO\_INTR\_STATUS67****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS67**

Bits	Name	Description
0	INTR_STATUS	

**0x03943010 TLMM\_GPIO\_ID\_STATUS67****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS67**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03944000 TLMM\_GPIO\_CFG68****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG68**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03944004 TLMM\_GPIO\_IN\_OUT68****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT68**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03944008 TLMM\_GPIO\_INTR\_CFG68****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG68**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394400C TLMM\_GPIO\_INTR\_STATUS68****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS68**

Bits	Name	Description
0	INTR_STATUS	

**0x03944010 TLMM\_GPIO\_ID\_STATUS68****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS68**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03945000 TLMM\_GPIO\_CFG69****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG69**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03945004 TLMM\_GPIO\_IN\_OUT69****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT69**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03945008 TLMM\_GPIO\_INTR\_CFG69****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG69**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394500C TLMM\_GPIO\_INTR\_STATUS69****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS69**

Bits	Name	Description
0	INTR_STATUS	

**0x03945010 TLMM\_GPIO\_ID\_STATUS69****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS69**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03946000 TLMM\_GPIO\_CFG70****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG70**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03946004 TLMM\_GPIO\_IN\_OUT70****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT70**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03946008 TLMM\_GPIO\_INTR\_CFG70****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG70**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394600C TLMM\_GPIO\_INTR\_STATUS70****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS70**

Bits	Name	Description
0	INTR_STATUS	

**0x03946010 TLMM\_GPIO\_ID\_STATUS70****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS70**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03947000 TLMM\_GPIO\_CFG71****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG71**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG71 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03947004 TLMM\_GPIO\_IN\_OUT71****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT71**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03947008 TLMM\_GPIO\_INTR\_CFG71****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG71**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG71 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394700C TLMM\_GPIO\_INTR\_STATUS71****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS71**

Bits	Name	Description
0	INTR_STATUS	

**0x03947010 TLMM\_GPIO\_ID\_STATUS71****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS71**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03948000 TLMM\_GPIO\_CFG72****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG72**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03948004 TLMM\_GPIO\_IN\_OUT72****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT72**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03948008 TLMM\_GPIO\_INTR\_CFG72****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG72**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394800C TLMM\_GPIO\_INTR\_STATUS72****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS72**

Bits	Name	Description
0	INTR_STATUS	

**0x03948010 TLMM\_GPIO\_ID\_STATUS72****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS72**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03949000 TLMM\_GPIO\_CFG73****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG73**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x03949004 TLMM\_GPIO\_IN\_OUT73****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT73**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x03949008 TLMM\_GPIO\_INTR\_CFG73****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG73**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394900C TLMM\_GPIO\_INTR\_STATUS73****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS73**

Bits	Name	Description
0	INTR_STATUS	

**0x03949010 TLMM\_GPIO\_ID\_STATUS73****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS73**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0394A000 TLMM\_GPIO\_CFG74****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG74**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0394A004 TLMM\_GPIO\_IN\_OUT74****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT74**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0394A008 TLMM\_GPIO\_INTR\_CFG74****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG74**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394A00C TLMM\_GPIO\_INTR\_STATUS74****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS74**

Bits	Name	Description
0	INTR_STATUS	

**0x0394A010 TLMM\_GPIO\_ID\_STATUS74****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS74**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0394B000 TLMM\_GPIO\_CFG75****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG75**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	

**TLMM\_GPIO\_CFG75 (cont.)**

Bits	Name	Description
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0394B004 TLMM\_GPIO\_IN\_OUT75****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT75**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0394B008 TLMM\_GPIO\_INTR\_CFG75****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG75**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE

**TLMM\_GPIO\_INTR\_CFG75 (cont.)**

Bits	Name	Description
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394B00C TLMM\_GPIO\_INTR\_STATUS75****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS75**

Bits	Name	Description
0	INTR_STATUS	

**0x0394B010 TLMM\_GPIO\_ID\_STATUS75****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS75**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0394C000 TLMM\_GPIO\_CFG76****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_CFG76**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0394C004 TLMM\_GPIO\_IN\_OUT76****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT76**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0394C008 TLMM\_GPIO\_INTR\_CFG76****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_INTR\_CFG76**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394C00C TLMM\_GPIO\_INTR\_STATUS76****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS76**

Bits	Name	Description
0	INTR_STATUS	

**0x0394C010 TLMM\_GPIO\_ID\_STATUS76****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_ID\_STATUS76**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0394D000 TLMM\_GPIO\_CFG77****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG77**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0394D004 TLMM\_GPIO\_IN\_OUT77****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_IN\_OUT77**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0394D008 TLMM\_GPIO\_INTR\_CFG77****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG77**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394D00C TLMM\_GPIO\_INTR\_STATUS77****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS77**

Bits	Name	Description
0	INTR_STATUS	

**0x0394D010 TLMM\_GPIO\_ID\_STATUS77****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS77**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x0394E000 TLMM\_GPIO\_CFG78****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_CFG78**

Bits	Name	Description
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	DRV_STRENGTH	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	FUNC_SEL	
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0394E004 TLMM\_GPIO\_IN\_OUT78****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_GPIO\_IN\_OUT78**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0394E008 TLMM\_GPIO\_INTR\_CFG78****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000E2**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_CFG78**

Bits	Name	Description
8	DIR_CONN_EN	0x1: ENABLE 0x0: DISABLE
7:5	TARGET_PROC	0x0: SENSORS 0x1: LPA_DSP 0x2: RPM 0x3: HMSS 0x4: GSS 0x5: TZ 0x6: CDSP 0x7: NONE
4	INTR_RAW_STATUS_EN	0x1: ENABLE 0x0: DISABLE
3:2	INTR_DECT_CTL	0x0: LEVEL 0x1: POS_EDGE 0x2: NEG_EDGE 0x3: DUAL_EDGE
1	INTR_POL_CTL	0x1: POLARITY_1 0x0: POLARITY_0
0	INTR_ENABLE	0x1: ENABLE 0x0: DISABLE

**0x0394E00C TLMM\_GPIO\_INTR\_STATUS78****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_INTR\_STATUS78**

Bits	Name	Description
0	INTR_STATUS	

**0x0394E010 TLMM\_GPIO\_ID\_STATUS78****Type:** R**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_GPIO\_ID\_STATUS78**

Bits	Name	Description
0	GPIO_ID_STATUS	

**0x03996000 TLMM\_NORTH\_MPM\_WAKEUP\_INT\_EN\_0****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_MPM\_WAKEUP\_INT\_EN\_0**

Bits	Name	Description
24	GPIO_78	0x0: DISABLE 0x1: ENABLE
23	GPIO_77	0x0: DISABLE 0x1: ENABLE
22	GPIO_76	0x0: DISABLE 0x1: ENABLE
21	GPIO_75	0x0: DISABLE 0x1: ENABLE
20	GPIO_74	0x0: DISABLE 0x1: ENABLE

**TLMM\_NORTH\_MPM\_WAKEUP\_INT\_EN\_0 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
19	GPIO_73	0x0: DISABLE 0x1: ENABLE
18	GPIO_72	0x0: DISABLE 0x1: ENABLE
17	GPIO_71	0x0: DISABLE 0x1: ENABLE
16	GPIO_70	0x0: DISABLE 0x1: ENABLE
15	GPIO_69	0x0: DISABLE 0x1: ENABLE
14	GPIO_68	0x0: DISABLE 0x1: ENABLE
13	GPIO_67	0x0: DISABLE 0x1: ENABLE
12	GPIO_66	0x0: DISABLE 0x1: ENABLE
11	GPIO_60	0x0: DISABLE 0x1: ENABLE
10	GPIO_54	0x0: DISABLE 0x1: ENABLE
9	GPIO_25	0x0: DISABLE 0x1: ENABLE
8	GPIO_13	0x0: DISABLE 0x1: ENABLE
7	GPIO_10	0x0: DISABLE 0x1: ENABLE
6	GPIO_9	0x0: DISABLE 0x1: ENABLE
5	SRST_N	0x0: DISABLE 0x1: ENABLE
4	SDC2_CMD	0x0: DISABLE 0x1: ENABLE
3	SDC2_DATA_3	0x0: DISABLE 0x1: ENABLE
2	SDC2_DATA_1	0x0: DISABLE 0x1: ENABLE
1	SDC1_DATA_3	0x0: DISABLE 0x1: ENABLE

**TLMM\_NORTH\_MPM\_WAKEUP\_INT\_EN\_0 (cont.)**

Bits	Name	Description
0	SDC1_DATA_1	0x0: DISABLE 0x1: ENABLE

**0x03997000 TLMM\_NORTH\_CLK\_GATE\_EN****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_CLK\_GATE\_EN**

Bits	Name	Description
2	AHB_HCLK_EN	0x1: ENABLE 0x0: DISABLE
1	SUMMARY_INTR_EN	0x1: ENABLE 0x0: DISABLE
0	CRIF_READ_EN	0x1: ENABLE 0x0: DISABLE

**0x03997004 TLMM\_NORTH\_IE\_CTRL\_DISABLE****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_IE\_CTRL\_DISABLE**

Bits	Name	Description
0	IE_CTRL_DISABLE	0x1: DISABLE 0x0: ENABLE

**0x03998000 TLMM\_INT\_JTAG\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_INT\_JTAG\_CTL**

Bits	Name	Description
0	TAP_SEL	0x0: ENABLE_MSM_ONLY 0x1: ENABLE_ALL_CHAINS

**0x03998004 TLMM\_ETM\_MODE****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_por\_ares**TLMM\_ETM\_MODE**

Bits	Name	Description
1:0	TRACE_OVER_SDC2	0x0: MODE0 0x1: MODE1 0x2: MODE2 0x3: MODE3

**0x03999000 TLMM\_MODE\_PULL\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000005**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_MODE\_PULL\_CTL**

Bits	Name	Description
3:2	MODE_1_PULL	
1:0	MODE_0_PULL	

**0x0399A000 TLMM\_SDC1\_HDRV\_PULL\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00008BDB**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_SDC1\_HDRV\_PULL\_CTL**

Bits	Name	Description
16:15	SDC1_RCLK_PULL	
14:13	SDC1_CLK_PULL	
12:11	SDC1_CMD_PULL	
10:9	SDC1_DATA_PULL	
8:6	SDC1_CLK_HDRV	
5:3	SDC1_CMD_HDRV	
2:0	SDC1_DATA_HDRV	

**0x0399B000 TLMM\_SDC2\_HDRV\_PULL\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000BDB**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_SDC2\_HDRV\_PULL\_CTL**

Bits	Name	Description
15:14	SDC2_CLK_PULL	
13	SDC2_HYS_CTL	0x0: DISABLE 0x1: ENABLE
12:11	SDC2_CMD_PULL	
10:9	SDC2_DATA_PULL	
8:6	SDC2_CLK_HDRV	
5:3	SDC2_CMD_HDRV	
2:0	SDC2_DATA_HDRV	

**0x0399D000 TLMM\_JTAG\_HDRV\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000003DB**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_JTAG\_HDRV\_CTL**

Bits	Name	Description
9:8	TDI_PULL	
7:6	TMS_PULL	
5:3	TMS_HDRV	
2:0	TDO_HDRV	

**0x0399E000 TLMM\_RESOUT\_HDRV\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_RESOUT\_HDRV\_CTL**

Bits	Name	Description
2:0	RESOUT_N_HDRV	

**0x0399F000 TLMM SNDWIRE\_SLIMBUS\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM SNDWIRE\_SLIMBUS\_CTL**

Bits	Name	Description
3:2	SNDWIRE_DATA_SR_CTL_EN	
1:0	SNDWIRE_CLK_SR_CTL_E_N	

**0x039A0000 TLMM\_BTFM\_SLIMBUS\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000000**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_BTFM\_SLIMBUS\_CTL**

Bits	Name	Description
1:0	SR_CTL_EN	

**0x039A1000 TLMM\_UFS\_REF\_CLK\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x0000000B**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_UFS\_REF\_CLK\_CTL**

Bits	Name	Description
4:3	UFS_REF_CLK_PULL	
2:0	UFS_REF_CLK_HDRV	

**0x039A2000 TLMM\_DDR\_RESET\_N\_HDRV\_PULL\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000003**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_DDR\_RESET\_N\_HDRV\_PULL\_CTL**

Bits	Name	Description
2:0	DDR_RESET_N_HDRV	

**0x039A3000 TLMM\_UFS\_RESET\_CTL****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x0000000B**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_UFS\_RESET\_CTL**

Bits	Name	Description
4:3	UFS_RESET_PULL	
2:0	UFS_RESET_HDRV	

**0x039A3004 TLMM\_UFS\_RESET****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000001**Reset Name:** gcc\_tlmm\_por\_ares**TLMM\_UFS\_RESET**

Bits	Name	Description
0	UFS_RESET	

**0x039A5000+ TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_SENSORS, n=[0..1]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_SENSORS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x039A6000+ TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP, n=[0..5]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x000000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x039A7000+ TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_RPM, n=[0..0]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_RPM**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x039A8000+ TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_HMSS, n=[0..7]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_HMSS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x039A9000+ TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_GSS, n=[0..1]****0x4\*n****Type:** RW**Clock:** gcc\_tlmm\_ahb\_clk**Reset State:** 0x00000100**Reset Name:** gcc\_tlmm\_ahb\_ares**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_GSS**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

**0x039AA000+TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_CDSP, n=[0..1]**

**0x4\*n**

**Type:** RW

**Clock:** gcc\_tlmm\_ahb\_clk

**Reset State:** 0x00000100

**Reset Name:** gcc\_tlmm\_ahb\_ares

**TLMM\_NORTH\_DIR\_CONN\_INTRn\_CFG\_CDSP**

Bits	Name	Description
8	POLARITY	0x1: POLARITY_1 0x0: POLARITY_0
6:0	GPIO_SEL	

2019-09-18 18:26:30 PDT  
xiaoyongjie15@huqin.com

## 4 Clock controller registers

---

### 0x05065100 GPUCC\_RCG\_SRC\_ACTIVE\_CTL

Type: RW

Clock: AHB CLK

Reset State: 0x0000001F

RCG\_SRC\_ACTIVE

#### GPUCC\_RCG\_SRC\_ACTIVE\_CTL

Bits	Name	Description
4	GPU_PLL1_MAIN_ACTIVE_SW_OVERRIDE	RESERVED
3	GPU_PLL0_MAIN_ACTIVE_SW_OVERRIDE	RESERVED
2	GFX_ZMEAS_ACTIVE_SW_OVERRIDE	GFX_ZMEAS_ACTIVE_SW_OVERRIDE
1	RESERVED	RESERVED
0	RESERVED	RESERVED

### 0x05065104 GPUCC\_RCG\_SRC\_ACTIVE\_STATUS

Type: R

Clock: AHB CLK

Reset State: 0x0000001F

RCG\_SRC\_ACTIVE

#### GPUCC\_RCG\_SRC\_ACTIVE\_STATUS

Bits	Name	Description
4	GPU_PLL1_MAIN_ACTIVE	RESERVED
3	GPU_PLL0_MAIN_ACTIVE	RESERVED

**GPUCC\_RCG\_SRC\_ACTIVE\_STATUS (cont.)**

Bits	Name	Description
2	GFX_ZMEAS_ACTIVE	GFX_ZMEAS_ACTIVE
1	RESERVED	RESERVED
0	RESERVED	RESERVED

**0x05065130 GPUCC\_GX\_DOMAIN\_MISC****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

Misc registers for Oxili Ocmem in GX rail

**GPUCC\_GX\_DOMAIN\_MISC**

Bits	Name	Type	Description
21	RAC_ENR_STATUS	R	GDSC in gfx wrapper that controls RAC status for enr 0 : Not Active 1 : Active
20	RAC_ENF_STATUS	R	GDSC in gfx wrapper that controls RAC status for enf 0 : Not Active 1 : Active
19	SPTP_ENR_STATUS	R	GDSC in gfx wrapper that controls SPTP status for enr 0 : Not Active 1 : Active
18	SPTP_ENF_STATUS	R	GDSC in gfx wrapper that controls SPTP status for enf 0 : Not Active 1 : Active
4	GPU_GX_GMEM_RESET		GPU_GX GMEM reset with GX powers on 0 : Not Active 1 : Active
0	GPU_GX_GMEM_CLAMP_I_O		GPU_GX GMEM GX clamp I/O enable 0 : Not Active 1 : Active

**0x05065134 GPUCC\_GX\_DOMAIN\_MISC2****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

Misc registers for GFX core in GX rail

#### **GPUCC\_GX\_DOMAIN\_MISC2**

Bits	Name	Description
0	GFX_CLAMP_MEM	This is a software bit (used as backup) for clamp_mem in GFX HM 0 : Not Active 1 : Active

#### **0x05065150 GPUCC\_GFX3D\_ZMEAS\_CONFIG**

**Type:** RW

**Clock:** AHB CLK

**Reset State:** 0x00000000

Zmeas config registers for the GFX3D clock network

#### **GPUCC\_GFX3D\_ZMEAS\_CONFIG**

Bits	Name	Description
31:24	RESERVED	Reserved
23:5	ZMEAS_COUNT	19-bit count value for setting the clock divides value. This directly determines the resonance cycle time period.
4	RESERVE_BIT_4	Reserved bits
3	ZMEAS_CLK_R_ENA	Gates the output clock of the macro regardless of how bypass is set. There needs to be an active FSM clock (zmeas_en = 1) in order for this enable to take effect.
2	ZMEAS_CGC_ENA	Module software bit which activates the internal clock CGC so that activity is present only when the macro is in use. When set low the internal FSM clock is not active to save power.
1	ZMEAS_BYPASS	Active high enable which passes the input clock to the units output pin clk_divn so as to appear continuous and undivided. This pin function can be gated by the async_clk_r_ena pin as described below.
0	ZMEAS_SW_RESET	Active high software reset.

#### **0x05065160 GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_REQ\_SW**

**Type:** RW

**Clock:** AHB CLK

**Reset State:** 0x00000000

GPU GDS\_HW\_CTL Halt Request Register for SMMU client software Bypass mode

**GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_REQ\_SW**

Bits	Name	Description
0	SMMU_HALT_REQ_SW	SMMU Halt Request from software 0: No Halt Request 1: Halt Request asserted

**0x05065164 GPUCC\_GDS\_HW\_CTL\_DVM\_HALT\_REQ\_SW****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

GPU GDS\_HW\_CTL Halt Request Register for DVM client software Bypass mode

**GPUCC\_GDS\_HW\_CTL\_DVM\_HALT\_REQ\_SW**

Bits	Name	Description
0	DVM_HALT_REQ_SW	DVM Halt Request from software 0: No Halt Request 1: Halt Request asserted

**0x05065168 GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_STATUS****Type:** R**Clock:** AHB CLK**Reset State:** 0x00000001

SMMU Halt Status Register

**GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_STATUS**

Bits	Name	Description
1	SMMU_HALT_ACK	SMMU HALT ACK Status from hardware 0: No Acknowledge for the Halt Request 1: Halt Request Acknowledged
0	SMMU_HALT_IDLE	SMMU IDLE Status from hardware 0: Busy 1: Idle

**0x0506516C GPUCC\_GDS\_HW\_CTL\_DVM\_NIU\_STATUS****Type:** R**Clock:** AHB CLK**Reset State:** 0x00000000

DVM NIU Power Down Idle Status Register

**GPUCC\_GDS\_HW\_CTL\_DVM\_NIU\_STATUS**

Bits	Name	Description
1	DVM_NIU_IDLEACK	DVM NIU Power Down ACK from hardware 0: Power Down Prohibited 1: Power Down Allowed The software can power down the NIU only when DVM_NIU_IDLEACK and DVM_NIU_IDLE are both high
0	DVM_NIU_IDLE	DVM NIU IDLE Status 0: Busy 1: Idle

**0x05065200 GPUCC\_GPU\_CX\_GDSC\_EN\_ACK\_CTL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

GPU CX GDSC Enable Ack software Control

**GPUCC\_GPU\_CX\_GDSC\_EN\_ACK\_CTL**

Bits	Name	Type	Description
5	EN_FEW_ACK_SW		Enable Few Ack from software 0: Software value of Enable Few Ack is 0 1: Software value of Enable Few Ack is 1
4	EN_REST_ACK_SW		Enable Rest Ack from software 0: Software value of Enable Rest Ack is 0 1: Software value of Enable Rest Ack is 1
1	EN_FEW_ACK_STATUS	R	Hardware Status of Enable Few Ack 0: Enable Few Ack is de-asserted 1: Enable Few Ack is asserted
0	EN_REST_ACK_STATUS	R	Hardware Status of Enable Rest Ack 0: Enable Rest Ack is de-asserted 1: Enable Rest Ack is asserted

**0x05065204 GPUCC\_GPU\_GX\_GDSC\_EN\_ACK\_CTL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

GPU GX GDSC Enable Ack software Control

**GPUCC\_GPU\_GX\_GDSC\_EN\_ACK\_CTL**

Bits	Name	Type	Description
5	EN_FEW_ACK_SW		Enable Few Ack from software 0: Software value of Enable Few Ack is 0 1: Software value of Enable Few Ack is 1
4	EN_REST_ACK_SW		Enable Rest Ack from software 0: Software value of Enable Rest Ack is 0 1: Software value of Enable Rest Ack is 1
1	EN_FEW_ACK_STATUS	R	Hardware Status of Enable Few Ack 0: Enable Few Ack is de-asserted 1: Enable Few Ack is asserted
0	EN_REST_ACK_STATUS	R	Hardware Status of Enable Rest Ack 0: Enable Rest Ack is de-asserted 1: Enable Rest Ack is asserted

**0x05065240 GPUCC\_FREQUENCY\_LIMITER\_IRQ\_STATUS****Type:** R**Clock:** AHB CLK**Reset State:** 0x00000000

Frequency Limiter Interrupt STATUS

**GPUCC\_FREQUENCY\_LIMITER\_IRQ\_STATUS**

Bits	Name	Description
0	STATUS	Frequency Limiter Interrupt Status 0: Interrupt is not asserted 1: Interrupt is asserted

**0x05065244 GPUCC\_FREQUENCY\_LIMITER\_IRQ\_MASK****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

## Frequency Limiter Interrupt MASK

**GPUCC\_FREQUENCY\_LIMITER\_IRQ\_MASK**

Bits	Name	Description
0	MASK	Frequency Limiter Mask bit 0: Interrupt is Masked 1: Interrupt is allowed

**0x05065248 GPUCC\_FREQUENCY\_LIMITER\_IRQ\_CLEAR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

Frequency Limiter Interrupt CLEAR

**GPUCC\_FREQUENCY\_LIMITER\_IRQ\_CLEAR**

Bits	Name	Description
0	CLEAR	Writing 1 to this register will clear the Frequency Limiter Interrupt

**0x05065250 GPUCC\_IDLE\_REQ\_FSM\_STATE****Type:** R**Clock:** AHB CLK**Reset State:** 0x00000000

Idle Request FSM State

**GPUCC\_IDLE\_REQ\_FSM\_STATE**

Bits	Name	Description
1:0	FSM_STATE	Idle Request FSM State

**0x05066000 GPUCC\_GPU\_CX\_BCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

BCR Registers for GPU

**GPUCC\_GPU\_CX\_BCR**

Bits	Name	Description
0	BLK_ARES	The block asynchronous reset bit, when set, the block clocks are disabled and the ares_out signals go active. 0 : Not Active 1 : Active

**0x05066004 GPUCC\_GPU\_CX\_GDSCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00222001

GDS Registers for GPU

**GPUCC\_GPU\_CX\_GDSCR**

Bits	Name	Type	Description
31	PWR_ON	R	PWR State 0 : Power collapsed 1 : Power on
30:27	GDSC_STATE	R	GDS FSM State 0000 : PowerOff 0001 : AssertAresAndRampUp_en_few 0010 : DeassertAres 0011 : RampUp_en_rest 0100 : PowerOn 0101 : UnclampIO 0110 : NoRetainFF 0111 : RestoreFF 1000 : NA 1001 : NA 1010 : NA 1011 : NA 1100 : DisableClk 1101 : RetainFF 1110 : NA 1111 : ClampIOAndSaveFF

**GPUCC\_GPU\_CX\_GDSCR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
23:20	EN_REST_WAIT		Wait cntr for the remaining PD switches to power on 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle
19:16	EN_FEW_WAIT		Wait cntr for the first few PD switches to power on 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle

**GPUCC\_GPU\_CX\_GDSCR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
15:12	CLK_DIS_WAIT		Wait while all clocks in this PD are turned off 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle
11	RETAIN_FF_ENABLE		Retain ff enable 0 : Disabled - Default during chip power up 1 : Enabled - Reconfigured before power down
10	RESTORE		Restore 0 : Not Active 1 : Active
9	SAVE		Save 0 : Not Active 1 : Active
8	RETAIN		Retain 0 : Not Active 1 : Active
7	EN_REST		Enable Rest 0 : Not Active 1 : Active
6	EN_FEW		Enable Few 0 : Not Active 1 : Active
5	CLAMP_IO		Clamp I/O 0 : Not Active 1 : Active
4	CLK_DISABLE		Clk Disable 0 : Not Active 1 : Active

**GPUCC\_GPU\_CX\_GDSCR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
3	PD_ARES		Reset 0 : Not Active 1 : Active
2	SW_OVERRIDE		Software Override 0 : GDSC sequence based on hardware FSM 1 : GDSC sequence based on software
1	HW_CONTROL		Hardware Control 0 : Power off by hardware trigger 1 : Power on by hardware trigger
0	SW_COLLAPSE		Software Collapse 0 : Power on through software 1 : Power off through software, default

**0x05066008 GPUCC\_GPU\_CX\_GDS\_HW\_CTRL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00005FF4**GPU\_CX\_GDS\_HW\_CTRL****GPUCC\_GPU\_CX\_GDS\_HW\_CTRL**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
31	PWR_ON_STATUS	R	PWR_ON_STATUS
29	DVM_HALT1_PWR_DOWN_ACK_STATUS	R	DVM_HALT1_PWR_DOWN_ACK_STATUS
28	DVM_HALT1_PWR_UP_ACK_STATUS	R	DVM_HALT1_PWR_UP_ACK_STATUS
27	HALT1_PWR_DOWN_ACK_STATUS	R	HALT1_PWR_DOWN_ACK_STATUS
23	HALT1_PWR_UP_ACK_STATUS	R	HALT1_PWR_UP_ACK_STATUS
19	HALT2_PWR_DOWN_ACK_STATUS	R	HALT2_PWR_DOWN_ACK_STATUS
15	HALT2_PWR_UP_ACK_STATUS	R	HALT2_PWR_UP_ACK_STATUS
14	COLLAPSE_OUT	R	COLLAPSE_OUT
12:5	HALT_ACK_TIMEOUT		HALT_ACK_TIMEOUT
4:1	GDS_HW_STATE	R	GDS_HW_STATE

**GPUCC\_GPU\_CX\_GDS\_HW\_CTRL (cont.)**

Bits	Name	Type	Description
0	SW_OVERRIDE		SW_OVERRIDE

**0x05066020 GPUCC\_CXO\_CBCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

CBC Registers for CXO for GPU CX and GX

**GPUCC\_CXO\_CBCR**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off 0 : Clock is ON 1 : Clock is OFF
0	CLK_ENABLE		Software Clock Enable Bit 0 : Clock Disabled. 1 : Clock Enabled.

**0x05066070 GPUCC\_GFX3D\_CMD\_RCGR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

Root Clock Generator Command Register GFX3D

**GPUCC\_GFX3D\_CMD\_RCGR**

Bits	Name	Type	Description
31	ROOT_OFF	R	The ROOT_OFF status allows software to determine if the generator is actively producing a clock 0 : Active 1 : Not Active
4	DIRTY_CFG_RCGR	R	Dirty bit for the CFG_RCGR register. This bit is active if the CFG_RCGR register is written, and will clear after the update bit asserts. 0 : Not Active 1 : Active

**GPUCC\_GFX3D\_CMD\_RCGR (cont.)**

Bits	Name	Type	Description
1	ROOT_EN		The ROOT_EN bit controls the output of the clock generator. This enable may be ORed with a hardware clock enable and then sent to the RCG. 0 : Disabled 1 : Enabled
0	UPDATE		This UPDATE bit is software and hardware modified. Software sets the bit when the configuration values are programmed, and when they have taken effect, then hardware clears the bit to 0. 0 : Not Active 1 : Active

**0x05066074 GPUCC\_GFX3D\_CFG\_RCGR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

Clock Generator Config Register for GFX3D

**GPUCC\_GFX3D\_CFG\_RCGR**

Bits	Name	Description
16	RCGLITE_DISABLE	Bit to control GFX3D RCGLite feature 0 : Feature Enabled 1 : Feature Disabled
10:8	SRC_SEL	Root Source Select 000 : cxo 001 : mmpll0 010 : mmpll9 011 : mmpll2 100 : mmpll8 101 : gcc_gpll0 110 : zmeas_clk 111 :

**GPUCC\_GFX3D\_CFG\_RCGR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
4:0	SRC_DIV	<p>The SRC_DIV bits define the integer and half integer portion of the divider setting.</p> <p>00000 : Div 1      00001 : Div 1      00010 : Div 1.5      00011 : Div 2      00100 : Div 2.5      00101 : Div 3      00110 : Div 3.5      00111 : Div 4      01000 : Div 4.5      01001 : Div 5      01010 : Div 5.5      01011 : Div 6      01100 : Div 6.5      01101 : Div 7      01110 : Div 7.5      01111 : Div 8      10000 : Div 8.5      10001 : Div 9      10010 : Div 9.5      10011 : Div 10      10100 : Div 10.5      10101 : Div 11      10110 : Div 11.5      10111 : Div 12      11000 : Div 12.5      11001 : Div 13      11010 : Div 13.5      11011 : Div 14      11100 : Div 14.5      11101 : Div 15      11110 : Div 15.5      11111 : Div 16</p>

**0x05066090 GPUCC\_GPU\_GX\_BCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

BCR Registers for GPU\_GX

**GPUCC\_GPU\_GX\_BCR**

Bits	Name	Description
0	BLK_ARES	The block asynchronous reset bit, when set, the block clocks are disabled and the ares_out signals go active. 0 : Not Active 1 : Active

**0x05066094 GPUCC\_GPU\_GX\_GDSCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00222001

GDS Registers for GPU\_GX

**GPUCC\_GPU\_GX\_GDSCR**

Bits	Name	Type	Description
31	PWR_ON	R	PWR State 0 : Power collapsed 1 : Power on
30:27	GDSC_STATE	R	GDS FSM State 0000 : PowerOff 0001 : AssertAresAndRampUp_en_few 0010 : DeassertAres 0011 : RampUp_en_rest 0100 : PowerOn 0101 : UnclampIO 0110 : NoRetainFF 0111 : RestoreFF 1000 : NA 1001 : NA 1010 : NA 1011 : NA 1100 : DisableClk 1101 : RetainFF 1110 : NA 1111 : ClampIOAndSaveFF

**GPUCC\_GPU\_GX\_GDSCR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
23:20	EN_REST_WAIT		Wait cntr for the remaining PD switches to power on 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle
19:16	EN_FEW_WAIT		Wait cntr for the first few PD switches to power on 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle

**GPUCC\_GPU\_GX\_GDSCR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
15:12	CLK_DIS_WAIT		Wait while all clocks in this PD are turned off 0000 : cntr value is cxo x 1 cycle 0001 : cntr value is cxo x 2 cycle 0010 : cntr value is cxo x 4 cycle [default] 0011 : cntr value is cxo x 8 cycle 0100 : cntr value is cxo x 16 cycle 0101 : cntr value is cxo x 32 cycle 0110 : cntr value is cxo x 64 cycle 0111 : cntr value is cxo x 128 cycle 1000 : cntr value is cxo x 256 cycle 1001 : cntr value is cxo x 512 cycle 1010 : cntr value is cxo x 1024 cycle 1011 : cntr value is cxo x 2048 cycle 1100 : cntr value is cxo x 2048 cycle 1101 : cntr value is cxo x 2048 cycle 1110 : cntr value is cxo x 2048 cycle 1111 : cntr value is cxo x 2048 cycle
11	RETAIN_FF_ENABLE		Retain ff enable 0 : Disabled - Default during chip power up 1 : Enabled - Reconfigured before power down
10	RESTORE		Restore 0 : Not Active 1 : Active
9	SAVE		Save 0 : Not Active 1 : Active
8	RETAIN		Retain 0 : Not Active 1 : Active
7	EN_REST		Enable Rest 0 : Not Active 1 : Active
6	EN_FEW		Enable Few 0 : Not Active 1 : Active
5	CLAMP_IO		Clamp I/O 0 : Not Active 1 : Active
4	CLK_DISABLE		Clk Disable 0 : Not Active 1 : Active

**GPUCC\_GPU\_GX\_GDSCR (cont.)**

Bits	Name	Type	Description
3	PD_ARES		Reset 0 : Not Active 1 : Active
2	SW_OVERRIDE		Software Override 0 : GDSC sequence based on hardware FSM 1 : GDSC sequence based on software
1	HW_CONTROL		Hardware Control 0 : Power off by hardware trigger 1 : Power on by hardware trigger
0	SW_COLLAPSE		Software Collapse 0 : Power on through software 1 : Power off through software, default

**0x05066098 GPUCC\_GFX3D\_CBCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80008620

PSCBC Registers for GPU\_GX\_GFX3D

**GPUCC\_GFX3D\_CBCR**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off 0 : Clock is ON 1 : Clock is OFF
30:28	RESERVED		
15	RESERVED		
14	FORCE_MEM_CORE_ON		Force Mem Core bits to be on 0 : Memory Core can be powered down by the clock branch slp control (default) 1 : Force Memory Core to be powered on
13	FORCE_MEM_PERIPH_ON		Force Mem Periph logics to be on 0 : Memory Periphery logic can be powered down by the clock branch slp control (default) 1 : Force Memory Periphery logic to be powered on
12	FORCE_MEM_PERIPH_OF		Force Mem Periph logics to be off 0 : Memory Periphery logic is powered on by the clock branch slp control (default) 1 : Force Memory Periphery logic to be powered off

**GPUCC\_GFX3D\_CBCR (cont.)**

Bits	Name	Type	Description
11:8	WAKEUP		<p>This latency formula is valid for w[3:0] = 0001 to 1111  If async_hw/sw_clk_en is toggled from LOW to HIGH in the middle of clock cycle (@negedge of clk_in) , the latency is = <math>2*(16*w[3:0]) + 5.5</math> * ( Timeperiod of clk_in)  If async_hw/sw_clk_en is toggled from LOW to HIGH at the posedge of the clock cycle (@posedge of clk_in) , the latency = <math>2*(16*w[3:0]) + 6</math> * ( Timeperiod of clk_in)  Here the mem_core_on will be asserted first followed by mem_periph_on after a certain delay controlled by w[3:0]. Then after some delay the clk_out starts toggling.  w[3:0] async_clk_en to clk_out   async_clk_en to mem_periph_on   mem_core_on to mem_periph_on latency  0 7 5 1  1 38 20 16  2 70 36 32 (default)  3 102 52 48  4 134 68 64  5 166 84 80  6 198 100 96  7 230 116 112  8 262 132 128  9 294 148 144  10 326 164 160  11 358 180 176  12 390 196 192  13 422 212 208  14 454 228 224  15 486 244 240 </p>

**GPUCC\_GFX3D\_CBCR (cont.)**

Bits	Name	Type	Description
7:4	SLEEP		<p>This latency formula is valid for s[3:0] = 0001 to 1111      clk_out is gated immediately , but the mem_periph_on will be turned off after a delay first followed by mem_core_on after a delay</p> <p>If async_hw/sw_clk_en is toggled from HIGH to LOW in the @posedge of clk_in / in the middle of clock cycle (@negedge of clk_in)</p> <p>s[3:0] async_clk_en to mem_periph_on   async_clk_en to mem_core_on   mem_periph_on to mem_core_on</p> <p>0 4 5 1      1 20 37 17      2 36 69 33 (default)      3 52 101 49      4 68 133 65      5 84 165 81      6 100 197 97      7 116 229 113      8 132 261 129      9 148 293 145      10 164 325 161      11 180 357 177      12 196 389 193      13 212 421 209      14 228 453 225      15 244 485 241</p>
1	HW_CTL		<p>Hardware Clock Control Bit, used for PwrActive signal from FSM</p> <p>0 : PwrActive Disabled.      1 : PwrActive Enabled.</p>
0	CLK_ENABLE		<p>Software Clock Enable Bit</p> <p>0 : Clock Disabled      1 : Clock Enabled</p>

**0x050660B0 GPUCC\_RBBMTIMER\_CMD\_RCGR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

Root Clock Generator Command Register RBBMTIMER

**GPUCC\_RBBMTIMER\_CMD\_RCGR**

Bits	Name	Type	Description
31	ROOT_OFF	R	The ROOT_OFF status allows software to determine if the generator is actively producing a clock 0 : Active 1 : Not Active
4	DIRTY_CFG_RCGR	R	Dirty bit for the CFG_RCGR register. This bit is active if the CFG_RCGR register is written, and will clear after the update bit asserts. 0 : Not Active 1 : Active
1	ROOT_EN		The ROOT_EN bit controls the output of the clock generator. This enable may be ORed with a hardware clock enable and then sent to the RCG. 0 : Disabled 1 : Enabled
0	UPDATE		This UPDATE bit is software and hardware modified. Software sets the bit when the configuration values are programmed, and when they have taken effect, then hardware clears the bit to 0. 0 : Not Active 1 : Active

**0x050660B4 GPUCC\_RBBMTIMER\_CFG\_RCGR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000

Clock Generator Config Register for RBBMTIMER

**GPUCC\_RBBMTIMER\_CFG\_RCGR**

Bits	Name	Description
10:8	SRC_SEL	Root Source Select 000 : cxo 001 : mmpll0 010 : gnd 011 : gnd 100 : gnd 101 : gcc_gpll0 110 : gcc_gpll0_div2 111 :

**GPUCC\_RBBMTIMER\_CFG\_RCGR (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
4:0	SRC_DIV	<p>The SRC_DIV bits define the integer and half integer portion of the divider setting.</p> <p>00000 : Div 1      00001 : Div 1      00010 : Div 1.5      00011 : Div 2      00100 : Div 2.5      00101 : Div 3      00110 : Div 3.5      00111 : Div 4      01000 : Div 4.5      01001 : Div 5      01010 : Div 5.5      01011 : Div 6      01100 : Div 6.5      01101 : Div 7      01110 : Div 7.5      01111 : Div 8      10000 : Div 8.5      10001 : Div 9      10010 : Div 9.5      10011 : Div 10      10100 : Div 10.5      10101 : Div 11      10110 : Div 11.5      10111 : Div 12      11000 : Div 12.5      11001 : Div 13      11010 : Div 13.5      11011 : Div 14      11100 : Div 14.5      11101 : Div 15      11110 : Div 15.5      11111 : Div 16</p>

**0x050660D0 GPUCC\_RBBMTIMER\_CBCR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

CBC Registers for GPU\_GX\_RBBMTIMER

**GPUCC\_RBBMTIMER\_CBCR**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off 0 : Clock is ON 1 : Clock is OFF
0	CLK_ENABLE		Software Clock Enable Bit 0 : Clock Disabled 1 : Clock Enabled

**0x05066200 GPUCC\_GFX3D\_AON\_DIV\_CTRL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

Register to control Divide Ratio of CDIV inside GPU for GFX3D\_AON clk

**GPUCC\_GFX3D\_AON\_DIV\_CTRL**

Bits	Name	Description
1:0	DIV_BY	Divider Setting 00 : Div 1 01 : Div 2 10 : Div 3 11 : Div 4

**0x05066220 GPUCC\_GFX\_HM\_BHS\_STRENGTH\_CTRL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x0000003F

Register to control GFX HM BHS Strength

**GPUCC\_GFX\_HM\_BHS\_STRENGTH\_CTRL**

Bits	Name	Description
5:2	BHS_REST_STRENGTH	4 bits for enable rest strength selection
1:0	BHS_FEW_STRENGTH	2 bits for enable few strength selection

**0x05066230 GPUCC\_GFX\_SPTP\_BHS\_STRENGTH\_CTRL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x0000003F

Register to control GFX SPTP BHS Strength

**GPUCC\_GFX\_SPTP\_BHS\_STRENGTH\_CTRL**

Bits	Name	Description
5:2	BHS_REST_STRENGTH	4 bits for enable rest strength selection
1:0	BHS_FEW_STRENGTH	2 bits for enable few strength selection

**0x05066240 GPUCC\_GFX\_RAC\_BHS\_STRENGTH\_CTRL****Type:** RW**Clock:** AHB CLK**Reset State:** 0x0000003F

Register to control GFX RAC BHS Strength

**GPUCC\_GFX\_RAC\_BHS\_STRENGTH\_CTRL**

Bits	Name	Description
5:2	BHS_REST_STRENGTH	4 bits for enable rest strength selection
1:0	BHS_FEW_STRENGTH	2 bits for enable few strength selection

**0x05068000 GPUCC\_HYP\_VOTE\_GPU\_SMMU\_CLK****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

Registers for SMMU\_CLK

**GPUCC\_HYP\_VOTE\_GPU\_SMMU\_CLK**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off Off 0 : Clock is ON 1 : Clock is OFF

**GPUCC\_HYP\_VOTE\_GPU\_SMMU\_CLK (cont.)**

Bits	Name	Type	Description
0	CLK_ENABLE		Software Clock Enable Bit 0 : Clock Disabled 1 : Clock Enabled

**0x05068004 GPUCC\_HYP\_VOTE\_GPU\_SMMU\_GDS****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

Registers for SMMU\_GDS

**GPUCC\_HYP\_VOTE\_GPU\_SMMU\_GDS**

Bits	Name	Type	Description
31	PWR_ON	R	PWR State 0 : Power collapsed 1 : Power on
0	SW_COLLAPSE		Software Collapse 0 : Power on through software 1 : Power off through software, default

**0x05069000 GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_CLK****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

Registers for SMMU\_CLK

**GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_CLK**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off Off 0 : Clock is ON 1 : Clock is OFF
0	CLK_ENABLE		Software Clock Enable Bit 0 : Clock Disabled 1 : Clock Enabled

**0x05069004 GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_GDS****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

Registers for SMMU\_GDS

**GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_GDS**

Bits	Name	Type	Description
31	PWR_ON	R	PWR State 0 : Power collapsed 1 : Power on
0	SW_COLLAPSE		Software Collapse 0 : Power on through software 1 : Power off through software, default

**0x0506A000 GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_CLK****Type:** RW**Clock:** AHB CLK**Reset State:** 0x80000000

Registers for SMMU\_CLK

**GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_CLK**

Bits	Name	Type	Description
31	CLK_OFF	R	Branch Clock Off Off 0 : Clock is ON 1 : Clock is OFF
0	CLK_ENABLE		Software Clock Enable Bit 0 : Clock Disabled 1 : Clock Enabled

**0x0506A004 GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_GDS****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001

Registers for SMMU\_GDS

**GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_GDS**

Bits	Name	Type	Description
31	PWR_ON	R	PWR State 0 : Power collapsed 1 : Power on
0	SW_COLLAPSE		Software Collapse 0 : Power on through software 1 : Power off through software, default

**0x0506D000 GPUCC\_GDS\_HW\_CTRL\_IRQ\_STATUS****Type:** R**Clock:** AHB CLK**Reset State:** 0x00000000**GDS\_HW\_CTRL\_IRQ\_STATUS****GPUCC\_GDS\_HW\_CTRL\_IRQ\_STATUS**

Bits	Name	Description
0	GPU	GPU

**0x0506D004 GPUCC\_GDS\_HW\_CTRL\_IRQ\_MASK****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000001**GDS\_HW\_CTRL\_IRQ\_MASK****GPUCC\_GDS\_HW\_CTRL\_IRQ\_MASK**

Bits	Name	Description
0	GPU	GPU

**0x0506D008 GPUCC\_GDS\_HW\_CTRL\_IRQ\_CLEAR****Type:** RW**Clock:** AHB CLK**Reset State:** 0x00000000**GDS\_HW\_CTRL\_IRQ\_CLEAR**

**GPUCC\_GDS\_HW\_CTRL\_IRQ\_CLEAR**

Bits	Name	Description
0	GPU	Writing 1 to this register will clear the Interrupt

Qualcomm  
2019-09-18 18:26:30 PDT  
xiaoyongjie15@huqin.com

# 5 Mobile display subsystem registers

---

## 0x0C900010 MMSS\_MDSS\_HW\_INTR\_STATUS

Type: R

Reset State: 0x00000000

MDSS Toplevel Interrupt Collection Status Register

Since there are a number of display IP related interrupt sources, this register indicates which display IP core has an interrupt line asserted to help out the display driver

### MMSS\_MDSS\_HW\_INTR\_STATUS

Bits	Name	Description
12	EDP	0x0: EDP Interrupt line is NOT asserted 0x1: EDP Interrupt line is asserted
8	HDMI	0x0: HDMI Interrupt line is NOT asserted 0x1: HDMI Interrupt line is asserted
5	DSI1	0x0: DSI1 Interrupt line is NOT asserted 0x1: DSI1 Interrupt line is asserted
4	DSI0	0x0: DSI0 Interrupt line is NOT asserted 0x1: DSI0 Interrupt line is asserted
2	ROT	0x0: ROT Interrupt line is NOT asserted 0x1: ROT Interrupt line is asserted
0	MDP	0x0: MDP Interrupt line is NOT asserted 0x1: MDP Interrupt line is asserted

## 0x0C900014 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_0

Type: RW

Reset State: 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_0**

Bits	Name	Description
31:0	MDSS_SCRATCH_0	32 bit scratch register

**0x0C900018 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_1****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_1**

Bits	Name	Description
31:0	MDSS_SCRATCH_1	32 bit scratch register

**0x0C90001C MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_2****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_2**

Bits	Name	Description
31:0	MDSS_SCRATCH_2	32 bit scratch register

**0x0C900020 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_3****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_3**

Bits	Name	Description
31:0	MDSS_SCRATCH_3	32 bit scratch register

**0x0C900024 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_4****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_4**

Bits	Name	Description
31:0	MDSS_SCRATCH_4	32 bit scratch register

**0x0C900028 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_5****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_5**

Bits	Name	Description
31:0	MDSS_SCRATCH_5	32 bit scratch register

**0x0C90002C MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_6****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_6**

Bits	Name	Description
31:0	MDSS_SCRATCH_6	32 bit scratch register

**0x0C900030 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_7****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_7**

Bits	Name	Description
31:0	MDSS_SCRATCH_7	32 bit scratch register

**0x0C900034 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_8****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_8**

Bits	Name	Description
31:0	MDSS_SCRATCH_8	32 bit scratch register

**0x0C900038 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_9****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_9**

Bits	Name	Description
31:0	MDSS_SCRATCH_9	32 bit scratch register

**0x0C90003C MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_10****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_10**

Bits	Name	Description
31:0	MDSS_SCRATCH_10	32 bit scratch register

**0x0C900040 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_11****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_11**

Bits	Name	Description
31:0	MDSS_SCRATCH_11	32 bit scratch register

**0x0C900044 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_12****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_12**

Bits	Name	Description
31:0	MDSS_SCRATCH_12	32 bit scratch register

**0x0C900048 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_13****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_13**

Bits	Name	Description
31:0	MDSS_SCRATCH_13	32 bit scratch register

**0x0C90004C MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_14****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_14**

Bits	Name	Description
31:0	MDSS_SCRATCH_14	32 bit scratch register

**0x0C900050 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_15****Type:** RW**Reset State:** 0x00000000

MDSS scratch register

**MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_15**

Bits	Name	Description
31:0	MDSS_SCRATCH_15	32 bit scratch register

**0x0C900060 MMSS\_MDSS\_HW\_MDSS\_VBIF\_UBWC\_CLK\_CTRL****Type:** RW**Reset State:** 0x00000000

uBWC Encoder hardware Version Register standardized according to structure

**MMSS\_MDSS\_HW\_MDSS\_VBIF\_UBWC\_CLK\_CTRL**

Bits	Name	Description
0	UBWC_DISABLE	Global disable for UBWC-D cores within VBIF Used to clock gate Qualcomm® Snapdragon™ Universal Bandwidth Compression (UBWC) core when not required for any traffic Signal can only be programmed when VBIF is IDLE and no dynamic switching is allowed Would be used for use cases that guarantee no UBWC enablement to save some power by gating off internal logic

**0x0C900080 MMSS\_MDSS\_HW\_MDSS\_CLK\_CTRL****Type:** RW**Reset State:** 0x00000000

MDSS clock gating control

**MMSS\_MDSS\_HW\_MDSS\_CLK\_CTRL**

Bits	Name	Description
8	ROT_CLK_STOP	write 1 to stop ROT clock, a Rotator register write or read access will turn on Qualcomm® Snapdragon™ Display Engine clock
0	SDE_CLK_STOP	write 1 to stop Snapdragon Display Engine/MDP clock, a Snapdragon Display Engine or DSI register write or read write will turn on Snapdragon Display Engine clock

**0x0C900084 MMSS\_MDSS\_HW\_MDSS\_CLK\_STATUS****Type:** R**Reset State:** 0x00000000

MDSS clock gating status

**MMSS\_MDSS\_HW\_MDSS\_CLK\_STATUS**

Bits	Name	Description
8	ROT_CLK_STATUS	1 indicates Rotator clock is active
0	SDE_CLK_STATUS	1 indicates Snapdragon Display Engine clock is active

**0x0C900090 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_0****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_0**

Bits	Name	Description
0	MDSS_SEMAPHORE_0	<p>1 bit semaphore register</p> <p>Read Operation:</p> <p>If current status is 0, the read operation will change the status to 1 and return 0</p> <p>If current status is 1, no update action is taken and a status of 1 is returned</p> <p>Write Operation:</p> <p>Writing 1 takes no actions</p> <p>Writing 0, changes the status to 0 if already 0, then no action is taken</p>

**0x0C900094 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_1****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_1**

Bits	Name	Description
0	MDSS_SEMAPHORE_1	1 bit semaphore register

**0x0C900098 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_2****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_2**

Bits	Name	Description
0	MDSS_SEMAPHORE_2	1 bit semaphore register

**0x0C90009C MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_3****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_3**

Bits	Name	Description
0	MDSS_SEMAPHORE_3	1 bit semaphore register

**0x0C9000A0 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_4****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_4**

Bits	Name	Description
0	MDSS_SEMAPHORE_4	1 bit semaphore register

**0x0C9000A4 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_5****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_5**

Bits	Name	Description
0	MDSS_SEMAPHORE_5	1 bit semaphore register

**0x0C9000A8 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_6****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_6**

Bits	Name	Description
0	MDSS_SEMAPHORE_6	1 bit semaphore register

**0x0C9000AC MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_7****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_7**

Bits	Name	Description
0	MDSS_SEMAPHORE_7	1 bit semaphore register

**0x0C9000B0 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_8****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_8**

Bits	Name	Description
0	MDSS_SEMAPHORE_8	1 bit semaphore register

**0x0C9000B4 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_9****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_9**

Bits	Name	Description
0	MDSS_SEMAPHORE_9	1 bit semaphore register

**0x0C9000B8 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_10****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_10**

Bits	Name	Description
0	MDSS_SEMAPHORE_10	1 bit semaphore register

**0x0C9000BC MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_11****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_11**

Bits	Name	Description
0	MDSS_SEMAPHORE_11	1 bit semaphore register

**0x0C9000C0 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_12****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_12**

Bits	Name	Description
0	MDSS_SEMAPHORE_12	1 bit semaphore register

**0x0C9000C4 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_13****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_13**

Bits	Name	Description
0	MDSS_SEMAPHORE_13	1 bit semaphore register

**0x0C9000C8 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_14****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_14**

Bits	Name	Description
0	MDSS_SEMAPHORE_14	1 bit semaphore register

**0x0C9000CC MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_15****Type:** RW**Reset State:** 0x00000000

MDSS hardware semaphore register

**MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_15**

Bits	Name	Description
0	MDSS_SEMAPHORE_15	1 bit semaphore register

**0x0C990010 MMSS\_DP\_SW\_RESET****Type:** RW**Reset State:** 0x00000000

Software Reset for logic as noted in descriptions. Set bit to 1 for reset, set back to 0 for normal operation

double buffered: NO

**MMSS\_DP\_SW\_RESET**

Bits	Name	Description
3	SDP_SW_RESET	Sync reset for SDP generation logic, excluding MSA/VSC/PPS.
2	AUD_SW_RESET	Sync reset for audio logic.
1	HDCP_SW_RESET	Sync reset for HDCP 1.3/2.2 logic. It is to be used in unauthenticated state. For HDCP2.2, applying this reset on-the-fly during hdcp2.2 authentication state will cause immediate blocking of audio and video data, encryption disabled until the end of frame and it may cause the sink to assert RxStatus.reauth_req.
0	DP_SW_RESET	Sync reset for entire controller. Manually set the register values to default if necessary to replicate hard reset of the register clock domain. (all flops except for registers and TPG are reset)

**0x0C990014 MMSS\_DP\_PHY\_CTRL****Type:** RW**Reset State:** 0x00000000

DP PHY control register

**MMSS\_DP\_PHY\_CTRL**

Bits	Name	Description
2	SW_RESET	Reset value sent to DP PHY

**0x0C990018 MMSS\_DP\_CLK\_CTRL****Type:** RW**Reset State:** 0x00000000

Clock gating control for mainlink

double buffered: NO

**MMSS\_DP\_CLK\_CTRL**

Bits	Name	Description
3	LINK_CLK_FORCE_ON	Link clock gater control. Set to 1 to force clocks on. Set to 0 for hardware clock gating
2	CRYPTO_CLK_FORCE_ON	Crypto clock gater control. Set to 1 to force clocks on. Set to 0 for hardware clock gating
1	AUX_CLK_FORCE_ON	AUX/SCLK clock gater control. Set to 1 to force clocks on. Set to 0 for hardware clock gating
0	PIXEL_CLK_FORCE_ON	Pixel clock gater control. Set to 1 to force clocks on. Set to 0 for hardware clock gating

**0x0C99001C MMSS\_DP\_CLK\_ACTIVE****Type:** R**Reset State:** 0x00000000

Clock active indicator

double buffered: NO

**MMSS\_DP\_CLK\_ACTIVE**

Bits	Name	Description
14	CRYPTO_CLK_ACTIVE	Indicator that the crypto clock is active
13	HDCP1P3_SCLK_CLK_ACTIVE	Indicator that the HDCP1.3 branch of SCLK is active
12	HDCP_SCLK_CLK_ACTIVE	Indicator that the HDCP general control SCLK is active
11	AUX_CLK_ACTIVE	Indicator that the AUX clock is active

**MMSS\_DP\_CLK\_ACTIVE (cont.)**

Bits	Name	Description
10	TPG_PIXEL_CLK_ACTIVE	Indicator that the TPG pixel clock is active
9	PIXEL_CLK_ACTIVE	Indicator that the pixel clock is active
8	PIXEL0_CLK_ACTIVE	Indicator that the pixel lane 0 clock is active
7	PIXEL1_CLK_ACTIVE	Indicator that the pixel lane 1 clock is active
6	PIXEL23_CLK_ACTIVE	Indicator that the pixel lane 2/3 clock is active
3	HDCP2P2_LINK_CLK_ACTIVE	Indicator that the HDCP2.2 link clock is active
2	HDCP1P3_LINK_CLK_ACTIVE	Indicator that the HDCP1.3 link clock is active
1	IF_LINK_CLK_ACTIVE	Indicator that the link interface clock is active
0	MAIN_LINK_CLK_ACTIVE	Indicator that the main clock is active

**0x0C990020 MMSS\_DP\_INTERRUPT\_STATUS****Type:** RW**Reset State:** 0x00000000

DP AUX/PLL related interrupt status register

**MMSS\_DP\_INTERRUPT\_STATUS**

Bits	Name	Type	Description
29	DPPHY_AUX_ERROR_MASK		Mask bit for DPPHY_AUX_ERROR_INT. Set to 1 to enable interrupt
28	DPPHY_AUX_ERROR_ACK	W	Acknowledge bit. Write 1 to clear DPPHY_AUX_ERROR_INT interrupt
27	DPPHY_AUX_ERROR_INT	R	DPPHY AUX detected an error and issued an interrupt. Clear the controller interrupt first, read status bits of the PHY, clear the PHY interrupt, and then resume/restart AUX/I2C transactions
26	PLL_UNLOCK_DET_MASK		Mask bit for PLL_UNLOCK_DET_INT. Set to 1 to enable interrupt
25	PLL_UNLOCK_DET_ACK	W	Acknowledge bit. Write 1 to clear PLL_UNLOCK_DET_INT interrupt
24	PLL_UNLOCK_DET_INT	R	PLL in DP PHY has unlocked
20	AUX_NACK_DURING_I2C_MASK		Mask bit for AUX_NACK_DURING_I2C_INT. Set to 1 to enable interrupt
19	AUX_NACK_DURING_I2C_ACK	W	Acknowledge bit. Write 1 to clear AUX_NACK_DURING_I2C_INT interrupt

**MMSS\_DP\_INTERRUPT\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
18	AUX_NACK_DURING_I2C_INT	R	AUX NACK'ed during an I2C transaction
17	WRONG_READDATA_COUNT_MASK		Mask bit for WRONG_ReadData_COUNT_INT. Set to 1 to enable interrupt
16	WRONG_READDATA_COUNT_ACK	W	Acknowledge bit. Write 1 to clear WRONG_ReadData_COUNT_INT interrupt
15	WRONG_READDATA_COUNT_INT	R	The number of read data bytes does not match the requested number of bytes
14	CONSECUTIVE_NACK_DEFER_MASK		Mask bit for CONSECUTIVE_NACK_DEFER_INT. Set to 1 to enable interrupt
13	CONSECUTIVE_NACK_DEFER_ACK	W	Acknowledge bit. Write 1 to clear CONSECUTIVE_NACK_DEFER_INT interrupt
12	CONSECUTIVE_NACK_DEFER_INT	R	AUX/I2C has NACK'ed/DEFER'ed beyond the number of allowed attempts set by AUX_limits
11	CONSECUTIVE_TIMEOUT_MASK		Mask bit for CONSECUTIVE_TIMEOUT_INT. Set to 1 to enable interrupt
10	CONSECUTIVE_TIMEOUT_ACK	W	Acknowledge bit. Write 1 to clear CONSECUTIVE_TIMEOUT_INT interrupt
9	CONSECUTIVE_TIMEOUT_INT	R	AUX/I2C transaction have timed out consecutively and the number of attempts exceeded the allowed number of attempts defined in AUX_limits
8	WRONG_ADDRESS_MASK		Mask bit for WRONG_ADDRESS_INT. Set to 1 to enable interrupt
7	WRONG_ADDRESS_ACK	W	Acknowledge bit. Write 1 to clear WRONG_ADDRESS_INT interrupt
6	WRONG_ADDRESS_INT	R	A wrong address in an AUX/I2C transaction was attempted
5	AUX_I2C_DONE_MASK		Mask bit for AUX_I2C_done_INT. Set to 1 to enable interrupt
4	AUX_I2C_DONE_ACK	W	Acknowledge bit. Write 1 to clear AUX_I2C_done_INT interrupt
3	AUX_I2C_DONE_INT	R	AUX/I2C transaction(s) are completed successfully

**0x0C990024 MMSS\_DP\_INTERRUPT\_STATUS\_2****Type:** RW**Reset State:** 0x00000000

DP MAINLINK related interrupt status register

**MMSS\_DP\_INTERRUPT\_STATUS\_2**

Bits	Name	Type	Description
27	ENCRYPTION_NOT_READ_Y_MASK		Mask bit for ENCRYPTION_NOT_READY interrupt set to 1 to enable interrupt
26	ENCRYPTION_NOT_READ_Y_ACK	W	Acknowledge bit for ENCRYPTION_NOT_READY - write 1 to clear
25	ENCRYPTION_NOT_READ_Y	R	HDCP_STATUS.AUTH_STATUS has changed from 1 or 2 (authenticated) to 0 (non-authenticated)
24	ENCRYPTION_READY_MASK		Mask bit for ENCRYPTION_READY interrupt - set to 1 to enable interrupt
23	ENCRYPTION_READY_ACK	W	Acknowledge bit for ENCRYPTION_READY - write 1 to clear
22	ENCRYPTION_READY	R	HDCP_STATUS.AUTH_STATUS has changed from 0 (non-authenticated) to 1 or 2 (authenticated)
21	HDCP_AUTH_FAIL_INFO_ACK	W	Acknowledge bit for HDCP Authentication Failure Info field - write 1 to clear
20	HDCP_AUTH_FAIL_MASK		Mask bit for HDCP Authentication Lost interrupt set to 1 to enable interrupt
19	HDCP_AUTH_FAIL_ACK	W	Acknowledge bit for HDCP Authentication Lost bit - write 1 to clear
18	HDCP_AUTH_FAIL_INT	R	HDCP Authentication Lost interrupt Status
17	HDCP_AUTH_SUCCESS_MASK		Mask bit for HDCP Authentication Success interrupt - set to 1 to enable interrupt
16	HDCP_AUTH_SUCCESS_ACK	W	Acknowledge bit for HDCP Authentication Success bit - write 1 to clear
15	HDCP_AUTH_SUCCESS_INT	R	HDCP Authentication Success interrupt status
11	CRC_UPDATED_MASK		Mask bit for CRC_UPDATED_INT. Set to 1 to enable interrupt
10	CRC_UPDATED_ACK	W	Acknowledge bit. Write 1 to clear CRC_UPDATED_INT interrupt
9	CRC_UPDATED_INT	R	CRC (MISR_LANE0, MISR_LANE1, MISR_LANE2, MISR_LANE3) and FILLER_COUNT have been updated in DP MISR
8	FRAME_END_MASK		Mask bit for FRAME_END_INT. Set to 1 to enable interrupt
7	FRAME_END_ACK	W	Acknowledge bit. Write 1 to clear FRAME_END_INT interrupt
6	FRAME_END_INT	R	The end of a frame at the very last active pixel symbol
5	IDLE_PATTERNS_SENT_MASK		Mask bit for IDLE_PATTERNS_SENT_INT. Set to 1 to enable interrupt

**MMSS\_DP\_INTERRUPT\_STATUS\_2 (cont.)**

Bits	Name	Type	Description
4	IDLE_PATTERNS_SENT_ACK	W	Acknowledge bit. Write 1 to clear IDLE_PATTERNS_SENT_INT interrupt
3	IDLE_PATTERNS_SENT_IN_T	R	Required number of idle patterns are sent after a software push for idle
2	READY_FOR_VIDEO_MASK		Mask bit for READY_FOR_VIDEO_INT. Set to 1 to enable interrupt
1	READY_FOR_VIDEO_ACK	W	Acknowledge bit. Write 1 to clear READY_FOR_VIDEO_INT interrupt
0	READY_FOR_VIDEO_INT	R	DP hardware is ready to receive video after a software command to start video

**0x0C990028 MMSS\_DP\_INTERRUPT\_STATUS\_3****Type:** RW**Reset State:** 0x00000000

DP MAINLINK related interrupt status register

**MMSS\_DP\_INTERRUPT\_STATUS\_3**

Bits	Name	Type	Description
11	AUD_SAM_DROP_MASK		Audio sample drop interrupt mask. When 1 written to this bit, it AUD_SAM_DROP_INT interrupt will toggle the interrupt line
10	AUD_SAM_DROP_ACK	W	Audio sample drop interrupt acknowledge. When 1 written to this bit, it will generate a pulse to clear the AUD_SAM_DROP_INT
9	AUD_SAM_DROP_INT	R	Audio sample drop interrupt. This goes hi when the audio data tag commiing in from lpass does not match the count in audio engine
8	AUD_FIFO_URUN_MASK		Audio fifo underrun interrupt mask. When 1 written to this bit, it AUD_FIFO_URUN_INT interrupt will toggle the interrupt line
7	AUD_FIFO_URUN_ACK	W	Audio fifo underrun interrupt acknowledge. When 1 written to this bit, it will generate a pulse to clear the AUD_FIFO_URUN_INT
6	AUD_FIFO_URUN_INT	R	Audio fifo underrun interrupt. This goes hi when the audio fifo requests the data put there is no request and all the fifo entries are drained out
2	HDCP2P2_KEY_FIFO_UFLOW_MASK		Mask bit fo SECURE_HDCP2P2_KEY_FIFO_UFLOW interrupt set to 1 to enable interrupt (non-secure)

**MMSS\_DP\_INTERRUPT\_STATUS\_3 (cont.)**

Bits	Name	Type	Description
1	HDCP2P2_KEY_FIFO_UFL_OW_ACK	W	Acknowledge bit for SECURE_HDCP2P2_KEY_FIFO_UFLOW - write 1 to clear (non-secure)
0	HDCP2P2_KEY_FIFO_UFL_OW	R	Indicate that the AES key fifo has underflowed (non-secure)

**0x0C990200 MMSS\_DP\_DP\_HPD\_CTRL****Type:** RW**Reset State:** 0x00000000

HPD related control bits

**MMSS\_DP\_DP\_HPD\_CTRL**

Bits	Name	Description
31	HPD_PIN_POLARITY	HPD polarity, 0=HPD is active high, so going low is an interrupt, 1=HPD is active low, so going high is an interrupt
0	DP_HPD_EN	ALPHA: ENABLE DP HPD enable, 0=disable, 1=enable

**0x0C990204 MMSS\_DP\_DP\_HPD\_INT\_STATUS****Type:** R**Reset State:** 0x00000000

DP HPD interrupt status

**MMSS\_DP\_DP\_HPD\_INT\_STATUS**

Bits	Name	Description
31:29	DP_HPD_STATE_STATUS	Indicating the current event state: 000: Disconnected 001: Connect Pending 010: Connected 011: HPD I/O Glitch Count 100: IRQ HPD Pulse Count 101: HPD Replug Count
3	DP_HPD_UNPLUG_INT_STATUS	This event is meant to signal the downstream to upstream removal of a connected device, in order for the upstream device to cease communication downstream

**MMSS\_DP\_DP\_HPD\_INT\_STATUS (cont.)**

Bits	Name	Description
2	DP_HPD_REPLUG_INT_STATUS	This event is a Sink to Source 'Replug' notification generated to inform the Source that a broader change has occurred in the Sink such that the Source needs to start at the initial connect capability check of the Sink
1	DP_IRQ_HPD_INT_STATUS	This event is a Sink to Source interrupt request while the devices are deemed to be connected
0	DP_HPD_PLUG_INT_STATUS	This event is meant to signal the downstream to upstream presence of a connected device

**0x0C990208 MMSS\_DP\_DP\_HPD\_INT\_ACK****Type:** W**Reset State:** 0x00000000

DP HPD Interrupt Acknowledgement

**MMSS\_DP\_DP\_HPD\_INT\_ACK**

Bits	Name	Description
3	DP_HPD_UNPLUG_INT_ACK	Acknowledgement bit. Write 1 to clear DP_HPD_UNPLUG_INT_ACK
2	DP_HPD_REPLUG_INT_ACK	Acknowledgement bit. Write 1 to clear DP_HPD_REPLUG_INT_ACK
1	DP_IRQ_HPD_INT_ACK	Acknowledgement bit. Write 1 to clear DP_IRQ_HPD_INT
0	DP_HPD_PLUG_INT_ACK	Acknowledgement bit. Write 1 to clear DP_HPD_PLUG_INT

**0x0C99020C MMSS\_DP\_DP\_HPD\_INT\_MASK****Type:** RW**Reset State:** 0x00000000

DP HPD Interrupt Mask

**MMSS\_DP\_DP\_HPD\_INT\_MASK**

Bits	Name	Description
3	DP_HPD_UNPLUG_INT_MASK	Mask bit for DP_HPD_UNPLUG_INT. Set 1 to enable interrupt
2	DP_HPD_REPLUG_INT_MASK	Mask bit for DP_HPD_REPLUG_INT. Set 1 to enable interrupt
1	DP_IRQ_HPD_INT_MASK	Mask bit for DP_IRQ_HPD_INT. Set 1 to enable interrupt

**MMSS\_DP\_DP\_HPD\_INT\_MASK (cont.)**

Bits	Name	Description
0	DP_HPD_PLUG_INT_MASK	Mask bit for DP_HPD_PLUG_INT. Set 1 to enable interrupt

**0x0C990218 MMSS\_DP\_DP\_HPD\_REF\_TIMER****Type:** RW**Reset State:** 0x000000013

Microsecond Reference Timer

**MMSS\_DP\_DP\_HPD\_REF\_TIMER**

Bits	Name	Description
16	REF_TIMER_ENABLE	ALPHA: ENABLE Value to set the register in order to generate a strobe every microsecond.
15:0	REF_TIMER	Enable the reference timer

**0x0C99021C MMSS\_DP\_DP\_HPD\_EVENT\_TIME\_0****Type:** RW**Reset State:** 0x03E986A0

HPD Timer Control 0

**MMSS\_DP\_DP\_HPD\_EVENT\_TIME\_0**

Bits	Name	Description
31:18	DP_HPD_GLITCH_TIME	Amount of time in microseconds HPD line must be asserted for a glitch
17:0	DP_HPD_CONNECT_TIME	Amount of time in microseconds HPD line must be asserted for a connect

**0x0C990220 MMSS\_DP\_DP\_HPD\_EVENT\_TIME\_1****Type:** RW**Reset State:** 0x61A807D0

HPD Timer Control 1

**MMSS\_DP\_DPDHPD\_EVENT\_TIME\_1**

Bits	Name	Description
31:14	DP_HPD_DISCONNECT_TIME	Amount of time in microseconds HPD line must be asserted for a disconnected
13:0	DP_IRQ_HPD_MAX_TIME	Amount of time in microseconds HPD line must be asserted for a HPD IRQ

**0x0C990230 MMSS\_DP\_AUX\_CTRL****Type:** RW**Reset State:** 0x000000010

AUX control register

**MMSS\_DP\_AUX\_CTRL**

Bits	Name	Description
4	AUX_DPPHY_ERROR_STOP_TX	when set, upon an DPPHY_AUX_ERROR the controller stops, goes to idle mode, and waits for software interaction. 0x0: DISABLE 0x1: ENABLE
3	AUX_RX_ENDIANNESS	0x0: BIG 0x1: LITTLE
2	AUX_TX_ENDIANNESS	0x0: BIG 0x1: LITTLE
1	SW_AUX_RESET	Software reset to AUX controller
0	AUX_ENABLE	when set, enables the AUX controller and its clocks

**0x0C990234 MMSS\_DP\_AUX\_DATA****Type:** RW**Reset State:** 0x000000000

This register is used to read or write the AUX buffer

**MMSS\_DP\_AUX\_DATA**

Bits	Name	Type	Description
31	INDEX_WRITE	W	To write index field, set this bit to 1 while writing AUX_DATA.

**MMSS\_DP\_AUX\_DATA (cont.)**

Bits	Name	Type	Description
23:16	INDEX		Use to set index into AUX buffer for next read or current write, or to read index of current read or next write. Writable only when INDEX_WRITE=1.
15:8	DATA		Use to fill or read the AUX buffer
0	DATA_RW		Select whether buffer access will be a read or write. For writes, address auto-increments on write to AUX_DATA. For reads, address auto-increments on reads to AUX_DATA. 0x0: WRITE 0x1: READ

**0x0C990238 MMSS\_DP\_AUX\_TRANS\_CTRL****Type:** RW**Reset State:** 0x00000000

DP AUX transaction control register

**MMSS\_DP\_AUX\_TRANS\_CTRL**

Bits	Name	Description
11	NO_SEND_STOP	Do not send final address only transaction and thus no asserted MOT at end of I2C transaction (single I2C transaction mode only)
10	NO_SEND_ADDR	Do not send initial address only transaction (single I2C transaction mode only)
9	GO	GO bit from the software. When set, software HAS TO KEEP it set until an AUX/I2C interrupt occurs. Then software must first reset GO, write new transactions into cmd_fifo and then set the GO again, there should be enough time between resetting GO and setting it again for the hardware to detect a new GO
8	TR_TYPE	defines transaction type between AUX and I2C 0x0: AUX 0x1: I2C
7:0	NUM_TR	It should be programmed to the number of transactions - 1, number of transactions written into cmd_fifo. Software should take into account the size of the FIFO (144 bytes) and the fact that each command is 4 bytes instructions and LEN data bytes

**0x0C99023C MMSS\_DP\_TIMEOUT\_COUNT****Type:** RW**Reset State:** 0x0000021C0

DP timeout count

#### **MMSS\_DP\_TIMEOUT\_COUNT**

Bits	Name	Description
19:0	TIMEOUT_COUNT	Defines how many AUX_DP_CLKs should happen before one timeout is perceived by AUX cont. hardware

### **0x0C990240 MMSS\_DP\_AUX\_LIMITS**

**Type:** RW

**Reset State:** 0x0000A222

DP limits for AUX/I2C

#### **MMSS\_DP\_AUX\_LIMITS**

Bits	Name	Description
31:28	I2C_DEFER_LIMIT_MSB	During I2C operation, how many AUX/I2C defers are acceptable before an interrupt (MSB extension)
19:16	TIMEOUTS_LIMIT_MSB	defines how many consecutive timeouts before an interrupt (MSB extension)
15:12	I2C_DEFER_LIMIT	During I2C operation, how many AUX/I2C defers are acceptable before an interrupt
11:8	AUX_NACK_DEFER_LIMIT	defines how many AUX nack's/defer's are acceptable before an interrupt
7:4	AUX_READ_ZERO_LIMIT	defines how many AUX reads with no data are acceptable before an interrupt
3:0	TIMEOUTS_LIMIT	defines how many consecutive timeouts before an interrupt

### **0x0C990244 MMSS\_DP\_AUX\_STATUS**

**Type:** R

**Reset State:** 0x00000000

DP AUX STATUS

#### **MMSS\_DP\_AUX\_STATUS**

Bits	Name	Description
5	AUX_NACK_DURING_I2C	AUX NACK'ed during an I2C transaction
4	WRONG_READDATA_COUNT	The number of read data bytes does not match the requested number of bytes

**MMSS\_DP\_AUX\_STATUS (cont.)**

Bits	Name	Description
3	CONSECUTIVE_NACK_DEFER	AUX/I2C has NACK'ed/DEFER'ed beyond the number of allowed attempts set by AUX_limits
2	CONSECUTIVE_TIMEOUT	AUX/I2C transaction have timed out consecutively and the number of attempts exceeded the allowed number of attempts defined in AUX_limits
1	WRONG_ADDRESS	A wrong address in an AUX/I2C transaction was attempted
0	AUX_I2C_DONE	AUX/I2C transaction(s) are completed successfully

**0x0C990248 MMSS\_DP\_AUX\_MISR\_CTRL****Type:** RW**Reset State:** 0x00000000

AUX MISR CONTROL

**MMSS\_DP\_AUX\_MISR\_CTRL**

Bits	Name	Description
0	AUX_MISR_EN	Set to 1 to Enable

**0x0C99024C MMSS\_DP\_AUX\_MISR\_TX\_VALUE****Type:** R**Reset State:** 0xFFFFFFFF

AUX TX MISR VALUE

**MMSS\_DP\_AUX\_MISR\_TX\_VALUE**

Bits	Name	Description
31:0	MISR_AUX_TX	CRC32 value MISR for AUX Transmitted bytes over a batch of transactions between the rising edge of GO and the falling edge of GO all controlled by software

**0x0C990250 MMSS\_DP\_AUX\_MISR\_RX\_VALUE****Type:** R**Reset State:** 0xFFFFFFFF

AUX RX MISR VALUE

**MMSS\_DP\_AUX\_MISR\_RX\_VALUE**

Bits	Name	Description
31:0	MISR_AUX_RX	CRC32 value MISR for AUX Received bytes over a batch of transactions between the rising edge of GO and the falling edge of GO all controlled by software

**0x0C990400 MMSS\_DP\_MAINLINK\_CTRL****Type:** RW**Reset State:** 0x00000000

DP MAIN LINK CONTROL

**MMSS\_DP\_MAINLINK\_CTRL**

Bits	Name	Description
6	SW_FORCE_FIFO_PIXEL_CLOCK_ON	when set, if DP_MAINLINK_ENABLE is also set, the pixel clock to the FIFOs of all data lanes is forced on, THE Default VALUE is suitable for normal operation, setting this bit causes wasting extra power
5	SW_BYPASS_8B10B	when set, bypass 8b10b, pads the 2 msb bits with zero
4	SW_BYPASS_SCRAMBLER	when set, bypass scrambler
3	SW_BYPASS_INTERLANE_SKEW	when set, bypass lane skew insertion
2	RESERVED	RESERVED
1	SW_MAINLINK_RESET	when set, software resets the main link
0	DP_MAINLINK_ENABLE	This is set by software whenever DP is needed, it turns on clocks and it takes out the main link FSMs out of reset state to a ready-for-cmd state

**0x0C990404 MMSS\_DP\_STATE\_CTRL****Type:** RW**Reset State:** 0x00000000

DP software control of the main link, i.e. main link FSM

**MMSS\_DP\_STATE\_CTRL**

Bits	Name	Description
8	SW_PUSH_IDLE	when set, software pushes for idle state, this should not be used in normal operation

**MMSS\_DP\_STATE\_CTRL (cont.)**

Bits	Name	Description
7	SW_SEND_VIDEO	when set, software commands sending video
6		
5	SW_LINK_PRBS7	when set, software commands sending PRBS7 pattern
4	SW_LINK_SYMBOL_ERROR_RATE_MEASUREMENT	when set, software commands sending symbol error rate measurement pattern
3	SW_LINK_TRAINING_PATTERN4	
2	SW_LINK_TRAINING_PATTERN3	when set, software commands sending link training pattern3
1	SW_LINK_TRAINING_PATTERN2	when set, software commands sending link training pattern2
0	SW_LINK_TRAINING_PATTERN1	when set, software commands sending link training pattern1

**0x0C990408 MMSS\_DP\_CONFIGURATION\_CTRL****Type:** RW**Reset State:** 0x00004577

DP software configuration of the main ink

**MMSS\_DP\_CONFIGURATION\_CTRL**

Bits	Name	Description
14:13	LSCLK_DIV	Link Symbol Clk div 00: Link symbol clk frequency = Link rate clk frequency 01: Link symbol clk frequency = 1/2 Link rate clk frequency 10: Link symbol clk frequency = 1/4 Link rate clk frequency
12:11	RGB_YUV	Format 00: RGB 01: YUV420 10: YUV422 11: reserved
10	ASSR	Alternate scrambler seed reset enable for authentication 0x0: DISABLE 0x1: ENABLE
9:8	BPC	Bits Per Component 00: BPC=6 01: BPC=8 10: BPC=10

**MMSS\_DP\_CONFIGURATION\_CTRL (cont.)**

Bits	Name	Description
7	SEND_VSC	Video Stream Config should be sent 0x0: NO_VSC 0x1: SEND_VSC
6	ENHANCED_FRAMING	enhanced DP framing is enabled 0x0: NORMAL 0x1: ENHANCED
5:4	NUMBER_OF_LANES	number of DP main link lanes 0x0: ONE_LANE 0x1: TWO_LANE 0x2: ILLEGAL 0x3: FOUR_LANE
1	STATIC_DYNAMIC_COUNTER	Selecting static or dynamic Mvid: static is software programmed, dynamic is hardware calculated 0x0: DYNAMIC 0x1: STATIC

**0xC990410 MMSS\_DP\_SOFTWARE\_MVID****Type:** RW**Reset State:** 0x00000000

Mvid programmed by software to be used mainly in static pixel clock mode for video clock recovery

**MMSS\_DP\_SOFTWARE\_MVID**

Bits	Name	Description
23:0	SW_MVID	Double buffered

**0xC990418 MMSS\_DP\_SOFTWARE\_NVID****Type:** RW**Reset State:** 0x00000000

Nvid programmed by software to be used mainly in static pixel clock mode for video clock recovery

**MMSS\_DP\_SOFTWARE\_NVID**

Bits	Name	Description
23:0	SW_NVID	Double buffered

**0x0C99041C MMSS\_DP\_TOTAL\_HOR\_VER****Type:** RW**Reset State:** 0x00000000

Total horizontal (in pixels) and total vertical (in lines)

**MMSS\_DP\_TOTAL\_HOR\_VER**

Bits	Name	Description
31:16	TOTAL_VER	total vertical pixels
15:0	TOTAL_HOR	total horizontal pixels double buffered

**0x0C990420 MMSS\_DP\_START\_HOR\_VER\_FROM\_SYNC****Type:** RW**Reset State:** 0x00000000

Horizontal and vertical active start from the leading edges of HSYNC and VSYNC in pixels and line count respectively

**MMSS\_DP\_START\_HOR\_VER\_FROM\_SYNC**

Bits	Name	Description
31:16	START_FROM_VSYNC	
15:0	START_FROM_HSYNC	Double buffered

**0x0C990424 MMSS\_DP\_HSYNC\_VSYNC\_WIDTH\_POLARITY****Type:** RW**Reset State:** 0x00000000

Horizontal and vertical sync pulse width and polarity

**MMSS\_DP\_HSYNC\_VSYNC\_WIDTH\_POLARITY**

Bits	Name	Description
31	VSP	vertical sync polarity
30:16	VSW	vertical sync width
15	HSP	horizontal sync pulse width
14:0	HSW	horizontal sync width

**0x0C990428 MMSS\_DP\_ACTIVE\_HOR\_VER****Type:** RW**Reset State:** 0x00000000

Active video width(hor) and height(ver) in pixels and line count

**MMSS\_DP\_ACTIVE\_HOR\_VER**

Bits	Name	Description
31:16	ACTIVE_VER	
15:0	ACTIVE_HOR	

**0x0C99042C MMSS\_DP\_MISC1\_MISC0****Type:** RW**Reset State:** 0x00000000

MISC0 and MISC1 of main stream attribute, exactly compatible with the DP 1.2 standard pages 74, 75 and 78

**MMSS\_DP\_MISC1\_MISC0**

Bits	Name	Description
15:8	MISC1	b0: interlaced vertical total even -- 0=number of lines per interlaced frame (consisting of two fields) is an odd number , 1=number of lines per interlaced frame (consisting of two fields) is an even number. b2:1: stereo video attribute -- 00=no 3D stereo video in-band signaling done using this field, indicating either no 3D stereo video transported or the in-band signaling done using an SDP called Video_stream_configuration (VSC) packet, 01= For progressive video, the first frame is RIGHT EYE, for interlaced video, TOP field is RIGHT EYE and BOTTOM field is LEFT EYE, 10 =Reserved, 11= For progressive video, the first frame is LEFT EYE, for interlaced video, TOP field is LEFT EYE and BOTTOM field is RIGHT EYE. b6:4: RESERVED (set to 0s). b7: Y-only video
7:0	MISC0	b0: synchronous clock -- 0= Link clock and main video pixel clock are asynchronous, 1=link clock and main video stream clock synchronous, in this case the Mvid value must be constant regardless of link clock down spread enabled. b7:1: Colorimetry Indicator Field and bit 7 of MISC1

**0x0C990430 MMSS\_DP\_VALID\_BOUNDARY****Type:** RW**Reset State:** 0x00000000

valid boundaries that determine the relation between link clock and pixel clock in link and pixel clock domains

#### **MMSS\_DP\_VALID\_BOUNDARY**

Bits	Name	Description
25:16	DELAY_START_LINK	Number of link clock cycles that BE is delayed after a rising edge of DE. double buffered
6:0	VALID_BOUNDARY_LINK	(upper) valid boundary in the link clock domain, result must be between 1~64 double buffered

#### **0x0C990434 MMSS\_DP\_VALID\_BOUNDARY\_2**

Type: RW

Reset State: 0x00000000

parameters defined to be used with the boundary moderation enabled

#### **MMSS\_DP\_VALID\_BOUNDARY\_2**

Bits	Name	Description
23:20	LOWER_BOUNDARY_COUNT	number of cycles that lower valid boundary should be used before switching to the upper boundary, the resulting number is between 1 and 15. double buffered
19:16	UPPER_BOUNDARY_COUNT	number of cycles that upper valid boundary should be used before switching to the lower boundary, the resulting number is between 1 and 15. double buffered
7:1	VALID_LOWER_BOUNDARY_LINK	lower valid boundary in the link clock domain. The resulting number is between 1 and 64. double buffered
0	BOUNDARY_MODERATION_EN	When enabled, moderates the number of valid symbols within a TU so that the number of valid symbols per TU is more evenly distributed  0: BOUNDARY MODERATION DISABLE 1: BOUNDARY MODERATION ENABLE double buffered

**0x0C990438 MMSS\_DP\_LOGICAL2PHYSICAL\_LANE\_MAPPING****Type:** RW**Reset State:** 0x000000E4

Determines how each physical lane from dp cont2PHY gets its input, these are used for lane\_en's as well. It also determines the Endianness

**MMSS\_DP\_LOGICAL2PHYSICAL\_LANE\_MAPPING**

Bits	Name	Description
9	PIXEL_ENDIANNESS	Determines the Endianness for only pixels going from the DP controller out to the PHY and from there to the panel. Writing 1 into this field will cause ONLY the pixels be in little endian 0x0: BIG 0x1: LITTLE
8	MAINLINK_ENDIANNESS	Determines the Endianness for all data going from DP controller out to PHY and from there to the panel. Writing 1 into this field will cause all the data (pixels, blanking, etc.) transmitted on the mainlink to be little endian 0x0: BIG 0x1: LITTLE
7:6	LANE3_MAPPING	physical lane3 from contr. to phy gets its data from logical lane0, lane1, lane2, lane3 0x0: LANE0 0x1: LANE1 0x2: LANE2 0x3: LANE3
5:4	LANE2_MAPPING	physical lane2 from contr. to phy gets its data from logical lane0, lane1, lane2, lane3 0x0: LANE0 0x1: LANE1 0x2: LANE2 0x3: LANE3
3:2	LANE1_MAPPING	physical lane1 from contr. to phy gets its data from logical lane0, lane1, lane2, lane3 0x0: LANE0 0x1: LANE1 0x2: LANE2 0x3: LANE3
1:0	LANE0_MAPPING	physical lane0 from contr. to phy gets its data from logical lane0, lane1, lane2, lane3 0x0: LANE0 0x1: LANE1 0x2: LANE2 0x3: LANE3

**0x0C990444 MMSS\_DP\_MAINLINK\_LEVELS****Type:** RW**Reset State:** 0x00000A14

Determines several programmable levels used with the mainlink

**MMSS\_DP\_MAINLINK\_LEVELS**

Bits	Name	Description
20:8	IDLE_PATTERN_REPEATITION_NUMBER	After software pushes the controller to idle during video, hardware will send the number of idle patterns (each 8192 symbols) determined by this field before raising the idle_patterns_sent, hardware will not send any new video before the programmed number of idle patterns are sent, however software can issue the send video command and then poll for READY_for_VIDEO before enabling MDP
4:0	SAFE_TO_EXIT_LEVEL	This level determines two boundaries: 1) when sending idle patterns, if the sender just sent a BS sequence, a boundary must be reached before a new BS sequence due to start of Video (first frame) or a new MSA due to a new frame can start, the default value should work 2) HW_MVID is not allowed to be updated when the sender has not passed the safe_to_exit boundary to ensure a static hardware MVID is sent to the panel

**0x0C990448 MMSS\_DP\_MAINLINK\_LEVELS\_2****Type:** RW**Reset State:** 0x0000003C

Determines several programmable levels used with the mainlink

**MMSS\_DP\_MAINLINK\_LEVELS\_2**

Bits	Name	Description
11:8	VSC_REPEATITION_NUMBER	If SEND_VSC is set, this field determines in how many frames VSC for 3D will be sent. If it is set to 0, VSC for 3D will be sent for every frame.
7:0	MSA_TO_VSC_DELAY	After sending MSA, how many link cycles to wait before sending VSC if it is enabled by send_VSC register field, default should work. Zero is illegal.

**0x0C99044C MMSS\_DP\_TU****Type:** RW**Reset State:** 0x00000020

Transfer Unit parameters

#### **MMSS\_DP\_TU**

Bits	Name	Description
5:0	TU_SIZE_MINUS1	Must be programmed to the desired TU size minus , it should be between 31~63 double buffered

### **0x0C990450 MMSS\_DP\_DB\_CTRL**

**Type:** RW

**Reset State:** 0x00000000

Double Buffer Control for DP registers

#### **MMSS\_DP\_DB\_CTRL**

Bits	Name	Description
0	DB_ENABLE	When set, enables Double buffering of register fields noted as double buffered in the description. The registers are updated on the update signal from MDP. Registers include VALID_BOUNDARY, VALID_BOUNDARY_2. (excludes new registers added for DP specifically (audio, HDCP) + DSC. When clear, the registers are not double buffered.

### **0x0C990454 MMSS\_DP\_HBR2\_COMPLIANCE\_SCRAMBLER\_RESET**

**Type:** RW

**Reset State:** 0x00002000

HBR2 Compliance Scrambler Reset parameters

#### **MMSS\_DP\_HBR2\_COMPLIANCE\_SCRAMBLER\_RESET**

Bits	Name	Description
16	HBR2_ERM_PATTERN	This bit selects what pattern will be output when SW_LINK_SYMBOL_ERROR_RATE_MEASUREMENT is set  0: Error rate measurement pattern 1: HBR2 compliance EYE pattern

**MMSS\_DP\_HBR2\_COMPLIANCE\_SCRAMBLER\_RESET (cont.)**

Bits	Name	Description
15:0	SCRAMBLER_RESET_COUNT	<p>Count of number of scrambled 0 symbols to be output for every enhanced framing scrambler reset sequence (SR BF BF SR). Count includes the reset sequence.</p> <p>A value less than four causes scrambled 0 symbols to be output with no scrambler reset sequence. The same value shall be programmed into DPCD 24Ah-24Bh.</p> <p>Program MAINLINK_LEVELS:SAFE_TO_EXIT_LEVEL for HBR2 compliance pattern to 1 iff SCRAMBLER_RESET_COUNT=4, otherwise 2</p>

**0x0C990480 MMSS\_DP\_AUDIO\_TIMING\_GEN****Type:** RW**Reset State:** 0x00000000

Audio NAUD and MAUD generic values

**MMSS\_DP\_AUDIO\_TIMING\_GEN**

Bits	Name	Description
31:16	MAUD_GEN	Audio MAUD generic value
15:0	NAUD_GEN	Audio NAUD generic value

**0x0C990484 MMSS\_DP\_AUDIO\_TIMING\_RBR\_32****Type:** RW**Reset State:** 0x0400278D

Audio NAUD and MAUD for Reduced Bit Rate (RBR) 1.62Gbps @ SF 32 KHz

**MMSS\_DP\_AUDIO\_TIMING\_RBR\_32**

Bits	Name	Description
31:16	MAUD_32_RBR	Audio MAUD for RBR 32 KHz
15:0	NAUD_32_RBR	Audio NAUD for RBR 32 KHz

**0x0C990488 MMSS\_DP\_AUDIO\_TIMING\_HBR\_32****Type:** RW**Reset State:** 0x040041EB

Audio NAUD and MAUD for High Bit Rate (HBR) 2.72Gbps @ SF 32 KHz

**MMSS\_DP\_AUDIO\_TIMING\_HBR\_32**

Bits	Name	Description
31:16	MAUD_32_HBR	Audio MAUD for HBR 32 KHz
15:0	NAUD_32_HBR	Audio NAUD for HBR 32 KHz

**0x0C99048C MMSS\_DP\_AUDIO\_TIMING\_RBR\_44****Type:** RW**Reset State:** 0x031015F9

Audio NAUD and MAUD for Reduced Bit Rate (RBR) 1.62Gbps @ SF 44 KHz

**MMSS\_DP\_AUDIO\_TIMING\_RBR\_44**

Bits	Name	Description
31:16	MAUD_44_RBR	Audio MAUD for RBR 44 KHz
15:0	NAUD_44_RBR	Audio NAUD for RBR 44 KHz

**0x0C990490 MMSS\_DP\_AUDIO\_TIMING\_HBR\_44****Type:** RW**Reset State:** 0x0310249F

Audio NAUD and MAUD for High Bit Rate (HBR) 2.72Gbps @ SF 44 KHz

**MMSS\_DP\_AUDIO\_TIMING\_HBR\_44**

Bits	Name	Description
31:16	MAUD_44_HBR	Audio MAUD for HBR 44 KHz
15:0	NAUD_44_HBR	Audio NAUD for HBR 44 KHz

**0x0C990494 MMSS\_DP\_AUDIO\_TIMING\_RBR\_48****Type:** RW**Reset State:** 0x02000D2F

Audio NAUD and MAUD for Reduced Bit Rate (RBR) 1.62Gbps @ SF 48 KHz

**MMSS\_DP\_AUDIO\_TIMING\_RBR\_48**

Bits	Name	Description
31:16	MAUD_48_RBR	Audio MAUD for RBR 48 KHz
15:0	NAUD_48_RBR	Audio NAUD for RBR 48 KHz

**0x0C990498 MMSS\_DP\_AUDIO\_TIMING\_HBR\_48****Type:** RW**Reset State:** 0x020015F9

Audio NAUD and MAUD for High Bit Rate (HBR) 2.72Gbps @ SF 48 KHz

**MMSS\_DP\_AUDIO\_TIMING\_HBR\_48**

Bits	Name	Description
31:16	MAUD_48_HBR	Audio MAUD for HBR 48 KHz
15:0	NAUD_48_HBR	Audio NAUD for HBR 48 KHz

**0x0C9904A0 MMSS\_DP\_MISR\_CTRL****Type:** RW**Reset State:** 0x00000010

MISR Control

**MMSS\_DP\_MISR\_CTRL**

Bits	Name	Description
19:4	MISR_FRAME_COUNT	Determines how many frames to calculate CRC32 over for MISR. For a single frame, this register should be programmed to 1.
3	MISR_NO_FILLER	Exclude fillers in a TU from CRC32 calculation
2	MISR_ACTIVE_ONLY	Calculate CRC32 only for active area and not blanking areas
1	MISR_SW_RESET	MISR software reset, when it is set, MISR is reset, after a 0x1, a 0x0 must be written for normal operation
0	MISR_ENABLE	Enable for MISR

**0x0C9904A4 MMSS\_DP\_MISR\_VALUE\_LANE0****Type:** R**Reset State:** 0xFFFFFFFF

CRC32 value for lane0 over the programmed number of frames, can be read along with CRC\_UPDATED interrupt or FRAME\_END interrupt

**MMSS\_DP\_MISR\_VALUE\_LANE0**

Bits	Name	Description
31:0	MISR_LANE0	

**0x0C9904A8 MMSS\_DP\_MISR\_VALUE\_LANE1****Type:** R**Reset State:** 0xFFFFFFFF

CRC32 value for lane1 over the programmed number of frames, can be read along with CRC\_UPDATED interrupt or FRAME\_END interrupt

**MMSS\_DP\_MISR\_VALUE\_LANE1**

Bits	Name	Description
31:0	MISR_LANE1	

**0x0C9904AC MMSS\_DP\_MISR\_VALUE\_LANE2****Type:** R**Reset State:** 0xFFFFFFFF

CRC32 value for lane2 over the programmed number of frames, can be read along with CRC\_UPDATED interrupt or FRAME\_END interrupt

**MMSS\_DP\_MISR\_VALUE\_LANE2**

Bits	Name	Description
31:0	MISR_LANE2	

**0x0C9904B0 MMSS\_DP\_MISR\_VALUE\_LANE3****Type:** R**Reset State:** 0xFFFFFFFF

CRC32 value for lane3 over the programmed number of frames, can be read along with CRC\_UPDATED interrupt or FRAME\_END interrupt

#### **MMSS\_DP\_MISR\_VALUE\_LANE3**

Bits	Name	Description
31:0	MISR_LANE3	

#### **0x0C9904B4 MMSS\_DP\_MISR\_FILLER**

**Type:** R

**Reset State:** 0x00000000

Number of fillers over the programmed number of frames programmed in MISR\_CTRL

#### **MMSS\_DP\_MISR\_FILLER**

Bits	Name	Description
31:0	FILLER_COUNT	Number of fillers over the programmed of frames, can be read along with the CRC_UPDATED or FRAME_END interrupt

#### **0x0C9904B8 MMSS\_DP\_MISR\_WRONG\_FILLERS\_1**

**Type:** R

**Reset State:** 0x00000000

DP Mainlink's filler error detection register 1

#### **MMSS\_DP\_MISR\_WRONG\_FILLERS\_1**

Bits	Name	Description
31:24	TU_NUMBER	Indicates which was the first TU that had wrong size. TUs are counted from 1.
23:16	WRONG_FILLERS_COUNT	Indicates how many fillers in the programmed number of frames had a wrong size as explained in the WRONG_FILLER field, SHOULD ONLY BE READ WITH CRC_UPDATED INTERRUPT
0	WRONG_FILLER	If 1, it indicates a filler size unequal to the one programmed (TU size - valid_boundary_LINK) has been detected, the other fields of the this register and MISR_WRONG_FILLERS_2 should be used to fingure out where/when this happened, SHOULD ONLY BE READ WITH CRC_UPDATED INTERRUPT

**0x0C9904BC MMSS\_DP\_MISR\_WRONG\_FILLERS\_2****Type:** R**Reset State:** 0x00000000

DP Mainlink's filler error detection register 1

**MMSS\_DP\_MISR\_WRONG\_FILLERS\_2**

Bits	Name	Description
28:16	LINE_NUMBER	Indicates the line number for the first TU with wrong size, counting from 1. Line counter resets to 1 at the end of each frame, SHOULD ONLY BE READ WITH CRC_UPDATED INTERRUPT
15:0	FRAME_NUMBER	Indicates the frame number for the first TU with wrong size, counting from 1. Frame counter resets to 1 at each CRC_UPDATED, SHOULD ONLY BE READ WITH CRC_UPDATED INTERRUPT

**0x0C9904D0 MMSS\_DP\_MISR40\_CTRL****Type:** RW**Reset State:** 0x00000000

40 bit MISR Control

**MMSS\_DP\_MISR40\_CTRL**

Bits	Name	Type	Description
10	MISR40_STATUS_CLEAR	W	MISR status is cleared by writing 1 to this field.
9	MISR40_STATUS	R	Flag when MISR signature is updated.
8	MISR40_ENABLE	RW	1: enable MISR on the next vblank edge. 0: disable MISR on the next vblank edge.
7:0	MISR40_FRAME_COUNT	RW	Number of frames of data to be input to misr before capturing the signature. When this is set to 0, misr will be actively capturing until MISR40_ENABLE is off.

**0x0C9904D4 MMSS\_DP\_MISR40\_SIG\_LANE\_00****Type:** R**Reset State:** 0x00000000

Lane 0 MISR (lower 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_00**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904D8 MMSS\_DP\_MISR40\_SIG\_LANE\_01****Type:** R**Reset State:** 0x00000000

Lane 0 MISR (upper 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_01**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904DC MMSS\_DP\_MISR40\_SIG\_LANE\_10****Type:** R**Reset State:** 0x00000000

Lane 1 MISR (lower 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_10**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904E0 MMSS\_DP\_MISR40\_SIG\_LANE\_11****Type:** R**Reset State:** 0x00000000

Lane 1 MISR (upper 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_11**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904E4 MMSS\_DP\_MISR40\_SIG\_LANE\_20****Type:** R**Reset State:** 0x00000000

Lane 2 MISR (lower 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_20**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904E8 MMSS\_DP\_MISR40\_SIG\_LANE\_21****Type:** R**Reset State:** 0x00000000

Lane 2 MISR (upper 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_21**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904EC MMSS\_DP\_MISR40\_SIG\_LANE\_30****Type:** R**Reset State:** 0x00000000

Lane 3 MISR (lower 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_30**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C9904F0 MMSS\_DP\_MISR40\_SIG\_LANE\_31****Type:** R**Reset State:** 0x00000000

Lane 3 MISR (upper 20 bits data).

**MMSS\_DP\_MISR40\_SIG\_LANE\_31**

Bits	Name	Description
31:0	CAPTURED	32-bit MISR signature.

**0x0C990570 MMSS\_DP\_VSC\_HB0\_PB0\_HB1\_PB1****Type:** RW**Reset State:** 0x16070000

VSC Header Byte(HB) and Parity Byte(PB) ONLY for 3D config

**MMSS\_DP\_VSC\_HB0\_PB0\_HB1\_PB1**

Bits	Name	Description
31:24	PB1	This is the parity byte for HB1, it should be calculated using RS(15, 13) according to section 2.2.6 of DP 1.2 standard. It should be noted that 13 nibbles with value of 0 must be used as msb to calculate the RS(15,13) for HB.
23:16	HB1	This is the header byte for the VSC packet sent for 3D configurations. For the definition, please refer to page 94 of DP 1.2 standard.
15:8	PB0	This is the parity byte for HB0, it should be calculated using RS(15, 13) according to section 2.2.6 of DP 1.2 standard. It should be noted that 13 nibbles with value of 0 must be used as msb to calculate the RS(15,13) for HB.
7:0	HB0	This is the header byte for the VSC packet sent for 3D configurations. For the definition, please refer to page 94 of DP 1.2 standard.

**0x0C990574 MMSS\_DP\_VSC\_HB2\_PB2\_HB3\_PB3****Type:** RW**Reset State:** 0x67016701

VSC Header Byte(HB) and Parity Byte(PB) ONLY for 3D config

**MMSS\_DP\_VSC\_HB2\_PB2\_HB3\_PB3**

Bits	Name	Description
31:24	PB3	This is the parity byte for HB3, it should be calculated using RS(15, 13) according to section 2.2.6 of DP 1.2 standard. It should be noted that 13 nibbles with value of 0 must be used as msb to calculate the RS(15,13) for HB, default value should work

**MMSS\_DP\_VSC\_HB2\_PB2\_HB3\_PB3 (cont.)**

Bits	Name	Description
23:16	HB3	This is the header byte for the VSC packet sent for 3D configurations. For the definition, please refer to page 94 of DP 1.2 standard, default value should work.
15:8	PB2	This is the parity byte for HB2, it should be calculated using RS (15, 13) according to section 2.2.6 of DP 1.2 standard. It should be noted that 13 nibbles with value of 0 must be used as msb to calculate the RS(15,13) for HB, default value should work
7:0	HB2	This is the header byte for the VSC packet sent for 3D configurations. For the definition, please refer to page 94 of DP 1.2 standard, default value should work.

**0x0C990578 MMSS\_DP\_VSC\_DB0\_PB4****Type:** RW**Reset State:** 0x00000000

VSC Data Byte(DB) and Parity Byte(PB) ONLY for 3D config

**MMSS\_DP\_VSC\_DB0\_PB4**

Bits	Name	Description
15:8	PB4	This is the parity byte for DB0, it should be calculated using RS (15, 13) according to section 2.2.6 of DP 1.2 standard.
7:0	DB0	This is the dB for the VSC packet sent for 3D configurations. For the definition of bits, please refer to table 2-56, page 94 of DP 1.2 standard.

**0x0C990580 MMSS\_DP\_COMPRESSION\_MODE\_CTRL****Type:** RW**Reset State:** 0x00000000

Compression control for controller

double buffered: NO

**MMSS\_DP\_COMPRESSION\_MODE\_CTRL**

Bits	Name	Description
8	UPDATE_PPS	VSCEXT_SD_PPS_UPDATE: Trigger SDP values in registers, similar to a flush

**0x0C990600 MMSS\_DP\_AUDIO\_CFG****Type:** RW**Reset State:** 0x000000C0

DP Audio Engine Configuration

double buffered: NO

**MMSS\_DP\_AUDIO\_CFG**

Bits	Name	Description
8:3	FIFO_WATERMARK	Watermark for DP fifo. DMA requests when fifo level is < watermark When fifo level reaches watermark, DMA engine will not request additional audio sample data. There has been no change in the audio DMA logic - this is an updated and correct description 0x0: Reserved 0x1: when empty 0x18: Fill to max 24dec (default)
0	ENABLE	Enable the audio engine 0: Disabled 1: Enabled.

**0x0C99060C MMSS\_DP\_AUDIO\_PKT\_CTRL2****Type:** RW**Reset State:** 0x00000011

DP Audio Engine Configuration

double buffered: NO

**MMSS\_DP\_AUDIO\_PKT\_CTRL2**

Bits	Name	Description
13:12	HDCP_BLOCK_AUDIO_MODE	HDCP BLOCK AUDIO mode 0: Set the entire 3 bytes SDP payload to 0 (default) 1: Set the entire 3 bytes SDP payload to 0, and set AudioMute_Flag=1 in VB_ID 2: Set the entire 3 bytes SDP payload to 0, and set P=1, C=0, U=0, V=1 3: Set the entire 3 bytes SDP payload to 0, set AudioMute_Flag=1 in VB_ID, and set P=1, C=0, U=0, V=1.

**0x0C990610 MMSS\_DP\_AUDIO\_ACR\_CTRL****Type:** RW**Reset State:** 0x00004100

DP Audio Engine ACR Control

double buffered: NO

**MMSS\_DP\_AUDIO\_ACR\_CTRL**

Bits	Name	Description
30	AUDIO_TIMING_OVERRIDE	
5:4	SELECT	Link Speed select 0: 162 MHz RBR (default) 1: 270 MHz HBR 2: 540 MHz HBR2 3: Override enable for Maud_gen/Naud_gen

**0x0C990614 MMSS\_DP\_AUDIO\_CTRL\_RESET****Type:** RW**Reset State:** 0x00000001

DP Audio Engine FIFO Reset enable

double buffered: NO

**MMSS\_DP\_AUDIO\_CTRL\_RESET**

Bits	Name	Description
0	RESET_FIFO_DP_ENABLE	Audio Engine FIFO Reset possible enable 0: Reset not possible/enabled 1: Reset possible/enabled (default)

**0x0C990628 MMSS\_DP\_SDP\_CFG****Type:** RW**Reset State:** 0x00100006

SDP enables

double buffered: SOME

**MMSS\_DP\_SDPCFG**

Bits	Name	Description
20	AUDIO_INFOFRAME_SDPCFG_EN	Audio Infoframe SDP enable
18	GEN1_SDPCFG_EN	Generic 1 SDP enable (double buffered)
17	GEN0_SDPCFG_EN	Generic 0 SDP enable (double buffered)
16	VSCEXT_SDPCFG_EN	VSC extension SDP enable (double buffered)
6	AUDIO_ISRC_SDPCFG_EN	Audio ISRC SDP enable
5	AUDIO_COPY_MANAGEMENT_SDPCFG_EN	Audio copy management SDP enable
4	EXTENSION_SDPCFG_EN	Extension SDP enable
2	AUDIO_STREAM_SDPCFG_EN	Audio stream SDP enable
1	AUDIO_TIMESTAMP_SDPCFG_EN	Audio timestamp SDP enable

**0x0C99062C MMSS\_DP\_SDPCFG2****Type:** RW**Reset State:** 0x00000000

SDP enables

double buffered: NO

**MMSS\_DP\_SDPCFG2**

Bits	Name	Description
17	GENERIC1_SDPSIZE	Generic 1 SDP payload size: 0-16 DBs, 1-32 DBs
16	GENERIC0_SDPSIZE	Generic 0 SDP payload size: 0-16 DBs, 1-32 DBs
15	EXTN_SDPSIZE	Extension SDP payload size: 0-16 DBs, 1-32 DBs
1	AUDIO_STREAM_HB3_REG_SRC	Audio Stream SDP HB3 data src: 0-Metadata intf, 1-All from registers
0	IFRM_REGSRC	Audio Infoframe SDP data src: 0-Metadata intf, 1-All from registers

**0x0C990630 MMSS\_DP\_AUDIO\_TIMESTAMP\_0****Type:** RW**Reset State:** 0x00000000

Audio Time Stamp SDP contents

**MMSS\_DP\_AUDIO\_TIMESTAMP\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C990634 MMSS\_DP\_AUDIO\_TIMESTAMP\_1****Type:** RW**Reset State:** 0x00000000

Audio Time Stamp SDP contents

**MMSS\_DP\_AUDIO\_TIMESTAMP\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990640 MMSS\_DP\_AUDIO\_STREAM\_0****Type:** RW**Reset State:** 0x00000000

Audio Stream SDP contents

**MMSS\_DP\_AUDIO\_STREAM\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	ParityByte 0
7:0	HB0	Header Byte 0

**0x0C990644 MMSS\_DP\_AUDIO\_STREAM\_1****Type:** RW**Reset State:** 0x00000000

Audio Stream SDP contents

**MMSS\_DP\_AUDIO\_STREAM\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990650 MMSS\_DP\_EXTENSION\_0****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C990654 MMSS\_DP\_EXTENSION\_1****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3

**MMSS\_DP\_EXTENSION\_1 (cont.)**

Bits	Name	Description
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990658 MMSS\_DP\_EXTENSION\_2****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

**0x0C99065C MMSS\_DP\_EXTENSION\_3****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

**0x0C990660 MMSS\_DP\_EXTENSION\_4****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

#### **MMSS\_DP\_EXTENSION\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

#### **0x0C990664 MMSS\_DP\_EXTENSION\_5**

**Type:** RW

**Reset State:** 0x00000000

Extension SDP contents

#### **MMSS\_DP\_EXTENSION\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

#### **0x0C990668 MMSS\_DP\_EXTENSION\_6**

**Type:** RW

**Reset State:** 0x00000000

Extension SDP contents

#### **MMSS\_DP\_EXTENSION\_6**

Bits	Name	Description
31:24	BYTE19	Payload byte 19
23:16	BYTE18	Payload Byte 18
15:8	BYTE17	Payload Byte 17
7:0	BYTE16	Payload Byte 16

**0x0C99066C MMSS\_DP\_EXTENSION\_7****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_7**

Bits	Name	Description
31:24	BYTE23	Payload byte 23
23:16	BYTE22	Payload Byte 22
15:8	BYTE21	Payload Byte 21
7:0	BYTE20	Payload Byte 20

**0x0C990670 MMSS\_DP\_EXTENSION\_8****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_8**

Bits	Name	Description
31:24	BYTE27	Payload byte 27
23:16	BYTE26	Payload Byte 26
15:8	BYTE25	Payload Byte 25
7:0	BYTE24	Payload Byte 24

**0x0C990674 MMSS\_DP\_EXTENSION\_9****Type:** RW**Reset State:** 0x00000000

Extension SDP contents

**MMSS\_DP\_EXTENSION\_9**

Bits	Name	Description
31:24	BYTE31	Payload byte 31
23:16	BYTE30	Payload Byte 30

**MMSS\_DP\_EXTENSION\_9 (cont.)**

Bits	Name	Description
15:8	BYTE29	Payload Byte 29
7:0	BYTE28	Payload Byte 28

**0x0C990678 MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_0****Type:** RW**Reset State:** 0x00000000

Audio Copy Management SDP contents

**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C99067C MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_1****Type:** RW**Reset State:** 0x00000000

Audio Copy Management SDP contents

**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990680 MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_2****Type:** RW**Reset State:** 0x00000000

Audio Copy Management SDP contents

#### **MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

#### **0x0C990684 MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_3**

**Type:** RW

**Reset State:** 0x00000000

Audio Copy Management SDP contents

#### **MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

#### **0x0C990688 MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_4**

**Type:** RW

**Reset State:** 0x00000000

Audio Copy Management SDP contents

#### **MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

**0x0C99068C MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_5****Type:** RW**Reset State:** 0x00000000

Audio Copy Management SDP contents

**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

**0x0C990690 MMSS\_DP\_AUDIO\_ISRC\_0****Type:** RW**Reset State:** 0x00000000

Audio ISRC SDP contents

**MMSS\_DP\_AUDIO\_ISRC\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C990694 MMSS\_DP\_AUDIO\_ISRC\_1****Type:** RW**Reset State:** 0x00000000

Audio ISRC SDP contents

**MMSS\_DP\_AUDIO\_ISRC\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3

**MMSS\_DP\_AUDIO\_ISRC\_1 (cont.)**

Bits	Name	Description
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990698 MMSS\_DP\_AUDIO\_ISRC\_2****Type:** RW**Reset State:** 0x00000000

Audio ISRC SDP contents

**MMSS\_DP\_AUDIO\_ISRC\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

**0x0C99069C MMSS\_DP\_AUDIO\_ISRC\_3****Type:** RW**Reset State:** 0x00000000

Audio ISRC SDP contents

**MMSS\_DP\_AUDIO\_ISRC\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

**0x0C9906A0 MMSS\_DP\_AUDIO\_ISRC\_4****Type:** RW**Reset State:** 0x00000000

Audio ISRC SDP contents

#### **MMSS\_DP\_AUDIO\_ISRC\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

#### **0x0C9906A4 MMSS\_DP\_AUDIO\_ISRC\_5**

**Type:** RW

**Reset State:** 0x00000000

Audio ISRC SDP contents

#### **MMSS\_DP\_AUDIO\_ISRC\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

#### **0x0C9906A8 MMSS\_DP\_AUDIO\_INFOFRAME\_0**

**Type:** RW

**Reset State:** 0x00000000

Audio InfoFrame SDP contents

#### **MMSS\_DP\_AUDIO\_INFOFRAME\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C9906AC MMSS\_DP\_AUDIO\_INFOFRAME\_1****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

**MMSS\_DP\_AUDIO\_INFOFRAME\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C9906B0 MMSS\_DP\_AUDIO\_INFOFRAME\_2****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

**MMSS\_DP\_AUDIO\_INFOFRAME\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

**0x0C9906B4 MMSS\_DP\_AUDIO\_INFOFRAME\_3****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

**MMSS\_DP\_AUDIO\_INFOFRAME\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6

**MMSS\_DP\_AUDIO\_INFOFRAME\_3 (cont.)**

Bits	Name	Description
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

**0x0C9906B8 MMSS\_DP\_AUDIO\_INFOFRAME\_4****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

**MMSS\_DP\_AUDIO\_INFOFRAME\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

**0x0C9906BC MMSS\_DP\_AUDIO\_INFOFRAME\_5****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

**MMSS\_DP\_AUDIO\_INFOFRAME\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

**0x0C9906C0 MMSS\_DP\_AUDIO\_INFOFRAME\_6****Type:** RW**Reset State:** 0x00000000

Audio InfoFrame SDP contents

#### **MMSS\_DP\_AUDIO\_INFOFRAME\_6**

Bits	Name	Description
31:24	BYTE19	Payload byte 19
23:16	BYTE18	Payload Byte 18
15:8	BYTE17	Payload Byte 17
7:0	BYTE16	Payload Byte 16

#### **0x0C9906C4 MMSS\_DP\_AUDIO\_INFOFRAME\_7**

**Type:** RW

**Reset State:** 0x00000000

Audio InfoFrame SDP contents

#### **MMSS\_DP\_AUDIO\_INFOFRAME\_7**

Bits	Name	Description
31:24	BYTE23	Payload byte 23
23:16	BYTE22	Payload Byte 22
15:8	BYTE21	Payload Byte 21
7:0	BYTE20	Payload Byte 20

#### **0x0C9906C8 MMSS\_DP\_AUDIO\_INFOFRAME\_8**

**Type:** RW

**Reset State:** 0x00000000

Audio InfoFrame SDP contents

#### **MMSS\_DP\_AUDIO\_INFOFRAME\_8**

Bits	Name	Description
31:24	BYTE27	Payload byte 27
23:16	BYTE26	Payload Byte 26
15:8	BYTE25	Payload Byte 25
7:0	BYTE24	Payload Byte 24

**0x0C9906D0 MMSS\_DP\_VSCEXT\_0****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C9906D4 MMSS\_DP\_VSCEXT\_1****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C9906D8 MMSS\_DP\_VSCEXT\_2****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2

**MMSS\_DP\_VSCEXT\_2 (cont.)**

Bits	Name	Description
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

**0x0C9906DC MMSS\_DP\_VSCEXT\_3****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

**0x0C9906E0 MMSS\_DP\_VSCEXT\_4****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

**0x0C9906E4 MMSS\_DP\_VSCEXT\_5****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

#### **MMSS\_DP\_VSCEXT\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

#### **0x0C9906E8 MMSS\_DP\_VSCEXT\_6**

**Type:** RW

**Reset State:** 0x00000000

VSC Extension SDP contents

#### **MMSS\_DP\_VSCEXT\_6**

Bits	Name	Description
31:24	BYTE19	Payload byte 19
23:16	BYTE18	Payload Byte 18
15:8	BYTE17	Payload Byte 17
7:0	BYTE16	Payload Byte 16

#### **0x0C9906EC MMSS\_DP\_VSCEXT\_7**

**Type:** RW

**Reset State:** 0x00000000

VSC Extension SDP contents

#### **MMSS\_DP\_VSCEXT\_7**

Bits	Name	Description
31:24	BYTE23	Payload byte 23
23:16	BYTE22	Payload Byte 22
15:8	BYTE21	Payload Byte 21
7:0	BYTE20	Payload Byte 20

**0x0C9906F0 MMSS\_DP\_VSCEXT\_8****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_8**

Bits	Name	Description
31:24	BYTE27	Payload byte 27
23:16	BYTE26	Payload Byte 26
15:8	BYTE25	Payload Byte 25
7:0	BYTE24	Payload Byte 24

**0x0C9906F4 MMSS\_DP\_VSCEXT\_9****Type:** RW**Reset State:** 0x00000000

VSC Extension SDP contents

**MMSS\_DP\_VSCEXT\_9**

Bits	Name	Description
31:24	BYTE31	Payload byte 31
23:16	BYTE30	Payload Byte 30
15:8	BYTE29	Payload Byte 29
7:0	BYTE28	Payload Byte 28

**0x0C990700 MMSS\_DP\_GENERIC0\_0****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1

**MMSS\_DP\_GENERIC0\_0 (cont.)**

Bits	Name	Description
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C990704 MMSS\_DP\_GENERIC0\_1****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

**0x0C990708 MMSS\_DP\_GENERIC0\_2****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

**0x0C99070C MMSS\_DP\_GENERIC0\_3****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

#### **MMSS\_DP\_GENERIC0\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

#### **0x0C990710 MMSS\_DP\_GENERIC0\_4**

**Type:** RW

**Reset State:** 0x00000000

Generic 0 SDP contents

#### **MMSS\_DP\_GENERIC0\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

#### **0x0C990714 MMSS\_DP\_GENERIC0\_5**

**Type:** RW

**Reset State:** 0x00000000

Generic 0 SDP contents

#### **MMSS\_DP\_GENERIC0\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

**0x0C990718 MMSS\_DP\_GENERIC0\_6****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_6**

Bits	Name	Description
31:24	BYTE19	Payload byte 19
23:16	BYTE18	Payload Byte 18
15:8	BYTE17	Payload Byte 17
7:0	BYTE16	Payload Byte 16

**0x0C99071C MMSS\_DP\_GENERIC0\_7****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_7**

Bits	Name	Description
31:24	BYTE23	Payload byte 23
23:16	BYTE22	Payload Byte 22
15:8	BYTE21	Payload Byte 21
7:0	BYTE20	Payload Byte 20

**0x0C990720 MMSS\_DP\_GENERIC0\_8****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_8**

Bits	Name	Description
31:24	BYTE27	Payload byte 27
23:16	BYTE26	Payload Byte 26

**MMSS\_DP\_GENERIC0\_8 (cont.)**

Bits	Name	Description
15:8	BYTE25	Payload Byte 25
7:0	BYTE24	Payload Byte 24

**0x0C990724 MMSS\_DP\_GENERIC0\_9****Type:** RW**Reset State:** 0x00000000

Generic 0 SDP contents

**MMSS\_DP\_GENERIC0\_9**

Bits	Name	Description
31:24	BYTE31	Payload byte 31
23:16	BYTE30	Payload Byte 30
15:8	BYTE29	Payload Byte 29
7:0	BYTE28	Payload Byte 28

**0x0C990728 MMSS\_DP\_GENERIC1\_0****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_0**

Bits	Name	Description
31:24	PB1	Parity byte 1
23:16	HB1	Header Byte 1
15:8	PB0	Parity Byte 0
7:0	HB0	Header Byte 0

**0x0C99072C MMSS\_DP\_GENERIC1\_1****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

#### **MMSS\_DP\_GENERIC1\_1**

Bits	Name	Description
31:24	PB3	Parity byte 3
23:16	HB3	Header Byte 3
15:8	PB2	Parity Byte 2
7:0	HB2	Header Byte 2

#### **0x0C990730 MMSS\_DP\_GENERIC1\_2**

**Type:** RW

**Reset State:** 0x00000000

Generic 1 SDP contents

#### **MMSS\_DP\_GENERIC1\_2**

Bits	Name	Description
31:24	BYTE3	Payload byte 3
23:16	BYTE2	Payload Byte 2
15:8	BYTE1	Payload Byte 1
7:0	BYTE0	Payload Byte 0

#### **0x0C990734 MMSS\_DP\_GENERIC1\_3**

**Type:** RW

**Reset State:** 0x00000000

Generic 1 SDP contents

#### **MMSS\_DP\_GENERIC1\_3**

Bits	Name	Description
31:24	BYTE7	Payload byte 7
23:16	BYTE6	Payload Byte 6
15:8	BYTE5	Payload Byte 5
7:0	BYTE4	Payload Byte 4

**0x0C990738 MMSS\_DP\_GENERIC1\_4****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_4**

Bits	Name	Description
31:24	BYTE11	Payload byte 11
23:16	BYTE10	Payload Byte 10
15:8	BYTE9	Payload Byte 9
7:0	BYTE8	Payload Byte 8

**0x0C99073C MMSS\_DP\_GENERIC1\_5****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_5**

Bits	Name	Description
31:24	BYTE15	Payload byte 15
23:16	BYTE14	Payload Byte 14
15:8	BYTE13	Payload Byte 13
7:0	BYTE12	Payload Byte 12

**0x0C990740 MMSS\_DP\_GENERIC1\_6****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_6**

Bits	Name	Description
31:24	BYTE19	Payload byte 19
23:16	BYTE18	Payload Byte 18

**MMSS\_DP\_GENERIC1\_6 (cont.)**

Bits	Name	Description
15:8	BYTE17	Payload Byte 17
7:0	BYTE16	Payload Byte 16

**0x0C990744 MMSS\_DP\_GENERIC1\_7****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_7**

Bits	Name	Description
31:24	BYTE23	Payload byte 23
23:16	BYTE22	Payload Byte 22
15:8	BYTE21	Payload Byte 21
7:0	BYTE20	Payload Byte 20

**0x0C990748 MMSS\_DP\_GENERIC1\_8****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

**MMSS\_DP\_GENERIC1\_8**

Bits	Name	Description
31:24	BYTE27	Payload byte 27
23:16	BYTE26	Payload Byte 26
15:8	BYTE25	Payload Byte 25
7:0	BYTE24	Payload Byte 24

**0x0C99074C MMSS\_DP\_GENERIC1\_9****Type:** RW**Reset State:** 0x00000000

Generic 1 SDP contents

#### **MMSS\_DP\_GENERIC1\_9**

Bits	Name	Description
31:24	BYTE31	Payload byte 31
23:16	BYTE30	Payload Byte 30
15:8	BYTE29	Payload Byte 29
7:0	BYTE28	Payload Byte 28

#### **0x0C990A10 MMSS\_DP\_TIMING\_ENGINE\_EN**

**Type:** RW

**Reset State:** 0x00000000

BIST Display Interface Timing Engine Enable register

#### **MMSS\_DP\_TIMING\_ENGINE\_EN**

Bits	Name	Description
8	FRAME_CRC_EN	Enable frame CRC on active pixel. This CRC is collected per frame per component, CRC is recorded in PSR_RG, PSR_B in lclk. If RGB swizzle (RGB_MAPPING) and endian swap (PIXEL_ENDIANNESS) is turned on, they are applied to MDP output color component before feeding into this CRC
0	TIMING_ENGINE_EN_EN	Display Interface enable. Should be the LAST register programmed for the interface Be sure to program ALL retiming registers and setting up of the display pipeline BEFORE enabling this register. Upon disabling of this bit, hardware will stop driving the interface at the NEXT vsync boundary. A VSYNC interrupt WILL be generated but a VSYNC will NOT be generated on the interface

#### **0x0C990A14 MMSS\_DP\_INTF\_CONFIG**

**Type:** RW

**Reset State:** 0x00000000

BIST Display Interface Configuration register

**MMSS\_DP\_INTF\_CONFIG**

Bits	Name	Description
2	ACTIVE_V_EN	Active V Enable
1	ACTIVE_H_EN	Active H Enable
0	INTERLACE_MODE	0x0: Progressive Out 0x1: Interlace out

**0x0C990A18 MMSS\_DP\_HSYNC\_CTL****Type:** RW**Reset State:** 0x00000000

BIST Display Interface HSYNC Control

**MMSS\_DP\_HSYNC\_CTL**

Bits	Name	Description
31:16	HSYNC_PERIOD	HSYNC Period in pixel clocks. This is the start between start of HSYNC pulse and the start of the next HSYNC pulse. Note: $hsync\_period = width + h\_porch$ where width=output image width $h\_porch = (h\_back\_porch + h\_front\_porch)$ h_back_porch starts at the beginning of hsync pulse Please refer to VESA specifications to obtain accurate parameters
15:0	PULSE_WIDTH	HSYNC pulse width in pixel clocks

**0x0C990A1C MMSS\_DP\_VSYNC\_PERIOD\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_VSYNC\_PERIOD\_F0**

Bits	Name	Description
31:0	VSYNC_PERIOD	<p>VSYNC Period in pixel clocks. This is the time between start of vsync pulse and the start of the next vsync pulse.</p> <p>Note  <math>\text{vsync\_period} = \text{height} + \text{v\_porch}</math>  where  <math>\text{height} = \text{output\_image\_height}</math>  <math>\text{v\_porch} = (\text{v\_back\_porch} + \text{v\_front\_porch})</math>  <math>\text{v\_back\_porch}</math> starts at the beginning of vsync pulse.  Please refer to VESA specifications to obtain accurate parameters</p>

**0x0C990A20 MMSS\_DP\_VSYNC\_PERIOD\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_VSYNC\_PERIOD\_F1**

Bits	Name	Description
31:0	VSYNC_PERIOD	<p>VSYNC Period in pixel clocks.</p> <p>In progressive mode, this is the time between start of vsync pulse and the start of next vsync pulse</p> <p>Interlace mode this is the time between start of Field1 vsync pulse and the start of Field2 vsync pulse.</p> <p>Note:  <math>\text{vsync\_period} = \text{Height} + \text{v\_porch}</math>  where,  <math>\text{Height} = \text{Output image height}</math> <math>\text{nv\_porch} = (\text{v\_back\_porch} + \text{v\_front\_porch})</math> <math>\text{nv\_back\_porch}</math> starts at the beginning of vsync pulse</p>

**0x0C990A24 MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F0**

Bits	Name	Description
31:0	VSYNC_PERIOD	VSYNC Period width in pixel clocks

**0x0C990A28 MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F1**

Bits	Name	Description
31:0	VSYNC_PERIOD	VSYNC Period in pixel clocks for the 2nd field. NOT valid for progressive mode

**0x0C990A2C MMSS\_DP\_DISPLAY\_V\_START\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_DISPLAY\_V\_START\_F0**

Bits	Name	Description
31:0	DISPLAY_START_Y	Display Y Start Defines the time period between the field1 start of vsync pulse and the first displayed line position in pixel clocks display_start_y=v_back_porch v_porch=vsync_period+height v_porch=(v_back_porch+v_front_porch)

**0x0C990A30 MMSS\_DP\_DISPLAY\_V\_START\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_DISPLAY\_V\_START\_F1**

Bits	Name	Description
31:0	DISPLAY_START_Y	<p>Display Y Start</p> <p>Defines the start of the 2nd field of vsync pulse and the first displayed line position in pixel clocks</p> <p>display_start_y=v_back_porch</p> <p>v_porch=vsync_period-height</p> <p>v_porch=(v_back_porch+v_front_porch)</p>

**0x0C990A34 MMSS\_DP\_DISPLAY\_V\_END\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_DISPLAY\_V\_END\_F0**

Bits	Name	Description
31:0	DISPLAY_END_Y	Display Y End

**0x0C990A38 MMSS\_DP\_DISPLAY\_V\_END\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_DISPLAY\_V\_END\_F1**

Bits	Name	Description
31:0	DISPLAY_END_Y	Display Y End

**0x0C990A3C MMSS\_DP\_ACTIVE\_V\_START\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_ACTIVE\_V\_START\_F0**

Bits	Name	Description
31:0	ACTIVE_START_Y	Active Y Start

**0x0C990A40 MMSS\_DP\_ACTIVE\_V\_START\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_ACTIVE\_V\_START\_F1**

Bits	Name	Description
31:0	ACTIVE_START_Y	Active Y Start

**0x0C990A44 MMSS\_DP\_ACTIVE\_V\_END\_F0****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 0

**MMSS\_DP\_ACTIVE\_V\_END\_F0**

Bits	Name	Description
31:0	ACTIVE_END_Y	Active Y End

**0x0C990A48 MMSS\_DP\_ACTIVE\_V\_END\_F1****Type:** RW**Reset State:** 0x00000000

BIST Display Interface VSYNC Control for field 1

**MMSS\_DP\_ACTIVE\_V\_END\_F1**

Bits	Name	Description
31:0	ACTIVE_END_Y	Active Y End

**0x0C990A4C MMSS\_DP\_DISPLAY\_HCTL****Type:** RW**Reset State:** 0x00000000

BIST Display Interface HSYNC Control for field 1

**MMSS\_DP\_DISPLAY\_HCTL**

Bits	Name	Description
31:16	DISPLAY_END_X	Display End X
15:0	DISPLAY_START_X	Display Start X

**0x0C990A50 MMSS\_DP\_ACTIVE\_HCTL****Type:** RW**Reset State:** 0x00000000

BIST Display Interface HSYNC Control for field 1

**MMSS\_DP\_ACTIVE\_HCTL**

Bits	Name	Description
31:16	ACTIVE_END_X	Active End X
15:0	ACTIVE_START_X	Active Start X

**0x0C990A54 MMSS\_DP\_HSYNC\_SKEW****Type:** RW**Reset State:** 0x00000000

BIST Display Interface HSYNC Skew Control

double buffered: NO

**MMSS\_DP\_HSYNC\_SKEW**

Bits	Name	Description
15:0	Hsync_Skew	Hsync Skew Defines the number of pixel clocks HSYNC active edge is delayed from VSYNC active edge

**0x0C990A58 MMSS\_DP\_POLARITY\_CTL****Type:** RW**Reset State:** 0x00000000

BIST Display Interface Polarity Control

double buffered: NO

**MMSS\_DP\_POLARITY\_CTL**

Bits	Name	Description
2	DEN	DEN Polarity 0x0: Active HIGH 0x1: Active LOW
1	VSYNC	VSYNC Polarity 0x0: Active HIGH 0x1: Active LOW
0	Hsync	Hsync Polarity 0x0: Active HIGH 0x1: Active LOW

**0x0C990A80 MMSS\_DP\_RGB\_MAPPING****Type:** RW**Reset State:** 0x00000006

RGB mapping for the input data from MDP to DP

**MMSS\_DP\_RGB\_MAPPING**

Bits	Name	Description
5:4	B_MAPPING	Defines which bits of MDP data maps to B, normally default value is fine  00: FROM [7:0] 01: FROM [15:8] 10: FROM [23:16]
3:2	G_MAPPING	Defines which bits of MDP data maps to G, normally default value is fine  00: FROM [7:0] 01: FROM [15:8] 10: FROM [23:16]

**MMSS\_DP\_RGB\_MAPPING (cont.)**

Bits	Name	Description
1:0	R_MAPPING	Defines which bits of MDP data maps to R, normally default value is fine  00: FROM [7:0] 01: FROM [15:8] 10: FROM [23:16]

**0x0C9A8810 MMSS\_ROT\_ROTTOP\_CLK\_CTRL****Type:** RW**Reset State:** 0x00000004

Clock Gating Control

**MMSS\_ROT\_ROTTOP\_CLK\_CTRL**

Bits	Name	Description
16	ROT_VSYNC_CLK_DISABLE	0x0 : VSYNC Clock gating is OFF 0x1 : Force VSYNC clock branch OFF
8	ROT_AXI_CLK_DISABLE	0x0 : AXI Clock gating is OFF 0x1 : Force AXI clock branch OFF
3:2	ROT_CLK_HYSTESIS_CTRL	0x0 : Clock will run for extra 4 cycles before is gated OFF 0x1 : Clock will run for extra 8 cycles before is gated OFF 0x2 : Clock will run for extra 16 cycles before is gated OFF 0x3 : Clock will run for extra 32 cycles before is gated OFF
1	ROT_CLK_FORCE_OFF	0x0 : Clock gating is active if CLK_FORCE_ON=0 and CLK_FORCE_OFF=0 0x1 : Force clock branch OFF if CLK_FORCE_ON=0, otherwise setting is ignored
0	ROT_CLK_FORCE_ON	0x0 : Clock gating is active if CLK_FORCE_ON=0 and CLK_FORCE_OFF=0 0x1 : Force clock branch ON

**0x0C9A8814 MMSS\_ROT\_ROTTOP\_CLK\_STATUS****Type:** R**Reset State:** 0x00000000

Status of gated clock branches

**MMSS\_ROT\_ROTTOP\_CLK\_STATUS**

Bits	Name	Description
0	ROT_CLK_ACTIVE	

**0x0C9A8818 MMSS\_ROT\_ROTTOP\_ROT\_NEWRoi\_PRIOR\_TO\_START****Type:** RW**Reset State:** 0x00000001

The newroi prior to start is used to flush out incorrect states and such between frames. The newroi is sent prior to the start signals from the PMU's. The recommendation is to always have this bit enabled. The recommendation is to always have this bit enabled

**MMSS\_ROT\_ROTTOP\_ROT\_NEWRoi\_PRIOR\_TO\_START**

Bits	Name	Description
0	ENABLE	Enable the mode to send out newroi prior to start

**0x0C9A8820 MMSS\_ROT\_ROTTOP\_SW\_RESET****Type:** RW**Reset State:** 0x00000000

Software Reset of the given control operation

**MMSS\_ROT\_ROTTOP\_SW\_RESET**

Bits	Name	Description
0	RESET	Writing to this bit will trigger a soft reset for the described data path This register is self-clearing and will return 0x1 if soft reset sequence is not complete Upon completion, this will return 0x0

**0x0C9A8824 MMSS\_ROT\_ROTTOP\_SW\_RESET\_CTRL****Type:** RW**Reset State:** 0x00000038

The SW\_RESET\_CTRL register is used for the controls required for software reset

**MMSS\_ROT\_ROTTOP\_SW\_RESET\_CTRL**

Bits	Name	Description
7:0	HYSTERICIS_CTRL	This field is program to control how many core clock cycles the software reset is held for. The default is 56 clock cycles

**0x0C9A8828 MMSS\_ROT\_ROTTOP\_SW\_RESET\_OVERRIDE****Type:** RW**Reset State:** 0x00000000

Software Reset Override of the given control operation

**MMSS\_ROT\_ROTTOP\_SW\_RESET\_OVERRIDE**

Bits	Name	Description
0	RESET	<p>Writing to this bit will assert reset for the described data path</p> <p>This value contained in this register will be forced onto the data path</p> <p>It will reset the state machines within the scheduler as well for the given data path</p> <p>To use this register sequence, software shall write a value of 0x1 to this register to assert reset</p> <p>Once software wants to de-assert reset, software shall write a value of 0x0</p> <p>Software will need to ensure the duration of the value 0x1 is long enough to ensure the path is clean and all memory reads are complete</p> <p>The data path will not be triggered if this bit is asserted</p>

**0x0C9A8830 MMSS\_ROT\_ROTTOP\_INTR\_EN****Type:** RW**Reset State:** 0x00000000

ROT Interrupt Enable Control register

**MMSS\_ROT\_ROTTOP\_INTR\_EN**

Bits	Name	Description
0	ROT_DONE_EN	<p>Rotator Done</p> <p>0x0: Interrupt Disabled</p> <p>0x1: Interrupt Enabled</p>

**0x0C9A8834 MMSS\_ROT\_ROTTOP\_INTR\_STATUS****Type:** R**Reset State:** 0x00000000

ROT Interrupt Status Control register

**MMSS\_ROT\_ROTTOP\_INTR\_STATUS**

Bits	Name	Description
0	ROT_DONE	Rotator Done 0x0: 0x1: Rotation done

**0x0C9A8838 MMSS\_ROT\_ROTTOP\_INTR\_CLEAR****Type:** W**Reset State:** 0x00000000

ROT Interrupt Clear Control register

**MMSS\_ROT\_ROTTOP\_INTR\_CLEAR**

Bits	Name	Description
0	ROT_DONE_CLR	Rotator Done Clear 0x0: 0x1: Interrupt Clear

**0x0C9A8840 MMSS\_ROT\_ROTTOP\_START\_CTRL****Type:** W**Reset State:** 0x00000000

Rotator Start Signal for the given Control Operation

**MMSS\_ROT\_ROTTOP\_START\_CTRL**

Bits	Name	Description
0	START_CTRL	A write to this register will kick off Rotator

**0x0C9A8844 MMSS\_ROT\_ROTTOP\_STATUS****Type:** RW**Reset State:** 0x00000000

Rotator Status register

**MMSS\_ROT\_ROTTOP\_STATUS**

Bits	Name	Type	Description
31	ERROR_CLEAR	W	Clear error register
8	ERROR	R	Error register
0	ROT_BUSY	R	0x0: Rotator is IDLE 0x1: Rotator is BUSY

**0x0C9A8848 MMSS\_ROT\_ROTTOP\_OP\_MODE****Type:** RW**Reset State:** 0x00000000

Destination Operation Mode

double buffered: NO

**MMSS\_ROT\_ROTTOP\_OP\_MODE**

Bits	Name	Description
1	ROT_MODE	0x0: Rotate 0 0x1: Rotate 90
0	ROT_EN	0x0: Rotate Disabled 0x1: Rotate Enabled

**0x0C9A884C MMSS\_ROT\_ROTTOP\_DNSC****Type:** RW**Reset State:** 0x00000000

Rotator downscaler configuration

double buffered: NO

**MMSS\_ROT\_ROTTOP\_DNSC**

Bits	Name	Description
23:16	DNSC_FACTOR_V	Downscale factor in vertical direction 0: bypass 1: downscale by 1.5 2: downscale by 2 4: downscale by 4 8: downscale by 8 16: downscale by 16 32: downscale by 32 64: downscale by 64 0x0: V_BYPASS 0x1: V_DNSC_X1P5 0x2: V_DNSC_X2 0x4: V_DNSC_X4 0x8: V_DNSC_X8 0x10: V_DNSC_X16 0x20: V_DNSC_X32 0x40: V_DNSC_X64
7:0	DNSC_FACTOR_H	Downscale factor in horizontal direction 0: bypass 1: downscale by 1.5 2: downscale by 2 4: downscale by 4 8: downscale by 8 16: downscale by 16 32: downscale by 32 64: downscale by 64 0x0: H_BYPASS 0x1: H_DNSC_X1P5 0x2: H_DNSC_X2 0x4: H_DNSC_X4 0x8: H_DNSC_X8 0x10: H_DNSC_X16 0x20: H_DNSC_X32 0x40: H_DNSC_X64

**0x0C9A8900 MMSS\_ROT\_SSPP\_SRC\_SIZE****Type:** RW**Reset State:** 0x00000000

Source Size for the region of interest source image

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_SIZE**

Bits	Name	Description
31:16	ROI_H	Source Height of the Region Of Interest (ROI)
15:0	ROI_W	Source Width of the Region Of Interest (ROI)

**0x0C9A8908 MMSS\_ROT\_SSPP\_SRC\_XY****Type:** RW**Reset State:** 0x00000000

Source Offset relative to the top-left corner of the source image for each of the region of interest dimensions

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_XY**

Bits	Name	Description
31:16	SRC_Y	Y-offset of source ROI in the source image
15:0	SRC_X	X-offset of source ROI in the source image

**0x0C9A890C MMSS\_ROT\_SSPP\_OUT\_SIZE****Type:** RW**Reset State:** 0x00000000

Output Size of the layer.

Should always be explicitly set and same as source size if scaling is not enabled.

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_OUT\_SIZE**

Bits	Name	Description
31:16	DST_H	Height of Output ROI
15:0	DST_W	Width of Output ROI

**0x0C9A8910 MMSS\_ROT\_SSPP\_OUT\_XY****Type:** RW**Reset State:** 0x00000000

Output Offset relative to the top-left corner of the final output blended image

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_OUT\_XY**

Bits	Name	Description
31:16	DST_Y	Height of Output ROI
15:0	DST_X	Width of Output ROI

**0x0C9A8914 MMSS\_ROT\_SSPP\_SRC0\_ADDR****Type:** RW**Reset State:** 0x00000000

Plane 0 base address

For linear formats, this plane refers to ARGB/Y plane

For compressed UBWC formats, this plane refers to the compressed data plane

double buffered: YES

when: On frame boundaries (vsync or software trigger) when ADDR\_SYNC\_MODE is in frame mode and on line boundary if ADDR\_SYNC\_MODE in in line mode

**MMSS\_ROT\_SSPP\_SRC0\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address plane0

**0x0C9A8918 MMSS\_ROT\_SSPP\_SRC1\_ADDR****Type:** RW**Reset State:** 0x00000000

Plane 1 base address

For linear formats, this plane refers to UV or U plane

For compressed UBWC formats, this plane refers to the compressed UV data plane

double buffered: YES

when: On frame boundaries (vsync or software trigger) when ADDR\_SYNC\_MODE is in frame mode and on line boundary if ADDR\_SYNC\_MODE in in line mode

#### **MMSS\_ROT\_SSPP\_SRC1\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address of plane1

#### **0xC9A891C MMSS\_ROT\_SSPP\_SRC2\_ADDR**

**Type:** RW

**Reset State:** 0x00000000

Plane 2 base address

For linear formats, this plane refers to V plane

For compressed UBWC formats, this plane refers to the Y/RGB metadata plane for plane0

double buffered: YES

when: On frame boundaries (vsync or software trigger) when ADDR\_SYNC\_MODE is in frame mode and on line boundary if ADDR\_SYNC\_MODE in in line mode

#### **MMSS\_ROT\_SSPP\_SRC2\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address of plane2

#### **0xC9A8920 MMSS\_ROT\_SSPP\_SRC3\_ADDR**

**Type:** RW

**Reset State:** 0x00000000

Plane 3 base address

For compressed UBWC formats, this plane refers to the UV metadata plane for plane1

double buffered: YES

when: On frame boundaries (vsync or software trigger) when ADDR\_SYNC\_MODE is in frame mode and on line boundary if ADDR\_SYNC\_MODE in in line mode

**MMSS\_ROT\_SSPP\_SRC3\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address of plane3

**0x0C9A8924 MMSS\_ROT\_SSPP\_SRC\_YSTRIDE0****Type:** RW**Reset State:** 0x00000000

Stride Values

Double-buffered

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_YSTRIDE0**

Bits	Name	Description
31:16	SRCP1_YSTRIDE	Plane 1 Y stride in bytes
15:0	SRCP0_YSTRIDE	Plane 0 Y stride in bytes

**0x0C9A8928 MMSS\_ROT\_SSPP\_SRC\_YSTRIDE1****Type:** RW**Reset State:** 0x00000000

Stride Values

Double-buffered

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_YSTRIDE1**

Bits	Name	Description
31:16	SRCP3_YSTRIDE	Plane 3 Y stride in bytes
15:0	SRCP2_YSTRIDE	Plane 2 Y stride in bytes

**0x0C9A8930 MMSS\_ROT\_SSPP\_SRC\_FORMAT****Type:** RW**Reset State:** 0x00000000

Source Format Configuration for each virtual pipeline.

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_FORMAT**

Bits	Name	Description
31:30	FRAME_FORMAT	Frame format Note that cursor only supports linear formats 0x0: Linear, no tile 0x1: Macrotile A4x 0x2: Macrotile A5x 0x3: unused
25:23	SRC_CHROMA_SAMP	Source Chroma Sampling direction information 0x0: no subsampling; i.e. RGB/YUV 4:4:4 0x1: horizontal subsampled only; i.e. 4:2:2 - H2V1 0x2: vertical subsampled only; i.e. 4:2:2 - H1V2 0x3: both vertical and horizontal subsampled; i.e. 4:2:0 - N2V1
22	SOLID_FILL	Fetch feeds the lower pipeline with the CONSTANT_COLOR register as its output value. NOTHING is fetched Note that in order to use this feature then Interleaved FETCH_PLANES must be set as well and no chroma subsampling enabled (i.e. make it look like interleaved 32bpp)
21	VC1_REDUCE	VC1 range reduction enable ONLY valid for 8-bit YUV formats The reduce range is set by separate register VC1_RANGE and applied to pixel data regardless of source format
20:19	FETCH_PLANES	Determines the number of planes to fetch 0x0: Interleaved 0x1: Planar 0x2: Pseudo-planar
18	UNPACK_ALIGN	0x0: TO LSB 0x1: TO MSB
17	UNPACK_TIGHT	0x0: Loose 0x1: Tight
15	SRC_COLOR_SPACE	Source color space to distinguish formats Must be programmed independent of OP_MODE register 0x0: RGB 0x1: YUV

**MMSS\_ROT\_SSPP\_SRC\_FORMAT (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
14	UNPACK_DX_FORMAT	DX Format 0x0: Not DX Format 0x1: ARGB 2:10:10:10
13:12	UNPACK_COUNT	Valid unpacking pattern count 0x0 = 1 component 0x1 = 2 components ... 0x3 = 4 components Unpacking applies to interleaved plane (chroma plane in pseudo-planar)
11	ROT90	Source image is to be rotated 90 degrees
10:9	SRC_BPP	Effective source byte per pixel 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes This is used ONLY when unpacking the interleaved plane
8	SRCC3_EN	Alpha Enable 0x0: Source has NO alpha 0x1: Source has alpha
7:6	SRCC3	Number of bits for component 3 (Alpha ) 0x0: 1 bits 0x1: 4 bits 0x2: 6 bits 0x3: 8 bits
5:4	SRCC2	Number of bits for component 2 (R/Cr ) 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits
3:2	SRCC1	Number of bits for component 1 (B/Cb ) 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits
1:0	SRCC0	Number of bits for component 0 (G/Luma) 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits

**0x0C9A8934 MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN****Type:** RW**Reset State:** 0x00000000

Unpacking pattern, maximum of 4 pattern elements starting from LSB to MSB (0 to 4).

Default is ARGB or C3-C2-C0-C1.

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN**

Bits	Name	Description
25:24	ELEMENT3	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
17:16	ELEMENT2	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
9:8	ELEMENT1	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
1:0	ELEMENT0	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3

**0x0C9A8938 MMSS\_ROT\_SSPP\_SRC\_OP\_MODE****Type:** RW**Reset State:** 0x80000000

Operation Mode for each SSPP pipeline

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_SSPP\_SRC\_OP\_MODE**

Bits	Name	Description
31	SW_PIX_EXT_OVERRIDE	Software override for pixel extension values. 0x1: Use software programmed pixel extensions
30	SW_CROP_OVERRIDE	Software control for the crop values of the data read from the detile buffer Software is not required to enable this feature as all calculations are done in hardware. This register provides a backup and future flexibility to have software program these values 0x0: Hardware calculated crop values (recommended) 0x1: Software programmed crop values
25	SMOOTH_BUF_DISABLE	Smoothing buffer is disabled in non-scaling modes to save power. The smoothing buffer provides additional latency hiding to the source pipes and is enabled when scaling is performed in the source pipes. In non-scaling cases the smoothing buffer is disabled to save power consumption. Setting this bit to 1 provides an override to disable the automatic hardware disabling. Recommended value is to leave this value set to 0
24	HFLIP_BUF_DISABLE	The hflip buffer in the fetch unpack is required for horizontal flips and line repeats when scaling is enabled. It can also provide an additional line of latency hiding in non-hflip cases. In non hflip and non-scaling use cases the buffer is disabled by hardware to reduce power consumption. Setting this bit to 1 provides an override to disable the automatic hardware disabling. Recommended value is to leave this value set to 0
16	IGC_LUT_EN	Inverse Gamma Correction LUT Enable 0x0: Disabled - mapped Linearly from 8-bit to 12-bit 0x1: Enabled
14:13	FLIP_MODE	Flip Operation Note that hflip is NOT supported on cursor pipes as there is no flip buffer in the hardware Lower Bit: Left/Right horizontal flip Upper Bit: Up/Down vertical flip
0	BWC_DEC_EN	This is coupled with the source format to indicate that the source has been compressed and the bit stream needs to be decoded Note that there is no BWC formats supported on cursor pipes 0x0: NO Bandwidth de-compression 0x1: Bandwidth de-ompression is enabled

**0x0C9A893C MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR****Type:** RW**Reset State:** 0x00000000

Constant Color Register during solid fill mode

double buffered: NO

**MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR**

Bits	Name	Description
31:24	COLOR3	Constant color driven for color3 during solid fill mode.
23:16	COLOR2	Constant color driven for color2 during solid fill mode.
15:8	COLOR1	Constant color driven for color1 during solid fill mode.
7:0	COLOR0	Constant color driven for color0 during solid fill mode.

**0x0C9A8948 MMSS\_ROT\_SSPP\_FETCH\_CONFIG****Type:** RW**Reset State:** 0x00000087

SSPP Client Fetch Configuration per Virtual Pipeline double buffered: NO

This register must NOT be altered during operation and MDP must be idle when changing values

**MMSS\_ROT\_SSPP\_FETCH\_CONFIG**

Bits	Name	Description
20	ADDR_GEN_MODE	GPU A4x address generation mode mode 32 bpp formats 0x0: old mode 0x1: new mode
19:18	HIGHEST_BANK_BIT	GPU highest memory bank bit used This is a static setting for the system and should align with the GPU and memory support. This should not change during operation 0x0: 13 0x1: 14 0x2: 15 0x3: 16

**MMSS\_ROT\_SSPP\_FETCH\_CONFIG (cont.)**

Bits	Name	Description
17:16	BLOCK_SIZE	<p>Block Size for fetch and processing of BLOCK formats  Note that this only applies to blocks that operate in block mode which is limited to ROT source pipe. This setting has no effect for all other source pipes (ViG/RGB/DMA)</p> <p>If BLOCK_SIZE_EXT = 0x0 0x0: 128x128 (block size for all non-TP10 source formats)  0x1: 64x64 (block size for all non-TP10 source formats)  0x2: 96x96 (required setting for TP10 source format)  0x3: 48x48 (required setting for TP10 source format)  Others: Reserved / Not Supported</p> <p>If BLOCK_SIZE_EXT = 0x1 0x0: 192x192 (block size for all non-TP10 source formats)  0x1: 96x96 (block size for all non-TP10 source formats)  0x2: 144x144 (required setting for TP10 source format)  0x3: 72x72 (required setting for TP10 source format)  Others: Reserved / Not Supported</p>
15	BLOCK_SIZE_EXT	<p>Block Size Extention for fetch and processing of BLOCK formats  Note that this only applies to blocks that operate in block mode which is limited to ROT source pipe.</p> <p>This setting has no effect for all other source pipes (ViG/RGB/DMA)</p> <p>0x0 when ROTATOR DOWNSCALING FACTOR = 0/2/4/8/16/32/64  0x1 when ROTATOR DOWNSCALING FACTOR = 1.5</p>

**0x0C9A894C MMSS\_ROT\_SSPP\_VC1\_RANGE****Type:** RW**Reset State:** 0x000000707

VC1 Range Map for 8-bit YUV formats

not applicable for 10-bit formats

ONLY valid on ViG and DMA pipes and not applicable for RGB pipes

double buffered: NO

**MMSS\_ROT\_SSPP\_VC1\_RANGE**

Bits	Name	Description
10:8	RANGE_MAPUV	Chroma range map for VC1 range reduction, default of 7 is equivalent to setting it for main profile.
2:0	RANGE_MAPY	Luma range map for VC1 range reduction, default of 7 is equivalent to setting it for main profile.

**0x0C9A8960 MMSS\_ROT\_SSPP\_DANGER\_LUT****Type:** RW**Reset State:** 0x0055AAFF

LUT programming to generate danger level based on internal fill level

double buffered: YES

**MMSS\_ROT\_SSPP\_DANGER\_LUT**

Bits	Name	Description
31:30	FL_15	The danger signal generated when MDP is at Fill Level 15
29:28	FL_14	The danger signal generated when MDP is at Fill Level 14
27:26	FL_13	The danger signal generated when MDP is at Fill Level 13
25:24	FL_12	The danger signal generated when MDP is at Fill Level 12
23:22	FL_11	The danger signal generated when MDP is at Fill Level 11
21:20	FL_10	The danger signal generated when MDP is at Fill Level 10
19:18	FL_9	The danger signal generated when MDP is at Fill Level 9
17:16	FL_8	The danger signal generated when MDP is at Fill Level 8
15:14	FL_7	The danger signal generated when MDP is at Fill Level 7
13:12	FL_6	The danger signal generated when MDP is at Fill Level 6
11:10	FL_5	The danger signal generated when MDP is at Fill Level 5
9:8	FL_4	The danger signal generated when MDP is at Fill Level 4
7:6	FL_3	The danger signal generated when MDP is at Fill Level 3
5:4	FL_2	The danger signal generated when MDP is at Fill Level 2
3:2	FL_1	The danger signal generated when MDP is at Fill Level 1
1:0	FL_0	The danger signal generated when MDP is at Fill Level 0

**0x0C9A8964 MMSS\_ROT\_SSPP\_SAFE\_LUT****Type:** RW**Reset State:** 0x0000F000

LUT programming to generate safe level based on internal fill level

double buffered: YES

**MMSS\_ROT\_SSPP\_SAFE\_LUT**

Bits	Name	Description
15	FL_15	The safe signal generated when MDP is at Fill Level 15
14	FL_14	The safe signal generated when MDP is at Fill Level 14
13	FL_13	The safe signal generated when MDP is at Fill Level 13
12	FL_12	The safe signal generated when MDP is at Fill Level 12
11	FL_11	The safe signal generated when MDP is at Fill Level 11
10	FL_10	The safe signal generated when MDP is at Fill Level 10
9	FL_9	The safe signal generated when MDP is at Fill Level 9
8	FL_8	The safe signal generated when MDP is at Fill Level 8
7	FL_7	The safe signal generated when MDP is at Fill Level 7
6	FL_6	The safe signal generated when MDP is at Fill Level 6
5	FL_5	The safe signal generated when MDP is at Fill Level 5
4	FL_4	The safe signal generated when MDP is at Fill Level 4
3	FL_3	The safe signal generated when MDP is at Fill Level 3
2	FL_2	The safe signal generated when MDP is at Fill Level 2
1	FL_1	The safe signal generated when MDP is at Fill Level 1
0	FL_0	The safe signal generated when MDP is at Fill Level 0

**0x0C9A8968 MMSS\_ROT\_SSPP\_CREQ\_LUT****Type:** RW**Reset State:** 0x0055AAFF

LUT programming to generate creqpriority and creqprioritylvl signals to vbif based on internal fill level

double buffered: YES

**MMSS\_ROT\_SSPP\_CREQ\_LUT**

Bits	Name	Description
31:30	FL_15	The creq signal generated when MDP is at Fill Level 15
29:28	FL_14	The creq signal generated when MDP is at Fill Level 14
27:26	FL_13	The creq signal generated when MDP is at Fill Level 13
25:24	FL_12	The creq signal generated when MDP is at Fill Level 12
23:22	FL_11	The creq signal generated when MDP is at Fill Level 11

**MMSS\_ROT\_SSPP\_CREQ\_LUT (cont.)**

Bits	Name	Description
21:20	FL_10	The creq signal generated when MDP is at Fill Level 10
19:18	FL_9	The creq signal generated when MDP is at Fill Level 9
17:16	FL_8	The creq signal generated when MDP is at Fill Level 8
15:14	FL_7	The creq signal generated when MDP is at Fill Level 7
13:12	FL_6	The creq signal generated when MDP is at Fill Level 6
11:10	FL_5	The creq signal generated when MDP is at Fill Level 5
9:8	FL_4	The creq signal generated when MDP is at Fill Level 4
7:6	FL_3	The creq signal generated when MDP is at Fill Level 3
5:4	FL_2	The creq signal generated when MDP is at Fill Level 2
3:2	FL_1	The creq signal generated when MDP is at Fill Level 1
1:0	FL_0	The creq signal generated when MDP is at Fill Level 0

**0x0C9A896C MMSS\_ROT\_SSPP\_QOS\_CTRL****Type:** RW**Reset State:** 0x00000001

QOS control to enable danger safe generation and creqpriority generation to vbif

double buffered: YES

**MMSS\_ROT\_SSPP\_QOS\_CTRL**

Bits	Name	Description
21:20	CREQ_VBLANK	creq value generated to vbif during vertical blanking
16	VBLANK_EN	Enable mode to send CREQ_VBLANK as creq and DANGER_VBLANK as danger during vertical blanking Note that safe is not altered during vblank and sent as is
5:4	DANGER_VBLANK	danger value generated during vertical blanking
0	DANGER_SAFE_EN	Enable danger safe generation  If disabled the pipe will always indicate safe=1 and danger=0 regardless of internal buffer state. The creqpriority generation is not affected by this register programming

**0x0C9A8970 MMSS\_ROT\_SSPP\_SRC\_ADDR\_SW\_STATUS****Type:** RW**Reset State:** 0x00000000

Security Status of the source buffer plane

This is programmed by software to indicate if the buffer is NON-SECURE or SECURE

This information will be presented to VBIF per transaction and reflected as part of the memory transaction

NOTE that the entire surface is considered secure if any of the bits are set

The security status of each plane should be programmed to identical values since planes for a given surface would not be separated between secure and non-secure regions

double buffered: YES

**MMSS\_ROT\_SSPP\_SRC\_ADDR\_SW\_STATUS**

Bits	Name	Description
3	SRC3_ADDR_CP	<p>This is the secure status bit for source plane 3 (if used)  If the format does not fetch from that plane then this value is ignored  Memory transactions will not complete if this status does not correspond to system programming in the rest of the path to memory  Software should program SRC0_ADDR_CP, SRC1_ADDR_CP, SRC2_ADDR_CP, SRC3_ADDR_CP to the same value  0x0: Non Content Protected Buffer State  0x1: Content Protected Buffer State</p>
2	SRC2_ADDR_CP	<p>This is the secure status bit for source plane 2 (if used)  If the format does not fetch from that plane then this value is ignored  Memory transactions will not complete if this status does not correspond to system programming in the rest of the path to memory  Software should program SRC0_ADDR_CP, SRC1_ADDR_CP, SRC2_ADDR_CP, SRC3_ADDR_CP to the same value  0x0: Non Content Protected Buffer State  0x1: Content Protected Buffer State</p>

**MMSS\_ROT\_SSPP\_SRC\_ADDR\_SW\_STATUS (cont.)**

Bits	Name	Description
1	SRC1_ADDR_CP	<p>This is the secure status bit for source plane 1 (if used)  If the format does not fetch from that plane then this value is ignored  Memory transactions will not complete if this status does not correspond to system programming in the rest of the path to memory  Software should program SRC0_ADDR_CP, SRC1_ADDR_CP, SRC2_ADDR_CP, SRC3_ADDR_CP to the same value  0x0: Non Content Protected Buffer State  0x1: Content Protected Buffer State</p>
0	SRC0_ADDR_CP	<p>This is the secure status bit for source plane 0 (if used)  If the format does not fetch from that plane then this value is ignored  Memory transactions will not complete if this status does not correspond to system programming in the rest of the path to memory  Software should program SRC0_ADDR_CP, SRC1_ADDR_CP, SRC2_ADDR_CP, SRC3_ADDR_CP to the same value  0x0: Non Content Protected Buffer State  0x1: Content Protected Buffer State</p>

**0x0C9A89A4 MMSS\_ROT\_SSPP\_CURRENT\_SRC0\_ADDR****Type:** R**Reset State:** 0x00000000

Current Plane 0 base address that Hardware is using

double buffered: N/A

**MMSS\_ROT\_SSPP\_CURRENT\_SRC0\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address plane0

**0x0C9A89A8 MMSS\_ROT\_SSPP\_CURRENT\_SRC1\_ADDR****Type:** R**Reset State:** 0x00000000

Plane 1 base address that Hardware is using

double buffered: N/A

**MMSS\_ROT\_SSPP\_CURRENT\_SRC1\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address plane1

**0x0C9A89AC MMSS\_ROT\_SSPP\_CURRENT\_SRC2\_ADDR****Type:** R**Reset State:** 0x00000000

Plane 2 base address that Hardware is using

double buffered: N/A

**MMSS\_ROT\_SSPP\_CURRENT\_SRC2\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address plane2

**0x0C9A89B0 MMSS\_ROT\_SSPP\_CURRENT\_SRC3\_ADDR****Type:** R**Reset State:** 0x00000000

Plane 3 base address that Hardware is using

double buffered: N/A

**MMSS\_ROT\_SSPP\_CURRENT\_SRC3\_ADDR**

Bits	Name	Description
31:0	ADDR	Base byte address plane3

**0x0C9A89F0 MMSS\_ROT\_SSPP\_FILL\_LEVELS****Type:** R**Reset State:** 0x00000000

Detail Buffer fill levels values presented to fetch read control

double buffered: N/A

**MMSS\_ROT\_SSPP\_FILL\_LEVELS**

Bits	Name	Description
19:16	PLANE1	This is the fill level for plane 1 which represents the UV buffer. Note that in planar YUV mode, the fill level for plane1 is identical as plane2
3:0	PLANE0	This is the fill level for plane 0 which represents the RGB/Y buffer.

**0x0C9A89F4 MMSS\_ROT\_SSPP\_STATUS****Type:** R**Reset State:** 0x00000000

Status registers to indicate processing state of blocks in fetch unpack

double buffered: N/A

**MMSS\_ROT\_SSPP\_STATUS**

Bits	Name	Description
16	UBWC_IDLE	UBWC decoder idle signal 0= ubwc busy 1=ubwc idle
9	C12_UNPACK_REQ	color 1/2 request from downstream 0=no valid color 1=valid color
8	C03_UNPACK_REQ	color 0/3 request from downstream 0=no valid color 1=valid color
5	C12_UNPACK_VALID	color 1/2 valid to downstream 0=no valid color 1=valid color
4	C03_UNPACK_VALID	color 0/3 valid to downstream 0=no valid color 1=valid color
1	FETCH_IDLE	Fetch idle signal 0= fetch busy 1=fetch idle
0	UNPACK_IDLE	Unpack idle signal 0= unpack busy 1=unpack idle

**0x0C9A89F8 MMSS\_ROT\_SSPP\_UNPACK\_LINE\_COUNT****Type:** R**Reset State:** 0x00000000**MMSS\_ROT\_SSPP\_UNPACK\_LINE\_COUNT**

Bits	Name	Description
31:16	C1C2_COUNT	color 1/2 line count
15:0	C0C3_COUNT	color 0/3 line count

**0x0C9A89FC MMSS\_ROT\_SSPP\_UNPACK\_BLK\_COUNT****Type:** R**Reset State:** 0x00000000**MMSS\_ROT\_SSPP\_UNPACK\_BLK\_COUNT**

Bits	Name	Description
31:16	C1C2_COUNT	color 1/2 block count
15:0	C0C3_COUNT	color 0/3 block count

**0x0C9A8A00 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_LR****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register

double buffered: YES

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_LR**

Bits	Name	Description
31:24	RIGHT_OVF	Signed Value for Right Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
23:16	RIGHT_RPT	Unsigned Value for Right Pixel Repeat value.

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_LR (cont.)**

Bits	Name	Description
15:8	LEFT_OVF	Signed Value for Left Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
7:0	LEFT_RPT	Unsigned Value for Left Pixel Repeat value.

**0x0C9A8A04 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_TB****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register  
double buffered: YES**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_TB**

Bits	Name	Description
31:24	BOTTOM_OVF	Signed Value for Bottom Overfetch value. +ve value indicates overfetch -ve value indicates line drop
23:16	BOTTOM_RPT	Unsigned Value for Bottom Pixel Repeat value.
15:8	TOP_OVF	Signed Value for Top Overfetch value. +ve value indicates overfetch -ve value indicates line drop
7:0	TOP_RPT	Unsigned Value for TOP Line Repeat value.

**0x0C9A8A08 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_REQ\_PIXELS****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for pixel height/width of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register  
double buffered: YES

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_REQ\_PIXELS**

Bits	Name	Description
31:16	TOP_BOTTOM	Unsigned Value for pixel height post pixel extension.
15:0	LEFT_RIGHT	Unsigned Value for pixel width post pixel extension.

**0x0C9A8A10 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_LR****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register  
double buffered: YES**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_LR**

Bits	Name	Description
31:24	RIGHT_OVF	Signed Value for Right Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
23:16	RIGHT_RPT	Unsigned Value for Right Pixel Repeat value.
15:8	LEFT_OVF	Signed Value for Left Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
7:0	LEFT_RPT	Unsigned Value for Left Pixel Repeat value.

**0x0C9A8A14 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_TB****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register  
double buffered: YES

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_TB**

Bits	Name	Description
31:24	BOTTOM_OVF	Signed Value for Bottom Overfetch value. +ve value indicates overfetch -ve value indicates line drop
23:16	BOTTOM_RPT	Unsigned Value for Bottom Pixel Repeat value.
15:8	TOP_OVF	Signed Value for Top Overfetch value. +ve value indicates overfetch -ve value indicates line drop
7:0	TOP_RPT	Unsigned Value for TOP Line Repeat value.

**0x0C9A8A18 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_REQ\_PIXELS****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for pixel height/width of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register

double buffered: YES

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_REQ\_PIXELS**

Bits	Name	Description
31:16	TOP_BOTTOM	Unsigned Value for pixel height post pixel extension.
15:0	LEFT_RIGHT	Unsigned Value for pixel width post pixel extension.

**0x0C9A8A20 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_LR****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register

double buffered: YES

**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_LR**

Bits	Name	Description
31:24	RIGHT_OVF	Signed Value for Right Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
23:16	RIGHT_RPT	Unsigned Value for Right Pixel Repeat value.
15:8	LEFT_OVF	Signed Value for Left Overfetch value. +ve value indicates overfetch -ve value indicates pixel drop
7:0	LEFT_RPT	Unsigned Value for Left Pixel Repeat value.

**0x0C9A8A24 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_TB****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for Left/Right of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register  
double buffered: YES**MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_TB**

Bits	Name	Description
31:24	BOTTOM_OVF	Signed Value for Bottom Overfetch value. +ve value indicates overfetch -ve value indicates line drop
23:16	BOTTOM_RPT	Unsigned Value for Bottom Pixel Repeat value.
15:8	TOP_OVF	Signed Value for Top Overfetch value. +ve value indicates overfetch -ve value indicates line drop
7:0	TOP_RPT	Unsigned Value for TOP Line Repeat value.

**0x0C9A8A28 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_REQ\_PIXELS****Type:** RW**Reset State:** 0x00000000

Software Pixel Extension Override for pixel height/width of Color 0

These value only take effect if SW\_PIX\_EXT\_OVERRIDE is set in SRC\_OP\_MODE register

double buffered: YES

#### **MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_REQ\_PIXELS**

Bits	Name	Description
31:16	TOP_BOTTOM	Unsigned Value for pixel height post pixel extension.
15:0	LEFT_RIGHT	Unsigned Value for pixel width post pixel extension.

#### **0x0C9A8A30 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER**

**Type:** RW

**Reset State:** 0x00000000

Traffic Shaper Configuration for source pipe

double buffered: YES

#### **MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER**

Bits	Name	Description
31	EN	Traffic Shaper enable
27	PREFILL_MODE	Mode to use traffic shaper during prefill only 0x0=traffic shape always 0x1=traffic shape until prefill ends
7:0	BYTES_PER_CLOCK	Determines how many bytes each vsync_clk is equivalent to. This is an 8.0 byte value and is reference to a stable never changing clock A value of 0x01 implies 1 byte every 19.2 MHz pulse A value of 0xFF implies 255 bytes every 19.2 MHz pulse which is the largest programmable value

#### **0x0C9A8A34 MMSS\_ROT\_SSPP\_CDP\_CNTL**

**Type:** RW

**Reset State:** 0x0000000F

Client Driven Pre-Fetch Control

double buffered: YES

**MMSS\_ROT\_SSPP\_CDP\_CNTL**

Bits	Name	Description
3	PRELOAD_ADDR_AHEAD_CNT	address requests ahead control 0=32 requests ahead 1=64 requests ahead
2	TILE_MODE_AMORTIZE_EN	Amortization period control for A5x tiled formats 0=32/64 requests ahead 1=full amortization over tile rows
1	UBWC_META_CDP_EN	UBWC metadata address preload enable 0=disable address preload 1=enable address preload
0	CDP_EN	Client Driven prefetch enable 0=disable address preload 1=enable address preload

**0xC9A8A38 MMSS\_ROT\_SSPP\_UBWC\_ERROR\_STATUS****Type:** RW**Reset State:** 0x00000000

UBWC error status information

double buffered: N/A

**MMSS\_ROT\_SSPP\_UBWC\_ERROR\_STATUS**

Bits	Name	Type	Description
31	CODE_CLEAR	W	Write field to clear the error code capture register. Any write will clear the code and error count fields
23:16	ERROR_COUNT	R	Number of unique error codes since last clear. Clamps to all 1's to avoid rollover aliasing
7:0	CODE	R	Represents the first error received since last clear. See UBWC for complete error code definition but during normal decode errors are not expected.

**0xC9A8A40 MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C0C3****Type:** RW**Reset State:** 0x00000000

Software control to enable the horizontal crop module in unpack

Software is not required to enable this feature as all calculations are done in hardware.

This register provides a backup and future flexibility to have software program these values double buffered: YES

### **MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C0C3**

Bits	Name	Description
31:16	START_X	The location of the first valid pre-crop pixel
15:0	CROP_WIDTH	The output width of the color0/color3 component after cropping

### **0x0C9A8A44 MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C1C2**

**Type:** RW

**Reset State:** 0x00000000

Software control to enable the horizontal crop module in unpack

Software is not required to enable this feature as all calculations are done in hardware.

This register provides a backup and future flexibility to have software program these values double buffered: YES

### **MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C1C2**

Bits	Name	Description
31:16	START_X	The location of the first valid pre-crop pixel
15:0	CROP_WIDTH	The output width of the color1/color2 component after cropping

### **0x0C9A8A48 MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C0C3**

**Type:** RW

**Reset State:** 0x00000000

Software control to enable the vertical crop module in unpack

Software is not required to enable this feature as all calculations are done in hardware.

This register provides a backup and future flexibility to have software program these values double buffered: YES

### **MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C0C3**

Bits	Name	Description
31:16	START_Y	The location of the first valid Plane 0 pre-crop line

**0x0C9A8A4C MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C1C2****Type:** RW**Reset State:** 0x00000000

Software control to enable the vertical crop module in unpack

Software is not required to enable this feature as all calculations are done in hardware.

This register provides a backup and future flexibility to have software program these values  
double buffered: YES**MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C1C2**

Bits	Name	Description
31:16	START_Y	The location of the first valid Plane 1 pre-crop line

**0x0C9A8A50 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_PREFILL****Type:** RW**Reset State:** 0x00000000

Traffic shaper prefill

double buffered: YES

**MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_PREFILL**

Bits	Name	Description
31:0	COUNT	Prefill counter in # of 19.2MHz pulses

**0x0C9A8A54 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_REC1\_PREFILL****Type:** RW**Reset State:** 0x00000000

Traffic shaper prefill for rectangle 1 in time-multiplexed mode

double buffered: YES

**MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_REC1\_PREFILL**

Bits	Name	Description
31:0	COUNT	Prefill counter in # of 19.2MHz pulses

**0x0C9A8A60 MMSS\_ROT\_SSPP\_OUT\_SIZE\_REC1****Type:** RW**Reset State:** 0x00000000

Output Size of the layer assigned to the rectangle 1

double buffered: YES

**MMSS\_ROT\_SSPP\_OUT\_SIZE\_REC1**

Bits	Name	Description
31:16	REC1_DST_H	Height of Output ROI
15:0	REC1_DST_W	Width of Output ROI

**0x0C9A8A64 MMSS\_ROT\_SSPP\_OUT\_XY\_REC1****Type:** RW**Reset State:** 0x00000000

Output Offset relative to the top-left corner of the final output blended image

of the layer assigned to the rectangle 1

double buffered: YES

**MMSS\_ROT\_SSPP\_OUT\_XY\_REC1**

Bits	Name	Description
31:16	REC1_DST_Y	Height of Output ROI
15:0	REC1_DST_X	Width of Output ROI

**0x0C9A8A68 MMSS\_ROT\_SSPP\_SRC\_XY\_REC1****Type:** RW**Reset State:** 0x00000000

Source Offset relative to the top-left corner of the source image for each of the region of interest dimensions

of the layer assigned to the rectangle 1

double buffered: YES

**MMSS\_ROT\_SSPP\_SRC\_XY\_REC1**

Bits	Name	Description
31:16	REC1_SRC_Y	Y-offset of source ROI in the source image
15:0	REC1_SRC_X	X-offset of source ROI in the source image

**0x0C9A8A6C MMSS\_ROT\_SSPP\_SRC\_SIZE\_REC1****Type:** RW**Reset State:** 0x00000000

Source Size for the region of interest source image

of the layer assigned to the rectangle 1

double buffered: YES

**MMSS\_ROT\_SSPP\_SRC\_SIZE\_REC1**

Bits	Name	Description
31:16	REC1_ROI_H	Source Height of the Region Of Interest (ROI)
15:0	REC1_ROI_W	Source Width of the Region Of Interest (ROI)

**0x0C9A8A70 MMSS\_ROT\_SSPP\_MULTI\_REC\_OP\_MODE****Type:** RW**Reset State:** 0x00000000

Multi-rectangle mode configuration register

double buffered: YES

**MMSS\_ROT\_SSPP\_MULTI\_REC\_OP\_MODE**

Bits	Name	Description
2	MULTI_REC_MODE	0 = Multi-rectangle parallel fetch enabled. PD_MEM is logically divided into two halves 1 = Multi-rectangle time-multiplexed fetch. Each rectangle gets full access to PD_MEM
1:0	MULTI_REC_ENABLE	00 = Multi-rectangle not enabled 01 = REC0 is enabled 10 = REC1 is enabled 11 = REC0 and REC1 are both enabled

**0x0C9A8A74 MMSS\_ROT\_SSPP\_SRC\_FORMAT\_REC1****Type:** RW**Reset State:** 0x00000000

Source Format Configuration for rectangle 1.

Note that although separate format register control exists the only valid configuration is when rec0 format is equal to rec1 format

The hardware does NOT support different formats for each rectangle

double buffered: YES

**MMSS\_ROT\_SSPP\_SRC\_FORMAT\_REC1**

Bits	Name	Description
31:30	REC1_FRAME_FORMAT	Frame format In the current revision, there is no support for tiled formats 0x0: Linear, no tile 0x1: Macrotile A4x 0x2: Macrotile A5x 0x3: unused
21	REC1_VC1_REDUCE	VC1 range reduction enable ONLY valid on ViG and DMA pipes and not applicable for RGB pipes The reduce range is set by separate register VC1_RANGE and applied to pixel data regardless of source format
18	REC1_UNPACK_ALIGN	0x0: TO LSB 0x1: TO MSB
17	REC1_UNPACK_TIGHT	0x0: Loose 0x1: Tight
13:12	REC1_UNPACK_COUNT	Valid unpacking pattern count 0x0 = 1 component 0x1 = 2 components ... 0x3 = 4 components Unpacking applies to interleaved plane (chroma plane in pseudo-planar)
11	REC1_ROT90	Source image is to be rotated 90 degrees
10:9	REC1_SRC_BPP	Effective source byte per pixel 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes This is used ONLY when unpacking the interleaved plane

**0x0C9A8A78 MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN\_REC1****Type:** RW**Reset State:** 0x00000000

Unpacking pattern for rectangle 1.

Different unpack patterns per rectangle is supported but they must be the same format as programmed in the SRC\_FORMAT registers

double buffered: YES

**MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN\_REC1**

Bits	Name	Description
25:24	REC1_ELEMENT3	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
17:16	REC1_ELEMENT2	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
9:8	REC1_ELEMENT1	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
1:0	REC1_ELEMENT0	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3

**0x0C9A8A7C MMSS\_ROT\_SSPP\_SRC\_OP\_MODE\_REC1****Type:** RW**Reset State:** 0x80000000

Operation Mode for each SSPP pipeline for rectangle 1

Note that although separate op mode register control exists the only valid configuration is when rec0 format is equal to rec1 format

Both rectangles must have identical orientation

HFLIP is not supported in multi-rectangle configurations

double buffered: YES

when: On frame boundaries (vsync or software trigger)

#### **MMSS\_ROT\_SSPP\_SRC\_OP\_MODE\_REC1**

Bits	Name	Description
31	REC1_SW_PIX_EXT_OVER_RIDE	Software override for pixel extension values. 0x0: 0x1: Use software programmed pixel extensions
30	REC1_SW_CROP_OVERRIDE	Software control for the crop values of the data read from the detile buffer  Software is not required to enable this feature as all calculations are done in hardware.  This register provides a backup and future flexibility to have software program these values 0x0: Hardware calculated crop values (recommended) 0x1: Software programmed crop values
25	REC1_SMOOTH_BUF_DISABLE	Smoothing buffer is disabled in non-scaling modes to save power. The smoothing buffer provides additional latency hiding to the source pipes and is enabled when scaling is performed in the source pipes. In non-scaling cases the smoothing buffer is disabled to save power consumption.  Setting this bit to 1 provides an override to disable the automatic hardware disabling. Recommended value is to leave this value set to 0
24	REC1_HFLIP_BUF_DISABLE	The hflip buffer in the fetch unpack is required for horizontal flips and line repeats when scaling is enabled.  It can also provide an additional line of latency hiding in non-hflip cases.  In non hflip and non-scaling use cases the buffer is disabled by hardware to reduce power consumption.  Setting this bit to 1 provides an override to disable the automatic hardware disabling. Recommended value is to leave this value set to 0
16	REC1_IGC_LUT_EN	Inverse Gamma Correction LUT Enable 0x0: Disabled - mapped Linearly from 8-bit to 12-bit 0x1: Enabled
14:13	REC1_FLIP_MODE	Flip Operation Lower Bit: Left/Right flip Upper Bit: Up/Down flip
0	REC1_BWC_DEC_EN	This is coupled with the source format to indicate that the source has been compressed and the bit stream needs to be decoded 0x0: NO Bandwidth de-compression 0x1: Bandwidth de-ompression is enabled

**0x0C9A8A80 MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR\_REC1****Type:** RW**Reset State:** 0x00000000

Constant Color Register during solid fill mode for rectangle 1 used only in multirectangle mode  
double buffered: NO

**MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR\_REC1**

Bits	Name	Description
31:24	REC1_COLOR3	Constant color driven for color3 during solid fill mode.
23:16	REC1_COLOR2	Constant color driven for color2 during solid fill mode.
15:8	REC1_COLOR1	Constant color driven for color1 during solid fill mode.
7:0	REC1_COLOR0	Constant color driven for color0 during solid fill mode.

**0x0C9A8B00 MMSS\_ROT\_WB\_DST\_FORMAT****Type:** RW**Reset State:** 0x00000000

Memory Write Back Output Format

double buffered: YES

**MMSS\_ROT\_WB\_DST\_FORMAT**

Bits	Name	Description
31:30	FRAME_FORMAT	FRAME_FORMAT 0x0: Linear mode 0x1: Tiled mode (A4x) 0x2: Tiled mode (A5x)
25:23	DST_CHROMA_SAMP	Chroma Sampling 0x0: RGB/4:4:4 0x1: 4:2:2 - H2V1 0x2: 4:2:2 - H1V2 0x3: 4:2:0 - N2V1 0x4: 4:2:0 - N1V2
21	PACK_DX_FORMAT	DX Format 0x0: Not DX Format 0x1: ARGB 2:10:10:10

**MMSS\_ROT\_WB\_DST\_FORMAT (cont.)**

Bits	Name	Description
20:19	WRITE_PLANES	Number of planes 0x0: Interleaved 0x1: Planar 0x2: Pseudo-planar
18	PACK_ALIGN	0x0: LSB 0x1: MSB
17	PACK_TIGHT	0x0: Loose 0x1: Tight
15	DST_COLOR_SPACE	Source color space to distinguish formats Must be programmed independent of OP_MODE register 0x0: RGB 0x1: YUV
14	DST_ALPHA_X	0x0: Use alpha value from pipe 0x1: Use alpha value from register
13:12	PACK_COUNT	Valid packing elements count 0x0 = 1 element 0x1 = 2 elements 0x2 = 3 elements 0x3 = 4 elements
10:9	DST_BPP	Effective byte per pixel 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes
8	DSTC3_EN	0x1: Destination has alpha
7:6	DSTC3_OUT	Number of bits for component 3 output 0x0: 1 bit 0x1: 4 bits 0x2: 6 bits 0x3: 8 bits
5:4	DSTC2_OUT	Number of bits for component 2 output 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits
3:2	DSTC1_OUT	Number of bits for component 1 output 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits

**MMSS\_ROT\_WB\_DST\_FORMAT (cont.)**

Bits	Name	Description
1:0	DSTC0_OUT	Number of bits for component 0 output 0x0: 4 bits 0x1: 5 bits 0x2: 6 bits 0x3: 8 bits

**0x0C9A8B04 MMSS\_ROT\_WB\_DST\_OP\_MODE****Type:** RW**Reset State:** 0x00000000

Destination Operation Mode

double buffered: YES

**MMSS\_ROT\_WB\_DST\_OP\_MODE**

Bits	Name	Description
0	UBWC_ENC_EN	0x0: No Bandwidth compression 0x1: Bandwidth Compression is enabled

**0x0C9A8B08 MMSS\_ROT\_WB\_DST\_PACK\_PATTERN****Type:** RW**Reset State:** 0x00000000

Write Back Pack Pattern

double buffered: YES

**MMSS\_ROT\_WB\_DST\_PACK\_PATTERN**

Bits	Name	Description
25:24	ELEMENT3	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
17:16	ELEMENT2	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3

**MMSS\_ROT\_WB\_DST\_PACK\_PATTERN (cont.)**

Bits	Name	Description
9:8	ELEMENT1	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3
1:0	ELEMENT0	0x0: Color0 0x1: Color1 0x2: Color2 0x3: Color3

**0x0C9A8B0C MMSS\_ROT\_WB\_DST0\_ADDR****Type:** RW**Reset State:** 0x00000000

Write Back Destination Address

For linear formats, this plane refers to ARGB/Y plane

For compressed UBWC formats, this plane refers to the compressed data plane

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_WB\_DST0\_ADDR**

Bits	Name	Description
31:0	ADDR	Destination Address (byte-aligned)

**0x0C9A8B10 MMSS\_ROT\_WB\_DST1\_ADDR****Type:** RW**Reset State:** 0x00000000

Write Back Destination Address

For linear formats, this plane refers to UV or U plane

For compressed UBWC formats, this plane refers to the compressed UV data plane

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_WB\_DST1\_ADDR**

Bits	Name	Description
31:0	ADDR	Destination Address (byte-aligned)

**0x0C9A8B14 MMSS\_ROT\_WB\_DST2\_ADDR****Type:** RW**Reset State:** 0x00000000

Write Back Destination Address

For linear formats, this plane refers to V plane

For compressed UBWC formats, this plane refers to the Y/RGB metadata plane for plane0

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_WB\_DST2\_ADDR**

Bits	Name	Description
31:0	ADDR	Destination Address (byte-aligned)

**0x0C9A8B18 MMSS\_ROT\_WB\_DST3\_ADDR****Type:** RW**Reset State:** 0x00000000

Write Back Destination Address

For compressed UBWC formats, this plane refers to the UV metadata plane for plane1

double buffered: YES

when: On frame boundaries (vsync or software trigger)

**MMSS\_ROT\_WB\_DST3\_ADDR**

Bits	Name	Description
31:0	ADDR	Destination Address (byte-aligned)

**0x0C9A8B1C MMSS\_ROT\_WB\_DST\_YSTRIDE0****Type:** RW**Reset State:** 0x00000000

Stride Values

Double-buffered

double buffered: YES

**MMSS\_ROT\_WB\_DST\_YSTRIDE0**

Bits	Name	Description
31:16	DST1_YSTRIDE	Plane 1 Y stride in bytes
15:0	DST0_YSTRIDE	Plane 0 Y stride in bytes

**0x0C9A8B20 MMSS\_ROT\_WB\_DST\_YSTRIDE1****Type:** RW**Reset State:** 0x00000000

Stride Values

Double-buffered

double buffered: YES

**MMSS\_ROT\_WB\_DST\_YSTRIDE1**

Bits	Name	Description
31:16	DST3_YSTRIDE	Plane 3 Y stride in bytes
15:0	DST2_YSTRIDE	Plane 2 Y stride in bytes

**0x0C9A8B40 MMSS\_ROT\_WB\_TRAFFIC\_SHAPER\_WR\_CLIENT****Type:** RW**Reset State:** 0x00000000

Traffic Shaper Configuration for write client0

This is mapped to WB0 write client and not tied to a panel

double buffered: YES

**MMSS\_ROT\_WB\_TRAFFIC\_SHAPER\_WR\_CLIENT**

Bits	Name	Description
31	EN	Traffic Shaper enable
7:0	BYTES_PER_CLOCK	Determines how many bytes each vsync_clk is equivalent to. This is an 8.0 byte value and is reference to a stable never changing clock A value of 0x01 implies 1 byte every 19.2 MHz pulse A value of 0xFF implies 255 bytes every 19.2 MHz pulse which is the largest programmable value

**0x0C9A8B48 MMSS\_ROT\_WB\_DST\_WRITE\_CONFIG****Type:** RW**Reset State:** 0x00000000

Write-back Configuration double buffered: YES

This register must NOT be altered during operation and MDP must be idle when changing values

**MMSS\_ROT\_WB\_DST\_WRITE\_CONFIG**

Bits	Name	Description
10	ADDR_GEN_MODE	GPU address generation mode (A4x) 0x0: old mode; i.e. A420 mode 0x1: new mode; i.e. A430 mode
9:8	HIGHEST_BANK_BIT	GPU highest memory bank bit used This is a static setting for the system and should align with the GPU and memory support. This should not change during operation 0x0: 13 0x1: 14 0x2: 15 0x3: 16
7:6	PRIORITY_LEVEL	Priority Level. Sideband priority reflecting the status of VBIF client. 0x3 - max priority 0x0 - min priority
5:4	REQ_PRIORITY	Request Priority. In-band priority level assigned when VBIF transaction is created. 0x3 - max priority 0x0 - min priority

**0x0C9A8B74 MMSS\_ROT\_WB\_OUT\_SIZE****Type:** RW**Reset State:** 0x00000000

Output Size

Should be same as source is scaling is NOT enabled

double buffered: YES

**MMSS\_ROT\_WB\_OUT\_SIZE**

Bits	Name	Description
31:16	DST_H	Height of Output ROI
15:0	DST_W	Width of Output ROI

**0x0C9A8B78 MMSS\_ROT\_WB\_DST\_ALPHA\_X\_VALUE****Type:** RW**Reset State:** 0x00000000

Alpha value used for XRGB formats

double buffered: YES

**MMSS\_ROT\_WB\_DST\_ALPHA\_X\_VALUE**

Bits	Name	Description
7:0	VALUE	Alpha value used for XRGB formats

**0x0C9A8DB0 MMSS\_ROT\_WB\_DST\_ADDR\_SW\_STATUS****Type:** RW**Reset State:** 0x00000000

Security Status of the destination buffer plane

This provides a software override to the hardware calculated content protection sent to VBIF

This value is logically or ed with the source content protection produced by hardware

This information will be presented to VBIF per transaction and reflected as part of the memory transaction

Note that the VBIF context banks need to be properly programmed for the output to be successful

double buffered: YES

**MMSS\_ROT\_WB\_DST\_ADDR\_SW\_STATUS**

Bits	Name	Description
0	DST_ADDR_CP	This is the secure status bit for destination plane 0 (if used) Memory transactions will not complete if this status does not correspond to system programming in the rest of the path to memory 0x0: Non Content Protected Buffer State 0x1: Content Protected Buffer State

**0xC9A8DB4 MMSS\_ROT\_WB\_CDP\_CNTL****Type:** RW**Reset State:** 0x0000000B

Client Driven Pre-Fetch Control

double buffered: YES

**MMSS\_ROT\_WB\_CDP\_CNTL**

Bits	Name	Description
3	PRELOAD_ADDR_AHEAD_CNT	Data/metadata address preload ahead count 0=32 requests ahead 1=64 requests ahead
1	UBWC_META_CDP_EN	Metadata address preload enable 0=disable address preload 1=enable address preload
0	CDP_EN	Data address preload enable 0=disable address preload 1=enable address preload

**0xC9A8DB8 MMSS\_ROT\_WB\_STATUS****Type:** R**Reset State:** 0x00000000

Status register to indicate processing state of blocks in write-back

double buffered: NO

**MMSS\_ROT\_WB\_STATUS**

Bits	Name	Description
0	UBWC_IDLE	The external UBWC encoder is IDLE.

**0x0C9A8DBC MMSS\_ROT\_WB\_UBWC\_ERROR\_STATUS****Type:** RW**Reset State:** 0x00000000

UBWC error status information

double buffered: NO

**MMSS\_ROT\_WB\_UBWC\_ERROR\_STATUS**

Bits	Name	Type	Description
31	CODE_CLEAR	W	Write field to clear the error code capture register. Any write will clear the code and error count fields
23:16	ERROR_COUNT	R	Number of unique error codes since last clear. Clamps to all 1's to avoid rollover aliasing
7:0	CODE	R	Represents the first error received since last clear. See UBWC for complete error code definition but during normal decode errors are not expected.

**0x0C9A8DC0 MMSS\_ROT\_WB\_OUT\_IMG\_SIZE****Type:** RW**Reset State:** 0x00000000

Output Size for the image in which the ROI resides

double buffered: YES

**MMSS\_ROT\_WB\_OUT\_IMG\_SIZE**

Bits	Name	Description
31:16	IMG_H	Image Height. Note that this is NOT the ROI height but refers to the surface image.
15:0	IMG_W	Image Width. Note that this is NOT the ROI width but refers to the surface image.

**0x0C9A8DC4 MMSS\_ROT\_WB\_OUT\_XY****Type:** RW**Reset State:** 0x00000000

Output offset relative to the top-left corner of the destination image for each of the region of interest dimensions

double buffered: YES

**MMSS\_ROT\_WB\_OUT\_XY**

Bits	Name	Description
31:16	DST_Y	Y-offset of out ROI in the destination image
15:0	DST_X	X-offset of out ROI in the destination image

**0x0C9A8E00 MMSS\_ROT\_REGDMA\_RAM\_REGDMA\_CMD\_RAM\_i, i=[0..2047]  
+4\*i**

Type: RW

Reset State: 0x00000000

**MMSS\_ROT\_REGDMA\_RAM\_REGDMA\_CMD\_RAM\_i**

Bits	Name	Description
31:0	REGDMA_CMD_DATA	

**0x0C9AAE04 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_OP\_MODE**

Type: RW

Reset State: 0x00000000

MDP REGDMA operation mode

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_OP\_MODE**

Bits	Name	Description
16	HALT_EN	enable/disable REGDMA HALT function
8	SECURE_EN	enable/disable REGDMA SECURE function
0	REGDMA_EN	enable/disable REGDMA

**0x0C9AAE10 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_SUBMIT**

Type: RW

Reset State: 0x00000000

Queue 0 command descriptor

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_SUBMIT**

Bits	Name	Description
31:30	INT_EN	0x0: no interrupt 0x1: INT_0 interrupt 0x2: INT_1 interrupt 0x3: INT_2 interrupt
29:27	TRIG_SEL	command trigger selection: only support 0x0 immediate trigger
23:14	CMD_LENGTH	command length: Dword aligned
13:0	CMD_OFFSET	command offset in command RAM: Dword aligned

**0x0C9AAE14 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_STATUS****Type:** R**Reset State:** 0x000000500

Queue 0 status

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_STATUS**

Bits	Name	Description
11:8	STATUS	command queue fifo: almost_full, full, almost_empty, empty
5:0	FULLNESS	command queue occupy level

**0x0C9AAE18 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_SUBMIT****Type:** RW**Reset State:** 0x000000000

Queue 1 command descriptor

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_SUBMIT**

Bits	Name	Description
31:30	INT_EN	0x0: no interrupt 0x1: INT_0 interrupt 0x2: INT_1 interrupt 0x3: INT_2 interrupt
29:27	TRIG_SEL	command trigger selection: only support 0x0 immediate trigger
23:14	CMD_LENGTH	command length: Dword aligned
13:0	CMD_OFFSET	command offset in command RAM: Dword aligned

**0x0C9AAE1C MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_STATUS****Type:** R**Reset State:** 0x00000500

Queue 1 status

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_STATUS**

Bits	Name	Description
11:8	STATUS	command queue fifo almost_full, full, almost_empty, empty
5:0	FULLNESS	command queue occupy level

**0x0C9AAE20 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_0****Type:** RW**Reset State:** 0x00000000

Secure block region 0 low address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_0**

Bits	Name	Description
19:0	BLOCK_ADDRESS_LO	region 0 low address

**0x0C9AAE24 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_0****Type:** RW**Reset State:** 0x00000000

Secure block region 0 high address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_0**

Bits	Name	Description
19:0	BLOCK_ADDRESS_HI	region 0 high address

**0x0C9AAE28 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_1****Type:** RW**Reset State:** 0x00000000

Secure block region 1 low address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_1**

Bits	Name	Description
19:0	BLOCK_ADDRESS_LO	region 1 low address

**0x0C9AAE2C MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_1****Type:** RW**Reset State:** 0x00000000

Secure block region 1 high address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_1**

Bits	Name	Description
19:0	BLOCK_ADDRESS_HI	region 1 high address

**0x0C9AAE30 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_2****Type:** RW**Reset State:** 0x00000000

Secure block region 2 low address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_2**

Bits	Name	Description
19:0	BLOCK_ADDRESS_LO	region 2 low address

**0x0C9AAE34 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_2****Type:** RW**Reset State:** 0x00000000

Secure block region 2 high address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_2**

Bits	Name	Description
19:0	BLOCK_ADDRESS_HI	region 2 high address

**0x0C9AAE38 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_3****Type:** RW**Reset State:** 0x00000000

Secure block region 3 low address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_3**

Bits	Name	Description
19:0	BLOCK_ADDRESS_LO	region 3 low address

**0x0C9AAE3C MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_3****Type:** RW**Reset State:** 0x00000000

Secure block region 3 high address

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_3**

Bits	Name	Description
19:0	BLOCK_ADDRESS_HI	region 3 high address

**0x0C9AAE40 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL****Type:** W**Reset State:** 0x00000000

This register is used to control Watchdog Timer in regdma

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL**

Bits	Name	Description
0	REGDMA_WD_TIMER_CLE_AR	Clear the timer

**0x0C9AAE44 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL2****Type:** RW**Reset State:** 0x00000002

This register is used to control Watchdog Timer in regdma

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL2**

Bits	Name	Description
8	WD_TIMER_HEARTBEAT_ENABLE	Enable timer heartbeat
4:1	WD_TIMER_GRANULARITY	Configurable # of clock ticks granularity per main counter increment 0x0: 8 clock ticks 0x1: 16 clock ticks (default) 0x2: 24 clock ticks 0x3: 32 clock ticks 0x4: 40 clock ticks 0x5: 48 clock ticks 0x6: 56 clock ticks 0x7: 64 clock ticks 0x8: 72 clock ticks 0x9: 80 clock ticks 0xa: 88 clock ticks 0xb: 96 clock ticks 0xc: 104 clock ticks 0xd: 112 clock ticks 0xe: 120 clock ticks 0xf: 128 clock ticks
0	WD_TIMER_ENABLE	Enable the timer

**0x0C9AAE48 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_LOAD\_VALUE****Type:** RW**Reset State:** 0x00000000

The timer count up to value that would trigger WD interrupt

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_LOAD\_VALUE**

Bits	Name	Description
31:0	REGDMA_WD_TIMER_LOAD_VALUE	Timer count value 0x0 : Free running mode, timer counter wraps Others: Timer count up to value in granularity of 16 clock tick periods eg. 0x2 : Translates into $2 \times 16 = 32$ clock tick periods

**0x0C9AAE4C MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_STATUS\_VALUE****Type:** R**Reset State:** 0x00000000

The current timer count value

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_STATUS\_VALUE**

Bits	Name	Description
31:0	REGDMA_WD_TIMER_STATUS_VALUE	Timer count value in granularity of 16 clock tick periods. Timer counts up eg. 0x2 : Translates into 2*16=32 clock tick periods

**0x0C9AAE50 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_STATUS****Type:** R**Reset State:** 0x00000000

regdma interrupts status

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_STATUS**

Bits	Name	Description
19	REGDMA_WATCHDOG_INT	regdma watchdog interrupt status
18	REGDMA_INVALID_DESCRIPTOR	invalid command descriptor status
17	REGDMA_INCOMPLETE_CMD	incomplete opcode status
16	REGDMA_INVALID_CMD	invalid opcode status
10	REGDMA_DONE_QUEUE_1_INT2	queue 1 command INT2 status
9	REGDMA_DONE_QUEUE_1_INT1	queue 1 command INT1 status
8	REGDMA_DONE_QUEUE_1_INT0	queue 1 command INT0 status
2	REGDMA_DONE_QUEUE_0_INT2	queue 0 command INT2 status
1	REGDMA_DONE_QUEUE_0_INT1	queue 0 command INT1 status
0	REGDMA_DONE_QUEUE_0_INT0	queue 0 command INT0 status

**0x0C9AAE54 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_EN****Type:** RW**Reset State:** 0x00000000

regdma interrupts enable

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_EN**

Bits	Name	Description
19	REGDMA_WATCHDOG_INT_EN	enable/disable regdma watchdog interrupt
18	REGDMA_INVALID_DESCRIPTOR_EN	enable/disable invalid command descriptor
17	REGDMA_INCOMPLETE_CMD_EN	enable/disable incomplete opcode
16	REGDMA_INVALID_CMD_EN	enable/disable invalid opcode
10	REGDMA_DONE_QUEUE_1_INT2_EN	enable/disable queue 1 command INT2
9	REGDMA_DONE_QUEUE_1_INT1_EN	enable/disable queue 1 command INT1
8	REGDMA_DONE_QUEUE_1_INT0_EN	enable/disable queue 1 command INT0
2	REGDMA_DONE_QUEUE_0_INT2_EN	enable/disable queue 0 command INT2
1	REGDMA_DONE_QUEUE_0_INT1_EN	enable/disable queue 0 command INT1
0	REGDMA_DONE_QUEUE_0_INT0_EN	enable/disable queue 0 command INT0

**0x0C9AAE58 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_CLEAR****Type:** W**Reset State:** 0x00000000

clear dregdma interrupts

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_CLEAR**

Bits	Name	Description
19	REGDMA_WATCHDOG_INT_CLEAR	clear regdma watchdog interrupt

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_CLEAR (cont.)**

Bits	Name	Description
18	REGDMA_INVALID_DESCRIPTOR_CLEAR	clear invalid command descriptor
17	REGDMA_INCOMPLETE_COMMAND_CLEAR	clear incomplete opcode
16	REGDMA_INVALID_CMD_CLEAR	clear invalid opcode
10	REGDMA_DONE_QUEUE_1_INT2_CLEAR	clear queue 1 command INT2
9	REGDMA_DONE_QUEUE_1_INT1_CLEAR	clear queue 1 command INT1
8	REGDMA_DONE_QUEUE_1_INT0_CLEAR	clear queue 1 command INT0
2	REGDMA_DONE_QUEUE_0_INT2_CLEAR	clear queue 0 command INT2
1	REGDMA_DONE_QUEUE_0_INT1_CLEAR	clear queue 0 command INT1
0	REGDMA_DONE_QUEUE_0_INT0_CLEAR	clear queue 0 command INT0

**0x0C9AAE5C MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_STATUS****Type:** R**Reset State:** 0x00000000

regdma busy status

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_STATUS**

Bits	Name	Description
0	REGDMA_BUSY	regdma is busy

**0x0C9AAE60 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INVALID\_CMD\_RAM\_OFFSET****Type:** R**Reset State:** 0x00000000

RAM offset for invalid command descriptor or opcode

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INVALID\_CMD\_RAM\_OFFSET**

Bits	Name	Description
11:0	INVALID_CMD_RAM_OFFSET	RAM offset for invalid command descriptor or opcode

**0x0C9AAE64 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_FSM\_STATE****Type:** R**Reset State:** 0x00000000

regdma internal state machine

**MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_FSM\_STATE**

Bits	Name	Description
27:24	REGDMA_CSR_ARBITER_STATE	host and regdma arbiter state
18:16	REGDMA_COMMAND_EXECUTION_STATE	opcode execution state
11:8	REGDMA_COMMAND_DECODE_STATE	opcode decode state
2:0	REGDMA_DESCRIPTOR_ARBITER_STATE	command descriptor arbiter state

**0x0C9AAF04 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE0****Type:** RW**Reset State:** 0x00000000

QDSS ATB Bus Data Selection for data\_id 31:0

**MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE0**

Bits	Name	Description
31:0	ID_EN	Specify profile data enable for data_id 31:0 - little-endian

**0x0C9AAF08 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE1****Type:** RW**Reset State:** 0x00000000

QDSS ATB Bus Data Selection for data\_id 63:32

**MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE1**

Bits	Name	Description
31:0	ID_EN	Specify profile data enable for data_id 63:32 - little-endian

**0x0C9AAF0C MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE2****Type:** RW**Reset State:** 0x00000000

QDSS ATB Bus Data Selection for data\_id 95:64

**MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE2**

Bits	Name	Description
31:0	ID_EN	Specify profile data enable for data_id 95:64 - little-endian

**0x0C9AAF10 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE3****Type:** RW**Reset State:** 0x00000000

QDSS ATB Bus Data Selection for data\_id 127:96

**MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE3**

Bits	Name	Description
31:0	ID_EN	Specify profile data enable for data_id 127:96 - little-endian

**0x0C9AAF14 MMSS\_ROT\_QDSS\_QDSS\_CLK\_CTRL****Type:** RW**Reset State:** 0x00000004

Clock Gating Control

**MMSS\_ROT\_QDSS\_QDSS\_CLK\_CTRL**

Bits	Name	Description
3:2	QDSS_CLK_HYSTESIS_C_TRL	0x0 : Clock will run for extra 4 cycles before is gated OFF 0x1 : Clock will run for extra 8 cycles before is gated OFF 0x2 : Clock will run for extra 16 cycles before is gated OFF 0x3 : Clock will run for extra 32 cycles before is gated OFF

**MMSS\_ROT\_QDSS\_QDSS\_CLK\_CTRL (cont.)**

Bits	Name	Description
1	QDSS_CLK_FORCE_OFF	0x0 : Clock gating is active if CLK_FORCE_ON=0 and CLK_FORCE_OFF=0 0x1 : Force clock branch OFF if CLK_FORCE_ON=0, otherwise setting is ignored
0	QDSS_CLK_FORCE_ON	0x0 : Clock gating is active if CLK_FORCE_ON=0 and CLK_FORCE_OFF=0 0x1 : Force clock branch ON

**0x0C9AAF18 MMSS\_ROT\_QDSS\_QDSS\_CLK\_STATUS****Type:** R**Reset State:** 0x00000000

Status of gated clock branches

**MMSS\_ROT\_QDSS\_QDSS\_CLK\_STATUS**

Bits	Name	Description
0	QDSS_CLK_ACTIVE	QDSS clock

**0x0C9AAF20 MMSS\_ROT\_QDSS\_QDSS\_PULSE\_TRIGGER****Type:** W**Reset State:** 0x00000000

Write trigger to initiate MDP transfer of QDSS data. Analogous to spdm\_pulse\_trigger signal.

QDSS\_CONFIG.TRIGGER\_LEVEL bit must be set to 0x1 in order to use this trigger

**MMSS\_ROT\_QDSS\_QDSS\_PULSE\_TRIGGER**

Bits	Name	Description
0	ENABLE	Write trigger Enable

# 6 Serial peripheral master interface registers

**0x0800A700+ SPMI\_PERIPHm\_2OWNER\_TABLE\_REG, m=[0..511]**  
**0x4\*m**

**Type:** RW  
**Clock:** AHB\_CLK  
**Reset State:** 0x00000000

Peripheral 2 owner Table; Each Peripheral has 4 bits indicating its associated owner. The APID2PPID is used verify that each mapping table request is given the correct APID, APID2PPID not relevant when RPU is removed (versions 2.1.0 and above).

## SPMI\_PERIPHm\_2OWNER\_TABLE\_REG

Bits	Name	Type	Description
27:16	APID2PPID	R	APID to PPID mapping
2:0	PERIPH2OWNER	RW	Owner index for peripheral m.

**0x0800AF00+ SPMI\_MAPPING\_TABLE\_REGk, k=[0..510]**  
**0x4\*k**

**Type:** RW  
**Clock:** AHB\_CLK  
**Reset State:** 0x00000000

Mapping table register; this table is used for conversion of the 12-bit physical peripheral ID (4-bit SID + 8 high address bits, PPID) to the 9-bit application peripheral ID (APID).

## SPMI\_MAPPING\_TABLE\_REGk

Bits	Name	Description
23:20	BIT_INDEX	Mapping logic table contents.
19	BRANCH_RESULT_0_FLAG	Mapping logic table contents.
18:10	BRANCH_RESULT_0	Mapping logic table contents.
9	BRANCH_RESULT_1_FLAG	Mapping logic table contents.
8:0	BRANCH_RESULT_1	Mapping logic table contents.

**0x0800B700 SPMI\_MID\_REG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

This register defines the MID (master ID) of the SPMI master on the bus.

**SPMI\_MID\_REG**

Bits	Name	Description
1:0	MID	This register contains the SPMI master MID.

**0x0800B704 SPMI\_CFG\_REG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000020

This register holds different configuration of the SPMI-controller.

**SPMI\_CFG\_REG**

Bits	Name	Description
16:13	ARBITER_CTRL	Reserved bits for arbiter control need.
12	GENI_SPMI_BUS_IN_MASK_DIS	when '0'(default), Geni sdata_in and sclk_in interfaces will be enabled only during the time Geni is operative or in BYPASS mode; else ('1') Geni 'see' SPMI bus_in all the time (needed for protecting CX during sleep/collapse power modes)
11	ASYNC_SSC_DET_INT_DIS	when '0'(default) mean the spmi_m async_ssc_detect use the internal indication for SSC, this can be used also for multi master (unlike ssc_detect).
10	SSC_Q1_DELAY_DIS	when set, disable the delay that placed between Q1 to Q2 regs in the SSC_DETECT
9	ARB_UNKNOWN_EN	when '0'(default) mean that if ssc_window_en='1', ssc will be detect only when master_win slave_win(suitable for single master) if it set to '1'ssc will be detected when master_win slave_win arb_unknown(suitable for multi master). If ssc_window_en='0' arb_unknown_en is dont_care.
8	SSC_WINDOW_EN	when set, SSC is detected only in specific window when it is expected in order to reduce glitch risk
7	SSC_DET_INT_DIS	when '0'(default) mean the spmi_m use the internal indication for SSC (suitable for single master, when set use the external bus for ssc detection suitable for multi master).
6	FORCE_MPM_CLK_REQ_IMM	When set request clock for configuration, to enable transferring the values to the ser_clock domain

**SPMI\_CFG\_REG (cont.)**

Bits	Name	Description
5	FORCE_ARB_AFTER_MASTER_TO	When set, BOM starts an arbitration after false master arbitration (ire. SSC hasn't been asserted within 64 us after an arbitration win).
4	BUS_IDLE_CONN_MODE	When asserted, sdata is deasserted on a rising edge when connecting on bus idle. When deasserted, sdata is deasserted on the falling edge.
3	FORCE_MASTER_WRITE_ON_ERROR	When set, the arbiter will treat a slave command as master write in case of a parity error in a command frame.
2	FORCE_MPM_CLK_REQ	When set, the mpm_clk_req is high.
1	ARBITER_BYPASS	Set to enable bypass of SPMI bus arbitration. Used in case where SPMI-controller is a single master and there are no RCS slaves on the bus. Clear to disable the bypass.
0	ARBITER_ENABLE	Set to enable arbitration process (MID must be valid at time of assertion). All other bits in SPMI_CFG_REG are allowed to change value only while ARBITER_ENABLE bit remains low. Deassertion of this bit must be followed by arst to avoid errors

**0x0800B708 SPMI\_SEC\_DISABLE\_REG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

This register disables the use of all security features. Used for boot initialization.

**SPMI\_SEC\_DISABLE\_REG**

Bits	Name	Description
0	DISABLE_SECURITY	Set to '1' to disable and bypass all security features. Set to '0' to enable them.

**0x0800B710 SPMI\_CGC\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI clock gating control register.

**SPMI\_CGC\_CTRL**

Bits	Name	Description
6	MAP_LOGIC_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
5	RPU_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
4	MWB_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
3	PIC_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
2	PAC_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
1	CFG_AHB_BRIDGE_WRCLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open
0	CFG_AHB_BRIDGE_CLK_CGC_ON	When set, hardware CGC control enabled. When clear, the CGC is always open

**0x0800B714 SPMI\_MWB\_ENABLE\_REG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

This register enables master write buffer. It is used in order to prevent receiving an PMIC interrupt until the PIC database is initialized.

**SPMI\_MWB\_ENABLE\_REG**

Bits	Name	Description
0	MWB_ENABLE	Set to '1' to enable master swrite buffer (allow PMIC interrupts to enter the PIC. Set to '0' to disable.

**0x0800B800 SPMI\_PROTOCOL\_IRQ\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI-protocol interrupt status register. The bits in this register are sticky. In order to clear some bit in this register, corresponding bit should be set in the SPMI\_PROTOCOL\_IRQ\_CLEAR register.

**SPMI\_PROTOCOL\_IRQ\_STATUS**

Bits	Name	Description
11	ARBITER_DISCONNECTED	Disconnection was detected.
10	ARBITER_CONNECTED	Connection was detected.
9	PERIH_IRQ_LOST	A Peripheral interrupt was lost.
8	UNEXPECTED_SSC	Unexpected SSC was detected.
7	NO_RESPONSE_DATA_FRAME_DETECTED	No response data-frame was detected.
6	NO_RESPONSE_CMD_FRAME_DETECTED	No response frame was detected.
5	FALSE_MASTER_ARBITRATION_WIN	Master has won the arbitration but didn't initiate any command sequence within TBT.
4	FALSE_BUS_REQUEST	Bus arbitration request was asserted but no master/slave won.
3	UNSUPPORTED_COMMAND	Unsupported SPMI command was detected.
2	DATA_ADDR_FRAME_PARITY_ERROR	A parity error was detected in a data or address frame driven by an external master/slave.
1	SLAVE_CMD_FRAME_PARITY_ERROR	A parity error was detected in a command-frame driven by a slave.
0	MASTER_CMD_FRAME_PARITY_ERROR	A parity error was detected in a command-frame driven by an external master.

**0x0800B804 SPMI\_PROTOCOL\_IRQ\_ENABLE****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI protocol interrupt enable register. This register is used to enable the corresponding functions in the SPMI\_PROTOCOL\_IRQ\_STATUS register. Setting (1) a bit in the SPMI\_PROTOCOL\_IRQ\_ENABLE register causes an interrupt to be generated, if the corresponding bit in the SPMI\_PROTOCOL\_IRQ\_STATUS register is set. Clearing (0) a bit in the SPMI\_PROTOCOL\_IRQ\_ENABLE register causes the setting of the corresponding bit in the SPMI\_PROTOCOL\_IRQ\_STATUS register to have no effect on the interrupt.

**SPMI\_PROTOCOL\_IRQ\_ENABLE**

Bits	Name	Description
11	ARBITER_DISCONNECTED	Enable bit of ARBITER_DISCONNECTED interrupt.
10	ARBITER_CONNECTED	Enable bit of ARBITER_CONNECTED interrupt.
9	PERIH_IRQ_LOST	Enable bit of PERIH_IRQ_LOST interrupt.

**SPMI\_PROTOCOL\_IRQ\_ENABLE (cont.)**

Bits	Name	Description
8	UNEXPECTED_SSC	
7	NO_RESPONSE_DATA_FRAME_DETECTED	Enable bit of NO_RESPONSE_DATA_FRAME_DETECTED interrupt.
6	NO_RESPONSE_CMD_FRAME_DETECTED	Enable bit of NO_RESPONSE_CMD_FRAME_DETECTED interrupt.
5	FALSE_MASTER_ARBITRATION_WIN	Enable bit of FALSE_MASTER_ARBITRATION_WIN interrupt.
4	FALSE_BUS_REQUEST	Enable bit of FALSE_BUS_REQUEST interrupt.
3	UNSUPPORTED_COMMAND	Enable bit of UNSUPPORTED_COMMAND interrupt.
2	DATA_ADDR_FRAME_PARITY_ERROR	Enable bit of DATA_ADDR_FRAME_PARITY_ERROR interrupt.
1	SLAVE_CMD_FRAME_PARITY_ERROR	Enable bit of SLAVE_CMD_FRAME_PARITY_ERROR interrupt.
0	MASTER_CMD_FRAME_PARITY_ERROR	Enable bit of MASTER_CMD_FRAME_PARITY_ERROR interrupt.

**0x0800B808 SPMI\_PROTOCOL\_IRQ\_CLEAR****Type:** W**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI protocol interrupt clear register. Setting one of this register bits generates a pulse which clears corresponding bit in SPMI\_PROTOCOL\_IRQ\_STATUS register.

**SPMI\_PROTOCOL\_IRQ\_CLEAR**

Bits	Name	Description
11	ARBITER_DISCONNECTED	Clear bit of ARBITER_DISCONNECTED interrupt.
10	ARBITER_CONNECTED	Clear bit of ARBITER_CONNECTED interrupt.
9	PERIH_IRQ_LOST	Clear bit of PERIH_IRQ_LOST interrupt.
8	UNEXPECTED_SSC	Clear bit of UNEXPECTED_SSC interrupt.
7	NO_RESPONSE_DATA_FRAME_DETECTED	Clear bit of NO_RESPONSE_DATA_FRAME_DETECTED interrupt.
6	NO_RESPONSE_CMD_FRAME_DETECTED	Clear bit of NO_RESPONSE_CMD_FRAME_DETECTED interrupt.
5	FALSE_MASTER_ARBITRATION_WIN	Clear bit of FALSE_MASTER_ARBITRATION_WIN interrupt.

**SPMI\_PROTOCOL\_IRQ\_CLEAR (cont.)**

Bits	Name	Description
4	FALSE_BUS_REQUEST	Clear bit of FALSE_BUS_REQUEST interrupt.
3	UNSUPPORTED_COMMAND	Clear bit of UNSUPPORTED_COMMAND interrupt.
2	DATA_ADDR_FRAME_PARITY_ERROR	Clear bit of DATA_ADDR_FRAME_PARITY_ERROR interrupt.
1	SLAVE_CMD_FRAME_PARITY_ERROR	Clear bit of SLAVE_CMD_FRAME_PARITY_ERROR interrupt.
0	MASTER_CMD_FRAME_PARITY_ERROR	Clear bit of MASTER_CMD_FRAME_PARITY_ERROR interrupt.

**0x0800B80C SPMI\_PROTOCOL\_IRQ\_EN\_SET****Type:** W**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI protocol interrupt enable set register. This register is used to set the corresponding bits in the SPMI\_PROTOCOL\_IRQ\_ENABLE register. Setting one of this register bits generates a pulse which sets corresponding bit in SPMI\_PROTOCOL\_IRQ\_ENABLE register.

**SPMI\_PROTOCOL\_IRQ\_EN\_SET**

Bits	Name	Description
11	ARBITER_DISCONNECTED	Set-Enable bit of ARBITER_DISCONNECTED interrupt.
10	ARBITER_CONNECTED	Set-Enable bit of ARBITER_CONNECTED interrupt.
9	PERIH_IRQ_LOST	Set-Enable bit of PERIH_IRQ_LOST interrupt.
8	UNEXPECTED_SSC	Set-Enable bit of UNEXPECTED_SSC interrupt.
7	NO_RESPONSE_DATA_FRAME_DETECTED	Set-Enable bit of NO_RESPONSE_DATA_FRAME_DETECTED interrupt.
6	NO_RESPONSE_CMD_FRAME_DETECTED	Set-Enable bit of NO_RESPONSE_CMD_FRAME_DETECTED interrupt.
5	FALSE_MASTER_ARBITRATION_WIN	Set-Enable bit of FALSE_MASTER_ARBITRATION_WIN interrupt.
4	FALSE_BUS_REQUEST	Set-Enable bit of FALSE_BUS_REQUEST interrupt.
3	UNSUPPORTED_COMMAND	Set-Enable bit of UNSUPPORTED_COMMAND interrupt.
2	DATA_ADDR_FRAME_PARITY_ERROR	Set-Enable bit of DATA_ADDR_FRAME_PARITY_ERROR interrupt.
1	SLAVE_CMD_FRAME_PARITY_ERROR	Set-Enable bit of SLAVE_CMD_FRAME_PARITY_ERROR interrupt.

**SPMI\_PROTOCOL\_IRQ\_EN\_SET (cont.)**

Bits	Name	Description
0	MASTER_CMD_FRAME_PARITY_ERROR	Set-Enable bit of MASTER_CMD_FRAME_PARITY_ERROR interrupt.

**0x0800B810 SPMI\_PROTOCOL\_IRQ\_EN\_CLEAR****Type:** W**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI protocol interrupt enable clear register. This register is used to clear the corresponding bits in the SPMI\_PROTOCOL\_IRQ\_ENABLE register. Setting one of this register bits generates a pulse which clears corresponding bit in SPMI\_PROTOCOL\_IRQ\_ENABLE register.

**SPMI\_PROTOCOL\_IRQ\_EN\_CLEAR**

Bits	Name	Description
11	ARBITER_DISCONNECTED	Clear-Enable bit of ARBITER_DISCONNECTED interrupt.
10	ARBITER_CONNECTED	Clear-Enable bit of ARBITER_CONNECTED interrupt.
9	PERIH_IRQ_LOST	Clear-Enable bit of PERIH_IRQ_LOST interrupt.
8	UNEXPECTED_SSC	Clear-Enable bit of UNEXPECTED_SSC interrupt.
7	NO_RESPONSE_DATA_FRAME_DETECTED	Clear-Enable bit of NO_RESPONSE_DATA_FRAME_DETECTED interrupt.
6	NO_RESPONSE_CMD_FRAME_DETECTED	Clear-Enable bit of NO_RESPONSE_CMD_FRAME_DETECTED interrupt.
5	FALSE_MASTER_ARBITRATION_WIN	Clear-Enable bit of FALSE_MASTER_ARBITRATION_WIN interrupt.
4	FALSE_BUS_REQUEST	Clear-Enable bit of FALSE_BUS_REQUEST interrupt.
3	UNSUPPORTED_COMMAND	Clear-Enable bit of UNSUPPORTED_COMMAND interrupt.
2	DATA_ADDR_FRAME_PARITY_ERROR	Clear-Enable bit of DATA_ADDR_FRAME_PARITY_ERROR interrupt.
1	SLAVE_CMD_FRAME_PARITY_ERROR	Clear-Enable bit of SLAVE_CMD_FRAME_PARITY_ERROR interrupt.
0	MASTER_CMD_FRAME_PARITY_ERROR	Clear-Enable bit of MASTER_CMD_FRAME_PARITY_ERROR interrupt.

**0x0800C004 SPMI\_HW\_SW\_EVENTS\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI hardware/software events selector

**SPMI\_HW\_SW\_EVENTS\_CTRL**

Bits	Name	Description
2:0	HW_SW_EVENTS_SEL	when 3'd1 spmi_hw_sw_events is spmi hw events, when 3'd2 spmi_hw_sw_events is spmi sw events, when 3'd3 spmi_hw_sw_events is geni hw events, when 3'd4 spmi_hw_sw_events is geni sw events, else, spmi_hw_sw_events is 32'd0

**0x0800B754 SPMI\_CMPR\_EN\_REG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

This register enables comparison between the PPID the pac requests and the PPID corresponding to the APID it was given by the mapping logic. This register is not relevant when rpu is removed. (version 2.1.0. and above).

**SPMI\_CMPR\_EN\_REG**

Bits	Name	Description
0	CMPR_ENABLE	Set to '1' to enable PPID comparison using the APID2PPID logic

**0x0800C008+ SPMI\_HW\_SW\_EVENTS\_BITWISE\_REGj, j=[0..7]****0x4\*j****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

SPMI hardware/software Bitwise Events selector - to be used instead of SPMI\_HW\_SW\_EVENTS\_CTRL. for each bit in spmi\_hw\_sw\_events there are 2 control bits, a selector and an enable.

**SPMI\_HW\_SW\_EVENTS\_BITWISE\_REG**

Bits	Name	Description
31	HW_EVENTS_EN_D	when 0, bit is disabled, when 1, bit is enabled
24	HW_EVENTS_SEL_D	when 0, selects SPMI_HW_EVENTS, when 1, selects GENI_HW_EVENTS
23	HW_EVENTS_EN_C	when 0, bit is disabled, when 1, bit is enabled
16	HW_EVENTS_SEL_C	when 0, selects SPMI_HW_EVENTS, when 1, selects GENI_HW_EVENTS
15	HW_EVENTS_EN_B	when 0, bit is disabled, when 1, bit is enabled
8	HW_EVENTS_SEL_B	when 0, selects SPMI_HW_EVENTS, when 1, selects GENI_HW_EVENTS
7	HW_EVENTS_EN_A	when 0, bit is disabled, when 1, bit is enabled
0	HW_EVENTS_SEL_A	when 0, selects SPMI_HW_EVENTS, when 1, selects GENI_HW_EVENTS

**0x0A600000+ SPMI\_PIC\_OWNERm\_ACC\_STATUSn, m=[0..7], n=[0..15]**  
**4096\*m+4\*n**

Type: R

Clock: AHB\_CLK

Reset State: 0x00000000

Accumulated status register.

**SPMI\_PIC\_OWNERm\_ACC\_STATUSn**

Bits	Name	Description
31:0	INT_ACC_STATUS	512 accumulated interrupt bits; one bit per peripheral. The n-th register contains accumulated interrupts of peripherals [32*(n+1)-1:32*n]. m defines the owner

**0x0A400000+ SPMI\_PIC\_ACC\_ENABLEn, n=[0..511]**  
**4096\*n**

Type: RW

Clock: AHB\_CLK

Reset State: 0x00000000

Accumulated interrupt enable register.

**SPMI\_PIC\_ACC\_ENABLEn**

Bits	Name	Description
0	INT_ACC_ENABLE	512 interrupt enable bits. Each bit enables a peripherals accumulated interrupt.

**0x0A400004+ SPMI\_PIC\_IRQ\_STATUSn, n=[0..511]  
4096\*n**

**Type:** R  
**Clock:** AHB\_CLK  
**Reset State:** 0x00000000

Peripheral interrupt status register.

**SPMI\_PIC\_IRQ\_STATUSn**

Bits	Name	Description
7:0	INT_STATUS	512 bytes of interrupts. Byte per peripheral to indicate individual interrupts.

**0x0A400008+ SPMI\_PIC\_IRQ\_CLEARn, n=[0..511]  
4096\*n**

**Type:** W  
**Clock:** AHB\_CLK  
**Reset State:** 0x00000000

Interrupt clear register.

**SPMI\_PIC\_IRQ\_CLEARn**

Bits	Name	Description
7:0	INT_CLEAR	512 bytes of interrupt clear commands. Byte per peripheral to clear individual interrupts.

# 7 Universal serial bus registers

---

**0xA8FC000+USB30\_QSRAM\_REGS\_QSRAM\_REG\_n, n=[0..63]**  
**0x4\*n**

**Type:** RW

**Clock:** HCLK

**Reset State:** 0x00000000

General Purpose Registers.

These registers are all mapped to a dedicated RAM.

Accessible as an AHB slave for read, and write.

The address space allocation is 256 Bytes.

## **USB30\_QSRAM\_REGS\_QSRAM\_REG\_n**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
31:0	QSRAM_REG	Reserved Register

## **0xA8F8804 USB30\_QSCRATCH\_CTRL\_REG**

**Type:** RW

**Clock:** MASTER\_CLK

**Reset State:** 0x00000190

## **USB30\_QSCRATCH\_CTRL\_REG**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
9:8	BC_XCVR_SELECT	Battery charge mode XCVR_SELECT to PHY.
7	BC_TERM_SELECT	Battery charge mode TERM_SELECT to PHY.
6	BC_TX_VALID	Battery charge mode TX_VALID to PHY.
5:4	BC_OPMODE	Battery charge mode OPMODE to PHY.
3	BC_DMPULLDOWN	Battery charge mode DMPULLDOWN to PHY.
2	BC_DPPULLDOWN	Battery charge mode DPPULLDOWN to PHY.

**USB30\_QSCRATCH\_CTRL\_REG (cont.)**

Bits	Name	Description
1	BC_IDPULLUP	Battery charge mode IDPULLUP to PHY.
0	BC_SEL	Battery charge mode selection to port 1: (0): normal operation mode. (1): battery charge mode. Default: normal operation mode.

**0xA8F8808 USB30\_QSCRATCH\_GENERAL\_CFG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000004**USB30\_QSCRATCH\_GENERAL\_CFG**

Bits	Name	Description
9	PIPE3_SET_PHYSTATUS_SW	When PIPE3_PHYSTATUS is high, Setting the PIPE3_SET_PHYSTATUS_SW bit will assert the PIPE_PHYSTATUS controller input (0) PHY status software is 0 (1) PHY status software is asserted.
8	PIPE_UTMI_CLK_DIS	Assert to disable the clk from cgc which gets either the utmi_clk or the pipe_clk as it's inputs. When the USB2 is used and the USB3 PHY is not connected need to gate the PIPE clk before switching the Mux to UTMI CLK and de-asserting the PHY STATUS. (0): clk is enabled (1): clk is gated off Default: clock is on. 0x0: PIPE_UTMI_CLK_EN 0x1: PIPE_UTMI_CLK_DISABLE
3	PIPE3_PHYSTATUS_SW	when asserted PIPE3_SET_PHYSTATUS_SW will override the pipe3_Phystatus and the pipe3_Phystatus_async controller inputs. (0): no effect. (1): deassert pipe3_Phystatus, pipe3_Phystatus_async controller inputs when PIPE3_SET_PHYSTATUS_SW=0 Default: no effect. 0x0: NO_SW_OVERRIDE 0x1: PHYSTATUS_SW_OVERRIDE

**USB30\_QSCRATCH\_GENERAL\_CFG (cont.)**

Bits	Name	Description
2	QSRAM_EN	Set QSRAM operation mode: (0): QSRAM inactive. (1): QSRAM active. Default: QSRAM active 0x0: QSRAM_DISABLED 0x1: QSRAM_ENABLED
1	DBM_EN	Set DBM operation mode: (0): DBM inactive. (1): DBM active. Default: DBM inactive 0x0: DBM_DISABLED 0x1: DBM_ENABLED
0	PIPE_UTMI_CLK_SEL	Select between pipe3 clock and utmi clock: (0): pipe3 clock. (1): utmi clock (singleport core: usb30_utmi_clk, multiport core: usb30_utmi_1_clk). Default: pipe3 clock. 0x0: SELECT_PIPE3_CLOCK 0x1: SELECT_UTMI_CLOCK

**0xA8F8810 USB30\_QSCRATCH\_HS\_PHY\_CTRL****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x10000000

HS PHY Control Register

**USB30\_QSCRATCH\_HS\_PHY\_CTRL**

Bits	Name	Description
28	SW_SESSVLD_SEL	(0): utmiotg_valid, utmisrp_bvalid and inverted utmisrp_sessend are driven by outside ports; (1): utmiotg_valid, utmisrp_bvalid and inverted utmisrp_sessend are driven by utmiotg_vbusvalid from qscratch
20	UTMI_OTG_VBUS_VALID	Vbus Valid to USB3.0 Controller this register also assert the bvalid controller input

**0xA8F881C USB30\_QSCRATCH\_CHARGING\_DET\_OUTPUT****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_CHARGING\_DET\_OUTPUT**

Bits	Name	Description
9:8	LINESTATE	LINESTATE value

**0xA8F8820 USB30\_QSCRATCH\_ALT\_INTERRUPT\_EN****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_ALT\_INTERRUPT\_EN**

Bits	Name	Description
12	LINESTATE_INTEN	Enables linestate interrupt

**0xA8F8824 USB30\_QSCRATCH\_HS\_PHY\_IRQ\_STAT****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_HS\_PHY\_IRQ\_STAT**

Bits	Name	Description
12	LINESTATE_INTLCH	indicates linestate change

**0xA8F8828 USB30\_QSCRATCH\_CGCTL\_REG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000

Enable clock gating:

General: Enabled - CGC is active and clock is gated when needed,

Disabled - CGC is bypassed i.e. clock is running

**USB30\_QSCRATCH\_CGCTL\_REG**

Bits	Name	Description
8	DBM_REGFILE_CGC_EN_N	Enable disable the dbm regfile CGC (0) CGC is enabled (default) (1) CGC is disabled.
7	QSCRATCH_CLK_CGC_EN_N	Enable disable the qscratch regfile CGC (0) CGC is enabled (default) (1) CGC is disabled.
6	LSP_DBC_CGC_EN_N	Enable disable the LSP DBC CGC (0) CGC is enabled (default) (1) CGC is disabled
5	LSP_CGC_EN_N	Enable disable the LSP CGCs for host and device (0) CGCs are enabled (default) (1) CGCs are disabled
3	QUALCOMM TECHNOLOGIES, INC. (QTI) DD02482169_FIX_DIS	(0): QTI DD02482169 fix is enabled. (1): fix is disabled. Default: fix is enabled.
2	BAM_CGC_EN	Enable disable the BAM NDP CGC (0): CGC is disabled (default) (1): CGC is enabled
1	DBM_FSM_EN	(0): DBM FSMs CGC disable. (1): DBM FSMs CGC enable. Default: DBM FSMs CGC disable.

**0xA8F8830 USB30\_QSCRATCH\_SS\_PHY\_CTRL****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_SS\_PHY\_CTRL**

Bits	Name	Description
24	LANE0_PWR_PRESENT	indicates vbus present, not same as otgssessvld

**0xA8F8858 USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_STAT****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000

Async Set and status read

**USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_STAT**

Bits	Name	Description
24	USB30_GSI_L1_EXIT_IRQ_STAT	Set by hardware. Reset by software (through writing to the register). When in L1 and there was DEPCMD command this bit is set (1) - L1 exit. (0) - L1 exit did not occurred
18	USB30_LINESTATE_INTSTS	Set by hardware. Cleared by software (Clear on write). (1) - LINESTATE change detected (0) - LINESTATE change has not occurred
14	PME_IRQ_STAT	Set by hardware. Reset by software (Clear on write). (1) - Synopsys core generate pme. (0) - Synopsys core didn't generate pme
13	LPM_OUT_L1_IRQ_STAT	Set by hardware. Reset by software (Clear on write). DBM sets the status when pipe is available. (1) - L1 exit occurred. (0) - L1 exit did not occurred
12	LPM_OUT_RX_ELECidle IRQ_STAT	Set by hardware. Reset by software (Clear on write). Done through usb30_rx_elecidle signal. (1) - rx elec idle end occurred (U3 Exit). (0) - rx elec idle end did not occurred
5	LPM_OUT_L2_IRQ_STAT	Set by hardware. Reset by software (Clear on write). Done through usb30_utmi_suspend_n signal (1) - Clock Request while in L2 suspend mode. (0) - software enabled clock (L2 Exit)
4	LPM_IN_L2_IRQ_STAT	Set by hardware. Reset by software (Clear on write). Done through usb30_utmi_suspend_n signal (1) - Clock gating while in L2 suspend mode. (0) - software gated clock (L2 Enter)
3	POWERDOWN_OUT_P3_IRQ_STAT	Set by hardware. Reset by software (Clear on write). Done through usb30_pipe3_powerdown signal. (1) - Clock Request while in P3 suspend mode. (0) - software enabled clock (P3 Exit)
2	POWERDOWN_IN_P3_IRQ_STAT	Set by hardware. Reset by software (Clear on write). Done through usb30_pipe3_powerdown signal. (1) - Clock gating while in P3 suspend mode. (0) - software gated clock (P3 Enter)

**0xA8F885C USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_MASK****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000

Mask interrupt Register

**USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_MASK**

Bits	Name	Description
24	USB30_GSI_L1_EXIT_IRQ_MASK	When Set the L1_EXIT_IRQ is enabled
18	USB30_LINESTATE_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
14	PME_IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
13	LPM_OUT_L1_IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
12	LPM_OUT_RX_ELECIDLE IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
5	LPM_OUT_L2_IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
4	LPM_IN_L2_IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
3	POWERDOWN_OUT_P3 IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')
2	POWERDOWN_IN_P3_IRQ_MASK	When set, the interrupt is exposed to I/O. default: the interrupt is masked ('0')

**0xA8F8860 USB30\_QSCRATCH\_HW\_SW\_EVT\_CTRL\_REG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000001

Hardware/software Event Control Register

**USB30\_QSCRATCH\_HW\_SW\_EVT\_CTRL\_REG**

Bits	Name	Description
8	SW_EVT_MUX_SEL	Software Event Mux Select: 1'b0: (default) software Events Bus is USB Controller 1'b1: Software Events Bus is DBM. 0x0: SW_EVENT_BUS_USB_CTRL 0x1: SW_EVENT_BUS_DBM
5:4	HW_EVT_MUX_CTRL	Hardware Event Mux Control: 2'b00: (default) hardware Events Bus is DBM 2'b10: Hardware Events Bus is SS software Events Bus in reversed order 2'b11: Hardware Events Bus is HS software Events Bus in reversed order 0x0: HW_EVT_IS_DBM_EVT 0x2: HW_EVT_IS_SS_SW_EVT 0x3: HW_EVT_IS_HS_SW_EVT
0	EVENT_BUS_HALT	This bit is to allow software to halt event bus toggle (for power savings) (0): Event Bus toggle. (1): Event Bus halt. 0x0: EVENT_BUS_TOGGLE 0x1: EVENT_BUS_HALT

**0xA8F8868 USB30\_QSCRATCH\_FLADJ\_30MHZ\_REG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000020

FLADJ\_30MHZ Register

**USB30\_QSCRATCH\_FLADJ\_30MHZ\_REG**

Bits	Name	Description
5:0	FLADJ_30MHZ_VALUE	

**0xA8F886C USB30\_QSCRATCH\_M\_AW\_USER\_REG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000122

M\_AW\_USER Register

**USB30\_QSCRATCH\_M\_AW\_USER\_REG**

Bits	Name	Description
11	AW_MEMTYPE_1_SEL	Hardware Event Mux Control: 1'b0: (default) AXI write MEMTYPE[1] comes from AW_MEMTYPE[1]. 1'b1: AXI write MEMTYPE[1] comes from usb30_m_awcache_ctrl[1]
8	AW_NOALLOCATE	This field gives the ability to control AXI write NOALLOCATE by software Default value is 1'b1
6:4	AW_MEMTYPE	This field gives the ability to control AXI write MEMTYPE by software Default value is 3'b010
3:0	AW_CACHE	This field gives the ability to control AXI write CACHE by software Default value is 4'b0010

**0xA8F8870 USB30\_QSCRATCH\_M\_AR\_USER\_REG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x000000122

M\_AR\_USER Register

**USB30\_QSCRATCH\_M\_AR\_USER\_REG**

Bits	Name	Description
11	AR_MEMTYPE_1_SEL	Hardware Event Mux Control: 1'b0: (default) AXI read MEMTYPE[1] comes from AR_MEMTYPE[1]. 1'b1: AXI read MEMTYPE[1] comes from usb30_m_arcache_ctrl[1]
8	AR_NOALLOCATE	This field gives the ability to control AXI read NOALLOCATE by software Default value is 1'b1
6:4	AR_MEMTYPE	This field gives the ability to control AXI read MEMTYPE by software Default value is 3'b010
3:0	AR_CACHE	This field gives the ability to control AXI read CACHE by software Default value is 4'b0010

**0x0A8F88B4+USB30\_QSCRATCH\_QSCRTCH\_REG\_n, n=[0..4]****0x4\*n****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_QSCRTCH\_REG\_n**

Bits	Name	Description
31:0	QSCRTCH_REG	Reserved Register

**0x0A8F88F0 USB30\_QSCRATCH\_SS\_QMP\_PHY\_CTRL****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000001

Register for QMP SS PHY control signals

**USB30\_QSCRATCH\_SS\_QMP\_PHY\_CTRL**

Bits	Name	Description
0	USB3QMP_PHY_RESET_EN	cc_usb3_ss_phy_ares(usb3phy_reset) responsible to reset all the qmp registers, this register can gate the reset signal to the PHY. This software bit by default enabled (for POR reset) and can be disabled after POR

**0x0A8F88F4 USB30\_QSCRATCH\_SNPS\_CORE\_CFG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000

Used to control SNPS core constant iOS for primary core

**USB30\_QSCRATCH\_SNPS\_CORE\_CFG**

Bits	Name	Description
0	SNPS_CORE_PME_EN	Enable the core to assert pme_generation.

**0xA8F88F8 USB30\_QSCRATCH\_USB30\_STS\_REG****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000003D

USB30 QSCRATCH register

**USB30\_QSCRATCH\_USB30\_STS\_REG**

Bits	Name	Description
5	USB30_CTRL_SLEEP_N	The core assert this active low signal to indicate that the core is in sleep (L1) state and the PHY can switch off some of its internal logic
4	USB30_CTRL_SUSPEND_N	utmi_suspend from the controller. When this bit is low the PHY is in suspend
3	USB30_CTRL_L1_SUSPEND_N	This active low signal is asserted from the core indicating that the core is in Sleep (L1) state. In sleep state, the PHY can use this signal to enter deep low power mode. Deep low power mode is wherein the PHY is able to switch off its internal logic as well as the oscillator circuitry. The PHY clock to core is switched off. This signal anded with the usb30_ctrl_suspend_n is routed to the PHY SUSPENDMD
2	USB30_PHY_STATUS	PIPE3 PHYSTATUS from the PHY
1:0	USB30_OPMODE	synchronized version (2 master clock) of the utmi_opmode signal from the controller. To prevent re-convergence issues need to read this field 2 times. When the value identical there is no re-convergence and the value is valid

**0xA8F88FC USB30\_QSCRATCH\_USB30\_GSI\_GENERAL\_CFG****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_USB30\_GSI\_GENERAL\_CFG**

Bits	Name	Description
20	USB30_GSI_RESTART_DBL_PNTR	When asserted the dbl content will reload the value from the USB30_GSI_RING_BASE_ADDR_L register (As long as the USB30_BLOCK_GSI_RISE_PLS[i]=0)
15:13	USB30_BLOCK_GSI_DBL_RELOAD	Software can block the loading of USB30_GSI_RING_BASE_ADDR_L into the DBL pointer content per GSI EP channel (e.g 100 will block the reload of the third GSI channel)
12	USB30_GSI_CLK_EN	Software asserts this bit to enable the GSI IF Clock

**USB30\_QSCRATCH\_USB30\_GSI\_GENERAL\_CFG (cont.)**

Bits	Name	Description
10	USB30_GSI_SUPPORT_LEADING_DATA	when low the arbiter does not support leading data in the controller QTI DD01933059
9	USB30_GSI_RESP_BLOCK_GRNT	when asserted the arbiter waits for the gsi repose before granting the bus to the USB or BAM
5:4	USB30_GSI_AWLOCK	gsi awlock
1	USB30_BLOCK_GSI_WR_GO	When asserted gsi_wr_go will not be asserted. if there is GSI DBL in progress it will be completed but a new GSI DBL will not occur.
0	USB30_GSI_EN	Enable bit for the USB GSI sniffing feature

**0xA8F8900 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_L****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000000

USB30\_GSI Last Event address pointer (Lower bits)

**USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_L**

Bits	Name	Description
31:0	USB30_GSI_EVT_POINTER_L	Last Event address (Write-back address in device mode) pointer to be written to the GSI register (Lower bits)

**0xA8F8904 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_H****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000000

USB30\_GSI Last Event Data (Higher bits are relevant for AXI\_ADDR\_WIDTH&gt;32)

**USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_H**

Bits	Name	Description
3:0	USB30_GSI_EVT_POINTER_H	Last Event address (Write-back address in device mode) pointer to be written to the GSI register (Higher bits-relevant for AXI_ADDR_WIDTH>32)

**0xA8F8908 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_L****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000000

USB30\_GSI Last Event Error Low

**USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_L**

Bits	Name	Description
31:0	USB30_GSI_EVT_ON_ERR_L	Last Event address (Write-back address in device mode) pointer written to the GSI register and resulted in response error (low bits)

**0xA8F890C USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_H****Type:** R**Clock:** MASTER\_CLK**Reset State:** 0x00000000

USB30\_GSI Last Event Error High (relevant for AXI\_ADDR\_WIDTH&gt;32)

**USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_H**

Bits	Name	Description
3:0	USB30_GSI_EVT_ON_ERR_H	Last Event address (Write-back address in device mode) pointer written to the GSI register and resulted in response error (MSB bits-relevant for AXI_ADDR_WIDTH>32)

**0xA8F8910+ USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Ln, n=[0..2]  
0x4\*n****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000**USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Ln**

Bits	Name	Description
31:0	USB30_GSI_DBL_ADDR_L	Target GSI register Addr per Event Ring (32 LSB)

**0x0A8F8920+ USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Hn, n=[0..2]**  
**0x4\*n**

**Type:** RW  
**Clock:** MASTER\_CLK  
**Reset State:** 0x00000000

SB30\_GSI Target Addr per Event High (relevant for AXI\_ADDR\_WIDTH>32)

**USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Hn**

Bits	Name	Description
3:0	USB30_GSI_DBL_ADDR_H	Target GSI register Addr per Event Ring (high-relevant for AXI_ADDR_WIDTH>32)

**0x0A8F8930+ USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Ln, n=[0..2]**  
**0x4\*n**

**Type:** RW  
**Clock:** MASTER\_CLK  
**Reset State:** 0x00000000

USB30\_GSI Target Addr per Event Low

**USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Ln**

Bits	Name	Description
31:0	USB30_GSI_RING_BASE_A DDR_L	Start address of the TRB ring (NA for Host) LSB bits  Upon initialization software should initialize the TRB captured address to TRB queue base address  This will ensure correct behavior even when events arrive before TRB processing started  (e.g., upon first StartTransfer command which generates StartTransferComplete event)

**0x0A8F8944+ USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Hn, n=[0..2]**  
**0x4\*n**

**Type:** RW  
**Clock:** MASTER\_CLK  
**Reset State:** 0x00000000

USB30\_GSI Target Addr per Event High (relevant for AXI\_ADDR\_WIDTH>32)

**USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Hn**

Bits	Name	Description
3:0	USB30_GSI_RING_BASE_A DDR_H	Start address of the TRB ring (NA for Host) MSB bits-relevant for AXI_ADDR_WIDTH>32  Upon initialization software should initialize the TRB captured address to TRB queue base address  This will ensure correct behavior even when events arrive before TRB processing started  (e.g., upon first Stper Event Ring (low))

**0x0A8F8950+ USB30\_QSCRATCH\_USB30\_GSI\_DEPCMD\_ADDR\_L\_IPA\_EPn, n=[0..1]**  
**0x4\*n**

**Type:** RW

**Clock:** MASTER\_CLK

**Reset State:** 0x00000000

USB30\_GSI DEPCMD Addr per IPA IN EPs

**USB30\_QSCRATCH\_USB30\_GSI\_DEPCMD\_ADDR\_L\_IPA\_EPn**

Bits	Name	Description
20:0	USB30_GSI_DEPCMD_ADD R_L_IPA_EP	DEPCMD address inside the USB core used for of IPA in EP (Address is relative and not absolute. Used to identify DEPCMD during L1 (lsb bits))

**0x0A8F8970+ USB30\_QSCRATCH\_USB30\_IMODn, n=[0..1]**  
**0x4\*n**

**Type:** RW

**Clock:** MASTER\_CLK

**Reset State:** 0x00000000

USB30\_IMOD register

**USB30\_QSCRATCH\_USB30\_IMODn**

Bits	Name	Description
12	USB30_IMOD_EE_EN	Enable filtering usb30_ctrl_irq[n]
10:0	USB30_IMOD_EE_CNT	configurable filter for the ee[n] interrupt (in USB30_IMOD0_CNT*usec/clk_freq units if the USB30_USEC_CNT is configured to the master clk frequency the USB30_IMOD_EE will be in usec units)

**0xA8F8980 USB30\_QSCRATCH\_USB30\_USEC\_CNT****Type:** RW**Clock:** MASTER\_CLK**Reset State:** 0x00000000

USB30\_USEC\_CNT register

**USB30\_QSCRATCH\_USB30\_USEC\_CNT**

Bits	Name	Description
7:0	USB30_USEC_CNT	configurable counter to achieve 1 usec counter. This reg should be configured according to the master clk frequency(=freq[Mhz]) e.g for 125Mhz clk these register should be configured to 125

**0xA904000 USB30\_BAM\_CTRL****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00020000

BAM Control register allows global controls for the BAM.

**USB30\_BAM\_CTRL**

Bits	Name	Description
20	BAM_MESS_ONLY_CANCEL_WB	When active (1) - BAM will DROP former descriptor write-backing and Interrupt firing when former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT ans size 0 arrived (messaging only). When legacy mode is used (0) - This said a case then the former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT and size 0 arrived (messaging only). In this case the FIFO-pointer has been already advanced before the ack_on_sucess arrives and therefore we do write-back to former descriptor and fire interrupt. 1'b1 - Drop WB and Interrupt 1'b0 - legacy.
19	CACHE_MISS_ERR_RESP_EN	When set to '1', upon local ahb access results with cache miss, the bam_ndp will not stall the bus, and finish the access with error response. This bit is relevant for BAM_NDP only.

**USB30\_BAM\_CTRL (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
18:17	LOCAL_CLK_GATING	<p>These Bits enables power saving by using a local clock gating cell.</p> <p>Bit 17: 1'b1 - CGC is on, so that the clock is controled by the hardware - this is the reset value. 1'b0 - CGC is off so that the clock is free runing</p> <p>Bit 18: Reserved.</p>
16	IBC_DISABLE	<p>This Bit enables power saving by disabling the inactivity clock timer/counter. This is useful when:</p> <ol style="list-style-type: none"> <li>1) The BAM is Enabled but idling for long periods.</li> <li>2) The BAM is Enabled and running but the inactivity timers/counters are not required by the software for operating the BAM.</li> </ol> <p>When the BAM is Disabled, it automatically shuts down those timers to save power.</p> <p>1'b1 - Enable Power Saving Mode 1'b0 - Disable Power Saving Mode</p>
15	BAM_CACHED_DESC_STORE	<p>This Bit enables storage of a cached Descriptor into RAM when doing Context Switch and loading it back when returning to this pipe. This is relevant for System Modes Only.</p> <p>BAM to BAM Consumer mode doesn't have this option.</p> <p>Enabling this makes the Context Switch process slightly longer but reduces AHB Descriptor Fetch time later on. It also reduces the number of descriptor fetches done by the BAM if BAM is treating more than 2 pipes.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p> <p>Available in BAM only</p>
14:13	BAM_DESC_CACHE_SEL	<p>This Selector chooses which Descriptor Caching mode will be used. This is relevant for System Producer or Consumer modes only.</p> <p>2'b00 - Cache when current descriptor has less than 64 bytes left. 2'b01 - Cache when current descriptor has less than 128 bytes left. 2'b10 - Immediate Auto cache - cache will happen whenever there are no descriptors cached.</p> <p>This might cause unnecessary descriptors reads when switching between more than 2 pipes.</p> <p>It is the most effective when each pipe processes more bytes than there are in an average descriptor (small descriptors, big block size), or when working with 2 or less pipes.</p> <p>2'b11 - Cache when descriptor is needed only. This results in 1 descriptor present in cache at most.</p> <p>Available in BAM only</p>
11:5	RESERVED	RESERVED

**USB30\_BAM\_CTRL (cont.)**

Bits	Name	Description
4	BAM_EN_ACCUM	When Enabled, BAM will accumulate data written by the peripheral in Direct Mode into INCR8 Bursts. This is an optimization feature for Producer Direct Mode cases. 1'b1 - Enabled 1'b0 - Disabled Available in BAM only
1	BAM_EN	After reset the BAM wakes up in Disabled Mode. While Disabled, BAM operates in Legacy mode (all the transfers from the peripheral go around the BAM as if it doesn't exist) Software Enables this bit to allow BAM operation. This bit doesn't affect the PIPE enable bit in BAM_P_CTRL register. The only possibility to Disable the BAM after it has been enabled is by Hardware or Software reset. 1'b1 - Enabled 1'b0 - Disabled
0	BAM_SW_RST	This will reset the BAM & all Pipes. Software may use this to reset the BAM. As long as the bit remains high the BAM remains in reset state. Software should clear this bit 1'b1 - Reset state 1'b0 - Normal state

**0xA904008 USB30\_BAM\_DESC\_CNT\_TRSHLD****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000001

This Register holds a threshold value for the counter summing the Size of the Descriptors Provided. This value is Global for all pipes but it is relevant for System Producer BAM mode and Both Consumer modes, where Descriptors summing takes place in order to provide the information to the peripheral via pipe bytes free (producer) or pipe bytes available interfaces.

**USB30\_BAM\_DESC\_CNT\_TRSHLD**

Bits	Name	Description
15:0	CNT_TRSHLD	Threshold value. The maximum allowed value is 32kByte. The minimum allowed value is 1 (value 0 is not allowed). Available in BAM only

**0xA904018 USB30\_BAM\_IRQ\_CLR****Type:** W**Clock:** bam\_clk**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

**USB30\_BAM\_IRQ\_CLR**

Bits	Name	Description
4	BAM_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
3	BAM_EMPTY_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM-Lite only
2	BAM_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
1	BAM_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM only

**0xA90401C USB30\_BAM\_IRQ\_EN****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Enable bits for the Interrupts in the BAM Interrupt Status register.

**USB30\_BAM\_IRQ\_EN**

Bits	Name	Description
4	BAM_TIMER_EN	Enables Inactivity timer interrupt by writing this bit 1'b1 - Enable 1'b0 - Disable
3	BAM_EMPTY_EN	1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
2	BAM_ERROR_EN	1'b1 - Enable 1'b0 - Disable

**USB30\_BAM\_IRQ\_EN (cont.)**

Bits	Name	Description
1	BAM_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable Available in BAM only

**0xA904040 USB30\_BAM\_TIMER****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This register counts the idle time of the BAM (all the pipes are in idle state).

**USB30\_BAM\_TIMER**

Bits	Name	Description
15:0	TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0xA904044 USB30\_BAM\_TIMER\_CTRL****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The register can be written and read only when INACTIVITY\_TIMERS\_SUPPORTED generic equals to 1.

**USB30\_BAM\_TIMER\_CTRL**

Bits	Name	Description
31	TIMER_RST	To begin using the inactivity timer the software should write the threshold and write 1 to this bit. writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the TIMER_THREOLD value 1'b1 - Active 1'b0 - Reset
30	TIMER_RUN	Will be used along with TIMER_MODE Not implemented at this time. Set to zero(0)
29	TIMER_MODE	Not implemented at this time. Set to zero(0)

**USB30\_BAM\_TIMER\_CTRL (cont.)**

Bits	Name	Description
15:0	TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

**0xA904084 USB30\_BAM\_CNFG\_BITS\_2****Type:** RW**Clock:** bam\_clk**Reset State:** 0x0000000F

This Register holds the BAM configuration bits. It is highly recommended to follow the directions for each bit and set it accordingly.

**USB30\_BAM\_CNFG\_BITS\_2**

Bits	Name	Description
3	SUP_GRP_LOCKER_RST_SUPPORT	Fixing a bug when the super-group locker pipe is reset and the last CD was with a different super-group. Disable/Enable in BAM-NDP/BAM-Lite only.
2	ACTIVE_PIPE_RST_SUPPORT	Supporting pipe-reset when the pipe is not necessarily quiet. Disable/Enable in BAM-NDP/BAM-Lite only.
1	NO_SW_OFFSET_REVERT_BACK	When doing Write-back do not revert back the BAM_P_SW_OFSTS pointer.
0	CNFG_NO_ACCEPT_AT_FIFO_FULL	Hold the back pressure of accept_ack_on_success in a case of fifo is going to be full. The back-pressure will be hold until after the indication of descriptor-fifo-full (pipe_bytes_ctrl[0]) is asserted (CR701084). Disable/Enable in BAM-NDP only.

**0xA905000 USB30\_BAM\_REVISION****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds a hard coded revision number of the BAM RTL. This corresponds to the BAM target number in the IMS. Also additional BAM settings are reflected here.

**USB30\_BAM\_REVISION**

Bits	Name	Description
31:24	INACTIV_TMR_BASE	This indicates the width of the inactivity timers base counter.
23	CMD_DESC_EN	This indicates BAM has Command Descriptor feature enabled in hardware.

**USB30\_BAM\_REVISION (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
22:21	DESC_CACHE_DEPTH	This field indicates the Per-Pipe-Descriptor-Cache. could be 1-4: 2'b00 - Descriptor Cache Depth of 1. 2'b01 - Descriptor Cache Depth of 2. 2'b10 - Descriptor Cache Depth of 3. 2'b11 - Descriptor Cache Depth of 4. Valid only for BAM_NDP.
20	NUM_INACTIV_TMRS	This bit is checked only if INACTIV_TMRS_EXST = 1. This bit indicates how many inactivity timers are implemented according to value of INACTIVITY_TIMERS_SUPPORTED generic: 1'b1 - One timer for global BAM activity (INACTIVITY_TIMERS_SUPPORTED = 1) 1'b0 - MAX_PIPES timers for each pipe activity (INACTIVITY_TIMERS_SUPPORTED = 2)
19	INACTIV_TMRS_EXST	This indicates BAM has inactivity timers. 1'b1 - Timers available (INACTIVITY_TIMERS_SUPPORTED = 1 or 2) 1'b0 - No timers available (INACTIVITY_TIMERS_SUPPORTED = 0)
18	HIGH_FREQUENCY_BAM	This indicates BAM has Double Sampling on the Ack On Success interface. 1'b1 - Ack On Success double sampled 1'b0 - No double sampling of Ack On Success
17	BAM_HAS_NO_BYPASS	This field indicates BAM has no bypass on the AHB Data bus, from the Peripheral AHB Master to the fabric. When Bypass is available, it is the default routing when BAM is disabled. 1'b1 - No Bypass 1'b0 - Bypass Exists
16	SECURED	This field indicates BAM/BAM-Lite has security support (APU inside) 1'b1 - Secured 1'b0 - Not Secured
15	USE_VMIDMT	This field indicates BAM has VMIDMT supported in hardware.
14	AXI_ACTIVE	This field indicates BAM_NDP uses internal AXI bridge on the master port.
13:12	CE_BUFFER_SIZE	This field indicates the size (in Data words) of the buffer which stores the command elements. Each command element includes 4 words. 2'b00 - 4 Words (one command element). 2'b10 - 8 Words (2 command elements). 2'b11 - 16 Words (4 command elements). Valid Only if CMD_DESC_EN = 1'b1.

**USB30\_BAM\_REVISION (cont.)**

Bits	Name	Description
11:8	NUM_EES	This indicates the Number of Interrupt Lines (Execution Environments) supported by the BAM. When 1 EE is supported, only the BAM_IRQ_SRCS*_EE0 registers exist. When 4 EEs are supported, all BAM_IRQ_SRCS*_EEn registers exist for n=[0..3].

**0xA905008 USB30\_BAM\_NUM\_PIPES****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds a the hard coded number of pipes in the BAM. Since each BAM may have a different number of pipes instantiated in hardware, this register allows the software to know this parameter. The reset state value doesn't have any meaning in this register since it is a constant.

**USB30\_BAM\_NUM\_PIPES**

Bits	Name	Description
31:24	BAM_NON_PIPE_GRP	This field contains the number of BAM Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM
23:16	PERIPH_NON_PIPE_GRP	This field contains the number of Peripheral's Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM
15:14	BAM_DATA_ADDR_BUS_WIDTH	This field indicates BAM data bus address width: 2'b00 - 32bit address width 2'b01 - 36bit address width 2'b1x - Reserved for future widening
7:0	BAM_NUM_PIPES	This field contains the number of pipes available in this BAM

**0xA905024 USB30\_BAM\_AHB\_MASTER\_ERR\_CTRLS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **USB30\_BAM\_AHB\_MASTER\_ERR\_CTRLS**

Bits	Name	Description
22:18	BAM_ERR_HVMID	HVMID
17	BAM_ERR_DIRECT_MODE	DIRECT MODE
16:12	BAM_ERR_HCID	HCID
11:8	BAM_ERR_HPROT	HPROT
7:5	BAM_ERR_HBURST	HBURST
4:3	BAM_ERR_HSIZE	HSIZE
2	BAM_ERR_HWRITE	HWRITE
1:0	BAM_ERR_HTRANS	HTRANS

#### **0xA905028 USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR**

Bits	Name	Description
31:0	BAM_ERR_ADDR	HADDR

#### **0xA90502C USB30\_BAM\_AHB\_MASTER\_ERR\_DATA**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **USB30\_BAM\_AHB\_MASTER\_ERR\_DATA**

Bits	Name	Description
31:0	BAM_ERR_DATA	HDATA

#### **0xA905100 USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

#### **USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

Bits	Name	Description
31:0	BAM_ERR_ADDR	32 LSB of HADDR

#### **0xA905104 USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

**USB30\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB**

Bits	Name	Description
3:0	BAM_ERR_ADDR	4 MSB of Address (bits 35 to 32) of HADDR

**0xA906000 USB30\_BAM\_TRUST\_REG****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any software reset. Only hardware reset applies. When using a secured BAM (APU integrated inside - SECURED field in the BAM\_REVISION register) it is reset by the XPU hardware Reset. When using a non-secured BAM, it is reset by the BAM hardware Reset.

**USB30\_BAM\_TRUST\_REG**

Bits	Name	Description
13	LOCK_EE_CTRL	This bit controls if the EE setting defined in the TRUST registers will be taken into account for the pipe lock grouping decoding. 1'b0 - Only P_LOCK_GROUP is checked for pipe locking. 1'b1 - Both BAM_P_EE and P_LOCK_GROUP are checked for pipe locking.
12:8	BAM_VMid	
7	BAM_RST_BLOCK	When enabled, the BAM Global software reset will not be available for usage. This is in order to deny Global software reset requests by the Global security group. 1'b1 - Enable 1'b0 - Disable
2:0	BAM_EE	This Field Indicates the EE # (0-7) to which the BAM Interrupt belongs to. The pipe interrupts will fire on the EE # set at BAM_P_TRUST_REGn registers. The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register. 3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7

**0x0A907000+ USB30\_BAM\_IRQ\_SRCS\_EEn, n=[0..3]****4096\*n****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking this register, software reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register has an alias - BAM\_IRQ\_SRCS register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**USB30\_BAM\_IRQ\_SRCS\_EEn**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0x0A907004+ USB30\_BAM\_IRQ\_SRCS\_MSK\_EEn, n=[0..3]****4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register has an alias - BAM\_IRQ\_SRCS\_MSK register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**USB30\_BAM\_IRQ\_SRCS\_MSK\_EEn**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)

**USB30\_BAM\_IRQ\_SRCS\_MSK\_EEn (cont.)**

Bits	Name	Description
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

**0xA907008+ USB30\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn, n=[0..3]  
4096\*n****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

The register shows the interrupts sources like (BAM\_IRQ\_SRCS) although the interrupts from the BAM and the pipes are not masked on the lower level. This register has an alias - BAM\_IRQ\_SRCS\_UNMASKED register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**USB30\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

**0xA90700C+USB30\_BAM\_PIPE\_ATTR\_EEn, n=[0..3]  
4096\*n****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

The register shows the pipes attributed to a specific EE, and an indication if BAM is enabled.

The reset value written above is true only if EE>0. For BAM\_PIPE\_ATTR\_EE0 the default value is: Bits [MAX\_PIPES-1:0] = 1 , Bits [31: MAX\_PIPES] = 0 .

**USB30\_BAM\_PIPE\_ATTR\_EEn**

Bits	Name	Description
31	BAM_ENABLED	1'b1 - BAM is Enabled. 1'b0 - BAM Disabled.
30:0	P_ATTR	If bit 'i' == 1'b1 - Pipe 'i' is attributed to this EE.

**0xA907010 USB30\_BAM\_IRQ\_SRCS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking out this register, software reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register points to the physical BAM\_IRQ\_SRCS\_EE0 register.

**USB30\_BAM\_IRQ\_SRCS**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0xA907014 USB30\_BAM\_IRQ\_SRCS\_MSK****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register points to the physical BAM\_IRQ\_SRCS\_MSK\_EE0 register.

**USB30\_BAM\_IRQ\_SRCS\_MSK**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

**0xA907018 USB30\_BAM\_IRQ\_SRCS\_UNMASKED****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

The register shows the interrupts sources like (BAM\_IRQ\_SRCS) although the interrupts from the BAM and the pipes are not masked on the lower level.

This register points to the physical BAM\_IRQ\_SRCS\_UNMASKED\_EE0 register.

**USB30\_BAM\_IRQ\_SRCS\_UNMASKED**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

**0xA906020+ USB30\_BAM\_P\_TRUST\_REGn, n=[0..3]****0x4\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any software reset. Only hardware reset applies. When using a secured BAM (APU integrated inside) it is reset by the xPU hardware Reset. When using a non-secured BAM, it is reset by the BAM hardware Reset.

**USB30\_BAM\_P\_TRUST\_REGn**

Bits	Name	Description
12:8	BAM_P_VMID	
7:3	BAM_P_SUP_GROUP	Super-Group is the upper division of the pipes compared to the pipe-group division. When any pipe locks or unlocks the BAM, all the pipes which are in a different super-group will NOT be affected. Only the pipes within the same super-group of the locker are affected. Usually pipes which belongs to different peripherals which share the same BAM, would be in a different super-groups

**USB30\_BAM\_P\_TRUST\_REGn (cont.)**

Bits	Name	Description
2:0	BAM_P_EE	<p>This Field Indicates the EE # (0-7) to which the Pipe Interrupt belongs to. The BAM interrupt will fire to the EE # set at BAM_P_TRUST_REGn registers.</p> <p>The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register.</p> <p>3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7</p>

**0xA917000+ USB30\_BAM\_P\_CTRLn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Control register provides various controls for the pipe.

**USB30\_BAM\_P\_CTRLn**

Bits	Name	Description
20:16	P_LOCK_GROUP	<p>Pipe's lock group.</p> <p>Upon a lock request on this pipe, all pipes related to different pipe group and different EE will be locked.</p>
11	P_WRITE_NWD	<p>BAM-Lite feature.</p> <p>Applicable to BAM2BAM producer pipes only.</p> <p>When this bit is set for producer B2B pipe, NWD bit (bit number 28 in the second word of descriptor) will be written with EOT into the generated descriptor for the consumer usage.</p> <p>This bit is not applicable in pipes belong to a Peripheral which uses messaging_only (e.g. USB2).</p>
10:9	P_PREFETCH_LIMIT	<p>Bam Lite feature</p> <p>Selects the number of bytes to be prefetched by the Bam Lite, once a peripheral requests to read data from a pipe.</p> <p>This feature is relevant for Consumer modes only.</p> <p>2'b00 - 32 bytes at most (INCR8) 2'b01 - 16 bytes at most (INCR4) 2'b10 - 4 bytes at most (SINGLE)</p> <p>Available in BAM-Lite only</p>

**USB30\_BAM\_P\_CTRLn (cont.)**

Bits	Name	Description
8:7	P_AUTO_EOB_SEL	Bam Lite feature. Selects the number of bytes to be processed before creating End Of Block indication internally by the BAM. Descriptors generated by this feature (bam to bam producer mode) will always be of the chosen size, unless an End Of Transfer, causes a smaller descriptor to be created. 2'b00 - 512 Bytes 2'b01 - 256 Bytes 2'b10 - 128 Bytes 2'b11 - 64 Bytes Available in BAM-Lite only
6	P_AUTO_EOB	BAM Lite feature. When Enabled, the BAM Lite will ignore any incoming End Of Block indications from the peripheral (if any). Instead, the BAM Lite will generate it's own End Of Block indications internally, every P_AUTO_EOB_SEL bytes. This feature is relevant for Producer Bam 2 Bam modes only. It is a must for peripherals which are not able to produce End Of Block indications. 1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
5	P_SYS_MODE	This bit sets the pipe to work in System Mode, where the Software is the second side of the pipe. 1'b1 - System mode. 1'b0 - BAM to BAM mode.
4	P_SYS_STRM	Streaming Enable. Relevant to System Producer BAM mode only. When enabled, BAM will not close descriptors with End Of Transfer notifications received from peripheral, but continue writing the data to the same descriptor. 1'b1 - Enable Streaming Mode 1'b0 - None-Streaming Mode. This field is NOT supported in BAM-NDP (i.e. no streaming mode in NDP).
3	P_DIRECTION	This bit deNOTE:s pipe direction. 1'b1 - The pipe can be written (producer mode) 1'b0 - The pipe can be read (consumer mode)
1	P_EN	1'b1 - Enable Pipe 1'b0 - Disable Pipe

**0xA917004+ USB30\_BAM\_P\_RSTn, n=[0..3]**

4096\*n

**Type:** W**Clock:** bam\_clk**Reset State:** 0x00000000

Pipe Reset register provides Pipe Reset ability. Software needs to write 1 to this register to make the Pipe Reset. Writing 0 to this register will finish the Pipe Reset. Writing zero is needed prior to Peripheral reconfiguration for the same pipe.

NOTE: Software can invoke pipe reset only after it ensure that the pipe is 'quiet'.

('quiet' - No data transaction or command-descriptor is active at the time.)

BAM\_P\_RSTn, n=[0..30]

**USB30\_BAM\_P\_RSTn**

Bits	Name	Description
0	P_SW_RST	This resets the pipe and its registers, (Both flip-flops and RAM). Software can invoke pipe reset only after it ensure that the pipe is 'quiet'. 1'b1 - Reset 1'b0 - Do Nothing

**0xA917008+ USB30\_BAM\_P\_HALTN, n=[0..3]**

4096\*n

**Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Pipe Halt register Enables/Disables the Halt Sequence.

This is a self-modifying register.

It also supports pipe-empty indication and force-descriptor-fifo-full.

**USB30\_BAM\_P\_HALTN**

Bits	Name	Type	Description
4	P_FORCE_DESC_FIFO_FU_LL	RW	When software asserts this bit, descriptor-fifo-full status will be shown to the peripheral on pipe_bytes_avail_ctrl[0]. This is to make the peripheral to think that there is no room for more descriptors. Available only in BAM-NDP and only for Producer BAM-to-BAM pipes.
3	P_PIPE_EMPTY	R	This bit indicates that the Descriptor-FIFO is now empty. It is a read-only bit. Available in BAM-NDP only. Else it is tied high.

**USB30\_BAM\_P\_HALTN (cont.)**

Bits	Name	Type	Description
2	P_LAST_DESC_ZLT	R	This bit indicates that the last created descriptor is with zero-length size. It is a read only bit and effective only for Producer BAM2BAM mode. Available in BAM-NDP only. Else it is tied high.
1	P_PROD_HALTED	RW	This bit is used to inform the Consumer about the Producer being in Halt mode now. This is a part of the Halt procedure. 1'b1 - Sets this bit to 1 1'b0 - Does Nothing This bit will be cleared by the hardware. Not Available in BAM-Lite
0	P_HALT	RW	When Enabled, the Pipe will enter Halt Mode. 1'b1 - Enable Halt 1'b0 - Disable Halt For B2B Producer pipes, this bit is self modifying. It doesn't need the software to clear it. Not Available in BAM-Lite

**0xA917010+ USB30\_BAM\_P\_IRQ\_STTSn, n=[0..3]**

4096\*n

**Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

Pipe Interrupt status register describing the reason for the ongoing interrupt. This is an informative only register and therefore read only. It also has informative bits which are not interrupts.

Clearing the interrupt bits is done by writing to P\_IRQ\_CLR register.

**USB30\_BAM\_P\_IRQ\_STTSn**

Bits	Name	Description
7	P_HRESP_ERR_IRQ	This interrupt Indicates that an Erroneous HResponse has been received by the AHB Master. Additional Information about the error may be read at BAM_AHB_MASTER_ERR_* Registers. Clearing this interrupt also clears the BAM_AHB_MASTER_ERR_* Registers. Available in BAM only
6	P_PIPE_RST_ERROR_IRQ	Unsuccessful Pipe Reset operation. Pipe reset timer expired.
5	P_TRNSFR_END_IRQ	For producer this interrupt happens whenever the last AHB burst of a transfer happens For Consumer when ack_on_success closes a transfer

**USB30\_BAM\_P\_IRQ\_STTSn (cont.)**

Bits	Name	Description
4	P_ERR_IRQ	Pipe Error interrupt indicates that an error has happened. Error reason is not yet defined. Interrupt not in use.
3	P_OUT_OF_DESC_IRQ	Out of descriptors interrupt has several meanings depending on the operation mode: Producer - no free space in the descriptors fifo for adding a descriptor Consumer - no descriptors in the descriptors fifo for processing I/O Vectors Producer - No descriptors to read (and therefore no space for data to write to the pipe) I/O Vectors Consumer - No descriptors to read (and therefore no data to read from pipe)
2	P_WAKE_IRQ	Wake peripheral interrupt signals the driver to wake up the peripheral (USB for example) for transmission via the current pipe. This interrupt is issued whenever an Event comes to the specific pipe.
1	P_TIMER_IRQ	Inactivity timer has reached its threshold value Interrupt
0	P_PRCSD_DESC_IRQ	This Interrupt rises when BAM finishes processing an I/O Vector which has INT bit selected

**0xA917014+ USB30\_BAM\_P\_IRQ\_CLRn, n=[0..3]  
4096\*n****Type:** W**Clock:** bam\_clk**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

BAM\_P\_IRQ\_CLRn, n=[0..30]

**USB30\_BAM\_P\_IRQ\_CLRn**

Bits	Name	Description
7	P_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
6	P_PIPE_RST_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
5	P_TRNSFR_END_CLR	1'b1 - Clear 1'b0 - Unchanged
4	P_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
3	P_OUT_OF_DESC_CLR	1'b1 - Clear 1'b0 - Unchanged

**USB30\_BAM\_P\_IRQ\_CLRn (cont.)**

Bits	Name	Description
2	P_WAKE_CLR	Clear Wake peripheral interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
1	P_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
0	P_PRCSD_DESC_CLR	Clear Descriptor Processed interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged

**0xA917018+ USB30\_BAM\_P\_IRQ\_ENn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Enable bits for the Interrupts in the Pipe Interrupt Status register.

**USB30\_BAM\_P\_IRQ\_ENn**

Bits	Name	Description
7	P_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable
6	P_PIPE_RST_ERROR_EN	1'b1 - Enable 1'b0 - Disable
5	P_TRNSFR_END_EN	1'b1 - Enable 1'b0 - Disable
4	P_ERR_EN	1'b1 - Enable 1'b0 - Disable
3	P_OUT_OF_DESC_EN	1'b1 - Enable 1'b0 - Disable
2	P_WAKE_EN	Enable Wake peripheral interrupt 1'b1 - Enable 1'b0 - Disable
1	P_TIMER_EN	Enable Inactivity timer Interrupt 1'b1 - Enable 1'b0 - Disable
0	P_PRCSD_DESC_EN	Enable Descriptor Processed Interrupt 1'b1 - Enable 1'b0 - Disable

**0xA91701C+USB30\_BAM\_P\_TIMERn, n=[0..3]****4096\*n****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This register counts the idle time of the pipe.

BAM\_P\_TIMERn, n=[0..30]

**USB30\_BAM\_P\_TIMERn**

Bits	Name	Description
15:0	P_TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0xA917020+ USB30\_BAM\_P\_TIMER\_CTRLn, n=[0..3]****4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The resolution of the pipe inactivity timers are in units set by a common counter. This common counter is controlled by a PARAMETER value of the BAM, and uses a separate clock, the inactivity\_timers\_clk. This clock can be slower than the bam\_clk. The intent of the design is to use the sleep\_clk, which is an always on 32KHz clock. This allows the bam\_clk to be turned-off, while the base counter keeps counting. The width of this counter, as well of the rollover value are set using BAM PARAMETER. These values, taking the inactivity\_timers\_clk frequency define the pulse frequency that the base counter gives, and the threshold value can be calculated and set to interrupt after a specific time, regardless of the bam\_clk frequency, and independent of clock power save features of the bam\_clk. The timers continue counting even if the clock is turned off due to BAM dynamic hardware clock request voting.

The timer configuration is based on the inactivity\_timers\_clk period and the INACTIVITY\_TIMER\_WIDTH parameter constant. These parameters should be taken into account when setting this register. For example: for inactivity\_timer\_clk period of 1us and parameter is 3 and P\_TIMER\_TRSHLD is 10 will indicate the  $2^3 * 1\text{us} * 10$  which is 80us of inactivity in a pipe before sending an interrupt.

The general formula is:  $2^{\text{INACTIVITY\_TIMER\_WIDTH}} * \text{clock\_period} * \text{P\_TIMER\_TRSHLD}$ .

**USB30\_BAM\_P\_TIMER\_CTRLn**

Bits	Name	Description
31	P_TIMER_RST	To begin using the inactivity timer the software should write the threshold and write 1 to this bit. Writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the P_TIMER_THRESHOLD value 1'b1 - Active 1'b0 - Reset
30	P_TIMER_RUN	Will be used along with P_TIMER_MODE Not implemented at this time. Set to zero(0)
29	P_TIMER_MODE	Not implemented at this time. Set to zero(0)
15:0	P_TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

**0x0A917024+ USB30\_BAM\_P\_PRDCR\_SDBNDn, n=[0..3]**

4096\*n

**Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register reflects the current status of the Producer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

**USB30\_BAM\_P\_PRDCR\_SDBND**

Bits	Name	Description
24	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_producer. relevant for system mode only. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
20	BAM_P_TOGGLE	Pipe Bytes Free Toggle value. The toggle polarity of the publication (each publication changes its polarity).
19:16	BAM_P_CTRL	Pipe Bytes Free Ctrl value. The possible values are only: 0000 or 0001 (means desc-fifo-full).
15:0	BAM_P_BYTES_FREE	Pipe Bytes Free value. The accumulated bytes_free which are available for peripheral to use

**0x0A917028+ USB30\_BAM\_P\_CNSMR\_SDBNDn, n=[0..3]****4096\*n****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register reflects the current status of the Consumer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

**USB30\_BAM\_P\_CNSMR\_SDBNDn**

Bits	Name	Description
30	BAM_P_ACCEPT_ACK_ON_SUCCESS_TOGGLE	accept_ack_on_sucess toggle. This bit is relevant for BAM_NDP only.
29:28	BAM_P_ACK_ON_SUCCES_S_CTRL	ack_on_sucess control. This bit is relevant for BAM_NDP only.
27	BAM_P_ACK_ON_SUCCES_S_TOGGLE	ack_on_sucess toggle. This bit is relevant for BAM/BAM_NDP only.
26	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_consumer. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
25	BAM_P_NWD_TOGGLE	notify_when_done toggle
24	BAM_P_NWD_TOGGLE_R	notify_when_done toggle sampled
23	BAM_P_WAIT_4_ACK	waiting_for_ack_bytes_avail_r. BAM is waiting for Peripheral to acknowledge the former publish.
22	BAM_P_ACK_TOGGLE	ack_bytes_avail_toggle. The Peripheral's acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
21	BAM_P_ACK_TOGGLE_R	ack_bytes_avail_toggle_r. The Peripheral's sampled acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
20	BAM_P_TOGGLE	Pipe Bytes Avail Toggle value. The toggle polarity of the publication (each publication changes its polarity).
19:16	BAM_P_CTRL	Pipe Bytes Avail Ctrl value. The possible values of BAM_P_CTRL field are: 0100 - Means immediate command (could be only in bam-ndp). 0001 - Means End-Of-Transaction (EOT). 0011 - means End-Of-Transaction with Notify-When-Done (NWD). 0000 - Neither Immediate-command nor EOT nor NWD.
15:0	BAM_P_BYTES_AVAIL	Pipe Bytes AVAIL value. The size of the current publish.

**0xA917800+ USB30\_BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..3]****4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register deNOTE:s the pointer Offset of the first Not Acknowledged Descriptor. Meaning all Descriptors prior to this one have been processed by the BAM. This register is only used by the Software. After receiving an interrupt, software reads this register in order to know what descriptors has been processed.

When the BAM works as Producer, Software will usually read this value after Descriptor Processed Interrupt or after the End of Transfer Interrupt.

When the BAM works as Consumer, Software will usually read this value after Descriptor Processed Interrupt.

NOTE: This is non relevant in BAM to BAM modes.

NOTE: Although being Writable, Software should never write to this register.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..30]

**USB30\_BAM\_P\_SW\_OFSTS<sub>n</sub>**

Bits	Name	Description
31:16	SW_OFST_IN_DESC	Offset inside the Descriptor. This value is used by the Software only when a BAM is working as System Producer. Then the software can read this value to know how many Bytes were written to the current descriptor if it was not closed down. This is useful for Streaming mode. This Field is NOT available in BAM-NDP
15:0	SW_DESC_OFST	Descriptor FIFO offset.

**0xA91782C+USB30\_BAM\_P\_EVNT\_DEST\_ADDR<sub>n</sub>, n=[0..3]****4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_EVNT\_DEST\_ADDRn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe.

#### **0xA917930+ USB30\_BAM\_P\_EVNT\_DEST\_ADDR\_LSBn, n=[0..3] 4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_EVNT\_DEST\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	32 LSB of Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

#### **0xA917934+ USB30\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn, n=[0..3] 4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_EVNT_DEST_ADDR	4 MSB of Address (bits 35 to 32) of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

#### **0xA917818+ USB30\_BAM\_P\_EVNT\_REGn, n=[0..3]**

4096\*n

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

This Register is where Read/Write events are sent to. If current BAM works as Producer, it will receive Read events into this register from the Consumer. If the Consumer is another BAM, it will write read events to this register. If the Consumer is the Software, it will prepare descriptors in the descriptor FIFO and update the DESC\_FIFO\_PEER\_OFST value in this register.

If current BAM works as Consumer, opposite behavior is assumed.

BAM\_P\_EVNT\_REGn, n=[0..30]

#### **USB30\_BAM\_P\_EVNT\_REGn**

Bits	Name	Description
31:16	P_BYTES_CONSUMED	This field is used only in the B2B mode. 15 LSB bits of this field indicate the number of bits consumed by the Consumer. Consumer sends this value to the Producer as a part of a read event, for the Producer to know how many bytes were consumed. The MSB is virtual-event written by the software in pipe-halt procedure. When working in System mode, this value should be set to zero. BAM to BAM Producer writes zero in this value.
15:0	P_DESC_FIFO_PEER_OFST	This field indicates the Descriptor Fifo Offset Pointer of the Peer BAM (The BAM or the Software which creates descriptors). For example when current BAM works as consumer for the current pipe, Producer BAM (or the software) updates this value to let the current pipe know about the new descriptors added in the Descriptor Fifo.

**0xA91781C+USB30\_BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Address of the Descriptor Fifo beginning.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**USB30\_BAM\_P\_DESC\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	Address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0xA917910+ USB30\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the LSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**USB30\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	32 LSB of address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0xA917914+ USB30\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the MSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DESC_FIFO_ADDR	4 MSB of address (bits 35 to 32) of Address of the Descriptors Fifo. This address must be 8 bytes aligned.

#### **0xA917820+ USB30\_BAM\_P\_FIFO\_SIZESn, n=[0..3] 4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

The least significant field is the Descriptor Fifo Size.

The most significant field is the Data Fifo Size.

NOTE: This Descriptor Fifo Size is used by all modes. The Data Fifo Size is non relevant in System Modes.

BAM\_P\_FIFO\_SIZESn, n=[0..30]

#### **USB30\_BAM\_P\_FIFO\_SIZESn**

Bits	Name	Description
31:16	P_DATA_FIFO_SIZE	Size of the Data Address Space. ALWAYS write 16'b0 to this field of P_DATA_FIFO_SIZE unless you work in BAM2BAM producer mode (also known as peer mode). NOTE: Data fifo size should not be bigger than 32KB.
15:0	P_DESC_FIFO_SIZE	Size of the Descriptors Fifo. Must be 8 byte aligned. Each descriptor is 8 bytes. Size of the descriptor fifo must contain an integer number of Descriptors. Size of descriptor fifo must be 16 bytes and above.

#### **0xA917824+ USB30\_BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..3] 4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_DATA\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	Data Address Space beginning.

**0xA917920+ USB30\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	32 LSB of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0xA917924+ USB30\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

#### **USB30\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DATA_FIFO_ADDR	4 MSB of Address (bits 35 to 32) of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0x0A917828+ USB30\_BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register configures the threshold value for Read/Write event generation by the BAM towards another BAM. When aBAM pushes/pops this number of bytes to/from the pipe an event to the other BAM will be generated. A counter summing up the bytes pushed/popped is reset when event is issued, thus restarting count for the threshold.

Events are also issued after End Of Transfer received from the peripheral (Producer case), non regarding whether the pushed bytes count has reached the threshold or not.

This register is non relevant in System Modes.

BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..30]

**USB30\_BAM\_P\_EVNT\_GEN\_TRSHLDn**

Bits	Name	Description
15:0	P_TRSHLD	Threshold value. The maximum allowed value is 32kByte.

**0x0A917804+ USB30\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..30]

**USB30\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n**

Bits	Name	Description
31:16	AU_PSM_ACCUMED	PSM: The acknowledged (EOB/T) number of bytes since last Write Event.  AU: The accumulation of the descriptor sizes for the present ack_on_success event. This is never bigger than the ACKED field. Once the ACCUMED values are bigger than Event Threshold, an Event will be issued, the ACKED value will decrease by the ACCUMED size and this value will be zeroed.  This also serves as the Bytes Consumed value in the Event.

**USB30\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n (cont.)**

Bits	Name	Description
15:0	AU_ACKED	AU: The left over of the number of bytes that were received via the ack_on_success, that did not cause a descriptor closure. This value is down counting upon issuing Read Event. PSM:

**0xA917808+ USB30\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..3]  
4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

**BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..30]****USB30\_BAM\_P\_PSM\_CNTXT\_2\_n**

Bits	Name	Description
31	PSM_DESC_VALID	The Descriptor Valid bit provided by the Writeback state machine to Pipe state machine.
30	PSM_DESC_IRQ	The Descriptor Interrupt bit provided by the Writeback state machine to pipe state machine.
29	PSM_DESC_IRQ_DONE	The Descriptor Interrupt needed selection bit.
28:25	PSM_GENERAL_BITS	General Context Switch Bits
24:22	PSM_CONS_STATE	State of the Pipe Consumer state machine.
21:19	PSM_PROD_SYS_STATE	State of the Pipe Producer System state machine.
18:16	PSM_PROD_B2B_STATE	State of the Pipe Producer BAM to BAM state machine.
15:0	PSM_DESC_SIZE	The Descriptor Size provided by the Writeback state machine to pipe state machine.

**0xA91780C+USB30\_BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..3]  
4096\*n****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **USB30\_BAM\_P\_PSM\_CNTXT\_3\_n**

Bits	Name	Description
31:0	PSM_DESC_ADDR	The Data Address provided in the current descriptor from the Writeback state machine.

**0xA917900+ USB30\_BAM\_P\_PSM\_CNTXT\_3\_LSBn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **USB30\_BAM\_P\_PSM\_CNTXT\_3\_LSBn**

Bits	Name	Description
31:0	PSM_DESC_ADDR	32 LSB of The Data Address provided in the current descriptor from the Writeback state machine.

**0xA917904+ USB30\_BAM\_P\_PSM\_CNTXT\_3\_MSBn, n=[0..3]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

**USB30\_BAM\_P\_PSM\_CNTXT\_3\_MSBn**

Bits	Name	Description
3:0	PSM_DESC_ADDR	4 MSB of address (bits 35 to 32) of the Data Address provided in the current descriptor from the Writeback state machine.

**0xA917810+ USB30\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..3]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..30]

**USB30\_BAM\_P\_PSM\_CNTXT\_4\_n**

Bits	Name	Description
31:16	PSM_DESC_OFST	The running Descriptor Fifo Offset of the Writeback state machine
15:0	PSM_SAVED_ACCUMED_SIZE	The Saved Accumulated Size from the Pipe state machine.

**0xA917814+ USB30\_BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..3]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..30]

**USB30\_BAM\_P\_PSM\_CNTXT\_5\_n**

Bits	Name	Description
31:16	PSM_BLOCK_BYTE_CNT	The Byte count inside the current Block.
15:0	PSM_OFST_IN_DESC	The Offset inside a register. This is used by pipe state machine to compute offset inside a register after retransmission

**0xA917830+ USB30\_BAM\_P\_DF\_CNTXT\_n, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register is used on in BAM-NDP core.

NOTE: Irrelevant for BAM-Lite.

BAM\_P\_DF\_CNTXT\_n, n=[0..30]

**USB30\_BAM\_P\_DF\_CNTXT\_n**

Bits	Name	Description
31:16	WB_ACCUMULATED	Relevant in B2B Producer mode only. This field stores the current running accumulation of the total amount of data that has been written to the data fifo. When it goes bigger than event_threshold a write event occurs.
15:0	DF_DESC_OFST	Holds the descriptor offset of the Descriptor-Fetcher block which is responsible for fetching descriptors before publishing them to the peripheral.

**0xA917834+ USB30\_BAM\_P\_RETR\_CNTXT\_n, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

This is being stored by Ack Update state machine in consumer mode or by the Writeback state machine in producer mode.

This is being loaded by the Writeback state machine when retransmission happens.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_RETR\_CNTXT\_n, n=[0..30]

**USB30\_BAM\_P\_RETR\_CNTXT\_n**

Bits	Name	Description
31:16	RETR_DESC_OFST	Descriptor Fifo Offset for retransmission.
15:0	RETR_OFST_IN_DESC	The Offset inside the Descriptor for retransmission.

**0xA917838+ USB30\_BAM\_P\_SI\_CNTXT\_n, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_SI\_CNTXT\_n, n=[0..30]

**USB30\_BAM\_P\_SI\_CNTXT\_n**

Bits	Name	Description
15:0	SI_DESC_OFST	The Descriptor Fifo Offset pointer stored by the Sideband Inform state machine.

**0xC010A00 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_CTRL**

**Type:** RW  
**Clock:** WCLK  
**Reset State:** 0x00000000

Type-C control

**PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_CTRL**

Bits	Name	Description
4	ARCVR_DTCT_DUAL_EN	Enables autonomous mode receiver detect to be performed on both lanes simultaneously. Only takes effect when in autonomous mode with arcvr_dtct_cm_event_sel set to '1' (in detach state looking for an attach).
2	PORTSELECT_POLARITY	Type-C port select polarity, 1 - invert polarity of portselect from PMIC
1	SW_PORTSELECT_MX	Type-C port select sw mux, 1 - portselect 0 - hardware control
0	SW_PORTSELECT	Type-C port select sw override; only applies if mux is set

**0xC010A04 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_STATUS**

**Type:** R  
**Clock:** WCLK  
**Reset State:** 0x00000000

Type-C Status

**PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_STATUS**

Bits	Name	Description
5:4	ARCVR_DTCT_OUTCOME	Indicates the outcome of autonomous mode receiver detect for portA (bit0) and portB (bit1): 0 - detach, 1 - attach. When arcvr_dtct_dual_en is a '1', can be used to determine which lane caused an autonomous mode receiver detect attach interrupt.
1	PORTSELECT_LATCHED	Value of latched portselect inside of PHY. Latching of the portselect pin value occurs when the PHY exits reset.
0	PORTSELECT_RAW	Value of portselect pin at input to PHY.

**0x0C010A18 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_PLACEHOLDER\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Placeholder register chain Status

**PERIPH\_SS\_USB3PHY\_PCS\_MISC\_PLACEHOLDER\_STATUS**

Bits	Name	Description
0	PLACEHOLDER_STATUS	

**0x0C010C00 PERIPH\_SS\_USB3PHY\_PCS\_SW\_RESET****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001

Contains PHY software reset bit

**PERIPH\_SS\_USB3PHY\_PCS\_SW\_RESET**

Bits	Name	Description
7:1	RESERVED	Reserved
0	SW_RESET	Perform software reset of PCS/SerDes.

**0x0C010C04 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_DOWN\_CONTROL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Controls powerdown and endpoint refclk disable

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_DOWN\_CONTROL**

Bits	Name	Description
7:2	RESERVED	Reserved
1	REFCLK_DRV_DSBL_B	Active low disable for Endpoint RefClk driver; tri-states clock driver outputs
0	SW_PWRDN_B	Active low enable for powering down the QMP Serdes.

**0x0C010C0C PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V0****Type:** RW**Clock:** WCLK**Reset State:** 0x0000009F

Defines drive levels for Tx margin V0

**PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V0**

Bits	Name	Description
7	TX_LARGE_AMP_V0	Tx large amp control for TxMargin = 000b
6:5	TX_IDLE_LVL_V0	Tx idle drive level for TxMargin = 000b
4:0	TXMGN_V0	Tx drive level for TxMargin = 000b.

**0x0C010C10 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V1****Type:** RW**Clock:** WCLK**Reset State:** 0x0000009E

Defines drive levels for Tx margin V1

**PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V1**

Bits	Name	Description
7	TX_LARGE_AMP_V1	Tx large amp control for TxMargin = 001b

**PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V1 (cont.)**

Bits	Name	Description
6:5	TX_IDLE_LVL_V1	Tx idle drive level for TxMargin = 001b
4:0	TXMGN_V1	Tx drive level for TxMargin = 001b.

**0x0C010C14 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V2****Type:** RW**Clock:** WCLK**Reset State:** 0x000000B0

Defines drive levels for Tx margin V2

**PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V2**

Bits	Name	Description
7	TX_LARGE_AMP_V2	Tx large amp control for TxMargin = 010b
6:5	TX_IDLE_LVL_V2	Tx idle drive level for TxMargin = 010b
4:0	TXMGN_V2	Tx drive level for TxMargin = 010b.

**0x0C010C18 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V3****Type:** RW**Clock:** WCLK**Reset State:** 0x00000057

Defines drive levels for Tx margin V3

**PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V3**

Bits	Name	Description
7	TX_LARGE_AMP_V3	Tx large amp control for TxMargin = 011b
6:5	TX_IDLE_LVL_V3	Tx idle drive level for TxMargin = 011b
4:0	TXMGN_V3	Tx drive level for TxMargin = 011b.

**0x0C010C1C PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V4****Type:** RW**Clock:** WCLK**Reset State:** 0x00000069

Defines drive levels for Tx margin V4

#### **PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V4**

Bits	Name	Description
7	TX_LARGE_AMP_V4	Tx large amp control for TxMargin > 011b
6:5	TX_IDLE_LVL_V4	Tx idle drive level for TxMargin > 011b
4:0	TXMGN_V4	Tx drive level for TxMargin > 011b.

#### **0x0C010C20 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_LS**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000069

Defines drive levels for Tx margin Low\_Swing

#### **PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_LS**

Bits	Name	Description
7	TX_LARGE_AMP_LS	Tx large amp control when TxSwing = 1 (low swing).
6:5	TX_IDLE_LVL_LS	Tx idle drive level when TxSwing = 1 (low swing).
4:0	TXMGN_LS	Tx drive level when TxSwing = 1 (low swing).

#### **0x0C010C24 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V0**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000017

Defines de-emphasis settings for Tx -6db de-emphasis V0

#### **PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V0**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_V0	Tx drive deemphasis for TxDemph = 00b when TxMargin = 000b & TxSwing = 0.

**0x0C010C28 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V0****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000F

Defines de-emphasis settings for Tx -3.5db de-emphasis V0

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V0**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_V0	Tx drive deemph for TxDemph = 01b when TxMargin = 000b & TxSwing = 0.

**0x0C010C2C PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000016

Defines de-emphasis settings for Tx -6db de-emphasis V1

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V1**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_V1	Tx drive deemph for TxDemph = 00b when TxMargin = 001b & TxSwing = 0.

**0x0C010C30 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V1****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000F

Defines de-emphasis settings for Tx -3.5db de-emphasis V1

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V1**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_V1	Tx drive deemph for TxDemph = 01b when TxMargin = 001b & TxSwing = 0.

**0x0C010C34 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V2****Type:** RW**Clock:** WCLK**Reset State:** 0x000000011

Defines de-emphasis settings for Tx -6db de-emphasis V2

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V2**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_V2	Tx drive deemph for TxDemph = 00b when TxMargin = 010b & TxSwing = 0.

**0x0C010C38 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V2****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000C

Defines de-emphasis settings for Tx -3.5db de-emphasis V2

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V2**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_V2	Tx drive deemph for TxDemph = 01b when TxMargin = 010b & TxSwing = 0.

**0x0C010C3C PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V3****Type:** RW**Clock:** WCLK**Reset State:** 0x000000019

Defines de-emphasis settings for Tx -6db de-emphasis V3

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V3**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_V3	Tx drive deemph for TxDemph = 00b when TxMargin = 011b & TxSwing = 0.

**0x0C010C40 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V3****Type:** RW**Clock:** WCLK**Reset State:** 0x000000011

Defines de-emphasis settings for Tx -3.5db de-emphasis V3

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V3**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_V3	Tx drive deemph for TxDemph = 01b when TxMargin = 011b & TxSwing = 0.

**0x0C010C44 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V4****Type:** RW**Clock:** WCLK**Reset State:** 0x000000010

Defines de-emphasis settings for Tx -6db de-emphasis V4

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V4**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_V4	Tx drive deemph for TxDemph = 00b when TxMargin > 011b & TxSwing = 0.

**0x0C010C48 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V4****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000B

Defines de-emphasis settings for Tx -3.5db de-emphasis V4

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V4**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_V4	Tx drive deemph for TxDemph = 01b when TxMargin > 011b & TxSwing = 0.

**0x0C010C4C PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_LS****Type:** RW**Clock:** WCLK**Reset State:** 0x000000010

Defines de-emphasis settings for Tx -6db de-emphasis Low\_Swing

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_LS**

Bits	Name	Description
4:0	TXDEEMPH_M6DB_LS	Tx drive deemphasis for TxDemph = 00b when TxSwing = 1.

**0x0C010C50 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_LS****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000B

Defines de-emphasis settings for Tx -3.5db de-emphasis Low\_Swing

**PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_LS**

Bits	Name	Description
4:0	TXDEEMPH_M3P5DB_LS	Tx drive deemphasis for TxDemph = 01b when TxSwing = 1.

**0x0C010C54 PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_DRIVE****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Configures endpoint reference clock driver

**PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_DRIVE**

Bits	Name	Description
5	REFCLK_DRV_OFF_MODE_SEL	Selects behavior of Endpoint RefClk driver when output is off: 0 - clock lines are pulled low when driver is off, 1 - clock lines are idle (common mode maintained) when driver is off. (PCIe only function.)
4:3	REFCLK_CAPSEL	Endpoint RefClk decoupling cap on/off control: 0 - on, 1 - off
2	REFCLK_DRV_SW_EN	Software enable of Endpoint RefClk driver. (PCIe only function.)
1:0	REFCLK_DRV_LVL	Endpoint RefClk drive level. (PCIe only function.)

**0x0C010C58 PERIPH\_SS\_USB3PHY\_PCS\_RX\_IDLE\_DTCT\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000004C

Configures Rx idle detection

**PERIPH\_SS\_USB3PHY\_PCS\_RX\_IDLE\_DTCT\_CNTRL**

Bits	Name	Description
7:4	RX_IDLE_LOW_COUNT_VA L	Defines number of Rclk cycles in a row rx_sigdet must be low before filtered rx_sigdet indicates idle.
3	EIOS_DTCT_CFG	Selects when Rx goes inactive on EIOS detection: 0 - when 1st EIOS (2nd when 5.0GT/s rate) detected, 1 - when last consecutive EIOS detected (must be at least two for 5.0GT/s rate).
2	RX_IDLE_ON_EIOS_EN	Enables detection of EIOS for determining when Rx is idle.
1	RX_IDLE_DTCT_G2S_EN	Enables use of filtered rx_sigdet at gen2 speed (5.0GT/s) for determining when Rx is idle.
0	RX_IDLE_DTCT_G1S_EN	Enables use of filtered rx_sigdet at gen1 speed (2.5GT/s) for determining when Rx is idle. (PCIe only function.)

**0x0C010C5C PERIPH\_SS\_USB3PHY\_PCS\_RATE\_SLEW\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002

Defines slew-rate settings for Transmitter

**PERIPH\_SS\_USB3PHY\_PCS\_RATE\_SLEW\_CNTRL**

Bits	Name	Description
5:3	RATE1_SLEW_CNTRL	Configures QMP Serdes slew control for rate 1 (PCIE Gen-2 and USB3)
2:0	RATE0_SLEW_CNTRL	Configures QMP Serdes slew control for rate 0 (PCIE Gen-1)

**0x0C010C60 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG1****Type:** RW**Clock:** WCLK**Reset State:** 0x000000A3

Configures PHY power states (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG1**

Bits	Name	Description
7	SYSCLK_OFF_IN_L1SS_EN	In the L1SS power state, allows the SysClk buffer and internal clock to be disabled if *phy_pclkack_n is high and (refclk_drv_sw_en = '0' or refclk_drv_sw_en = '1' and *phy_rxelecidle_disable is high). When set, it always causes the state machine clock source to switch from SysClk to AuxClk while PHY is in the L1SS power state. This bit should not be set to a '1' if the CLKREQ# asserted to REFCLK on time of 400ns is required in L1.1 substate. (PCIe only function.)
6	PLL_ON_IN_L1P1_EN	In the L1SS (L1.1) power state, prevents the PLL from being disabled. The L1.1 substate is defined as L1SS state with *phy_rxelecidle_disable high and *phy_txcommonmode_disable low. This bit should be set to a '1' in order to meet the CLKREQ# asserted to REFCLK on time requirement of 400ns. This bit is ignored and the PLL is disabled when the L1SS (L1.2) power state is entered (denoted by *phy_rxelecidle_disable high and *phy_txcommonmode_disable high). This bit has no effect when refclk_drv_sw_en = '0'. (PCIe only function.)
5	RXELECIDLE_LOW_ON_L1SS_EXIT_EN	Selects the value of *phy_rxelecidle during remotely initiated L1SS exit from the time that the controller brings *phy_rxelecidle_disable low and the PHY PLL is locked until the controller brings *phy_pclkreq_n low and the PHY responds by bringing *phy_pclkack_n low. The PHY drives pceiphy_rxelecidle high in L1SS prior to this point. When rxelecidle_low_on_l1ss_exit_en is a '1', the PHY drives *phy_rxelecidle low during remotely initiated L1SS exit. When value is a '0', *phy_rxelecidle behaves normally (0 - Rx traffic detected, 1 - No Rx traffic detected). (PCIe only function.)
4	PU0_P0SU1_RXCLKS_ON	Prevents rx_i_en & c_pll_rxclk_en (and rx_q_en at gen2 speed (5.0GT/s)) from going low during idle in P0/U0 and P0s/U1 power states.
3	SW_BEACON_RX_EN	Enables filtering of Rx signal detection in the generation of *phy_rxelecidle during P2 power state. When enabled, *phy_rxelecidle goes low when a received beacon edge is detected and stays low until no edge is detected for a time period defined by {beacon_2_idle_time_h,beacon_2_idle_time_l}. When disabled, *phy_rxelecidle outputs the inversion of the unfiltered Rx signal detect block output. (PCIe only function.)
2	SW_BEACON_TX_EN	Enables beacon transmit during P2 power state. When enabled in P2 power state and *phy_txelecidle is low, cc_*phy_aux_clk is output from the TX pins. (PCIe only function.)
1	P2U3_CM_CTRL	Selects state of common mode driver during P2/U3 power state -> 0: Common mode driver always disabled while in P2/U3 power state, 1: Common mode driver disabled/enabled when *phy_txelecidle is low/high. (For PCIe, beacon is transmitted when *phy_txelecidle is low if beacon_tx_en is a '1'. For USB3, LFPS is transmitted when *phy_txelecidle is low.) Always set to '1'.
0	PU0_P1U2P2U3_WO_IDLE_EN	Enables transition from P0/U0 to P1/U2 or P2/U3 (or U1 in USB3 mode) without the need for receiver to be idle. Always set to '1'.

**0x0C010C64 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG2****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001B

Configures PHY power states (register 2)

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG2**

Bits	Name	Description
7	EP_REFCLK_WAKEUP_IN_P2_EN	Selects whether Endpoint RefClk driver can be re-enabled during the P2 power state by *phy_clkreq_in_n. If value is '1', the Endpoint RefClk driver re-enable process begins when *phy_clkreq_in_n is driven low (by the endpoint). The Endpoint RefClk driver turns on once the internal pll locks. If value is '0', Endpoint RefClk driver is only re-enabled during P2 exit. The Endpoint RefClk driver is always off when refclk_drv_sw_en = '0'. This bit must be set to a '1' if the CLKREQ# asserted to REFCLK on time of 400ns is required in the P2 power state. (PCIe only function.)
6:4	PHYCLK_REQ_N_CFG	Configures output *phy_phyclk_req_n: 000 - held low; 001 - high in P2/U3 when SysClk buffer and internal clock are disabled; otherwise, low; 010 - high in L1SS when SysClk buffer and internal clock are disabled; otherwise, low; 011 - high in P2/U3 and L1SS when SysClk buffer and internal clock are disabled; otherwise, low; 100 - high from P2/U3 entry until P2/U3 exit request (i.e. *phy_powerdown changes from 11b to 10b); otherwise, low; 101, 110, 111 - reserved. Configuration settings 001, 010 and 011 are used when a high on *phy_phyclk_req_n is intended to indicate that the PHY does not need its reference clock. Configuration setting 100 is used when a high on *phy_phyclk_req_n during P2/U3 is intended to indicate that the PHY is ready to shut off the Endpoint RefClk driver as soon as *phy_clkreq_in_n goes high. Configuration settings 001, 010 and 011 should not be used when *phy_phyclk_req_n is involved in the shutting off of the Endpoint RefClk driver. Configuration setting 100 should be used instead.
3	SYSCLK_OFF_IN_P2U3_EN	In the P2/U3 power state, allows the SysClk buffer and internal clock to be disabled if refclk_drv_sw_en = '0' or refclk_drv_sw_en = '1' and the selected clkreq is high. When set, it always causes the state machine clock source to switch from SysClk to AuxClk while PHY is in the P2/U3 power state. This bit should not be set to a '1' if the CLKREQ# asserted to REFCLK on time of 400ns is required in the P2/U3 power state.
2	PLL_ON_IN_P2_EN	In the P2 power state, prevents the PLL from being disabled. This bit should be set to a '1' in order to meet the CLKREQ# asserted to REFCLK on time requirement of 400ns. This bit has no effect when refclk_drv_sw_en = '0'. (PCIe only function.)

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG2 (cont.)**

Bits	Name	Description
1:0	P2U3_CLKREQ_SEL	Selects source for clkreq in P2/U3 power state. The selected clkreq is used during P2/U3 power state for two purposes; the first is to shut off the Endpoint RefClk driver (only if selected source is high and *phy_clkreq_in_n is high) and the second is to disable the SysClk buffer and internal clock (only after Endpoint RefClk drive is shut off, if sysclk_off_in_p2u3_en = '1'). AuxClk becomes the power state machine clock source when the SysClk buffer and internal clock are disabled. The Endpoint RefClk driver is always off when refclk_drv_sw_en = '0'. Clkreq Source: 00 - held low (disabled), 01 - *phy_pclkreq_n, 10 - *phy_clk_req_n, 11 - *phy_clkreq_in_n.

**0x0C010C68 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG3****Type:** RW**Clock:** WCLK**Reset State:** 0x000000FF

Configures PHY power states (register 3)

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG3**

Bits	Name	Description
7	EBM1_SKP_ADD_STS_EN	Selects whether SKP added indicated on *phy_rxstatus when in USB3 mode, ebufmode = '1' (empty buffer model selected) and SKP OS added due to buffer empty condition (USB3 only function): 0 - SKP added not indicated, 1: SKP added indicated.
6	RX_CM_DSBL_ON_RXTERM_LOW	Selects whether the internal Rx common mode driver is disabled when *phy_rxtermination is low: 0 - remains enabled, 1 - disabled. (USB3 only function.)
5	U2_CM_CTRL	Selects state of common mode driver during U2 power state -> 0: Common mode driver always disabled while in U2 power state, 1: Common mode driver disabled/enabled when *phy_txelecidle is low/high. (LFPS is transmitted when *phy_txelecidle is low.) (USB3 only function.) Always set to '1'.
4	RX_SS_DET_EN	Enables de-assertion of CDR reset in U0 power state when SS traffic is detected. If not enabled, then CDR reset will be de-asserted when rx_ss_wd_time expires. (USB3 only function.)
3:0	RX_SS_WD_TIME	Defines maximum CDR reset assert time in U0 power state (default = ~15.4us): time = 256*rx_ss_wd_time + 15 PCLK cycles or approximately rx_ss_wd_time microseconds. Valid only if rx_ss_det_en is de-asserted. (USB3 only function.)

**0x0C010C6C PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG4****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001

Configures PHY power states (register 4)

**PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG4**

Bits	Name	Description
4	PCLKREQ_IGNORE_IN_P1	When '1', ignores pclkreq going high during transition of P1->P0. If pclkreq goes high before phystatus toggles, the PHY could shut off the pipe clk temporarily.
3	PCLK_ON_IN_P2_EN	When a '1', prevents pipe clock from being disabled in P2. This is only valid when PLL_ON_IN_P2_EN is a '1', REFCLK_DRV_DSBL_B is a 1, and REFCLK_DRV_SW_EN is a 1.
2	PCLK_ON_IN_L1SS_EN	When a '1', prevents pipe clock from being disabled in L1SS. This is only valid when PLL_ON_IN_L1P[12]_EN is a '1', REFCLK_DRV_DSBL_B is a 1, and REFCLK_DRV_SW_EN is a 1. (PCIE only)
1	PLL_ON_IN_L1P2_EN	When a '1', pll is kept on during L1SS (L1.2). This is only valid when REFCLK_DRV_DSBL_B is a 1 and REFCLK_DRV_SW_EN is a 1. (PCIE only)
0	RXELECIDLE_DSBL_CAPTURE_EN	When '1', enables asynchronous capture and hold of rising edge of pciephy_rxecidle_disable by PCS SM in order to guarantee that high pulse is not missed due to sampling of signal by low frequency clock (SysClk or AuxClk).

**0x0C010C70 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_L****Type:** RW**Clock:** WCLK**Reset State:** 0x000000F1

Defines the receiver detect delay in P1/U2 states (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_L**

Bits	Name	Description
7:0	RCVR_DTCT_DLY_P1U2_L	{rcvr_dtct_dly_p1u2_h,rcvr_dtct_dly_p1u2_l} define the delay in PCLK cycles from TxDetectRx going high in P1/U2 power state until the receiver detection circuit is queried and the result placed on RxStatus (and PhyStatus toggled). Default set to 4us when f(PCLK)=62.5MHz. Delay = T(PCLK)/2 * ({rcvr_dtct_dly_p1u2_h,rcvr_dtct_dly_p1u2_l} + 3).

**0x0C010C74 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_H****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001

Defines the receiver detect delay in P1/U2 states (high-order bits)

**PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_H**

Bits	Name	Description
7:4	RESERVED	Reserved
3:0	RCVR_DTCT_DLY_P1U2_H	{rcvr_dtct_dly_p1u2_h,rcvr_dtct_dly_p1u2_l} define the delay in PCLK cycles from TxDetectRx going high in P1/U2 power state until the receiver detection circuit is queried and the result placed on RxStatus (and PhyStatus toggled). Default set to 4us when f(PCLK)=62.5MHz. Delay = T(PCLK)/2 * ({rcvr_dtct_dly_p1u2_h,rcvr_dtct_dly_p1u2_l} + 3).

**0x0C010C78 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_U3\_L****Type:** RW**Clock:** WCLK**Reset State:** 0x00000040

Defines the receiver detect delay in U3 state (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_U3\_L**

Bits	Name	Description
7:0	RCVR_DTCT_DLY_U3_L	{rcvr_dtct_dly_u3_h,rcvr_dtct_dly_u3_l} define the delay in FLL_CLK cycles from TxDetectRx going high in U3 power state until the receiver detection circuit is queried and the result placed on RxStatus (and PhyStatus toggled). Default set to 4us when f(FLL_CLK)=16.67MHz. Delay = T(FLL_CLK) * ({rcvr_dtct_dly_u3_h,rcvr_dtct_dly_u3_l} + 3). (USB3 only function.)

**0x0C010C80 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000080

Configures symbol lock detect (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG1**

Bits	Name	Description
7:0	LOCK_HOLDOFF_TIME_L	{lock_holdoff_time_h, lock_holdoff_time_l} define the number of RClk cycles to wait after CDR reset is released until symbol lock detection is enabled.

**0x0C010C84 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG2****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002

Configures symbol lock detect (register 2)

**PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG2**

Bits	Name	Description
7:0	LOCK_COUNT_VAL_L	{lock_count_val_h, lock_count_val_l} define the number of commas needed at same location consecutively to attain lock; 0 - one comma, 1 - two commas, etc. If enabling the watch-dog timer in Lock_Detect_Config4, the value of {lock_count_val_h, lock_count_val_l} must be less than 0x3E8.

**0x0C010C88 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG3****Type:** RW**Clock:** WCLK**Reset State:** 0x00000040

Configures symbol lock detect (register 3)

**PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG3**

Bits	Name	Description
7:6	LOCK_KEEP_VAL	Defines number of commas received at different location consecutively before lock state machine indicates unlock.
5:4	LOCK_COUNT_VAL_H	{lock_count_val_h, lock_count_val_l} define the number of commas needed at same location consecutively to attain lock; 0 - one comma, 1 - two commas, etc. If enabling the watch-dog timer in Lock_Detect_Config4, the value of {lock_count_val_h, lock_count_val_l} must be less than 0x3E8.
3:0	LOCK_HOLDOFF_TIME_H	{lock_holdoff_time_h, lock_holdoff_time_l} define the number of RClk cycles to wait after CDR reset is released until symbol lock detection is enabled.

**0x0C010C8C PERIPH\_SS\_USB3PHY\_PCS\_TSYNC\_RSYNC\_TIME****Type:** RW**Clock:** WCLK**Reset State:** 0x00000044

Defines Tsync and Rsync times

**PERIPH\_SS\_USB3PHY\_PCS\_TSYNC\_RSYNC\_TIME**

Bits	Name	Description
7:4	TSYNC_DLY_TIME	Defines how many PCLK cycles a Tsync operation requires (and the Rsync assert time) after a rate change and before transitioning from P1/U2 to P0/U0. The number of PCLK cycles required is doubled when running at Gen2 speed.
3:0	ACTIVE_RSYNC_TIME	Defines RSync assert time in PCLK cycles when exiting idle state in P0/U0 and P0s/U1 power states and during P2/U3 to P0/U0 transitions. The Rsync assert time is doubled in terms of PCLK cycles when running at Gen2 speed.

**0x0C010C90 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_LOW\_2\_IDLE\_TIME****Type:** RW**Clock:** WCLK**Reset State:** 0x00000008

Defines Rx signal detect low to idle time

**PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_LOW\_2\_IDLE\_TIME**

Bits	Name	Description
7:0	SIGDET_LOW_2_IDLE_TIME	Defines how many PCLK cycles filtered rx_sigdet must be low before receiver is placed in idle state when in P0/U0 or P0s/U1 power state. (PCIe only function.)

**0x0C010C94 PERIPH\_SS\_USB3PHY\_PCS\_BEACON\_2\_IDLE\_TIME\_L****Type:** RW**Clock:** WCLK**Reset State:** 0x00000080

Defines beacon detect off filter time (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_BEACON\_2\_IDLE\_TIME\_L**

Bits	Name	Description
7:0	BEACON_2_IDLE_TIME_L	{beacon_2_idle_time_h,beacon_2_idle_time_l} define how many SysClk/AuxClk cycles rx_sigdet must be low before *phy_rxidle goes high in P2 state (set to > 16us). Default value gives 20us at 19.2MHz. (PCIe only function.)

**0x0C010C9C PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_TIME\_SYSCLK****Type:** RW**Clock:** WCLK**Reset State:** 0x00000026

Defines power-up delay and wake-up reset duration for SysClk

**PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_TIME\_SYSCLK**

Bits	Name	Description
7:0	PWRUP_RESET_DLY_TIME_SYSCLK	Defines how many SysClk cycles to wait for bias to stabilize on powerup and when returning from L1SS and P2/U3 power states. Used when SysClk is clocking the power state logic (which is always the case during powerup). Set to value > 25h and so delay time is greater than 2us.

**0x0C010CA0 PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_TIME\_AUXCLK****Type:** RW**Clock:** WCLK**Reset State:** 0x00000004

Defines power-up reset duration for AuxClk

**PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_TIME\_AUXCLK**

Bits	Name	Description
7:0	PWRUP_RESET_DLY_TIME_AUXCLK	Defines how many AuxClk cycles to wait for bias to stabilize on powerup. Used when AuxClk is clocking the power state logic. Set to a value greater than 03h.

**0x0C010CA4 PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUXCLK****Type:** RW**Clock:** WCLK**Reset State:** 0x00000004

Defines wake-up reset duration for AuxClk

**PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUXCLK**

Bits	Name	Description
7:0	LP_WAKEUP_DLY_TIME_A UXCLK	Defines how many AuxClk cycles to wait when returning from the P2/U3 power states. Used when AuxClk is clocking the power state logic. Set to a value that results in the same duration as pwrup_reset_dly_time_sysclk, with the added restriction that the value must be greater than 03h.

**0x0C010CAC PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_DET\_HIGH\_COUNT\_VAL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Defines filtering of LFPS detect used by PCS SM

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_DET\_HIGH\_COUNT\_VAL**

Bits	Name	Description
7:4	RESERVED	Reserved
3:0	LFPS_DET_HIGH_COUNT_VAL	Enables digital filtering of the LFPS detect used by the PCS state machines. Defines number of PCLK edges in a row LFPS detect must be high before the PCS state machines see it as high. A value of 0 results in no filtering. The LFPS detect signal that affects the value of *phy_rxelecidle is unfiltered. (USB3 only function.)

**0x0C010CB0 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX\_ECSTART\_EQTLOCK****Type:** RW**Clock:** WCLK**Reset State:** 0x00000086

Defines LFPS transmit extension in U1 & U2 and CDR lock time during equalization training

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX\_ECSTART\_EQTLOCK**

Bits	Name	Description
7:4	RXEQTRAINING_LOCK_TIME	When *phy_rxeqtraining is high, defines amount of time to allow for CDR lock before starting Rx Eq training (default = ~8us): lock time = 0.5 * {value, 8'h0F} PCLK cycles. (USB3 only function.)
3:0	LFPS_TX_END_CNT_START	Defines starting value of down counter used to extend the LFPS transmit time in U1 and U2 power states; this extension is needed so that LFPS will continue to transmit during a transition to the U0 power state. Value indicates a delay in terms of PCLK cycles. (USB3 only function.)

**0x0C010CB4 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX\_END\_CNT\_P2U3\_START****Type:** RW**Clock:** WCLK**Reset State:** 0x000000020

Defines LFPS transmit extension in U3

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX\_END\_CNT\_P2U3\_START**

Bits	Name	Description
7:0	LFPS_TX_END_CNT_U3_START	Defines starting value of down counter used to extend the LFPS transmit time in the U3 power state; this extension is needed so that LFPS will continue to transmit during a transition to the U0 power state. Value indicates a delay in terms of FLL cycles. (USB3 only function.)

**0x0C010CB8 PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_WAIT\_TIME****Type:** RW**Clock:** WCLK**Reset State:** 0x000000075

Defines equalization training wait time

**PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_WAIT\_TIME**

Bits	Name	Description
7:0	RXEQTRAINING_WAIT_TIME	When *phy_rxeqtraining goes high, defines amount of time to wait before enabling CDR lock (default = 30us): wait time = 0.5 * {value, 6'h0F} PCLK cycles. (USB3 only function.)

**0x0C010CBC PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_RUN\_TIME****Type:** RW**Clock:** WCLK**Reset State:** 0x00000007A

Defines equalization training run time

**PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_RUN\_TIME**

Bits	Name	Description
7:0	RXEQTRAINING_RUN_TIME	When *phy_rxeqtraining is high, defines amount of time to allow for Rx Eq training once CDR lock time has expired (default = ~2ms): wait time = 0.5 * {value, 12'h00F} PCLK cycles. Training ends when run time expires or *phy_rxeqtraining is brought low. (USB3 only function.)

**0x0C010CC0 PERIPH\_SS\_USB3PHY\_PCS\_TXONESZEROS\_RUN\_LENGTH****Type:** RW**Clock:** WCLK**Reset State:** 0x00000007

Defines Tx ones/zeros run length

**PERIPH\_SS\_USB3PHY\_PCS\_TXONESZEROS\_RUN\_LENGTH**

Bits	Name	Description
7:4	RESERVED	Reserved
3:0	TXONESZEROS_RUN_LENGTH	Defines the number of consecutive 1's/0's transmitted in USB3 compliance mode when *phy_txoneszeros is high: number of bits = 20 * (reg_val + 1). Register value should only be changed while *phy_txoneszeros is low. (USB3 only function.)

**0x0C010CC4 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002

Configures FLL operation

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL1**

Bits	Name	Description
7:4	FLL_ANA_CTRL	FLL analog control bits.

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL1 (cont.)**

Bits	Name	Description
3	RESERVED	RESERVED
2	FLL_BKGRND_CAL_EN	Enables background adjustment of FLL DAC value during FLL operation.
1	FLL_CAL_ON_STRTUP_EN	Enables calibration during start up.
0	FLL_MAN_MODE	Enables manual mode, where FLL DAC value is given by fll_man_code<7:0>.

**0x0C010CC8 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL2****Type:** RW**Clock:** WCLK**Reset State:** 0x000000081

Defines FLL output divide value

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL2**

Bits	Name	Description
7	FLL_PER_AUTO_MODE	Forces fll on periodically in autonomous mode (power-savings mode)
6	FLL_AO_AUTO_MODE	Forces fll on all the time in autonomous mode
5	FLL_CAL_DONE_MASK	Ignores check for completed FLL calibration in the PS_R1 state
4	FLL_ANA_EN_FORCE_SET	Set fll_ana_en
3	FLL_DIG_EN_FORCE_CLR	Clear fll_dig_en
2:0	FLL_DIV	Defines FLL divider value: $f(FLL) = f(FLL \text{ raw}) / (fll\_div + 1)$

**0x0C010CCC PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_L****Type:** RW**Clock:** WCLK**Reset State:** 0x00000009

Defines FLL count value (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_L**

Bits	Name	Description
7:0	FLL_CNT_VAL_L	{fll_cnt_val_h,fll_cnt_val_l} determine the FLL clock frequency by defining the number of FLL clock cycles during one cycle of qmp_PCIE_usb3_32khz_clk. f(FLL) = 32KHz * {fll_cnt_val_h,fll_cnt_val_l}. Default value of 209h results in f(FLL)=16.67MHz when f(qmp_PCIE_usb3_32khz_clk)=32KHz. Only effective when calibration used.

**0x0C010CD0 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_H\_TOL****Type:** RW**Clock:** WCLK**Reset State:** 0x000000A2

Defines FLL count value (high-order bits) and tolerance

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_H\_TOL**

Bits	Name	Description
7	USE_CNT_VAL_TOL_FORGND	Use the count tolerance value when checking the FLL count during foreground calibration. Enabling this will allow the binary search to converge faster.
6:4	FLL_CNT_VAL_TOL	Defines count tolerance used during background calibration. If count falls within {fll_cnt_val_h,fll_cnt_val_l} +/- fll_cnt_val_tol, then no adjustment to the FLL frequency is made.
3:0	FLL_CNT_VAL_H	{fll_cnt_val_h,fll_cnt_val_l} determine the FLL clock frequency by defining the number of FLL clock cycles during one cycle of qmp_PCIE_usb3_32khz_clk. f(FLL) = 32KHz * {fll_cnt_val_h,fll_cnt_val_l}. Default value of 209h results in f(FLL)=16.67MHz when f(qmp_PCIE_usb3_32khz_clk)=32KHz. Only effective when calibration used.

**0x0C010CD4 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_MAN\_CODE****Type:** RW**Clock:** WCLK**Reset State:** 0x00000040

Defines FLL DAC manual code

**PERIPH\_SS\_USB3PHY\_PCS\_FLL\_MAN\_CODE**

Bits	Name	Description
7:0	FLL_MAN_CODE	Defines code sent to FLL DAC when both fll_man_mode = '1'.

**0x0C010CD8 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000C0

Configures autonomous mode

**PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL**

Bits	Name	Description
7	AUTO_RX_SIGDET_PER_EN	When a 1, forces the rx_sigdet to be enabled once every 32khz refclk cycle only for a period of FLL cycles specified by LFPS_Per_Timer_Val to reduce the power during autonomous mode. Meant to be used in conjunction with FLL_Cntrl2.fil_per_auto_mode.
6	SIGDET_ON_WITH_NO_AUTO_LFPS_P2U3	When a 1, forces the rx_sigdet to be enabled at all times in U3 if autonomous mode is enabled but alfps_dtct_en is cleared.
5	AUTO_RX_SIGDET_AO	When a 1, auto_rx_sigdet_ao forces the transceiver signal detect to be enabled at all times during autonomous mode if alfps_dtct_en is set
4	ARCVR_DTCT_EVENT_SEL	Determines event that generates an interrupt for autonomous receiver detect: 0 - when receiver detect outcome is detach; 1 - when receiver detect outcome is attach
3	ARCVR_DTCT_CM_CTRL	Determines whether the transmitter common mode driver can be disabled in U3 mode when autonomous receiver is enabled: 0 - common mode driver is disabled when common mode is not being re-established and receiver detect is not being performed; 1 - common mode driver is always enabled.
2	RESERVED	Reserved
1	RESERVED	Reserved
0	RESERVED	Reserved

**0x0C010CDC PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_CLEAR****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Clears autonomous mode interrupt

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_CLEAR**

Bits	Name	Description
1	RESERVED	Reserved

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_CLEAR (cont.)**

Bits	Name	Description
0	LFPS_RXTERM_IRQ_CLEA R	Write a '1' and then a '0' to clear and release the autonomous lfps detect / receive detect interrupt.

**0x0C010CE0 PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_EN\_PERIOD****Type:** RW**Clock:** WCLK**Reset State:** 0x00000064

Defines time between autonomous mode receiver detect operations

**PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_EN\_PERIOD**

Bits	Name	Description
7:0	ARCVR_DTCT_EN_PERIOD	Defines amount of time in between receiver detect operations in autonomous receiver detect mode. Time = {arcvr_dtct_en_period, 00010b} * 31.25us. (default = 100.0625ms) The delay defined here must be greater than the combined delays defined by arcvr_dtct_cm_dly and rcvr_dtct_dly_u3_h/l.

**0x0C010CE4 PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_CM\_DLY****Type:** RW**Clock:** WCLK**Reset State:** 0x00000008

Defines time for establishing common mode voltage during autonomous mode receiver detect

**PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_CM\_DLY**

Bits	Name	Description
7:4	RESERVED	Reserved
3:0	ARCVR_DTCT_CM_DLY	Defines amount of time to allow for establishment of common mode voltage before receiver detect is performed in autonomous receiver detect mode. Delay = {arcvr_dtct_cm_dly, 00001b} * 31.25us. (default = 8.03125ms)

**0x0C010CE8 PERIPH\_SS\_USB3PHY\_PCS\_ALFPS\_DEGLITCH\_VAL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000010

Defines autonomous mode LFPS detect filtering

#### **PERIPH\_SS\_USB3PHY\_PCS\_ALFPS\_DEGLITCH\_VAL**

Bits	Name	Description
7:6	RESERVED	Reserved
5:0	ALFPS_DEGLITCH_VAL	Defines how long a received LFPS must be before an interrupt will be generated while operating in autonomous LFPS detect mode. Minimum duration = (1/f(FLL)) (value + 1). The following relationship must be true when configuring this register: LFPS_PER_TIMER_VAL - SIGDET_STARTUP_TIMER_VAL >= ALFPS_DEGLITCH_VAL

#### **0x0C010CEC PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL1**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000006

Provides input signal override values (register 1)

#### **PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL1**

Bits	Name	Description
7	SW_TXSWING	Value muxed onto *phy_txswing when sw_txswing_mx = '1'.
6:4	SW_TXMARGIN	Values muxed onto *phy_txmargin<2:0> when sw_txmgn_mx = '1'.
3:2	SW_TXDEEMPH	Value muxed onto *phy_txdeemph<1:0> when sw_txdeemph_mx = '1'.
1:0	SW_POWERDOWN	Values muxed onto *phy_powerdown<1:0> when sw_powerdown_mx = '1'.

#### **0x0C010CF0 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL2**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000001

Provides input signal override values (register 2)

#### **PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL2**

Bits	Name	Description
7	SW_RATE	Value muxed onto *phy_rate when sw_rate_mx = '1'.

**PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL2 (cont.)**

Bits	Name	Description
6	SW_TXCM_DISABLE	Value muxed onto *phy_txcommonmode_disable when sw_txcm_disable_mx = '1'.
5	SW_RXELECIDLE_DISABLE	Value muxed onto *phy_rxelecidle_disable when sw_rxelecidle_disable_mx = '1'.
4	SW_PCLKREQ_N	Value muxed onto *phy_pclkreq_n when sw_pclkreq_n_mx = '1'.
3	SW_TXCOMPLIANCE	Value muxed onto *phy_txcompliance when sw_txcompliance_mx = '1'.
2	SW_TXDETECTRX_LOOPBACK	Value muxed onto *phy_txdetectrx_loopback when sw_txdtctrx_lpb_mx = '1'.
1	SW_RXPOLARITY	Value muxed onto *phy_rxpolarity when sw_rxpolarity_mx = '1'.
0	SW_TXELECIDLE	Value muxed onto *phy_txelecidle when sw_txelecidle_mx = '1'.

**0x0C010CF4 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL3****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides input signal override values (register 3)

**PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL3**

Bits	Name	Description
7:5	RESERVED	Reserved
4	SW_RXTERMINATION	Value muxed onto *phy_rxtermination when sw_rxtermination_mx = '1'.
3	SW_RXEQTRAINING	Value muxed onto *phy_rxeqtraining when sw_rxeqtraining_mx = '1'.
2	SW_TXONESZEROS	Value muxed onto *phy_txoneszeros when sw_txoneszeros_mx = '1'.
1	SW_EBUFMODE	Value muxed onto *phy_ebufmode when sw_ebufmode_mx = '1'.
0	SW_PHYMODE	Value muxed onto *phy_phymode when sw_phymode_mx = '1'.

**0x0C010CF8 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Enables input signal override (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL1**

Bits	Name	Description
7	SW_TXSWING_MX	Mux control for *phy_txswing.
4	SW_TXMGN_MX	Mux control for *phy_txmargin<2:0>.
2	SW_TXDEEMPH_MX	Mux control for *phy_txdeemph<1:0>.
0	SW_POWERDOWN_MX	Mux control for *phy_powerdown<1:0>.

**0x0C010CFC PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL2****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Enables input signal override (register 2)

**PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL2**

Bits	Name	Description
7	SW_RATE_MX	Mux control for *phy_rate.
6	SW_TXCM_DISABLE_MX	Mux control for *phy_txcommonmode_disable.
5	SW_RXELECIDLE_DISABLE_MX	Mux control for *phy_rxelecidle_disable.
4	SW_PCLKREQ_N_MX	Mux control for *phy_pclkreq_n.
3	SW_TXCOMPLIANCE_MX	Mux control for *phy_txcompliance.
2	SW_TXDTCTRX_LPB_MX	Mux control for *phy_txdetectrx_loopback.
1	SW_RXPOLARITY_MX	Mux control for *phy_rxpolarity.
0	SW_TXELECIDLE_MX	Mux control for *phy_txelecidle.

**0x0C010D00 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL3****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Enables input signal override (register 3)

**PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL3**

Bits	Name	Description
7:5	RESERVED	Reserved
4	SW_RXTERMINATION_MX	Mux control for *phy_rxtermination.
3	SW_RXEQTRAINING_MX	Mux control for *phy_rxeqtraining.
2	SW_TXONESZEROS_MX	Mux control for *phy_txoneszeros.
1	SW_EBUFMODE_MX	Mux control for *phy_ebufmode.
0	SW_PHYMODE_MX	Mux control for *phy_phymode.

**0x0C010D04 PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_SW\_CTRL1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides output signal override values (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_SW\_CTRL1**

Bits	Name	Description
3	SW_RXELECidle	Values muxed onto *phy_rxecidle when sw_rxecidle_mx = '1'.
2	SW_PHYCLK_REQ_N	Value muxed onto *phy_phyclk_req_n when sw_phyclk_req_n_mx = '1'.
1	SW_PCLKACK_N	Values muxed onto *phy_pclkack_n when sw_pclkack_n_mx = '1'.
0	SW_PHYSTATUS	Value muxed onto *phy_phystatus when sw_phystatus_mx = '1'.

**0x0C010D08 PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_MX\_CTRL1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Enables output signal override (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_MX\_CTRL1**

Bits	Name	Description
3	SW_RXELECidle_MX	Mux control for *phy_rxecidle
2	SW_PHYCLK_REQ_N_MX	Mux control for *phy_phyclk_req_n

**PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_MX\_CTRL1 (cont.)**

Bits	Name	Description
1	SW_PCLKACK_N_MX	Mux control for *phy_pclkack_n
0	SW_PHYSTATUS_MX	Mux control for *phy_phystatus

**0x0C010D2C PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_CONTROL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides software Control of Common Block Reset

**PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_CONTROL**

Bits	Name	Description
3	COMRST_FORCE_CLKREQ_LOW_L1SS	Forces CLKREQ# low causing a remote exit from L1SS
2	COMRST_AUTO_ENTER_P2	Automatically enter the P2 state without Controller intervention when a Common Block Reset is asserted
1	FORCE_COM_RESET_ACK	Forces the ACK signal from the Controller to the PHY
0	FORCE_COM_RESET_REQ	Forces the REQ signal from the PHY to the Controller

**0x0C010D34 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY0****Type:** RW**Clock:** WCLK**Reset State:** 0x00000040

Defines PCS BIST PRBS polynomial (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY0**

Bits	Name	Description
7:0	PRBS_POLY0	PRBS polynomial tap definition - low order byte.

**0x0C010D38 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002

Defines PCS BIST PRBS polynomial (high-order byte)

#### **PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY1**

Bits	Name	Description
7:0	PRBS_POLY1	PRBS polynomial tap definition - high order byte.

#### **0x0C010D3C PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED0**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x000000FF

Defines PCS BIST PRBS initial value (low-order byte)

#### **PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED0**

Bits	Name	Description
7:0	PRBS_SEED0	Value of initial PRBS value - low order byte.

#### **0x0C010D40 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED1**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000003

Defines PCS BIST PRBS initial value (high-order byte)

#### **PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED1**

Bits	Name	Description
7:0	PRBS_SEED1	Value of initial PRBS value - high order byte.

#### **0x0C010D44 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT\_CTRL**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000005

Defines PCS BIST fixed pattern K values

**PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT\_CTRL**

Bits	Name	Description
7:4	RESERVED	Reserved
3	FIXED_PAT3_K	Designates fixed_pat3 as either data (0) or control (1)
2	FIXED_PAT2_K	Designates fixed_pat2 as either data (0) or control (1)
1	FIXED_PAT1_K	Designates fixed_pat1 as either data (0) or control (1)
0	FIXED_PAT0_K	Designates fixed_pat0 as either data (0) or control (1)

**0x0C010D48 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT0****Type:** RW**Clock:** WCLK**Reset State:** 0x000000BC

Defines PCS BIST Fixed pattern symbol 0

**PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT0**

Bits	Name	Description
7:0	FIXED_PAT0	Value of first symbol of fixed pattern.

**0x0C010D4C PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT1****Type:** RW**Clock:** WCLK**Reset State:** 0x000000B5

Defines PCS BIST Fixed pattern symbol 1

**PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT1**

Bits	Name	Description
7:0	FIXED_PAT1	Value of second symbol of fixed pattern.

**0x0C010D50 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT2****Type:** RW**Clock:** WCLK**Reset State:** 0x000000BC

Defines PCS BIST Fixed pattern symbol 2

**PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT2**

Bits	Name	Description
7:0	FIXED_PAT2	Value of third symbol of fixed pattern.

**0x0C010D54 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT3****Type:** RW**Clock:** WCLK**Reset State:** 0x0000004A

Defines PCS BIST Fixed pattern symbol 3

**PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT3**

Bits	Name	Description
7:0	FIXED_PAT3	Value of fourth symbol of fixed pattern.

**0x0C010D58 PERIPH\_SS\_USB3PHY\_PCS\_COM\_CLK\_SWITCH\_CTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001

Defines controls for the switchover from SysClk to AuxClk in the Qserver Common block

**PERIPH\_SS\_USB3PHY\_PCS\_COM\_CLK\_SWITCH\_CTRL**

Bits	Name	Description
0	COM_CLK_SWITCH_EN	Enable the wait for the Qserver Common to switch from the SysClk to the AuxClk after shutting down the PLL

**0x0C010D5C PERIPH\_SS\_USB3PHY\_PCS\_ELECIDLE\_DLY\_SEL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000001

Defines delay for elecidle signal

**PERIPH\_SS\_USB3PHY\_PCS\_ELECIDLE\_DLY\_SEL**

Bits	Name	Description
1:0	ELECIDLE_DLY_SEL	Delay in TCLK cycles for elecidle signal to force common mode on TXP/M serial pins: 00: 2 cycle delay, 01: 3 cycle delay, 10: 4 cycle delay , 11: 5 cycle delay

**0x0C010D70 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_SOURCE\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Indicates source of autonomous mode interrupt

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_SOURCE\_STATUS**

Bits	Name	Description
2	CLR_IRQ_BUSY	If set, an utonomous lfps detect / receive detect interrupt clear operation is occurring. LFPS_RXTERM_IRQ_Clear cannot be set until clr_irq_busy is cleared.
1	LFPS_DETECT_IRQ	Autonomous LFPS detect interrupt active
0	RCVR_DETECT_IRQ	Autonomous receiver detect interrupt active

**0x0C010D74 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides PCS status (register 1)

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS**

Bits	Name	Description
7	ACTIVE_STATE_STATUS	Set to a '1' when the PHY has completed power up and is in the P1/U2 state. Goes to '0' and back to '1' when the PHY is transitioning from P2/U3 to P0/U0/P1/U2 and L1SS to P1.
6	PHYSTATUS	Reflects the value of *phy_phystatus.
5:4	PCS_POWERDOWN	Indicates the current powerdown state: 00 - P0/U0, 01 - P0s/U1, 10 - P1/L1SS/U2, 11 - P2/U3

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS (cont.)**

Bits	Name	Description
3	FREEZE_POWERDOWN	Set to '1' when *phy_powerdown value differs from pcs_powerdown. This indicates that the link is requesting a powerdown state change.
2:0	POWER_STATE	Indicates the internal power state: 000 - Reset0 (wait for bias on and PLL enable), 001 - Reset1 (wait for calibration and PLL lock to complete), 010 - L1SS, 011 - P2/U3 to P0/U0 transition, 100 - P0/U0, 101 - P0s/U1, 110 - P1/U2, 111 - P2/U3.

**0x0C010D78 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides PCS status (register 2)

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS2**

Bits	Name	Description
7	POWERPRESENT	Indicates value of PHY input *phy_vbusstatus (passed through to *phy_powerpresent). (USB3 only function.)
6	RX_EQUALIZATION_IN_PROGRESS	Indicates when Rx equalization training is in progress. (USB3 only function.)
5	PCS_LFPS_DET	Indicates when LPFS is being received. (USB3 only function.)
4	FLL_CLK_EN	Indicates when FLL clock generation circuit is enabled. (USB3 only function.)
3	L1SS_MODE	Indicates value of *phy_pclkack_n (high when in L1SS mode). (PCIe only function.)
2	RX_SAMPLER_CAL_IN_PROGRESS	Indicates when Rx sampler calibration is in progress.
1	REC_DETECT_DONE	Goes high when receiver detect operation completes. Stays high until beginning of next receiver detect operation.
0	REC_DETECT_OUTCOME	Indicates outcome of receiver detect operation (value latched).

**0x0C010D7C PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS3****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides PCS status (register 3)

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS3**

Bits	Name	Description
3	RX_SAMPLER_CAL_DONE	Indicates when Rx sampler calibration is finished.
2	PIPE_CLK_EN	Indicates when the Pipe clock is enabled
1	PCS_RATE	Indicates the current rate: 0 - 2.5Gbps (Gen1), 1 - 5Gbps (Gen2)
0	SYSCLK_ENABLED	Indicates if the sysclk is enabled. When it is not enabled, it can be shut down via software

**0x0C010D80 PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Common Block Reset Status

**PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_STATUS**

Bits	Name	Description
2	COM_RESET_INITIATED	Indicates if this port is the one that initiated the Common Block Reset
1	COM_RESET_ACK	Provides the state of the hardware com_reset_ack signal from the PHY to the Controller
0	COM_RESET_REQ	Provides the state of the hardware com_reset_req signal from the PHY to the Controller

**0x0C010D84 PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

SysClk Lock Detector Status

**PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_STATUS**

Bits	Name	Description
0	SYSCLK_ACTIVE	Indicates if the sysclk is active and locked to the correct frequency as determined by the Osc_Dtct_* registers

**0x0C010D88 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID0****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Indicates RevID STEP value (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID0**

Bits	Name	Description
7:0	STEP_7_0	Revision ID0 - low order byte of STEP.

**0x0C010D8C PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID1****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Indicates RevID STEP value (high-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID1**

Bits	Name	Description
7:0	STEP_15_8	Revision ID1 - high order byte of STEP.

**0x0C010D90 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Indicates RevID MINOR value (low-order byte)

**PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID2**

Bits	Name	Description
7:0	MINOR_7_0	Revision ID2 - low order byte of MINOR.

**0x0C010D94 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID3****Type:** R**Clock:** WCLK**Reset State:** 0x00000010

Indicates RevID MINOR value (high-order bits) and MAJOR value

**PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID3**

Bits	Name	Description
7:4	MAJOR	Revision ID3 - MAJOR.
3:0	MINOR_11_8	Revision ID3 - high order nibble of MINOR.

**0x0C010DA8 PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUXCLK\_MSB****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Defines wake-up reset duration for AuxClk (MSB for the 13-bit register)

**PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUXCLK\_MSB**

Bits	Name	Description
4:0	LP_WAKEUP_DLY_TIME_A UXCLK_MSB	Defines the MSB for the value stating how many AuxClk cycles to wait when returning from the P2/U3 power states. Used when AuxClk is clocking the power state logic. Set to a value that results in the same duration as pwrup_reset_dly_time_sysclk, with the added restriction that the value must be greater than 03h.

**0x0C010DAC PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_ACTIONS****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002

Defines the actions effected by the Sysclk Oscillation Detector logic

**PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_ACTIONS**

Bits	Name	Description
3	CLR_OSC_DTCT_STATUS	A 0->1 transition on this field clears the OSC_DTCT_STATUS.SYSCLK_ACTIVE register
2	RESERVED	RESERVED

**PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_ACTIONS (cont.)**

Bits	Name	Description
1	RESERVED	RESERVED
0	RESERVED	RESERVED

**0x0C010DB0 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000007

Provides control for Sigdet-related signals

**PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_CNTRL**

Bits	Name	Description
3	PRE_L1SS_MASK_SIGDET_EN	Enables masking of Rx signal detect during transition to L1SS power state; pciephy_rxidle is held high during this time.
2	P2_MASK_DSBL_SIGDET	Disables Sigdet circuit in Analog when masking is enabled during P2 power state
1	PRE_P2_MASK_SIGDET_EN	Enables masking of Rx signal detect during transition to P2 power state; pciephy_rxidle is held high during this time.
0	P2_MASK_SIGDET_EN	Enables masking of Rx signal detect during P2 power state; pciephy_rxidle is held high during this time.

**0x0C010DB4 PERIPH\_SS\_USB3PHY\_PCS\_IDAC\_CAL\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides control for IDAC Calibration

**PERIPH\_SS\_USB3PHY\_PCS\_IDAC\_CAL\_CNTRL**

Bits	Name	Description
1	FORCE_IDAC_INITIAL_CAL	Forces the IDAC calibration without starting up the PCS (useful for Qserdes BIST)
0	FORCE_IDAC_RECAL	Enables the IDAC calibration to be rerun on the exit of a low-power state

**0x0C010DB8 PERIPH\_SS\_USB3PHY\_PCS\_CMN\_ACK\_OUT\_SEL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides control for the Qserver Cmn Ack signal indicating the switchover from Sysclk to Auxclk

**PERIPH\_SS\_USB3PHY\_PCS\_CMN\_ACK\_OUT\_SEL**

Bits	Name	Description
1:0	CMN_ACK_OUT_SEL	Enables extra cycles of delay before responding to the cmn_ack_out signal

**0x0C010DC0 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

SysClk Lock Detector Status

**PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_STATUS**

Bits	Name	Description
1	ALFPS_DTCT_EN	Indicates if the PHY is in autonomous mode with lfps detection enabled.
0	ARCVR_DTCT_EN	Indicates if the PHY is in autonomous mode with receiver detect enabled.

**0x0C010DC4 PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Provides control for driving the Endpoint Reference clock (PCIe only)

**PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_CNTRL**

Bits	Name	Description
6	EPCLK_ALWAYS_ON_EN	Enables constant driving of the endpoint refclk regardless of the state of CLKREQ#. This will keep the PLL active during low-power modes.

**PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_CNTRL (cont.)**

Bits	Name	Description
5	EPCLK_P2_DLY_EN	Enables a programmable timer to determine when to start driving the endpoint refclk on P2 exit. Mutually exclusive with epclk_p2_pre_pll_lock_en.
4	RESERVED	RESERVED
3	EPCLK_L1P2_DLY_EN	Enables a programmable timer to determine when to start driving the endpoint refclk on the exit of L1.2. Mutually exclusive with epclk_l1p2_pre_pll_lock_en.
2	RESERVED	RESERVED
1	EPCLK_L1P1_DLY_EN	Enables a programmable timer to determine when to start driving the endpoint refclk on the exit of L1.1. Mutually exclusive with epclk_l1p1_pre_pll_lock_en.
0	RESERVED	RESERVED

**0x0C010DD0 PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_L****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Defines LSByte of delay time in microseconds to begin driving endpoint refclk on L1SS and P2 exit

**PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_L**

Bits	Name	Description
7:0	EPCLK_DLY_COUNT_VAL_7_0	epclk_dly_count_val+1 is the delay time in microseconds to begin driving endpoint refclk on L1SS or P2 exit. Is used when epclk_*_dly_en is asserted.

**0x0C010DD4 PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_H****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Defines MSByte of delay time in microseconds to begin driving endpoint refclk on L1SS and P2 exit

**PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_H**

Bits	Name	Description
3:0	EPCLK_DLY_COUNT_VAL_11_8	epclk_dly_count_val+1 is the delay time in microseconds to begin driving endpoint refclk on L1SS or P2 exit. Is used when epclk_*_dly_en is asserted.

**0x0C010DD8 PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_LVL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000009

Sets the RX Sigdet Threshold Level for the different modes

**PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_LVL**

Bits	Name	Description
7:4	RX_SIGDET_LVL_LP	Sets the RX Sigdet Threshold Level for low-power mode
3:0	RX_SIGDET_LVL_NORMAL	Sets the RX Sigdet Threshold Level for normal mode

**0x0C010DDC PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK\_LSB****Type:** RW**Clock:** WCLK**Reset State:** 0x00000004

Defines L1SS wake-up reset duration for AuxClk

**PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK LSB**

Bits	Name	Description
7:0	L1SS_WAKEUP_DLY_TIME_AUXCLK_7_0	Defines how many AuxClk cycles to wait when returning from the L1SS power state. Used when AuxClk is clocking the power state logic. Set to a value that results in the same duration as pwrup_reset_dly_time_sysclk, with the added restriction that the value must be greater than 03h.

**0x0C010DE0 PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK\_MSB****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Defines L1SS wake-up reset duration for AuxClk

#### **PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK\_MSB**

Bits	Name	Description
4:0	L1SS_WAKEUP_DLY_TIME_AUXCLK_12_8	Defines how many AuxClk cycles to wait when returning from the L1SS power state. Used when AuxClk is clocking the power state logic. Set to a value that results in the same duration as pwrup_reset_dly_time_sysclk, with the added restriction that the value must be greater than 03h.

#### **0x0C010DE4 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL2**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000002

Configures autonomous mode

#### **PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL2**

Bits	Name	Description
1	AUTO_MODE_MASK_RXEL_ECIDLE	When a 1, causes the PHY to mask rxecidle in autonomous mode.
0	AUTO_MODE_IGNORE_PIPE_SIGS	When a 1, causes the PHY to stay in autonomous mode as long as *phy_auto_mode_en is high and to ignore the PIPE interface signals. The PHY will still need to be in U3 in order to enter autonomous mode.

#### **0x0C010DE8 PERIPH\_SS\_USB3PHY\_PCS\_RXTERMINATION\_DLY\_SEL**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000001

Provides delay control when asserting the PIPE RxTermination input.

#### **PERIPH\_SS\_USB3PHY\_PCS\_RXTERMINATION\_DLY\_SEL**

Bits	Name	Description
1:0	RXTERMINATION_DLY_SEL	Enables extra cycles of delay when asserting RxTermination (de-assertion is not affected). This is used to mask glitches on the RxSigdet

**0x0C010DEC PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_PER\_TIMER\_VAL****Type:** RW**Clock:** WCLK**Reset State:** 0x000000043

Provides period duration of FLL in power savings mode

**PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_PER\_TIMER\_VAL**

Bits	Name	Description
7:0	LFPS_PER_TIMER_VAL	Provides the number of FLL clock cycles that the FLL and RX SigDet will be enabled when FLL_Cntrl2.fll_per_auto_mode is a 1. The following relationship must be true when configuring this register: LFPS_PER_TIMER_VAL - SIGDET_STARTUP_TIMER_VAL >= ALFPS_DEGLITCH_VAL

**0x0C010DF0 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_STARTUP\_TIMER\_VAL****Type:** RW**Clock:** WCLK**Reset State:** 0x000000022

Provides startup time for RX SigDet to stabilize

**PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_STARTUP\_TIMER\_VAL**

Bits	Name	Description
7:0	SIGDET_STARTUP_TIMER_VAL	Provides the number of FLL clock cycles for the RX SigDet to stabilize before being sampled when FLL_Cntrl2.fll_per_auto_mode is a 1 and Autonomous_Mode_Ctrl.auto_rx_sigdet_per_en is a 1. The following relationship must be true when configuring this register: LFPS_PER_TIMER_VAL - SIGDET_STARTUP_TIMER_VAL >= ALFPS_DEGLITCH_VAL

**0x0C010DF4 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG4****Type:** RW**Clock:** WCLK**Reset State:** 0x000000007

Configures symbol lock detect (register 4)

**PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG4**

Bits	Name	Description
7	GEN2_LOCK_WDT_EN	Enables use of watch-dog timer in Gen2 to reset receive path if symbol lock takes longer than (gen12_lock_wdt_val + 1) microseconds. (Gen2)
6	GEN1_LOCK_WDT_EN	Enables use of watch-dog timer in Gen1 to reset receive path if symbol lock takes longer than (gen12_lock_wdt_val + 1) microseconds. (Gen1)
5:0	GEN12_LOCK_WDT_VAL	Defines Gen1/2 watch-dog timer count value in (n+1) 1us increments. (Gen1/2)

**0x0C010DF8 PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_DTCT\_CNTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000004

Configures Rx Sigdet detection

**PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_DTCT\_CNTRL**

Bits	Name	Description
3:0	RX_SIGDET_HIGH_COUNT_VAL	Defines number of internal CmbClk cycles in a row rx_sigdet must be high before it is considered active. A value of 0x0 disables any filtering.

**0x0C010DFC PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides PCS status (register 4)

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4**

Bits	Name	Description
3	DISPARITY_ERROR	Indicates that a disparity error has occurred.
2	ELASTIC_BUFFER_OFLOW	Indicates that the elastic buffer has overflowed.
1	ELASTIC_BUFFER_UFLOW	Indicates that the elastic buffer has underflowed.
0	DEC_ERR	Indicates that an 8b10b error has occurred.

**0x0C010E00 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4\_CLEAR****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

Clears PCS status register 4

**PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4\_CLEAR**

Bits	Name	Description
0	STATUS4_CLEAR	Resets all the error indicators in PCS_STATUS4 to b0.

**0x0C010E04 PERIPH\_SS\_USB3PHY\_PCS\_DEC\_ERROR\_COUNT\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides 8b10b Error Counter

**PERIPH\_SS\_USB3PHY\_PCS\_DEC\_ERROR\_COUNT\_STATUS**

Bits	Name	Description
7	DEC_ERR_CNT_OFLOW	Binary counter overflow. More than 128 errors have occurred.
6:0	DEC_ERR_CNT	7-bit binary counter for 8b10b errors. Clears upon phy reset. NOTE: The PHY data bus to the transceiver is 20 bits wide with two simultaneous 10-bit symbols. If both symbols have an error, the counter only increments by one. This register should be used as a general indication of the 8b10b error count.

**0x0C010E08 PERIPH\_SS\_USB3PHY\_PCS\_COMMA\_POS\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

Provides Comma Position Status

**PERIPH\_SS\_USB3PHY\_PCS\_COMMA\_POS\_STATUS**

Bits	Name	Description
7:0	COMMA_POS	8-bit decimal current position of the comma lock. Valid decimal values are 0 to 40.

**0x0C011000 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID0****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID0**

Bits	Name	Description
7:0	STEP_7_0	READ ONLY STEP[7:0]: indicates a change in the hardware which is not intended to impact software compatibility.

**0x0C011004 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID1****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID1**

Bits	Name	Description
7:0	STEP_15_8	READ ONLY STEP[15:8]: indicates a change in the hardware which is not intended to impact software compatibility.

**0x0C011008 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID2**

Bits	Name	Description
7:0	MINOR_7_0	READ ONLY Minor[7:0]: indicates expanded functionality. Minor version adds functionality while being backward compatible for existing features.

**0x0C01100C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID3****Type:** R**Clock:** WCLK**Reset State:** 0x00000030

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID3**

Bits	Name	Description
7:4	MAJOR	READ ONLY Major[3:0]: indicates different interface version. Major version changes are not backward compatible. There will be a new programming guide document per major revision.
3:0	MINOR_11_8	READ ONLY Minor[11:8]: indicates expanded functionality. Minor version adds functionality while being backward compatibility for existing features.

**0x0C011010 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG****Type:** RW**Clock:** WCLK**Reset State:** 0x00000005**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG**

Bits	Name	Description
4	CORE_RESET_TSYNC	core_reset_tsync
3	PLL_START	- 1: enable PLL block.
2	PLL_RESET	To kick off PLL a low-high-low pulse is needed; The pulse width is 32 reference clocks long
1	SW_RESET	- 1: all PHY digital is reset except control register.
0	RETIMING_ENABLE	Rising edge of retime enable will re-lock the re-time logic. - 0: Disable the re-timer logic (read path disabled). - 1: Enable the re-time logic (rising edge causes the read gray-code address generation to re-lock to the write path).

**0x0C011014 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG\_1****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000F**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG\_1**

Bits	Name	Description
7	SW_PORTSELECT	Type_C portselect sw override; only applies if mux is set to 2'b00

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG\_1 (cont.)**

Bits	Name	Description
6:5	SW_PORTSELECT_MX	Type_C portselect sw mux, 00: portselect from PMIC (includes the sw polarity inversion), 01: latched portselect (includes the sw polarity inversion), 10: sw_portselect 11: same as 00 condition
4	PORTSELECT_POLARITY	Type_C portselect polarity, 1: invert polarity of portselect from PMIC
3	RT_BUFFER_LANE3_EN	enable retime buffer lane 3 clocks. 1: enable, 0: clock gate the clocks
2	RT_BUFFER_LANE2_EN	enable retime buffer lane 2 clocks. 1: enable, 0: clock gate the clocks
1	RT_BUFFER_LANE1_EN	enable retime buffer lane 1 clocks. 1: enable, 0: clock gate the clocks
0	RT_BUFFER_LANE0_EN	enable retime buffer lane 0 clocks. 1: enable, 0: clock gate the clocks

**0x0C011018 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_PD\_CTL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000038**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_PD\_CTL**

Bits	Name	Description
7:6	SPARE_PWRDN_B	
5	PLL_PWRDN_B	pwrdn_b for PLL
4	LANE_2_3_PWRDN_B	pwrdn_b for Lane 2 & 3
3	LANE_0_1_PWRDN_B	pwrdn_b for Lane 0 & 1
2	AUX_PWRDN_B	pwrdn_b for AUX CH.
1	PSR_PWRDN	- 1: power down tx driver and gate off serializer clock
0	PWRDN_B	- 0: power down all analog blocks and gated all clocks

**0x0C01101C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MODE****Type:** RW**Clock:** WCLK**Reset State:** 0x000000CC

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MODE**

Bits	Name	Description
7	LANE_TXCLK_EN	0: gate-disable muxed clock from lanes (I0_tclk, I1_tclk)
6	RETIME_BUFFER_SIZE	0: 3 entires 1: 5 entries
5:4	LINK_CLOCK_SEL	00: I0_tclk 01: I1_tclk 10: reserved 11: reserved
3:2	PAR_RATE	- 11 - 40 bit parallel interface; - 10 - 10 bit parallel interface; - 01 - 20 bit parallel interface; - 00 - 10 bit parallel interface
1:0	TX_BAND	Sets the speed band for the TX path. - 00 - Full rate; - 01 - Half rate; - 10 - quarter rate; - 11 - Eighth rate

**0x0C011020 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG0****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG0**

Bits	Name	Description
3	AUX_FORCE_TX_EN	force aux TX driver enable.
2	AUX_BYPASS	bypass mode; 1: digital control signals directly connect from link controller to analog block; 0: normal;.
1:0	RESERVED	Reserved

**0x0C011024 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG1**

Bits	Name	Description
7:0	AUX_DCTRL_7_0	<p>&lt;2:0&gt; Tx swing control 000: Swing &lt; 300mv; 001, 010, 011, 100 Swing within the spec; 101,110,111: High swing, max ~ 700mv in TT</p> <p>&lt;4:3&gt; DC offset controls bits</p> <p>&lt;5&gt; Enable term resistor (dctrl&lt;5&gt;&amp;&amp;rx_en) 1: term R is controlled by tx_en and dctrl&lt;7&gt; 0: term R is enabled as long as rx_en is high.</p> <p>&lt;6&gt; Enable PN swap 1: PN swaped. 0: No PN swap.</p> <p>&lt;7&gt; Enable rx_term(dctrl&lt;7&gt;&amp;&amp;tx_en) 1: term resistor not controlled by rx_en and dctrl&lt;5&gt; 0: term resistor is enabled as long as tx_en is high</p>

**0x0C011028 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG2****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG2**

Bits	Name	Description
2:0	AUX_DCTRL_10_8	<p>&lt;0&gt; Enable dc_offset_en 1: disable dc_offset circuit 0: if rx_en is high and jtag_mode_en is low, dc_offset is enabled.</p> <p>&lt;2:1&gt; Reserved</p>

**0x0C01102C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG3****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG3**

Bits	Name	Description
7	AUX_FORCE_RX_EN	force aux RX driver enable.
6:4	AUX_TX_PRECHARGE	Zero bits count for pre-charge phase. Range is from 10 to 16. - 000: 10; 001: 11; 010: 12; 011: 13; - 100: 14; 101: 15; 110: 16
3	AUX_PARALLEL_LBK	A MUX select; 1: parallel loopback mode; 0: normal mode;

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG3 (cont.)**

Bits	Name	Description
2	AUX_BYPASS_MANCH_LBK	A MUX select; 1: bypass manchester loopback mode; 0: normal mode;
1	AUX_SERIAL_LBK	A MUX select; 1: serial loopback mode; 0: normal mode;
0	AUX_NEAREND_LBK	A MUX select; 1: near-end loopback mode; 0: normal mode

**0x0C011030 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG4****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000A**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG4**

Bits	Name	Description
5:0	AUX_TX_MANCH_HALFCYC	Reference clock counts for counting half cycle of Manchester bit period. $Ref\_clk / (\text{Manchester\_bit\_rate} * 2)$ For example: $19.2\text{MHz} / (1\text{MHz} * 2) = 9.6$ then round up to 10;

**0x0C011034 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG5****Type:** RW**Clock:** WCLK**Reset State:** 0x00000028**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG5**

Bits	Name	Description
7:0	AUX_RX_MANCH_TWOCYC	Reference clock counts for counting two cycle of Manchester bit period. $(Ref\_clk * 2) / \text{Manchester\_bit\_rate}$ Recommendation: 19.2MHz: 8'd39; 27MHz: 8'd54;

**0x0C011038 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG6****Type:** RW**Clock:** WCLK**Reset State:** 0x0000000A

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG6**

Bits	Name	Description
5:0	AUX_RX_MANCH_HALFCYC C	Reference clock counts for counting half cycle of Manchester bit period. Ref_clk/(Manchester_bit_rate * 2) Recommendation: 19.2MHz: 6'd10; 27MHz: 6'd14;

**0x0C01103C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG7****Type:** RW**Clock:** WCLK**Reset State:** 0x00000003**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG7**

Bits	Name	Description
3:0	AUX_RX_MANCH_CYCJITT ER	Reference clock counts for counting jitter on RX. Recommendation: 19.2MHz: 4'd6; 27MHz: 4'd9;

**0x0C011040 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG8****Type:** RW**Clock:** WCLK**Reset State:** 0x000000B7**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG8**

Bits	Name	Description
7:4	AUX_TX_PREAMBLE	Zero bits count for TX preamble phase. Range is from 5 to 20. - 0000:5; 0001:6; 0010:7; 0011:8; - 0100:9; 0101:10; 0110: 11; - 0111: 12; 1000: 13; 1001: 14; 1010: 15; - 1011: 16; 1100: 17; 1101: 18; 1110: 19; 1111: 20
3:0	AUX_RX_PREAMBLE	Zero bits count for RX preamble phase. Range is from 1 to 16. - 0000:1; 0001:2; 0010:3; 0011:4; - 0100:5; 0101:6; 0110: 7; - 0111: 8; 1000: 9; 1001: 10; 1010: 11; - 1011: 12; 1100: 13; 1101: 14; 1110: 15; 1111: 16

**0x0C011044 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG9****Type:** RW**Clock:** WCLK**Reset State:** 0x00000003**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG9**

Bits	Name	Description
7	OVERRIDE_TWOCYC	By default, calibration logic uses 4 * preamble pulse width as two cycle pulse width. To override this parameter by CSR, this bit should be set
6	OVERRIDE_HALFCYC	By default, calibration logic uses preamble range from 6 to 13 cycles of reference clock. To override this paramters with CSR, this bit should be set
5	OVERRIDE_AUX_TIMING	To override automatic calibrated aux timing parameters, this bit should be set
4	USE_OLD_AUX_RX	To use old aux-rx design, this bit should be set
3:0	AUX_RX_ZERO_IDLE_CNT	Maximum allowed clock counts for counting no activity on the line between SYNC to STOP pattern. MAX_IDLE_CNT = (aux_rx_zero_idle_cn + 1) * 40 AUX_CLOCK cycles.

**0x0C011048 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_MASK****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_MASK**

Bits	Name	Description
4	AUX_TX_REQ_ERR_MASK	Mask bit for REQ. error on TX. - 0: Mask interrupt - 1: Enable interrupt
3	AUX_RX_ALIGN_ERR_MASK	Mask bit for ALIGN error on RX. - 0: Mask interrupt - 1: Enable interrupt
2	AUX_RX_SYNC_ERR_MASK	Mask bit for SYNC pattern error on R. - 0: Mask interrupt - 1: Enable interrupt
1	AUX_RX_DEC_ERR_MASK	Mask bit for data decode error on R. - 0: Mask interrupt - 1: Enable interrupt

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_MASK (cont.)**

Bits	Name	Description
0	AUX_RX_STOP_ERR_MASK	Mask bit for STOP pattern error on R. - 0: Mask interrupt - 1: Enable interrupt

**0x0C01104C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_CLEAR****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_CLEAR**

Bits	Name	Description
7	AUX_GLOBE_REQ_CLR_CMD	- 1:clear bits in register EDPPHY_GLB_AUX_INTERRUPT_STATUS simultaneously based on the following err_clr[4:0] bits. - 0: no impact clear procedure: step1: set AUX_TX_REQ_ERR_CLR/AUX_RX_ALIGN_ERR_CLR/AUX_RX_SYNC_ERR_CLR/AUX_RX_DEC_ERR_CLR/AUX_RX_STOP_ERR_CLR bits to 1 step2: set AUX_GLOBE_REQ_CLR_CMD to 1 step3: read the register EDPPHY_GLB_AUX_INTERRUPT_STATUS bits to confirm status clearing step4: set AUX_GLOBE_REQ_CLR_CMD to 0 step5: set AUX_TX_REQ_ERR_CLR/AUX_RX_ALIGN_ERR_CLR/AUX_RX_SYNC_ERR_CLR/AUX_RX_DEC_ERR_CLR/AUX_RX_STOP_ERR_CLR bits to 0
4	AUX_TX_REQ_ERR_CLR	Clear bit for REQ. error on TX. - 1: clear interrupt
3	AUX_RX_ALIGN_ERR_CLR	Clear bit for ALIGN error on RX. - 1: clear interrupt
2	AUX_RX_SYNC_ERR_CLR	Clear bit for SYNC pattern error on RX. - 1: clear interrupt
1	AUX_RX_DEC_ERR_CLR	Clear bit for data decode error on RX. - 1: clear interrupt
0	AUX_RX_STOP_ERR_CLR	Clear bit for STOP pattern error on RX. - 1: clear interrupt

**0x0C011068 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_VCO\_DIV****Type:** RW**Clock:** WCLK**Reset State:** 0x00000002**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_VCO\_DIV**

Bits	Name	Description
3	FORCE_CLKBUF_EN	Force c_ext_clkbuf_en signal to CMN
1:0	VCO_DIV_CLK_SEL	Clock selection signal: 00: clkout=0; 01: clkdiv2; 10: clkdiv4; 11:clkdiv8

**0x0C01106C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_LANE\_CTL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_LANE\_CTL**

Bits	Name	Description
2	L0_CORE_TXCLK_EN	txclk_en to lane
0	RESERVED	RESERVED

**0x0C011078 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE0****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE0**

Bits	Name	Description
7:0	L0_PRBS_SEED_7_0	First Byte of PRBS SEED.

**0x0C01107C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE1**

Bits	Name	Description
1:0	L0_PRBS_SEED_9_8	Second Byte of PRBS SEED.

**0x0C011088 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_LANE\_CTL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_LANE\_CTL**

Bits	Name	Description
2	L1_CORE_TXCLK_EN	txclk_en to lane
0	RESERVED	RESERVED

**0x0C011094 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE0****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE0**

Bits	Name	Description
7:0	L1_PRBS_SEED_7_0	First Byte of PRBS SEED.

**0x0C011098 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE1****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE1**

Bits	Name	Description
1:0	L1_PRBS_SEED_9_8	Second Byte of PRBS SEED.

**0x0C0110A4 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_CTRL****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_CTRL**

Bits	Name	Description
1	MISR_CLEAR	MISR clear
0	MISR_ENABLE	- 1: Enable MISR. - 0: Disable MISR.

**0x0C0110BC PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_STATUS**

Bits	Name	Description
4	AUX_TX_REQ_ERR	READ ONLY REQ. signal is late from controller on TX. When delay from aux_tx_ack to next aux_tx_req is larger than 160 aux clock (19.2MHz) cycles, the tx_req_err interrupt is asserted.
3	AUX_RX_ALIGN_ERR	READ ONLY Received less than 8 bits data prior to STOP pattern on RX.
2	AUX_RX_SYNC_ERR	READ ONLY SYNC pattern error on RX. NO_DOC
1	AUX_RX_DEC_ERR	READ ONLY Data decode error on RX.
0	AUX_RX_STOP_ERR	READ ONLY STOP pattern error on RX.

**0x0C0110C0 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_STATUS**

Bits	Name	Description
3	PORTSELECT_RAW	READ ONLY Value of portselect pin at input to PHY.
2	PORTSELECT_LATCHED	READ ONLY Value of latched portselect inside PHY. Latching of the portselect pin value occurs when the PHY exist reset.
1	PHY_READY	READ ONLY - 1: PHY is ready to send data.
0	TSYNC_DONE	READ ONLY - 1: core_reset_tsync operation is done.

**0x0C0110E8 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_STATUS****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_STATUS**

Bits	Name	Description
0	MISR_CAPTURED	READ ONLY MISR captured signal.

**0x0C0110EC PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS000****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS000**

Bits	Name	Description
7:0	L0_MISR_7_0	READ ONLY MISR output.

**0x0C0110F0 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS001****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS001**

Bits	Name	Description
7:0	L0_MISR_15_8	READ ONLY MISR output.

**0x0C0110F4 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS010****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS010**

Bits	Name	Description
7:0	L0_MISR_23_16	READ ONLY MISR output.

**0x0C0110F8 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS011****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS011**

Bits	Name	Description
7:0	L0_MISR_31_24	READ ONLY MISR output.

**0x0C0110FC PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS100****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS100**

Bits	Name	Description
7:0	L0_MISR_39_32	READ ONLY MISR output.

**0x0C011100 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS101****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS101**

Bits	Name	Description
7:0	L0_MISR_47_40	READ ONLY MISR output.

**0x0C011104 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS110****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS110**

Bits	Name	Description
7:0	L0_MISR_55_48	READ ONLY MISR output.

**0x0C011108 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS111****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS111**

Bits	Name	Description
7:0	L0_MISR_63_56	READ ONLY MISR output.

**0x0C01110C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS000****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS000**

Bits	Name	Description
7:0	L1_MISR_7_0	READ ONLY MISR output.

**0x0C011110 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS001****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS001**

Bits	Name	Description
7:0	L1_MISR_15_8	READ ONLY MISR output.

**0x0C011114 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS010****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS010**

Bits	Name	Description
7:0	L1_MISR_23_16	READ ONLY MISR output.

**0x0C011118 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS011****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS011**

Bits	Name	Description
7:0	L1_MISR_31_24	READ ONLY MISR output.

**0x0C01111C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS100****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS100**

Bits	Name	Description
7:0	L1_MISR_39_32	READ ONLY MISR output.

**0x0C011120 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS101****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS101**

Bits	Name	Description
7:0	L1_MISR_47_40	READ ONLY MISR output.

**0x0C011124 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS110****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS110**

Bits	Name	Description
7:0	L1_MISR_55_48	READ ONLY MISR output.

**0x0C011128 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS111****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS111**

Bits	Name	Description
7:0	L1_MISR_63_56	READ ONLY MISR output.

**0x0C01112C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS000****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS000**

Bits	Name	Description
7:0	L2_MISR_7_0	READ ONLY MISR output.

**0x0C011130 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS001****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS001**

Bits	Name	Description
7:0	L2_MISR_15_8	READ ONLY MISR output.

**0x0C011134 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS010****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS010**

Bits	Name	Description
7:0	L2_MISR_23_16	READ ONLY MISR output.

**0x0C011138 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS011****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS011**

Bits	Name	Description
7:0	L2_MISR_31_24	READ ONLY MISR output.

**0x0C01113C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS100****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS100**

Bits	Name	Description
7:0	L2_MISR_39_32	READ ONLY MISR output.

**0x0C011140 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS101****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS101**

Bits	Name	Description
7:0	L2_MISR_47_40	READ ONLY MISR output.

**0x0C011144 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS110****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS110**

Bits	Name	Description
7:0	L2_MISR_55_48	READ ONLY MISR output.

**0x0C011148 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS111****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS111**

Bits	Name	Description
7:0	L2_MISR_63_56	READ ONLY MISR output.

**0x0C01114C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS000****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS000**

Bits	Name	Description
7:0	L3_MISR_7_0	READ ONLY MISR output.

**0x0C011150 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS001****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS001**

Bits	Name	Description
7:0	L3_MISR_15_8	READ ONLY MISR output.

**0x0C011154 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS010****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS010**

Bits	Name	Description
7:0	L3_MISR_23_16	READ ONLY MISR output.

**0x0C011158 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS011****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS011**

Bits	Name	Description
7:0	L3_MISR_31_24	READ ONLY MISR output.

**0x0C01115C PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS100****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS100**

Bits	Name	Description
7:0	L3_MISR_39_32	READ ONLY MISR output.

**0x0C011160 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS101****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS101**

Bits	Name	Description
7:0	L3_MISR_47_40	READ ONLY MISR output.

**0x0C011164 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS10****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS10**

Bits	Name	Description
7:0	L3_MISR_55_48	READ ONLY MISR output.

**0x0C011168 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS11****Type:** R**Clock:** WCLK**Reset State:** 0x00000000**PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS11**

Bits	Name	Description
7:0	L3_MISR_63_56	READ ONLY MISR output.

**0x0C016008 PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_LOWER****Type:** RW**Clock:** hclk**Reset State:** 0x00000000

AHB2PHY\_BROADCAST\_EN\_CFG\_LOWER register configures the PHY chip select lower broadcast enables.

**PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_LOWER**

Bits	Name	Description
31:0	BROADCAST_EN_LOWER	ENABLE(1) DISABLE(0)  bit[0] : broadcast for the chip select 0 (o_phy_cs[0]) bit[1] : broadcast for the chip select 1 (o_phy_cs[1]) bit[2] : broadcast for the chip select 2 (o_phy_cs[2]) ..... bit[31] : broadcast for the chip select (o_phy_cs[31])

**0x0C01600C PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_UPPER****Type:** RW**Clock:** hclk**Reset State:** 0x00000000

AHB2PHY\_BROADCAST\_EN\_CFG\_UPPER register configures the PHY chip select upper broadcast enables.

**PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_UPPER**

Bits	Name	Description
29:0	BROADCAST_EN_UPPER	ENABLE(1) DISABLE(0)  bit[0] : broadcast for the chip select 0 (o_phy_cs[32]) bit[1] : broadcast for the chip select 1 (o_phy_cs[33]) bit[2] : broadcast for the chip select 2 (o_phy_cs[34]) ..... bit[29] : broadcast for the chip select (o_phy_cs[61])

**0x0C016010 PERIPH\_SS\_AHB2PHY\_TOP\_CFG****Type:** RW**Clock:** hclk**Reset State:** 0x00000000

AHB2PHY\_TOP\_CFG register configures AHB2PHY PHY wait states.

**PERIPH\_SS\_AHB2PHY\_TOP\_CFG**

Bits	Name	Description
5:4	WRITE_WAIT_STATES	0x0: NO_WAIT_STATES 0x1: ONE_WAIT_STATE 0x2: TWO_WAIT_STATES 0x3: THREE_WAIT_STATES
1:0	READ_WAIT_STATES	0x0: NO_WAIT_STATES 0x1: ONE_WAIT_STATE 0x2: TWO_WAIT_STATES 0x3: THREE_WAIT_STATES

**0x0C017000+ PERIPH\_SS\_AHB2PHY\_BROADCAST\_ADDRESS\_SPACE\_n, n=[0..255]  
0x4\*n**

**Type:** RW

**Clock:** hclk

**Reset State:** 0x00000000

A write or read access to this space activates all PHY chip selects that have their broadcast enable bit set to a 1.

**PERIPH\_SS\_AHB2PHY\_BROADCAST\_ADDRESS\_SPACE\_n**

Bits	Name	Description
31:0	BROADCAST_SPACE	The entire 1 Kbyte address space for this slave chip select is dedicated to the Broadcast function. If a write or read access is performed in this address space the chip selects of all PHY slaves that have been enabled will be active during the access. A PHY slave is enabled for broadcast by setting the BROADCAST_EN bit in the AHB2PHY_BROADCAST_EN_CFG register. Each PHY slave has a dedicated bit in this register.

# 8 Secure digital card registers

---

## 0x0C0C4000 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_0\_2

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

Argument 2 Register in SD Host Controller Standard Specification.

### PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_0\_2

Bits	Name	Description
31:0	ARG_2	Argument 2 field

## 0x0C0C4004 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_4\_6

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Block Size Register(15-0)

-Block Count Register(31-16)

### PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_4\_6

Bits	Name	Description
31:16	BLK_CNT_FOR_CUR_TRA_N	Blocks Count For Current Transfer field from Block Count Register
14:12	BLK_SIZE_HST_SDMA_BU_F	Host SDMA Buffer Boundary field from Block Size Register
11:00	BLK_SIZE_TRANS	Transfer Block Size field from Block Size Register

**0x0C0C4008 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_8\_A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Argument 1 Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_8\_A**

Bits	Name	Description
31:00	CMD_ARG_1	Command Argument 1 field

**0x0C0C400C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_C\_E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

- Transfer Mode Register(15-0)

- Command Register(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_C\_E**

Bits	Name	Description
29:24	CMD_INDX	Command Index field from Command Register
23:22	CMD_TYPE	Command Type field from Command Register
21	CMD_DATA_PRESENT_SEL	Data Present Select field from Command Register
20	CMD_INDX_CHECK_EN	Command Index Check Enable field from Command Register
19	CMD_CRC_CHECK_EN	Command CRC Check Enable field from Command Register
17:16	CMD_RESP_TYPE_SEL	Response Type Select field from Command Register
5	TRANS_MODE_MULTI_SINGLE_BLK_SEL	Multi/Single Block Select from Transfer Mode Register
4	TRANS_MODE_DATA_DIRECTION_SEL	Data Transfer Direction Select from Transfer Mode Register
3:2	TRANS_MODE_AUTO_CMD_EN	Auto CMD Enable from Transfer Mode Register
1	TRANS_MODE_BLK_CNT_EN	Block Count Enable field from Transfer Mode Register
0	TRANS_MODE_DMA_EN	DMA Enable field from Transfer Mode Register

**0x0C0C4010 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_10\_12****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_10\_12**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Response bits 0-31

**0x0C0C4014 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_14\_16****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_14\_16**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 31-63

**0x0C0C4018 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_18\_1A****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_18\_1A**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 64-95

**0x0C0C401C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_1C\_1E****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_1C\_1E**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 96-127

**0x0C0C4020 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_20\_22****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Buffer Data Port Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_20\_22**

Bits	Name	Description
31:24	BUF_DATA_PORT_3	Byte 3 of Buffer Data
23:16	BUF_DATA_PORT_2	Byte 2 of Buffer Data
15:8	BUF_DATA_PORT_1	Byte 1 of Buffer Data
7:0	BUF_DATA_PORT_0	Byte 0 of Buffer Data

**0x0C0C4024 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_24\_26****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x01F800F0

Present State Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_24\_26**

Bits	Name	Description
25	SIGANLING_18_SWITCHING_STS	Bit it is cleared, by hardware, when host driver writes any value to 1.8V Signaling Enable field, indicating that a signal switching operation has started. Bit is set, by hardware, when switching operation succeeds.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_24\_26 (cont.)**

Bits	Name	Description
24	PRESENT_STATE_CMD_LINR_SIGNAL_LEVEL	CMD Linr Signal Level field
23:20	PRESENT_STATE_DAT_3_0_LINE_SIGNAL_LEVEL	DAT[3:0] Line Signal Level field
19	PRESENT_STATE_WR_PROTTECT_SWITCH_PIN_LEVEL	Write Protect Switch Pin Level field
18	PRESENT_STATE_CARD_DETECT_PIN_LEVEL	Card Detect Pin Level field
17	PRESENT_STATE_CARD_STABLE	Card State Stable field
16	PRESENT_STATE_CARD_INSERTED	Card Inserted field
11	PRESENT_STATE_BUF_RD_EN	Buffer Read Enable field
10	PRESENT_STATE_BUF_W_R_EN	Buffer Write Enable field
9	PRESENT_STATE_RD_TRANS_ACT	Read Transfer Active field
8	PRESENT_STATE_WR_TRANS_ACT	Write Transfer Active field
7:4	PRESENT_STATE_DAT_7_4_LINE_SIGNAL_LEVEL	DAT[7:4] Line Signal Level field
3	PRESENT_STATE_RETUNING_REQ	Re-Tuning Request field
2	PRESENT_STATE_DAT_LIN_E_ACT	DAT Linr Active field
1	PRESENT_STATE_CMD_INHIBIT_DAT	Command Inhibit (DAT) field
0	PRESENT_STATE_CMD_INHIBIT_CMD	Command Inhibit (CMD) field

**0x0C0C4028 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_28\_2A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of four registers from the SD Host Controller Standard Specification:

-Host Control 1 Register(7-0)

-Power Control Register(15-8)

-Block Gap Control Register(23-16)

-Wakeup Control Register(31-24)

### **PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_28\_2A**

Bits	Name	Description
26	WKUP_EVENT_EN_ON_SD_CARD_REMOVAL	Wakeup Event Enable On SD Card Removal field from Wakeup Control Register
25	WKUP_EVENT_EN_ON_SD_CARD_INSERTION	Wakeup Event Enable On SD Card Insertion field from Wakeup Control Register
24	WKUP_EVENT_EN_ON_SD_CARD_INT	Wakeup Event Enable On Card Interrupt field from Wakeup Control Register
19	BLK_GAP_CTL_INT	Interrupt At Block Gap field from Block Gap Control Register
18	BLK_GAP_CTL_RD_WAIT	Read Wait Control field from Block Gap Control Register
17	BLK_GAP_CTL_CONTINUE_REQ	Continue Request field from Block Gap Control Register
16	BLK_GAP_CTL_STOP_GAP_REQ	Stop At Block Gap Request field from Block Gap Control Register
11:9	PWR_CTL_SD_BUS_VOLTA GE_SEL	SD Bus Voltage Select field from Power Control Register
8	PWR_CTL_SD_BUS_PWR	SD Bus Power field from Power Control Register
7	HST_CTL1_CARD_DETECT_SIGNAL_SEL	Card Detect Signal Selection field from Host Control 1 Register
6	RESERVED	RESERVED
5	HST_CTL1_EXTENDED_DA TA_TRANS_WIDTH	Extended Data Transfer Width field from Host Control 1 Register
4:3	HST_CTL1_DMA_SEL	DMA Select field from Host Control 1 Register
2	HST_CTL1_HS_EN	High Speed Enable field from Host Control 1 Register
1	HST_CTL1_DATA_TRANS_WIDTH	Data Transfer Width field from Host Control 1 Register
0	HST_CTL1_LED_CTL	LED Control field from Host Control 1 Register

### **0x0C0C402C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_2C\_2E**

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000001

Combination of three registers from the SD Host Controller Standard Specification:

-Clock Control Register(15-0)

-Timeout Control Register(23-16)

-Software Reset Register(31-24)

### **PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_2C\_2E**

Bits	Name	Type	Description
26	SW_RST_DAT_LINE	RW	Software Reset For DAT Line field from Software Reset Register
25	SW_RST_CMD_LINE	RW	Software Reset For CMD Line field from Software Reset Register
24	SW_RST_FOR_ALL	RW	Software Reset For All field from Software Reset Register
19:16	DATA_TIMEOUT_COUNTE_R	RW	Data Timeout Counter Value field from Timeout Control Register. The reserved value 0xF is translated to maximum number of cycles - 2^32-1
15:8	CLK_CTL_SDCLK_FREQ_SEL	RW	SDCLK Frequency Select field from Clock Control Register
7:6	CLK_CTL_SDCLK_FREQ_SEL_MSB	RW	Upper Bits of SDCLK Frequency Select field from Clock Control Register
5	CLK_CTL_GEN_SEL	RW	Clock Generator Select field from Clock Control Register
2	CLK_CTL_SDCLK_EN	RW	SD Clock Enable field from Clock Control Register
1	CLK_CTL_INTERNAL_CLK_STABLE	R	Internal Clock Stable field from Clock Control Register
0	CLK_CTL_INTERNAL_CLK_EN	RW	Internal Clock Enable field from Clock Control Register

### **0x0C0C4030 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_30\_32**

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Normal Interrupt Status Register(15-0)

-Error Interrupt Status Register(31-16)

### **PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_30\_32**

Bits	Name	Type	Description
31	ERR_INT_STS_VENDOR_SPECIFIC_ERR	RW	Vendor Specific Error Status field from Error Interrupt Status Register
30	ERR_INT_STS_VENDOR_SPECIFIC_STAT	RW	Vendor Specific Error/Status field from HC_VENDOR_SPECIFIC_INT_STS register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_30\_32 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
26	ERR_INT_STS_TUNING_E RR	RW	Tuning Error field from Error Interrupt Status Register
25	ERR_INT_STS_ADMA_ERR	RW	ADMA Error field from Error Interrupt Status Register
24	ERR_INT_STS_AUTO_CMD_ _ERR	RW	Auto CMD Error field from Error Interrupt Status Register
23	ERR_INT_STS_CURRENT_ LIMIT_ERR	RW	Current Limit Error field from Error Interrupt Status Register
22	ERR_INT_STS_DATA_END_ _BIT_ERR	RW	Data End Bit Error field from Error Interrupt Status Register
21	ERR_INT_STS_DATA_CRC_ _ERR	RW	Data CRC Error field from Error Interrupt Status Register
20	ERR_INT_STS_DATA_TIME OUT_ERR	RW	Data Timeout Error field from Error Interrupt Status Register
19	ERR_INT_STS_CMD_INDX_ _ERR	RW	Command Index Error field from Error Interrupt Status Register
18	ERR_INT_STS_CMD_END_ BIT_ERR	RW	Command End Bit Error field from Error Interrupt Status Register
17	ERR_INT_STS_CMD_CRC_ _ERR	RW	Command CRC Error field from Error Interrupt Status Register
16	ERR_INT_STS_CMD_TIME OUT_ERR	RW	Command Timeoutout Error field from Error Interrupt Status Register
15	NORMAL_INT_STS_ERR_I NT	R	Error Interrupt field from Normal Interrupt Status Register
14	NORMAL_INT_STS_CMD_ QUEUE	R	Command Queueing Event field from Normal Interrupt Status Register
12	NORMAL_INT_STS_RETUN ING_EVENT	R	Re-Tuning Event field from Normal Interrupt Status Register
11	NORMAL_INT_STS_INT_C	R	INT_C field from Normal Interrupt Status Register
10	NORMAL_INT_STS_INT_B	R	INT_B field from Normal Interrupt Status Register
9	NORMAL_INT_STS_INT_A	R	INT_A field from Normal Interrupt Status Register
8	NORMAL_INT_STS_CARD_ INT	R	Card Interrupt field from Normal Interrupt Status Register
7	NORMAL_INT_STS_CARD_ REMOVAL	RW	Card Removal field from Normal Interrupt Status Register
6	NORMAL_INT_STS_CARD_ INSERTION	RW	Card Insertion field from Normal Interrupt Status Register
5	NORMAL_INT_STS_BUF_R D_READY	RW	Buffer Read Ready field from Normal Interrupt Status Register
4	NORMAL_INT_STS_BUF_ WR_READY	RW	Buffer Write Ready field from Normal Interrupt Status Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_30\_32 (cont.)**

Bits	Name	Type	Description
3	NORMAL_INT_STS_DMA_INTERRUPT	RW	DMA Interrupt field from Normal Interrupt Status Register
2	NORMAL_INT_STS_BLK_GAP_EVENT	RW	Block Gap Event field from Normal Interrupt Status Register
1	NORMAL_INT_STS_TRANS_COMPLETE	RW	Transfer Complete field from Normal Interrupt Status Register
0	NORMAL_INT_STS_CMD_COMPLETE	RW	Command Complete field from Normal Interrupt Status Register

**0x0C0C4034 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_34\_36****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

- Normal Interrupt Status Enable Register(15-0)
- Error Interrupt Status Enable Register(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_34\_36**

Bits	Name	Description
31	ERR_INT_STS_EN_VENDOR_SPECIFIC_ERR	Vendor Specific Error Status Enable field from Error Interrupt Status Enable Register
30	ERR_INT_STS_EN_VENDOR_SPECIFIC_STAT	Vendor Specific Error/Status Enable field for ERR_INT_STS_VENDOR_SPECIFIC_STAT interrupt
26	ERR_INT_STS_EN_TUNING_ERR	Tuning Error Status Enable field from Error Interrupt Status Enable Register
25	ERR_INT_STS_EN_ADMA_ERR	ADMA Error Status Enable field from Error Interrupt Status Enable Register
24	ERR_INT_STS_EN_AUTO_CMD_ERR	Auto CMD Error Status Enable field from Error Interrupt Status Enable Register
23	ERR_INT_STS_EN_CURRENT_LIMIT_ERR	Current Limit Error Status Enable field from Error Interrupt Status Enable Register
22	ERR_INT_STS_EN_DATA_END_BIT_ERR	Data End Bit Error Status Enable field from Error Interrupt Status Enable Register
21	ERR_INT_STS_EN_DATA_CRC_ERR	Data CRC Error Status Enable field from Error Interrupt Status Enable Register
20	ERR_INT_STS_EN_DATA_TIMEOUT	Data Timeout Error Status Enable field from Error Interrupt Status Enable Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_34\_36 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
19	ERR_INT_STS_EN_CMD_INDX_ERR	Command Index Error Status Enable field from Error Interrupt Status Enable Register
18	ERR_INT_STS_EN_CMD-END_BIT_ERR	Command End Bit Error Status Enable field from Error Interrupt Status Enable Register
17	ERR_INT_STS_EN_CMD_CRC_ERR	Command CRC Error Status Enable field from Error Interrupt Status Enable Register
16	ERR_INT_STS_EN_CMD_TIMEOUT	Command Timeoutout Error Status Enable field from Error Interrupt Status Enable Register
14	NORMAL_INT_STS_EN_CMD_QUEUE	Command Queueing Event Status Enable field from Normal Interrupt Status Enable Register
12	NORMAL_INT_STS_EN_RETUNING_EVENT	Re-Tuning Event Status Enable field from Normal Interrupt Status Enable Register
11	NORMAL_INT_STS_EN_INT_C	INT_C Status Enable field from Normal Interrupt Status Enable Register
10	NORMAL_INT_STS_EN_INT_B	INT_B Status Enable field from Normal Interrupt Status Enable Register
9	NORMAL_INT_STS_EN_INT_A	INT_A Status Enable field from Normal Interrupt Status Enable Register
8	NORMAL_INT_STS_EN_CARD_INT	Card Interrupt Status Enable field from Normal Interrupt Status Enable Register
7	NORMAL_INT_STS_EN_CARD_REMOVAL	Card Removal Status Enable field from Normal Interrupt Status Enable Register
6	NORMAL_INT_STS_EN_CARD_INSERTION	Card Insertion Status Enable field from Normal Interrupt Status Enable Register
5	NORMAL_INT_STS_EN_BUFRD_READY	Buffer Read Ready Status Enable field from Normal Interrupt Status Enable Register
4	NORMAL_INT_STS_EN_BUFWR_READY	Buffer Write Ready Status Enable field from Normal Interrupt Status Enable Register
3	NORMAL_INT_STS_EN_DMA_INT	DMA Interrupt Status Enable field from Normal Interrupt Status Enable Register
2	NORMAL_INT_STS_EN_BLOCK_GAP_EVENT	Block Gap Event Status Enable field from Normal Interrupt Status Enable Register
1	NORMAL_INT_STS_EN_TRANS_COMPLETE	Transfer Complete Status Enable field from Normal Interrupt Status Enable Register
0	NORMAL_INT_STS_EN_CMD_COMPLETE	Command Complete Status Enable field from Normal Interrupt Status Enable Register

**0x0C0C4038 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_38\_3A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

- Normal Interrupt Signal Enable Register(15-0)

- Error Interrupt Signal Enable Register(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_38\_3A**

Bits	Name	Type	Description
31	ERR_INT_SIGNAL_EN_VEN_DOR_SPECIFIC_ERR	RW	Vendor Specific Error Signal Enable field from Error Interrupt Signal Enable Register
30	ERR_INT_SIGNAL_EN_VEN_DOR_SPECIFIC_STAT	RW	Vendor Specific Error Signal Enable field for ERR_INT_STS_VENDOR_SPECIFIC_STAT interrupt
26	ERR_INT_SIGNAL_EN_TUNING_ERR	RW	Tuning Error Signal Enable field from Error Interrupt Signal Enable Register
25	ERR_INT_SIGNAL_EN_ADMA_ERR	RW	ADMA Error Signal Enable field from Error Interrupt Signal Enable Register
24	ERR_INT_SIGNAL_EN_AUTO_CMD_ERR	RW	Auto CMD Error Signal Enable field from Error Interrupt Signal Enable Register
23	ERR_INT_SIGNAL_EN_CURRENT_LIMIT_ERR	RW	Current Limit Error Signal Enable field from Error Interrupt Signal Enable Register
22	ERR_INT_SIGNAL_EN_DATA_END_BIT_ERR	RW	Data End Bit Error Signal Enable field from Error Interrupt Signal Enable Register
21	ERR_INT_SIGNAL_EN_DATA_CRC_ERR	RW	Data CRC Error Signal Enable field from Error Interrupt Signal Enable Register
20	ERR_INT_SIGNAL_EN_DATA_TIMEOUT	RW	Data Timeout Error Signal Enable field from Error Interrupt Signal Enable Register
19	ERR_INT_SIGNAL_EN_COMMAND_INDEX_ERR	RW	Command Index Error Signal Enable field from Error Interrupt Signal Enable Register
18	ERR_INT_SIGNAL_EN_COMMAND_END_BIT_ERR	RW	Command End Bit Error Signal Enable field from Error Interrupt Signal Enable Register
17	ERR_INT_SIGNAL_EN_COMMAND_CRC_ERR	RW	Command CRC Error Signal Enable field from Error Interrupt Signal Enable Register
16	ERR_INT_SIGNAL_EN_COMMAND_TIMEOUT	RW	Command Timeout Error Signal Enable field from Error Interrupt Signal Enable Register
15	NORMAL_INT_SIGNAL_EN_ERROR_INTERRUPT	R	Error Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
14	NORMAL_INT_SIGNAL_EN_COMMAND_QUEUE	RW	Command Queueing Signal Enable field from Normal Interrupt Signal Enable Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_38\_3A (cont.)**

Bits	Name	Type	Description
12	NORMAL_INT_SIGNAL_EN _RETUNING_EVENT	RW	Re-Tuning Event Signal Enable field from Normal Interrupt Signal Enable Register
11	NORMAL_INT_SIGNAL_EN _INT_C	RW	INT_C Signal Enable field from Normal Interrupt Signal Enable Register
10	NORMAL_INT_SIGNAL_EN _INT_B	RW	INT_B Signal Enable field from Normal Interrupt Signal Enable Register
9	NORMAL_INT_SIGNAL_EN _INT_A	RW	INT_A Signal Enable field from Normal Interrupt Signal Enable Register
8	NORMAL_INT_SIGNAL_EN _CARD_INT	RW	Card Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
7	NORMAL_INT_SIGNAL_EN _CARD_REMOVAL	RW	Card Removal Signal Enable field from Normal Interrupt Signal Enable Register
6	NORMAL_INT_SIGNAL_EN _CARD_INSERTION	RW	Card Insertion Signal Enable field from Normal Interrupt Signal Enable Register
5	NORMAL_INT_SIGNAL_EN _BUF_RD_READY	RW	Buffer Read Ready Signal Enable field from Normal Interrupt Signal Enable Register
4	NORMAL_INT_SIGNAL_EN _BUF_WR_READY	RW	Buffer Write Ready Signal Enable field from Normal Interrupt Signal Enable Register
3	NORMAL_INT_SIGNAL_EN _DMA_INT	RW	DMA Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
2	NORMAL_INT_SIGNAL_EN _BLK_GAP_EVENT	RW	Block Gap Event Signal Enable field from Normal Interrupt Signal Enable Register
1	NORMAL_INT_SIGNAL_EN _TRANS_COMPLETE	RW	Transfer Complete Signal Enable field from Normal Interrupt Signal Enable Register
0	NORMAL_INT_SIGNAL_EN _CMD_COMPLETE	RW	Command Complete Signal Enable field from Normal Interrupt Signal Enable Register

**0x0C0C403C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_3C\_3E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Auto CMD Error Status Register(15-0)

-Host Control 2 Register(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_3C\_3E**

Bits	Name	Type	Description
31	HST_CTL2_PRESET_VALUE_EN	RW	Preset Value Enable field from Host Control2 Register
30	HST_CTL2_ASYNC_INT_EN	RW	Asynchronous Interrupt Enable field from Host Control2 Register
23	HST_CTL2_SAMPL_CLK_SEL	RW	Sampling Clock Select field from Host Control2 Register
22	HST_CTL2_EXEC_TUNING	RW	Execute Tuning field from Host Control2 Register
21:20	HST_CTL2_DRIVER_STRENGTH_SEL	RW	Driver Strength Select field from Host Control2 Register
19	HST_CTL2_SIGNALING_1_8_EN	RW	1.8V Signaling Enable field from Host Control2 Register
18:16	HST_CTL2_UHS_MODE_SEL	RW	UHS Mode Select field from Host Control2 Register
7	AUTO_CMD_NOT_ISSUED_BY_AUTO_CMD12	R	Command Not issued By Auto CMD12 Error field from Auto CMD Error Status Register
4	AUTO_CMD_IDX_ERR	R	Auto CMD Index Error field from Auto CMD Error Status Register
3	AUTO_CMD_END_BIT_ERR	R	Auto CMD End Bit Error field from Auto CMD Error Status Register
2	AUTO_CMD_CRC_ERR	R	Auto CMD CRC Error field from Auto CMD Error Status Register
1	AUTO_CMD_TIMEOUT	R	Auto CMD Timeout Error field from Auto CMD Error Status Register
0	AUTO_CMD12_NOT_EXEC	R	Auto CMD12 Not Executed field from Auto CMD Error Status Register

**0x0C0C4040 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_40\_42****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x3029C8B2

The 32 LSBs of the Capabilities Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_40\_42**

Bits	Name	Description
31:30	CAPABILITIES_SLOT_TYPE	Slot Type field from Capabilities Register
29	CAPABILITIES_ASYNC_INT_SUPPORT	Asynchronous Interrupt Support field from Capabilities Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_40\_42 (cont.)**

Bits	Name	Description
28	CAPABILITIES_SYS_BUS_SUPPORT_64_BIT	64 bit System Bus Support field from Capabilities Register
26	CAPABILITIES_VOLTAGE_SUPPORT_1_8V	Voltage Support 1.8v field from Capabilities Register
25	CAPABILITIES_VOLTAGE_SUPPORT_3_0V	Voltage Support 3.0v field from Capabilities Register
24	CAPABILITIES_VOLTAGE_SUPPORT_3_3V	Voltage Support 3.3v field from Capabilities Register
23	CAPABILITIES_SUSPEND_RESUME_SUPPORT	Suspend Resume Support field from Capabilities Register
22	CAPABILITIES_SDMA_SUPPORT	SDMA Support field from Capabilities Register
21	CAPABILITIES_HS_SUPPORT	High Speed Support field from Capabilities Register
19	CAPABILITIES_ADMA2_SUPPORT	ADMA2 Support field from Capabilities Register
18	CAPABILITIES_SUPPORT_8_BIT	8 bit Supported For Embedded Device field from Capabilities Register
17:16	CAPABILITIES_MAX_BLK_LENGTH	Max Block Length field from Capabilities Register
15:8	CAPABILITIES_BASE_SDCLK_FREQ	Base Clock Frequency For SD Clock field from Capabilities Register
7	CAPABILITIES_TIMEOUT_CLOCK_UNIT	Timeout Clock Unit field from Capabilities Register
5:0	CAPABILITIES_TIMEOUT_CLOCK_FREQ	Timeout Clock Frequency field from Capabilities Register

**0x0C0C4044 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_44\_46****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00008007

The 32 MSBs of the Capabilities Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_44\_46**

Bits	Name	Description
23:16	CAPABILITIES_CLK_MULTIPLIER	Clock Multiplier field from Capabilities Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_44\_46 (cont.)**

Bits	Name	Description
15:14	CAPABILITIES_RETUNING_MODE	Re-Tuning Modes field from Capabilities Register
13	CAPABILITIES_USE_TUNING_FOR_SDR50	Use Tuning for SDR50 field from Capabilities Register
11:8	CAPABILITIES_TIMER_CNT_FOR_RETUNING	Timer Count for Re-Tuning field from Capabilities Register
6	CAPABILITIES_DRIVER_TYPE_D_SUPPORT	Driver Type D Support field from Capabilities Register
5	CAPABILITIES_DRIVER_TYPE_C_SUPPORT	Driver Type C Support field from Capabilities Register
4	CAPABILITIES_DRIVER_TYPE_A_SUPPORT	Driver Type A Support field from Capabilities Register
2	CAPABILITIES_DDR_50_SUPPORT	DDR50 Support field from Capabilities Register
1	CAPABILITIES_SDR_104_SUPPORT	SDR104 Support field from Capabilities Register
0	CAPABILITIES_SDR_50_SUPPORT	SDR50 Support field from Capabilities Register

**0x0C0C4048 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_48\_4A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Maximum Current Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_48\_4A**

Bits	Name	Description
23:16	MAX_CURRENT_1_8V	Maximum Current for 1.8v field from Maximum Current Capabilities Register
15:08	MAX_CURRENT_3_0V	Maximum Current for 3.0v field from Maximum Current Capabilities Register
07:00	MAX_CURRENT_3_3V	Maximum Current for 3.3v field from Maximum Current Capabilities Register

**0x0C0C4050 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_50\_52****Type:** W**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

- Force Event Register for Auto CMD Error Status(15-0)

- Force Event Register for Error Interrupt Status(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_50\_52**

Bits	Name	Description
31:28	FORCE_EVENT_FOR_ERR_VENDOR_SPECIFIC_ERR_STS	Force Event for Vendor Specific Error Status field from Force Event Register for Error Interrupt Status
25	FORCE_EVENT_FOR_ERR_ADMA	Force Event for ADMA Error field from Force Event Register for Error Interrupt Status
24	FORCE_EVENT_FOR_ERR_AUTO_CMD	Force Event for Auto CMD Error field from Force Event Register for Error Interrupt Status
23	FORCE_EVENT_FOR_ERR_CURRENT_LIMIT	Force Event for Current Limit Error field from Force Event Register for Error Interrupt Status
22	FORCE_EVENT_FOR_ERR_DATA_END_BIT	Force Event for Data End Bit Error field from Force Event Register for Error Interrupt Status
21	FORCE_EVENT_FOR_ERR_DATA_CRC	Force Event for Data CRC Error field from Force Event Register for Error Interrupt Status
20	FORCE_EVENT_FOR_ERR_DATA_TIMEOUT	Force Event for Data Timeout Error field from Force Event Register for Error Interrupt Status
19	FORCE_EVENT_FOR_ERR_CMD_INDX	Force Event for Command Index Error field from Force Event Register for Error Interrupt Status
18	FORCE_EVENT_FOR_ERR_CMD_END_BIT	Force Event for Command End Bit Error field from Force Event Register for Error Interrupt Status
17	FORCE_EVENT_FOR_ERR_CMD_CRC	Force Event for Command CRC Error field from Force Event Register for Error Interrupt Status
16	FORCE_EVENT_FOR_ERR_CMD_TIMEOUT	Force Event for Command Timeout Error field from Force Event Register for Error Interrupt Status
7	FORCE_EVENT_FOR_CMD_NOT_ISSUED_BY_AUTO_CMD12	Force Event for Command Not Issued By Auto CMD12 Error field from Force Event Register for Auto CMD Error Status
4	FORCE_EVENT_FOR_AUTO_CMD_INDX_ERR	Force Event for Auto CMD Index Error field from Force Event Register for Auto CMD Error Status
3	FORCE_EVENT_FOR_AUTO_CMD_END_BIT_ERR	Force Event for Auto CMD End Bit Error field from Force Event Register for Auto CMD Error Status

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_50\_52 (cont.)**

Bits	Name	Description
2	FORCE_EVENT_FOR_AUTO_CMD_CRC_ERR	Force Event for Auto CMD CRC Error field from Force Event Register for Auto CMD Error Status
1	FORCE_EVENT_FOR_AUTO_CMD_TIMEOUT_ERR	Force Event for Auto CMD Timeout Error field from Force Event Register for Auto CMD Error Status
0	FORCE_EVENT_FOR_AUTO_CMD12_NOT_EXEC	Force Event for Auto CMD12 Not Executed field from Force Event Register for Auto CMD Error Status

**0x0C0C4054 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_54\_56****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 7 LSBs are the ADMA Error Status Register in SD Host Controller Standard Specification.  
The Bits 31-8 are unused.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_54\_56**

Bits	Name	Description
2	ADMA_LENGTH_MISMATCH_H_ERR	ADMA Length Mismatch Error field from ADMA Error Status Register
1:0	ADMA_ERR_STATE	ADMA Error State field from ADMA Error Status Register

**0x0C0C4058 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_58\_5A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 32 LSBs of the ADMA System Address Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_58\_5A**

Bits	Name	Description
31:00	ADMA_SYS_ADDRESS	ADMA System Address field from ADMA System Address Register

**0x0C0C405C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_5C\_5E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 32 MSBs of the ADMA System Address Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_5C\_5E**

Bits	Name	Description
31:00	ADMA_SYS_ADDRESS_64	upper 32 bits of ADMA System Address field from ADMA System Address Register

**0x0C0C4060 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_60\_62****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00040000

Combination of two registers from the SD Host Controller Standard Specification:

- Preset Value for Initialization Register(15-0)

- Preset Value for Default Speed Register(31-16)

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_60\_62**

Bits	Name	Description
31:30	DEFAULT_SPEED_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for Default Speed
26	DEFAULT_SPEED_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for Default Speed
25:16	DEFAULT_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for Default Speed
15:14	INIT_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for Initialization
10	INIT_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for Initialization
9:0	INIT_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for Initialization

**0x0C0C4064 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_64\_66****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00040002

Combination of two registers from the SD Host Controller Standard Specification:

- Preset Value for High Speed Register(15-0)

- Preset Value for SDR12 Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_64\_66**

Bits	Name	Description
31:30	SDR12_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR12
26	SDR12_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR12
25:16	SDR12_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR12
15:14	HS_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for High Speed
10	HS_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for High Speed
9:0	HS_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for High Speed

**0x0C0C4068 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_68\_6A****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00010002

Combination of two registers from the SD Host Controller Standard Specification:

- Preset Value for SDR25 Register(15-0)

- Preset Value for SDR50 Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_68\_6A**

Bits	Name	Description
31:30	SDR50_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR50
26	SDR50_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR50
25:16	SDR50_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR50

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_68\_6A (cont.)**

Bits	Name	Description
15:14	SDR25_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR25
10	SDR25_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR25
9:0	SDR25_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR25

**0x0C0C406C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_6C\_6E****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00020000

Combination of two registers from the SD Host Controller Standard Specification:

-Preset Value for SDR104 Register(15-0)

-Preset Value for DDR50 Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_6C\_6E**

Bits	Name	Description
31:30	DDR50_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for DDR50
26	DDR50_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for DDR50
25:16	DDR50_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for DDR50
15:14	SDR104_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR104
10	SDR104_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR104
9:0	SDR104_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR104

**0x0C0C40E0 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_E0\_E2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Shared Bus Control Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_E0\_E2**

Bits	Name	Type	Description
30:24	SHARED_BUS_CTL_BACK_END_PWR	RW	Back End Power Control field from Shared Bus Control Register
22:20	SHARED_BUS_INT_PIN_SEL	RW	
18:16	SHARED_BUS_CTL_CLK_PIN_SEL	RW	Clock Pin Select field from Shared Bus Control Register
14:8	SHARED_BUS_CTL_BUS_WIDTH_PRESET	R	Bus Width Preset field from Shared Bus Control Register
5:4	SHARED_BUS_CTL_NUM_INTERRUPT_INPUT_PINS	R	Number Of Interrupt Input Pins field from Shared Bus Control Register
2:0	SHARED_BUS_CTL_NUM_CLK_PINS	R	Number of Clock Pins field from Shared Bus Control Register

**0x0C0C4200 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x6000642C

This register configures the DLL's inputs.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG**

Bits	Name	Description
31	SDC4_DIS_DOUT	When this bit equals to '0', the data which comes out from the dll is valid. When it equals to '1' this data isn't valid. The inversion of this bit is connected to dll's input - dll_en_dout
30	DLL_RST	Setting this bit to 1 resets cm_dll_sdc4. cm_dll_sdc4 should be reset every time the MCLK frequency is changed.
29	PDN	Power Down Value 0 - analog blocks are enabled Value 1 - analog blocks are powered down (default)
28	CK_INTP_SEL	Selects interpolator output
27	CK_INTP_EN	Enable clock interpolation for finer resolution

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
26:24	MCLK_FREQ	Frequency of MCLK Value 000 -100 - 112 (MHz) Value 001 -112 - 125 Value 010 -125 - 137 Value 011 -137 - 150 Value 100 -150 - 162 Value 101 -162 - 175 Value 110 -175 - 187 Value 111 -187 - 200
23:20	CDR_SELEXT	CDR phase select (gray coded) Value 0000 - phase 0 Value 0001 - phase 1 Value 0011 - phase 2 Value 0010 - phase 3 Value 0110 - phase 4 Value 0111 - phase 5 Value 0101 - phase 6 Value 0100 - phase 7 Value 1100 - phase 8 Value 1101 - phase 9 Value 1111 - phase 10 Value 1110 - phase 11 Value 1010 - phase 12 Value 1011 - phase 13 Value 1001 - phase 14 Value 1000 - phase 15
19	CDR_EXT_EN	Enable external control of cdr phase select
18	CK_OUT_EN	Enable output clock (default value is '1')
17	CDR_EN	Enable CDR function
16	DLL_EN	Enable DLL function
15:14	CDR_UPD_RATE	CDR update rate, low pass filtering window of CDR Max update rate: Value 00 - 0.5 MCLK frequency Value 01 - 0.25 MCLK Frequency (default) Value 10 - 0.125 MCLK frequency Value 11 - 1/16 MCLK Frequency
13:12	DLL_UPD_RATE	DLL update rate Value 00 - sdc4_mclk/10 Value 01 - sdc4_mclk/20 Value 10 - sdc4_mclk/40 (default) Value 11 - sdc4_mclk/80

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG (cont.)**

Bits	Name	Description
11:10	DLL_PHASE_DET	DLL phase detector low pass average window Value 00 - 2 cycles Value 01 - 4 cycles (default) Value 10 - 8 cycles Value 11 - 16 cycles
9	CDR_ALGORITHM_SEL	CDR algorithm select Value 0 - Edge (default) Value 1 - Level
8	DLY_LINE_SWITCH_CLK	value 0 (default) - Uses MCLK for switching between two delay-lines in programmable RCLK delay generation. value 1 - Uses TCXO clock for interpolating between two delay-lines in programmable RCLK delay generation.
7:6	CDR_PHASE_SEL_MODE	00 (default) - CDR phase is selected using all 8 input data 01 - CDR phase is selected using DATA_IN<1> only 10 - CDR phase is selected using DATA_IN<2> only 11 - CDR phase is selected using DATA_IN<3> only
5	MCLK_GATING_ENABLE	value '1' (default) - MCLK gating Enabled. '0' - No MCLK gating
4	FINE_PHASE_ENABLE	value '0' (default) - fine phase mode disabled. '1' - enable fine phase mode
3:2	CDR_FINE_PHASE	Default Value = 11 When fine phase mode is enabled, these two bits together with SDC4_CDR_SELEXT are used to generate the 64 phases Selected phase : 4*SDC4_CDR_SELEXT(Decimal Value)+ DECIMAL Equivalent of CONFIG<2:3>
1	RESERVED	RESERVED
0	CMD_DAT_TRACK_SEL	Select DAT or CMD lines for CDR tracking Value 0 - track DAT inputs (default) Value 1 - track CMD input

**0x0C0C4208 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_STATUS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

This register is connected to cm\_dll\_sdc4's status output.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_STATUS**

Bits	Name	Description
12	RESERVED	RESERVED

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_STATUS (cont.)**

Bits	Name	Description
11	DDR_DLL_LOCK_JDR	DLL lock for HS400 operation
10:9	RESERVED	RESERVED
8	SDC4_DLL_LOCK_ATPG	SDC4_DLL_LOCK output from DLL after a clock mux with sampled JDRs for ATPG mode. This bit helps collecting coverage on DLL's output
7	DLL_LOCK	DLL lock status Value 0 - Not locked Value 1 - Locked
6:3	CDR_PHASE	CDR phase select (gray coded) Value 0000 - phase 0 Value 0001 - phase 1 Value 0011 - phase 2 Value 0010 - phase 3 Value 0110 - phase 4 Value 0111 - phase 5 Value 0101 - phase 6 Value 0100 - phase 7 Value 1100 - phase 8 Value 1101 - phase 9 Value 1111 - phase 10 Value 1110 - phase 11 Value 1010 - phase 12 Value 1011 - phase 13 Value 1001 - phase 14 Value 1000 - phase 15
2	DDLL_COARSE_CAL	Value 1 - done Value 0 - not done
0	DDR_DLL_LOCK	DLL lock for HS400 operation

**0x0C0C4210 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0xF88218A8**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2**

Bits	Name	Type	Description
31	QSB_AXI_ABURST	RW	Driving QSB AXI output
30:28	NUM_OUTSTANDING_DATA	RW	Configuring the number of outstanding transactions for the data. The descriptors accesses have more one outstanding request

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
27	PROCEED_AXI_AFTER_ER_R	RW	When error occurs on AXI master the traffic can continue or terminate. If this bit is set then the traffic continues and more requests are transmitted on the address channel
26	QSB_AXI_AFULL_CALC	RW	If this bit is set, then afull isn't '0' constantly but can be equal to '1' if all the beats have 8 bytes on the strobe lines
25	ONE_MID_SUPPORT	RW	If this bit is set, then only 1 MID is used for descriptors and data
24:22	QSB_AXI_READ_MEMTYP_E	RW	Driving VMIDMT's input for read accesses
21	HC_SW_RST_REQ	RW	Before executing HC_SW_RST_FOR_ALL or HC_SW_RST_FOR_DAT this bit can be set and SW needs to polls on it until it's being cleared. The bit is cleared after AXI master (QMB) has finished processing the outstanding transactions and it's in IDLE state. This bit can be useful for HPI feature.
20	HC_SW_RST_WAIT_IDLE_DIS	RW	When this bit is cleared, the HC_SW_RST is executed only after the AXI master (QMB) is in IDLE state. This bit Can be used for HPI feature instead of polling on HC_SW_RST_REQ bit. This bit should be cleared for SW reset from unknown states.
19	SDCC5_HALT_REQ	R	Reading SDCC5_HALT_REQ input
18	SDCC5_HALT_ACK	R	Reading SDCC5_HALT_ACK output
17	SDCC5_M_IDLE	R	Reading SDCC5_M_IDLE output
16	SDCC5_HALT_ACK_SW_EN	RW	Enabling the SW value for SDCC5_HALT_ACK
15	SDCC5_HALT_ACK_SW	RW	Driving SDCC5_HALT_ACK from SW and not by HW
14	SDCC5_M_IDLE_DIS	RW	If this bit is set, then sdcc5_m_idle will be tied low
13	QSB_AXI_INTERLEAVING_EN	RW	If the bit is set, then the controller can support interleaving between two bursts.This option is disabled by default.
12	QSB_AXI_TRANSIENT	RW	Driving VMIDMT's input
11	QSB_AXI_PROTNs	RW	Driving VMIDMT's input
10:9	QSB_AXI_REQPRIORITY	RW	Driving VMIDMT's input
8:6	QSB_AXI_MEMTYPE	RW	Driving VMIDMT's input
5	QSB_AXI_NOALLOCATE	RW	Driving VMIDMT's input
4	QSB_AXI_INNERSHARED	RW	Driving VMIDMT's input
3	QSB_AXI_SHARED	RW	Driving VMIDMT's input
2	QSB_AXI_OOWR	RW	Driving QSB AXI output
1	QSB_AXI_OOORD	RW	Driving QSB AXI output

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2 (cont.)**

Bits	Name	Type	Description
0	QSB_AXI_AFULL	RW	Driving QSB AXI output

**0x0C0C4214 PERIPH\_SS\_SDC1\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Last AHB address phase before response error was detected

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0**

Bits	Name	Description
31	CMDQ_TASK_DESC_ERRO R	CMDQ task descriptor AXI response error
30	ADMA_VALID_ERROR	descriptor not valid
29	ADMA_RESP_ERROR	descriptor read error.
28	DATA_RESP_ERROR	AXI data read response error.
27	DATA_WR_RESP_ERROR	AAXI data write response error.
26	ADMA_LEN_MISMATCH	ADMA length mismatch error
25:13	RESP_ADDR_MSB	response error address - MSBs
12:0	RESP_ADDR_LSB	response error address - LSBs

**0x0C0C421C PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x3029C8B2**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0**

Bits	Name	Description
31:30	VS_CAPABILITIES_SLOT_T YPE	Slot Type field from Capabilities Register
29	VS_CAPABILITIES_ASYNC_ INT_SUPPORT	Asynchronous Interrupt Support field from Capabilities Register
28	VS_CAPABILITIES_SYS_BU S_SUPPORT_64_BIT	64 bit System Bus Support field from Capabilities Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0 (cont.)**

Bits	Name	Description
26	VS_CAPABILITIES_VOLTAGE_SUPPORT_1_8V	Voltage Support 1.8v field from Capabilities Register
25	VS_CAPABILITIES_VOLTAGE_SUPPORT_3_0V	Voltage Support 3.0v field from Capabilities Register
24	VS_CAPABILITIES_VOLTAGE_SUPPORT_3_3V	Voltage Support 3.3v field from Capabilities Register
23	VS_CAPABILITIES_SUSPEND_RESUME_SUPPORT	Suspend Resume Support field from Capabilities Register
22	VS_CAPABILITIES_SDMA_SUPPORT	SDMA Support field from Capabilities Register
21	VS_CAPABILITIES_HS_SUPPORT	High Speed Support field from Capabilities Register
19	VS_CAPABILITIES_ADMA2_SUPPORT	ADMA2 Support field from Capabilities Register
18	VS_CAPABILITIES_SUPPORT_8_BIT	8 bit Supported For Embedded Device field from Capabilities Register
17:16	VS_CAPABILITIES_MAX_BLOCK_LENGTH	Max Block Length field from Capabilities Register
15:8	VS_CAPABILITIES_BASE_SDCLK_FREQ	Base Clock Frequency For SD Clock field from Capabilities Register
7	VS_CAPABILITIES_TIMEOUT_CLK_UNIT	Timeout Clock Unit field from Capabilities Register
5:0	VS_CAPABILITIES_TIMEOUT_CLK_FREQ	Timeout Clock Frequency field from Capabilities Register

**0x0C0C4220 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0200808F**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1**

Bits	Name	Type	Description
31:24	SPEC_VERSION	RW	driving HC_SPEC_VERSION_NUM field in HC_REG_FC_FE register
23:16	VS_CAPABILITIES_CLK_MULTIPLIER	RW	Clock Multiplier field from Capabilities Register
15:14	VS_CAPABILITIES_TUNING_MODE	RW	Re-Tuning Modes field from Capabilities Register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1 (cont.)**

Bits	Name	Type	Description
13	VS_CAPABILITIES_USE_TUNING_FOR_SDR50	RW	Use Tuning for SDR50 field from Capabilities Register
12	VS_CAPABILITIES_FUSE_ICE_DISABLE	R	reflects the value of ICE disable fuse. When it is '1' - ICE is disabled for this controller
11:8	VS_CAPABILITIES_TIMER_CNT_FOR_RETUNING	RW	Timer Count for Re-Tuning field from Capabilities Register
7	VS_CAPABILITIES_CMDQ_SUPPORT	R	CMDQ Support field from Capabilities Register - READ ONLY BIT
6	VS_CAPABILITIES_DRIVER_TYPE_D_SUPPORT	RW	Driver Type D Support field from Capabilities Register
5	VS_CAPABILITIES_DRIVER_TYPE_C_SUPPORT	RW	Driver Type C Support field from Capabilities Register
4	VS_CAPABILITIES_DRIVER_TYPE_A_SUPPORT	RW	Driver Type A Support field from Capabilities Register
3	VS_CAPABILITIES_HS400_SUPPORT	R	HS400 Support field from Capabilities Register - READ ONLY BIT
2	VS_CAPABILITIES_DDR_50_SUPPORT	RW	DDR50 Support field from Capabilities Register
1	VS_CAPABILITIES_SDR_104_SUPPORT	RW	SDR104 Support field from Capabilities Register
0	VS_CAPABILITIES_SDR_50_SUPPORT	RW	SDR50 Support field from Capabilities Register

**0x0C0C4224 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG****Type:** RW**Clock:** HCLK**Reset State:** 0x00000000

Configuration bits for e.MMC5.0 DDR200 mode.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG**

Bits	Name	Description
10	FF_CLK_DIS	The ff_clk to CDC/TBY4 is enabled by default. It can be disabled by setting this bit. The ff_clk is enabled to CDC and TBY4 according to CDC_T4_DLY_SEL field
9	VOLTAGE_MUX_SEL	select between 1.8v and 1.2v signals on sdcc5 IO interface with device. default is 0 to choose 1.8v

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG (cont.)**

Bits	Name	Description
8:7	CDC_TRAFFIC_SEL	Selecting the method of cdc_traffic's assertions: Value 0 (default) - auto calibration with stand by feature. Value 1 - SW calibration before any transaction. Value 2 - automatic calibration with HW assertion of cdc_traffic
6	START_CDC_TRAFFIC	This bit starts the cdc traffic assertion after calibration done is received
5	CRC_TOKEN_SAMPL_FALL_EDGE	If set, the start bit of incoming CRC token in DDR200 mode is sampled in falling edge. If this bit is cleared (default) then the start bit is detected in rising edge
4	DATIN_SAMPL_FALL_EDGE	If set, the start bit of incoming data in DDR200 mode is sampled in falling edge. If this bit is cleared (default) then the start bit is detected in rising edge
3	RESERVED	RESERVED
2	CMDIN_EDGE_SEL	Select sampling edge of CMD input when CMDIN_RCLK_EN is 1: 0 (default) - falling edge of RCLK, 1 - rising edge of RCLK
1	CMDIN_RCLK_EN	Enable CMD input sampling with RCLK: 0 (default) - CMD input is sampled by SD DLL output clock, 1 - CMD input is sampled by delayed RCLK
0	CDC_T4_DLY_SEL	Select block for RCLK delay: 0 (default) - CM_DLL_SDC4 CDCLP533 and CM_TBY4

**0x0C0C4240 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG****Type:** R**Clock:** PCLK**Reset State:** 0x00000000

HC Spec Defines power and IO voltage switch bits. The information is sent by a dedicated interrupt to the QTI SW driver for handling the power and voltage tasks with PMIC. The different tasks are defined in this register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG**

Bits	Name	Description
3	IO_HIGH_V	SDCC driver orders that the I/O signaling level be switched to 3.0v. This interrupt is triggered when SDCC driver writes '0' to offset 03Eh in HC register (bit 3) or when SW reset for all is handled and last supported voltage was 1.8V. This logic is enabled when SWITCHABLE_SIGNALING_VOLTAGE=true
2	IO_LOW_V	SDCC driver orders that the I/O signaling level be switched to 1.8v. This interrupt is triggered when SDCC driver writes '1' to offset 03Eh in HC register (bit 3). This logic is enabled when SWITCHABLE_SIGNALING_VOLTAGE=true

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG (cont.)**

Bits	Name	Description
1	BUS_ON	SDCC driver orders that the SD bus power be switched on. This interrupt is triggered when SDCC driver writes '1' to offset 029h in HC register (bit 0).
0	BUS_OFF	SDCC driver orders that the SD bus power be switched off. This interrupt is triggered when SDCC driver writes '0' to offset 029h in HC register(bit 0) or when SW reset for all is handled.

**0x0C0C4244 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG****Type:** RW**Clock:** PCLK**Reset State:** 0x00000000

Mask register for MCI\_PWRCTL\_STATUS\_REG

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG**

Bits	Name	Description
3	IO_HIGH_V	'1' - interrupt is enabled / '0' - interrupt is masked.
2	IO_LOW_V	'1' - interrupt is enabled / '0' - interrupt is masked.
1	BUS_ON	'1' - interrupt is enabled / '0' - interrupt is masked.
0	BUS_OFF	'1' - interrupt is enabled / '0' - interrupt is masked.

**0x0C0C4248 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG****Type:** W**Clock:** PCLK**Reset State:** 0x00000000

Clear register for MCI\_PWRCTL\_STATUS\_REG

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG**

Bits	Name	Description
3	IO_HIGH_V	Writing '1' clears bit 03 in PWRCTL_STATUS_REG
2	IO_LOW_V	Writing '1' clears bit 02 in PWRCTL_STATUS_REG
1	BUS_ON	Writing '1' clears bit 01 in PWRCTL_STATUS_REG
0	BUS_OFF	Writing '1' clears bit 00 in PWRCTL_STATUS_REG

**0x0C0C4260 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00030022

AXI values and other vendor specific functions

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4**

Bits	Name	Description
23	DLL_CDR_EN_OVERRIDE	in CMDQ mode, the CQE controls on CDR_EN input of DLL. This bit is allow SW to override CQE control over CDR_EN with original value from HC_REG_DLL_CONFIG. Default is '0'
22:20	UARM_PRIORITY_MODE	Priority for Axi master read clients (ADMA_DESC_FETCH, CMDQ_TASK_DESC_FETCH, RAM_DATA_CLIENT)
19	SBFE_UAWM_DIS	Axi Write error disable
18	SBFE_UARM_DIS	Axi read error disable
17:16	AXI_MAX_BURST_LENGTH	Controls the max length of AXI burst initiated by SDCC master. AXI Burst length = $2^{(AXI\_MAX\_BURST\_LENGTH+4)}$ . Default = 3 for 128Byte bursts for best performance
15	DISABLE_CRYPTO	When set, HC_CMDQ_CAPABILITIES.CRYPTO_SUPPORT will be overridden to '0' even if controller includes ICE
14	WRAP_ERROR	When the bit is set, the core responds with errors on WRAP transaction in configuration bus.
13	RX_FLOW_TIMING	Configuration bit which selects the cycle which RxFlowControl will be asserted when UHS mode is used Value 0 (default) - RxFlowControl is asserted one clock before the end bit is received in DPSM (NextBlkTCnt in DPSM equals to 1). Value 1 - RxFlowControl is asserted two clocks before the end bit is received in DPSM (NextBlkTCnt in DPSM equals to 2).
12	PROGDONE_WO_CMD_RESP	'0' (Default): we wait for cmd response end to detect progdone (legacy behavior). '1': we start looking for progdone without waiting for cmd response end but after 10 cycles from end bit of command
11:10	HC_AXI_ARLOCK	AXI ARLOCK value. shared for QMB and SCM. Default = 0x0
9:8	HC_AXI_AWLOCK	AXI AWLOCK value. shared for QMB and SCM. Default = 0x0
7:4	HC_AXI_ARCACHE	AXI ARCACHE value. shared for QMB and SCM. Default = 0x2
3:0	HC_AXI_AWCACHE	AXI AWCACHE value. shared for QMB and SCM. Default = 0x2

**0x0C0C4254 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0020A000

Second register for DLL's configuration

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_2**

Bits	Name	Description
22	DLL_OUTPUT_TO_PAD	'
21	DLL_CLOCK_DISABLE	'1' (default) - DLL clock is disabled. '0' - dll clock has legacy clock enable. SW must keep this bit set until correct clock frequency is delivered to DLL
19	LOW_FREQ_MODE	'0' (default) - High Frequency Mode (frequency range 175MHz - 208MHz) '1' - Low Frequency Mode (Clock frequency from 87.5MHz - 104MHz)
18	FLL_CYCLE_CNT	Number of TCXO clock cycles for which FLL counts before calculating error. 0 (default) - 4 TCXO Clock Cycles.1 - 8 TCXO Clock Cycles
17:10	MCLK_FREQ_CALC	Round off to nearest integer [(Freq of MCLK / Freq of TCXO) * FLL_CYCLE_CNT(4/8)]. Example : Freq of MCLK = 200MHz Freq of TCXO = 19.2MHz FLL_CYCLE_CNT = 0. Round off [(200/19.2)*4] = 41.666 = 42 Bits.
3:2	DDR_TRAFFIC_INIT_SEL	Selecting how DDR_TRAFFIC_INIT will be driven to DLL. Value 00 - HW drives the DDR_TRAFFIC_INIT when DPSM is active and when CPSM is active if RCLK is expected during response. Value 01 - The value is driven by SW by SW_DDR_TRAFFIC_INIT bit. Value 10 - DDR_TRAFFIC_INIT to DLL is driven when sdc4_mclk is not gated. off.
1	DDR_TRAFFIC_INIT_SW	The signal should be high 3-4 cycles before RCLK is toggling. It should be high when MCLK is toggling.
0	DDR_CAL_EN	The signal is set after the configuration of MCI_DDR_CONFIG is finished. When the signal is low the TCXO domain is being reset.

**0x0C0C4258 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_3****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Third register for DLL's configuration

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_3**

Bits	Name	Description
7:0	SDC4_CONFIG_MSB	bit[23:16] of sdc4_config bus. for more info check data sheets in go/dll

**0x0C0C425C PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DDR\_CONFIG****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x80040873

Configuration of ddr\_config input of DLL

**PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DDR\_CONFIG**

Bits	Name	Description
31	PRG_DLY_EN	Enables Programmable delay for RCLK_IN
30	EXT_PRG_RCLK_DLY_EN	Enables external control of Programmable RCLK delayline
29:27	EXT_PRG_RCLK_DLY_CODE	Used only if EXT_PRG_RCLK_DLY_EN is set. Configuring Programmable RCLK Delayline process code number.
26:21	EXT_PRG_RCLK_DLY	Used only if EXT_PRG_RCLK_DLY_EN is set. Programmable RCLK Delay line fine delay control.
20:12	TCXO_CYCLES_DLY_LINE	Number of TCXO clock cycles between finish of one delay line calibration and beginning of next delay line calibration.
11:9	TCXO_CYCLES_CNT	Number of TCXO clock cycles for which down counter is enabled for before making each inc/dec decision.
8:0	PRG_RCLK_DLY	These bits are controlled the programmable delay line. This bus is binary coded. Decimal value = (TCXO period * Decimal value of TCXO_CYCLES_CNT)/(2*delay needed). For 1.25ns the formula is (52n*4/2*1.25) = 83 in decimal.

**0x0C0C42E4 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AW\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

AXI Address channel counter, counts a successful write address transaction

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AW\_MON**

Bits	Name	Type	Description
17	HC_AXI_AW_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_AW_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_AW_CNTR	R	counts a successful write address transaction

**0x0C0C42E8 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_W\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Write data channel counter, counts a successful write data transaction

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_W\_MON**

Bits	Name	Type	Description
17	HC_AXI_W_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_W_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_W_CNTR	R	counts a successful write data transaction

**0x0C0C42EC PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_B\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Write response channel counter, counts a successful write transaction

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_B\_MON**

Bits	Name	Type	Description
17	HC_AXI_B_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_B_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_B_CNTR	R	counts a successful write transaction

**0x0C0C42F0 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AR\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Address channel counter, counts a successful read address transaction

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AR\_MON**

Bits	Name	Type	Description
17	HC_AXI_AR_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_AR_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_AR_CNTR	R	counts a successful read address transaction

**0x0C0C42F4 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_R\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Read data channel counter, counts a successful read data transaction

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_R\_MON**

Bits	Name	Type	Description
17	HC_AXI_R_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_R_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_R_CNTR	R	counts a successful read data transaction

**0x0C0C435C PERIPH\_SS\_SDC1\_SDCC\_DATA\_COUNT****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

DPSM data counter register

**PERIPH\_SS\_SDC1\_SDCC\_DATA\_COUNT**

Bits	Name	Description
27:0	DATACOUNT	Value of data counter in MciDPSM block. Represents remaining data of transaction.

**0x0C0C42F8 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt status register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS**

Bits	Name	Description
21	ERR_INT_STS_START_BIT_ERROR	Start Bit Error flag
20	ERR_INT_STS_FIFO_FULL_WR	
19	ERR_INT_STS_FIFO_EMPT_Y_RD	FIFO read attempt when FIFO was empty
18	ERR_INT_STS_AUTO_CMD19_TIMEOUT	Auto CMD19 timeout
17	ERR_INT_STS_BOOT_TIMEOUT	Data wasn't received within the valid time (according to BOOT_DATA_TIMER) from the start of boot operation.
16	ERR_INT_STS_BOOT_ACK_ERR	Acknowledge pattern wasn't received correctly or not within the valid time (according to BOOT_ACK_TIMER) from the start of boot operation.
0	NORMAL_INT_STS_BOOT_ACK_REC	Acknowledge pattern was received correctly.

**0x0C0C42FC PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt status enable register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS\_EN**

Bits	Name	Description
21	ERR_INT_STS_EN_START_BIT_ERROR	START_BIT_ERROR status enable
20	ERR_INT_STS_EN_FIFO_FULL_WR	FIFO_FULL_WR status enable
19	ERR_INT_STS_EN_FIFO_EMPTY_RD	IFO_EMPTY_RD status enable
18	ERR_INT_STS_EN_AUTO_CMD19_TIMEOUT	AUTO_CMD19_TIMEOUT status enable
17	ERR_INT_STS_EN_BOOT_TIMEOUT	BOOT_TIMEOUT status enable
16	ERR_INT_STS_EN_BOOT_ACK_ERR	BOOT_ACK_ERR status enable
0	NORMAL_INT_STS_EN_BOOT_ACK_REC	BOOT_ACK_REC status enable

**0x0C0C4300 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_SIG\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt signal enable register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_SIG\_EN**

Bits	Name	Description
21	ERR_INT_SIGNAL_EN_START_BIT_ERROR	START_BIT_ERROR signal enable
20	ERR_INT_SIGNAL_EN_FIFO_FULL_WR	FIFO_FULL_WR signal enable
19	ERR_INT_SIGNAL_EN_FIFO_EMPTY_RD	IFO_EMPTY_RD signal enable
18	ERR_INT_SIGNAL_EN_AUTO_CMD19_TIMEOUT	AUTO_CMD19_TIMEOUT signal enable
17	ERR_INT_SIGNAL_EN_BOOT_TIMEOUT	BOOT_TIMEOUT signal enable
16	ERR_INT_SIGNAL_EN_BOOT_ACK_ERR	BOOT_ACK_ERR signal enable
0	NORMAL_INT_SIGNAL_EN_BOOT_ACK_REC	BOOT_ACK_REC signal enable

**0x0C0C430C PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Handling the boot operation.

**PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT**

Bits	Name	Description
3	EARLY_ASSERT_CMD_LINE	If set to '1' then the CMD line is asserted by HW when all the data was received on the SD bus
2	BOOT_ACK_EN	If set to '1' then Host waits for acknowledge pattern after initiating the boot operation.
1	BOOT_EN	When this bit is asserted, the boot operation is initiated in both of the modes.
0	BOOT_MODE	If set to '1' then CMD line is low during the boot operation. If boot_mode = '0', then CMD0 with the argument 0xFFFFFFF is sent.

**0x0C0C4310 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_ACK\_TIMER****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Timer for boot operation - the time until receiving the acknowledge pattern.

**PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_ACK\_TIMER**

Bits	Name	Description
31:0	BOOT_ACK_TIMER	Timer for counting the cycles from initiating the boot operation until the acknowledge pattern is accepted.

**0x0C0C4314 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_DATA\_TIMER****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

boot data timer register.

**PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_DATA\_TIMER**

Bits	Name	Description
31:0	BOOT_DATA_TIMER	Number of cycles from initiating the boot operation until the first data is received

**0x0C0C4320 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_GENERICS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Generics list of each SDCC instance.

**PERIPH\_SS\_SDC1\_SDCC\_SDCC\_GENERICS**

Bits	Name	Description
26	USE_SCM	current SDCC instance includes SCM
25	USE_XPU	current SDCC instance includes XPU
24	USE_VMIDMT	current SDCC instance includes VMIDMT
23	USE_CMD_QUE	current SDCC instance supports CMDQ (eMMC5.1)
22	HS400_SUPPORT	current SDCC instance supports HS400 timing mode
21	USE_ICE	current SDCC instance includes ICE
20:17	SD_DATA_WIDTH	Number of DAT lines Value 4 - 4 DAT lines (SD, SDIO) Value 8 - 8 DAT lines (MMC, eMMC)
16:4	RAM_SIZE	Size of RAM size: Optional Values: 512, 1024, 2048 or 4096 bytes.
3:1	NUM_OF_DEV	Number of eSD or eSDIO devices supported on shared SD bus. Can be set from 1 to 4.
0	USE_DLL_SDC4	Enables the instantiation of cm_dll_sdc4. Value 0 - cm_dll_sdc4 isn't integrated with SDCC5 Value 1 - cm_dll_sdc4 is used with SDCC5.

**0x0C0C4324 PERIPH\_SS\_SDC1\_SDCC\_FIFO\_STATUS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00007800

Status bit of FIFO's fill level. Can be used for long accesses to FIFO using the FIFO\_ALT register

**PERIPH\_SS\_SDC1\_SDCC\_FIFO\_STATUS**

Bits	Name	Description
18	RX_FIFO_512	512 bytes are ready in FIFO during RX transaction
17	RX_FIFO_256	256 bytes are ready in FIFO during RX transaction
16	RX_FIFO_128	128 bytes are ready in FIFO during RX transaction
15	RX_FIFO_64	64 bytes are ready in FIFO during RX transaction
14	TX_FIFO_512	available space in FIFO for 512 bytes during TX transaction
13	TX_FIFO_256	available space in FIFO for 256 bytes during TX transaction
12	TX_FIFO_128	available space in FIFO for 128 bytes during TX transaction
11	TX_FIFO_64	available space in FIFO for 64 bytes during TX transaction
10:0	FIFO_FILL_LEVEL	reflects the actual fill level of FIFO DPRAM during TX or RX transaction. Supported values are 0, 4, 8, ..., 1020, 1024

**0x0C0C4400+PERIPH\_SS\_SDC1\_SDCC\_HC\_FIFO\_ALTn, n=[0..255]****0x4\*n****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Alternative register to MCI\_FIFO (0x80-0xBC). The SW can use this register range (0x400-0x7FC) for accessing all the RAM range (1024 bytes) in FIFO mode. The advantage is that any byte in the RAM can be accessed without repeating the same address

**PERIPH\_SS\_SDC1\_SDCC\_HC\_FIFO\_ALTn**

Bits	Name	Description
31:0	DATA	WORD is written or received to/from the RAM in FIFO mode

**0x0C0C4800 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

ICE Vendor Specific Controls

**PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL**

Bits	Name	Description
0	ICE_SW_RST_EN	Enable the ICE SW Reset. If this bit is '0', ICE Core could not be reset through SW. If this bit is '1', ICE Core will receive HCLK SW reset as its sync reset and ICE sync_rst_ack will be part of HCLK SW out-of-reset logic

**0x0C0C5004 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CAPABILITIES****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x100020C0

This register is reserved for capability indication

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CAPABILITIES**

Bits	Name	Description
28	CRYPTO_SUPPORT	Indicates whether the host controller supports cryptographic operations. '1' means the controller supports encryption operation
15:12	INT_TIMER_CLK_FREQ_MULTIPLIER	INT_TIMER_CLK_FREQ_MULTIPLIER and INT_TIMER_CLK_FREQ_VALUE indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. Field Value Description: 0h = 0.001 MHz 1h = 0.01 MHz 2h = 0.1 MHz 3h = 1 MHz 4h = 10 MHz Other values are reserved
09:00	INT_TIMER_CLK_FREQ_VALUE	INT_TIMER_CLK_FREQ_MULTIPLIER and INT_TIMER_CLK_FREQ_VALUE indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_STATUS (CMD13) polling. The clock frequency is calculated as INT_TIMER_CLK_FREQ_VALUE * INT_TIMER_CLK_FREQ_MULTIPLIER. For example, to encode 19.2 MHz, INT_TIMER_CLK_FREQ_VALUE shall be C0h (= 192 decimal) and INT_TIMER_CLK_FREQ_MULTIPLIER shall be 2h (0.1 MHz): 192 * 0.1 MHz = 19.2 MHz.

**0x0C0C5008 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONFIGURATION****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

This register controls CQE behavior affecting the general operation of command queueing 269 module or operation of multiple tasks in the same time

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONFIGURATION**

Bits	Name	Description
12	DIRECT_CMD_EN	this bit indicates to hw if Task Descr in slot #31 of the TDL is a Data Transfer Task Descr, or a Direct Command Task Descriptor.CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor.Bit Value Description 1 = Task descriptor in slot #31 is a DCMD Task Descriptor,0 = Task descriptor in slot #31 is a Data Transfer Task Descriptor
01	CRYPTO_GENERAL_ENABLE	Enable bit for crypto engine. '0' - Disable ICE for all data transfers. '1' - enable ICE if 'crypto_enable' field in task descriptor is '1'
08	TASK_DESCR_SIZE	This bit indicates whether task descr size is 128 or 64 bits, This bit can only be configured when CmdQ Enable bit is '1' (command queueing is disabled,1- Task descr size is 128 bits.0-Task descriptor size is 64 bits
00	CMDQ_EN	Software shall write '1' to this bit in order to enable command queueing mode. When this bit is 0, CQE is disabled and software controls the eMMC bus using the legacy eMMC host controller.Before software writes '1' to this bit, software shall verify that the eMMC host controller is in idle state and there are no commands or data transfers ongoing. When software wants to exit command queueing mode, it shall clear all previous tasks if such exist before setting this bit to 0

**0x0C0C500C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONTROL****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

clear ,halt

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONTROL**

Bits	Name	Description
00	HALT	Host sw shall write '1' to bit when it wants to acquire sw control over eMMC bus and disable CQE from issuing commands on the bus.For example, issuing a Discard Task command (CMDQ_TASK_MGMT) When sw writes '1', CQE shall complete the ongoing task if such a task is in progress.Once the task is completed and CQE is in idle state,CQE shall not issue new commands and indicate so to sw by setting this bit to 1. Sw may poll on this bit until it is set to 1, and may only then send commands on the eMMC bus.In order to exit halt state (i.e. resume CQE activity),sw shall clear this bit

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONTROL (cont.)**

Bits	Name	Description
08	CLEAR_ALL_TASKS	Software shall write '1' this bit when it wants to clear all the tasks sent to the device. This bit can only be written when CQE is in halt state (i.e. Halt bit is 1). When software writes '1', the value of the register is updated to 1, and CQE shall reset CQTDBR register and all other context information for all unfinished tasks. Then CQE will clear this bit. Software should poll on this bit until it is set to back 0 and may then resume normal operation. CQE does not communicate to the device that the tasks were cleared. It is software's responsibility to order the device to discard the tasks in its queue using CMDQ_TASK_MGMT command. Writing '1' to this register shall have no effect.

**0x0C0C5010 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

cmdq\_int\_status

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS**

Bits	Name	Description
06	DEVICE_EXCEPTION	This status bit is asserted when an R1 response is received to any command (except CMD13) where EXCEPTION_EVENT bit (bit 6) is set, indicating an exception by the device
05	RESERVED	Reserved
04	RESERVED	Reserved
03	TASK_CLEARED	This bit is asserted (if CQISTE.TCL=1) when a task clear operation is completed by CQE.
02	RESP_ERR_DETECTED_IN T	Bit is asserted (if CQISE.RED=1) when a response is received with an error bit set in the device status field
01	TASK_COMPLETE_INT	This bit is asserted (if CQISTE.TCC=1) with at least one of the following:(1) A task is completed and the INT bit is set in its Task Descriptor,(2) Interrupt caused by Interrupt Coalescing logic
00	HALT_COMPLETE_INT	This status bit is asserted (if CQISTE.HAC=1) when halt bit in CQTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state

**0x0C0C5014 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

status\_en

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS\_EN**

Bits	Name	Description
06	DEVICE_EXCEPTION_STATUS_EN	1:CQIS.DEE will be set when its interrupt condition is active, 0 - CQIS.DEE is disabled
05		
04		
03	TASK_CLEARED_STATUS_EN	1:CQIS.TCL will be set when its interrupt condition is active, 0 - CQIS.TCL is disabled
02	RESP_ERR_DETECTED_STATUS_EN	1:CQIS.RED will be set when its interrupt condition is active, 0 - CQIS.RED is disabled
01	TASK_COMPLETE_STATUS_EN	1:CQIS.TCC will be set when its interrupt condition is active, 0 - CQIS.TCC is disabled
00	HALT_COMPLETE_STATUS_EN	1:CQIS.HAC will be set when its interrupt condition is active, 0 - CQIS.HAC is disabled

**0x0C0C5018 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_SIGNAL\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

signal\_en

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_SIGNAL\_EN**

Bits	Name	Description
06	DEVICE_EXCEPTION_SIGNAL_EN	When set and CQIS.DEE is asserted, the CQE shall generate an interrupt
05		
04		
03	TASK_CLEARED_SIGNAL_EN	Task Cleared Signal Enable (TCL), When set and CQIS.TCL is asserted, the CQE shall generate an interrupt
02	RESP_ERR_DETECTED_SIGNAL_EN	Response Error Detected Signal Enable (TCC), When set and CQIS.RED is asserted, the CQE shall generate an interrupt

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_SIGNAL\_EN (cont.)**

Bits	Name	Description
01	TASK_COMPLETE_SIGNAL_EN	When set and CQIS.TCC is asserted, the CQE shall generate an interrupt
00	HALT_COMPLETE_SIGNAL_EN	When set and CQIS.HAC is asserted, the CQE shall generate an interrupt

**0x0C0C501C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_COALESCING****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

coalescing\_int

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_COALESCING**

Bits	Name	Type	Description
31	INT_COALESCING_EN	RW	When set to '1' by sw, cmd responses are neither counted nor timed. Interrupts are still triggered by completion of tasks with INT bit set. When set to '1', the interrupt coalescing mechanism is enabled and coalesced interrupts are generated
20	INT_COALESCING_STATUS_BIT	R	This bit indicates to software whether any responses have been received and counted towards interrupt aggregation (i.e., ICSB is set if and only if IC counter > 0,1 = At least one command has been received and counted (IC counter >0),0 - No commands has been received since last counter reset (IC counter =0))
16	COUNTER_AND_TIMER_RESET	W	When host driver writes '1', the interrupt aggregation timer and counter are reset
15	INT_COALESCING_COUNTER_THRES_WEN	W	When SW writes '1', the value ICCTH is updated with the contents written at the same cycle. When SW writes '0', the value in ICCTH is not updated.
12:08	INT_COALESCING_COUNTER_THRES	RW	Software uses this field to configure the number of responses that are required to generate an interrupt.Counter Operation: As tasks without INT bit set are completes, they are counted by CQE. The counter is reset by software during the interrupt service routine.The counter stops counting when it reaches the value configured in ICCTH.The maximum allowed value is 31, NOTE: When ICCTH is 0, responses are not counted, and counting-based interrupts are not generated.In order to write to this field, the ICPWEN bit must be set at the same write operation
07	INT_COALESCING_TIMOEOUT_VALUE_WEN	W	When SW writes '1', the value ICTOVAL is updated with the contents written at the same cycle. When SW writes '0', the value in ICTOVAL is not updated.

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_COALESCING (cont.)**

Bits	Name	Type	Description
06:00	INT_COALESCING_TIMOE OUT_VALUE	RW	Software uses this field to configure the maximum time allowed between a response on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when a task without INT bit set is completed, after the timer was reset. The timer stops when it reaches the value configured in ICTORVAL field. NOTE: When ICTORVAL is 0, the timer is not running, and timer-based interrupts are not generated. The Time units in this field are 53.33 us. Therefore, writing 0x01 represents a time-out value of 53.33 us, and writing 0xFF represents a time-out value of 13.6 ms.

**0x0C0C5020 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_LSB****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

base address register - LSB

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_LSB**

Bits	Name	Description
31:0	TASK_DESC_LIST_BASE_A DDR	This register stores the LSB bits (bits 31:0) of the byte addr of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address shall be set on 8-Byte boundary. The lower 3 bits of this register shall be set to 0 by software and shall be ignored by CQE.

**0x0C0C5024 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_MSB****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

base address register - MSB

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_MSB**

Bits	Name	Description
3:0	TASK_DESC_LIST_BASE_A DDR	This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This register is reserved when using 32-bit addressing mode

**0x0C0C5028 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DOORBELL****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

doorbell register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DOORBELL**

Bits	Name	Description
31:0	CMDQ_TASK_DOORBELL	Software shall configure TDLBA and TDLBAU, and enable CQE in CQCFC before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. CQE always processes tasks in-order according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks ,by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to '1' by CQE in one of the following events:(a) When a task execution is completed (with success or error),(b) The task is cleared using CQTCLR register,(c) All tasks are cleared using CQTIL register,(d) CQE is disabled using CQCFC register. Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission:CQE shall process the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution will be based on said processing order. Writing 0 by software shall have no impact on the hardware, and will not change the value of the register bit.

**0x0C0C502C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_COMPLETE\_INT****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

task\_notify

#### PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_COMPLETE\_INT

Bits	Name	Description
31:0	TASK_COMPLETE_NOTIFICATION	CQE shall set bit n of this register, at the same time it clears bit n of CQTDBR. When a task execution is completed (with success or error). When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

#### 0x0C0C5030 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_QUEUE\_STATUS

Type: R

Clock: same rate as HCLK

Reset State: 0x00000000

This register stores the most recent value of the device's queue status.

#### PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_QUEUE\_STATUS

Bits	Name	Description
31:0	CARD_QUEUE_STATUS	Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command, i.e. the device's queue status.

#### 0x0C0C5034 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_PENDING\_TASKS

Type: R

Clock: same rate as HCLK

Reset State: 0x00000000

pending

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_PENDING\_TASKS**

Bits	Name	Description
31:0	PENDING_TASKS	Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task hasn't been executed yet. CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution. Software needs to read this register in the task-discard procedure, when the controller is halted, to determine if the task is queued in the device. If the task is queued, the driver sends a CMDQ_TASK_MGMT (CMD48) to the device ordering it to discard the task. Then software clears the task in the CQE. Only then the controller sw orders CQE to resume its operation using CQTIL register

**0x0C0C5038 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_CLEAR****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

clear task

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_CLEAR**

Bits	Name	Description
31:0	CMDQ_TASK_CLEAR	Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFC register Halt bit. When software writes '1' to a bit in this register, CQE updates the value to '1', and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once clear operation is complete. Software should poll on the CQTCLR until it is cleared to verify clear operation was complete. Writing to this register only clears the task in the CQE and does not have impact on the device. In order to discard the task in the device, host software shall send CMDQ_TASK_MGMT while CQE is still in Halt state.

**0x0C0C503C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DESC\_PROC\_ERROR****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Task descriptor processing error

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DESC\_PROC\_ERROR**

Bits	Name	Description
31:0	TASK_DESC_PROC_ERRO R	Bit n of this register indicates that Crypto Configuration Invalid error was detected when processing a Task Descriptor with Task ID=n

**0x0C0C5040 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_1****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00011000

timer\_value

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_1**

Bits	Name	Description
19:16	SEND_STATUS_CMD_BLK_CNTR	This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue. A value of n means CQE shall send status command on the CMD line, during the transfer of data block BLOCK_CNT-n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. A value of 0 means that SEND_QUEUE_STATUS (CMD13) command shall not be sent during the transaction. Instead it will be sent only when the data lines are idle. A value of 1 means that STATUS command is to be sent during the last block of the transaction.
15:00	SEND_STATUS_CMD_IDLE_TIMER	This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicating that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS.clock whose frequency is specified in the Status Timer Clock Frequency field CQCAP register. The minimum value is 0001h and the maximum value is FFFFh

**0x0C0C5044 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

send status by blk\_cnt

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_2**

Bits	Name	Description
15:00	SEND_QUEUE_STATUS_RCA	This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_STATUS (CMD13) command argument.CQE shall copy this field to bits 31:16 of the argument when transmitting SEND_QUEUE_STATUS (CMD13) command.

**0x0C0C5048 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_DCMD\_RESPONSE****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

resp\_direct

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_DCMD\_RESPONSE**

Bits	Name	Description
31:00	DIRECT_CMD_LAST_RESP	This register contains the resp of cmd generated by the last direct-command (DCMD) task which was sent.CQE shall update this register when it receives the resp for a DCMD task.This register is considered valid only after bit 31 of CQTDBR register is cleared by CQE

**0x0C0C5050 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_RESP\_MODE\_ERR\_MASK****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0xFDF9A080

resp err detection

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_RESP\_MODE\_ERR\_MASK**

Bits	Name	Description
31:00	RESP_MODE_ERR_MASK	this bit is used as an interrupt on the device status filed which is received in R1/R1b responses. 1 = When a R1/R1b response is received, with bit i in the device status set, a RED interrupt is generated. 0 = When a R1/R1b response is received, bit i in the device status is ignored. the reset value of this register is set to trigger an interrupt on all Error type bits in the device status.NOTE: Responses to CMD13 (SQS) encode the QSR, so they are ignored by this logic

**0x0C0C5054 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_ERR\_INFO****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

cmdq\_task\_err

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_ERR\_INFO**

Bits	Name	Description
31	TRANS_ERR_VALID	This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a data transfer is in progress when the error is detected/indicated, the bit is set to 1. If a no data transfer is in progress when the error is detected/indicated, the bit is cleared to 0.
28:24	TRANS_ERR_TASK_ID	This field indicates the ID of the task which was executed on the data lines when an error occurred. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
21:16	TRANS_ERR_CMD_INDEX	This field indicates the index of the command which was executed on the data lines when an error occurred. The index shall be set to EXECUTE_READ_TASK (CMD46) or EXECUTE_WRITE_TASK (CMD47) according to the data direction. The field is updated if a data transfer is in progress when an error is detected by CQE, or indicated by eMMC controller.
15	CMD_ERR_VALID	This bit is updated when an error is detected by CQE, or indicated by eMMC controller. If a command transaction is in progress when the error is detected/indicated, the bit is set to 1. If a no command transaction is in progress when the error is detected/indicated, the bit is cleared to 0.
12:08	CMD_ERR_TASK_ID	This field indicates the ID of the task which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.
05:00	CMD_ERR_INDEX	This field indicates the index of the command which was executed on the command line when an error occurred. The field is updated if a command transaction is in progress when an error is detected by CQE, or indicated by eMMC controller.

**0x0C0C5058 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_INDEX****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

cmdq\_resp\_index

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_INDEX**

Bits	Name	Description
05:00	LAST_CMD_RESP_INDEX	This field stores the index of the last received command response. CQE shall update the value every time a command response is received.

**0x0C0C505C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_ARGUMENT****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

resp\_arg

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_ARGUMENT**

Bits	Name	Description
31:00	LAST_CMD_RESP_ARGUMENT	This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

**0x0C0C5078 PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_STATUS****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

non-CMDQ Crypto interrupt status register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_STATUS**

Bits	Name	Description
01	INVALID_CRYPTO_CFG	invalid crypto configuration error
00	GENERAL_CRYPTO_ERR	general crypto error

**0x0C0C507C PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_ENABLE****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

non-CMDQ Crypto interrupt status enable register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_ENABLE**

Bits	Name	Description
01	INVALID_CRYPTO_CFG_EN	invalid crypto configuration error enable
00	GENERAL_CRYPTO_ERR_EN	general crypto error enable

**0x0C0C5A00 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_CFG****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x000000043

command queue vendor specific configuration

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_CFG**

Bits	Name	Type	Description
31	CMDQ_SEND_STATUS_TRIGGER	W	Default = '0', When set - CMD13 will be sent
30	CMDQ_MCLK_SW_ENABLE	RW	Default = '0', When set - mclk will be free running in CMDQ mode
29	CMDQ_HCLK_SW_ENABLE	RW	Default = '0', When set - hclk will be free running in CMDQ mode
28:27	CMD_ARBITER_WEIGHT_DCMD_W_TIM	RW	arbitration rule weight for Direct command with timing in command queuing mode. Default = '0'
26:25	CMD_ARBITER_WEIGHT_EXEC_TASK	RW	arbitration rule weight for EXEC_TASK command (CMD46/47) in command queuing mode. Default = '0'
24:23	CMD_ARBITER_WEIGHT_DCMD_WO_TIM	RW	arbitration rule weight for Direct command without timing in command queuing mode. Default = '0'
22:21	CMD_ARBITER_WEIGHT_SEND_STATUS	RW	arbitration rule weight for SEND_STATUS command (CMD13) in command queuing mode. Default = '0'
20:19	CMD_ARBITER_WEIGHT_QUEUE_TASK	RW	arbitration rule weight for QUEUE_TASK commands (CMD44/45) in command queuing mode. Default = '0'
18	SEND_STAT_POLLING_ALWAYS_ON	RW	When set CQE will keep sending periodic CMD13 for QSR feedback all the time, including during task execution. Default = '0'
14	DISABLE_RST_ON_CMDQ_EN_RISE	RW	This field disable RESET originated from cmdq_sw_enable rise. Default is '0'
13	DISABLE_RST_ON_CMDQ_EN_FALL	RW	This field disable RESET originated from cmdq_sw_enable fall. Default is '0'
12	SEND_STATUS_REQ_MASK_DIS	RW	This field disable masking of SEND_STATUS_REQ in last blocks of a transaction. Default is '0'

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_CFG (cont.)**

Bits	Name	Type	Description
11:09	SEND_STATUS_REQ_MAS K_CNT	RW	Towards the end of the transaction if another task is ready we would like to mask CMD13 request to not keep CMD46/47 waiting. Default is 0x1 (1 blocks).
07:05	QUEUE_PARAMS_REQ_M ASK_CNT	RW	Towards the end of the transaction if another task is ready we would like to mask CMD44/45 request to allow other CMDs to be sent during the long execution. Default is 0x2 (2 blocks).
04	DCMD_REQ_MASK_DISAB LE	RW	This field disable masking of DCMD with timing through the execution. Default is '0'
03:00	DCMD_REQ_MASK_DATACT NT	RW	For DCMD with timing - we want to mask the DCMD request to allow other CMDs to be sent during the long execution. DCMD will happen be treated only when DATACNT < DCMD_REQ_MASK_DATACNT. Default is 0x2 (2 blocks).
08	QUEUE_PARAMS_REQ_M ASK_DIS	RW	This field disable masking of QUEUE_PARAMS_REQ in last blocks of a transaction. Default is '0'

**0x0C0C5A04 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_CFG2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0000009B

command queue vendor specific configuration second register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_CFG2**

Bits	Name	Description
14	CHECK_AVOID_TXFLWCTL _ALWAYS	When this bit is set we will always check if there is a full block ready in the RAM before the starting to send the data to the device. Default = '0'
13	CHECK_AVOID_TXFLWCTL _IGNORE	When this bit is set the check for a full block ready in the RAM before the starting to send the data to the device will be ignored. Default = '0'
12:00	CMD_VALID_WINDOW_CNT	This field sets the number of cycles before block ends that command can be sent - i.e. 'command_can_be_sent = #cycles_to_blk_end > CMD_VALID_WINDOW_CNT'. Default is 155 cycles, assuming N-cr of 64 cycles. For CARDs with lower N-cr, this field can be updated accordingly

**0x0C0C5A08 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CMD\_INDICES****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0DBEEB6C

command queue commands indices register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CMD\_INDICES**

Bits	Name	Description
29:24	SEND_STATUS_INDEX	command index of SEND_STATUS. default value is 13 (CMD13).
23:18	EXECUTE_WRITE_TASK_I_NDEX	command index of EXECUTE_WRITE_TASK. default value is 47 (CMD47).
17:12	EXECUTE_READ_TASK_IN_DEX	command index of EXECUTE_READ_TASK. default value is 46 (CMD46).
11:06	QUEUED_TASK_ADDRESS_INDEX	command index of QUEUED_TASK_ADDRESS. default value is 45 (CMD45).
05:00	QUEUED_TASK_PARAMS_I_NDEX	command index of QUEUED_TASK_PARAMS. default value is 44 (CMD44).

**0x0C0C5A0C PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CAPABILITIES****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0000020C0

Vendor specific R/W register for RO capalities register

**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CAPABILITIES**

Bits	Name	Type	Description
28	CRYPTO_SUPPORT	R	Indicates whether the host controller supports cryptographic operations. '1' means the controller supports encryption operation
15:12	INT_TIMER_CLK_FREQ_MULTIPLIER	RW	clk freq of clock used for int coalescing and send status polling period.
09:00	INT_TIMER_CLK_FREQ_VALUE	RW	clk freq of clock used for int coalescing and send status polling period.

**0x0C0C5A58 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_COUNTERS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000**PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VENDOR\_SPECIFIC\_COUNTERS**

Bits	Name	Description
05:00	INT_COAL_COUNTER	Interrupt coalescing counter

**0x0C0C8000 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CONTROL****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

Global ICE configuration register.

This register belongs to the GLOBAL\_CONTROL resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_CONTROL**

Bits	Name	Description
3:0	GENERAL_PURPOSE	RETENTION Bits reserved for ECOs.

**0x0C0C8004 PERIPH\_SS\_SDC1\_SDCC\_ICE\_RESET****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

Global ICE reset registers.

This register belongs to the GLOBAL\_CONTROL resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_RESET**

Bits	Name	Type	Description
16	CONTROLLER_RESET	R	This status bit reflects the client clock sync-reset control state which is driven by the client itself (Usually by the SW). Once the client requests a reset, it is also used to reset the core clock domain side in the ICE. 0x1: RESET_ON 0x0: RESET_OFF
8	IGNORE_CONTROLLER_RESET	RW	RETENTION This control bit allows the ICE to select whether the client clock sync-reset request from the controller should be used to reset the core clock datapath side. If client reset is ignored, ICE_RESET must be used to reset the core clock domain side. 0x1: IGNORE 0x0: USE
0	ICE_RESET	RW	This reset bit allows the reset of the ICE datapath residing on the core clock domain. It does not reset the ICE configuration registers or the configuration AHB2AHB bridge. It must be used jointly with the controller reset if IGNORE_CONTROLLER_RESET is set. SW should set/unset this bit both for enabling and disabling the reset. ICE_RESET must be enabled prior to setting the storage controller reset. 0x1: RESET_ON 0x0: RESET_OFF

**0x0C0C8010 PERIPH\_SS\_SDC1\_SDCC\_ICE\_FUSE\_SETTING****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register describes the ICE Fuse settings.

This register belongs to the GLOBAL\_CONTROL resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_FUSE\_SETTING**

Bits	Name	Description
2	ICE_FORCE_HW_KEY1_FUSE	ICE is force to use HW key 1 setting.
1	ICE_FORCE_HW_KEY0_FUSE	ICE is force to use HW key 0 setting. HW Key 0 has priority over HW Key 1.

**0x0C0C8030 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_STTS****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register reflects the Invalid Configuration Error status.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_STTS**

Bits	Name	Description
5	STREAM2_CAPIDX_OUT_OF_RANGE	Capability index provided in the Crypto Configuration is out of range. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
4	STREAM1_CAPIDX_OUT_OF_RANGE	Capability index provided in the Crypto Configuration is out of range. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
3	STREAM2_NOT_EXPECTED_CFG_UPD	Crypto Configuration has been updated in middle of transfer. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
2	STREAM1_NOT_EXPECTED_CFG_UPD	Crypto Configuration has been updated in middle of transfer. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
1	STREAM2_PARTIALLY_SET_KEY_USED	Attempt to use an uninitialized key identified. SALT_KEY_STTS and/or DATA_KEY_STTS field of the SW key is not fully set.
0	STREAM1_PARTIALLY_SET_KEY_USED	Attempt to use an uninitialized key identified. SALT_KEY_STTS and/or DATA_KEY_STTS field of the SW key is not fully set.

**0x0C0C8034 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_MASK****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000**RETENTION**

This register controls the masking of the Invalid Configuration Error status interface in the ICE\_INVALID\_CCFG\_ERR\_STTS register. NOTE: that masked Error status bits do not reach the Host Controller but are still reflected in the Error status register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_MASK**

Bits	Name	Description
5	STREAM2_CAPIIDX_OUT_OF_RANGE	Masks the STREAM2_CAPIIDX_OUT_OF_RANGE error indication when set.
4	STREAM1_CAPIIDX_OUT_OF_RANGE	Masks the STREAM1_CAPIIDX_OUT_OF_RANGE error indication when set.
3	STREAM2_NOT_EXPECTED_CFG_UPD	Masks the STREAM2_NOT_EXPECTED_CFG_UPD error indication when set.
2	STREAM1_NOT_EXPECTED_CFG_UPD	Masks the STREAM1_NOT_EXPECTED_CFG_UPD error indication when set.
1	STREAM2_PARTIALLY_SET_KEY_USED	Masks the STREAM2_partially_set_key_used error status bit when set.
0	STREAM1_PARTIALLY_SET_KEY_USED	Masks the STREAM1_partially_set_key_used error status bit when set.

**0x0C0C8038 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_CLR****Type:** W**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register clears the ICE\_INVALID\_CCFG\_ERR\_STTS register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_CLR**

Bits	Name	Description
5	STREAM2_CAPIIDX_OUT_OF_RANGE	Clear the STREAM2_CAPIIDX_OUT_OF_RANGE error indication when set.
4	STREAM1_CAPIIDX_OUT_OF_RANGE	Clear the STREAM1_CAPIIDX_OUT_OF_RANGE error indication when set.
3	STREAM2_NOT_EXPECTED_CFG_UPD	Clear the STREAM2_NOT_EXPECTED_CFG_UPD error indication when set.
2	STREAM1_NOT_EXPECTED_CFG_UPD	Clear the STREAM1_NOT_EXPECTED_CFG_UPD error indication when set.
1	STREAM2_PARTIALLY_SET_KEY_USED	Clears the STREAM2_partially_set_key_used error indication when set.
0	STREAM1_PARTIALLY_SET_KEY_USED	Clears the STREAM1_partially_set_key_used error indication when set.

**0x0C0C8040 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_STTS****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register reflects the General Error status.

This register belongs to the GLOBAL\_CONTROL resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_STTS**

Bits	Name	Description
15	STREAM2_NOT_EXPECTED_NEW_TRNS	New Transfer (DBO=0) provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
14	STREAM1_NOT_EXPECTED_NEW_TRNS	New Transfer (DBO=0) provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
13	STREAM2_NOT_EXPECTED_CONF_IDX	Context Index provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
12	STREAM1_NOT_EXPECTED_CONF_IDX	Context Index provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
11	STREAM2_NOT_EXPECTED_ENC_SEL	Encryption Selector provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
10	STREAM1_NOT_EXPECTED_ENC_SEL	Encryption Selector provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
9	STREAM2_NOT_EXPECTED_DBO	Data Byte Offset provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
8	STREAM1_NOT_EXPECTED_DBO	Data Byte Offset provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
5	STREAM2_NOT_EXPECTED_DUN	Data Unit Number provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.
4	STREAM1_NOT_EXPECTED_DUN	Data Unit Number provided by the controller is not as expected. ICE Enters intentional deadlock state when this happens and has to go through Reset+Init sequence.

**0x0C0C8044 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_MASK****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

## RETENTION

This register controls the masking of the General Error indications to the Host Controller in the ICE\_GENERAL\_ERR\_STTS register. NOTE: that masked Error indications do not reach the Host Controller but are still reflected in the General Error status register.

This register belongs to the GLOBAL\_CONTROL resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_MASK**

Bits	Name	Description
15	STREAM2_NOT_EXPECTED_NEW_TRNS	Masks the STREAM2_NOT_EXPECTED_NEW_TRNS error indication when set.
14	STREAM1_NOT_EXPECTED_NEW_TRNS	Masks the STREAM1_NOT_EXPECTED_NEW_TRNS error indication when set.
13	STREAM2_NOT_EXPECTED_CONF_IDX	Masks the STREAM2_NOT_EXPECTED_CONF_IDX error indication when set.
12	STREAM1_NOT_EXPECTED_CONF_IDX	Masks the STREAM1_NOT_EXPECTED_CONF_IDX error indication when set.
11	STREAM2_NOT_EXPECTED_ENC_SEL	Masks the STREAM2_NOT_EXPECTED_ENC_SEL error indication when set.
10	STREAM1_NOT_EXPECTED_ENC_SEL	Masks the STREAM1_NOT_EXPECTED_ENC_SEL error indication when set.
9	STREAM2_NOT_EXPECTED_DBO	Masks the STREAM2_NOT_EXPECTED_DBO error indication when set.
8	STREAM1_NOT_EXPECTED_DBO	Masks the STREAM1_NOT_EXPECTED_DBO error indication when set.
5	STREAM2_NOT_EXPECTED_DUN	Masks the STREAM2_NOT_EXPECTED_DUN error indication when set.
4	STREAM1_NOT_EXPECTED_DUN	Masks the STREAM1_NOT_EXPECTED_DUN error indication when set.

**0x0C0C8048 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_CLR****Type:** W**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register clears the ICE\_GENERAL\_ERR\_STTS register.

This register belongs to the GLOBAL\_CONTROL resource group.

#### **PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_CLR**

Bits	Name	Description
15	STREAM2_NOT_EXPECTED_NEW_TRNS	Clear the STREAM2_NOT_EXPECTED_NEW_TRNS error indication when set.
14	STREAM1_NOT_EXPECTED_NEW_TRNS	Clear the STREAM1_NOT_EXPECTED_NEW_TRNS error indication when set.
13	STREAM2_NOT_EXPECTED_CONF_IDX	Clear the STREAM2_NOT_EXPECTED_CONF_IDX error indication when set.
12	STREAM1_NOT_EXPECTED_CONF_IDX	Clear the STREAM1_NOT_EXPECTED_CONF_IDX error indication when set.
11	STREAM2_NOT_EXPECTED_ENC_SEL	Clear the STREAM2_NOT_EXPECTED_ENC_SEL error indication when set.
10	STREAM1_NOT_EXPECTED_ENC_SEL	Clear the STREAM1_NOT_EXPECTED_ENC_SEL error indication when set.
9	STREAM2_NOT_EXPECTED_DBO	Clear the STREAM2_NOT_EXPECTED_DBO error indication when set.
8	STREAM1_NOT_EXPECTED_DBO	Clear the STREAM1_NOT_EXPECTED_DBO error indication when set.
5	STREAM2_NOT_EXPECTED_DUN	Clear the STREAM2_NOT_EXPECTED_DUN error indication when set.
4	STREAM1_NOT_EXPECTED_DUN	Clear the STREAM1_NOT_EXPECTED_DUN error indication when set.

#### **0x0C0C8050 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYNDROME1**

**Type:** R

**Clock:** cc\_ice\_core\_clk

**Reset State:** 0x00000000

Error Syndrome register. This register reflects the control parameters received by the ICE from the storage controller. The value of this register is only valid when an Error Indication is triggered.

#### **PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYNDROME1**

Bits	Name	Description
31:16	LENGTH	Length
15	TRANSFER_COMPLETE	Transfer complete indication
14	BYPASS_SEL	Bypass selector
10	ENCRYPT_SEL	Encryption/Decryption selector

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYNDROME1 (cont.)**

Bits	Name	Description
9:5	TRANSFER_INDEX	Transfer Context Index
4:0	CONTEXT_INDEX	Configuration Context Index

**0x0C0C8054 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYNDROME2****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

Error Syndrome register. This register reflects the control parameters received by the ICE from the storage controller. The value of this register is only valid when an Error Indication is triggered.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYNDROME2**

Bits	Name	Description
31:16	DBO	DBO (16lsbs)
15:0	CDU_BASE_NUM	CDU Base Number (16 lsbs)

**0x0C0C8058 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYNDROME1****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

Error Syndrome register. This register reflects the control parameters received by the ICE from the storage controller. The value of this register is only valid when an Error Indication is triggered.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYNDROME1**

Bits	Name	Description
31:16	LENGTH	Length
15	TRANSFER_COMPLETE	Transfer complete indication
14	BYPASS_SEL	Bypass selector
10	ENCRYPT_SEL	Encryption/Decryption selector
9:5	TRANSFER_INDEX	Transfer Context Index
4:0	CONTEXT_INDEX	Configuration Context Index

**0x0C0C805C PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYNDROME2****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

Error Syndrome register. This register reflects the control parameters received by the ICE from the storage controller. The value of this register is only valid when an Error Indication is triggered.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYNDROME2**

Bits	Name	Description
31:16	DBO	DBO (16lsbs)
15:0	CDU_BASE_NUM	CDU Base Number (16 lsbs)

**0x0C0C8074 PERIPH\_SS\_SDC1\_SDCC\_ICE\_BYPASS\_STATUS****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x80000000

ICE bypass status. This register indicate if ICE will be bypassed. When BIST fails, ICE will always be bypassed, IF BIST pass, it's controlled by global\_bypass register

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_BYPASS\_STATUS**

Bits	Name	Description
31	BYPASS_STATUS	ICE Bypass status

**0x0C0CA000 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_n, r=[0..31], n=[0..15]  
+0x80\*r+0x4\*****n****Type:** W**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000**RETENTION**

Crypto Configuration DW0 to DW15. Crypto Key (CRYPTOKEY): Specifies the key to be used for this configuration.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTOCFG\_r\_n**

Bits	Name	Description
31:0	CRYPTOKEY	The specific key layout is defined according to the key size and algorithm specified in the Crypto Capability with index value specified in CAPIDX. When configuring CRYPTOKEY field software shall write the entire key from DW0 to DW15, sequentially, in one atomic set of operations. To regions of CRYPTOKEY which are unused, according to key size and algorithm selection, software shall write zeros.

**0x0C0CA040 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTOCFG\_r\_16, r=[0..31]  
+0x80\*r**

Type: RW

Clock: cc\_ice\_core\_clk

Reset State: 0x00000000

## RETENTION

Crypto Configuration DW16

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTOCFG\_r\_16**

Bits	Name	Description
31	CFGE	Configuration Enable: This field is used by software to enable/disable a Crypto Configuration usage by software 0x1: CFGE_ENABLED (Configuration Enabled. Transactions using this Configuration can be executed) 0x0: CFGE_DISABLED (Configuration Disabled. Transactions using this Crypto Configuration are terminated with error by host controller)
15:8	CAPIDX	Crypto Capability Index : Specifies the index of the Crypto Capability to be used for this configuration. Values allowed are between 0 and CCAP.CC-1
7:0	DUSIZE	Data Unit Size: Size of data unit used with this configuration, encoded in one-hot encoding, analogous to bitmask used in CRYPTOCAP.SDUSB field. When bit j in this field (j=0.....7) is set, a data unit size of 512*2 <sup>j</sup> bytes is selected. Bit j may only be set if the same bit is also set in the SDUSB field of the capability referenced in CAPIDX field.

**0x0C0CA044 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_17, r=[0..31]  
+0x80\*r**

**Type:** RW

**Clock:** cc\_ice\_core\_clk

**Reset State:** 0x00000000

RETENTION

Crypto Configuration DW17

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_17**

Bits	Name	Description
21:20	KEY_SELECTION	<p>This field selects the key to be used for processing the data. Any other but listed mode below is not supported and not allowed. HW will default to SW Key in case of illegal value set.</p> <p>Key0 &amp; Key1 HW keys are provided via the HW wiring from the Security Controller.</p> <p>A change in this field is only allowed when there is absolutely no traffic being processed by the current context.</p> <p>SW must use the SW_FORCED_CONTEXT_SWITCH when changing this setting.</p> <ul style="list-style-type: none"> <li>0x0: USE_KEY0_HW_KEY (Use HW Key0 from Security Controller)</li> <li>0x1: USE_KEY1_HW_KEY (Use HW Key1 from Security Controller)</li> <li>0x3: USE_LUT_SW_KEY (Use SW configured look-up table key. This key is selected using key index which is provided by the SW.)</li> </ul>
17	DECR_BYPASS	<p>This controls the ICE bypass mode for Decryption operations. Setting the decryption bypass mode allows the data to skip decryption, while data can still be encrypted on the opposite direction.</p> <p>A change in this field is only allowed when there is absolutely no traffic being processed by the current context.</p> <p>SW must use the SW_FORCED_CONTEXT_SWITCH when changing this setting.</p> <ul style="list-style-type: none"> <li>0x1: BYPASS_ENABLE (Bypass Enabled)</li> <li>0x0: BYPASS_DISABLE (Bypass Disabled)</li> </ul>

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_17 (cont.)**

Bits	Name	Description
16	ENCR_BYPASS	<p>This controls the ICE bypass mode for Encryption operations. Setting the encryption bypass mode allows the data to skip encryption, while data can still be decrypted on the opposite direction.</p> <p>A change in this field is only allowed when there is absolutely no traffic being processed by the current context.</p> <p>SW must use the SW_FORCED_CONTEXT_SWITCH when changing this setting.</p> <p>0x1: BYPASS_ENABLE (Bypass Enabled) 0x0: BYPASS_DISABLE (Bypass Disabled)</p>

**0x0C0CB000 PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_STTS, r=[0..31]  
+0x8\*r****Type:** R**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register reflects the LUT key status settings. SW shall observe changes in this register to make sure the key has been properly set.

This status register may be cleared by using the ICE\_CONTEXT\_LUT\_CLR register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_STTS**

Bits	Name	Description
15:0	CRYPTOKEY_STTS	<p>Each bit represents that a 32 bit portion of the key has been set in the key registers.</p> <p>For example, bit 0 represents the ICE_CRYPTO CFG_r_0 status. If 0, ICE_CRYPTO CFG_r_0 has never been set before. If 1, ICE_CRYPTO CFG_r_0 has been set at least once.</p>

**0x0C0CB004 PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_CLR, r=[0..31]  
+0x8\*r****Type:** W**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register clears the ICE\_CONTEXT\_LUT\_STTS status bits. SW shall make sure to clear the status before setting the key, so that the status will reflect the latest key setting attempt.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_CLR**

Bits	Name	Description
15:0	CRYPTOKEY_CLR	Writing 1 to each bit of this field resets the appropriate bit in the CRYPTOKEY_SSTS field of the ICE_CONTEXT_LUT_STTS register. Writing 0 has no effect.

**0x0C0CB104 PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_MASK****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0xC0000000**RETENTION**

This register controls the masking of the ICE Secure interrupts in the ICE\_SEC\_IRQ\_STTS register. NOTE: that masked interrupts do not reach the interrupt controller but are still reflected in the interrupts status register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_MASK**

Bits	Name	Description
31	KEYS_RAM_RESET_COMPLETED	Masks the KEYS_RAM_RESET_COMPLETED interrupt when set.
30		

**0x0C0CB108 PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_CLR****Type:** W**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

This register clears the ICE Secure interrupts set in the ICE\_SEC\_IRQ\_STTS register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_CLR**

Bits	Name	Description
31	KEYS_RAM_RESET_COMPLETED	Clears the KEYS_RAM_RESET_COMPLETED interrupt when set.
30		

**0x0C0CB10C PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_CONTROL****Type:** RW**Clock:** cc\_ice\_core\_clk**Reset State:** 0x00000000

## RETENTION

LUT common configuration register.

This register belongs to the CONTEXT\_LUT resource group.

**PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_CONTROL**

Bits	Name	Description
5	CRYPTO CFG_VS_BITS_EN	<p>This bit enables the VSB of CRYPTO CFG. When this bit value is DISABLED: KEY_SELECTION=USE_LUT_SW_KEY DECR_BYPASS=BYPASS_DISABLE ENCR_BYPASS=BYPASS_DISABLE When this bit is ENABLED, VSB bits values are based on CRYPTO CFG configuration. 0x1: ENABLED (Vendor Specific Bits values of CRYPTO CFG are functional) 0x0: DISABLED (Vendor Specific Bits values of CRYPTO CFG are not functional)</p>
4	SW_FORCED_CONTEXT_SWITCH	<p>This bit allows SW to force a context switch to be performed by the engine, with each encryption packet to be processed. Setting this bit is not required in normal operation. Keeping this bit always high, adds additional latency to the crypto engine and can affect throughput. 0x1: FORCE (Context Switch is forced to happen for each encryption packet) 0x0: AUTO (Context Switch is managed automatically by the engine.)</p>
3:1	GENERAL_PURPOSE	Bits reserved for ECOs

**0x0C084000 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_0\_2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Argument 2 Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_0\_2**

Bits	Name	Description
31:0	ARG_2	Argument 2 field

**0x0C084004 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_4\_6****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Block Size Register(15-0)

-Block Count Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_4\_6**

Bits	Name	Description
31:16	BLK_CNT_FOR_CUR_TRA_N	Blocks Count For Current Transfer field from Block Count Register
14:12	BLK_SIZE_HST_SDMA_BU_F	Host SDMA Buffer Boundary field from Block Size Register
11:00	BLK_SIZE_TRANS	Transfer Block Size field from Block Size Register

**0x0C084008 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_8\_A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Argument 1 Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_8\_A**

Bits	Name	Description
31:00	CMD_ARG_1	Command Argument 1 field

**0x0C08400C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_C\_E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

- Transfer Mode Register(15-0)

- Command Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_C\_E**

Bits	Name	Description
29:24	CMD_INDX	Command Index field from Command Register
23:22	CMD_TYPE	Command Type field from Command Register
21	CMD_DATA_PRESENT_SEL	Data Present Select field from Command Register
20	CMD_INDX_CHECK_EN	Command Index Check Enable field from Command Register
19	CMD_CRC_CHECK_EN	Command CRC Check Enable field from Command Register
17:16	CMD_RESP_TYPE_SEL	Response Type Select field from Command Register
5	TRANS_MODE_MULTI_SIN GLE_BLK_SEL	Multi/Single Block Select from Transfer Mode Register
4	TRANS_MODE_DATA_DIRECTION_SEL	Data Transfer Direction Select from Transfer Mode Register
3:2	TRANS_MODE_AUTO_CMD_EN	Auto CMD Enable from Transfer Mode Register
1	TRANS_MODE_BLK_CNT_EN	Block Count Enable field from Transfer Mode Register
0	TRANS_MODE_DMA_EN	DMA Enable field from Transfer Mode Register

**0x0C084010 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_10\_12****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_10\_12**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Response bits 0-31

**0x0C084014 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_14\_16****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_14\_16**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 31-63

**0x0C084018 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_18\_1A****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_18\_1A**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 64-95

**0x0C08401C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_1C\_1E****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Response Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_1C\_1E**

Bits	Name	Description
31:0	CMD_RESP	Command Response field - Long Response bits 96-127

**0x0C084020 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_20\_22****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Buffer Data Port Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_20\_22**

Bits	Name	Description
31:24	BUF_DATA_PORT_3	Byte 3 of Buffer Data
23:16	BUF_DATA_PORT_2	Byte 2 of Buffer Data
15:8	BUF_DATA_PORT_1	Byte 1 of Buffer Data
7:0	BUF_DATA_PORT_0	Byte 0 of Buffer Data

**0x0C084024 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_24\_26****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x01F800F0

Present State Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_24\_26**

Bits	Name	Description
25	SIGANLING_18_SWITCHING_STS	Bit it is cleared, by hardware, when host driver writes any value to 1.8V Signaling Enable field, indicating that a signal switching operation has started. Bit is set, by hardware, when switching operation succeeds.
24	PRESENT_STATE_CMD_LINE_SIGNAL_LEVEL	CMD Linr Signal Level field
23:20	PRESENT_STATE_DAT_3_0_LINE_SIGNAL_LEVEL	DAT[3:0] Line Signal Level field
19	PRESENT_STATE_WR_PROTECT_SWITCH_PIN_LEVEL	Write Protect Switch Pin Level field
18	PRESENT_STATE_CARD_DETECT_PIN_LEVEL	Card Detect Pin Level field
17	PRESENT_STATE_CARD_STATE_STABLE	Card State Stable field
16	PRESENT_STATE_CARD_INSERTED	Card Inserted field

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_24\_26 (cont.)**

Bits	Name	Description
11	PRESENT_STATE_BUF_RD_EN	Buffer Read Enable field
10	PRESENT_STATE_BUF_WR_EN	Buffer Write Enable field
9	PRESENT_STATE_RD_TRANS_ACT	Read Transfer Active field
8	PRESENT_STATE_WR_TRANS_ACT	Write Transfer Active field
7:4	PRESENT_STATE_DAT_7_4_LINE_SIGNAL_LEVEL	DAT[7:4] Line Signal Level field
3	PRESENT_STATE_RETUNING_REQ	Re-Tuning Request field
2	PRESENT_STATE_DAT_LIN_E_ACT	DAT Linr Active field
1	PRESENT_STATE_CMD_INHIBIT_DAT	Command Inhibit (DAT) field
0	PRESENT_STATE_CMD_INHIBIT_CMD	Command Inhibit (CMD) field

**0x0C084028 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_28\_2A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of four registers from the SD Host Controller Standard Specification:

- Host Control 1 Register(7-0)
- Power Control Register(15-8)
- Block Gap Control Register(23-16)
- Wakeup Control Register(31-24)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_28\_2A**

Bits	Name	Description
26	WKUP_EVENT_EN_ON_SD_CARD_REMOVAL	Wakeup Event Enable On SD Card Removal field from Wakeup Control Register
25	WKUP_EVENT_EN_ON_SD_CARD_INSERTION	Wakeup Event Enable On SD Card Insertion field from Wakeup Control Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_28\_2A (cont.)**

Bits	Name	Description
24	WKUP_EVENT_EN_ON_SD_CARD_INT	Wakeup Event Enable On Card Interrupt field from Wakeup Control Register
19	BLK_GAP_CTL_INT	Interrupt At Block Gap field from Block Gap Control Register
18	BLK_GAP_CTL_RD_WAIT	Read Wait Control field from Block Gap Control Register
17	BLK_GAP_CTL_CONTINUE_REQ	Continue Request field from Block Gap Control Register
16	BLK_GAP_CTL_STOP_GAP_REQ	Stop At Block Gap Request field from Block Gap Control Register
11:9	PWR_CTL_SD_BUS_VOLTAGE_SEL	SD Bus Voltage Select field from Power Control Register
8	PWR_CTL_SD_BUS_PWR	SD Bus Power field from Power Control Register
7	HST_CTL1_CARD_DETECT_SIGNAL_SEL	Card Detect Signal Selection field from Host Control 1 Register
6	RESERVED	RESERVED
5	HST_CTL1_EXTENDED_DATA_TRANS_WIDTH	Extended Data Transfer Width field from Host Control 1 Register
4:3	HST_CTL1_DMA_SEL	DMA Select field from Host Control 1 Register
2	HST_CTL1_HS_EN	High Speed Enable field from Host Control 1 Register
1	HST_CTL1_DATA_TRANS_WIDTH	Data Transfer Width field from Host Control 1 Register
0	HST_CTL1_LED_CTL	LED Control field from Host Control 1 Register

**0x0C08402C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_2C\_2E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000001

Combination of three registers from the SD Host Controller Standard Specification:

- Clock Control Register(15-0)
- Timeout Control Register(23-16)
- Software Reset Register(31-24)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_2C\_2E**

Bits	Name	Type	Description
26	SW_RST_DAT_LINE	RW	Software Reset For DAT Line field from Software Reset Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_2C\_2E (cont.)**

Bits	Name	Type	Description
25	SW_RST_CMD_LINE	RW	Software Reset For CMD Line field from Software Reset Register
24	SW_RST_FOR_ALL	RW	Software Reset For All field from Software Reset Register
19:16	DATA_TIMEOUT_COUNTE_R	RW	Data Timeout Counter Value field from Timeout Control Register. The reserved value 0xF is translated to maximum number of cycles - 2^32-1
15:8	CLK_CTL_SDCLK_FREQ_SEL	RW	SDCLK Frequency Select field from Clock Control Register
7:6	CLK_CTL_SDCLK_FREQ_SEL_MSB	RW	Upper Bits of SDCLK Frequency Select field from Clock Control Register
5	CLK_CTL_GEN_SEL	RW	Clock Generator Select field from Clock Control Register
2	CLK_CTL_SDCLK_EN	RW	SD Clock Enable field from Clock Control Register
1	CLK_CTL_INTERNAL_CLK_STABLE	R	Internal Clock Stable field from Clock Control Register
0	CLK_CTL_INTERNAL_CLK_EN	RW	Internal Clock Enable field from Clock Control Register

**0x0C084030 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_30\_32****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Normal Interrupt Status Register(15-0)

-Error Interrupt Status Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_30\_32**

Bits	Name	Type	Description
31	ERR_INT_STS_VENDOR_SPECIFIC_ERR	RW	Vendor Specific Error Status field from Error Interrupt Status Register
30	ERR_INT_STS_VENDOR_SPECIFIC_STAT	RW	Vendor Specific Error/Status field from HC_VENDOR_SPECIFIC_INT_STS register
26	ERR_INT_STS_TUNING_ERR	RW	Tuning Error field from Error Interrupt Status Register
25	ERR_INT_STS_ADMA_ERR	RW	ADMA Error field from Error Interrupt Status Register
24	ERR_INT_STS_AUTO_CMD_ERR	RW	Auto CMD Error field from Error Interrupt Status Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_30\_32 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
23	ERR_INT_STS_CURRENT_LIMIT_ERR	RW	Current Limit Error field from Error Interrupt Status Register
22	ERR_INT_STS_DATA_END_BIT_ERR	RW	Data End Bit Error field from Error Interrupt Status Register
21	ERR_INT_STS_DATA_CRC_ERR	RW	Data CRC Error field from Error Interrupt Status Register
20	ERR_INT_STS_DATA_TIME_OUT_ERR	RW	Data Timeout Error field from Error Interrupt Status Register
19	ERR_INT_STS_CMD_INDX_ERR	RW	Command Index Error field from Error Interrupt Status Register
18	ERR_INT_STS_CMD_END_BIT_ERR	RW	Command End Bit Error field from Error Interrupt Status Register
17	ERR_INT_STS_CMD_CRC_ERR	RW	Command CRC Error field from Error Interrupt Status Register
16	ERR_INT_STS_CMD_TIME_OUT_ERR	RW	Command Timeoutout Error field from Error Interrupt Status Register
15	NORMAL_INT_STS_ERR_INTERRUPT	R	Error Interrupt field from Normal Interrupt Status Register
14	NORMAL_INT_STS_CMD_QUEUE	R	Command Queueing Event field from Normal Interrupt Status Register
12	NORMAL_INT_STS_RETUNING_EVENT	R	Re-Tuning Event field from Normal Interrupt Status Register
11	NORMAL_INT_STS_INT_C	R	INT_C field from Normal Interrupt Status Register
10	NORMAL_INT_STS_INT_B	R	INT_B field from Normal Interrupt Status Register
9	NORMAL_INT_STS_INT_A	R	INT_A field from Normal Interrupt Status Register
8	NORMAL_INT_STS_CARD_INT	R	Card Interrupt field from Normal Interrupt Status Register
7	NORMAL_INT_STS_CARD_REMOVAL	RW	Card Removal field from Normal Interrupt Status Register
6	NORMAL_INT_STS_CARD_INSERTION	RW	Card Insertion field from Normal Interrupt Status Register
5	NORMAL_INT_STS_BUF_RD_READY	RW	Buffer Read Ready field from Normal Interrupt Status Register
4	NORMAL_INT_STS_BUF_WR_READY	RW	Buffer Write Ready field from Normal Interrupt Status Register
3	NORMAL_INT_STS_DMA_INTERRUPT	RW	DMA Interrupt field from Normal Interrupt Status Register
2	NORMAL_INT_STS_BLK_GAP_EVENT	RW	Block Gap Event field from Normal Interrupt Status Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_30\_32 (cont.)**

Bits	Name	Type	Description
1	NORMAL_INT_STS_TRANS_COMPLETE	RW	Transfer Complete field from Normal Interrupt Status Register
0	NORMAL_INT_STS_CMD_COMPLETE	RW	Command Complete field from Normal Interrupt Status Register

**0x0C084034 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_34\_36****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Normal Interrupt Status Enable Register(15-0)

-Error Interrupt Status Enable Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_34\_36**

Bits	Name	Description
31	ERR_INT_STS_EN_VENDOR_SPECIFIC_ERR	Vendor Specific Error Status Enable field from Error Interrupt Status Enable Register
30	ERR_INT_STS_EN_VENDOR_SPECIFIC_STAT	Vendor Specific Error/Status Enable field for ERR_INT_STS_VENDOR_SPECIFIC_STAT interrupt
26	ERR_INT_STS_EN_TUNING_ERR	Tuning Error Status Enable field from Error Interrupt Status Enable Register
25	ERR_INT_STS_EN_ADMA_ERR	ADMA Error Status Enable field from Error Interrupt Status Enable Register
24	ERR_INT_STS_EN_AUTO_CMD_ERR	Auto CMD Error Status Enable field from Error Interrupt Status Enable Register
23	ERR_INT_STS_EN_CURRENT_LIMIT_ERR	Current Limit Error Status Enable field from Error Interrupt Status Enable Register
22	ERR_INT_STS_EN_DATA_END_BIT_ERR	Data End Bit Error Status Enable field from Error Interrupt Status Enable Register
21	ERR_INT_STS_EN_DATA_CRC_ERR	Data CRC Error Status Enable field from Error Interrupt Status Enable Register
20	ERR_INT_STS_EN_DATA_TIMEOUT	Data Timeout Error Status Enable field from Error Interrupt Status Enable Register
19	ERR_INT_STS_EN_COMMAND_INDEX_ERR	Command Index Error Status Enable field from Error Interrupt Status Enable Register
18	ERR_INT_STS_EN_COMMAND_END_BIT_ERR	Command End Bit Error Status Enable field from Error Interrupt Status Enable Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_34\_36 (cont.)**

Bits	Name	Description
17	ERR_INT_STS_EN_CMD_CRC_ERR	Command CRC Error Status Enable field from Error Interrupt Status Enable Register
16	ERR_INT_STS_EN_CMD_TI_MEOUT	Command Timeoutout Error Status Enable field from Error Interrupt Status Enable Register
14	NORMAL_INT_STS_EN_CMD_QUEUE	Command Queueing Event Status Enable field from Normal Interrupt Status Enable Register
12	NORMAL_INT_STS_EN_RE_TUNING_EVENT	Re-Tuning Event Status Enable field from Normal Interrupt Status Enable Register
11	NORMAL_INT_STS_EN_INT_C	INT_C Status Enable field from Normal Interrupt Status Enable Register
10	NORMAL_INT_STS_EN_INT_B	INT_B Status Enable field from Normal Interrupt Status Enable Register
9	NORMAL_INT_STS_EN_INT_A	INT_A Status Enable field from Normal Interrupt Status Enable Register
8	NORMAL_INT_STS_EN_CARD_INT	Card Interrupt Status Enable field from Normal Interrupt Status Enable Register
7	NORMAL_INT_STS_EN_CARD_REMOVAL	Card Removal Status Enable field from Normal Interrupt Status Enable Register
6	NORMAL_INT_STS_EN_CARD_INSERTION	Card Insertion Status Enable field from Normal Interrupt Status Enable Register
5	NORMAL_INT_STS_EN_BUF_RD_READY	Buffer Read Ready Status Enable field from Normal Interrupt Status Enable Register
4	NORMAL_INT_STS_EN_BUF_WR_READY	Buffer Write Ready Status Enable field from Normal Interrupt Status Enable Register
3	NORMAL_INT_STS_EN_DMA_INT	DMA Interrupt Status Enable field from Normal Interrupt Status Enable Register
2	NORMAL_INT_STS_EN_BLOCK_GAP_EVENT	Block Gap Event Status Enable field from Normal Interrupt Status Enable Register
1	NORMAL_INT_STS_EN_TRANS_COMPLETE	Transfer Complete Status Enable field from Normal Interrupt Status Enable Register
0	NORMAL_INT_STS_EN_CMD_COMPLETE	Command Complete Status Enable field from Normal Interrupt Status Enable Register

**0x0C084038 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_38\_3A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Normal Interrupt Signal Enable Register(15-0)

-Error Interrupt Signal Enable Register(31-16)

### **PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_38\_3A**

Bits	Name	Type	Description
31	ERR_INT_SIGNAL_EN_VEN_DOR_SPECIFIC_ERR	RW	Vendor Specific Error Signal Enable field from Error Interrupt Signal Enable Register
30	ERR_INT_SIGNAL_EN_VEN_DOR_SPECIFIC_STAT	RW	Vendor Specific Error Signal Enable field for ERR_INT_STS_VENDOR_SPECIFIC_STAT interrupt
26	ERR_INT_SIGNAL_EN_TUNING_ERR	RW	Tuning Error Signal Enable field from Error Interrupt Signal Enable Register
25	ERR_INT_SIGNAL_EN_ADMA_ERR	RW	ADMA Error Signal Enable field from Error Interrupt Signal Enable Register
24	ERR_INT_SIGNAL_EN_AUTO_CMD_ERR	RW	Auto CMD Error Signal Enable field from Error Interrupt Signal Enable Register
23	ERR_INT_SIGNAL_EN_CURRENT_LIMIT_ERR	RW	Current Limit Error Signal Enable field from Error Interrupt Signal Enable Register
22	ERR_INT_SIGNAL_EN_DATA_END_BIT_ERR	RW	Data End Bit Error Signal Enable field from Error Interrupt Signal Enable Register
21	ERR_INT_SIGNAL_EN_DATA_CRC_ERR	RW	Data CRC Error Signal Enable field from Error Interrupt Signal Enable Register
20	ERR_INT_SIGNAL_EN_DATA_TIMEOUT	RW	Data Timeout Error Signal Enable field from Error Interrupt Signal Enable Register
19	ERR_INT_SIGNAL_EN_COMMAND_INDEX_ERR	RW	Command Index Error Signal Enable field from Error Interrupt Signal Enable Register
18	ERR_INT_SIGNAL_EN_COMMAND_END_BIT_ERR	RW	Command End Bit Error Signal Enable field from Error Interrupt Signal Enable Register
17	ERR_INT_SIGNAL_EN_COMMAND_CRC_ERR	RW	Command CRC Error Signal Enable field from Error Interrupt Signal Enable Register
16	ERR_INT_SIGNAL_EN_COMMAND_TIMEOUT	RW	Command Timeoutout Error Signal Enable field from Error Interrupt Signal Enable Register
15	NORMAL_INT_SIGNAL_EN_ERR_INT	R	Error Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
14	NORMAL_INT_SIGNAL_EN_CMD_QUEUE	RW	Command Queueing Signal Enable field from Normal Interrupt Signal Enable Register
12	NORMAL_INT_SIGNAL_EN_RETUNING_EVENT	RW	Re-Tuning Event Signal Enable field from Normal Interrupt Signal Enable Register
11	NORMAL_INT_SIGNAL_EN_INT_C	RW	INT_C Signal Enable field from Normal Interrupt Signal Enable Register
10	NORMAL_INT_SIGNAL_EN_INT_B	RW	INT_B Signal Enable field from Normal Interrupt Signal Enable Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_38\_3A (cont.)**

Bits	Name	Type	Description
9	NORMAL_INT_SIGNAL_EN_INT_A	RW	INT_A Signal Enable field from Normal Interrupt Signal Enable Register
8	NORMAL_INT_SIGNAL_EN_CARD_INT	RW	Card Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
7	NORMAL_INT_SIGNAL_EN_CARD_REMOVAL	RW	Card Removal Signal Enable field from Normal Interrupt Signal Enable Register
6	NORMAL_INT_SIGNAL_EN_CARD_INSERTION	RW	Card Insertion Signal Enable field from Normal Interrupt Signal Enable Register
5	NORMAL_INT_SIGNAL_EN_BUF_RD_READY	RW	Buffer Read Ready Signal Enable field from Normal Interrupt Signal Enable Register
4	NORMAL_INT_SIGNAL_EN_BUF_WR_READY	RW	Buffer Write Ready Signal Enable field from Normal Interrupt Signal Enable Register
3	NORMAL_INT_SIGNAL_EN_DMA_INT	RW	DMA Interrupt Signal Enable field from Normal Interrupt Signal Enable Register
2	NORMAL_INT_SIGNAL_EN_BLK_GAP_EVENT	RW	Block Gap Event Signal Enable field from Normal Interrupt Signal Enable Register
1	NORMAL_INT_SIGNAL_EN_TRANS_COMPLETE	RW	Transfer Complete Signal Enable field from Normal Interrupt Signal Enable Register
0	NORMAL_INT_SIGNAL_EN_CMD_COMPLETE	RW	Command Complete Signal Enable field from Normal Interrupt Signal Enable Register

**0x0C08403C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_3C\_3E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Auto CMD Error Status Register(15-0)

-Host Control 2 Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_3C\_3E**

Bits	Name	Type	Description
31	HST_CTL2_PRESET_VALUE_EN	RW	Preset Value Enable field from Host Control2 Register
30	HST_CTL2_ASYNC_INT_EN	RW	Asynchronous Interrupt Enable field from Host Control2 Register
23	HST_CTL2_SAMPL_CLK_SEL	RW	Sampling Clock Select field from Host Control2 Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_3C\_3E (cont.)**

Bits	Name	Type	Description
22	HST_CTL2_EXEC_TUNING	RW	Execute Tuning field from Host Control2 Register
21:20	HST_CTL2_DRIVER_STRENGTH_SEL	RW	Driver Strength Select field from Host Control2 Register
19	HST_CTL2_SIGNALING_1_8_EN	RW	1.8V Signaling Enable field from Host Control2 Register
18:16	HST_CTL2_UHS_MODE_SEL	RW	UHS Mode Select field from Host Control2 Register
7	AUTO_CMD_NOT_ISSUED_BY_AUTO_CMD12	R	Command Not issued By Auto CMD12 Error field from Auto CMD Error Status Register
4	AUTO_CMD_INDX_ERR	R	Auto CMD Index Error field from Auto CMD Error Status Register
3	AUTO_CMD_END_BIT_ER	R	Auto CMD End Bit Error field from Auto CMD Error Status Register
2	AUTO_CMD_CRC_ERR	R	Auto CMD CRC Error field from Auto CMD Error Status Register
1	AUTO_CMD_TIMEOUT	R	Auto CMD Timeout Error field from Auto CMD Error Status Register
0	AUTO_CMD12_NOT_EXEC	R	Auto CMD12 Not Executed field from Auto CMD Error Status Register

**0x0C084040 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_40\_42****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x3029C8B2

The 32 LSBs of the Capabilities Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_40\_42**

Bits	Name	Description
31:30	CAPABILITIES_SLOT_TYPE	Slot Type field from Capabilities Register
29	CAPABILITIES_ASYNC_INT_SUPPORT	Asynchronous Interrupt Support field from Capabilities Register
28	CAPABILITIES_SYS_BUS_SUPPORT_64_BIT	64 bit System Bus Support field from Capabilities Register
26	CAPABILITIES_VOLTAGE_SUPPORT_1_8V	Voltage Support 1.8v field from Capabilities Register
25	CAPABILITIES_VOLTAGE_SUPPORT_3_0V	Voltage Support 3.0v field from Capabilities Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_40\_42 (cont.)**

Bits	Name	Description
24	CAPABILITIES_VOLTAGE_SUPORT_3_3V	Voltage Support 3.3v field from Capabilities Register
23	CAPABILITIES_SUSPEND_RESUME_SUPPORT	Suspend Resume Support field from Capabilities Register
22	CAPABILITIES_SDMA_SUPPORT	SDMA Support field from Capabilities Register
21	CAPABILITIES_HS_SUPPORT	High Speed Support field from Capabilities Register
19	CAPABILITIES_ADMA2_SUPPORT	ADMA2 Support field from Capabilities Register
18	CAPABILITIES_SUPPORT_8_BIT	8 bit Supported For Embedded Device field from Capabilities Register
17:16	CAPABILITIES_MAX_BLK_LENGTH	Max Block Length field from Capabilities Register
15:8	CAPABILITIES_BASE_SDCLK_FREQ	Base Clock Frequency For SD Clock field from Capabilities Register
7	CAPABILITIES_TIMEOUT_CLK_UNIT	Timeout Clock Unit field from Capabilities Register
5:0	CAPABILITIES_TIMEOUT_CLK_FREQ	Timeout Clock Frequency field from Capabilities Register

**0x0C084044 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_44\_46****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00008007

The 32 MSBs of the Capabilities Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_44\_46**

Bits	Name	Description
23:16	CAPABILITIES_CLK_MULTIPLIER	Clock Multiplier field from Capabilities Register
15:14	CAPABILITIES_RETUNING_MODE	Re-Tuning Modes field from Capabilities Register
13	CAPABILITIES_USE_TUNING_FOR_SDR50	Use Tuning for SDR50 field from Capabilities Register
11:8	CAPABILITIES_TIMER_CNT_FOR_RETUNING	Timer Count for Re-Tuning field from Capabilities Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_44\_46 (cont.)**

Bits	Name	Description
6	CAPABILITIES_DRIVER_TYPE_D_SUPPORT	Driver Type D Support field from Capabilities Register
5	CAPABILITIES_DRIVER_TYPE_C_SUPPORT	Driver Type C Support field from Capabilities Register
4	CAPABILITIES_DRIVER_TYPE_A_SUPPORT	Driver Type A Support field from Capabilities Register
2	CAPABILITIES_DDR_50_SUPPORT	DDR50 Support field from Capabilities Register
1	CAPABILITIES_SDR_104_SUPPORT	SDR104 Support field from Capabilities Register
0	CAPABILITIES_SDR_50_SUPPORT	SDR50 Support field from Capabilities Register

**0x0C084048 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_48\_4A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Maximum Current Register in SD Host Controller Standard Specification.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_48\_4A**

Bits	Name	Description
23:16	MAX_CURRENT_1_8V	Maximum Current for 1.8v field from Maximum Current Capabilities Register
15:08	MAX_CURRENT_3_0V	Maximum Current for 3.0v field from Maximum Current Capabilities Register
07:00	MAX_CURRENT_3_3V	Maximum Current for 3.3v field from Maximum Current Capabilities Register

**0x0C084050 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_50\_52****Type:** W**Clock:** same rate as HCLK**Reset State:** 0x00000000

Combination of two registers from the SD Host Controller Standard Specification:

-Force Event Register for Auto CMD Error Status(15-0)

-Force Event Register for Error Interrupt Status(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_50\_52**

Bits	Name	Description
31:28	FORCE_EVENT_FOR_ERR_VENDOR_SPECIFIC_ERR_STS	Force Event for Vendor Specific Error Status field from Force Event Register for Error Interrupt Status
25	FORCE_EVENT_FOR_ERR_ADMA	Force Event for ADMA Error field from Force Event Register for Error Interrupt Status
24	FORCE_EVENT_FOR_ERR_AUTO_CMD	Force Event for Auto CMD Error field from Force Event Register for Error Interrupt Status
23	FORCE_EVENT_FOR_ERR_CURRENT_LIMIT	Force Event for Current Limit Error field from Force Event Register for Error Interrupt Status
22	FORCE_EVENT_FOR_ERR_DATA_END_BIT	Force Event for Data End Bit Error field from Force Event Register for Error Interrupt Status
21	FORCE_EVENT_FOR_ERR_DATA_CRC	Force Event for Data CRC Error field from Force Event Register for Error Interrupt Status
20	FORCE_EVENT_FOR_ERR_DATA_TIMEOUT	Force Event for Data Timeout Error field from Force Event Register for Error Interrupt Status
19	FORCE_EVENT_FOR_ERR_CMD_INDX	Force Event for Command Index Error field from Force Event Register for Error Interrupt Status
18	FORCE_EVENT_FOR_ERR_CMD_END_BIT	Force Event for Command End Bit Error field from Force Event Register for Error Interrupt Status
17	FORCE_EVENT_FOR_ERR_CMD_CRC	Force Event for Command CRC Error field from Force Event Register for Error Interrupt Status
16	FORCE_EVENT_FOR_ERR_CMD_TIMEOUT	Force Event for Command Timeout Error field from Force Event Register for Error Interrupt Status
7	FORCE_EVENT_FOR_CMD_NOT_ISSUED_BY_AUTO_CMD12	Force Event for Command Not Issued By Auto CMD12 Error field from Force Event Register for Auto CMD Error Status
4	FORCE_EVENT_FOR_AUTO_CMD_INDX_ERR	Force Event for Auto CMD Index Error field from Force Event Register for Auto CMD Error Status
3	FORCE_EVENT_FOR_AUTO_CMD_END_BIT_ERR	Force Event for Auto CMD End Bit Error field from Force Event Register for Auto CMD Error Status
2	FORCE_EVENT_FOR_AUTO_CMD_CRC_ERR	Force Event for Auto CMD CRC Error field from Force Event Register for Auto CMD Error Status
1	FORCE_EVENT_FOR_AUTO_CMD_TIMEOUT_ERR	Force Event for Auto CMD Timeout Error field from Force Event Register for Auto CMD Error Status
0	FORCE_EVENT_FOR_AUTO_CMD12_NOT_EXEC	Force Event for Auto CMD12 Not Executed field from Force Event Register for Auto CMD Error Status

**0x0C084054 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_54\_56****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 7 LSBs are the ADMA Error Status Register in SD Host Controller Standard Specification.  
The Bits 31-8 are unused.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_54\_56**

Bits	Name	Description
2	ADMA_LENGTH_MISMATCH_H_ERR	ADMA Length Mismatch Error field from ADMA Error Status Register
1:0	ADMA_ERR_STATE	ADMA Error State field from ADMA Error Status Register

**0x0C084058 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_58\_5A****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 32 LSBs of the ADMA System Address Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_58\_5A**

Bits	Name	Description
31:00	ADMA_SYS_ADDRESS	ADMA System Address field from ADMA System Address Register

**0x0C08405C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_5C\_5E****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

The 32 MSBs of the ADMA System Address Register in SD Host Controller Standard Specification

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_5C\_5E**

Bits	Name	Description
31:00	ADMA_SYS_ADDRESS_64	upper 32 bits of ADMA System Address field from ADMA System Address Register

**0x0C084060 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_60\_62****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00040000

Combination of two registers from the SD Host Controller Standard Specification:

- Preset Value for Initialization Register(15-0)

- Preset Value for Default Speed Register(31-16)

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_60\_62**

Bits	Name	Description
31:30	DEFAULT_SPEED_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for Default Speed
26	DEFAULT_SPEED_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for Default Speed
25:16	DEFAULT_SPEED_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for Default Speed
15:14	INIT_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for Initialization
10	INIT_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for Initialization
9:0	INIT_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for Initialization

**0x0C084064 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_64\_66****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00040002

Combination of two registers from the SD Host Controller Standard Specification:

- Preset Value for High Speed Register(15-0)

- Preset Value for SDR12 Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_64\_66**

Bits	Name	Description
31:30	SDR12_DRIVER_STRENGT_H_SEL	Driver Strength Select Value field from Preset Value for SDR12
26	SDR12_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR12

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_64\_66 (cont.)**

Bits	Name	Description
25:16	SDR12_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR12
15:14	HS_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for High Speed
10	HS_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for High Speed
9:0	HS_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for High Speed

**0x0C084068 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_68\_6A****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00010002

Combination of two registers from the SD Host Controller Standard Specification:

-Preset Value for SDR25 Register(15-0)

-Preset Value for SDR50 Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_68\_6A**

Bits	Name	Description
31:30	SDR50_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR50
26	SDR50_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR50
25:16	SDR50_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR50
15:14	SDR25_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR25
10	SDR25_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR25
9:0	SDR25_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR25

**0x0C08406C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_6C\_6E****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00020000

Combination of two registers from the SD Host Controller Standard Specification:

-Preset Value for SDR104 Register(15-0)

-Preset Value for DDR50 Register

#### **PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_6C\_6E**

Bits	Name	Description
31:30	DDR50_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for DDR50
26	DDR50_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for DDR50
25:16	DDR50_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for DDR50
15:14	SDR104_DRIVER_STRENGTH_SEL	Driver Strength Select Value field from Preset Value for SDR104
10	SDR104_CLK_GEN_SEL	Clock Generator Select Value field from Preset Value for SDR104
9:0	SDR104_SDCLK_FREQ_SEL	SDCLK Frequency Select Value field from Preset Value for SDR104

#### **0x0C0840E0 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_E0\_E2**

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

Shared Bus Control Register in SD Host Controller Standard Specification.

#### **PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_E0\_E2**

Bits	Name	Type	Description
30:24	SHARED_BUS_CTL_BACK_END_PWR	RW	Back End Power Control field from Shared Bus Control Register
22:20	SHARED_BUS_INT_PIN_SEL	RW	
18:16	SHARED_BUS_CTL_CLK_PIN_SEL	RW	Clock Pin Select field from Shared Bus Control Register
14:8	SHARED_BUS_CTL_BUS_WIDTH_PRESET	R	Bus Width Preset field from Shared Bus Control Register
5:4	SHARED_BUS_CTL_NUM_INTERRUPT_INPUT_PINS	R	Number Of Interrupt Input Pins field from Shared Bus Control Register
2:0	SHARED_BUS_CTL_NUM_CLK_PINS	R	Number of Clock Pins field from Shared Bus Control Register

**0x0C0840FC PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_FC\_FE****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00020000

Combination of two registers from the SD Host Controller Standard Specification:

- Slot Interrupt Status Register(15-0)

- Host Controller Version Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_FC\_FE**

Bits	Name	Type	Description
31:24	RESERVED		RESERVED
23:16	RESERVED		RESERVED
7:0	INT_SIGNAL_FOR_EACH_SLOT	RW	Interrupt Signal For Each Slot field from Slot Interrupt Status Register

**0x0C084200 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x6000642C

This register configures the DLL's inputs.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG**

Bits	Name	Description
31	SDC4_DIS_DOUT	When this bit equals to '0', the data which comes out from the dll is valid. When it equals to '1' this data isn't valid. The inversion of this bit is connected to dll's input - dll_en_dout
30	DLL_RST	Setting this bit to 1 resets cm_dll_sdc4. cm_dll_sdc4 should be reset every time the MCLK frequency is changed.
29	PDN	Power Down Value 0 - analog blocks are enabled Value 1 - analog blocks are powered down (default)
28	CK_INTP_SEL	Selects interpolator output
27	CK_INTP_EN	Enable clock interpolation for finer resolution

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
26:24	MCLK_FREQ	Frequency of MCLK Value 000 -100 - 112 (MHz) Value 001 -112 - 125 Value 010 -125 - 137 Value 011 -137 - 150 Value 100 -150 - 162 Value 101 -162 - 175 Value 110 -175 - 187 Value 111 -187 - 200
23:20	CDR_SELEXT	CDR phase select (gray coded) Value 0000 - phase 0 Value 0001 - phase 1 Value 0011 - phase 2 Value 0010 - phase 3 Value 0110 - phase 4 Value 0111 - phase 5 Value 0101 - phase 6 Value 0100 - phase 7 Value 1100 - phase 8 Value 1101 - phase 9 Value 1111 - phase 10 Value 1110 - phase 11 Value 1010 - phase 12 Value 1011 - phase 13 Value 1001 - phase 14 Value 1000 - phase 15
19	CDR_EXT_EN	Enable external control of cdr phase select
18	CK_OUT_EN	Enable output clock (default value is '1')
17	CDR_EN	Enable CDR function
16	DLL_EN	Enable DLL function
15:14	CDR_UPD_RATE	CDR update rate, low pass filtering window of CDR Max update rate: Value 00 - 0.5 MCLK frequency Value 01 - 0.25 MCLK Frequency (default) Value 10 - 0.125 MCLK frequency Value 11 - 1/16 MCLK Frequency
13:12	DLL_UPD_RATE	DLL update rate Value 00 - sdc4_mclk/10 Value 01 - sdc4_mclk/20 Value 10 - sdc4_mclk/40 (default) Value 11 - sdc4_mclk/80

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG (cont.)**

Bits	Name	Description
11:10	DLL_PHASE_DET	DLL phase detector low pass average window Value 00 - 2 cycles Value 01 - 4 cycles (default) Value 10 - 8 cycles Value 11 - 16 cycles
9	CDR_ALGORITHM_SEL	CDR algorithm select Value 0 - Edge (default) Value 1 - Level
8	DLY_LINE_SWITCH_CLK	value 0 (default) - Uses MCLK for switching between two delay-lines in programmable RCLK delay generation. value 1 - Uses TCXO clock for interpolating between two delay-lines in programmable RCLK delay generation.
7:6	CDR_PHASE_SEL_MODE	00 (default) - CDR phase is selected using all 8 input data 01 - CDR phase is selected using DATA_IN<1> only 10 - CDR phase is selected using DATA_IN<2> only 11 - CDR phase is selected using DATA_IN<3> only
5	MCLK_GATING_ENABLE	value '1' (default) - MCLK gating Enabled. '0' - No MCLK gating
4	FINE_PHASE_ENABLE	value '0' (default) - fine phase mode disabled. '1' - enable fine phase mode
3:2	CDR_FINE_PHASE	Default Value = 11 When fine phase mode is enabled, these two bits together with SDC4_CDR_SELEXT are used to generate the 64 phases Selected phase : 4*SDC4_CDR_SELEXT(Decimal Value)+ DECIMAL Equivalent of CONFIG<2:3>
1	RESERVED	RESERVED
0	CMD_DAT_TRACK_SEL	Select DAT or CMD lines for CDR tracking Value 0 - track DAT inputs (default) Value 1 - track CMD input

**0x0C084208 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_STATUS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

This register is connected to cm\_dll\_sdc4's status output.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_STATUS**

Bits	Name	Description
12	RESERVED	RESERVED

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_STATUS (cont.)**

Bits	Name	Description
11	DDR_DLL_LOCK_JDR	DLL lock for HS400 operation
10:9	RESERVED	RESERVED
8	SDC4_DLL_LOCK_ATPG	SDC4_DLL_LOCK output from DLL after a clock mux with sampled JDRs for ATPG mode. This bit helps collecting coverage on DLL's output
7	DLL_LOCK	DLL lock status Value 0 - Not locked Value 1 - Locked
6:3	CDR_PHASE	CDR phase select (gray coded) Value 0000 - phase 0 Value 0001 - phase 1 Value 0011 - phase 2 Value 0010 - phase 3 Value 0110 - phase 4 Value 0111 - phase 5 Value 0101 - phase 6 Value 0100 - phase 7 Value 1100 - phase 8 Value 1101 - phase 9 Value 1111 - phase 10 Value 1110 - phase 11 Value 1010 - phase 12 Value 1011 - phase 13 Value 1001 - phase 14 Value 1000 - phase 15
2	DDLL_COARSE_CAL	Value 1 - done Value 0 - not done
0	DDR_DLL_LOCK	DLL lock for HS400 operation

**0x0C084210 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0xF88218A8**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2**

Bits	Name	Type	Description
31	QSB_AXI_ABURST	RW	Driving QSB AXI output
30:28	NUM_OUTSTANDING_DATA	RW	Configuring the number of outstanding transactions for the data. The descriptors accesses have more one outstanding request

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
27	PROCEED_AXI_AFTER_ER_R	RW	When error occurs on AXI master the traffic can continue or terminate. If this bit is set then the traffic continues and more requests are transmitted on the address channel
26	QSB_AXI_AFULL_CALC	RW	If this bit is set, then afull isn't '0' constantly but can be equal to '1' if all the beats have 8 bytes on the strobe lines
25	ONE_MID_SUPPORT	RW	If this bit is set, then only 1 MID is used for descriptors and data
24:22	QSB_AXI_READ_MEMTYP_E	RW	Driving VMIDMT's input for read accesses
21	HC_SW_RST_REQ	RW	Before executing HC_SW_RST_FOR_ALL or HC_SW_RST_FOR_DAT this bit can be set and SW needs to polls on it until it's being cleared. The bit is cleared after AXI master (QMB) has finished processing the outstanding transactions and it's in IDLE state. This bit can be useful for HPI feature.
20	HC_SW_RST_WAIT_IDLE_DIS	RW	When this bit is cleared, the HC_SW_RST is executed only after the AXI master (QMB) is in IDLE state. This bit Can be used for HPI feature instead of polling on HC_SW_RST_REQ bit. This bit should be cleared for SW reset from unknown states.
19	SDCC5_HALT_REQ	R	Reading SDCC5_HALT_REQ input
18	SDCC5_HALT_ACK	R	Reading SDCC5_HALT_ACK output
17	SDCC5_M_IDLE	R	Reading SDCC5_M_IDLE output
16	SDCC5_HALT_ACK_SW_EN	RW	Enabling the SW value for SDCC5_HALT_ACK
15	SDCC5_HALT_ACK_SW	RW	Driving SDCC5_HALT_ACK from SW and not by HW
14	SDCC5_M_IDLE_DIS	RW	If this bit is set, then sdcc5_m_idle will be tied low
13	QSB_AXI_INTERLEAVING_EN	RW	If the bit is set, then the controller can support interleaving between two bursts.
12	QSB_AXI_TRANSIENT	RW	Driving VMIDMT's input
11	QSB_AXI_PROTNs	RW	Driving VMIDMT's input
10:9	QSB_AXI_REQPRIORITY	RW	Driving VMIDMT's input
8:6	QSB_AXI_MEMTYPE	RW	Driving VMIDMT's input
5	QSB_AXI_NOALLOCATE	RW	Driving VMIDMT's input
4	QSB_AXI_INNERSHARED	RW	Driving VMIDMT's input
3	QSB_AXI_SHARED	RW	Driving VMIDMT's input
2	QSB_AXI_OOWR	RW	Driving QSB AXI output
1	QSB_AXI_OOORD	RW	Driving QSB AXI output

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2 (cont.)**

Bits	Name	Type	Description
0	QSB_AXI_AFULL	RW	Driving QSB AXI output

**0x0C084214 PERIPH\_SS\_SDC2\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Last AHB address phase before response error was detected

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0**

Bits	Name	Description
31	CMDQ_TASK_DESC_ERRO R	CMDQ task descriptor AXI response error
30	ADMA_VALID_ERROR	descriptor not valid
29	ADMA_RESP_ERROR	descriptor read error.
28	DATA_RESP_ERROR	AXI data read response error.
27	DATA_WR_RESP_ERROR	AAXI data write response error.
26	ADMA_LEN_MISMATCH	ADMA length mismatch error
25:13	RESP_ADDR_MSB	response error address - MSBs
12:0	RESP_ADDR_LSB	response error address - LSBs

**0x0C08421C PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x3029C8B2

driving the HC\_REG\_40\_42 register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0**

Bits	Name	Description
31:30	VS_CAPABILITIES_SLOT_T YPE	Slot Type field from Capabilities Register
29	VS_CAPABILITIES_ASYNC_ INT_SUPPORT	Asynchronous Interrupt Support field from Capabilities Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0 (cont.)**

Bits	Name	Description
28	VS_CAPABILITIES_SYS_BU S_SUPPORT_64_BIT	64 bit System Bus Support field from Capabilities Register
26	VS_CAPABILITIES_VOLTAG E_SUPPORT_1_8V	Voltage Support 1.8v field from Capabilities Register
25	VS_CAPABILITIES_VOLTAG E_SUPPORT_3_0V	Voltage Support 3.0v field from Capabilities Register
24	VS_CAPABILITIES_VOLTAG E_SUPPORT_3_3V	Voltage Support 3.3v field from Capabilities Register
23	VS_CAPABILITIES_SUSPE ND_RESUME_SUPPORT	Suspend Resume Support field from Capabilities Register
22	VS_CAPABILITIES_SDMA_ SUPPORT	SDMA Support field from Capabilities Register
21	VS_CAPABILITIES_HS_SUP PORT	High Speed Support field from Capabilities Register
19	VS_CAPABILITIES_ADMA2_ SUPPORT	ADMA2 Support field from Capabilities Register
18	VS_CAPABILITIES_SUPPO RT_8_BIT	8 bit Supported For Embedded Device field from Capabilities Register
17:16	VS_CAPABILITIES_MAX_BL K_LENGTH	Max Block Length field from Capabilities Register
15:8	VS_CAPABILITIES_BASE_S DCLK_FREQ	Base Clock Frequency For SD Clock field from Capabilities Register
7	VS_CAPABILITIES_TIMEOUT T_CLK_UNIT	Timeout Clock Unit field from Capabilities Register
5:0	VS_CAPABILITIES_TIMEOUT T_CLK_FREQ	Timeout Clock Frequency field from Capabilities Register

**0x0C084220 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0200808F

driving the HC\_REG\_44\_46 register and spec version

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1**

Bits	Name	Type	Description
31:24			
23:16	VS_CAPABILITIES_CLK_M ULTIPLIER	RW	Clock Multiplier field from Capabilities Register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1 (cont.)**

Bits	Name	Type	Description
15:14	VS_CAPABILITIES_RETUNING_MODE	RW	Re-Tuning Modes field from Capabilities Register
13	VS_CAPABILITIES_USE_TUNING_FOR_SDR50	RW	Use Tuning for SDR50 field from Capabilities Register
12	VS_CAPABILITIES_FUSE_ICE_DISABLE	R	reflects the value of ICE disable fuse. When it is '1' - ICE is disabled for this controller
11:8	VS_CAPABILITIES_TIMER_CNT_FOR_RETUNING	RW	Timer Count for Re-Tuning field from Capabilities Register
7	VS_CAPABILITIES_CMDQ_SUPPORT	R	CMDQ Support field from Capabilities Register - READ ONLY BIT
6	VS_CAPABILITIES_DRIVER_TYPE_D_SUPPORT	RW	Driver Type D Support field from Capabilities Register
5	VS_CAPABILITIES_DRIVER_TYPE_C_SUPPORT	RW	Driver Type C Support field from Capabilities Register
4	VS_CAPABILITIES_DRIVER_TYPE_A_SUPPORT	RW	Driver Type A Support field from Capabilities Register
3	VS_CAPABILITIES_HS400_SUPPORT	R	HS400 Support field from Capabilities Register - READ ONLY BIT
2	VS_CAPABILITIES_DDR_50_SUPPORT	RW	DDR50 Support field from Capabilities Register
1	VS_CAPABILITIES_SDR_104_SUPPORT	RW	SDR104 Support field from Capabilities Register
0	VS_CAPABILITIES_SDR_50_SUPPORT	RW	SDR50 Support field from Capabilities Register

**0x0C084224 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG****Type:** RW**Clock:** HCLK**Reset State:** 0x00000000

Configuration bits for e.MMC5.0 DDR200 mode.

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG**

Bits	Name	Description
10	FF_CLK_DIS	The ff_clk to CDC/TBY4 is enabled by default. It can be disabled by setting this bit. The ff_clk is enabled to CDC and TBY4 according to CDC_T4_DLY_SEL field
9	VOLTAGE_MUX_SEL	select between 1.8v and 1.2v signals on sdcc5 IO interface with device. default is 0 to choose 1.8v

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_DDR200\_CFG (cont.)**

Bits	Name	Description
8:7	CDC_TRAFFIC_SEL	Selecting the method of cdc_traffic's assertions: Value 0 (default) - auto calibration with stand by feature. Value 1 - SW calibration before any transaction. Value 2 - automatic calibration with HW assertion of cdc_traffic
6	START_CDC_TRAFFIC	This bit starts the cdc traffic assertion after calibration done is received
5	CRC_TOKEN_SAMPL_FALL_EDGE	If set, the start bit of incoming CRC token in DDR200 mode is sampled in falling edge. If this bit is cleared (default) then the start bit is detected in rising edge
4	DATIN_SAMPL_FALL_EDGE	If set, the start bit of incoming data in DDR200 mode is sampled in falling edge. If this bit is cleared (default) then the start bit is detected in rising edge
3	RESERVED	RESERVED
2	CMDIN_EDGE_SEL	Select sampling edge of CMD input when CMDIN_RCLK_EN is 1: 0 (default) - falling edge of RCLK, 1 - rising edge of RCLK
1	CMDIN_RCLK_EN	Enable CMD input sampling with RCLK: 0 (default) - CMD input is sampled by SD DLL output clock, 1 - CMD input is sampled by delayed RCLK
0	CDC_T4_DLY_SEL	Select block for RCLK delay: 0 (default) - CM_DLL_SDC4 CDCLP533 and CM_TBY4

**0x0C084240 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG****Type:** R**Clock:** PCLK**Reset State:** 0x00000000

HC Spec Defines power and IO voltage switch bits. The information is sent by a dedicated interrupt to the QTI SW driver for handling the power and voltage tasks with PMIC. The different tasks are defined in this register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG**

Bits	Name	Description
3	IO_HIGH_V	SDCC driver orders that the I/O signaling level be switched to 3.0v. This interrupt is triggered when SDCC driver writes '0' to offset 03Eh in HC register (bit 3) or when SW reset for all is handled and last supported voltage was 1.8V. This logic is enabled when SWITCHABLE_SIGNALING_VOLTAGE=true
2	IO_LOW_V	SDCC driver orders that the I/O signaling level be switched to 1.8v. This interrupt is triggered when SDCC driver writes '1' to offset 03Eh in HC register (bit 3). This logic is enabled when SWITCHABLE_SIGNALING_VOLTAGE=true

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG (cont.)**

Bits	Name	Description
1	BUS_ON	SDCC driver orders that the SD bus power be switched on. This interrupt is triggered when SDCC driver writes '1' to offset 029h in HC register (bit 0).
0	BUS_OFF	SDCC driver orders that the SD bus power be switched off. This interrupt is triggered when SDCC driver writes '0' to offset 029h in HC register(bit 0) or when SW reset for all is handled.

**0x0C084244 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG****Type:** RW**Clock:** PCLK**Reset State:** 0x00000000

Mask register for MCI\_PWRCTL\_STATUS\_REG

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG**

Bits	Name	Description
3	IO_HIGH_V	'1' - interrupt is enabled / '0' - interrupt is masked.
2	IO_LOW_V	'1' - interrupt is enabled / '0' - interrupt is masked.
1	BUS_ON	'1' - interrupt is enabled / '0' - interrupt is masked.
0	BUS_OFF	'1' - interrupt is enabled / '0' - interrupt is masked.

**0x0C084248 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG****Type:** W**Clock:** PCLK**Reset State:** 0x00000000

Clear register for MCI\_PWRCTL\_STATUS\_REG

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG**

Bits	Name	Description
3	IO_HIGH_V	Writing '1' clears bit 03 in PWRCTL_STATUS_REG
2	IO_LOW_V	Writing '1' clears bit 02 in PWRCTL_STATUS_REG
1	BUS_ON	Writing '1' clears bit 01 in PWRCTL_STATUS_REG
0	BUS_OFF	Writing '1' clears bit 00 in PWRCTL_STATUS_REG

**0x0C084250 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC3****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x02226040

Vendor specific register #3

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC3**

Bits	Name	Type	Description
31:30	BUSY_CHECK_VALID_PERIOD	RW	Configuring the wait period between CRC token to BUSY period. Value 0 - waiting 7 cycles. The values of BUSY_CHECK_VALID_PERIOD is added to 7 cycles
29	BUSY_CHECK_VALID_ALWAYS	RW	In DDR200 mode, the end of busy should be checked after the CRC token was received. There is a counter of 4 cycles for this purpose. If this bit is set then the end of busy will be checked right after the crc token.
28	ADMA_INT_DATA	RW	If this bit is set, then the ADMA interrupt is set when the data was already transferred. If this bit is cleared, then the ADMA interrupt is set when the requests related to the descriptors were finished on address channel before the data itself is processed
27	IRQ_PCLK_DIS	RW	From tag sdcc5_p3q3r50, the interrupt and status bits are handled in PCLK domain only. If this bit is set, then the legacy architecture is used
26	CLOCK_AFTER_CMDEND_DIS	RW	
25	SDCC_CLK_EXT_EN	RW	
24:23	SD_DEV_SEL	RW	Select the active device if more than one device are connected in shared bus mode. (range: '00' - '11').
22	SD_CLK_STABLE	R	When changing timing mode from/to DDR, this bit will be set to '0'. it will be set to '1' after the SD clock was division is completed
21	TCXO_SW_RST_EN	RW	If set (1), SW_RESET (reset_a) will wait for TCXO reset complete ACK to be deasserted. default = 0.
19	HS400_BLK_END_RST_DISABLE	RW	Disable reset of HS400 Data receivers at end of block in RX transaction. Default = '0'.
18:16	DATEN_HS400_INPUT_MASK_CNT	RW	how many cycles after DATEN keep masking of input DATA bit at HS400.#cycles = 4 + CMDEN_HS400_INPUT_MASK_CNT. Default = 2
15:13	CMDEN_HS400_INPUT_MASK_CNT	RW	how many cycles after CMDEN keep masking of input CMD bit at HS400.#cycles = 4 + CMDEN_HS400_INPUT_MASK_CNT. Default = 2

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC3 (cont.)**

Bits	Name	Type	Description
12	DATEN_HS400_INPUT_MA_SK_DIS	RW	Disable masking of input DATA bits at HS400 mode while the controller drives the bus is disabled. Default = '0'.
11	CMDEN_HS400_INPUT_MA_SK_DIS	RW	Disable masking of input CMD bit at HS400 mode while the controller drives the bus is disabled. Default = '0'.
10	HC_FIFO_ALT_ENABLE	RW	Selecting the FIFO MODE method in HC - '0' (Default) - regular Buffer data read according to SDHCI. '1' - using MCI_FIFO_ALT
9	WIDEBUS_MASK_DISABLE	RW	When this bit is set, masking the LSBs according to bus width selection (8/4/1 DATA lines) is disabled
8	TXFLOWCONTROL_WITH_NO_TX	RW	When this bit is set, txflowcontrol keeps legacy behavior and can be on after the transaction had finished. Default is '0' - qualify txflowcontrol with txactive
7	MCLK_DURING_SW_RST_REQ_DIS	RW	When this bit is set, the mclk will not be automatically enabled during SW reset request. Default is '0' - enable the clock during SW reset request
6	HS200_ASYNC_FIFO_WR_CLK_EN	RW	When this bit is set, the wr_clk for async_fifo in HS200/HS400 will not be gated internally. default is '1'. In order to save idle power this bit can be cleared
5	IGNORE_START_BIT_ERR	RW	When this bit is set, Start Bit error will not be looked at. default is '0'
3	PWRSAVE_DLL	RW	When this bit is set, DLL will get clock during transactions but not in idle. When this bit is cleared, DLL will get clock all the time so sdc4_dll_lock won't be de-asserted
2	SDIO_TRANS	RW	When this bit is set in SDIO transaction, START_BIT_ERR status bit won't be raised. The reason is that interrupt period which is de-asserted only after the RX command can cause to wrong START_BIT_ERR detection. The START_BIT_ERR bit won't appear in MCI_STATUS when SDIO_TRANS is set
1	HCLK_IDLE_GATING	RW	Additional CGC for HCLK was added which gates the HCLK for status bits. If this bit is set then HCLK is gated for these registers during IDLE
0	MCLK_IDLE_GATING	RW	Additional CGC for MCLK was added which gates the MCLK for status bits. If this bit is set then MCLK is gated for these registers during IDLE

**0x0C084260 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00030022

AXI values and other vendor specific functions

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4**

Bits	Name	Description
23	DLL_CDR_EN_OVERRIDE	in CMDQ mode, the CQE controls on CDR_EN input of DLL. This bit is allow SW to override CQE control over CDR_EN with original value from HC_REG_DLL_CONFIG. Default is '0'
22:20	UARM_PRIORITY_MODE	Priority for Axi master read clients (ADMA_DESC_FETCH, CMDQ_TASK_DESC_FETCH, RAM_DATA_CLIENT)
19	SBFE_UAWM_DIS	Axi Write error disable
18	SBFE_UARM_DIS	Axi read error disable
17:16	AXI_MAX_BURST_LENGTH	Controls the max length of AXI burst initiated by SDCC master. AXI Burst length = $2^{(AXI\_MAX\_BURST\_LENGTH+4)}$ . Default = 3 for 128Byte bursts for best performance
15	DISABLE_CRYPTO	When set, HC_CMDQ_CAPABILITIES.CRYPTO_SUPPORT will be overridden to '0' even if controller includes ICE
14	WRAP_ERROR	When the bit is set, the core responds with errors on WRAP transaction in configuration bus.
13	RX_FLOW_TIMING	Configuration bit which selects the cycle which RxFlowControl will be asserted when UHS mode is used Value 0 (default) - RxFlowControl is asserted one clock before the end bit is received in DPSM (NextBlkTCnt in DPSM equals to 1). Value 1 - RxFlowControl is asserted two clocks before the end bit is received in DPSM (NextBlkTCnt in DPSM equals to 2).
12	PROGDONE_WO_CMD_RESP	'0' (Default): we wait for cmd response end to detect progdone (legacy behavior). '1': we start looking for progdone without waiting for cmd response end but after 10 cycles from end bit of command
11:10	HC_AXI_ARLOCK	AXI ARLOCK value. shared for QMB and SCM. Default = 0x0
9:8	HC_AXI_AWLOCK	AXI AWLOCK value. shared for QMB and SCM. Default = 0x0
7:4	HC_AXI_ARCACHE	AXI ARCACHE value. shared for QMB and SCM. Default = 0x2
3:0	HC_AXI_AWCACHE	AXI AWCACHE value. shared for QMB and SCM. Default = 0x2

**0x0C084254 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_2****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x0020A000

Second register for DLL's configuration

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_2**

Bits	Name	Description
22	DLL_OUTPUT_TO_PAD	RESERVED

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_2 (cont.)**

Bits	Name	Description
21	DLL_CLOCK_DISABLE	'1' (default) - DLL clock is disabled. '0' - dll clock has legacy clock enable. SW must keep this bit set until correct clock frequency is delivered to DLL
19	LOW_FREQ_MODE	'0' (default) - High Frequency Mode (frequency range 175MHz - 208MHz) '1' - Low Frequency Mode (Clock frequency from 87.5MHz - 104MHz)
18	FLL_CYCLE_CNT	Number of TCXO clock cycles for which FLL counts before calculating error. 0 (default) - 4 TCXO Clock Cycles.1 - 8 TCXO Clock Cycles
17:10	MCLK_FREQ_CALC	Round off to nearest integer [(Freq of MCLK / Freq of TCXO) * FLL_CYCLE_CNT(4/8)]. Example : Freq of MCLK = 200MHz Freq of TCXO = 19.2MHz FLL_CYCLE_CNT = 0. Round off [(200/19.2)*4] = 41.666 = 42 Bits.
3:2	DDR_TRAFFIC_INIT_SEL	Selecting how DDR_TRAFFIC_INIT will be driven to DLL. Value 00 - HW drives the DDR_TRAFFIC_INIT when DPSM is active and when CPSM is active if RCLK is expected during response. Value 01 - The value is driven by SW by SW_DDR_TRAFFIC_INIT bit. Value 10 - DDR_TRAFFIC_INIT to DLL is driven when sdc4_mclk is not gated. off.
1	DDR_TRAFFIC_INIT_SW	The signal should be high 3-4 cycles before RCLK is toggling. It should be high when MCLK is toggling.
0	DDR_CAL_EN	The signal is set after the configuration of MCI_DDR_CONFIG is finished. When the signal is low the TCXO domain is being reset.

**0x0C084258 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_3****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Third register for DLL's configuration

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_3**

Bits	Name	Description
7:0	SDC4_CONFIG_MSB	bit[23:16] of sdc4_config bus. for more info check data sheets in go/dll

**0x0C08425C PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DDR\_CONFIG****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x80040873

Configuration of ddr\_config input of DLL

**PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DDR\_CONFIG**

Bits	Name	Description
31	PRG_DLY_EN	Enables Programmable delay for RCLK_IN
30	EXT_PRG_RCLK_DLY_EN	Enables external control of Programmable RCLK delayline
29:27	EXT_PRG_RCLK_DLY_CODE	Used only if EXT_PRG_RCLK_DLY_EN is set. Configuring Programmable RCLK Delayline process code number.
26:21	EXT_PRG_RCLK_DLY	Used only if EXT_PRG_RCLK_DLY_EN is set. Programmable RCLK Delay line fine delay control.
20:12	TCXO_CYCLES_DLY_LINE	Number of TCXO clock cycles between finish of one delay line calibration and beginning of next delay line calibration.
11:9	TCXO_CYCLES_CNT	Number of TCXO clock cycles for which down counter is enabled for before making each inc/dec decision.
8:0	PRG_RCLK_DLY	These bits are controlled the programmable delay line. This bus is binary coded. Decimal value = (TCXO period * Decimal value of TCXO_CYCLES_CNT)/(2^delay needed). For 1.25ns the formula is (52n*4/2*1.25) = 83 in decimal.

**0x0C0842E4 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AW\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

AXI Address channel counter, counts a successful write address transaction

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AW\_MON**

Bits	Name	Type	Description
17	HC_AXI_AW_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_AW_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_AW_CNTR	R	counts a successful write address transaction

**0x0C0842E8 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_W\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Write data channel counter, counts a successful write data transaction

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_W\_MON**

Bits	Name	Type	Description
17	HC_AXI_W_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_W_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_W_CNTR	R	counts a successful write data transaction

**0x0C0842EC PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_B\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Write response channel counter, counts a successful write transaction

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_B\_MON**

Bits	Name	Type	Description
17	HC_AXI_B_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_B_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_B_CNTR	R	counts a successful write transaction

**0x0C0842F0 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AR\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Address channel counter, counts a successful read address transaction

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_AR\_MON**

Bits	Name	Type	Description
17	HC_AXI_AR_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_AR_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_AR_CNTR	R	counts a successful read address transaction

**0x0C0842F4 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_R\_MON****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Read data channel counter, counts a successful read data transaction

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AXI\_R\_MON**

Bits	Name	Type	Description
17	HC_AXI_R_CNTR_EOT_RST	RW	'1' - this counter is reset at the end of each transfer
16	HC_AXI_R_CNTR_SOT_RST	RW	'1' - this counter is reset at the start of each transfer
7:0	HC_AXI_R_CNTR	R	counts a successful read data transaction

**0x0C08435C PERIPH\_SS\_SDC2\_SDCC\_DATA\_COUNT****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

DPSM data counter register

**PERIPH\_SS\_SDC2\_SDCC\_DATA\_COUNT**

Bits	Name	Description
27:0	DATACOUNT	Value of data counter in McIDPSM block. Represents remaining data of transaction.

**0x0C0842F8 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt status register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS**

Bits	Name	Description
21	ERR_INT_STS_START_BIT_ERROR	Start Bit Error flag
20	ERR_INT_STS_FIFO_FULL_WR	
19	ERR_INT_STS_FIFO_EMPTY_RD	FIFO read attempt when FIFO was empty
18	ERR_INT_STS_AUTO_CMD19_TIMEOUT	Auto CMD19 timeout
17	ERR_INT_STS_BOOT_TIMEOUT	Data wasn't received within the valid time (according to BOOT_DATA_TIMER) from the start of boot operation.
16	ERR_INT_STS_BOOT_ACK_ERR	Acknowledge pattern wasn't received correctly or not within the valid time (according to BOOT_ACK_TIMER) from the start of boot operation.
0	NORMAL_INT_STS_BOOT_ACK_REC	Acknowledge pattern was received correctly.

**0x0C0842FC PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt status enable register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS\_EN**

Bits	Name	Description
21	ERR_INT_STS_EN_START_BIT_ERROR	START_BIT_ERROR status enable
20	ERR_INT_STS_EN_FIFO_FULL_WR	FIFO_FULL_WR status enable
19	ERR_INT_STS_EN_FIFO_EMPTY_RD	FIFO_EMPTY_RD status enable

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_STS\_EN (cont.)**

Bits	Name	Description
18	ERR_INT_STS_EN_AUTO_CMD19_TIMEOUT	AUTO_CMD19_TIMEOUT status enable
17	ERR_INT_STS_EN_BOOT_TIMEOUT	BOOT_TIMEOUT status enable
16	ERR_INT_STS_EN_BOOT_ACK_ERR	BOOT_ACK_ERR status enable
0	NORMAL_INT_STS_EN_BOOT_ACK_REC	BOOT_ACK_REC status enable

**0x0C084300 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_SIG\_EN****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

vendor specific interrupt signal enable register

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_INT\_SIG\_EN**

Bits	Name	Description
21	ERR_INT_SIGNAL_EN_START_BIT_ERROR	START_BIT_ERROR signal enable
20	ERR_INT_SIGNAL_EN_FIFO_FULL_WR	FIFO_FULL_WR signal enable
19	ERR_INT_SIGNAL_EN_FIFO_EMPTY_RD	IFO_EMPTY_RD signal enable
18	ERR_INT_SIGNAL_EN_AUTO_CMD19_TIMEOUT	AUTO_CMD19_TIMEOUT signal enable
17	ERR_INT_SIGNAL_EN_BOOT_TIMEOUT	BOOT_TIMEOUT signal enable
16	ERR_INT_SIGNAL_EN_BOOT_ACK_ERR	BOOT_ACK_ERR signal enable
0	NORMAL_INT_SIGNAL_EN_BOOT_ACK_REC	BOOT_ACK_REC signal enable

**0x0C08430C PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Handling the boot operation.

### **PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT**

Bits	Name	Description
3	EARLY_ASSERT_CMD_LINE	If set to '1' then the CMD line is asserted by HW when all the data was received on the SD bus
2	BOOT_ACK_EN	If set to '1' then Host waits for acknowledge pattern after initiating the boot operation.
1	BOOT_EN	When this bit is asserted, the boot operation is initiated in both of the modes.
0	BOOT_MODE	If set to '1' then CMD line is low during the boot operation. If boot_mode = '0', then CMD0 with the argument 0xFFFFFFFFFA is sent.

### **0x0C084310 PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT\_ACK\_TIMER**

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

Timer for boot operation - the time until receiving the acknowledge pattern.

### **PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT\_ACK\_TIMER**

Bits	Name	Description
31:0	BOOT_ACK_TIMER	Timer for counting the cycles from initiating the boot operation until the acknowledge pattern is accepted.

### **0x0C084314 PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT\_DATA\_TIMER**

**Type:** RW

**Clock:** same rate as HCLK

**Reset State:** 0x00000000

boot data timer register.

### **PERIPH\_SS\_SDC2\_SDCC\_SDCC\_BOOT\_DATA\_TIMER**

Bits	Name	Description
31:0	BOOT_DATA_TIMER	Number of cycles from initiating the boot operation until the first data is received

**0x0C084320 PERIPH\_SS\_SDC2\_SDCC\_SDCC\_GENERICS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00000000

Generics list of each SDCC instance.

**PERIPH\_SS\_SDC2\_SDCC\_SDCC\_GENERICS**

Bits	Name	Description
26	USE_SCM	current SDCC instance includes SCM
25	USE_XPU	current SDCC instance includes XPU
24	USE_VMIDMT	current SDCC instance includes VMIDMT
23	USE_CMD_QUE	current SDCC instance supports CMDQ (eMMC5.1)
22	HS400_SUPPORT	current SDCC instance supports HS400 timing mode
21	USE_ICE	current SDCC instance includes ICE
20:17	SD_DATA_WIDTH	Number of DAT lines Value 4 - 4 DAT lines (SD, SDIO) Value 8 - 8 DAT lines (MMC, eMMC)
16:4	RAM_SIZE	Size of RAM size: Optional Values: 512, 1024, 2048 or 4096 bytes.
3:1	NUM_OF_DEV	Number of eSD or eSDIO devices supported on shared SD bus. Can be set from 1 to 4.
0	USE_DLL_SDC4	Enables the instantiation of cm_dll_sdc4. Value 0 - cm_dll_sdc4 isn't integrated with SDCC5 Value 1 - cm_dll_sdc4 is used with SDCC5.

**0x0C084324 PERIPH\_SS\_SDC2\_SDCC\_FIFO\_STATUS****Type:** R**Clock:** same rate as HCLK**Reset State:** 0x00007800

Status bit of FIFO's fill level. Can be used for long accesses to FIFO using the FIFO\_ALT register

**PERIPH\_SS\_SDC2\_SDCC\_FIFO\_STATUS**

Bits	Name	Description
18	RX_FIFO_512	512 bytes are ready in FIFO during RX transaction
17	RX_FIFO_256	256 bytes are ready in FIFO during RX transaction
16	RX_FIFO_128	128 bytes are ready in FIFO during RX transaction

**PERIPH\_SS\_SDC2\_SDCC\_FIFO\_STATUS (cont.)**

Bits	Name	Description
15	RX_FIFO_64	64 bytes are ready in FIFO during RX transaction
14	TX_FIFO_512	available space in FIFO for 512 bytes during TX transaction
13	TX_FIFO_256	available space in FIFO for 256 bytes during TX transaction
12	TX_FIFO_128	available space in FIFO for 128 bytes during TX transaction
11	TX_FIFO_64	available space in FIFO for 64 bytes during TX transaction
10:0	FIFO_FILL_LEVEL	reflects the actual fill level of FIFO DPRAM during TX or RX transaction. Supported values are 0, 4, 8, ..., 1020, 1024

**0x0C084400+ PERIPH\_SS\_SDC2\_SDCC\_HC\_FIFO\_ALTn, n=[0..255]****0x4\*n****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

Alternative register to MCI\_FIFO (0x80-0xBC). The SW can use this register range (0x400-0x7FC) for accessing all the RAM range (1024 bytes) in FIFO mode. The advantage is that any byte in the RAM can be accessed without repeating the same address

**PERIPH\_SS\_SDC2\_SDCC\_HC\_FIFO\_ALTn**

Bits	Name	Description
31:0	DATA	WORD is written or received to/from the RAM in FIFO mode

**0x0C084800 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL****Type:** RW**Clock:** same rate as HCLK**Reset State:** 0x00000000

ICE Vendor Specific Controls

**PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL**

Bits	Name	Description
0	ICE_SW_RST_EN	Enable the ICE SW Reset. If this bit is '0', ICE Core could not be reset through SW. If this bit is '1', ICE Core will receive HCLK SW reset as its sync reset and ICE sync_rst_ack will be part of HCLK SW out-of-reset logic

# 9 BAM low-speed peripheral registers

## 0x0C144000 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CTRL

Type: RW

Clock: bam\_clk

Reset State: 0x00020000

BAM Control register allows global controls for the BAM.

### PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CTRL

Bits	Name	Description
20	BAM_MESS_ONLY_CANCEL_WB	<p>When active (1) - BAM will DROP former descriptor write-backing and Interrupt firing when former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT and size 0 arrived (messaging only).</p> <p>When legacy mode is used (0) - This said a case then the former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT and size 0 arrived (messaging only). In this case the FIFO-pointer has been already advanced before the ack_on_sucess arrives and therefore we do write-back to former descriptor and fire interrupt.</p> <p>1'b1 - Drop WB and Interrupt 1'b0 - legacy.</p>
19	CACHE_MISS_ERR_RESP_EN	<p>When set to '1', upon local ahb access results with cache miss, the bam_ndp will not stall the bus, and finish the access with error response.</p> <p>This bit is relevant for BAM_NDP only.</p>
18:17	LOCAL_CLK_GATING	<p>These Bits enables power saving by using a local clock gating cell.</p> <p>Bit 17:</p> <p>1'b1 - CGC is on, so that the clock is controlled by the HW - this is the reset value.</p> <p>1'b0 - CGC is off so that the clock is free running</p> <p>Bit 18:</p> <p>Reserved.</p>

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CTRL (cont.)**

Bits	Name	Description
16	IBC_DISABLE	<p>This Bit enables power saving by disabling the inactivity clock timer/counter. This is useful when:</p> <ol style="list-style-type: none"> <li>1) The BAM is Enabled but idling for long periods.</li> <li>2) The BAM is Enabled and running but the inactivity timers/counters are not required by the SW for operating the BAM.</li> </ol> <p>When the BAM is Disabled, it automatically shuts down those timers to save power.</p> <p>1'b1 - Enable Power Saving Mode 1'b0 - Disable Power Saving Mode</p>
15	BAM_CACHED_DESC_STORE	<p>This Bit enables storage of a cached Descriptor into RAM when doing Context Switch and loading it back when returning to this pipe. This is relevant for System Modes Only.</p> <p>BAM to BAM Consumer mode doesn't have this option.</p> <p>Enabling this makes the Context Switch process slightly longer but reduces AHB Descriptor Fetch time later on. It also reduces the number of descriptor fetches done by the BAM if BAM is treating more than 2 pipes.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p> <p>Available in BAM only</p>
14:13	BAM_DESC_CACHE_SEL	<p>This Selector chooses which Descriptor Caching mode will be used. This is relevant for System Producer or Consumer modes only.</p> <p>2'b00 - Cache when current descriptor has less than 64 bytes left. 2'b01 - Cache when current descriptor has less than 128 bytes left. 2'b10 - Immediate Auto cache - cache will happen whenever there are no descriptors cached.</p> <p>This might cause unnecessary descriptors reads when switching between more than 2 pipes.</p> <p>It is the most effective when each pipe processes more bytes than there are in an average descriptor (small descriptors, big block size), or when working with 2 or less pipes.</p> <p>2'b11 - Cache when descriptor is needed only. This results in 1 descriptor present in cache at most.</p> <p>Available in BAM only</p>
11:5	RESERVED	RESERVED
4	BAM_EN_ACCUM	<p>When Enabled, BAM will accumulate data written by the peripheral in Direct Mode into INCR8 Bursts.</p> <p>This is an optimization feature for Producer Direct Mode cases.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p> <p>Available in BAM only</p>

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CTRL (cont.)**

Bits	Name	Description
1	BAM_EN	<p>After reset the BAM wakes up in Disabled Mode. While Disabled, BAM operates in Legacy mode (all the transfers from the peripheral go around the BAM as if it doesn't exist) Software Enables this bit to allow BAM operation. This bit doesn't affect the PIPE enable bit in BAM_P_CTRL register. The only possibility to Disable the BAM after it has been enabled is by Hardware or Software reset.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p>
0	BAM_SW_RST	<p>This will reset the BAM &amp; all Pipes. Software may use this to reset the BAM. As long as the bit remains high the BAM remains in reset state.</p> <p>Software should clear this bit</p> <p>1'b1 - Reset state 1'b0 - Normal state</p>

**0x0C144008 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds a threshold value for the counter summing the Size of the Descriptors Provided. This value is Global for all pipes but it is relevant for System Producer BAM mode and Both Consumer modes, where Descriptors summing takes place in order to provide the information to the peripheral via pipe bytes free (producer) or pipe bytes available interfaces.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD**

Bits	Name	Description
15:0	CNT_TRSHLD	<p>Threshold value. The maximum allowed value is 32kByte. The minimum allowed value is 1 (value 0 is not allowed). Available in BAM only</p>

**0x0C144014 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_STTS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

Interrupt status register describing the reason for the ongoing interrupt. This is an informative only register and therefore read only.

Clearing the interrupt bits is done by writing to BAM\_IRQ\_CLR register.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_STTS**

Bits	Name	Description
4	BAM_TIMER_IRQ	Inactivity timer has reached its threshold value Interrupt
3	BAM_EMPTY_IRQ	It should not be used in normal SW flows. The interrupt indicates the BAMs buffer is empty. This bit can become high only when BAM_DATA_ERASE
2	BAM_ERROR_IRQ	This Indicates a Fatal Error has happened in the BAM. Currently this might happen due to an UnSuccessful Pipe Reset operation.
1	BAM_HRESP_ERR_IRQ	This interrupt Indicates that an Erroneous HResponse has been received by the AHB Master. Additional Information about the error may be read at BAM_AHB_MASTER_ERR_* Registers. Clearing this interrupt also clears the BAM_AHB_MASTER_ERR_* Registers. Available in BAM only

#### **0x0C144018 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_CLR**

**Type:** W

**Clock:** bam\_clk

**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_CLR**

Bits	Name	Description
4	BAM_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
3	BAM_EMPTY_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM-Lite only
2	BAM_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
1	BAM_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM only

**0x0C14401C PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_EN****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Enable bits for the Interrupts in the BAM Interrupt Status register.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_EN**

Bits	Name	Description
4	BAM_TIMER_EN	Enables Inactivity timer interrupt by writing this bit 1'b1 - Enable 1'b0 - Disable
3	BAM_EMPTY_EN	1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
2	BAM_ERROR_EN	1'b1 - Enable 1'b0 - Disable
1	BAM_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable Available in BAM only

**0x0C144040 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This register counts the idle time of the BAM (all the pipes are in idle state).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER**

Bits	Name	Description
15:0	TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0x0C144044 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER\_CTRL****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The register can be written and read only when INACTIVITY\_TIMERS\_SUPPORTED generic equals to 1.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER\_CTRL**

Bits	Name	Description
31	TIMER_RST	To begin using the inactivity timer the SW should write the threshold and write 1 to this bit. writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the TIMER_THRESOLD value 1'b1 - Active 1'b0 - Reset
30	TIMER_RUN	Will be used along with TIMER_MODE Not implemented at this time. Set to zero(0)
29	TIMER_MODE	Not implemented at this time. Set to zero(0)
15:0	TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

#### **0x0C144084 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CNFG\_BITS\_2**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x0000000F

This Register holds the BAM configuration bits. It is highly recommended to follow the directions for each bit and set it accordingly.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CNFG\_BITS\_2**

Bits	Name	Description
3	SUP_GRP_LOCKER_RST_SUPPORT	
2	ACTIVE_PIPE_RST_SUPPORT	Supporting pipe-reset when the pipe is not necessarily quiet. Disable/Enable in BAM-NDP/BAM-Lite only.
1	NO_SW_OFFSET_REVERT_BACK	When doing Write-back do not revert back the BAM_P_SW_OFSTS pointer.
0	CNFG_NO_ACCEPT_AT_FIFO_FULL	Hold the back pressure of accept_ack_on_success in a case of fifo is going to be full. The back-pressure will be held until after the indication of descriptor-fifo-full (pipe_bytes_ctrl[0]) is asserted (CR701084). Disable/Enable in BAM-NDP only.

**0x0C145000 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_REVISION****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds a hard coded revision number of the BAM RTL. This corresponds to the BAM target number in the IMS. Also additional BAM settings are reflected here.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_REVISION**

Bits	Name	Description
31:24	INACTIV_TMR_BASE	This indicates the width of the inactivity timers base counter.
23	CMD_DESC_EN	This indicates BAM has Command Descriptor feature enabled in HW.
22:21	DESC_CACHE_DEPTH	This field indicates the Per-Pipe-Descriptor-Cache. could be 1-4: 2'b00 - Descriptor Cache Depth of 1. 2'b01 - Descriptor Cache Depth of 2. 2'b10 - Descriptor Cache Depth of 3. 2'b11 - Descriptor Cache Depth of 4. Valid only for BAM_NDP.
20	NUM_INACTIV_TMRS	This bit is checked only if INACTIV_TMRS_EXST = 1. This bit indicates how many inactivity timers are implemented according to value of INACTIVITY_TIMERS_SUPPORTED generic: 1'b1 - One timer for global BAM activity (INACTIVITY_TIMERS_SUPPORTED = 1) 1'b0 - MAX_PIPES timers for each pipe activity (INACTIVITY_TIMERS_SUPPORTED = 2)
19	INACTIV_TMRS_EXST	This indicates BAM has inactivity timers. 1'b1 - Timers available (INACTIVITY_TIMERS_SUPPORTED = 1 or 2) 1'b0 - No timers available (INACTIVITY_TIMERS_SUPPORTED = 0)
18	HIGH_FREQUENCY_BAM	This indicates BAM has Double Sampling on the Ack On Success interface. 1'b1 - Ack On Success double sampled 1'b0 - No double sampling of Ack On Success
17	BAM_HAS_NO_BYPASS	This field indicates BAM has no bypass on the AHB Data bus, from the Peripheral AHB Master to the fabric. When Bypass is available, it is the default routing when BAM is disabled. 1'b1 - No Bypass 1'b0 - Bypass Exists

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_REVISION (cont.)**

Bits	Name	Description
16	SECURED	This field indicates BAM/BAM-Lite has security support (APU inside) 1'b1 - Secured 1'b0 - Not Secured
15	USE_VMIDMT	This field indicates BAM has VMIDMT supported in HW.
14	AXI_ACTIVE	This field indicates BAM_NDP uses internal AXI bridge on the master port.
13:12	CE_BUFFER_SIZE	This field indicates the size (in Data words) of the buffer which stores the command elements. Each command element includes 4 words. 2'b00 - 4 Words (one command element). 2'b10 - 8 Words (2 command elements). 2'b11 - 16 Words (4 command elements). Valid Only if CMD_DESC_EN = 1'b1.
11:8	NUM_EES	This indicates the Number of Interrupt Lines (Execution Environments) supported by the BAM. When 1 EE is supported, only the BAM_IRQ_SRCS*_EE0 registers exist. When 4 EEs are supported, all BAM_IRQ_SRCS*_EEn registers exist for n=[0..3].

**0x0C145008 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_NUM\_PIPES****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds the hard coded number of pipes in the BAM. Since each BAM may have a different number of pipes instantiated in HW, this register allows the SW to know this parameter. The reset state value doesn't have any meaning in this register since it is a constant.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_NUM\_PIPES**

Bits	Name	Description
31:24	BAM_NON_PIPE_GRP	This field contains the number of BAM Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM
23:16	PERIPH_NON_PIPE_GRP	This field contains the number of Peripheral's Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_NUM\_PIPES (cont.)**

Bits	Name	Description
15:14	BAM_DATA_ADDR_BUS_WI_DTH	This field indicates BAM data bus address width: 2'b00 - 32bit address width 2'b01 - 36bit address width 2'b1x - Reserved for future widening
7:0	BAM_NUM_PIPES	This field contains the number of pipes available in this BAM

**0x0C145024 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_CTRLS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_CTRLS**

Bits	Name	Description
22:18	BAM_ERR_HVMID	HVMID
17	BAM_ERR_DIRECT_MODE	DIRECT MODE
16:12	BAM_ERR_HCID	HCID
11:8	BAM_ERR_HPROT	HPROT
7:5	BAM_ERR_HBURST	HBURST
4:3	BAM_ERR_HSIZE	HSIZE
2	BAM_ERR_HWRITE	HWRITE
1:0	BAM_ERR_HTRANS	HTRANS

**0x0C145028 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR**

Bits	Name	Description
31:0	BAM_ERR_ADDR	HADDR

#### **0x0C14502C PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_DATA**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_DATA**

Bits	Name	Description
31:0	BAM_ERR_DATA	HDATA

#### **0x0C145100 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

Bits	Name	Description
31:0	BAM_ERR_ADDR	32 LSB of HADDR

**0x0C145104 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB**

Bits	Name	Description
3:0	BAM_ERR_ADDR	4 MSB of Address (bits 35 to 32) of HADDR

**0x0C146000 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TRUST\_REG****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any SW reset. Only HW reset applies. When using a secured BAM (APU integrated inside - SECURED field in the BAM\_REVISION register) it is reset by the XPU HW Reset. When using a non-secured BAM, it is reset by the BAM HW Reset.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TRUST\_REG**

Bits	Name	Description
13	LOCK_EE_CTRL	This bit controls if the EE setting defined in the TRUST registers will be taken into account for the pipe lock grouping decoding. 1'b0 - Only P_LOCK_GROUP is checked for pipe locking. 1'b1 - Both BAM_P_EE and P_LOCK_GROUP are checked for pipe locking.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TRUST\_REG (cont.)**

Bits	Name	Description
12:8	BAM_VMid	
7	BAM_RST_BLOCK	When enabled, the BAM Global SW reset will not be available for usage. This is in order to deny Global SW reset requests by the Global security group. 1'b1 - Enable 1'b0 - Disable
2:0	BAM_EE	This Field Indicates the EE # (0-7) to which the BAM Interrupt belongs to. The pipe interrupts will fire on the EE # set at BAM_P_TRUST_REGn registers. The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register. 3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7

**0x0C147000+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn, n=[0..3]**  
**4096\*n**

Type: R

Clock: bam\_clk

Reset State: 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking this register, SW reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register has an alias - BAM\_IRQ\_SRCS register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0x0C147004+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK\_EEn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register has an alias - BAM\_IRQ\_SRCS\_MSK register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK\_EEn**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

**0x0C147008+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn, n=[0..3]  
4096\*n**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

The register shows the interrupts sources like (BAM\_IRQ\_SRCS) although the interrupts from the BAM and the pipes are not masked on the lower level. This register has an alias - BAM\_IRQ\_SRCS\_UNMASKED register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

**0x0C14700C+PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn, n=[0..3]  
4096\*n**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

The register shows the pipes attributed to a specific EE, and an indication if BAM is enabled.

The reset value written above is true only if EE>0. For BAM\_PIPE\_ATTR\_EE0 the default value is: Bits [MAX\_PIPES-1:0] = 1 , Bits [31: MAX\_PIPES] = 0 .

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn**

Bits	Name	Description
31	BAM_ENABLED	1'b1 - BAM is Enabled. 1'b0 - BAM Disabled.
30:0	P_ATTR	If bit 'i' == 1'b1 - Pipe 'i' is attributed to this EE.

**0x0C147010 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking out this register, SW reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register points to the physical BAM\_IRQ\_SRCS\_EE0 register.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0x0C147014 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register points to the physical BAM\_IRQ\_SRCS\_MSK\_EE0 register.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

#### **0x0C147018 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register points to the physical BAM\_IRQ\_SRCS\_UNMASKED\_EE0 register.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

#### **BAM PIPE configuration and interrupts registers**

#### **0x0C146020+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TRUST\_REGn, n=[0..11]**

**0x4\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any SW reset. Only HW reset applies. When using a secured BAM (APU integrated inside) it is reset by the XPU HW Reset. When using a non-secured BAM, it is reset by the BAM HW Reset.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TRUST\_REGn**

Bits	Name	Description
12:8	BAM_P_VMID	
7:3	BAM_P_SUP_GROUP	<p>Super-Group is the upper division of the pipes compared to the pipe-group division.</p> <p>When any pipe locks or unlocks the BAM, all the pipes which are in a different super-group will NOT be affected. Only the pipes within the same super-group of the locker are affected.</p> <p>Usually pipes which belongs to different peripherals which share the same BAM, would be in a different super-groups</p>
2:0	BAM_P_EE	<p>This Field Indicates the EE # (0-7) to which the Pipe Interrupt belongs to. The BAM interrupt will fire to the EE # set at BAM_P_TRUST_REGn registers.</p> <p>The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register.</p> <p>3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7</p>

**0x0C157000+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CTRLn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Control register provides various controls for the pipe.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CTRLn**

Bits	Name	Description
20:16	P_LOCK_GROUP	<p>Pipe's lock group.</p> <p>Upon a lock request on this pipe, all pipes related to different pipe group and different EE will be locked.</p>

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CTRLn (cont.)**

Bits	Name	Description
11	P_WRITE_NWD	BAM-Lite feature. Applicable to BAM2BAM producer pipes only. When this bit is set for producer B2B pipe, NWD bit (bit number 28 in the second word of descriptor) will be written with EOT into the generated descriptor for the consumer usage. This bit is not applicable in pipes belong to a Peripheral which uses messaging_only (e.g. USB2).
10:9	P_PREFETCH_LIMIT	Bam Lite feature Selects the number of bytes to be prefetched by the Bam Lite, once a peripheral requests to read data from a pipe. This feature is relevant for Consumer modes only. 2'b00 - 32 bytes at most (INCR8) 2'b01 - 16 bytes at most (INCR4) 2'b10 - 4 bytes at most (SINGLE) Available in BAM-Lite only
8:7	P_AUTO_EOB_SEL	Bam Lite feature. Selects the number of bytes to be processed before creating End Of Block indication internally by the BAM. Descriptors generated by this feature (bam to bam producer mode) will always be of the chosen size, unless an End Of Transfer, causes a smaller descriptor to be created. 2'b00 - 512 Bytes 2'b01 - 256 Bytes 2'b10 - 128 Bytes 2'b11 - 64 Bytes Available in BAM-Lite only
6	P_AUTO_EOB	BAM Lite feature. When Enabled, the BAM Lite will ignore any incoming End Of Block indications from the peripheral (if any). Instead, the BAM Lite will generate it's own End Of Block indications internally, every P_AUTO_EOB_SEL bytes. This feature is relevant for Producer Bam 2 Bam modes only. It is a must for peripherals which are not able to produce End Of Block indications. 1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
5	P_SYS_MODE	This bit sets the pipe to work in System Mode, where the Software is the second side of the pipe. 1'b1 - System mode. 1'b0 - BAM to BAM mode.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CTRLn (cont.)**

Bits	Name	Description
4	P_SYS_STRM	Streaming Enable. Relevant to System Producer BAM mode only. When enabled, BAM will not close descriptors with End Of Transfer notifications received from peripheral, but continue writing the data to the same descriptor. 1'b1 - Enable Streaming Mode 1'b0 - None-Streaming Mode. This field is NOT supported in BAM-NDP (i.e. no streaming mode in NDP).
3	P_DIRECTION	This bit deNOTE:s pipe direction. 1'b1 - The pipe can be written (producer mode) 1'b0 - The pipe can be read (consumer mode)
1	P_EN	1'b1 - Enable Pipe 1'b0 - Disable Pipe

**0xC157004+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RSTn, n=[0..11]  
4096\*n**

**Type:** W

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Reset register provides Pipe Reset ability. Software needs to write 1 to this register to make the Pipe Reset. Writing 0 to this register will finish the Pipe Reset. Writing zero is needed prior to Peripheral reconfiguration for the same pipe.

NOTE: Software can invoke pipe reset only after it make sure that the pipe is 'quiet'.

('quiet' - No data transaction or command-descriptor is active at the time.)

BAM\_P\_RSTn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RSTn**

Bits	Name	Description
0	P_SW_RST	This resets the pipe and it's registers, (Both Flip-Flops and RAM). Software can invoke pipe reset only after it make sure that the pipe is 'quiet'. 1'b1 - Reset 1'b0 - Do Nothing

**0xC157008+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_HALTn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000008

Pipe Halt register Enables/Disables the Halt Sequence. This is a self-modifying register.

It also supports pipe-empty indication and force-descriptor-fifo-full.

### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_HALTN**

Bits	Name	Type	Description
4	P_FORCE_DESC_FIFO_FULL	RW	When SW asserts this bit, descriptor-fifo-full status will be shown to the peripheral on pipe_bytes_avail_ctrl[0]. This is to make the peripheral to think that there is no room for more descriptors. Available only in BAM-NDP and only for Producer BAM-to-BAM pipes.
3	P_PIPE_EMPTY	R	This bit indicates that the Descriptor-FIFO is now empty. It is a read-only bit. Available in BAM-NDP only. Else it is tied high.
2	P_LAST_DESC_ZLT	R	This bit indicates that the last created descriptor is with zero-length size. It is a read only bit and effective only for Producer BAM2BAM mode. Available in BAM-NDP only. Else it is tied high.
1	P_PROD_HALTED	RW	This bit is used to inform the Consumer about the Producer being in Halt mode now. This is a part of the Halt procedure. 1'b1 - Sets this bit to 1 1'b0 - Does Nothing This bit will be cleared by the HW. Not Available in BAM-Lite
0	P_HALT	RW	When Enabled, the Pipe will enter Halt Mode. 1'b1 - Enable Halt 1'b0 - Disable Halt For B2B Producer pipes, this bit is self modifying. It doesn't need the SW to clear it. Not Available in BAM-Lite

### **0x0C157010+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn, n=[0..11] 4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Interrupt status register describing the reason for the ongoing interrupt. This is an informative only register and therefore read only. It also has informative bits which are not interrupts.

Clearing the interrupt bits is done by writing to P\_IRQ\_CLR register.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn**

Bits	Name	Description
7	P_HRESP_ERR_IRQ	This interrupt Indicates that an Erroneous HResponse has been received by the AHB Master. Additional Information about the error may be read at BAM_AHB_MASTER_ERR_* Registers. Clearing this interrupt also clears the BAM_AHB_MASTER_ERR_* Registers. Available in BAM only
6	P_PIPE_RST_ERROR_IRQ	Unsuccessful Pipe Reset operation. Pipe reset timer expired.
5	P_TRNSFR_END_IRQ	For producer this interrupt happens whenever the last AHB burst of a transfer happens For Consumer when ack_on_success closes a transfer
4	P_ERR_IRQ	Pipe Error interrupt indicates that an error has happened. Error reason is not yet defined. Interrupt not in use.
3	P_OUT_OF_DESC_IRQ	Out of descriptors interrupt has several meanings depending on the operation mode: Producer - no free space in the descriptors fifo for adding a descriptor Consumer - no descriptors in the descriptors fifo for processing IO Vectors Producer - No descriptors to read (and therefore no space for data to write to the pipe) IO Vectors Consumer - No descriptors to read (and therefore no data to read from pipe)
2	P_WAKE_IRQ	Wake peripheral interrupt signals the driver to wake up the peripheral (USB for example) for transmission via the current pipe. This interrupt is issued whenever an Event comes to the specific pipe.
1	P_TIMER_IRQ	Inactivity timer has reached its threshold value Interrupt
0	P_PRCSD_DESC_IRQ	This Interrupt rises when BAM finishes processing an IO Vector which has INT bit selected

**0x0C157014+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn, n=[0..11]  
4096\*n**

**Type:** W

**Clock:** bam\_clk

**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

**BAM\_P\_IRQ\_CLRn, n=[0..30]**

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn**

Bits	Name	Description
7	P_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
6	P_PIPE_RST_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
5	P_TRNSFR_END_CLR	1'b1 - Clear 1'b0 - Unchanged
4	P_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
3	P_OUT_OF_DESC_CLR	1'b1 - Clear 1'b0 - Unchanged
2	P_WAKE_CLR	Clear Wake peripheral interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
1	P_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
0	P_PRCSD_DESC_CLR	Clear Descriptor Processed interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged

**0x0C157018+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Enable bits for the Interrupts in the Pipe Interrupt Status register.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn**

Bits	Name	Description
7	P_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable
6	P_PIPE_RST_ERROR_EN	1'b1 - Enable 1'b0 - Disable
5	P_TRNSFR_END_EN	1'b1 - Enable 1'b0 - Disable
4	P_ERR_EN	1'b1 - Enable 1'b0 - Disable

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn (cont.)**

Bits	Name	Description
3	P_OUT_OF_DESC_EN	1'b1 - Enable 1'b0 - Disable
2	P_WAKE_EN	Enable Wake peripheral interrupt 1'b1 - Enable 1'b0 - Disable
1	P_TIMER_EN	Enable Inactivity timer Interrupt 1'b1 - Enable 1'b0 - Disable
0	P_PRCSD_DESC_EN	Enable Descriptor Processed Interrupt 1'b1 - Enable 1'b0 - Disable

**0xC15701C+PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMERn, n=[0..11]**  
**4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register counts the idle time of the pipe.

**BAM\_P\_TIMERn, n=[0..30]**

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMERn**

Bits	Name	Description
15:0	P_TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0xC157020+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRLn, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The resolution of the pipe inactivity timers are in units set by a common counter. This common counter is controlled by a PARAMETER value of the BAM, and uses a separate clock, the inactivity\_timers\_clk. This clock can be slower than the bam\_clk. The intent of the design is to use the sleep\_clk, which is an always on 32KHz clock. This allows the bam\_clk to be turned-off, while the base counter keeps counting. The width of this counter, as well of the rollover value are set using BAM PARAMETER. These values, taking the inactivity\_timers\_clk frequency define the

pulse frequency that the base counter gives, and the threshold value can be calculated and set to interrupt after a specific time, regardless of the bam\_clk frequency, and independent of clock power save features of the bam\_clk. The timers continue counting even if the clock is turned off due to BAM dynamic hardware clock request voting.

The timer configuration is based on the inactivity\_timers\_clk period and the INACTIVITY\_TIMER\_WIDTH parameter constant. These parameters should be taken into account when setting this register. For example: for inactivity\_timer\_clk period of 1us and parameter is 3 and P\_TIMER\_THRESHOLD is 10 will indicate the  $2^{3*1\text{us}} * 10$  which is 80us of inactivity in a pipe before sending an interrupt.

The general formula is:  $2^{\text{INACTIVITY\_TIMER\_WIDTH}} * \text{clock\_period} * \text{P\_TIMER\_TRSHLD}$ .

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRLn**

Bits	Name	Description
31	P_TIMER_RST	To begin using the inactivity timer the SW should write the threshold and write 1 to this bit. Writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the P_TIMER_THRESHOLD value 1'b1 - Active 1'b0 - Reset
30	P_TIMER_RUN	Will be used along with P_TIMER_MODE Not implemented at this time. Set to zero(0)
29	P_TIMER_MODE	Not implemented at this time. Set to zero(0)
15:0	P_TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

#### **0x0C157024+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn, n=[0..11] 4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register reflects the current status of the Producer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn**

Bits	Name	Description
24	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_producer. relevant for system mode only. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
20	BAM_P_TOGGLE	Pipe Bytes Free Toggle value. The toggle polarity of the publication (each publication changes its polarity).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn (cont.)**

Bits	Name	Description
19:16	BAM_P_CTRL	Pipe Bytes Free Ctrl value. The possible values are only: 0000 or 0001 (means desc-fifo-full).
15:0	BAM_P_BYTES_FREE	Pipe Bytes Free value. The accumulated bytes_free which are available for peripheral to use

**0x0C157028+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn, n=[0..11]  
4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register reflects the current status of the Consumer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn**

Bits	Name	Description
30	BAM_P_ACCEPT_ACK_ON_SUCCESS_TOGGLE	accept_ack_on_sucess toggle. This bit is relevant for BAM_NDP only.
29:28	BAM_P_ACK_ON_SUCCES_S_CTRL	ack_on_sucess control. This bit is relevant for BAM_NDP only.
27	BAM_P_ACK_ON_SUCCES_TOGGLE	ack_on_sucess toggle. This bit is relevant for BAM/BAM_NDP only.
26	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_consumer. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
25	BAM_P_NWD_TOGGLE	notify_when_done toggle
24	BAM_P_NWD_TOGGLE_R	notify_when_done toggle sampled
23	BAM_P_WAIT_4_ACK	waiting_for_ack_bytes_avail_r. BAM is waiting for Peripheral to acknowledge the former publish.
22	BAM_P_ACK_TOGGLE	ack_bytes_avail_toggle. The Peripheral's acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
21	BAM_P_ACK_TOGGLE_R	ack_bytes_avail_toggle_r. The Peripheral's sampled acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
20	BAM_P_TOGGLE	Pipe Bytes Avail Toggle value. The toggle polarity of the publication (each publication changes its polarity).

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn (cont.)**

Bits	Name	Description
19:16	BAM_P_CTRL	Pipe Bytes Avail Ctrl value. The possible values of BAM_P_CTRL field are: 0100 - Means immediate command (could be only in bam-ndp). 0001 - Means End-Of-Transaction (EOT). 0011 - means End-Of-Transaction with Notify-When-Done (NWD). 0000 - Neither Immediate-command nor EOT nor NWD.
15:0	BAM_P_BYTES_AVAIL	Pipe Bytes AVAIL value. The size of the current publish.

**0x0C157800+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register deNOTE:s the pointer Offset of the first Not Acknowledged Descriptor. Meaning all Descriptors prior to this one have been processed by the BAM. This register is only used by the Software. After receiving an interrupt, software reads this register in order to know what descriptors has been processed.

When the BAM works as Producer, Software will usually read this value after Descriptor Processed Interrupt or after the End of Transfer Interrupt.

When the BAM works as Consumer, Software will usually read this value after Descriptor Processed Interrupt.

NOTE: This is non relevant in BAM to BAM modes.

NOTE: Although being Writable, Software should never write to this register.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SW\_OFSTS<sub>n</sub>**

Bits	Name	Description
31:16	SW_OFST_IN_DESC	Offset inside the Descriptor. This value is used by the Software only when a BAM is working as System Producer. Then the SW can read this value to know how many Bytes were written to the current descriptor if it was not closed down. This is useful for Streaming mode. This Field is NOT available in BAM-NDP
15:0	SW_DESC_OFST	Descriptor FIFO offset.

**0x0C15782C+PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDRn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe.

**0x0C157930+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR LSBn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR LSBn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	32 LSB of Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

**0x0C157934+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_EVNT_DEST_ADDR	4 MSB of Address (bits 35 to 32) of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

**0x0C157818+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This Register is where Read/Write events are sent to. If current BAM works as Producer, it will receive Read events into this register from the Consumer. If the Consumer is another BAM, it will write read events to this register. If the Consumer is the Software, it will prepare descriptors in the descriptor FIFO and update the DESC\_FIFO\_PEER\_OFST value in this register.

If current BAM works as Consumer, opposite behavior is assumed.

BAM\_P\_EVNT\_REGn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn**

Bits	Name	Description
31:16	P_BYTES_CONSUMED	This field is used only in the B2B mode. 15 LSB bits of this field indicate the number of bytes consumed by the Consumer. Consumer sends this value to the Producer as a part of a read event, for the Producer to know how many bytes were consumed. The MSB is virtual-event written by the SW in pipe-halt procedure. When working in System mode, this value should be set to zero. BAM to BAM Producer writes zero in this value.
15:0	P_DESC_FIFO_PEER_OFST	This field indicates the Descriptor Fifo Offset Pointer of the Peer BAM (The BAM or the Software which creates descriptors). For example when current BAM works as consumer for the current pipe, Producer BAM (or the software) updates this value to let the current pipe know about the new descriptors added in the Descriptor Fifo.

**0x0C15781C+PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Address of the Descriptor Fifo beginning.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	Address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0x0C157910+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the LSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	32 LSB of address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0x0C157914+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the MSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DESC_FIFO_ADDR	4 MSB of address (bits 35 to 32) of Address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0x0C157820+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

The least significant field is the Descriptor Fifo Size.

The most significant field is the Data Fifo Size.

NOTE: This Descriptor Fifo Size is used by all modes. The Data Fifo Size is non relevant in System Modes.

BAM\_P\_FIFO\_SIZESn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn**

Bits	Name	Description
31:16	P_DATA_FIFO_SIZE	Size of the Data Address Space. ALWAYS write 16'b0 to this field of P_DATA_FIFO_SIZE unless you work in BAM2BAM producer mode (also known as peer mode). NOTE: Data fifo size should not be bigger than 32KB.
15:0	P_DESC_FIFO_SIZE	Size of the Descriptors Fifo. Must be 8 byte aligned. Each descriptor is 8 bytes. Size of the descriptor fifo must contain an integer number of Descriptors. Size of descriptor fifo must be 16 bytes and above.

**0xC157824+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	Data Address Space beginning.

**0xC157920+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	32 LSB of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0x0C157924+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DATA_FIFO_ADDR	4 MSB of Address (bits 35 to 32) of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0x0C157828+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register configures the threshold value for Read/Write event generation by the BAM towards another BAM. When aBAM pushes/pops this number of bytes to/from the pipe an event to the other BAM will be generated. A counter summing up the bytes pushed/popped is reset when event is issued, thus restarting count for the threshold.

Events are also issued after End Of Transfer received from the peripheral (Producer case), non regarding whether the pushed bytes count has reached the threshold or not.

This register is non relevant in System Modes.

BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn**

Bits	Name	Description
15:0	P_TRSHLD	Threshold value. The maximum allowed value is 32kByte.

**BAM PIPE internal state registers**

**0x0C157804+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n**

Bits	Name	Description
31:16	AU_PSM_ACCUMED	PSM: The acknowledged (EOB/T) number of bytes since last Write Event.  AU: The accumulation of the descriptor sizes for the present ack_on_success event. This is never bigger than the ACKED field. Once the ACCUMED values are bigger than Event Threshold, an Event will be issued, the ACKED value will decrease by the ACCUMED size and this value will be zeroed.  This also serves as the Bytes Consumed value in the Event.
15:0	AU_ACKED	AU: The left over of the number of bytes that were received via the ack_on_success, that did not cause a descriptor closure. This value is down counting upon issuing Read Event.  PSM:

**0x0C157808+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n**

Bits	Name	Description
31	PSM_DESC_VALID	The Descriptor Valid bit provided by the Writeback state machine to Pipe state machine.
30	PSM_DESC_IRQ	The Descriptor Interrupt bit provided by the Writeback state machine to pipe state machine.
29	PSM_DESC_IRQ_DONE	The Descriptor Interrupt needed selection bit.
28:25	PSM_GENERAL_BITS	General Context Switch Bits
24:22	PSM_CONS_STATE	State of the Pipe Consumer state machine.
21:19	PSM_PROD_SYS_STATE	State of the Pipe Producer System state machine.
18:16	PSM_PROD_B2B_STATE	State of the Pipe Producer BAM to BAM state machine.
15:0	PSM_DESC_SIZE	The Descriptor Size provided by the Writeback state machine to pipe state machine.

**0xC15780C+PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..11]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_n**

Bits	Name	Description
31:0	PSM_DESC_ADDR	The Data Address provided in the current descriptor from the Writeback state machine.

**0xC157900+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_LSBn, n=[0..11]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_LSBn**

Bits	Name	Description
31:0	PSM_DESC_ADDR	32 LSB of The Data Address provided in the current descriptor from the Writeback state machine.

#### **0xC157904+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_MSBn, n=[0..11] 4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_MSBn**

Bits	Name	Description
3:0	PSM_DESC_ADDR	4 MSB of address (bits 35 to 32) of the Data Address provided in the current descriptor from the Writeback state machine.

#### **0xC157810+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..11] 4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n**

Bits	Name	Description
31:16	PSM_DESC_OFST	The running Descriptor Fifo Offset of the Writeback state machine
15:0	PSM_SAVED_ACCUMED_SIZE	The Saved Accumulated Size from the Pipe state machine.

**0xC157814+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n**

Bits	Name	Description
31:16	PSM_BLOCK_BYTE_CNT	The Byte count inside the current Block.
15:0	PSM_OFST_IN_DESC	The Offset inside a register. This is used by pipe state machine to compute offset inside a register after retransmission

**0xC157830+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register is used on in BAM-NDP core.

NOTE: Irrelevant for BAM-Lite.

BAM\_P\_DF\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n**

Bits	Name	Description
31:16	WB_ACCUMULATED	Relevant in B2B Producer mode only. This field stores the current running accumulation of the total amount of data that has been written to the data fifo. When it goes bigger than event_threshold a write event occurs.

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n (cont.)**

Bits	Name	Description
15:0	DF_DESC_OFST	Holds the descriptor offset of the Descriptor-Fetcher block which is responsible for fetching descriptors before publishing them to the peripheral.

**0x0C157834+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

This is being stored by Ack Update state machine in consumer mode or by the Writeback state machine in producer mode.

This is being loaded by the Writeback state machine when retransmission happens.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_RETR\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n**

Bits	Name	Description
31:16	RETR_DESC_OFST	Descriptor Fifo Offset for retransmission.
15:0	RETR_OFST_IN_DESC	The Offset inside the Descriptor for retransmission.

**0x0C157838+ PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_SI\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n**

Bits	Name	Description
15:0	SI_DESC_OFST	The Descriptor Fifo Offset pointer stored by the Sideband Inform state machine.

**0x0C16F000 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR1****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_MR1 register is the UART mode register #1. It is used, along with UART\_DM\_MR2, to configure the operational mode of the UART.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR1**

Bits	Name	Description
31:8	AUTO_RFR_LEVEL1	<p>These bits are used, along with field AUTO_RFR_LEVEL0 to program the level in the receive FIFO at which the Ready-For-Receive signal (RFR_N) toggles, if programmed to do so. When RX-FIFO level is greater than the value in fields AUTO_RFR_LEVEL1 and AUTO_RFR_LEVEL0, signal will indicate Not-Ready (RFR_N high). Otherwise signal will indicate Ready RFR_N low.</p> <p>Configured value should be higher/equal than RFWR value, otherwise the watermark will never be reached.</p> <p>Value of this register is in words and can be programmed from 1 to <math>2^{\text{RAM\_ADDR\_WIDTH}}</math>.</p> <p>- Only RAM_ADDR_WIDTH +1:8 bits of this field are generated.</p>
7	RX_RDY_CTL	<p>Setting (1) this bit enables the automatic ready-for-receiving (RFR_N) control for the receiver. RFR_N is turned off, or set (1), when a valid start bit is received and the RX channel FIFO is at the level programmed in bits [31:8,5:0] of this register. When the FIFO level falls back to the programmed level, the RFR_N signal is turned on, or clear (0). When this feature is off, the RFR_N signal can be used by normal signal port bit manipulation.</p>
6	CTS_CTL	<p>When this bit is set(1), the transmitter checks the CTS_N input to determine whether to begin transmission of a new character. If CTS_N is low, the character is sent. Otherwise the transmitter continues marking until CTS_N goes low, then the next character is transmitted. A change on CTS_N during the transmission of a character has no effect on that character.</p> <p>When this bit is clear(0), the CTS_N input for the channel has no effect on the transmitter.</p>
5:0	AUTO_RFR_LEVEL0	Same description of field AUTO_RFR_LEVEL1.

**0x0C16F004 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR2****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR2**

Bits	Name	Description
10	RFR_CTS_LOOPBACK	Enables internal loopback between RFR_N of RX channel and CTS_N of TX channel if set. For this feature to be enabled, data loopback must also be enabled (LOOPBACK bit of this register).
9	RX_ERROR_CHAR_OFF	When this bit is set, characters received with parity or framing errors get discarded and not moved to the RX channel. Otherwise they enter RX channel as any valid character.
8	RX_BREAK_ZERO_CHAR_OFF	When this bit is set, the zero character received at rx_break event get discarded and not moved to the RX channel. Otherwise it enters RX-channel as any valid character.
7	LOOPBACK	When set - data output of transmitter is fed back to the data input of the receiver, i.e. TX-pin is connected to RX-pin. Might be used with flow-control loopback (RFR_CTS_LOOPBACK bit above).
6	ERROR_MODE	This bit controls the operation of the two status register bits for the RX channel (parity or framing error and received break). <ul style="list-style-type: none"> <li>- When clear (0), the UART operates in character mode and the status bits apply only to the last character received.</li> <li>- When set (1), the UART operates in block mode and both bits are sticky, i.e. they go high once an error/break-event is detected and go low only after reset error status command is issued.</li> </ul>
5:4	BITS_PER_CHAR	These bits determine how many data bits are transmitted or received per character, not including the start, stop, and parity bits. <ul style="list-style-type: none"> <li>0x0: ENUM_5_BITS (5 bits)</li> <li>0x1: ENUM_6_BITS (6 bits)</li> <li>0x2: ENUM_7_BITS (7 bits)</li> <li>0x3: ENUM_8_BITS (8 bits)</li> </ul>
3:2	STOP_BIT_LEN	This field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 9/16, 1, 1+ 9/16, and 2 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. This configuration only applies to UART/IrDA interfaces. <ul style="list-style-type: none"> <li>0x0: ENUM_0_563 (0.563 (9/16) bit times)</li> <li>0x1: ENUM_1_000_BIT_TIME (1.000 bit time)</li> <li>0x2: ENUM_1_563 (1.563 (1+9/16) bit times)</li> <li>0x3: ENUM_2_000_BIT_TIMES (2.000 bit times)</li> </ul>

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR2 (cont.)**

Bits	Name	Description
1:0	PARITY_MODE	<p>These bits determine which parity mode is used. The user can select between odd, even, space, or no parity.</p> <p>0x0: NO_PARITY (no parity - last data-bit is followed by a stop-bit)</p> <p>0x1: ODD_PARITY (odd parity - parity-bit ensures number of ones in data-bits+parity-bit is odd)</p> <p>0x2: EVEN_PARITY (even parity - parity-bit ensures number of ones in data-bits+parity-bit is even)</p> <p>0x3: SPACE_PARITY (space parity - parity is always '0')</p>

**0x0C16F008 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR\_SR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CSR or UART\_DM\_SR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR\_SR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CSR_SR_DEPRECATED	

**0x0C16F010 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR\_MISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CR or UART\_DM\_MISR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR\_MISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CR_MISR_DEPRECATED	

**0x0C16F014 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR\_ISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IMR or UART\_DM\_ISR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR\_ISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IMR_ISR_DEPRECATE	

**0x0C16F018 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IPR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0xFFFFFFF9F

The UART\_DM\_IPR register is the UART interrupt programming register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IPR**

Bits	Name	Description
31:7	STALE_TIMEOUT_MSB	The stale character time-out duration field contains a number from 1 to $2^{30} - 1$ . This number determines how many character times must elapse before a stale event is generated. This character time is defined as 10 times the bit rate. It is reset whenever a data bit is received. Never set this fields to 0 while RX channel is operational. Do not re-configure while an RX-transfer is active. NOTE: the discontinuity in the bit assignments.
6	SAMPLE_DATA	Currently not in use. Setting (1) to this bit enables the sample data mode. In this mode start bit is sampled along with the data of the UART character.
4:0	STALE_TIMEOUT_LSB	This bitfield is the LSbits of the STALE_TIMEOUT bitfield.

**0x0C16F01C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_TFWR register is the UART transmit FIFO watermark register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TFWR**

Bits	Name	Description
31:0	TFW	<p>These bits contain a number, between 0 and <math>2^{\text{RAM\_ADDR\_WIDTH}} - 1</math>, that determines the level of the transmit (TX) FIFO at which space_avail_req signal is asserted in BAM mode or TXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of TX words in the FIFO) is less than or equal to the value in TFWR. Notice that the value is in words.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH -1:0 bits are generated.</li> </ul>

**0x0C16F020 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RFWR register is the UART receive FIFO watermark register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RFWR**

Bits	Name	Description
31:0	RFW	<p>These bits contain a number, between 0 and <math>2^{\text{RAM\_ADDR\_WIDTH}} - 1</math>, that determines the level of the receive (RX) FIFO at which data_avail_req signal is asserted in BAM mode or RXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of words in the RX FIFO) is greater than the value in RFWR. Configured value should be lower/equal than AUTO_RFR_LEVEL value, otherwise the watermark will never be reached. Notice that the value is in words.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH -1:0 bits are generated.</li> </ul>

**0x0C16F024 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_HCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Configures Hunt-Character value.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_HCR**

Bits	Name	Description
7:0	DATA	The UART_DM_HCR register is the UART hunt character register, which contains the character for which the receiver looks to assert a hunt character received interrupt.

**0x0C16F034 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMRX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initializes RX transfer and configures maximal length in bytes of the transfer.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMRX**

Bits	Name	Description
24:0	RX_DM_CRCI_CHARS	A write to RX_DM_CRCI_CHARS initializes RX transfer in SW (FIFO) mode, with written value configuring the maximal length in bytes of the transfer. In BAM-mode, only the value held in this register impacts the data-flow and not the write-event. A write-value of 0 is not valid. Read of UART_DM_DMRX register is returns the number of characters that were received since the end of the last RX transfer.

**0x0C16F038 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IRDA or UART\_DM\_RX\_TOTAL\_SNAP.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IRDA_RX_TOTAL_SNAP_DEPRECATED	

**0x0C16F03C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMEN****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_DMEN register defines the DMA used by the core and the data packing mode for TX and RX channels.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMEN**

Bits	Name	Description
5	RX_SC_ENABLE	- Set (1) for work in Single-Character mode for RX channel (every character received is zero-padded into a word). - Clearing this bit requires resetting the receiver.
4	TX_SC_ENABLE	- Set (1) for work in Single-Character mode for TX channel (Only LSB byte of each word get transmitted). - Clearing this bit requires resetting the transmitter.
3	RX_BAM_ENABLE	- Set (1) this bit to enable RX BAM interface (BAM-mode). - Clear (0) this bit to disable RX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the receiver.
2	TX_BAM_ENABLE	- Set (1) this bit to enable TX BAM interface (BAM-mode). - Clear (0) this bit to disable TX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the transmitter.

**0x0C16F040 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_NO\_CHARS\_FOR\_TX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initialization of TX transfer in SW/FIFO mode.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_NO\_CHARS\_FOR\_TX**

Bits	Name	Description
23:0	TX_TOTAL_TRANS_LEN	The total number of characters for transmission. Before writing a new value, the TX FIFO must be empty as indicated by TX_READY interrupt or after a TX-channel reset. It is used by the TX-channel in SW/FIFO mode to calculate how many characters to transmit in the last word. Any additional writes to the TX FIFO beyond TX_TOTAL_TRANS_LEN characters will be discarded. A TX_READY interrupt is triggered after TX_TOTAL_TRANS_LEN number of characters were moved from the TX FIFO to the unpacking buffer (not all may have been transmitted at that point though).

**0x0C16F044 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BADR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

Configures FIFO division between RX-FIFO and TX-FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BADR**

Bits	Name	Description
31:2	RX_BASE_ADDR	<p>RX FIFO base address. Both FIFOs use the same RAM (<math>2^{\text{RAM\_ADDR\_WIDTH}}</math>, 32-bit entries). This register controls the division of the memory between the RX and TX FIFOs. The division must be a multiple of 4 entries for legacy reasons. The size of the TX FIFO equals RX_BASE_ADDR. The size of the RX FIFO is <math>2^{\text{RAM\_ADDR\_WIDTH}} - \text{RX\_BASE\_ADDR}</math>.</p> <p>* The default is RX_BASE_ADDR = <math>2^{(\text{RAM\_ADDR\_WIDTH} - 1)}</math></p> <p>* Only RAM_ADDR_WIDTH -1:2 bits are generated.</p>

**0x0C16F04C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

TX channel status register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TXFS**

Bits	Name	Description
31:14	TX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. - Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	TX_ASYNC_FIFO_STATE	TX asynchronous fifo status - number of characters in the TX asynchronous FIFO.
9:7	TX_BUFFER_STATE	TX unpacking buffer status - number of bytes in the unpacking buffer
6:0	TX_FIFO_STATE_LSB	TX FIFO status - number of words in the TX FIFO

**0x0C16F050 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

RX channel status register.

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RXFS**

Bits	Name	Description
31:14	RX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	RX_ASYNC_FIFO_STATE	RX asynchronous fifo status - number of characters in the RX asynchronous FIFO.
9:7	RX_BUFFER_STATE	RX packing buffer status - number of bytes in the packing buffer
6:0	RX_FIFO_STATE_LSB	The number of entries in the RX FIFO that have at least one valid word. It is incremented each time a word is moved from the packing register to the RX FIFO. It is decremented each time a word is read from the RX FIFO.  NOTE: The Uart does not keep track of non-valid characters in each word.

#### **0x0C16F06C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR\_VAL**

**Type:** R

**Clock:** WR\_CLK

**Reset State:** Undefined

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR\_VAL**

Bits	Name	Description
9:0	VAL	Current MISR state

#### **0x0C16F080 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SIM\_CFG**

**Type:** RW

**Clock:** AHB\_CLK

**Reset State:** 0x00000000

The UART\_DM\_SIM\_CFG register is used to configure the SIM interface for the UART.

\* This register is generated only when SIM\_GLUE\_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SIM\_CFG**

Bits	Name	Description
17	UIM_TX_MODE	When this bit is set (1), any re-transmission signal from the UIM card will be ignored (The transmission portion of the SIM interface operates in block mode T=1). When clear (0), re-transmission will be executed upon request (the transmission portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
16	UIM_RX_MODE	When this bit is set (1), error-signal indication will not be generated by the receiver (the receive portion of the SIM interface operates in block mode T=1). When clear (0), error signal will be generated when a character error is detected (the receive portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
15:8	SIM_STOP_BIT_LEN	In UIM_IF mode, this field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 1 to 254 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. 0x1: ENUM_1_BIT_TIMES (1 bit times) 0x2: ENUM_2_BIT_TIMES (2 bit times) ..... 11 111110: 254 bit times)
7	SIM_CLK_ON	When this bit is set(1), the UIM_CLK is turned on at either the TD8 or TD4 frequency, as indicated by bit6(UIM_CLK_TD8_SEL). When this bit is cleared, the UIM_CLK is stopped either LOW or HIGH, as indicated by bit 5(UIM_CLK_STOP_HIGH).
6	SIM_CLK_TD8_SEL	This bit selects the UIM_CLK frequency at which the UIM_CLK is turned on. 0x1: TD8 (UIM_CLK runs at the TD8 frequency (1/2 the frequency of SIM-SRC_CLK)) 0x0: TD4 (UIM_CLK runs at the TD4 frequency (same frequency as SIM-SRC_CLK))
5	SIM_CLK_STOP_HIGH	This bit indicates at what point - high or low - the UIM_CLK will stop. 0x1: HIGH (high) 0x0: LOW (low)
3	MASK_RX	Setting (1) this bit masks off any 0 on the RX line, which prevents the receiver from seeing the START bit. Set (1) this bit if you do not want to receive the data you transmit.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SIM\_CFG (cont.)**

Bits	Name	Description
2	SWAP_D	Setting (1) this bit causes the UIM_IF to swap the 8 data bits (making MSB into LSB, bit 6 into bit1, and so on) before it is read by the microprocessor.
1	INV_D	Set (1) this bit to cause the UIM_IF to invert the 8 data bit before it is read by the microprocessor.
0	SIM_SEL	When set (1), UART_DM will operate in SIM/UIM interface mode. When cleared (0), SIM/UIM interface is disabled.

**0x0C16F0A0 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_CSR register is the UART clock selection register. This register is used to determine the bit rate for the transmitter and receiver. The transmitter and receiver can be clocked at different rates.

Data-rate of receiver/transmitter is UART-CLK frequency divided by the integer matching the configured value below:

0xF = 16

0xE = 32

0xD = 48

0xC = 64

0xB = 96

0xA = 128

0x9 = 192

0x8 = 256

0x7 = 384

0x6 = 512

0x5 = 768

0x4 = 1536

0x3 = 3072

0x2 = 6144

0x1 = 12288

0x0 = 24576

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR**

Bits	Name	Description
7:4	UART_RX_CLK_SEL	Use the values above to select the receive bit rate.
3:0	UART_TX_CLK_SEL	Use the values above to select the transmit bit rate.

#### **0x0C16F0A4 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SR**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

The UART\_DM\_SR register is the UART status register. This register is used to obtain the current state of the UART subsystem. This register is updated asynchronously.

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SR**

Bits	Name	Description
12	COMMAND_IN_PROGRESS	When 1 the command is in progress
11:10	TRANS_END_TRIGGER	RX-transfer-end event type. Stores trigger for last transfer end : '00' for no trigger, '01' for DMRX-event, '10' for timeout stale, '11' for SW stale.
9	TRANS_ACTIVE	Transfer Active bit.
8	RX_BREAK_START_LAST	When break start is detected, this bit is set (1). When break end was detected, this bit is reset (0).
7	HUNT_CHAR	When set (1), this bit indicates that the programmed character in the UART_DM_HCR register was received. If programmed, this condition causes the RXHUNT bit to be set (1) in the UART_DM_ISR register. This bit is cleared by issuing a reset error status command.
6	RX_BREAK	When set (1), this bit indicates that an all-zero character of the programmed length was received without a stop bit. If a break begins in the middle of a character, it is detected if the condition remains until the end of the next character time. Only one all zero character is written into the FIFO upon receiving the break. After the RX input returns high for at least half a bit time, then more characters are accepted by the receiver. Whenever the break condition starts or stops as defined above, the RXBREAK bit is set (1) in the UART_DM_ISR register. This bit is appended to the character in the receive FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SR (cont.)**

Bits	Name	Description
5	PAR_FRAME_ERR	A parity error occurs if parity is programmed to be checked and an incoming character is received with incorrect parity. A framing error occurs when the RX input is low while the stop bit is being sampled. This bit is sampled in the middle of the first stop bit position, regardless of the programmed length of the stop bit. This error bit is appended to the character in the receive FIFO.
4	UART_OVERRUN	An overrun error occurs if one or more incoming characters are lost. This happens when the receive FIFO is full, the packing buffer is full and another character has been received in the shift register. The character in the receive shift register is lost. Upon overrun, this status bit remains high until cleared by a reset error status command.
3	TXEMT	This bit is set (1) when the transmitter under-runs. It is set after the transmission of the stop bit at the end of a character, if there are no characters waiting to be sent. It is reset when a character is written to the transmit FIFO.
2	TXRDY	This bit indicates that the transmit FIFO has space in it. It is reset when the FIFO becomes full and set when space becomes available again. Any characters that are written to the FIFO while this bit is low is lost.
1	RXFULL	This bit is set (1) when the receive FIFO becomes full. It is reset when the CPU reads a word.
0	RXRDY	This bit is set (1) when there is a word in the receive FIFO waiting to be read. It is reset when the last word currently stored in the FIFO is read.

**0x0C16F0A8 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR**

Bits	Name	Description
11	CHANNEL_COMMAND_MSB	This is the msb of the CHANNEL_COMMAND bitfield.
10:8	GENERAL_COMMAND	Writing the appropriate value to these bits executes the commands that are listed in Table .
7:4	CHANNEL_COMMAND_LSB	Writing the appropriate value to these bits along with bit 11,executes the channel-command as listed in the register description.
3	UART_TX_DISABLE	This command terminates the operation of the transmitter after any character in the transmit shift register is sent.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR (cont.)**

Bits	Name	Description
2	UART_TX_EN	This command enables the operation of the transmitter.
1	UART_RX_DISABLE	This command immediately terminates the operation of the channel receiver and any incoming characters are lost. None of the receiver status bits are affected by this command and characters that are already in the receive FIFO remain there.
0	UART_RX_EN	This command enables the channel receiver.

**0x0C16F0AC PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR**

Bits	Name	Description
17:0	UART_MISR	The UART_DM_MISR register is the masked interrupt status register. A read from this register returns the 'AND' of the UART_DM_ISR and the UART_DM_IMR registers. This register is updated asynchronously. This bit field is misr <= (isr(16 DOWNTO 7) AND imr(16 DOWNTO 7)) & '0' & (isr(5 DOWNTO 0) AND imr(5 DOWNTO 0)).

**0x0C16F0B0 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IMR register is the UART interrupt mask register. This register is used to enable the corresponding functions in the UART\_DM\_ISR register. Setting (1) a bit in the UART\_DM\_IMR register causes an interrupt to be generated, if the corresponding bit in the UART\_DM\_ISR register is set. Clearing (0) a bit in the UART\_DM\_IMR register causes the setting of the corresponding bit in the UART\_DM\_ISR register to have no effect on the interrupt signal.

Bit 6 of the UART\_DM\_IMR register, CURRENT\_CTS, is slightly different. If the current value of the CTS is one (1), no interrupt is generated. However, the user can use bit 6 in this register to mask the CTS value when reading the UART\_DM\_MISR register or as a general-purpose bit.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Masking bit of the matching field interrupt bit
16	WWT_IRQ	Masking bit of the matching field interrupt bit
15	TXCOMP_IRQ	Masking bit of the matching field interrupt bit
14	RX_RD_ERROR_IRQ	Masking bit of the matching field interrupt bit
13	TX_WR_ERROR_IRQ	Masking bit of the matching field interrupt bit
12	PAR_FRAME_ERR_IRQ	Masking bit of the matching field interrupt bit
11	RXBREAK_END	Masking bit of the matching field interrupt bit
10	RXBREAK_START	Masking bit of the matching field interrupt bit
9	TX_DONE	Masking bit of the matching field interrupt bit
8	TX_ERROR	Masking bit of the matching field interrupt bit
7	TX_READY	Masking bit of the matching field interrupt bit
6	CURRENT_CTS	Masking bit of the matching field interrupt bit
5	DELTA_CTS	Masking bit of the matching field interrupt bit
4	RXLEV	Masking bit of the matching field interrupt bit
3	RXSTALE	Masking bit of the matching field interrupt bit
2	RXBREAK_CHANGE	Masking bit of the matching field interrupt bit
1	RXHUNT	Masking bit of the matching field interrupt bit
0	TXLEV	Masking bit of the matching field interrupt bit

**0x0C16F0B4 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_ISR register is the UART interrupt status register. This register provides the current status of the possible interrupt conditions. If one of these bits is set (1), and the corresponding bit in the UART\_DM\_IMR register is set (1), an interrupt is generated (with the exception of bit 6, CURRENT\_CTS). If the corresponding bit in the UART\_DM\_IMR register is clear (0), setting one of these bits has no effect on the UART interrupt request signal. This register is updated asynchronously.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	This interrupt is fires if: 1.Command protection is not used 2.A new command is issued while the previous command have not finished yet. Cleared by issuing CHANNEL_COMMAND = CLEAR_NO_FINISH_CMD_VIOL_IRQ (UART_DM_CR register).
16	WWT_IRQ	WWT timeout reached interrupt. Fired after WWT timer was enabled and no data was received during the configured time that follows. Cleared by issuing CHANNEL_COMMAND = CLEAR_WWT_IRQ (UART_DM_CR register).
15	TXCOMP_IRQ	TX-complete interrupt. Fired when TX channel finished transmitting all of TX-transfer's data and no new transfer was initiated. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_COMP_IRQ (UART_DM_CR register).
14	RX_RD_ERROR_IRQ	RX-FIFO empty read error. Fired when AHB master is trying to read from an empty RX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_RX_RD_ERROR_IRQ (UART_DM_CR register).
13	TX_WR_ERROR_IRQ	TX-FIFO full write error. Fired when AHB master is trying to write into a full TX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_WR_ERROR_IRQ (UART_DM_CR register).
12	PAR_FRAME_ERR_IRQ	This bit is asserted in the same condition at which PAR_FRAME_ERR status bit is asserted. Cleared by issuing CHANNEL_COMMAND = RESET_PAR_FRAME_ERR_IRQ (UART_DM_CR register).
11	RXBREAK_END	This bit is set (1) if the end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_END_IRQ (UART_DM_CR register).
10	RXBREAK_START	This bit is set (1) if the start of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_START_IRQ (UART_DM_CR register).
9	TX_DONE	This bit is used only in UIM/SIM_IF mode. Set (1) this bit when the last byte in the transmitter has been sent. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_DONE_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.
8	TX_ERROR	Only used in UIM_IF mode. Set (1) this bit to indicate that there has been parity error during transmission. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_ERROR_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR (cont.)**

Bits	Name	Description
7	TX_READY	This bit, when set(1), indicates that: 1. TX FIFO is empty. 2. The value written in NO_CHARS_FOR_TX register equals the number of bytes written to the TX FIFO since the register was updated. NOTE: There may be characters in the unpack buffer or in the shift register. Cleared by issuing GENERAL_COMMAND = RESET_TX_READY_IRQ (UART_DM_CR register)
6	CURRENT_CTS	This bit indicates the current state of the CTS input, high ('1') meaning peer is not ready for data reception, low ('0') - peer is ready. The toggling of this bit does not generate an interrupt.
5	DELTA_CTS	This bit, when set (1), indicates that the CTS input has changed states. Cleared by issuing CHANNEL_COMMAND = RESET_CTS_N (UART_DM_CR register).
4	RXLEV	This bit is set when a word is loaded into the receive FIFO that brings the total number of words in the FIFO above the programmed watermark level in the FIFO watermark register (UART_DM_RFWR). This bit is cleared after enough words have been read to bring the level equal to or below the programmed watermark level.
3	RXSTALE	This bit indicates that an RX-transfer has finished. Cleared by issuing CHANNEL_COMMAND = RESET_STALE_IRQ (UART_DM_CR register).
2	RXBREAK_CHANGE	This bit is set (1) if the start or end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_CHANGE_IRQ (UART_DM_CR register). A detected break is defined as an all-zeros character with an invalid stop bit AND LOW PARITY. The end of a break is defined as a mark condition (1) at least 1/2 bit width. A detected break occupies one position in the Rx FIFO.
1	RXHUNT	This bit is set (1) when an incoming character matches the value programmed into the UART_DM_HCR register. Cleared by issuing CHANNEL_COMMAND = RESET_ERROR_STATUS (UART_DM_CR register).
0	TXLEV	This bit is set (1) when a word which is transferred from the transmit FIFO to the transmit shift register brings the total number of words in the FIFO below or equal to the programmed watermark level in the UART_DM_TFWR register. This bit is cleared (0) after enough words have been written to the FIFO to bring the level above the programmed watermark level. Since this bit does not have a Clear command, during interrupt handling this bit must be masked (using the matching bit in UART_DM_IMR) until writing additional data for transmission is required. After data is written, this interrupt-bit should be unmasked.

**0x0C16F0B8 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IRDA register enables the IRDA function, selects a loopback, and allows inversion of RX and TX data. This register also controls the IRDA transceiver that optionally interfaces between the UART and the DP\_RX\_DATA and DP\_TX\_DATA pins.

The UART proper is a standard RX/TX configuration that converts the serial pin input and output lines to 8-bit data for the microprocessor. The configuration contains receive and transmit blocks, a control block, a bit rate generator for RX and TX, and FIFO interfaces with the microprocessor data bus. The IRDA register provides the means of switching IRDA circuits into the serial I/O lines and the IRDA function is normally switched out. The IRDA modulator/demodulator may be enabled or disabled under microprocessor control. When enabled, the RX and TX data paths have individual inversion select bits, so external true or inverted drivers may be used in the signal path.

\* The register is generated only when IRDA\_IFC\_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA**

Bits	Name	Description
4	MEDIUM_RATE_EN	- Set (1) for 1/4 bit-time pulse length (Medium rate) - Clear (0) for 3/16 bit-time pulse length (Low rate)
3	IRDA_LOOPBACK	- This bit loops the IRDA modulated TX line back into the IRDA RX line. The normal UART loopback mode must be off in order to see this effect.
2	INVERT_IRDA_TX	This bit inverts the polarity of the IRDA modulated signal on the DP_TX_DATA pin. - Set (1) this bit for an inverted polarity. - Clear (0) this bit for a non-inverted polarity.
1	INVERT_IRDA_RX	This bit inverts the polarity of the IRDA received signal on the DP_RX_DATA pin. - Set (1) this bit for inverted the polarity. - Clear (0) this bit for non-inverted polarity.
0	IRDA_EN	- Set (1) this bit to enable the IRDA transceiver. - Clear (0) this bit to bypass the IRDA transceiver and ignore the other bits of this register.

**0x0C16F0BC PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX\_TOTAL\_SNAP****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX\_TOTAL\_SNAP**

Bits	Name	Description
23:0	RX_TOTAL_BYTES	<ul style="list-style-type: none"> <li>•RX_TOTAL_SNAP register holds the number of characters received since the end of last Rx transfer. It is updated at the end of an Rx transfer.</li> <li>Rx transfer ends when one of the conditions is met: <ul style="list-style-type: none"> <li>- The number of characters which were received since the end of the previous transfer equals the value which was written to DMRX at 'Transfer initialization'.</li> <li>- A stale event occurred (flush operation already performed if was needed)</li> </ul> </li> </ul>

**0x0C16F0C0 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_WWT\_TIMEOUT****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_WWT\_TIMEOUT**

Bits	Name	Description
25	WWT_CYCLEREENABLE	When 1, a new received character will re-load the WWT counter instead of disabling it
24:0	WWT_TIMEOUT	timeout value for WWT mechanism. Minimal value should be 10 , otherwise the WWT_IRQ will be received even if two characters will follow each other

**0x0C16F0C4 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL**

Bits	Name	Description
23	UART_IRDA_CLK_CGC_OPEN	Forces UART IRDA-clock (uart_tx_clk) CGC to open when set.
22	UART_SIM_CLK_CGC_OPEN	Forces UART SIM-clock (uart_tx_clk) CGC to open when set.
21	UART_RX_CLK_CGC_OPEN	Forces UART RX-clock (uart_tx_clk) CGC to open when set.
20	UART_TX_CLK_CGC_OPEN	Forces UART TX-clock (uart_tx_clk) CGC to open when set.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL (cont.)**

Bits	Name	Description
14	AHB_RX_BAM_CLK_CGC_OPEN	Forces AHB RX-BAM-clock (hrx_bam_clk) CGC to open when set.
13	AHB_TX_BAM_CLK_CGC_OPEN	Forces AHB TX-BAM-clock (htx_bam_clk) CGC to open when set.
10	AHB_RX_CLK_CGC_OPEN	Forces AHB RX-clock (hrx_clk) CGC to open when set.
9	AHB_TX_CLK_CGC_OPEN	Forces AHB TX-clock (htx_clk) CGC to open when set.
8	AHB_WR_CLK_CGC_OPEN	Forces AHB write-clock (hwr_clk) CGC to open when set.
5	RX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to RX-DISABLE command/RX-RESET command only when set.
4	TX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to TX-DISABLE command/TX-RESET command only when set.
0	AHB_CLK_CGC_CLOSE	Setting this bit to '1' will completely gate AHB-clock to the core. Only allowed register acces during the duration this bit is set is the access to reset this bit. All other register accesses are not allowed and might result in errors. When set to '0', AHB clock is not gated at the root of the core and can be gated by the other gating features.

**0x0C16F0C8 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BCR**

Bits	Name	Description
6	IGNORE_CR_PROT_VIOL	Relevant only when command protection is disabled. If during on-going command a new command is issued the new command is disregarded. If the bit is disabled legacy behavior is used
5	RX_DMRX_1BYTE_RES_EN	Allows DMRX values other than a multiple of 16 if set.
4	RX_STALE_IRQ_DMRX_EQUAL	If set, Stale irq will be asserted when (DMRX==valid_char_count) and stale is not enabled.
2	RX_DMRX_LOW_EN	DMRX lower than valid_char_cnt logic is enabled if bit is set
1	STALE_IRQ_EMPTY	Stale interrupt when RX-FIFO is empty will fire if bit is set. This relates to the issue that was when state when SW reads all the characters from the fifo and timeout expired. In that case without this bit enable interrupt will not be fired
0	TX_BREAK_DISABLE	TX-pin value when transmitter disabled while in break state is '0' if set, else '1'

**0x0C16F0CC PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX\_TRANS\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX\_TRANS\_CTRL**

Bits	Name	Description
2	RX_DMRX_CYCLIC_EN	If set - Auto-Re-Activated RX-transfer will be re-activated with a DMRX value from the DMRX register, making a DMRX-transfer-end-event possible. Otherwise, Auto-Re-Activated RX-transfer cannot end with a DMRX event unless its value is written to the DMRX register throughout the transfer.
1	RX_TRANS_AUTO_RE_ACTIVE	RX-transfer will be automatically re-activated after last data of previous transfer was read - if set. Otherwise, transfers are only activated by writing to DMRX register.
0	RX_STALE_AUTO_RE_EN	RX-Stale automatic re-enable - on if set.

**0x0C16F0D4 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS**

Bits	Name	Description
29:28	TX_COMP_FSM	Value of TX-COMP FSM state : '00' for empty, '01' non-empty, '10' for empty-check, '11' for empty-check-fail.
26:24	RX_PACK_FSM	Value of RX-packing FSM state : '000' for idle, '001' for fifo_write, '010' for wait_ack, '011' for flush_check, '100' for flushing, '101' for flush_done.
21:20	RX_TRANS_FSM	Value of RX-transfer FSM state : '00' for inactive, '01' for active, '10' for active_dmrx, '11' for active_flush.
18:16	TX_TRANS_FSM	Value of TX-transfer FSM state : '000' for idle, '001' for bam_idle, '010' for active, '011' for inactive, '100' for last_trap, '101' for nwd.
14:12	RX_PRO_TRANS_END_FSM	Value of RX_PRO_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char_dphase, '101' for mess_only_dphase.
10:8	RX_PRO_ACTIVE_FSM	Value of RX_PRO_ACTIVE FSM state : '000' for disabled, '001' for isactive, '010' for wait_check_1, '011' for wait_check_1, '100' for pf_pending, '101' for wait_update_1, '110' for wait_update_2, '111' for inactive?

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS (cont.)**

Bits	Name	Description
6:4	TX_CON_TRANS_END_FS_M	Value of TX_CON_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char dphase, '101' for mess_only_dphase, '110' for normal_dphase.
0	RX_TRANSFER_ACTIVE	1 is RX transfer is currently active, 0 if inactive.

**0x0C16F0DC PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_GENERICS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_GENERICS**

Bits	Name	Description
7	GENERIC_BAM_IFC	Value of 'BAM_IFC_GEN' generic.
6	GENERIC_DM_IFC	Value of 'DM_IFC_GEN' generic.
5	GENERIC_IRDA_IFC	Value of 'IRDA_IFC_GEN' generic
4	GENERIC_SIM_GLUE	Value of 'SIM_GLUE_GEN' generic
3:0	GENERIC_RAM_ADDR_WIDHTH	Value of 'RAM_ADDR_WIDTH' generic.

**0x0C16F100 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

In order to support bursts of 4, 0x70-0x7C address space is reserved for the UART\_DM TX FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF**

Bits	Name	Description
31:0	UART_TF	The UART_TF register is the UART transmit FIFO register. This register is used to access the channel transmit FIFO. If the FIFO is full at the time of writing, the characters are lost and an interrupt is generated.

**0x0C16F104 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_2****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_2**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F108 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_3****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_3**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F10C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_4****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_4**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F110 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_5****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_5**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F114 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_6****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_6**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F118 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_7****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_7**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F11C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_8****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_8**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F120 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_9****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_9**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F124 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_10****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_10**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F128 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_11****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_11**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F12C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_12****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_12**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F130 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_13****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_13**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F134 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_14****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_14**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F138 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_15****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_15**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F13C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_16****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_16**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C16F140 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RF register is the UART receive FIFO register, which is used to access the channel receive FIFO. In order to support bursts of 16, 0x140-0x17C address space is reserved for the UART\_DM\_RX FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF**

Bits	Name	Description
31:0	UART_RF	This register returns the next word in the receive FIFO. After the read is completed, the FIFO read pointer is updated to make the next word available.

**0x0C16F144 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_2****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_2**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F148 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_3****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_3**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F14C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_4****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_4**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F150 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_5****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_5**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F154 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_6****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_6**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F158 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_7****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_7**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F15C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_8****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_8**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F160 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_9****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_9**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F164 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_10****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_10**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F168 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_11****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_11**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F16C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_12****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_12**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F170 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_13**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_13**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F174 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_14**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_14**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F178 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_15**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_15**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F17C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_16**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_16**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C16F180 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CFG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00001703

Configuration register for UIM-controller (linked with this UART\_DM). Reset value relevant to UIM-controller HW-reset and not a UART\_DM HW-reset. Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CFG**

Bits	Name	Description
15	BATT_ALARM_QUICK_DRO_P_EN	When set, deactivation sequence resulted from battery-alarm will drop all UIM-lines immediately. Otherwise, lines will be dropped gradually.
14		
13	SW_RESET	When set, UIM-controller will be at reset state.
12	MODE18	Configures PAD's working reference voltage (set to '0' for 3V, '1' for 1.8V).
10	PMIC_ALARM_EN	Enables messaging on the alarm-line to notify PMIC to drop card's voltage supply. Set to enable messaging, clear to disable.
9	BATT_ALARM_TRIGGER_EN	Enables triggering of deactivation sequence when battery-alarm detected. Set to enable triggering, clear to disable.
8	UIM_RMV_TRIGGER_EN	Enables triggering of deactivation sequence when card-removal detected. Set to enable triggering, clear to disable.
6	UIM_CARD_EVENTS_ENABLE	When cleared, UIM-controller will not react to any card-event (insertion/removal). When set, UIM-controller reacts to these events and considered active. This bit should be set only after/during UIM_PRESENT_POLARITY bit is configured so UIM-controller will not interpret card-events incorrectly due to a wrong polarity configuration (the reset-value of the polarity might not match the actual one).
5	UIM_PRESENT_POLARITY	Determines value interpretation of uim_present signal. Setting this bit to '1' indicates that a uim-card present in the uim-slot will return the value '1' on uim_present line & '0' when a card is not present. Setting this bit to '0' indicates that a uim-card present in the uim-slot will return the value '0' on uim_present line & '1' when a card is not present.
4:0	EVENT_DEBOUNCE_TIME	Configures debounce-time of card-removal/insertion events (in sleep-clk cycles).

**0x0C16F184 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CMD****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

Command register for UIM-controller (linked with this UART\_DM). Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CMD**

Bits	Name	Description
1	RECOVER_FROM_HW_DEACTIVATION	Writing '1' to this bit will cause the UIM-io-ctrl to exit override state and reflect UIM signals from UART_DM to the UIM card. Setting both bits of this register to '1' will cause neither of the commands to be executed.
0	INITIATE_HW_DEACTIVATION	Initiates a HW deactivation sequence when writing '1' to this bit. This executes the sequence on the uim_data, uim_clk & uim_rst_n lines and alerts the PMIC to cut the power-supply to the card afterwards. Setting both bits of this register to '1' will cause neither of the commands to be executed.

**0x0C16F188 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000002**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STATUS**

Bits	Name	Description
2	UIM_IO_WRITE_IN_PROGRESS	Asserted after UIM-write (UIM_CFG/UIM_CMD register write) while write is being propagated into the UIM-IO-controller. An additional read/write to UIM_CFG/UIM_CMD register should be done while high. De-asserted when write finishes propagating.
1	UIM_DEACTIVATION_STATUS	Indicates whether a HW deactivation sequence was performed and sequence recovery hasn't been executed yet. This bit is set when the sequence is initiated either by a HW trigger (batt_alarm, card-removal) or a SW trigger ('initiate_shutdown_sequence' bit in CMD register). Cleared after executing a 'recover_from_hw_deactivation' command.
0	CARD_PRESENCE	Indication on card presence in slot. High when card present, low otherwise.

**0x0C16F18C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Asserted when UIM-write had finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
3	HW_SEQUENCE_FINISH	Asserted when UIM-controller's HW-deactivation sequence has been initiated and finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
2	BATT_ALARM	Set to high when battery-alarm indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
1	UIM_CARD_INSERTION	Set to high when card insertion indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
0	UIM_CARD_REMOVAL	Set to high when card removal indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.

**0x0C16F190 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	See UART_DM_UIM_IRQ_ISR for field descriptions.
3	HW_SEQUENCE_FINISH	See UART_DM_UIM_IRQ_ISR for field descriptions.
2	BATT_ALARM	See UART_DM_UIM_IRQ_ISR for field descriptions.
1	UIM_CARD_INSERTION	See UART_DM_UIM_IRQ_ISR for field descriptions.
0	UIM_CARD_REMOVAL	See UART_DM_UIM_IRQ_ISR for field descriptions.

**0x0C16F194 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
3	HW_SEQUENCE_FINISH	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
2	BATT_ALARM	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
1	UIM_CARD_INSERTION	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
0	UIM_CARD_REMOVAL	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.

**0x0C16F198 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
3	HW_SEQUENCE_FINISH	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
2	BATT_ALARM	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
1	UIM_CARD_INSERTION	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
0	UIM_CARD_REMOVAL	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.

**0x0C16F19C PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_SET****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_SET**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C16F1A0 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C16F0E0 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

New added interrupt clear register

**PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR\_CLR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Clears NO_FINISH_CMD_VIOL bit in UART_DM_ISR

**0x0C170000 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR1****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_MR1 register is the UART mode register #1. It is used, along with UART\_DM\_MR2, to configure the operational mode of the UART.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR1**

Bits	Name	Description
31:8	AUTO_RFR_LEVEL1	<p>These bits are used, along with field AUTO_RFR_LEVEL0 to program the level in the receive FIFO at which the Ready-For-Receive signal (RFR_N) toggles, if programmed to do so.</p> <p>When RX-FIFO level is greater than the value in fields AUTO_RFR_LEVEL1 and AUTO_RFR_LEVEL0 , signal will indicate Not-Ready (RFR_N high). Otherwise signal will indicate Ready RFR_N low.</p> <p>Configured value should be higher/equal than RFWR value, otherwise the watermark will never be reached.</p> <p>Value of this register is in words and can be programmed from 1 to <math>2^{\text{RAM\_ADDR\_WIDTH}}</math>.</p> <p>- Only RAM_ADDR_WIDTH +1:8 bits of this field are generated.</p>
7	RX_RDY_CTL	<p>Setting (1) this bit enables the automatic ready-for-receiving (RFR_N) control for the receiver. RFR_N is turned off, or set (1), when a valid start bit is received and the RX channel FIFO is at the level programmed in bits [31:8,5:0] of this register. When the FIFO level falls back to the programmed level, the RFR_N signal is turned on, or clear (0). When this feature is off, the RFR_N signal can be used by normal signal port bit manipulation.</p>
6	CTS_CTL	<p>When this bit is set(1), the transmitter checks the CTS_N input to determine whether to begin transmission of a new character. If CTS_N is low, the character is sent. Otherwise the transmitter continues marking until CTS_N goes low, then the next character is transmitted. A change on CTS_N during the transmission of a character has no effect on that character.</p> <p>When this bit is clear(0), the CTS_N input for the channel has no effect on the transmitter.</p>
5:0	AUTO_RFR_LEVEL0	Same description of field AUTO_RFR_LEVEL1.

**0x0C170004 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR2****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR2**

Bits	Name	Description
10	RFR_CTS_LOOPBACK	Enables internal loopback between RFR_N of RX channel and CTS_N of TX channel if set. For this feature to be enabled, data loopback must also be enabled (LOOPBACK bit of this register).
9	RX_ERROR_CHAR_OFF	When this bit is set, characters received with parity or framing errors get discarded and not moved to the RX channel. Otherwise they enter RX channel as any valid character.
8	RX_BREAK_ZERO_CHAR_OFF	When this bit is set, the zero character received at rx_break event get discarded and not moved to the RX channel. Otherwise it enters RX-channel as any valid character.
7	LOOPBACK	When set - data output of transmitter is fed back to the data input of the receiver, i.e. TX-pin is connected to RX-pin. Might be used with flow-control loopback (RFR_CTS_LOOPBACK bit above).
6	ERROR_MODE	This bit controls the operation of the two status register bits for the RX channel (parity or framing error and received break). <ul style="list-style-type: none"> <li>- When clear (0), the UART operates in character mode and the status bits apply only to the last character received.</li> <li>- When set (1), the UART operates in block mode and both bits are sticky, i.e. they go high once an error/break-event is detected and go low only after reset error status command is issued.</li> </ul>
5:4	BITS_PER_CHAR	These bits determine how many data bits are transmitted or received per character, not including the start, stop, and parity bits. <ul style="list-style-type: none"> <li>0x0: ENUM_5_BITS (5 bits)</li> <li>0x1: ENUM_6_BITS (6 bits)</li> <li>0x2: ENUM_7_BITS (7 bits)</li> <li>0x3: ENUM_8_BITS (8 bits)</li> </ul>
3:2	STOP_BIT_LEN	This field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 9/16, 1, 1+ 9/16, and 2 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. This configuration only applies to UART/IrDA interfaces. <ul style="list-style-type: none"> <li>0x0: ENUM_0_563 (0.563 (9/16) bit times)</li> <li>0x1: ENUM_1_000_BIT_TIME (1.000 bit time)</li> <li>0x2: ENUM_1_563 (1.563 (1+9/16) bit times)</li> <li>0x3: ENUM_2_000_BIT_TIMES (2.000 bit times)</li> </ul>

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR2 (cont.)**

Bits	Name	Description
1:0	PARITY_MODE	<p>These bits determine which parity mode is used. The user can select between odd, even, space, or no parity.</p> <p>0x0: NO_PARITY (no parity - last data-bit is followed by a stop-bit)</p> <p>0x1: ODD_PARITY (odd parity - parity-bit ensures number of ones in data-bits+parity-bit is odd)</p> <p>0x2: EVEN_PARITY (even parity - parity-bit ensures number of ones in data-bits+parity-bit is even)</p> <p>0x3: SPACE_PARITY (space parity - parity is always '0')</p>

**0x0C170008 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR\_SR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CSR or UART\_DM\_SR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR\_SR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CSR_SR_DEPRECATED	

**0x0C170010 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR\_MISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CR or UART\_DM\_MISR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR\_MISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CR_MISR_DEPRECATED	

**0x0C170014 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR\_ISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IMR or UART\_DM\_ISR.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR\_ISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IMR_ISR_DEPRECATE	

**0x0C170018 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IPR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0xFFFFFFF9F

The UART\_DM\_IPR register is the UART interrupt programming register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IPR**

Bits	Name	Description
31:7	STALE_TIMEOUT_MSB	The stale character time-out duration field contains a number from 1 to $2^{30} - 1$ . This number determines how many character times must elapse before a stale event is generated. This character time is defined as 10 times the bit rate. It is reset whenever a data bit is received. Never set this fields to 0 while RX channel is operational. Do not re-configure while an RX-transfer is active. NOTE: the discontinuity in the bit assignments.
6	SAMPLE_DATA	Currently not in use. Setting (1) to this bit enables the sample data mode. In this mode start bit is sampled along with the data of the UART character.
4:0	STALE_TIMEOUT_LSB	This bitfield is the LSbits of the STALE_TIMEOUT bitfield.

**0x0C17001C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_TFWR register is the UART transmit FIFO watermark register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TFWR**

Bits	Name	Description
31:0	TFW	<p>These bits contain a number, between 0 and <math>2^{\text{RAM\_ADDR\_WIDTH}} - 1</math>, that determines the level of the transmit (TX) FIFO at which space_avail_req signal is asserted in BAM mode or TXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of TX words in the FIFO) is less than or equal to the value in TFWR. Notice that the value is in words.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH -1:0 bits are generated.</li> </ul>

**0x0C170020 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RFWR register is the UART receive FIFO watermark register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RFWR**

Bits	Name	Description
31:0	RFW	<p>These bits contain a number, between 0 and <math>2^{\text{RAM\_ADDR\_WIDTH}} - 1</math>, that determines the level of the receive (RX) FIFO at which data_avail_req signal is asserted in BAM mode or RXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of words in the RX FIFO) is greater than the value in RFWR. Configured value should be lower/equal than AUTO_RFR_LEVEL value, otherwise the watermark will never be reached. Notice that the value is in words.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH -1:0 bits are generated.</li> </ul>

**0x0C170024 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_HCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Configures Hunt-Character value.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_HCR**

Bits	Name	Description
7:0	DATA	The UART_DM_HCR register is the UART hunt character register, which contains the character for which the receiver looks to assert a hunt character received interrupt.

**0x0C170034 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMRX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initializes RX transfer and configures maximal length in bytes of the transfer.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMRX**

Bits	Name	Description
24:0	RX_DM_CRCI_CHARS	A write to RX_DM_CRCI_CHARS initializes RX transfer in SW (FIFO) mode, with written value configuring the maximal length in bytes of the transfer. In BAM-mode, only the value held in this register impacts the data-flow and not the write-event. A write-value of 0 is not valid. Read of UART_DM_DMRX register is returns the number of characters that were received since the end of the last RX transfer.

**0x0C170038 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IRDA or UART\_DM\_RX\_TOTAL\_SNAP.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IRDA_RX_TOTAL_SNAP_DEPRECATED	

**0x0C17003C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMEN****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_DMEN register defines the DMA used by the core and the data packing mode for TX and RX channels.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMEN**

Bits	Name	Description
5	RX_SC_ENABLE	- Set (1) for work in Single-Character mode for RX channel (every character received is zero-padded into a word). - Clearing this bit requires resetting the receiver.
4	TX_SC_ENABLE	- Set (1) for work in Single-Character mode for TX channel (Only LSB byte of each word get transmitted). - Clearing this bit requires resetting the transmitter.
3	RX_BAM_ENABLE	- Set (1) this bit to enable RX BAM interface (BAM-mode). - Clear (0) this bit to disable RX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the receiver.
2	TX_BAM_ENABLE	- Set (1) this bit to enable TX BAM interface (BAM-mode). - Clear (0) this bit to disable TX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the transmitter.

**0x0C170040 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initialization of TX transfer in SW/FIFO mode.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX**

Bits	Name	Description
23:0	TX_TOTAL_TRANS_LEN	The total number of characters for transmission. Before writing a new value, the TX FIFO must be empty as indicated by TX_READY interrupt or after a TX-channel reset. It is used by the TX-channel in SW/FIFO mode to calculate how many characters to transmit in the last word. Any additional writes to the TX FIFO beyond TX_TOTAL_TRANS_LEN characters will be discarded. A TX_READY interrupt is triggered after TX_TOTAL_TRANS_LEN number of characters were moved from the TX FIFO to the unpacking buffer (not all may have been transmitted at that point though).

**0x0C170044 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BADR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

Configures FIFO division between RX-FIFO and TX-FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BADR**

Bits	Name	Description
31:2	RX_BASE_ADDR	<p>RX FIFO base address. Both FIFOs use the same RAM (<math>2^{\text{RAM\_ADDR\_WIDTH}}</math>, 32-bit entries). This register controls the division of the memory between the RX and TX FIFOs. The division must be a multiple of 4 entries for legacy reasons, therefore bits 1:0 of this register are not used.</p> <p>The size of the TX FIFO equals RX_BASE_ADDR. The size of the RX FIFO is <math>2^{\text{RAM\_ADDR\_WIDTH}} - \text{RX\_BASE\_ADDR}</math>.</p> <ul style="list-style-type: none"> <li>* The default is RX_BASE_ADDR = <math>2^{(\text{RAM\_ADDR\_WIDTH} - 1)}</math></li> <li>* Only RAM_ADDR_WIDTH -1:2 bits are generated.</li> </ul>

**0x0C17004C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

TX channel status register.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TXFS**

Bits	Name	Description
31:14	TX_FIFO_STATE_MSB	<p>The msb of TX_FIFO_STATE bitfield.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.</li> </ul>
13:10	TX_ASYNC_FIFO_STATE	TX asynchronous fifo status - number of characters in the TX asynchronous FIFO.
9:7	TX_BUFFER_STATE	TX unpacking buffer status - number of bytes in the unpacking buffer
6:0	TX_FIFO_STATE_LSB	TX FIFO status - number of words in the TX FIFO

**0x0C170050 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

RX channel status register.

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RXFS**

Bits	Name	Description
31:14	RX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	RX_ASYNC_FIFO_STATE	RX asynchronous fifo status - number of characters in the RX asynchronous FIFO.
9:7	RX_BUFFER_STATE	RX packing buffer status - number of bytes in the packing buffer
6:0	RX_FIFO_STATE_LSB	The number of entries in the RX FIFO that have at least one valid word. It is incremented each time a word is moved from the packing register to the RX FIFO. It is decremented each time a word is read from the RX FIFO.  NOTE: The Uart does not keep track of non-valid characters in each word.

#### **0x0C170060 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_MODE**

Type: RW

Clock: WR\_CLK

Reset State: Undefined

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_MODE**

Bits	Name	Description
1:0	RESERVED	RESERVED

#### **0x0C170064 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_RESET**

Type: W

Clock: WR\_CLK

Reset State: Undefined

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_RESET**

Bits	Name	Description
0	RESET	

**0x0C17006C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_VAL****Type:** R**Clock:** WR\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_VAL**

Bits	Name	Description
9:0	VAL	Current MISR state

**0x0C170080 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SIM\_CFG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_SIM\_CFG register is used to configure the SIM interface for the UART.

\* This register is generated only when SIM\_GLUE\_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SIM\_CFG**

Bits	Name	Description
17	UIM_TX_MODE	When this bit is set (1), any re-transmission signal from the UIM card will be ignored (The transmission portion of the SIM interface operates in block mode T=1). When clear (0), re-transmission will be executed upon request (the transmission portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
16	UIM_RX_MODE	When this bit is set (1), error-signal indication will not be generated by the receiver (the receive portion of the SIM interface operates in block mode T=1). When clear (0), error signal will be generated when a character error is detected (the receive portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
15:8	SIM_STOP_BIT_LEN	In UIM_IF mode, this field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 1 to 254 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. 0x1: ENUM_1_BIT_TIMES (1 bit times) 0x2: ENUM_2_BIT_TIMES (2 bit times) ..... 11 111110: 254 bit times)

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SIM\_CFG (cont.)**

Bits	Name	Description
7	SIM_CLK_ON	When this bit is set(1), the UIM_CLK is turned on at either the TD8 or TD4 frequency, as indicated by bit6(UIM_CLK_TD8_SEL). When this bit is cleared, the UIM_CLK is stopped either LOW or HIGH, as indicated by bit 5(UIM_CLK_STOP_HIGH).
6	SIM_CLK_TD8_SEL	This bit selects the UIM_CLK frequency at which the UIM_CLK is turned on. 0x1: TD8 (UIM_CLK runs at the TD8 frequency (1/2 the frequency of SIM-SRC_CLK)) 0x0: TD4 (UIM_CLK runs at the TD4 frequency (same frequency as SIM-SRC_CLK))
5	SIM_CLK_STOP_HIGH	This bit indicates at what point - high or low - the UIM_CLK will stop. 0x1: HIGH (high) 0x0: LOW (low)
3	MASK_RX	Setting (1) this bit masks off any 0 on the RX line, which prevents the receiver from seeing the START bit. Set (1) this bit if you do not want to receive the data you transmit.
2	SWAP_D	Setting (1) this bit causes the UIM_IF to swap the 8 data bits (making MSB into LSB, bit 6 into bit1, and so on) before it is read by the microprocessor.
1	INV_D	Set (1) this bit to cause the UIM_IF to invert the 8 data bit before it is read by the microprocessor.
0	SIM_SEL	When set (1), UART_DM will operate in SIM/UIM interface mode. When cleared (0), SIM/UIM interface is disabled.

**0x0C1700A0 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_CSR register is the UART clock selection register. This register is used to determine the bit rate for the transmitter and receiver. The transmitter and receiver can be clocked at different rates.

Data-rate of receiver/transmitter is UART-CLK frequency divided by the integer matching the configured value below:

0xF = 16

0xE = 32

0xD = 48  
 0xC = 64  
 0xB = 96  
 0xA = 128  
 0x9 = 192  
 0x8 = 256  
 0x7 = 384  
 0x6 = 512  
 0x5 = 768  
 0x4 = 1536  
 0x3 = 3072  
 0x2 = 6144  
 0x1 = 12288  
 0x0 = 24576

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR**

Bits	Name	Description
7:4	UART_RX_CLK_SEL	Use the values above to select the receive bit rate.
3:0	UART_TX_CLK_SEL	Use the values above to select the transmit bit rate.

#### **0x0C1700A4 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SR**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

The UART\_DM\_SR register is the UART status register. This register is used to obtain the current state of the UART subsystem. This register is updated asynchronously.

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SR**

Bits	Name	Description
12	COMMAND_IN_PROGRESS	When 1 the command is in progress
11:10	TRANS_END_TRIGGER	RX-transfer-end event type. Stores trigger for last transfer end : '00' for no trigger, '01' for DMRX-event, '10' for timeout stale, '11' for SW stale.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SR (cont.)**

Bits	Name	Description
9	TRANS_ACTIVE	Transfer Active bit.
8	RX_BREAK_START_LAST	When break start is detected, this bit is set (1). When break end was detected, this bit is reset (0).
7	HUNT_CHAR	When set (1), this bit indicates that the programmed character in the UART_DM_HCR register was received. If programmed, this condition causes the RXHUNT bit to be set (1) in the UART_DM_ISR register. This bit is cleared by issuing a reset error status command.
6	RX_BREAK	When set (1), this bit indicates that an all-zero character of the programmed length was received without a stop bit. If a break begins in the middle of a character, it is detected if the condition remains until the end of the next character time. Only one all zero character is written into the FIFO upon receiving the break. After the RX input returns high for at least half a bit time, then more characters are accepted by the receiver. Whenever the break condition starts or stops as defined above, the RXBREAK bit is set (1) in the UART_DM_ISR register. This bit is appended to the character in the receive FIFO.
5	PAR_FRAME_ERR	A parity error occurs if parity is programmed to be checked and an incoming character is received with incorrect parity. A framing error occurs when the RX input is low while the stop bit is being sampled. This bit is sampled in the middle of the first stop bit position, regardless of the programmed length of the stop bit. This error bit is appended to the character in the receive FIFO.
4	UART_OVERRUN	An overrun error occurs if one or more incoming characters are lost. This happens when the receive FIFO is full, the packing buffer is full and another character has been received in the shift register. The character in the receive shift register is lost. Upon overrun, this status bit remains high until cleared by a reset error status command.
3	TXEMT	This bit is set (1) when the transmitter under-runs. It is set after the transmission of the stop bit at the end of a character, if there are no characters waiting to be sent. It is reset when a character is written to the transmit FIFO.
2	TXRDY	This bit indicates that the transmit FIFO has space in it. It is reset when the FIFO becomes full and set when space becomes available again. Any characters that are written to the FIFO while this bit is low is lost.
1	RXFULL	This bit is set (1) when the receive FIFO becomes full. It is reset when the CPU reads a word.
0	RXRDY	This bit is set (1) when there is a word in the receive FIFO waiting to be read. It is reset when the last word currently stored in the FIFO is read.

**0x0C1700A8 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_CR register is the UART command register. This register is used to issue specific commands to the UART subsystem. This register is updated asynchronously.

**CAUTION** Do not reset the transmitter and disable it at the same time. Do not reset the receiver and disable it at the same time.

Channel-commands:

0x00 - Null Command - Does nothing.

0x01 - Reset Receiver - Resets the receiver as if a hardware reset were issued. The receiver is disabled and the FIFO, packing buffer and shift registers are flushed.

0x02 - Reset Transmitter - Resets the transmitter as if a hardware reset were issued. The transmitter signal goes high (marking) and the FIFO, unpacking register and shift register are flushed.

0x03 - Reset error status - Clears the overrun error and hunt char received status bits in both the character and block error modes. In the block error mode, it clears the error status and received break.

0x04 - Reset break change interrupt - Clears the break change interrupt status bit.

0x05 - Start break - Forces the transmitter signal low. The transmitter must be enabled. If the transmitter is busy, the break is started when all characters in the transmit FIFO and the transmit shift register have been completely sent.

0x06 - Stop break - If executed while channel is breaking, this command causes the transmitter signal to go high. The signal remains high for at least one bit time before sending out a new character.

0x07 - Reset CTS\_N - Clears ISR bit 5.

0x08 - Reset stale interrupt - Clears the stale interrupt.

0x09 - Packet mode - Turns on the sample data mode, which causes the receiver to sample the receive data stream at 16 times the programmed baud rate. The data is sampled with the start of the start bit, or the first data bit, and continued until the marking state. To exit this state, write 1100 in the command field.

0x0C - Mode reset - Turns off the sample data mode.

0x0D - Set RFR - Asserts the ready for receiving signal (actual flow-control port is active-low (rfr\_n) and will go low following this command).

0x0E - Reset RFR - Deasserts the ready for receiving signal (actual flow-control port is active-low (rfr\_n) and will go high following this command).

- 0x0F - uart\_reset\_int - .
- 0x10 - Reset TX\_ERROR - Clears TX\_ERROR
- 0x11 - Clear TX\_DONE - Clears the TX\_DONE interrupt (ISR bit 9)
- 0x12 - Reset break start interrupt - Clears the break start interrupt status bit.
- 0x13 - Reset break end interrupt - Clears the break end interrupt status bit.
- 0x14 - Reset par\_frame\_err interrupt - Clears the par\_frame\_err interrupt status bit.
- 0x15 - Start RX BAM IFC - Initialize producer sidebands towards BAM after reset.
- 0x16 - Start TX BAM IFC - Initialize consumer sidebands towards BAM after reset.
- 0x17 - WWT counter enable - Enable WWT counter & IRQ.
- 0x18 - WWT counter disable - Disable WWT counter & IRQ.
- 0x19 - Clear TX\_WR\_ERROR IRQ - Clears TX write error interrupt status bit.
- 0x1A - Clear RX\_RD\_ERROR IRQ - Clears RX read error interrupt status bit.
- 0x1B - Clear TX\_COMP IRQ - Clears TX complete interrupt status bit.
- 0x1C - Clear WWT IRQ - Clears WWT timeout interrupt status bit.
- 0x1D - Clear NO\_FINISH\_CMD\_VIOL - Clear NO\_FINISH\_CMD\_VIOL interrupt status bit.

#### General-commands:

0x0 - Null Command - Does nothing.

0x1 - CR Protection Enable - Enables CR HW protection. When two consecutive writes to the CR are detected, the second write is delayed until the command of the first write is finished. The delay is done by de-asserting the AHB ready and this ensures that the first command completes and the second one will be executed right afterward.

0x2 - CR Protection Disable - Disables CR HW protection. SW is responsible for managing delay between writes to the CR register.

0x3 - Reset TX-Ready interrupt - Clears the TX\_READY interrupt.

0x4 - SW Force Stale - Causes a stale event (even if stale events are disabled).

0x5 - Enable Stale Event - Enables the stale event mechanism

0x6 - Disable Stale Event - Enables the stale event mechanism

0x7 - RESERVED

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR**

Bits	Name	Description
11	CHANNEL_COMMAND_MS_B	This is the msb of the CHANNEL_COMMAND bitfield.
10:8	GENERAL_COMMAND	Writing the appropriate value to these bits executes the commands that are listed in Table .
7:4	CHANNEL_COMMAND_LSB	Writing the appropriate value to these bits along with bit 11,executes the channel-command as listed in the register description.
3	UART_TX_DISABLE	This command terminates the operation of the transmitter after any character in the transmit shift register is sent.
2	UART_TX_EN	This command enables the operation of the transmitter.
1	UART_RX_DISABLE	This command immediately terminates the operation of the channel receiver and any incoming characters are lost. None of the receiver status bits are affected by this command and characters that are already in the receive FIFO remain there.
0	UART_RX_EN	This command enables the channel receiver.

**0x0C1700AC PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR**

Bits	Name	Description
17:0	UART_MISR	The UART_DM_MISR register is the masked interrupt status register. A read from this register returns the 'AND' of the UART_DM_ISR and the UART_DM_IMR registers. This register is updated asynchronously. This bit field is misr <= (isr(16 DOWNTO 7) AND imr(16 DOWNTO 7)) & '0' & (isr(5 DOWNTO 0) AND imr(5 DOWNTO 0)).

**0x0C1700B0 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IMR register is the UART interrupt mask register. This register is used to enable the corresponding functions in the UART\_DM\_ISR register. Setting (1) a bit in the UART\_DM\_IMR register causes an interrupt to be generated, if the corresponding bit in the UART\_DM\_ISR register is set. Clearing (0) a bit in the UART\_DM\_IMR register causes the

setting of the corresponding bit in the UART\_DM\_ISR register to have no effect on the interrupt signal.

Bit 6 of the UART\_DM\_IMR register, CURRENT\_CTS, is slightly different. If the current value of the CTS is one (1), no interrupt is generated. However, the user can use bit 6 in this register to mask the CTS value when reading the UART\_DM\_MISR register or as a general-purpose bit.

### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Masking bit of the matching field interrupt bit
16	WWT_IRQ	Masking bit of the matching field interrupt bit
15	TXCOMP_IRQ	Masking bit of the matching field interrupt bit
14	RX_RD_ERROR_IRQ	Masking bit of the matching field interrupt bit
13	TX_WR_ERROR_IRQ	Masking bit of the matching field interrupt bit
12	PAR_FRAME_ERR_IRQ	Masking bit of the matching field interrupt bit
11	RXBREAK_END	Masking bit of the matching field interrupt bit
10	RXBREAK_START	Masking bit of the matching field interrupt bit
9	TX_DONE	Masking bit of the matching field interrupt bit
8	TX_ERROR	Masking bit of the matching field interrupt bit
7	TX_READY	Masking bit of the matching field interrupt bit
6	CURRENT_CTS	Masking bit of the matching field interrupt bit
5	DELTA_CTS	Masking bit of the matching field interrupt bit
4	RXLEV	Masking bit of the matching field interrupt bit
3	RXSTALE	Masking bit of the matching field interrupt bit
2	RXBREAK_CHANGE	Masking bit of the matching field interrupt bit
1	RXHUNT	Masking bit of the matching field interrupt bit
0	TXLEV	Masking bit of the matching field interrupt bit

### **0x0C1700B4 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

The UART\_DM\_ISR register is the UART interrupt status register. This register provides the current status of the possible interrupt conditions. If one of these bits is set (1), and the corresponding bit in the UART\_DM\_IMR register is set (1), an interrupt is generated (with the exception of bit 6, CURRENT\_CTS). If the corresponding bit in the UART\_DM\_IMR register is

clear (0), setting one of these bits has no effect on the UART interrupt request signal. This register is updated asynchronously.

#### **PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	This interrupt is fires if: 1.Command protection is not used 2.A new command is issued while the previous command have not finished yet. Cleared by issuing CHANNEL_COMMAND = CLEAR_NO_FINISH_CMD_VIOL_IRQ (UART_DM_CR register).
16	WWT_IRQ	WWT timeout reached interrupt. Fired after WWT timer was enabled and no data was received during the configured time that follows. Cleared by issuing CHANNEL_COMMAND = CLEAR_WWT_IRQ (UART_DM_CR register).
15	TXCOMP_IRQ	TX-complete interrupt. Fired when TX channel finished transmitting all of TX-transfer's data and no new transfer was initiated. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_COMP_IRQ (UART_DM_CR register).
14	RX_RD_ERROR_IRQ	RX-FIFO empty read error. Fired when AHB master is trying to read from an empty RX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_RX_RD_ERROR_IRQ (UART_DM_CR register).
13	TX_WR_ERROR_IRQ	TX-FIFO full write error. Fired when AHB master is trying to write into a full TX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_WR_ERROR_IRQ (UART_DM_CR register).
12	PAR_FRAME_ERR_IRQ	This bit is asserted in the same condition at which PAR_FRAME_ERR status bit is asserted. Cleared by issuing CHANNEL_COMMAND = RESET_PAR_FRAME_ERR_IRQ (UART_DM_CR register).
11	RXBREAK_END	This bit is set (1) if the end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_END_IRQ (UART_DM_CR register).
10	RXBREAK_START	This bit is set (1) if the start of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_START_IRQ (UART_DM_CR register).
9	TX_DONE	This bit is used only in UIM/SIM_IF mode. Set (1) this bit when the last byte in the transmitter has been sent. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_DONE_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.
8	TX_ERROR	Only used in UIM_IF mode. Set (1) this bit to indicate that there has been parity error during transmission. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_ERROR_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR (cont.)**

Bits	Name	Description
7	TX_READY	This bit, when set(1), indicates that: 1. TX FIFO is empty. 2. The value written in NO_CHARS_FOR_TX register equals the number of bytes written to the TX FIFO since the register was updated. NOTE: There may be characters in the unpack buffer or in the shift register. Cleared by issuing GENERAL_COMMAND = RESET_TX_READY_IRQ (UART_DM_CR register)
6	CURRENT_CTS	This bit indicates the current state of the CTS input, high ('1') meaning peer is not ready for data reception, low ('0') - peer is ready. The toggling of this bit does not generate an interrupt.
5	DELTA_CTS	This bit, when set (1), indicates that the CTS input has changed states. Cleared by issuing CHANNEL_COMMAND = RESET_CTS_N (UART_DM_CR register).
4	RXLEV	This bit is set when a word is loaded into the receive FIFO that brings the total number of words in the FIFO above the programmed watermark level in the FIFO watermark register (UART_DM_RFWR). This bit is cleared after enough words have been read to bring the level equal to or below the programmed watermark level.
3	RXSTALE	This bit indicates that an RX-transfer has finished. Cleared by issuing CHANNEL_COMMAND = RESET_STALE_IRQ (UART_DM_CR register).
2	RXBREAK_CHANGE	This bit is set (1) if the start or end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_CHANGE_IRQ (UART_DM_CR register). A detected break is defined as an all-zeros character with an invalid stop bit AND LOW PARITY. The end of a break is defined as a mark condition (1) at least 1/2 bit width. A detected break occupies one position in the Rx FIFO.
1	RXHUNT	This bit is set (1) when an incoming character matches the value programmed into the UART_DM_HCR register. Cleared by issuing CHANNEL_COMMAND = RESET_ERROR_STATUS (UART_DM_CR register).
0	TXLEV	This bit is set (1) when a word which is transferred from the transmit FIFO to the transmit shift register brings the total number of words in the FIFO below or equal to the programmed watermark level in the UART_DM_TFWR register. This bit is cleared (0) after enough words have been written to the FIFO to bring the level above the programmed watermark level. Since this bit does not have a Clear command, during interrupt handling this bit must be masked (using the matching bit in UART_DM_IMR) until writing additional data for transmission is required. After data is written, this interrupt-bit should be unmasked.

**0x0C1700B8 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IRDA register enables the IRDA function, selects a loopback, and allows inversion of RX and TX data. This register also controls the IRDA transceiver that optionally interfaces between the UART and the DP\_RX\_DATA and DP\_TX\_DATA pins.

The UART proper is a standard RX/TX configuration that converts the serial pin input and output lines to 8-bit data for the microprocessor. The configuration contains receive and transmit blocks, a control block, a bit rate generator for RX and TX, and FIFO interfaces with the microprocessor data bus. The IRDA register provides the means of switching IRDA circuits into the serial I/O lines and the IRDA function is normally switched out. The IRDA modulator/demodulator may be enabled or disabled under microprocessor control. When enabled, the RX and TX data paths have individual inversion select bits, so external true or inverted drivers may be used in the signal path.

\* The register is generated only when IRDA\_IFC\_GEN generic equals 1.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA**

Bits	Name	Description
4	MEDIUM_RATE_EN	- Set (1) for 1/4 bit-time pulse length (Medium rate) - Clear (0) for 3/16 bit-time pulse length (Low rate)
3	IRDA_LOOPBACK	- This bit loops the IRDA modulated TX line back into the IRDA RX line. The normal UART loopback mode must be off in order to see this effect.
2	INVERT_IRDA_TX	This bit inverts the polarity of the IRDA modulated signal on the DP_TX_DATA pin. - Set (1) this bit for an inverted polarity. - Clear (0) this bit for a non-inverted polarity.
1	INVERT_IRDA_RX	This bit inverts the polarity of the IRDA received signal on the DP_RX_DATA pin. - Set (1) this bit for inverted the polarity. - Clear (0) this bit for non-inverted polarity.
0	IRDA_EN	- Set (1) this bit to enable the IRDA transceiver. - Clear (0) this bit to bypass the IRDA transceiver and ignore the other bits of this register.

**0x0C1700BC PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TOTAL\_SNAP****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TOTAL\_SNAP**

Bits	Name	Description
23:0	RX_TOTAL_BYTES	<ul style="list-style-type: none"> <li>•RX_TOTAL_SNAP register holds the number of characters received since the end of last Rx transfer. It is updated at the end of an Rx transfer.</li> <li>Rx transfer ends when one of the conditions is met: <ul style="list-style-type: none"> <li>- The number of characters which were received since the end of the previous transfer equals the value which was written to DMRX at 'Transfer initialization'.</li> <li>- A stale event occurred (flush operation already performed if was needed)</li> </ul> </li> </ul>

**0x0C1700C0 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_WWT\_TIMEOUT****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_WWT\_TIMEOUT**

Bits	Name	Description
25	WWT_CYCLEREENABLE	When 1, a new received character will re-load the WWT counter instead of disabling it
24:0	WWT_TIMEOUT	timeout value for WWT mechanism. Minimal value should be 10 , otherwise the WWT_IRQ will be received even if two characters will follow each other

**0x0C1700C4 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL**

Bits	Name	Description
23	UART_IRDA_CLK_CGC_OPEN	Forces UART IRDA-clock (uart_tx_clk) CGC to open when set.
22	UART_SIM_CLK_CGC_OPEN	Forces UART SIM-clock (uart_tx_clk) CGC to open when set.
21	UART_RX_CLK_CGC_OPEN	Forces UART RX-clock (uart_tx_clk) CGC to open when set.
20	UART_TX_CLK_CGC_OPEN	Forces UART TX-clock (uart_tx_clk) CGC to open when set.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL (cont.)**

Bits	Name	Description
14	AHB_RX_BAM_CLK_CGC_OPEN	Forces AHB RX-BAM-clock (hrx_bam_clk) CGC to open when set.
13	AHB_TX_BAM_CLK_CGC_OPEN	Forces AHB TX-BAM-clock (htx_bam_clk) CGC to open when set.
10	AHB_RX_CLK_CGC_OPEN	Forces AHB RX-clock (hrx_clk) CGC to open when set.
9	AHB_TX_CLK_CGC_OPEN	Forces AHB TX-clock (htx_clk) CGC to open when set.
8	AHB_WR_CLK_CGC_OPEN	Forces AHB write-clock (hwr_clk) CGC to open when set.
5	RX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to RX-DISABLE command/RX-RESET command only when set.
4	TX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to TX-DISABLE command/TX-RESET command only when set.
0	AHB_CLK_CGC_CLOSE	Setting this bit to '1' will completely gate AHB-clock to the core. Only allowed register acces during the duration this bit is set is the access to reset this bit. All other register accesses are not allowed and might result in errors. When set to '0', AHB clock is not gated at the root of the core and can be gated by the other gating features.

**0x0C1700C8 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BCR**

Bits	Name	Description
6	IGNORE_CR_PROT_VIOL	Relevant only when command protection is disabled. If during on-going command a new command is issued the new command is disregarded. If the bit is disabled legacy behavior is used
5	RX_DMRX_1BYTE_RES_EN	Allows DMRX values other than a multiple of 16 if set.
4	RX_STALE_IRQ_DMRX_EQUAL	If set, Stale irq will be asserted when (DMRX==valid_char_count) and stale is not enabled.
2	RX_DMRX_LOW_EN	DMRX lower than valid_char_cnt logic is enabled if bit is set
1	STALE_IRQ_EMPTY	Stale interrupt when RX-FIFO is empty will fire if bit is set. This relates to the issue that was when state when SW reads all the characters from the fifo and timeout expired. In that case without this bit enable interrupt will not be fired
0	TX_BREAK_DISABLE	TX-pin value when transmitter disabled while in break state is '0' if set, else '1'

**0x0C1700CC PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL**

Bits	Name	Description
2	RX_DMRX_CYCLIC_EN	If set - Auto-Re-Activated RX-transfer will be re-activated with a DMRX value from the DMRX register, making a DMRX-transfer-end-event possible. Otherwise, Auto-Re-Activated RX-transfer cannot end with a DMRX event unless its value is written to the DMRX register throughout the transfer.
1	RX_TRANS_AUTO_RE_ACTIVE	RX-transfer will be automatically re-activated after last data of previous transfer was read - if set. Otherwise, transfers are only activated by writing to DMRX register.
0	RX_STALE_AUTO_RE_EN	RX-Stale automatic re-enable - on if set.

**0x0C1700D4 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS**

Bits	Name	Description
29:28	TX_COMP_FSM	Value of TX-COMP FSM state : '00' for empty, '01' non-empty, '10' for empty-check, '11' for empty-check-fail.
26:24	RX_PACK_FSM	Value of RX-packing FSM state : '000' for idle, '001' for fifo_write, '010' for wait_ack, '011' for flush_check, '100' for flushing, '101' for flush_done.
21:20	RX_TRANS_FSM	Value of RX-transfer FSM state : '00' for inactive, '01' for active, '10' for active_dmrx, '11' for active_flush.
18:16	TX_TRANS_FSM	Value of TX-transfer FSM state : '000' for idle, '001' for bam_idle, '010' for active, '011' for inactive, '100' for last_trap, '101' for nwd.
14:12	RX_PRO_TRANS_END_FSM	Value of RX_PRO_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char_dphase, '101' for mess_only_dphase.
10:8	RX_PRO_ACTIVE_FSM	Value of RX_PRO_ACTIVE FSM state : '000' for disabled, '001' for isactive, '010' for wait_check_1, '011' for wait_check_1, '100' for pf_pending, '101' for wait_update_1, '110' for wait_update_2, '111' for inactive?

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS (cont.)**

Bits	Name	Description
6:4	TX_CON_TRANS_END_FS_M	Value of TX_CON_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char dphase, '101' for mess_only_dphase, '110' for normal_dphase.
0	RX_TRANSFER_ACTIVE	1 is RX transfer is currently active, 0 if inactive.

**0x0C1700DC PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_GENERICS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_GENERICS**

Bits	Name	Description
7	GENERIC_BAM_IFC	Value of 'BAM_IFC_GEN' generic.
6	GENERIC_DM_IFC	Value of 'DM_IFC_GEN' generic.
5	GENERIC_IRDA_IFC	Value of 'IRDA_IFC_GEN' generic
4	GENERIC_SIM_GLUE	Value of 'SIM_GLUE_GEN' generic
3:0	GENERIC_RAM_ADDR_WIDHTH	Value of 'RAM_ADDR_WIDTH' generic.

**0x0C170100 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

In order to support bursts of 4, 0x70-0x7C address space is reserved for the UART\_DM TX FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF**

Bits	Name	Description
31:0	UART_TF	The UART_TF register is the UART transmit FIFO register. This register is used to access the channel transmit FIFO. If the FIFO is full at the time of writing, the characters are lost and an interrupt is generated.

**0x0C170104 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_2****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_2**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170108 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_3****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_3**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C17010C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_4****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_4**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170110 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_5****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_5**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170114 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_6****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_6**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170118 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_7****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_7**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C17011C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_8****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_8**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170120 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_9****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_9**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170124 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_10****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_10**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170128 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_11****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_11**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C17012C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_12****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_12**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170130 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_13****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_13**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170134 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_14****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_14**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170138 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_15****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_15**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C17013C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_16****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_16**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C170140 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RF register is the UART receive FIFO register, which is used to access the channel receive FIFO. In order to support bursts of 16, 0x140-0x17C address space is reserved for the UART\_DM\_RX FIFO.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF**

Bits	Name	Description
31:0	UART_RF	This register returns the next word in the receive FIFO. After the read is completed, the FIFO read pointer is updated to make the next word available.

**0x0C170144 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_2****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_2**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170148 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_3****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_3**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C17014C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_4****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_4**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170150 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_5****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_5**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170154 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_6****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_6**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170158 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_7****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_7**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C17015C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_8****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_8**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170160 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_9**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_9**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170164 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_10**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_10**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170168 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_11**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_11**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C17016C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_12**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_12**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170170 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_13**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_13**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170174 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_14**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_14**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170178 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_15**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_15**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C17017C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_16**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_16**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C170180 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CFG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00001703

Configuration register for UIM-controller (linked with this UART\_DM). Reset value relevant to UIM-controller HW-reset and not a UART\_DM HW-reset. Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CFG**

Bits	Name	Description
15	BATT_ALARM_QUICK_DRO_P_EN	When set, deactivation sequence resulted from battery-alarm will drop all UIM-lines immediately. Otherwise, lines will be dropped gradually.
14		
13	SW_RESET	When set, UIM-controller will be at reset state.
12	MODE18	Configures PAD's working reference voltage (set to '0' for 3V, '1' for 1.8V).
10	PMIC_ALARM_EN	Enables messaging on the alarm-line to notify PMIC to drop card's voltage supply. Set to enable messaging, clear to disable.
9	BATT_ALARM_TRIGGER_EN	Enables triggering of deactivation sequence when battery-alarm detected. Set to enable triggering, clear to disable.
8	UIM_RMV_TRIGGER_EN	Enables triggering of deactivation sequence when card-removal detected. Set to enable triggering, clear to disable.
6	UIM_CARD_EVENTS_ENABLE	When cleared, UIM-controller will not react to any card-event (insertion/removal). When set, UIM-controller reacts to these events and considered active. This bit should be set only after/during UIM_PRESENT_POLARITY bit is configured so UIM-controller will not interpret card-events incorrectly due to a wrong polarity configuration (the reset-value of the polarity might not match the actual one).
5	UIM_PRESENT_POLARITY	Determines value interpretation of uim_present signal. Setting this bit to '1' indicates that a uim-card present in the uim-slot will return the value '1' on uim_present line & '0' when a card is not present. Setting this bit to '0' indicates that a uim-card present in the uim-slot will return the value '0' on uim_present line & '1' when a card is not present.
4:0	EVENT_DEBOUNCE_TIME	Configures debounce-time of card-removal/insertion events (in sleep-clk cycles).

**0x0C170184 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CMD****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

Command register for UIM-controller (linked with this UART\_DM). Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CMD**

Bits	Name	Description
1	RECOVER_FROM_HW_DEACTIVATION	Writing '1' to this bit will cause the UIM-io-ctrl to exit override state and reflect UIM signals from UART_DM to the UIM card. Setting both bits of this register to '1' will cause neither of the commands to be executed.
0	INITIATE_HW_DEACTIVATION	Initiates a HW deactivation sequence when writing '1' to this bit. This executes the sequence on the uim_data, uim_clk & uim_rst_n lines and alerts the PMIC to cut the power-supply to the card afterwards. Setting both bits of this register to '1' will cause neither of the commands to be executed.

**0x0C170188 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000002**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS**

Bits	Name	Description
2	UIM_IO_WRITE_IN_PROGRESS	Asserted after UIM-write (UIM_CFG/UIM_CMD register write) while write is being propagated into the UIM-IO-controller. An additional read/write to UIM_CFG/UIM_CMD register should be done while high. De-asserted when write finishes propagating.
1	UIM_DEACTIVATION_STATUS	Indicates whether a HW deactivation sequence was performed and sequence recovery hasn't been executed yet. This bit is set when the sequence is initiated either by a HW trigger (batt_alarm, card-removal) or a SW trigger ('initiate_shutdown_sequence' bit in CMD register). Cleared after executing a 'recover_from_hw_deactivation' command.
0	CARD_PRESENCE	Indication on card presence in slot. High when card present, low otherwise.

**0x0C17018C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Asserted when UIM-write had finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
3	HW_SEQUENCE_FINISH	Asserted when UIM-controller's HW-deactivation sequence has been initiated and finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
2	BATT_ALARM	Set to high when battery-alarm indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
1	UIM_CARD_INSERTION	Set to high when card insertion indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
0	UIM_CARD_REMOVAL	Set to high when card removal indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.

**0x0C170190 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	See UART_DM_UIM_IRQ_ISR for field descriptions.
3	HW_SEQUENCE_FINISH	See UART_DM_UIM_IRQ_ISR for field descriptions.
2	BATT_ALARM	See UART_DM_UIM_IRQ_ISR for field descriptions.
1	UIM_CARD_INSERTION	See UART_DM_UIM_IRQ_ISR for field descriptions.
0	UIM_CARD_REMOVAL	See UART_DM_UIM_IRQ_ISR for field descriptions.

**0x0C170194 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
3	HW_SEQUENCE_FINISH	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
2	BATT_ALARM	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
1	UIM_CARD_INSERTION	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
0	UIM_CARD_REMOVAL	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.

**0x0C170198 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
3	HW_SEQUENCE_FINISH	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
2	BATT_ALARM	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
1	UIM_CARD_INSERTION	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
0	UIM_CARD_REMOVAL	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.

**0x0C17019C PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1701A0 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1700E0 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

New added interrupt clear register

**PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR\_CLR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Clears NO_FINISH_CMD_VIOL bit in UART_DM_ISR

**0x0C175000 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- The contents of this register should only be changed when in the RESET\_STATE.
- The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	qup_data_in is not used and the value is a 'don't care'. The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	qup_data_out is not used and is held at the value zero. The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup will not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C175004 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C175008 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.

- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.
- c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

### PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_IO\_MODES

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_IO\_MODES (cont.)**

Bits	Name	Type	Description
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C17500C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C175014 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is scheduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Cancelling an i2c transfer requires using version 2 tags.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL (cont.)**

Bits	Name	Type	Description
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C175018 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL (cont.)**

Bits	Name	Description
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPT Y	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C17501C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C175020 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER_R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E_N	If set, enables input over run error generation.

**0x0C175028 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C175100 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C175104 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C17510C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C175110+ PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** W  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C175150 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C175154 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_WRITE\_CNT\_CURRENT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C175200 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

**0x0C175204 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_INPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

**0x0C175208 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

**0x0C17520C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_READ\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

**0x0C175214 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C175218+ PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc, c=[0..15]****0x4\*c****Type:** R**Clock:** crif\_clk**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C175300 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_CONFIG (cont.)**

Bits	Name	Type	Description
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C175304 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_IO\_CONTROL (cont.)**

Bits	Name	Description
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C175308 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

**0x0C17530C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C175310 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C175314 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C175318 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C17531C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C175330 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS (cont.)**

Bits	Name	Description
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C175334 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C175338 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_CFG (cont.)**

Bits	Name	Description
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPi Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N,unused bits>. Default format <unused bits,N>. Unused bits may be garbage from previous word in case N!=32

**0x0C175400 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clkok edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL (cont.)**

Bits	Name	Description
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: $I2C\_FS\_CLK = I2C\_CLK/(2*(FS\_DIVIDER\_VALUE+3))$ When HIGH_TIME_DIVIDER_VALUE!=0: $I2C\_FS\_CLK = I2C\_CLK/(FS\_DIVIDER\_VALUE+HIGH\_TIME\_DIVIDER\_VALUE+6)$

**0x0C175404 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
7:6	FAILED	RW	<p>Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example:</p> <p>Value 00: Byte n transmitted successfully, byte n+1 to begin transmission</p> <p>Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example)</p> <p>Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur)</p> <p>Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)</p>
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C175408 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C17540C PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the i2c mincore is idle. When in doubt SW can use SW_RESET to reset the mincore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C175410 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C175414 PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C176000 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- a. The contents of this register should only be changed when in the RESET\_STATE.
- b. The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	<p>value: 0000 null core</p> <p>value: 0001 SPI core</p> <p>value: 0010 I2C master controller</p> <p>value: 0011 reserved</p> <p>value: 0100 SPI Slave</p> <p>value: 0101 reserved</p> <p>value: 0110 reserved</p> <p>value: 0111 reserved</p>
7	NO_INPUT	qup_data_in is not used and the value is a 'don't care'. The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_CONFIG (cont.)**

Bits	Name	Description
6	NO_OUTPUT	qup_data_out is not used and is held at the value zero. The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup will not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C176004 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.
6	I2C_FLUSH	RW	Flushing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C176008 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.
- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.

c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_IO\_MODES**

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.

**PERIPH\_SS\_BLSP1\_BLSP\_QUPI\_QUPI\_IO\_MODES (cont.)**

Bits	Name	Type	Description
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C17600C PERIPH\_SS\_BLSP1\_BLSP\_QUPI\_QUPI\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the

AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

#### **0x0C176014 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_TRANSFER\_CANCEL**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is scheduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

#### **0x0C176018 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x000000C0

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL (cont.)**

Bits	Name	Description
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C17601C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER_R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C176020 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER_R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E_N	If set, enables input over run error generation.

**0x0C176028 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C176100 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C176104 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C17610C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C176110+ PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc, c=[0..15]  
0x4\*c****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C176150 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_WRITE\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

#### **0x0C176154 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_WRITE\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

#### **0x0C176200 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

**0x0C176204 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_INPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

**0x0C176208 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

**0x0C17620C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_READ\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

**0x0C176214 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C176218+ PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C176300 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_CONFIG**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C176304 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C176308 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

#### **0x0C17630C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

#### **PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C176310 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C176314 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C176318 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C17631C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C176330 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C176334 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C176338 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>. Unused bits may be garbage from previous word in case N!=32

**0x0C176400 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns ≈ MAX(Tsu;DAT)) the clock edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C176404 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C176408 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C17640C PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C176410 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C176414 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C177000 PERIPH\_SS\_BLSP1\_BLSP\_QUPI2\_QUPI\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- a. The contents of this register should only be changed when in the RESET\_STATE.
- b. The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP1\_BLSP\_QUPI2\_QUPI\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	qup_data_in is not used and the value is a 'don't care'. The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	qup_data_out is not used and is held at the value zero. The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup will not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C177004 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C177008 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.

- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.
- c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

### PERIPH\_SS\_BLSP1\_BLSP\_QUPI\_QUPI\_IO\_MODES

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_IO\_MODES (cont.)**

Bits	Name	Type	Description
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C17700C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C177014 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is scheduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_TRANSFER\_CANCEL (cont.)**

Bits	Name	Type	Description
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C177018 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgment is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgment, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes, recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes, recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL (cont.)**

Bits	Name	Description
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPT Y	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C17701C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C177020 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER_R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E_N	If set, enables input over run error generation.

**0x0C177028 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C177100 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C177104 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C17710C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C177110+ PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** W  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C177150 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C177154 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_CNT\_CURRENT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C177200 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

**0x0C177204 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_INPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

**0x0C177208 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

**0x0C17720C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_READ\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

**0x0C177214 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C177218+ PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc, c=[0..15]****0x4\*c****Type:** R**Clock:** crif\_clk**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C177300 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C177304 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_IO\_CONTROL (cont.)**

Bits	Name	Description
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_Polarity. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C177308 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

**0x0C17730C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C177310 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C177314 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C177318 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C17731C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C177330 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C177334 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN (cont.)**

Bits	Name	Description
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C177338 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits,N>. Unused bits may be garbage from previous word in case N!=32

**0x0C177400 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clcok edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C177404 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C177408 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C17740C PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C177410 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C177414 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C178000 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- The contents of this register should only be changed when in the RESET\_STATE.
- The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	qup_data_in is not used and the value is a 'don't care'. The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	qup_data_out is not used and is held at the value zero. The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup will not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C178004 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C178008 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.

- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.
- c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

### PERIPH\_SS\_BLSP1\_BLSP\_QUPI3\_QUPI\_MODES

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_IO\_MODES (cont.)**

Bits	Name	Type	Description
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C17800C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C178014 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is scheduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Cancelling an i2c transfer requires using version 2 tags.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_TRANSFER\_CANCEL (cont.)**

Bits	Name	Type	Description
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C178018 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000C0**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL (cont.)**

Bits	Name	Description
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C17801C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ERR	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C178020 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER_R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E_N	If set, enables input over run error generation.

**0x0C178028 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C178100 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C178104 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C17810C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C178110+ PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** W  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C178150 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C178154 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_CNT\_CURRENT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C178200 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

**0x0C178204 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_INPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

**0x0C178208 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_READ\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

**0x0C17820C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_READ\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

**0x0C178214 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C178218+ PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFOC, c=[0..15]****0x4\*c****Type:** R**Clock:** crif\_clk**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFOC**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C178300 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C178304 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_IO\_CONTROL (cont.)**

Bits	Name	Description
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_Polarity. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C178308 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

**0x0C17830C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C178310 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C178314 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C178318 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C17831C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C178330 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C178334 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN (cont.)**

Bits	Name	Description
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C178338 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>. Unused bits may be garbage from previous word in case N!=32

**0x0C178400 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clcok edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C178404 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C178408 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C17840C PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C178410 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C178414 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C184000 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CTRL****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00020000

BAM Control register allows global controls for the BAM.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CTRL**

Bits	Name	Description
20	BAM_MESS_ONLY_CANCEL_WB	<p>When active (1) - BAM will DROP former descriptor write-backing and Interrupt firing when former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT and size 0 arrived (messaging only).</p> <p>When legacy mode is used (0) - This said a case then the former descriptor has been closed (accumulated=acknowledged) and ack_on_sucess with EOT and size 0 arrived (messaging only). In this case the FIFO-pointer has been already advanced before the ack_on_sucess arrives and therefore we do write-back to former descriptor and fire interrupt.</p> <p>1'b1 - Drop WB and Interrupt 1'b0 - legacy.</p>
19	CACHE_MISS_ERR_RESP_EN	<p>When set to '1', upon local ahb access results with cache miss, the bam_ndp will not stall the bus, and finish the access with error response.</p> <p>This bit is relevant for BAM_NDP only.</p>
18:17	LOCAL_CLK_GATING	<p>These Bits enables power saving by using a local clock gating cell.</p> <p>Bit 17: 1'b1 - CGC is on, so that the clock is controled by the HW - this is the reset value. 1'b0 - CGC is off so that the clock is free runing</p> <p>Bit 18: Reserved</p>
16	IBC_DISABLE	<p>This Bit enables power saving by disabling the inactivity clock timer/counter. This is useful when:</p> <p>1) The BAM is Enabled but idling for long periods. 2) The BAM is Enabled and running but the inactivity timers/counters are not required by the SW for operating the BAM.</p> <p>When the BAM is Disabled, it automatically shuts down those timers to save power.</p> <p>1'b1 - Enable Power Saving Mode 1'b0 - Disable Power Saving Mode</p>

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CTRL (cont.)**

Bits	Name	Description
15	BAM_CACHED_DESC_STORE	<p>This Bit enables storage of a cached Descriptor into RAM when doing Context Switch and loading it back when returning to this pipe. This is relevant for System Modes Only.</p> <p>BAM to BAM Consumer mode doesn't have this option.</p> <p>Enabling this makes the Context Switch process slightly longer but reduces AHB Descriptor Fetch time later on. It also reduces the number of descriptor fetches done by the BAM if BAM is treating more than 2 pipes.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p> <p>Available in BAM only</p>
14:13	BAM_DESC_CACHE_SEL	<p>This Selector chooses which Descriptor Caching mode will be used. This is relevant for System Producer or Consumer modes only.</p> <p>2'b00 - Cache when current descriptor has less than 64 bytes left. 2'b01 - Cache when current descriptor has less than 128 bytes left. 2'b10 - Immediate Auto cache - cache will happen whenever there are no descriptors cached.</p> <p>This might cause unnecessary descriptors reads when switching between more than 2 pipes.</p> <p>It is the most effective when each pipe processes more bytes than there are in an average descriptor (small descriptors, big block size), or when working with 2 or less pipes.</p> <p>2'b11 - Cache when descriptor is needed only. This results in 1 descriptor present in cache at most.</p> <p>Available in BAM only</p>
11:5	RESERVED	RESERVED
4	BAM_EN_ACCUM	<p>When Enabled, BAM will accumulate data written by the peripheral in Direct Mode into INCR8 Bursts.</p> <p>This is an optimization feature for Producer Direct Mode cases.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p> <p>Available in BAM only</p>
1	BAM_EN	<p>After reset the BAM wakes up in Disabled Mode.</p> <p>While Disabled, BAM operates in Legacy mode (all the transfers from the peripheral go around the BAM as if it doesn't exist)</p> <p>Software Enables this bit to allow BAM operation.</p> <p>This bit doesn't affect the PIPE enable bit in BAM_P_CTRL register.</p> <p>The only possibility to Disable the BAM after it has been enabled is by Hardware or Software reset.</p> <p>1'b1 - Enabled 1'b0 - Disabled</p>

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CTRL (cont.)**

Bits	Name	Description
0	BAM_SW_RST	This will reset the BAM & all Pipes. Software may use this to reset the BAM. As long as the bit remains high the BAM remains in reset state. Software should clear this bit 1'b1 - Reset state 1'b0 - Normal state

**0x0C184008 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000001

This Register holds a threshold value for the counter summing the Size of the Descriptors Provided. This value is Global for all pipes but it is relevant for System Producer BAM mode and Both Consumer modes, where Descriptors summing takes place in order to provide the information to the peripheral via pipe bytes free (producer) or pipe bytes available interfaces.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD**

Bits	Name	Description
15:0	CNT_TRSHLD	Threshold value. The maximum allowed value is 32kByte. The minimum allowed value is 1 (value 0 is not allowed). Available in BAM only

**0x0C184014 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_STTS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

Interrupt status register describing the reason for the ongoing interrupt. This is an informative only register and therefore read only.

Clearing the interrupt bits is done by writing to BAM\_IRQ\_CLR register.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_STTS**

Bits	Name	Description
4	BAM_TIMER_IRQ	Inactivity timer has reached its threshold value Interrupt

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_STTS (cont.)**

Bits	Name	Description
3	BAM_EMPTY_IRQ	The interrupt indicates the BAMs buffer is empty. This bit can become high only when BAM_DATA_ERASE
2	BAM_ERROR_IRQ	This Indicates a Fatal Error has happened in the BAM. Currently this might happen due to an UnSuccessful Pipe Reset operation.
1	BAM_HRESP_ERR_IRQ	This interrupt Indicates that an Erroneous HResponse has been received by the AHB Master. Additional Information about the error may be read at BAM_AHB_MASTER_ERR_* Registers. Clearing this interrupt also clears the BAM_AHB_MASTER_ERR_* Registers. Available in BAM only

**0x0C184018 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_CLR****Type:** W**Clock:** bam\_clk**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_CLR**

Bits	Name	Description
4	BAM_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
3	BAM_EMPTY_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM-Lite only
2	BAM_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
1	BAM_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged Available in BAM only

**0x0C18401C PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_EN****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

Enable bits for the Interrupts in the BAM Interrupt Status register.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_EN**

Bits	Name	Description
4	BAM_TIMER_EN	Enables Inactivity timer interrupt by writing this bit 1'b1 - Enable 1'b0 - Disable
3	BAM_EMPTY_EN	1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
2	BAM_ERROR_EN	1'b1 - Enable 1'b0 - Disable
1	BAM_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable Available in BAM only

**0x0C184040 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This register counts the idle time of the BAM (all the pipes are in idle state).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER**

Bits	Name	Description
15:0	TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0x0C184044 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER\_CTRL****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The register can be written and read only when INACTIVITY\_TIMERS\_SUPPORTED generic equals to 1.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER\_CTRL**

Bits	Name	Description
31	TIMER_RST	To begin using the inactivity timer the SW should write the threshold and write 1 to this bit. writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the TIMER_THRESOLD value 1'b1 - Active 1'b0 - Reset
30	TIMER_RUN	Will be used along with TIMER_MODE Not implemented at this time. Set to zero(0)
29	TIMER_MODE	Not implemented at this time. Set to zero(0)
15:0	TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

#### **0x0C184084 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CNFG\_BITS\_2**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x0000000F

This Register holds the BAM configuration bits. It is highly recommended to follow the directions for each bit and set it accordingly.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CNFG\_BITS\_2**

Bits	Name	Description
3	SUP_GRP_LOCKER_RST_SUPPORT	Disable/Enable in BAM-NDP/BAM-Lite only.
2	ACTIVE_PIPE_RST_SUPPORT	Supporting pipe-reset when the pipe is not necessarily quiet. Disable/Enable in BAM-NDP/BAM-Lite only.
1	NO_SW_OFFSET_REVERT_BACK	When doing Write-back do not revert back the BAM_P_SW_OFSTS pointer.
0	CNFG_NO_ACCEPT_AT_FIFO_FULL	Hold the back pressure of accept_ack_on_success in a case of fifo is going to be full. The back-pressure will be held until after the indication of descriptor-fifo-full (pipe_bytes_ctrl[0]) is asserted (CR701084). Disable/Enable in BAM-NDP only.

**0x0C185000 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_REVISION****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds a hard coded revision number of the BAM RTL. This corresponds to the BAM target number in the IMS. Also additional BAM settings are reflected here.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_REVISION**

Bits	Name	Description
31:24	INACTIV_TMR_BASE	This indicates the width of the inactivity timers base counter.
23	CMD_DESC_EN	This indicates BAM has Command Descriptor feature enabled in HW.
22:21	DESC_CACHE_DEPTH	This field indicates the Per-Pipe-Descriptor-Cache. could be 1-4: 2'b00 - Descriptor Cache Depth of 1. 2'b01 - Descriptor Cache Depth of 2. 2'b10 - Descriptor Cache Depth of 3. 2'b11 - Descriptor Cache Depth of 4. Valid only for BAM_NDP.
20	NUM_INACTIV_TMRS	This bit is checked only if INACTIV_TMRS_EXST = 1. This bit indicates how many inactivity timers are implemented according to value of INACTIVITY_TIMERS_SUPPORTED generic: 1'b1 - One timer for global BAM activity (INACTIVITY_TIMERS_SUPPORTED = 1) 1'b0 - MAX_PIPES timers for each pipe activity (INACTIVITY_TIMERS_SUPPORTED = 2)
19	INACTIV_TMRS_EXST	This indicates BAM has inactivity timers. 1'b1 - Timers available (INACTIVITY_TIMERS_SUPPORTED = 1 or 2) 1'b0 - No timers available (INACTIVITY_TIMERS_SUPPORTED = 0)
18	HIGH_FREQUENCY_BAM	This indicates BAM has Double Sampling on the Ack On Success interface. 1'b1 - Ack On Success double sampled 1'b0 - No double sampling of Ack On Success
17	BAM_HAS_NO_BYPASS	This field indicates BAM has no bypass on the AHB Data bus, from the Peripheral AHB Master to the fabric. When Bypass is available, it is the default routing when BAM is disabled. 1'b1 - No Bypass 1'b0 - Bypass Exists

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_REVISION (cont.)**

Bits	Name	Description
16	SECURED	This field indicates BAM/BAM-Lite has security support (APU inside) 1'b1 - Secured 1'b0 - Not Secured
15	USE_VMIDMT	This field indicates BAM has VMIDMT supported in HW.
14	AXI_ACTIVE	This field indicates BAM_NDP uses internal AXI bridge on the master port.
13:12	CE_BUFFER_SIZE	This field indicates the size (in Data words) of the buffer which stores the command elements. Each command element includes 4 words. 2'b00 - 4 Words (one command element). 2'b10 - 8 Words (2 command elements). 2'b11 - 16 Words (4 command elements). Valid Only if CMD_DESC_EN = 1'b1.
11:8	NUM_EES	This indicates the Number of Interrupt Lines (Execution Environments) supported by the BAM. When 1 EE is supported, only the BAM_IRQ_SRCS*_EE0 registers exist. When 4 EEs are supported, all BAM_IRQ_SRCS*_EEn registers exist for n=[0..3].

**0x0C185008 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_NUM\_PIPES****Type:** R**Clock:** bam\_clk**Reset State:** Undefined

This Register holds the hard coded number of pipes in the BAM. Since each BAM may have a different number of pipes instantiated in HW, this register allows the SW to know this parameter. The reset state value doesn't have any meaning in this register since it is a constant.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_NUM\_PIPES**

Bits	Name	Description
31:24	BAM_NON_PIPE_GRP	This field contains the number of BAM Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM
23:16	PERIPH_NON_PIPE_GRP	This field contains the number of Peripheral's Non Pipe security groups (APU Entries). Zeroes when Non-Secured BAM

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_NUM\_PIPES (cont.)**

Bits	Name	Description
15:14	BAM_DATA_ADDR_BUS_WI_DTH	This field indicates BAM data bus address width: 2'b00 - 32bit address width 2'b01 - 36bit address width 2'b1x - Reserved for future widening
7:0	BAM_NUM_PIPES	This field contains the number of pipes available in this BAM

**0x0C185024 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_CTRLS****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_CTRLS**

Bits	Name	Description
22:18	BAM_ERR_HVMID	HVMID
17	BAM_ERR_DIRECT_MODE	DIRECT MODE
16:12	BAM_ERR_HCID	HCID
11:8	BAM_ERR_HPROT	HPROT
7:5	BAM_ERR_HBURST	HBURST
4:3	BAM_ERR_HSIZE	HSIZE
2	BAM_ERR_HWRITE	HWRITE
1:0	BAM_ERR_HTRANS	HTRANS

**0x0C185028 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR**

Bits	Name	Description
31:0	BAM_ERR_ADDR	HADDR

#### **0x0C18502C PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_DATA**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_DATA**

Bits	Name	Description
31:0	BAM_ERR_DATA	HDATA

#### **0x0C185100 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_LSB**

Bits	Name	Description
31:0	BAM_ERR_ADDR	32 LSB of HADDR

**0x0C185104 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB****Type:** R**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the information about the data error in the AHB Buffer during bus Access.

The value stored is the value of the first error occurred.

Clear the BAM\_HRESP\_ERR\_IRQ interrupt to clear the recorded error.

Available in BAM only

Relevant only on 36 bit address BAM.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ERR\_ADDR\_MSB**

Bits	Name	Description
3:0	BAM_ERR_ADDR	4 MSB of Address (bits 35 to 32) of HADDR

**0x0C186000 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TRUST\_REG****Type:** RW**Clock:** bam\_clk**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any SW reset. Only HW reset applies. When using a secured BAM (APU integrated inside - SECURED field in the BAM\_REVISION register) it is reset by the XPU HW Reset. When using a non-secured BAM, it is reset by the BAM HW Reset.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TRUST\_REG**

Bits	Name	Description
13	LOCK_EE_CTRL	This bit controls if the EE setting defined in the TRUST registers will be taken into account for the pipe lock grouping decoding. 1'b0 - Only P_LOCK_GROUP is checked for pipe locking. 1'b1 - Both BAM_P_EE and P_LOCK_GROUP are checked for pipe locking.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TRUST\_REG (cont.)**

Bits	Name	Description
12:8	BAM_VMid	
7	BAM_RST_BLOCK	When enabled, the BAM Global SW reset will not be available for usage. This is in order to deny Global SW reset requests by the Global security group. 1'b1 - Enable 1'b0 - Disable
2:0	BAM_EE	This Field Indicates the EE # (0-7) to which the BAM Interrupt belongs to. The pipe interrupts will fire on the EE # set at BAM_P_TRUST_REGn registers. The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register. 3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7

**0x0C187000+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn, n=[0..3]**  
**4096\*n**

Type: R

Clock: bam\_clk

Reset State: 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking this register, SW reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register has an alias - BAM\_IRQ\_SRCS register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0x0C187004+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK\_EEn, n=[0..3]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register has an alias - BAM\_IRQ\_SRCS\_MSK register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK\_EEn**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

**0x0C187008+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn, n=[0..3]  
4096\*n**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

The register shows the interrupts sources like (BAM\_IRQ\_SRCS) although the interrupts from the BAM and the pipes are not masked on the lower level. This register has an alias - BAM\_IRQ\_SRCS\_UNMASKED register.

Registers with n=[1..7] exist only when the BAM supports multiple EEs (NUM\_EES field in the BAM\_REVISION register).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

**0x0C18700C+PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn, n=[0..3]  
4096\*n**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

The register shows the pipes attributed to a specific EE, and an indication if BAM is enabled.

The reset value written above is true only if EE>0. For BAM\_PIPE\_ATTR\_EE0 the default value is: Bits [MAX\_PIPES-1:0] = 1 , Bits [31: MAX\_PIPES] = 0 .

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn**

Bits	Name	Description
31	BAM_ENABLED	1'b1 - BAM is Enabled. 1'b0 - BAM Disabled.
30:0	P_ATTR	If bit 'i' == 1'b1 - Pipe 'i' is attributed to this EE.

**0x0C187010 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS**

**Type:** R  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This Interrupt Status Register provides a single interrupt bit per each pipe in the BAM and a single bit for the BAM (not pipe related) interrupt. This register is read only and provides fast indication for the software to decide which pipe triggered the interrupt. After checking out this register, SW reads the BAM\_P\_IRQ\_STTSn register for the pipe interrupt reason and BAM\_IRQ\_STTS for the BAM interrupt reason.

This register points to the physical BAM\_IRQ\_SRCS\_EE0 register.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS**

Bits	Name	Description
31	BAM_IRQ	BAM Global interrupt
30:0	P_IRQ	An interrupt bit per each pipe in the BAM.

**0x0C187014 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

Interrupt Sources Mask register allows to mask the interrupts by their source, whether BAM or a specific pipe.

This register points to the physical BAM\_IRQ\_SRCS\_MSK\_EE0 register.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK**

Bits	Name	Description
31	BAM_IRQ_MSK	BAM Global Interrupt Mask 0x1: ENABLE_BAM_INTERRUPT (Enable BAM interrupt) 0x0: DISABLE_BAM_INTERRUPT (Disable BAM interrupt)
30:0	P_IRQ_MSK	An Interrupt Mask bit per each pipe in the BAM. 0x1: ENABLE_PIPE_INTERRUPT (Enable Pipe interrupt) 0x0: DISABLE_PIPE_INTERRUPT (Disable Pipe interrupt)

#### **0x0C187018 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

The register shows the interrupts sources like (BAM\_IRQ\_SRCS) although the interrupts from the BAM and the pipes are not masked on the lower level.

This register points to the physical BAM\_IRQ\_SRCS\_UNMASKED\_EE0 register.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED**

Bits	Name	Description
31	BAM_IRQ_UNMASKED	BAM Global Interrupt after Mask
30:0	P_IRQ_UNMASKED	An Interrupt bit per each pipe in the BAM, after Mask

#### **BAM PIPE configuration and interrupts registers**

#### **0x0C186020+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TRUST\_REGn, n=[0..11]**

**0x4\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register holds the BAM Trusted configurations. The BAM APU needs to protect this register at the highest security level.

This register cannot be reset by any SW reset. Only HW reset applies. When using a secured BAM (APU integrated inside) it is reset by the XPU HW Reset. When using a non-secured BAM, it is reset by the BAM HW Reset.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TRUST\_REGn**

Bits	Name	Description
12:8	BAM_P_VMID	
7:3	BAM_P_SUP_GROUP	<p>Super-Group is the upper division of the pipes compared to the pipe-group division.</p> <p>When any pipe locks or unlocks the BAM, all the pipes which are in a different super-group will NOT be affected. Only the pipes within the same super-group of the locker are affected.</p> <p>Usually pipes which belongs to different peripherals which share the same BAM, would be in a different super-groups</p>
2:0	BAM_P_EE	<p>This Field Indicates the EE # (0-7) to which the Pipe Interrupt belongs to. The BAM interrupt will fire to the EE # set at BAM_P_TRUST_REGn registers.</p> <p>The number of supported EEs is reflect in the NUM_EES field in the BAM_REVISION register.</p> <p>3'b000 - EE0 3'b001 - EE1 3'b010 - EE2 3'b011 - EE3 3'b100 - EE4 3'b101 - EE5 3'b110 - EE6 3'b111 - EE7</p>

**0x0C197000+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CTRLn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Control register provides various controls for the pipe.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CTRLn**

Bits	Name	Description
20:16	P_LOCK_GROUP	<p>Pipe's lock group.</p> <p>Upon a lock request on this pipe, all pipes related to different pipe group and different EE will be locked.</p>

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CTRLn (cont.)**

Bits	Name	Description
11	P_WRITE_NWD	BAM-Lite feature. Applicable to BAM2BAM producer pipes only. When this bit is set for producer B2B pipe, NWD bit (bit number 28 in the second word of descriptor) will be written with EOT into the generated descriptor for the consumer usage. This bit is not applicable in pipes belong to a Peripheral which uses messaging_only (e.g. USB2).
10:9	P_PREFETCH_LIMIT	Bam Lite feature Selects the number of bytes to be prefetched by the Bam Lite, once a peripheral requests to read data from a pipe. This feature is relevant for Consumer modes only. 2'b00 - 32 bytes at most (INCR8) 2'b01 - 16 bytes at most (INCR4) 2'b10 - 4 bytes at most (SINGLE) Available in BAM-Lite only
8:7	P_AUTO_EOB_SEL	Bam Lite feature. Selects the number of bytes to be processed before creating End Of Block indication internally by the BAM. Descriptors generated by this feature (bam to bam producer mode) will always be of the chosen size, unless an End Of Transfer, causes a smaller descriptor to be created. 2'b00 - 512 Bytes 2'b01 - 256 Bytes 2'b10 - 128 Bytes 2'b11 - 64 Bytes Available in BAM-Lite only
6	P_AUTO_EOB	BAM Lite feature. When Enabled, the BAM Lite will ignore any incoming End Of Block indications from the peripheral (if any). Instead, the BAM Lite will generate it's own End Of Block indications internally, every P_AUTO_EOB_SEL bytes. This feature is relevant for Producer Bam 2 Bam modes only. It is a must for peripherals which are not able to produce End Of Block indications. 1'b1 - Enable 1'b0 - Disable Available in BAM-Lite only
5	P_SYS_MODE	This bit sets the pipe to work in System Mode, where the Software is the second side of the pipe. 1'b1 - System mode. 1'b0 - BAM to BAM mode.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CTRLn (cont.)**

Bits	Name	Description
4	P_SYS_STRM	Streaming Enable. Relevant to System Producer BAM mode only. When enabled, BAM will not close descriptors with End Of Transfer notifications received from peripheral, but continue writing the data to the same descriptor. 1'b1 - Enable Streaming Mode 1'b0 - None-Streaming Mode. This field is NOT supported in BAM-NDP (i.e. no streaming mode in NDP).
3	P_DIRECTION	This bit deNOTE:s pipe direction. 1'b1 - The pipe can be written (producer mode) 1'b0 - The pipe can be read (consumer mode)
1	P_EN	1'b1 - Enable Pipe 1'b0 - Disable Pipe

**0xC197004+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RSTn, n=[0..11]**  
**4096\*n**

**Type:** W

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Reset register provides Pipe Reset ability. Software needs to write 1 to this register to make the Pipe Reset. Writing 0 to this register will finish the Pipe Reset. Writing zero is needed prior to Peripheral reconfiguration for the same pipe.

NOTE: Software can invoke pipe reset only after it make sure that the pipe is 'quiet'.

('quiet' - No data transaction or command-descriptor is active at the time.

BAM\_P\_RSTn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RSTn**

Bits	Name	Description
0	P_SW_RST	This resets the pipe and it's registers, (Both Flip-Flops and RAM). Software can invoke pipe reset only after it make sure that the pipe is 'quiet'. 1'b1 - Reset 1'b0 - Do Nothing

**0xC197008+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_HALTn, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Halt register Enables/Disables the Halt Sequence. This is a self-modifying register.

It also supports pipe-empty indication and force-descriptor-fifo-full.

### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_HALTN**

Bits	Name	Type	Description
4	P_FORCE_DESC_FIFO_FULL	RW	When SW asserts this bit, descriptor-fifo-full status will be shown to the peripheral on pipe_bytes_avail_ctrl[0]. This is to make the peripheral to think that there is no room for more descriptors. Available only in BAM-NDP and only for Producer BAM-to-BAM pipes.
3	P_PIPE_EMPTY	R	This bit indicates that the Descriptor-FIFO is now empty. It is a read-only bit. Available in BAM-NDP only. Else it is tied high.
2	P_LAST_DESC_ZLT	R	This bit indicates that the last created descriptor is with zero-length size. It is a read only bit and effective only for Producer BAM2BAM mode. Available in BAM-NDP only. Else it is tied high.
1	P_PROD_HALTED	RW	This bit is used to inform the Consumer about the Producer being in Halt mode now. This is a part of the Halt procedure. 1'b1 - Sets this bit to 1 1'b0 - Does Nothing This bit will be cleared by the HW. Not Available in BAM-Lite
0	P_HALT	RW	When Enabled, the Pipe will enter Halt Mode. 1'b1 - Enable Halt 1'b0 - Disable Halt For B2B Producer pipes, this bit is self modifying. It doesn't need the SW to clear it. Not Available in BAM-Lite

### **0x0C197010+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn, n=[0..11] 4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

Pipe Interrupt status register describing the reason for the ongoing interrupt. This is an informative only register and therefore read only. It also has informative bits which are not interrupts.

Clearing the interrupt bits is done by writing to P\_IRQ\_CLR register.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn**

Bits	Name	Description
7	P_HRESP_ERR_IRQ	This interrupt Indicates that an Erroneous HResponse has been received by the AHB Master. Additional Information about the error may be read at BAM_AHB_MASTER_ERR_* Registers. Clearing this interrupt also clears the BAM_AHB_MASTER_ERR_* Registers. Available in BAM only
6	P_PIPE_RST_ERROR_IRQ	Unsuccessful Pipe Reset operation. Pipe reset timer expired.
5	P_TRNSFR_END_IRQ	For producer this interrupt happens whenever the last AHB burst of a transfer happens For Consumer when ack_on_success closes a transfer
4	P_ERR_IRQ	Pipe Error interrupt indicates that an error has happened. Error reason is not yet defined. Interrupt not in use.
3	P_OUT_OF_DESC_IRQ	Out of descriptors interrupt has several meanings depending on the operation mode: Producer - no free space in the descriptors fifo for adding a descriptor Consumer - no descriptors in the descriptors fifo for processing IO Vectors Producer - No descriptors to read (and therefore no space for data to write to the pipe) IO Vectors Consumer - No descriptors to read (and therefore no data to read from pipe)
2	P_WAKE_IRQ	Wake peripheral interrupt signals the driver to wake up the peripheral (USB for example) for transmission via the current pipe. This interrupt is issued whenever an Event comes to the specific pipe.
1	P_TIMER_IRQ	Inactivity timer has reached its threshold value Interrupt
0	P_PRCSD_DESC_IRQ	This Interrupt rises when BAM finishes processing an IO Vector which has INT bit selected

**0x0C197014+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn, n=[0..11]  
4096\*n**

**Type:** W

**Clock:** bam\_clk

**Reset State:** 0x00000000

Writing to this register causes the interrupt to clear.

**BAM\_P\_IRQ\_CLRn, n=[0..30]**

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn**

Bits	Name	Description
7	P_HRESP_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
6	P_PIPE_RST_ERROR_CLR	1'b1 - Clear 1'b0 - Unchanged
5	P_TRNSFR_END_CLR	1'b1 - Clear 1'b0 - Unchanged
4	P_ERR_CLR	1'b1 - Clear 1'b0 - Unchanged
3	P_OUT_OF_DESC_CLR	1'b1 - Clear 1'b0 - Unchanged
2	P_WAKE_CLR	Clear Wake peripheral interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
1	P_TIMER_CLR	Clear Inactivity timer interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged
0	P_PRCSD_DESC_CLR	Clear Descriptor Processed interrupt by writing this bit 1'b1 - Clear 1'b0 - Unchanged

**0x0C197018+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

Enable bits for the Interrupts in the Pipe Interrupt Status register.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn**

Bits	Name	Description
7	P_HRESP_ERR_EN	1'b1 - Enable 1'b0 - Disable
6	P_PIPE_RST_ERROR_EN	1'b1 - Enable 1'b0 - Disable
5	P_TRNSFR_END_EN	1'b1 - Enable 1'b0 - Disable
4	P_ERR_EN	1'b1 - Enable 1'b0 - Disable

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn (cont.)**

Bits	Name	Description
3	P_OUT_OF_DESC_EN	1'b1 - Enable 1'b0 - Disable
2	P_WAKE_EN	Enable Wake peripheral interrupt 1'b1 - Enable 1'b0 - Disable
1	P_TIMER_EN	Enable Inactivity timer Interrupt 1'b1 - Enable 1'b0 - Disable
0	P_PRCSD_DESC_EN	Enable Descriptor Processed Interrupt 1'b1 - Enable 1'b0 - Disable

**0xC19701C+PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMERn, n=[0..11]**  
**4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register counts the idle time of the pipe.

BAM\_P\_TIMERn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMERn**

Bits	Name	Description
15:0	P_TIMER	Inactivity timer counter. 16 bits = 64k*0.1ms ~ 6 seconds

**0xC197020+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRLn, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register sets various controls and the threshold of the inactivity for interrupt triggering after long idle time.

The resolution of the pipe inactivity timers are in units set by a common counter. This common counter is controlled by a PARAMETER value of the BAM, and uses a separate clock, the inactivity\_timers\_clk. This clock can be slower than the bam\_clk. The intent of the design is to use the sleep\_clk, which is an always on 32KHz clock. This allows the bam\_clk to be turned-off, while the base counter keeps counting. The width of this counter, as well of the rollover value are set using BAM PARAMETER. These values, taking the inactivity\_timers\_clk frequency define the

pulse frequency that the base counter gives, and the threshold value can be calculated and set to interrupt after a specific time, regardless of the bam\_clk frequency, and independent of clock power save features of the bam\_clk. The timers continue counting even if the clock is turned off due to BAM dynamic hardware clock request voting.

The timer configuration is based on the inactivity\_timers\_clk period and the INACTIVITY\_TIMER\_WIDTH parameter constant. These parameters should be taken into account when setting this register. For example: for inactivity\_timer\_clk period of 1us and parameter is 3 and P\_TIMER\_THRHLD is 10 will indicate the  $2^{3*1\text{us}} * 10$  which is 80us of inactivity in a pipe before sending an interrupt.

The general formula is:  $2^{\text{INACTIVITY\_TIMER\_WIDTH}} * \text{clock\_period} * \text{P\_TIMER\_TRSHLD}$ .

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRLn**

Bits	Name	Description
31	P_TIMER_RST	To begin using the inactivity timer the SW should write the threshold and write 1 to this bit. Writing the value 0 will hold the timer reset until value 1 is written again. Then the timer will start down counting from the P_TIMER_THRESHOLD value 1'b1 - Active 1'b0 - Reset
30	P_TIMER_RUN	Will be used along with P_TIMER_MODE Not implemented at this time. Set to zero(0)
29	P_TIMER_MODE	Not implemented at this time. Set to zero(0)
15:0	P_TIMER_TRSHLD	Inactivity timer threshold for interrupt generation.

#### **0x0C197024+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn, n=[0..11] 4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register reflects the current status of the Producer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn**

Bits	Name	Description
24	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_producer. relevant for system mode only. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
20	BAM_P_TOGGLE	Pipe Bytes Free Toggle value. The toggle polarity of the publication (each publication changes its polarity).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn (cont.)**

Bits	Name	Description
19:16	BAM_P_CTRL	Pipe Bytes Free Ctrl value. The possible values are only: 0000 or 0001 (means desc-fifo-full).
15:0	BAM_P_BYTES_FREE	Pipe Bytes Free value. The accumulated bytes_free which are available for peripheral to use

**0x0C197028+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn, n=[0..11]  
4096\*n**

**Type:** R

**Clock:** bam\_clk

**Reset State:** 0x00000000

This Register reflects the current status of the Consumer Sideband interface as the peripheral sees it and some additional internal bits related to its management.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn**

Bits	Name	Description
30	BAM_P_ACCEPT_ACK_ON_SUCCESS_TOGGLE	accept_ack_on_sucess toggle. This bit is relevant for BAM_NDP only.
29:28	BAM_P_ACK_ON_SUCCES_S_CTRL	ack_on_sucess control. This bit is relevant for BAM_NDP only.
27	BAM_P_ACK_ON_SUCCES_TOGGLE	ack_on_sucess toggle. This bit is relevant for BAM/BAM_NDP only.
26	BAM_P_SB_UPDATED	sideband_inform_bytes_avail_updated_consumer. The sideband-Inform block which responsible for reading descriptor has updated the SB block (which responsible for publishing to peripheral) with all of its read descriptors.
25	BAM_P_NWD_TOGGLE	notify_when_done toggle
24	BAM_P_NWD_TOGGLE_R	notify_when_done toggle sampled
23	BAM_P_WAIT_4_ACK	waiting_for_ack_bytes_avail_r. BAM is waiting for Peripheral to acknowledge the former publish.
22	BAM_P_ACK_TOGGLE	ack_bytes_avail_toggle. The Peripheral's acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
21	BAM_P_ACK_TOGGLE_R	ack_bytes_avail_toggle_r. The Peripheral's sampled acknowledgement for the publish. This is a toggling signal (each ack will be inverted to its former).
20	BAM_P_TOGGLE	Pipe Bytes Avail Toggle value. The toggle polarity of the publication (each publication changes its polarity).

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn (cont.)**

Bits	Name	Description
19:16	BAM_P_CTRL	Pipe Bytes Avail Ctrl value. The possible values of BAM_P_CTRL field are: 0100 - Means immediate command (could be only in bam-ndp). 0001 - Means End-Of-Transaction (EOT). 0011 - means End-Of-Transaction with Notify-When-Done (NWD). 0000 - Neither Immediate-command nor EOT nor NWD.
15:0	BAM_P_BYTES_AVAIL	Pipe Bytes AVAIL value. The size of the current publish.

**0x0C197800+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register deNOTE:s the pointer Offset of the first Not Acknowledged Descriptor. Meaning all Descriptors prior to this one have been processed by the BAM. This register is only used by the Software. After receiving an interrupt, software reads this register in order to know what descriptors has been processed.

When the BAM works as Producer, Software will usually read this value after Descriptor Processed Interrupt or after the End of Transfer Interrupt.

When the BAM works as Consumer, Software will usually read this value after Descriptor Processed Interrupt.

NOTE: This is non relevant in BAM to BAM modes.

NOTE: Although being Writable, Software should never write to this register.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_SW\_OFSTS<sub>n</sub>, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SW\_OFSTS<sub>n</sub>**

Bits	Name	Description
31:16	SW_OFST_IN_DESC	Offset inside the Descriptor. This value is used by the Software only when a BAM is working as System Producer. Then the SW can read this value to know how many Bytes were written to the current descriptor if it was not closed down. This is useful for Streaming mode. This Field is NOT available in BAM-NDP
15:0	SW_DESC_OFST	Descriptor FIFO offset.

**0x0C19782C+PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDRn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe.

**0x0C197930+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR LSBn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR LSBn**

Bits	Name	Description
31:0	P_EVNT_DEST_ADDR	32 LSB of Address of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

**0x0C197934+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This register stores the Event Destination Address which is the address of BAM\_P\_EVNT\_REG in the BAM on the other side of the Pipe.

Producer BAM will send Write Events to this address.

Consumer BAM will send Read Events to this address.

This is relevant only for BAM to BAM modes. In System modes, BAM doesn't issue Write/Read events.

Relevant only on 36 bit address BAM.

BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_EVNT_DEST_ADDR	4 MSB of Address (bits 35 to 32) of the Event Register in the peer BAM (second end of the pipe). Driver will configure this when initializing the pipe. Relevant only on 36 bit address BAM.

**0x0C197818+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn, n=[0..11]  
4096\*n**

**Type:** RW  
**Clock:** bam\_clk  
**Reset State:** 0x00000000

This Register is where Read/Write events are sent to. If current BAM works as Producer, it will receive Read events into this register from the Consumer. If the Consumer is another BAM, it will write read events to this register. If the Consumer is the Software, it will prepare descriptors in the descriptor FIFO and update the DESC\_FIFO\_PEER\_OFST value in this register.

If current BAM works as Consumer, opposite behavior is assumed.

BAM\_P\_EVNT\_REGn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn**

Bits	Name	Description
31:16	P_BYTES_CONSUMED	This field is used only in the B2B mode. 15 LSB bits of this field indicate the number of bytes consumed by the Consumer. Consumer sends this value to the Producer as a part of a read event, for the Producer to know how many bytes were consumed. The MSB is virtual-event written by the SW in pipe-halt procedure. When working in System mode, this value should be set to zero. BAM to BAM Producer writes zero in this value.
15:0	P_DESC_FIFO_PEER_OFST	This field indicates the Descriptor Fifo Offset Pointer of the Peer BAM (The BAM or the Software which creates descriptors). For example when current BAM works as consumer for the current pipe, Producer BAM (or the software) updates this value to let the current pipe know about the new descriptors added in the Descriptor Fifo.

**0xC19781C+PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Address of the Descriptor Fifo beginning.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	Address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0xC197910+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn, n=[0..11]**  
**4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the LSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DESC_FIFO_ADDR	32 LSB of address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0x0C197914+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the MSB of Address of the Descriptor Fifo beginning.

Relevant only on 36 bit address BAM.

NOTE: This register is used by all modes.

BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DESC_FIFO_ADDR	4 MSB of address (bits 35 to 32) of Address of the Descriptors Fifo. This address must be 8 bytes aligned.

**0x0C197820+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

The least significant field is the Descriptor Fifo Size.

The most significant field is the Data Fifo Size.

NOTE: This Descriptor Fifo Size is used by all modes. The Data Fifo Size is non relevant in System Modes.

BAM\_P\_FIFO\_SIZESn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn**

Bits	Name	Description
31:16	P_DATA_FIFO_SIZE	Size of the Data Address Space. ALWAYS write 16'b0 to this field of P_DATA_FIFO_SIZE unless you work in BAM2BAM producer mode (also known as peer mode). NOTE: Data fifo size should not be bigger than 32KB.
15:0	P_DESC_FIFO_SIZE	Size of the Descriptors Fifo. Must be 8 byte aligned. Each descriptor is 8 bytes. Size of the descriptor fifo must contain an integer number of Descriptors. Size of descriptor fifo must be 16 bytes and above.

**0xC197824+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDRn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	Data Address Space beginning.

**0xC197920+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_LSBn**

Bits	Name	Description
31:0	P_DATA_FIFO_ADDR	32 LSB of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0x0C197924+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register stores the Beginning of the Data Address space.

This register is non relevant in System Modes.

Relevant only on 36 bit address BAM.

BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn**

Bits	Name	Description
3:0	P_DATA_FIFO_ADDR	4 MSB of Address (bits 35 to 32) of Data Address Space beginning. Relevant only on 36 bit address BAM.

**0x0C197828+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register configures the threshold value for Read/Write event generation by the BAM towards another BAM. When aBAM pushes/pops this number of bytes to/from the pipe an event to the other BAM will be generated. A counter summing up the bytes pushed/popped is reset when event is issued, thus restarting count for the threshold.

Events are also issued after End Of Transfer received from the peripheral (Producer case), non regarding whether the pushed bytes count has reached the threshold or not.

This register is non relevant in System Modes.

BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn**

Bits	Name	Description
15:0	P_TRSHLD	Threshold value. The maximum allowed value is 32kByte.

**BAM PIPE internal state registers**

**0x0C197804+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n**

Bits	Name	Description
31:16	AU_PSM_ACCUMED	PSM: The acknowledged (EOB/T) number of bytes since last Write Event.  AU: The accumulation of the descriptor sizes for the present ack_on_success event. This is never bigger than the ACKED field. Once the ACCUMED values are bigger than Event Threshold, an Event will be issued, the ACKED value will decrease by the ACCUMED size and this value will be zeroed.  This also serves as the Bytes Consumed value in the Event.
15:0	AU_ACKED	AU: The left over of the number of bytes that were received via the ack_on_success, that did not cause a descriptor closure. This value is down counting upon issuing Read Event.  PSM:

**0x0C197808+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n**

Bits	Name	Description
31	PSM_DESC_VALID	The Descriptor Valid bit provided by the Writeback state machine to Pipe state machine.
30	PSM_DESC_IRQ	The Descriptor Interrupt bit provided by the Writeback state machine to pipe state machine.
29	PSM_DESC_IRQ_DONE	The Descriptor Interrupt needed selection bit.
28:25	PSM_GENERAL_BITS	General Context Switch Bits
24:22	PSM_CONS_STATE	State of the Pipe Consumer state machine.
21:19	PSM_PROD_SYS_STATE	State of the Pipe Producer System state machine.
18:16	PSM_PROD_B2B_STATE	State of the Pipe Producer BAM to BAM state machine.
15:0	PSM_DESC_SIZE	The Descriptor Size provided by the Writeback state machine to pipe state machine.

**0xC19780C+PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..11]**  
**4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

NOTE: When using 2 pipes BAM, REVISION 0x03, this registers is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_n**

Bits	Name	Description
31:0	PSM_DESC_ADDR	The Data Address provided in the current descriptor from the Writeback state machine.

**0xC197900+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_LSBn, n=[0..11]**  
**4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_LSBn**

Bits	Name	Description
31:0	PSM_DESC_ADDR	32 LSB of The Data Address provided in the current descriptor from the Writeback state machine.

#### **0xC197904+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_MSBn, n=[0..11] 4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

Relevant only on 36 bit address BAM.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..30]

#### **PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_MSBn**

Bits	Name	Description
3:0	PSM_DESC_ADDR	4 MSB of address (bits 35 to 32) of the Data Address provided in the current descriptor from the Writeback state machine.

#### **0xC197810+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..11] 4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n**

Bits	Name	Description
31:16	PSM_DESC_OFST	The running Descriptor Fifo Offset of the Writeback state machine
15:0	PSM_SAVED_ACCUMED_SIZE	The Saved Accumulated Size from the Pipe state machine.

**0x0C197814+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n**

Bits	Name	Description
31:16	PSM_BLOCK_BYTE_CNT	The Byte count inside the current Block.
15:0	PSM_OFST_IN_DESC	The Offset inside a register. This is used by pipe state machine to compute offset inside a register after retransmission

**0x0C197830+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n, n=[0..11]  
4096\*n**

**Type:** RW

**Clock:** bam\_clk

**Reset State:** 0x00000000

This register is used on in BAM-NDP core.

NOTE: Irrelevant for BAM-Lite.

BAM\_P\_DF\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n**

Bits	Name	Description
31:16	WB_ACCUMULATED	Relevant in B2B Producer mode only. This field stores the current running accumulation of the total amount of data that has been written to the data fifo. When it goes bigger than event_threshold a write event occurs.

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n (cont.)**

Bits	Name	Description
15:0	DF_DESC_OFST	Holds the descriptor offset of the Descriptor-Fetcher block which is responsible for fetching descriptors before publishing them to the peripheral.

**0x0C197834+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n, n=[0..11]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: Irrelevant for BAM-Lite.

This is being stored by Ack Update state machine in consumer mode or by the Writeback state machine in producer mode.

This is being loaded by the Writeback state machine when retransmission happens.

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_RETR\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n**

Bits	Name	Description
31:16	RETR_DESC_OFST	Descriptor Fifo Offset for retransmission.
15:0	RETR_OFST_IN_DESC	The Offset inside the Descriptor for retransmission.

**0x0C197838+ PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n, n=[0..11]  
4096\*n**

Type: RW

Clock: bam\_clk

Reset State: 0x00000000

NOTE: When using 2 pipes BAM, REVISION 0x03, this register is Read Only.

BAM\_P\_SI\_CNTXT\_n, n=[0..30]

**PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n**

Bits	Name	Description
15:0	SI_DESC_OFST	The Descriptor Fifo Offset pointer stored by the Sideband Inform state machine.

**0x0C1AF000 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR1****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_MR1 register is the UART mode register #1. It is used, along with UART\_DM\_MR2, to configure the operational mode of the UART.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR1**

Bits	Name	Description
31:8	AUTO_RFR_LEVEL1	<p>These bits are used, along with field AUTO_RFR_LEVEL0 to program the level in the receive FIFO at which the Ready-For-Receive signal (RFR_N) toggles, if programmed to do so.</p> <p>When RX-FIFO level is greater than the value in fields AUTO_RFR_LEVEL1 and AUTO_RFR_LEVEL0 , signal will indicate Not-Ready (RFR_N high). Otherwise signal will indicate Ready RFR_N low.</p> <p>Configured value should be higher/equal than RFWR value, otherwise the watermark will never be reached.</p> <p>Value of this register is in words and can be programmed from 1 to <math>2^{\text{RAM\_ADDR\_WIDTH}}</math>.</p> <p>- Only RAM_ADDR_WIDTH +1:8 bits of this field are generated.</p>
7	RX_RDY_CTL	<p>Setting (1) this bit enables the automatic ready-for-receiving (RFR_N) control for the receiver. RFR_N is turned off, or set (1), when a valid start bit is received and the RX channel FIFO is at the level programmed in bits [31:8,5:0] of this register. When the FIFO level falls back to the programmed level, the RFR_N signal is turned on, or clear (0). When this feature is off, the RFR_N signal can be used by normal signal port bit manipulation.</p>
6	CTS_CTL	<p>When this bit is set(1), the transmitter checks the CTS_N input to determine whether to begin transmission of a new character. If CTS_N is low, the character is sent. Otherwise the transmitter continues marking until CTS_N goes low, then the next character is transmitted. A change on CTS_N during the transmission of a character has no effect on that character.</p> <p>When this bit is clear(0), the CTS_N input for the channel has no effect on the transmitter.</p>
5:0	AUTO_RFR_LEVEL0	Same description of field AUTO_RFR_LEVEL1.

**0x0C1AF004 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR2****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR2**

Bits	Name	Description
10	RFR_CTS_LOOPBACK	Enables internal loopback between RFR_N of RX channel and CTS_N of TX channel if set. For this feature to be enabled, data loopback must also be enabled (LOOPBACK bit of this register).
9	RX_ERROR_CHAR_OFF	When this bit is set, characters received with parity or framing errors get discarded and not moved to the RX channel. Otherwise they enter RX channel as any valid character.
8	RX_BREAK_ZERO_CHAR_OFF	When this bit is set, the zero character received at rx_break event get discarded and not moved to the RX channel. Otherwise it enters RX-channel as any valid character.
7	LOOPBACK	When set - data output of transmitter is fed back to the data input of the receiver, i.e. TX-pin is connected to RX-pin. Might be used with flow-control loopback (RFR_CTS_LOOPBACK bit above).
6	ERROR_MODE	This bit controls the operation of the two status register bits for the RX channel (parity or framing error and received break). <ul style="list-style-type: none"> <li>- When clear (0), the UART operates in character mode and the status bits apply only to the last character received.</li> <li>- When set (1), the UART operates in block mode and both bits are sticky, i.e. they go high once an error/break-event is detected and go low only after reset error status command is issued.</li> </ul>
5:4	BITS_PER_CHAR	These bits determine how many data bits are transmitted or received per character, not including the start, stop, and parity bits. <ul style="list-style-type: none"> <li>0x0: ENUM_5_BITS (5 bits)</li> <li>0x1: ENUM_6_BITS (6 bits)</li> <li>0x2: ENUM_7_BITS (7 bits)</li> <li>0x3: ENUM_8_BITS (8 bits)</li> </ul>
3:2	STOP_BIT_LEN	This field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 9/16, 1, 1+ 9/16, and 2 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. <ul style="list-style-type: none"> <li>0x0: ENUM_0_563 (0.563 (9/16) bit times)</li> <li>0x1: ENUM_1_000_BIT_TIME (1.000 bit time)</li> <li>0x2: ENUM_1_563 (1.563 (1+9/16) bit times)</li> <li>0x3: ENUM_2_000_BIT_TIMES (2.000 bit times)</li> </ul>

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR2 (cont.)**

Bits	Name	Description
1:0	PARITY_MODE	<p>These bits determine which parity mode is used. The user can select between odd, even, space, or no parity.</p> <p>0x0: NO_PARITY (no parity - last data-bit is followed by a stop-bit)</p> <p>0x1: ODD_PARITY (odd parity - parity-bit ensures number of ones in data-bits+parity-bit is odd)</p> <p>0x2: EVEN_PARITY (even parity - parity-bit ensures number of ones in data-bits+parity-bit is even)</p> <p>0x3: SPACE_PARITY (space parity - parity is always '0')</p>

**0x0C1AF008 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR\_SR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CSR or UART\_DM\_SR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR\_SR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CSR_SR_DEPR ECATED	

**0x0C1AF010 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR\_MISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CR or UART\_DM\_MISR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR\_MISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CR_MISR_DEP RECATED	

**0x0C1AF014 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR\_ISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IMR or UART\_DM\_ISR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR\_ISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IMR_ISR_DEPRECATE	

**0x0C1AF018 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IPR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0xFFFFFFF9F

The UART\_DM\_IPR register is the UART interrupt programming register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IPR**

Bits	Name	Description
31:7	STALE_TIMEOUT_MSB	The stale character time-out duration field contains a number from 1 to $2^{30} - 1$ . This number determines how many character times must elapse before a stale event is generated. This character time is defined as 10 times the bit rate. It is reset whenever a data bit is received. Never set this fields to 0 while RX channel is operational. Do not re-configure while an RX-transfer is active. NOTE: the discontinuity in the bit assignments.
6	SAMPLE_DATA	Currently not in use. Setting (1) to this bit enables the sample data mode. In this mode start bit is sampled along with the data of the UART character.
4:0	STALE_TIMEOUT_LSB	This bitfield is the LSbits of the STALE_TIMEOUT bitfield.

**0x0C1AF01C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_TFWR register is the UART transmit FIFO watermark register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TFWR**

Bits	Name	Description
31:0	TFW	These bits contain a number, between 0 and $2^{\text{RAM\_ADDR\_WIDTH}} - 1$ , that determines the level of the transmit (TX) FIFO at which space_avail_req signal is asserted in BAM mode or TXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of TX words in the FIFO) is less than or equal to the value in TFWR. Notice that the value is in words. - Only RAM_ADDR_WIDTH -1:0 bits are generated.

**0x0C1AF020 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RFWR register is the UART receive FIFO watermark register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RFWR**

Bits	Name	Description
31:0	RFW	These bits contain a number, between 0 and $2^{\text{RAM\_ADDR\_WIDTH}} - 1$ , that determines the level of the receive (RX) FIFO at which data_avail_req signal is asserted in BAM mode or RXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of words in the RX FIFO) is greater than the value in RFWR. Configured value should be lower/equal than AUTO_RFR_LEVEL value, otherwise the watermark will never be reached. Notice that the value is in words. - Only RAM_ADDR_WIDTH -1:0 bits are generated.

**0x0C1AF024 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_HCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Configures Hunt-Character value.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_HCR**

Bits	Name	Description
7:0	DATA	The UART_DM_HCR register is the UART hunt character register, which contains the character for which the receiver looks to assert a hunt character received interrupt.

**0x0C1AF034 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMRX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initializes RX transfer and configures maximal length in bytes of the transfer.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMRX**

Bits	Name	Description
24:0	RX_DM_CRCI_CHARS	A write to RX_DM_CRCI_CHARS initializes RX transfer in SW (FIFO) mode, with written value configuring the maximal length in bytes of the transfer. In BAM-mode, only the value held in this register impacts the data-flow and not the write-event. A write-value of 0 is not valid. Read of UART_DM_DMRX register is returns the number of characters that were received since the end of the last RX transfer.

**0x0C1AF038 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IRDA or UART\_DM\_RX\_TOTAL\_SNAP.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IRDA_RX_TOTAL_SNAP_DEPRECATED	

**0x0C1AF03C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMEN****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_DMEN register defines the DMA used by the core and the data packing mode for TX and RX channels.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMEN**

Bits	Name	Description
5	RX_SC_ENABLE	- Set (1) for work in Single-Character mode for RX channel (every character received is zero-padded into a word). - Clearing this bit requires resetting the receiver.
4	TX_SC_ENABLE	- Set (1) for work in Single-Character mode for TX channel (Only LSB byte of each word get transmitted). - Clearing this bit requires resetting the transmitter.
3	RX_BAM_ENABLE	- Set (1) this bit to enable RX BAM interface (BAM-mode). - Clear (0) this bit to disable RX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the receiver.
2	TX_BAM_ENABLE	- Set (1) this bit to enable TX BAM interface (BAM-mode). - Clear (0) this bit to disable TX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the transmitter.

**0x0C1AF040 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_NO\_CHARS\_FOR\_TX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initialization of TX transfer in SW/FIFO mode.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_NO\_CHARS\_FOR\_TX**

Bits	Name	Description
23:0	TX_TOTAL_TRANS_LEN	The total number of characters for transmission. Before writing a new value, the TX FIFO must be empty as indicated by TX_READY interrupt or after a TX-channel reset. It is used by the TX-channel in SW/FIFO mode to calculate how many characters to transmit in the last word. Any additional writes to the TX FIFO beyond TX_TOTAL_TRANS_LEN characters will be discarded. A TX_READY interrupt is triggered after TX_TOTAL_TRANS_LEN number of characters were moved from the TX FIFO to the unpacking buffer (not all may have been transmitted at that point though).

**0x0C1AF044 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BADR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

Configures FIFO division between RX-FIFO and TX-FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BADR**

Bits	Name	Description
31:2	RX_BASE_ADDR	<p>RX FIFO base address. Both FIFOs use the same RAM (<math>2^{\text{RAM\_ADDR\_WIDTH}}</math>, 32-bit entries). This register controls the division of the memory between the RX and TX FIFOs. The division must be a multiple of 4 entries for legacy reasons.</p> <p>The size of the TX FIFO equals RX_BASE_ADDR. The size of the RX FIFO is <math>2^{\text{RAM\_ADDR\_WIDTH}} - \text{RX\_BASE\_ADDR}</math>.</p> <p>* The default is RX_BASE_ADDR = <math>2^{(\text{RAM\_ADDR\_WIDTH} - 1)}</math></p> <p>* Only RAM_ADDR_WIDTH -1:2 bits are generated.</p>

**0x0C1AF04C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

TX channel status register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TXFS**

Bits	Name	Description
31:14	TX_FIFO_STATE_MSB	<p>The msb of TX_FIFO_STATE bitfield.</p> <p>- Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.</p>
13:10	TX_ASYNC_FIFO_STATE	TX asynchronous fifo status - number of characters in the TX asynchronous FIFO.
9:7	TX_BUFFER_STATE	TX unpacking buffer status - number of bytes in the unpacking buffer
6:0	TX_FIFO_STATE_LSB	TX FIFO status - number of words in the TX FIFO

**0x0C1AF050 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

RX channel status register.

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RXFS**

Bits	Name	Description
31:14	RX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	RX_ASYNC_FIFO_STATE	RX asynchronous fifo status - number of characters in the RX asynchronous FIFO.
9:7	RX_BUFFER_STATE	RX packing buffer status - number of bytes in the packing buffer
6:0	RX_FIFO_STATE_LSB	The number of entries in the RX FIFO that have at least one valid word. It is incremented each time a word is moved from the packing register to the RX FIFO. It is decremented each time a word is read from the RX FIFO.  NOTE: The Uart does not keep track of non-valid characters in each word.

#### **0x0C1AF064 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_RESET**

Type: W

Clock: WR\_CLK

Reset State: Undefined

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_RESET**

Bits	Name	Description
0	RESET	

#### **0x0C1AF06C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_VAL**

Type: R

Clock: WR\_CLK

Reset State: Undefined

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_VAL**

Bits	Name	Description
9:0	VAL	Current MISR state

**0x0C1AF070 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF or UART\_DM\_RF.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_DEPRECATED	

**0x0C1AF074 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_2\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_2 or UART\_DM\_RF\_2.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_2\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_2_DEPRECATED	

**0x0C1AF078 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_3\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_3 or UART\_DM\_RF\_3.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_3\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_3_DEPRECATED	

**0x0C1AF07C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_4\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_4 or UART\_DM\_RF\_4.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_4\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_4_DEPR ECATED	

**0x0C1AF080 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SIM\_CFG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_SIM\_CFG register is used to configure the SIM interface for the UART.

\* This register is generated only when SIM\_GLUE\_GEN generic equals 1.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SIM\_CFG**

Bits	Name	Description
17	UIM_TX_MODE	When this bit is set (1), any re-transmission signal from the UIM card will be ignored (The transmission portion of the SIM interface operates in block mode T=1). When clear (0), re-transmission will be executed upon request (the transmission portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
16	UIM_RX_MODE	When this bit is set (1), error-signal indication will not be generated by the receiver (the receive portion of the SIM interface operates in block mode T=1). When clear (0), error signal will be generated when a character error is detected (the receive portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SIM\_CFG (cont.)**

Bits	Name	Description
15:8	SIM_STOP_BIT_LEN	In UIM_IF mode, this field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 1 to 254 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. 0x1: ENUM_1_BIT_TIMES (1 bit times) 0x2: ENUM_2_BIT_TIMES (2 bit times) ..... 11 111110: 254 bit times)
7	SIM_CLK_ON	When this bit is set(1), the UIM_CLK is turned on at either the TD8 or TD4 frequency, as indicated by bit6(UIM_CLK_TD8_SEL). When this bit is cleared, the UIM_CLK is stopped either LOW or HIGH, as indicated by bit 5(UIM_CLK_STOP_HIGH).
6	SIM_CLK_TD8_SEL	This bit selects the UIM_CLK frequency at which the UIM_CLK is turned on. 0x1: TD8 (UIM_CLK runs at the TD8 frequency (1/2 the frequency of SIM-SRC_CLK)) 0x0: TD4 (UIM_CLK runs at the TD4 frequency (same frequency as SIM-SRC_CLK))
5	SIM_CLK_STOP_HIGH	This bit indicates at what point - high or low - the UIM_CLK will stop. 0x1: HIGH (high) 0x0: LOW (low)
3	MASK_RX	Setting (1) this bit masks off any 0 on the RX line, which prevents the receiver from seeing the START bit. Set (1) this bit if you do not want to receive the data you transmit.
2	SWAP_D	Setting (1) this bit causes the UIM_IF to swap the 8 data bits (making MSB into LSB, bit 6 into bit1, and so on) before it is read by the microprocessor.
1	INV_D	Set (1) this bit to cause the UIM_IF to invert the 8 data bit before it is read by the microprocessor.
0	SIM_SEL	When set (1), UART_DM will operate in SIM/UIM interface mode. When cleared (0), SIM/UIM interface is disabled.

**0x0C1AF0A0 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_CSR register is the UART clock selection register. This register is used to determine the bit rate for the transmitter and receiver. The transmitter and receiver can be clocked at different rates.

Data-rate of receiver/transmitter is UART-CLK frequency divided by the integer matching the configured value below:

0xF = 16

0xE = 32

0xD = 48

0xC = 64

0xB = 96

0xA = 128

0x9 = 192

0x8 = 256

0x7 = 384

0x6 = 512

0x5 = 768

0x4 = 1536

0x3 = 3072

0x2 = 6144

0x1 = 12288

0x0 = 24576

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR**

Bits	Name	Description
7:4	UART_RX_CLK_SEL	Use the values above to select the receive bit rate.
3:0	UART_TX_CLK_SEL	Use the values above to select the transmit bit rate.

#### **0x0C1AF0A4 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SR**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

The UART\_DM\_SR register is the UART status register. This register is used to obtain the current state of the UART subsystem. This register is updated asynchronously.

### **PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SR**

Bits	Name	Description
12	COMMAND_IN_PROGRESS	When 1 the command is in progress
11:10	TRANS_END_TRIGGER	RX-transfer-end event type. Stores trigger for last transfer end : '00' for no trigger, '01' for DMRX-event, '10' for timeout stale, '11' for SW stale.
9	TRANS_ACTIVE	Transfer Active bit.
8	RX_BREAK_START_LAST	When break start is detected, this bit is set (1). When break end was detected, this bit is reset (0).
7	HUNT_CHAR	When set (1), this bit indicates that the programmed character in the UART_DM_HCR register was received. If programmed, this condition causes the RXHUNT bit to be set (1) in the UART_DM_ISR register. This bit is cleared by issuing a reset error status command.
6	RX_BREAK	When set (1), this bit indicates that an all-zero character of the programmed length was received without a stop bit. If a break begins in the middle of a character, it is detected if the condition remains until the end of the next character time. Only one all zero character is written into the FIFO upon receiving the break. After the RX input returns high for at least half a bit time, then more characters are accepted by the receiver. Whenever the break condition starts or stops as defined above, the RXBREAK bit is set (1) in the UART_DM_ISR register. This bit is appended to the character in the receive FIFO.
5	PAR_FRAME_ERR	A parity error occurs if parity is programmed to be checked and an incoming character is received with incorrect parity. A framing error occurs when the RX input is low while the stop bit is being sampled. This bit is sampled in the middle of the first stop bit position, regardless of the programmed length of the stop bit. This error bit is appended to the character in the receive FIFO.
4	UART_OVERRUN	An overrun error occurs if one or more incoming characters are lost. This happens when the receive FIFO is full, the packing buffer is full and another character has been received in the shift register. The character in the receive shift register is lost. Upon overrun, this status bit remains high until cleared by a reset error status command.
3	TXEMT	This bit is set (1) when the transmitter under-runs. It is set after the transmission of the stop bit at the end of a character, if there are no characters waiting to be sent. It is reset when a character is written to the transmit FIFO.
2	TXRDY	This bit indicates that the transmit FIFO has space in it. It is reset when the FIFO becomes full and set when space becomes available again. Any characters that are written to the FIFO while this bit is low is lost.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SR (cont.)**

Bits	Name	Description
1	RXFULL	This bit is set (1) when the receive FIFO becomes full. It is reset when the CPU reads a word.
0	RXRDY	This bit is set (1) when there is a word in the receive FIFO waiting to be read. It is reset when the last word currently stored in the FIFO is read.

**0x0C1AF0A8 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR**

Bits	Name	Description
11	CHANNEL_COMMAND_MSB	This is the msb of the CHANNEL_COMMAND bitfield.
10:8	GENERAL_COMMAND	Writing the appropriate value to these bits executes the commands that are listed in Table .
7:4	CHANNEL_COMMAND_LSB	Writing the appropriate value to these bits along with bit 11,executes the channel-command as listed in the register description.
3	UART_TX_DISABLE	This command terminates the operation of the transmitter after any character in the transmit shift register is sent.
2	UART_TX_EN	This command enables the operation of the transmitter.
1	UART_RX_DISABLE	This command immediately terminates the operation of the channel receiver and any incoming characters are lost. None of the receiver status bits are affected by this command and characters that are already in the receive FIFO remain there.
0	UART_RX_EN	This command enables the channel receiver.

**0x0C1AF0AC PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR**

Bits	Name	Description
17:0	UART_MISR	The UART_DM_MISR register is the masked interrupt status register. A read from this register returns the 'AND' of the UART_DM_ISR and the UART_DM_IMR registers. This register is updated asynchronously. This bit field is misr <= (isr(16 Downto 7) AND imr(16 Downto 7)) & '0' & (isr(5 Downto 0) AND imr(5 Downto 0)).

**0x0C1AF0B0 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IMR register is the UART interrupt mask register. This register is used to enable the corresponding functions in the UART\_DM\_ISR register. Setting (1) a bit in the UART\_DM\_IMR register causes an interrupt to be generated, if the corresponding bit in the UART\_DM\_ISR register is set. Clearing (0) a bit in the UART\_DM\_IMR register causes the setting of the corresponding bit in the UART\_DM\_ISR register to have no effect on the interrupt signal.

Bit 6 of the UART\_DM\_IMR register, CURRENT\_CTS, is slightly different. If the current value of the CTS is one (1), no interrupt is generated. However, the user can use bit 6 in this register to mask the CTS value when reading the UART\_DM\_MISR register or as a general-purpose bit.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Masking bit of the matching field interrupt bit
16	WWT_IRQ	Masking bit of the matching field interrupt bit
15	TXCOMP_IRQ	Masking bit of the matching field interrupt bit
14	RX_RD_ERROR_IRQ	Masking bit of the matching field interrupt bit
13	TX_WR_ERROR_IRQ	Masking bit of the matching field interrupt bit
12	PAR_FRAME_ERR_IRQ	Masking bit of the matching field interrupt bit
11	RXBREAK_END	Masking bit of the matching field interrupt bit
10	RXBREAK_START	Masking bit of the matching field interrupt bit
9	TX_DONE	Masking bit of the matching field interrupt bit
8	TX_ERROR	Masking bit of the matching field interrupt bit
7	TX_READY	Masking bit of the matching field interrupt bit
6	CURRENT_CTS	Masking bit of the matching field interrupt bit

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR (cont.)**

Bits	Name	Description
5	DELTA_CTS	Masking bit of the matching field interrupt bit
4	RXLEV	Masking bit of the matching field interrupt bit
3	RXSTALE	Masking bit of the matching field interrupt bit
2	RXBREAK_CHANGE	Masking bit of the matching field interrupt bit
1	RXHUNT	Masking bit of the matching field interrupt bit
0	TXLEV	Masking bit of the matching field interrupt bit

**0x0C1AF0B4 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_ISR register is the UART interrupt status register. This register provides the current status of the possible interrupt conditions. If one of these bits is set (1), and the corresponding bit in the UART\_DM\_IMR register is set (1), an interrupt is generated (with the exception of bit 6, CURRENT\_CTS). If the corresponding bit in the UART\_DM\_IMR register is clear (0), setting one of these bits has no effect on the UART interrupt request signal. This register is updated asynchronously.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	This interrupt is fires if:  1.A new command is issued while the previous command have not finished yet. Cleared by issuing CHANNEL_COMMAND = CLEAR_NO_FINISH_CMD_VIOL_IRQ (UART_DM_CR register).
16	WWT_IRQ	WWT timeout reached interrupt. Fired after WWT timer was enabled and no data was received during the configured time that follows. Cleared by issuing CHANNEL_COMMAND = CLEAR_WWT_IRQ (UART_DM_CR register).
15	TXCOMP_IRQ	TX-complete interrupt. Fired when TX channel finished transmitting all of TX-transfer's data and no new transfer was initiated. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_COMP_IRQ (UART_DM_CR register).
14	RX_RD_ERROR_IRQ	RX-FIFO empty read error. Fired when AHB master is trying to read from an empty RX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_RX_RD_ERROR_IRQ (UART_DM_CR register).

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR (cont.)**

Bits	Name	Description
13	TX_WR_ERROR_IRQ	TX-FIFO full write error. Fired when AHB master is trying to write into a full TX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_WR_ERROR_IRQ (UART_DM_CR register).
12	PAR_FRAME_ERR_IRQ	This bit is asserted in the same condition at which PAR_FRAME_ERR status bit is asserted. Cleared by issuing CHANNEL_COMMAND = RESET_PAR_FRAME_ERR_IRQ (UART_DM_CR register).
11	RXBREAK_END	This bit is set (1) if the end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_END_IRQ (UART_DM_CR register).
10	RXBREAK_START	This bit is set (1) if the start of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_START_IRQ (UART_DM_CR register).
9	TX_DONE	<p>This bit is used only in UIM/SIM_IF mode. Set (1) this bit when the last byte in the transmitter has been sent. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_DONE_IRQ (UART_DM_CR register).</p> <ul style="list-style-type: none"> <li>- This bit is generated only when SIM_GLUE_GEN generic equals 1.</li> </ul>
8	TX_ERROR	<p>Only used in UIM_IF mode. Set (1) this bit to indicate that there has been parity error during transmission. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_ERROR_IRQ (UART_DM_CR register).</p> <ul style="list-style-type: none"> <li>- This bit is generated only when SIM_GLUE_GEN generic equals 1.</li> </ul>
7	TX_READY	<p>This bit, when set(1), indicates that:</p> <ol style="list-style-type: none"> <li>1. TX FIFO is empty.</li> <li>2. The value written in NO_CHARS_FOR_TX register equals the number of bytes written to the TX FIFO since the register was updated.</li> </ol> <p>NOTE: There may be characters in the unpack buffer or in the shift register.</p> <p>Cleared by issuing GENERAL_COMMAND = RESET_TX_READY_IRQ (UART_DM_CR register)</p>
6	CURRENT_CTS	This bit indicates the current state of the CTS input, high ('1') meaning peer is not ready for data reception, low ('0') - peer is ready. The toggling of this bit does not generate an interrupt.
5	DELTA_CTS	This bit, when set (1), indicates that the CTS input has changed states. Cleared by issuing CHANNEL_COMMAND = RESET_CTS_N (UART_DM_CR register).
4	RXLEV	This bit is set when a word is loaded into the receive FIFO that brings the total number of words in the FIFO above the programmed watermark level in the FIFO watermark register (UART_DM_RFWR). This bit is cleared after enough words have been read to bring the level equal to or below the programmed watermark level.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR (cont.)**

Bits	Name	Description
3	RXSTALE	This bit indicates that an RX-transfer has finished. Cleared by issuing CHANNEL_COMMAND = RESET_STALE_IRQ (UART_DM_CR register).
2	RXBREAK_CHANGE	This bit is set (1) if the start or end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_CHANGE_IRQ (UART_DM_CR register). A detected break is defined as an all-zeros character with an invalid stop bit AND LOW PARITY. The end of a break is defined as a mark condition (1) at least 1/2 bit width. A detected break occupies one position in the Rx FIFO.
1	RXHUNT	This bit is set (1) when an incoming character matches the value programmed into the UART_DM_HCR register. Cleared by issuing CHANNEL_COMMAND = RESET_ERROR_STATUS (UART_DM_CR register).
0	TXLEV	This bit is set (1) when a word which is transferred from the transmit FIFO to the transmit shift register brings the total number of words in the FIFO below or equal to the programmed watermark level in the UART_DM_TFWR register. This bit is cleared (0) after enough words have been written to the FIFO to bring the level above the programmed watermark level. Since this bit does not have a Clear command, during interrupt handling this bit must be masked (using the matching bit in UART_DM_IMR) until writing additional data for transmission is required. After data is written, this interrupt-bit should be unmasked.

**0x0C1AF0B8 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IRDA register enables the IRDA function, selects a loopback, and allows inversion of RX and TX data. This register also controls the IRDA transceiver that optionally interfaces between the UART and the DP\_RX\_DATA and DP\_TX\_DATA pins.

The UART proper is a standard RX/TX configuration that converts the serial pin input and output lines to 8-bit data for the microprocessor. The configuration contains receive and transmit blocks, a control block, a bit rate generator for RX and TX, and FIFO interfaces with the microprocessor data bus. The IRDA register provides the means of switching IRDA circuits into the serial I/O lines and the IRDA function is normally switched out. The IRDA modulator/demodulator may be enabled or disabled under microprocessor control. When enabled, the RX and TX data paths have individual inversion select bits, so external true or inverted drivers may be used in the signal path.

\* The register is generated only when IRDA\_IFC\_GEN generic equals 1.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA**

Bits	Name	Description
4	MEDIUM_RATE_EN	- Set (1) for 1/4 bit-time pulse length (Medium rate) - Clear (0) for 3/16 bit-time pulse length (Low rate)
3	IRDA_LOOPBACK	- This bit loops the IRDA modulated TX line back into the IRDA RX line. The normal UART loopback mode must be off in order to see this effect.
2	INVERT_IRDA_TX	This bit inverts the polarity of the IRDA modulated signal on the DP_TX_DATA pin. - Set (1) this bit for an inverted polarity. - Clear (0) this bit for a non-inverted polarity.
1	INVERT_IRDA_RX	This bit inverts the polarity of the IRDA received signal on the DP_RX_DATA pin. - Set (1) this bit for inverted the polarity. - Clear (0) this bit for non-inverted polarity.
0	IRDA_EN	- Set (1) this bit to enable the IRDA transceiver. - Clear (0) this bit to bypass the IRDA transceiver and ignore the other bits of this register.

**0x0C1AF0BC PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TOTAL\_SNAP****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TOTAL\_SNAP**

Bits	Name	Description
23:0	RX_TOTAL_BYTES	•RX_TOTAL_SNAP register holds the number of characters received since the end of last Rx transfer. It is updated at the end of an Rx transfer.  Rx transfer ends when one of the conditions is met: - The number of characters which were received since the end of the previous transfer equals the value which was written to DMRX at 'Transfer initialization'. - A stale event occurred (flush operation already performed if was needed)

**0x0C1AF0C0 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_WWT\_TIMEOUT****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_WWT\_TIMEOUT**

Bits	Name	Description
25	WWT_CYCLEREENABLE	When 1, a new received character will re-load the WWT counter instead of disabling it
24:0	WWT_TIMEOUT	timeout value for WWT mechanism. Minimal value should be 10 , otherwise the WWT_IRQ will be received even if two characters will follow each other

**0x0C1AF0C4 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL**

Bits	Name	Description
23	UART_IRDA_CLK_CGC_OPEN	Forces UART IRDA-clock (uart_tx_clk) CGC to open when set.
22	UART_SIM_CLK_CGC_OPEN	Forces UART SIM-clock (uart_tx_clk) CGC to open when set.
21	UART_RX_CLK_CGC_OPEN	Forces UART RX-clock (uart_tx_clk) CGC to open when set.
20	UART_TX_CLK_CGC_OPEN	Forces UART TX-clock (uart_tx_clk) CGC to open when set.
14	AHB_RX_BAM_CLK_CGC_OPEN	Forces AHB RX-BAM-clock (hrx_bam_clk) CGC to open when set.
13	AHB_TX_BAM_CLK_CGC_OPEN	Forces AHB TX-BAM-clock (htx_bam_clk) CGC to open when set.
10	AHB_RX_CLK_CGC_OPEN	Forces AHB RX-clock (hrx_clk) CGC to open when set.
9	AHB_TX_CLK_CGC_OPEN	Forces AHB TX-clock (htx_clk) CGC to open when set.
8	AHB_WR_CLK_CGC_OPEN	Forces AHB write-clock (hwr_clk) CGC to open when set.
5	RX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to RX-DISABLE command/RX-RESET command only when set.
4	TX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to TX-DISABLE command/TX-RESET command only when set.
0	AHB_CLK_CGC_CLOSE	Setting this bit to '1' will completely gate AHB-clock to the core. Only allowed register access during the duration this bit is set is the access to reset this bit. All other register accesses are not allowed and might result in errors. When set to '0', AHB clock is not gated at the root of the core and can be gated by the other gating features.

**0x0C1AF0C8 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BCR**

Bits	Name	Description
6	IGNORE_CR_PROT_VIOL	Relevant only when command protection is disabled. If during on-going command a new command is issued the new command is disregarded. If the bit is disabled legacy behavior is used
5	RX_DMRX_1BYTE_RES_EN	Allows DMRX values other than a multiple of 16 if set.
4	RX_STALE_IRQ_DMRX_EQUAL	If set, Stale irq will be asserted when (DMRX==valid_char_count) and stale is not enabled.
2	RX_DMRX_LOW_EN	DMRX lower than valid_char_cnt logic is enabled if bit is set
1	STALE_IRQ_EMPTY	Stale interrupt when RX-FIFO is empty will fire if bit is set. This relates to the issue that was when state when SW reads all the characters from the fifo and timeout expired. In that case without this bit enable interrupt will not be fired
0	TX_BREAK_DISABLE	TX-pin value when transmitter disabled while in break state is '0' if set, else '1'

**0x0C1AF0CC PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TRANS\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TRANS\_CTRL**

Bits	Name	Description
2	RX_DMRX_CYCLIC_EN	If set - Auto-Re-Activated RX-transfer will be re-activated with a DMRX value from the DMRX register, making a DMRX-transfer-end-event possible. Otherwise, Auto-Re-Activated RX-transfer cannot end with a DMRX event unless its value is written to the DMRX register throughout the transfer.
1	RX_TRANS_AUTO_RE_ACTIVE	RX-transfer will be automatically re-activated after last data of previous transfer was read - if set. Otherwise, transfers are only activated by writing to DMRX register.
0	RX_STALE_AUTO_RE_EN	RX-Stale automatic re-enable - on if set.

**0x0C1AF0D4 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS**

Bits	Name	Description
29:28	TX_COMP_FSM	Value of TX-COMP FSM state : '00' for empty, '01' non-empty, '10' for empty-check, '11' for empty-check-fail.
26:24	RX_PACK_FSM	Value of RX-packing FSM state : '000' for idle, '001' for fifo_write, '010' for wait_ack, '011' for flush_check, '100' for flushing, '101' for flush_done.
21:20	RX_TRANS_FSM	Value of RX-transfer FSM state : '00' for inactive, '01' for active, '10' for active_dmrx, '11' for active_flush.
18:16	TX_TRANS_FSM	Value of TX-transfer FSM state : '000' for idle, '001' for bam_idle, '010' for active, '011' for inactive, '100' for last_trap, '101' for nwd.
14:12	RX_PRO_TRANS_END_FSM	Value of RX_PRO_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char_dphase, '101' for mess_only_dphase.
10:8	RX_PRO_ACTIVE_FSM	Value of RX_PRO_ACTIVE FSM state : '000' for disabled, '001' for inactive, '010' for wait_check_1, '011' for wait_check_1, '100' for pf_pending, '101' for wait_update_1, '110' for wait_update_2, '111' for inactive?
6:4	TX_CON_TRANS_END_FSM	Value of TX_CON_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char_dphase, '101' for mess_only_dphase, '110' for normal_dphase.
0	RX_TRANSFER_ACTIVE	1 is RX transfer is currently active, 0 if inactive.

**0x0C1AF0DC PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_GENERICS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_GENERICS**

Bits	Name	Description
7	GENERIC_BAM_IFC	Value of 'BAM_IFC_GEN' generic.
6	GENERIC_DM_IFC	Value of 'DM_IFC_GEN' generic.
5	GENERIC_IRDA_IFC	Value of 'IRDA_IFC_GEN' generic
4	GENERIC_SIM_GLUE	Value of 'SIM_GLUE_GEN' generic

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_GENERICS (cont.)**

Bits	Name	Description
3:0	GENERIC_RAM_ADDR_WI_DTH	Value of 'RAM_ADDR_WIDTH' generic.

**0x0C1AF100 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

In order to support bursts of 4, 0x70-0x7C address space is reserved for the UART\_DM TX FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF**

Bits	Name	Description
31:0	UART_TF	The UART_TF register is the UART transmit FIFO register. This register is used to access the channel transmit FIFO. If the FIFO is full at the time of writing, the characters are lost and an interrupt is generated.

**0x0C1AF104 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_2****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_2**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF108 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_3****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_3**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF10C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_4****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_4**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF110 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_5****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_5**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF114 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_6****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_6**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF118 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_7****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_7**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF11C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_8****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_8**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF120 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_9****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_9**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF124 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_10****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_10**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF128 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_11****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_11**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF12C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_12****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_12**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF130 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_13****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_13**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF134 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_14****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_14**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF138 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_15****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_15**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF13C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_16****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_16**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1AF140 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RF register is the UART receive FIFO register, which is used to access the channel receive FIFO. In order to support bursts of 16, 0x140-0x17C address space is reserved for the UART\_DM RX FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF**

Bits	Name	Description
31:0	UART_RF	This register returns the next word in the receive FIFO. After the read is completed, the FIFO read pointer is updated to make the next word available.

**0x0C1AF144 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_2****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_2**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF148 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_3****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_3**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF14C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_4****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_4**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF150 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_5****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_5**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF154 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_6****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_6**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF158 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_7****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_7**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF15C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_8****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_8**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF160 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_9****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_9**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF164 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_10****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_10**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF168 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_11****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_11**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF16C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_12****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_12**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF170 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_13****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_13**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF174 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_14****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_14**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF178 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_15****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_15**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF17C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_16****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_16**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1AF184 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_CMD****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

Command register for UIM-controller (linked with this UART\_DM). Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_CMD**

Bits	Name	Description
1	RECOVER_FROM_HW_DE_ACTIVATION	Writing '1' to this bit will cause the UIM-io-ctrl to exit override state and reflect UIM signals from UART_DM to the UIM card. Setting both bits of this register to '1' will cause neither of the commands to be executed.
0	INITIATE_HW_DEACTIVATION	Initiates a HW deactivation sequence when writing '1' to this bit. This executes the sequence on the uim_data, uim_clk & uim_rst_n lines and alerts the PMIC to cut the power-supply to the card afterwards. Setting both bits of this register to '1' will cause neither of the commands to be executed.

**0x0C1AF188 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000002**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STATUS**

Bits	Name	Description
2	UIM_IO_WRITE_IN_PROGRESS	Asserted after UIM-write (UIM_CFG/UIM_CMD register write) while write is being propagated into the UIM-IO-controller. An additional read/write to UIM_CFG/UIM_CMD register should be done while high. De-asserted when write finishes propagating.
1	UIM_DEACTIVATION_STATUS	Indicates whether a HW deactivation sequence was performed and sequence recovery hasn't been executed yet. This bit is set when the sequence is initiated either by a HW trigger (batt_alarm, card-removal) or a SW trigger ('initiate_shutdown_sequence' bit in CMD register). Cleared after executing a 'recover_from_hw_deactivation' command.
0	CARD_PRESENCE	Indication on card presence in slot. High when card present, low otherwise.

**0x0C1AF18C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Asserted when UIM-write had finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
3	HW_SEQUENCE_FINISH	Asserted when UIM-controller's HW-deactivation sequence has been initiated and finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
2	BATT_ALARM	Set to high when battery-alarm indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
1	UIM_CARD_INSERTION	Set to high when card insertion indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
0	UIM_CARD_REMOVAL	Set to high when card removal indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.

**0x0C1AF190 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	See UART_DM_UIM_IRQ_ISR for field descriptions.
3	HW_SEQUENCE_FINISH	See UART_DM_UIM_IRQ_ISR for field descriptions.
2	BATT_ALARM	See UART_DM_UIM_IRQ_ISR for field descriptions.
1	UIM_CARD_INSERTION	See UART_DM_UIM_IRQ_ISR for field descriptions.
0	UIM_CARD_REMOVAL	See UART_DM_UIM_IRQ_ISR for field descriptions.

**0x0C1AF194 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
3	HW_SEQUENCE_FINISH	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
2	BATT_ALARM	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
1	UIM_CARD_INSERTION	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
0	UIM_CARD_REMOVAL	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.

**0x0C1AF198 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR (cont.)**

Bits	Name	Description
3	HW_SEQUENCE_FINISH	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
2	BATT_ALARM	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
1	UIM_CARD_INSERTION	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
0	UIM_CARD_REMOVAL	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.

**0x0C1AF19C PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_SET****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_SET**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1AF1A0 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR (cont.)**

Bits	Name	Description
1	UIM_CARD_INSERTION	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1AF0E0 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

New added interrupt clear register

**PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR\_CLR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Clears NO_FINISH_CMD_VIOL bit in UART_DM_ISR

**0x0C1B0000 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR1****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_MR1 register is the UART mode register #1. It is used, along with UART\_DM\_MR2, to configure the operational mode of the UART.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR1**

Bits	Name	Description
31:8	AUTO_RFR_LEVEL1	<p>These bits are used, along with field AUTO_RFR_LEVEL0 to program the level in the receive FIFO at which the Ready-For-Receive signal (RFR_N) toggles, if programmed to do so.</p> <p>When RX-FIFO level is greater than the value in fields AUTO_RFR_LEVEL1 and AUTO_RFR_LEVEL0 , signal will indicate Not-Ready (RFR_N high). Otherwise signal will indicate Ready RFR_N low.</p> <p>Configured value should be higher/equal than RFWR value, otherwise the watermark will never be reached.</p> <p>Value of this register is in words and can be programmed from 1 to <math>2^{\text{RAM\_ADDR\_WIDTH}}</math>.</p> <p>- Only RAM_ADDR_WIDTH +1:8 bits of this field are generated.</p>

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR1 (cont.)**

Bits	Name	Description
7	RX_RDY_CTL	Setting (1) this bit enables the automatic ready-for-receiving (RFR_N) control for the receiver. RFR_N is turned off, or set (1), when a valid start bit is received and the RX channel FIFO is at the level programmed in bits [31:8,5:0] of this register. When the FIFO level falls back to the programmed level, the RFR_N signal is turned on, or clear (0). When this feature is off, the RFR_N signal can be used by normal signal port bit manipulation.
6	CTS_CTL	When this bit is set(1), the transmitter checks the CTS_N input to determine whether to begin transmission of a new character. If CTS_N is low, the character is sent. Otherwise the transmitter continues marking until CTS_N goes low, then the next character is transmitted. A change on CTS_N during the transmission of a character has no effect on that character. When this bit is clear(0), the CTS_N input for the channel has no effect on the transmitter.
5:0	AUTO_RFR_LEVEL0	Same description of field AUTO_RFR_LEVEL1.

**0x0C1B0004 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR2****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR2**

Bits	Name	Description
10	RFR_CTS_LOOPBACK	Enables internal loopback between RFR_N of RX channel and CTS_N of TX channel if set. For this feature to be enabled, data loopback must also be enabled (LOOPBACK bit of this register).
9	RX_ERROR_CHAR_OFF	When this bit is set, characters received with parity or framing errors get discarded and not moved to the RX channel. Otherwise they enter RX channel as any valid character.
8	RX_BREAK_ZERO_CHAR_OFF	When this bit is set, the zero character received at rx_break event get discarded and not moved to the RX channel. Otherwise it enters RX-channel as any valid character.
7	LOOPBACK	When set - data output of transmitter is fed back to the data input of the receiver, i.e. TX-pin is connected to RX-pin. Might be used with flow-control loopback (RFR_CTS_LOOPBACK bit above).
6	ERROR_MODE	This bit controls the operation of the two status register bits for the RX channel (parity or framing error and received break). <ul style="list-style-type: none"> <li>- When clear (0), the UART operates in character mode and the status bits apply only to the last character received.</li> <li>- When set (1), the UART operates in block mode and both bits are sticky, i.e. they go high once an error/break-event is detected and go low only after reset error status command is issued.</li> </ul>

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR2 (cont.)**

Bits	Name	Description
5:4	BITS_PER_CHAR	<p>These bits determine how many data bits are transmitted or received per character, not including the start, stop, and parity bits.</p> <p>0x0: ENUM_5_BITS (5 bits) 0x1: ENUM_6_BITS (6 bits) 0x2: ENUM_7_BITS (7 bits) 0x3: ENUM_8_BITS (8 bits)</p>
3:2	STOP_BIT_LEN	<p>This field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 9/16, 1, 1+ 9/16, and 2 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length.</p> <p>0x0: ENUM_0_563 (0.563 (9/16) bit times) 0x1: ENUM_1_000_BIT_TIME (1.000 bit time) 0x2: ENUM_1_563 (1.563 (1+9/16) bit times) 0x3: ENUM_2_000_BIT_TIMES (2.000 bit times)</p>
1:0	PARITY_MODE	<p>These bits determine which parity mode is used. The user can select between odd, even, space, or no parity.</p> <p>0x0: NO_PARITY (no parity - last data-bit is followed by a stop-bit) 0x1: ODD_PARITY (odd parity - parity-bit ensures number of ones in data-bits+parity-bit is odd) 0x2: EVEN_PARITY (even parity - parity-bit ensures number of ones in data-bits+parity-bit is even) 0x3: SPACE_PARITY (space parity - parity is always '0')</p>

**0x0C1B0008 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR\_SR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CSR or UART\_DM\_SR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR\_SR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CSR_SR_DEPRECATED	

**0x0C1B0010 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR\_MISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_CR or UART\_DM\_MISR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR\_MISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_CR_MISR_DEPRECATED	

**0x0C1B0014 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR\_ISR\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IMR or UART\_DM\_ISR.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR\_ISR\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_IMR_ISR_DEPRECATED	

**0x0C1B0018 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IPR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0xFFFFF9F

The UART\_DM\_IPR register is the UART interrupt programming register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IPR**

Bits	Name	Description
31:7	STALE_TIMEOUT_MSB	The stale character time-out duration field contains a number from 1 to $2^{30} - 1$ . This number determines how many character times must elapse before a stale event is generated. This character time is defined as 10 times the bit rate. It is reset whenever a data bit is received. Never set this fields to 0 while RX channel is operational. Do not re-configure while an RX-transfer is active. NOTE: the discontinuity in the bit assignments.
6	SAMPLE_DATA	Currently not in use. Setting (1) to this bit enables the sample data mode. In this mode start bit is sampled along with the data of the UART character.
4:0	STALE_TIMEOUT_LSB	This bitfield is the LSbits of the STALE_TIMEOUT bitfield.

**0x0C1B001C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_TFWR register is the UART transmit FIFO watermark register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TFWR**

Bits	Name	Description
31:0	TFW	These bits contain a number, between 0 and $2^{\text{RAM\_ADDR\_WIDTH}} - 1$ , that determines the level of the transmit (TX) FIFO at which space_avail_req signal is asserted in BAM mode or TXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of TX words in the FIFO) is less than or equal to the value in TFW. Notice that the value is in words. - Only RAM_ADDR_WIDTH -1:0 bits are generated.

**0x0C1B0020 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RFWR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RFWR register is the UART receive FIFO watermark register.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RFWR**

Bits	Name	Description
31:0	RFW	<p>These bits contain a number, between 0 and <math>2^{\text{RAM\_ADDR\_WIDTH}} - 1</math>, that determines the level of the receive (RX) FIFO at which data_avail_req signal is asserted in BAM mode or RXLEV interrupt is asserted in non-BAM mode. The signal/interrupt is asserted when the FIFO level (number of words in the RX FIFO) is greater than the value in RFWR. Configured value should be lower/equal than AUTO_RFR_LEVEL value, otherwise the watermark will never be reached. Notice that the value is in words.</p> <ul style="list-style-type: none"> <li>- Only RAM_ADDR_WIDTH -1:0 bits are generated.</li> </ul>

**0x0C1B0024 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_HCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Configures Hunt-Character value.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_HCR**

Bits	Name	Description
7:0	DATA	The UART_DM_HCR register is the UART hunt character register, which contains the character for which the receiver looks to assert a hunt character received interrupt.

**0x0C1B0034 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMRX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initializes RX transfer and configures maximal length in bytes of the transfer.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMRX**

Bits	Name	Description
24:0	RX_DM_CRCI_CHARS	<p>A write to RX_DM_CRCI_CHARS initializes RX transfer in SW (FIFO) mode, with written value configuring the maximal length in bytes of the transfer. In BAM-mode, only the value held in this register impacts the data-flow and not the write-event. A write-value of 0 is not valid.</p> <p>Read of UART_DM_DMRX register returns the number of characters that were received since the end of the last RX transfer.</p>

## 0x0C1B0038 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED

**Type:** RW

**Clock:** AHB\_CLK

**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_IRDA or UART\_DM\_RX\_TOTAL\_SNAP.

### PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TOTAL\_SNAP\_DEPRECATED

Bits	Name	Description
31:0	UART_DM_IRDA_RX_TOTALL_SNAP_DEPRECATED	

## 0x0C1B003C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMEN

**Type:** RW

**Clock:** AHB\_CLK

**Reset State:** 0x00000000

The UART\_DM\_DMEN register defines the DMA used by the core and the data packing mode for TX and RX channels.

### PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMEN

Bits	Name	Description
5	RX_SC_ENABLE	- Set (1) for work in Single-Character mode for RX channel (every character received is zero-padded into a word). - Clearing this bit requires resetting the receiver.
4	TX_SC_ENABLE	- Set (1) for work in Single-Character mode for TX channel (Only LSB byte of each word get transmitted). - Clearing this bit requires resetting the transmitter.
3	RX_BAM_ENABLE	- Set (1) this bit to enable RX BAM interface (BAM-mode). - Clear (0) this bit to disable RX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the receiver.
2	TX_BAM_ENABLE	- Set (1) this bit to enable TX BAM interface (BAM-mode). - Clear (0) this bit to disable TX BAM interface (SW/FIFO mode). Clearing this bit requires resetting the transmitter.

**0x0C1B0040 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

Initialization of TX transfer in SW/FIFO mode.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX**

Bits	Name	Description
23:0	TX_TOTAL_TRANS_LEN	The total number of characters for transmission. Before writing a new value, the TX FIFO must be empty as indicated by TX_READY interrupt or after a TX-channel reset. It is used by the TX-channel in SW/FIFO mode to calculate how many characters to transmit in the last word. Any additional writes to the TX FIFO beyond TX_TOTAL_TRANS_LEN characters will be discarded. A TX_READY interrupt is triggered after TX_TOTAL_TRANS_LEN number of characters were moved from the TX FIFO to the unpacking buffer (not all may have been transmitted at that point though).

**0x0C1B0044 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BADR****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

Configures FIFO division between RX-FIFO and TX-FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BADR**

Bits	Name	Description
31:2	RX_BASE_ADDR	RX FIFO base address. Both FIFOs use the same RAM ( $2^{\text{RAM\_ADDR\_WIDTH}}$ , 32-bit entries). This register controls the division of the memory between the RX and TX FIFOs. The division must be a multiple of 4 entries for legacy reasons. The size of the TX FIFO equals RX_BASE_ADDR. The size of the RX FIFO is $2^{\text{RAM\_ADDR\_WIDTH}} - \text{RX\_BASE\_ADDR}$ . * The default is RX_BASE_ADDR = $2^{(\text{RAM\_ADDR\_WIDTH} - 1)}$ * Only RAM_ADDR_WIDTH - 1:2 bits are generated.

**0x0C1B004C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TXFS****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

TX channel status register.

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TXFS**

Bits	Name	Description
31:14	TX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. - Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	TX_ASYNC_FIFO_STATE	TX asynchronous fifo status - number of characters in the TX asynchronous FIFO.
9:7	TX_BUFFER_STATE	TX unpacking buffer status - number of bytes in the unpacking buffer
6:0	TX_FIFO_STATE_LSB	TX FIFO status - number of words in the TX FIFO

#### **0x0C1B0050 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RXFS**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

RX channel status register.

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RXFS**

Bits	Name	Description
31:14	RX_FIFO_STATE_MSB	The msb of TX_FIFO_STATE bitfield. Only RAM_ADDR_WIDTH +4 +3 -1:14 bits are generated.
13:10	RX_ASYNC_FIFO_STATE	RX asynchronous fifo status - number of characters in the RX asynchronous FIFO.
9:7	RX_BUFFER_STATE	RX packing buffer status - number of bytes in the packing buffer
6:0	RX_FIFO_STATE_LSB	The number of entries in the RX FIFO that have at least one valid word. It is incremented each time a word is moved from the packing register to the RX FIFO. It is decremented each time a word is read from the RX FIFO.  NOTE: The Uart does not keep track of non-valid characters in each word.

#### **0x0C1B0060 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_MODE**

**Type:** RW

**Clock:** WR\_CLK

**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_MODE**

Bits	Name	Description
1:0	RESERVED	RESERVED

**0x0C1B0064 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_RESET****Type:** W**Clock:** WR\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_RESET**

Bits	Name	Description
0	RESET	RESERVED

**0x0C1B006C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_VAL****Type:** R**Clock:** WR\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_VAL**

Bits	Name	Description
9:0	VAL	Current MISR state

**0x0C1B0070 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF or UART\_DM\_RF.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_DEPRECATED	

**0x0C1B0074 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_2\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_2 or UART\_DM\_RF\_2.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_2\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_2_DEPR ECATED	

**0x0C1B0078 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_3\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_3 or UART\_DM\_RF\_3.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_3\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_3_DEPR ECATED	

**0x0C1B007C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_4\_DEPRECATED****Type:** RW**Clock:** AHB\_CLK**Reset State:** Undefined

This register is deprecated and remains only for backward-compatibility. Use UART\_DM\_TF\_4 or UART\_DM\_RF\_4.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_4\_DEPRECATED**

Bits	Name	Description
31:0	UART_DM_TF_RF_4_DEPR ECATED	

**0x0C1B0080 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SIM\_CFG****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_SIM\_CFG register is used to configure the SIM interface for the UART.

\* This register is generated only when SIM\_GLUE\_GEN generic equals 1.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SIM\_CFG**

Bits	Name	Description
17	UIM_TX_MODE	When this bit is set (1), any re-transmission signal from the UIM card will be ignored (The transmission portion of the SIM interface operates in block mode T=1). When clear (0), re-transmission will be executed upon request (the transmission portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
16	UIM_RX_MODE	When this bit is set (1), error-signal indication will not be generated by the receiver (the receive portion of the SIM interface operates in block mode T=1). When clear (0), error signal will be generated when a character error is detected (the receive portion of the SIM interface operates in asynchronous transfer mode T=0). NOTE: that the operation of this function is conditional upon SIM_SEL field being set (1) to designate the UIM_IF mode of operation.
15:8	SIM_STOP_BIT_LEN	In UIM_IF mode, this field programs the duration of the stop bit that is appended to each transmitted character. The stop bit duration can be 1 to 254 bit times. The receiver only samples the stop bit once, regardless of the programmed stop bit length. 0x1: ENUM_1_BIT_TIMES (1 bit times) 0x2: ENUM_2_BIT_TIMES (2 bit times) ..... 11 111110: 254 bit times)
7	SIM_CLK_ON	When this bit is set(1), the UIM_CLK is turned on at either the TD8 or TD4 frequency, as indicated by bit6(UIM_CLK_TD8_SEL). When this bit is cleared, the UIM_CLK is stopped either LOW or HIGH, as indicated by bit 5(UIM_CLK_STOP_HIGH).
6	SIM_CLK_TD8_SEL	This bit selects the UIM_CLK frequency at which the UIM_CLK is turned on. 0x1: TD8 (UIM_CLK runs at the TD8 frequency (1/2 the frequency of SIM-SRC_CLK)) 0x0: TD4 (UIM_CLK runs at the TD4 frequency (same frequency as SIM-SRC_CLK))

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SIM\_CFG (cont.)**

Bits	Name	Description
5	SIM_CLK_STOP_HIGH	This bit indicates at what point - high or low - the UIM_CLK will stop. 0x1: HIGH (high) 0x0: LOW (low)
3	MASK_RX	Setting (1) this bit masks off any 0 on the RX line, which prevents the receiver from seeing the START bit. Set (1) this bit if you do not want to receive the data you transmit.
2	SWAP_D	Setting (1) this bit causes the UIM_IF to swap the 8 data bits (making MSB into LSB, bit 6 into bit1, and so on) before it is read by the microprocessor.
1	INV_D	Set (1) this bit to cause the UIM_IF to invert the 8 data bit before it is read by the microprocessor.
0	SIM_SEL	When set (1), UART_DM will operate in SIM/UIM interface mode. When cleared (0), SIM/UIM interface is disabled.

**0x0C1B00A0 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_CSR register is the UART clock selection register. This register is used to determine the bit rate for the transmitter and receiver. The transmitter and receiver can be clocked at different rates.

Data-rate of receiver/transmitter is UART-CLK frequency divided by the integer matching the configured value below:

0xF = 16

0xE = 32

0xD = 48

0xC = 64

0xB = 96

0xA = 128

0x9 = 192

0x8 = 256

0x7 = 384

0x6 = 512

0x5 = 768

0x4 = 1536

0x3 = 3072

0x2 = 6144

0x1 = 12288

0x0 = 24576

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR**

Bits	Name	Description
7:4	UART_RX_CLK_SEL	Use the values above to select the receive bit rate.
3:0	UART_TX_CLK_SEL	Use the values above to select the transmit bit rate.

#### **0x0C1B00A4 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SR**

Type: R

Clock: AHB\_CLK

Reset State: Undefined

The UART\_DM\_SR register is the UART status register. This register is used to obtain the current state of the UART subsystem. This register is updated asynchronously.

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SR**

Bits	Name	Description
12	COMMAND_IN_PROGRESS	When 1 the command is in progress
11:10	TRANS_END_TRIGGER	RX-transfer-end event type. Stores trigger for last transfer end : '00' for no trigger, '01' for DMRX-event, '10' for timeout stale, '11' for SW stale.
9	TRANS_ACTIVE	Transfer Active bit.
8	RX_BREAK_START_LAST	When break start is detected, this bit is set (1). When break end was detected, this bit is reset (0).
7	HUNT_CHAR	When set (1), this bit indicates that the programmed character in the UART_DM_HCR register was received. If programmed, this condition causes the RXHUNT bit to be set (1) in the UART_DM_ISR register. This bit is cleared by issuing a reset error status command.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SR (cont.)**

Bits	Name	Description
6	RX_BREAK	When set (1), this bit indicates that an all-zero character of the programmed length was received without a stop bit. If a break begins in the middle of a character, it is detected if the condition remains until the end of the next character time. Only one all zero character is written into the FIFO upon receiving the break.  After the RX input returns high for at least half a bit time, then more characters are accepted by the receiver. Whenever the break condition starts or stops as defined above, the RXBREAK bit is set (1) in the UART_DM_ISR register. This bit is appended to the character in the receive FIFO.
5	PAR_FRAME_ERR	A parity error occurs if parity is programmed to be checked and an incoming character is received with incorrect parity. A framing error occurs when the RX input is low while the stop bit is being sampled. This bit is sampled in the middle of the first stop bit position, regardless of the programmed length of the stop bit. This error bit is appended to the character in the receive FIFO.
4	UART_OVERRUN	An overrun error occurs if one or more incoming characters are lost. This happens when the receive FIFO is full, the packing buffer is full and another character has been received in the shift register. The character in the receive shift register is lost. Upon overrun, this status bit remains high until cleared by a reset error status command.
3	TXEMT	This bit is set (1) when the transmitter under-runs. It is set after the transmission of the stop bit at the end of a character, if there are no characters waiting to be sent. It is reset when a character is written to the transmit FIFO.
2	TXRDY	This bit indicates that the transmit FIFO has space in it. It is reset when the FIFO becomes full and set when space becomes available again. Any characters that are written to the FIFO while this bit is low is lost.
1	RXFULL	This bit is set (1) when the receive FIFO becomes full. It is reset when the CPU reads a word.
0	RXRDY	This bit is set (1) when there is a word in the receive FIFO waiting to be read. It is reset when the last word currently stored in the FIFO is read.

**0x0C1B00A8 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR**

Bits	Name	Description
11	CHANNEL_COMMAND_MS_B	This is the msb of the CHANNEL_COMMAND bitfield.
10:8	GENERAL_COMMAND	Writing the appropriate value to these bits executes the commands that are listed in Table .
7:4	CHANNEL_COMMAND_LSB	Writing the appropriate value to these bits along with bit 11,executes the channel-command as listed in the register description.
3	UART_TX_DISABLE	This command terminates the operation of the transmitter after any character in the transmit shift register is sent.
2	UART_TX_EN	This command enables the operation of the transmitter.
1	UART_RX_DISABLE	This command immediately terminates the operation of the channel receiver and any incoming characters are lost. None of the receiver status bits are affected by this command and characters that are already in the receive FIFO remain there.
0	UART_RX_EN	This command enables the channel receiver.

**0x0C1B00AC PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR**

Bits	Name	Description
17:0	UART_MISR	The UART_DM_MISR register is the masked interrupt status register. A read from this register returns the 'AND' of the UART_DM_ISR and the UART_DM_IMR registers. This register is updated asynchronously. This bit field is misr <= (isr(16 DOWNTO 7) AND imr(16 DOWNTO 7)) & '0' & (isr(5 DOWNTO 0) AND imr(5 DOWNTO 0)).

**0x0C1B00B0 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IMR register is the UART interrupt mask register. This register is used to enable the corresponding functions in the UART\_DM\_ISR register. Setting (1) a bit in the UART\_DM\_IMR register causes an interrupt to be generated, if the corresponding bit in the UART\_DM\_ISR register is set. Clearing (0) a bit in the UART\_DM\_IMR register causes the

setting of the corresponding bit in the UART\_DM\_ISR register to have no effect on the interrupt signal.

Bit 6 of the UART\_DM\_IMR register, CURRENT\_CTS, is slightly different. If the current value of the CTS is one (1), no interrupt is generated. However, the user can use bit 6 in this register to mask the CTS value when reading the UART\_DM\_MISR register or as a general-purpose bit.

### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Masking bit of the matching field interrupt bit
16	WWT_IRQ	Masking bit of the matching field interrupt bit
15	TXCOMP_IRQ	Masking bit of the matching field interrupt bit
14	RX_RD_ERROR_IRQ	Masking bit of the matching field interrupt bit
13	TX_WR_ERROR_IRQ	Masking bit of the matching field interrupt bit
12	PAR_FRAME_ERR_IRQ	Masking bit of the matching field interrupt bit
11	RXBREAK_END	Masking bit of the matching field interrupt bit
10	RXBREAK_START	Masking bit of the matching field interrupt bit
9	TX_DONE	Masking bit of the matching field interrupt bit
8	TX_ERROR	Masking bit of the matching field interrupt bit
7	TX_READY	Masking bit of the matching field interrupt bit
6	CURRENT_CTS	Masking bit of the matching field interrupt bit
5	DELTA_CTS	Masking bit of the matching field interrupt bit
4	RXLEV	Masking bit of the matching field interrupt bit
3	RXSTALE	Masking bit of the matching field interrupt bit
2	RXBREAK_CHANGE	Masking bit of the matching field interrupt bit
1	RXHUNT	Masking bit of the matching field interrupt bit
0	TXLEV	Masking bit of the matching field interrupt bit

### **0x0C1B00B4 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR**

**Type:** R

**Clock:** AHB\_CLK

**Reset State:** Undefined

The UART\_DM\_ISR register is the UART interrupt status register. This register provides the current status of the possible interrupt conditions. If one of these bits is set (1), and the corresponding bit in the UART\_DM\_IMR register is set (1), an interrupt is generated (with the exception of bit 6, CURRENT\_CTS). If the corresponding bit in the UART\_DM\_IMR register is

clear (0), setting one of these bits has no effect on the UART interrupt request signal. This register is updated asynchronously.

#### **PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	This interrupt is fires if:  1.A new command is issued while the previous command have not finished yet. Cleared by issuing CHANNEL_COMMAND = CLEAR_NO_FINISH_CMD_VIOL_IRQ (UART_DM_CR register).
16	WWT_IRQ	WWT timeout reached interrupt. Fired after WWT timer was enabled and no data was received during the configured time that follows. Cleared by issuing CHANNEL_COMMAND = CLEAR_WWT_IRQ (UART_DM_CR register).
15	TXCOMP_IRQ	TX-complete interrupt. Fired when TX channel finished transmitting all of TX-transfer's data and no new transfer was initiated. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_COMP_IRQ (UART_DM_CR register).
14	RX_RD_ERROR_IRQ	RX-FIFO empty read error. Fired when AHB master is trying to read from an empty RX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_RX_RD_ERROR_IRQ (UART_DM_CR register).
13	TX_WR_ERROR_IRQ	TX-FIFO full write error. Fired when AHB master is trying to write into a full TX-FIFO. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_WR_ERROR_IRQ (UART_DM_CR register).
12	PAR_FRAME_ERR_IRQ	This bit is asserted in the same condition at which PAR_FRAME_ERR status bit is asserted. Cleared by issuing CHANNEL_COMMAND = RESET_PAR_FRAME_ERR_IRQ (UART_DM_CR register).
11	RXBREAK_END	This bit is set (1) if the end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_END_IRQ (UART_DM_CR register).
10	RXBREAK_START	This bit is set (1) if the start of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_START_IRQ (UART_DM_CR register).
9	TX_DONE	This bit is used only in UIM/SIM_IF mode. Set (1) this bit when the last byte in the transmitter has been sent. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_DONE_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.
8	TX_ERROR	Only used in UIM_IF mode. Set (1) this bit to indicate that there has been parity error during transmission. Cleared by issuing CHANNEL_COMMAND = CLEAR_TX_ERROR_IRQ (UART_DM_CR register). - This bit is generated only when SIM_GLUE_GEN generic equals 1.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR (cont.)**

Bits	Name	Description
7	TX_READY	This bit, when set(1), indicates that: 1. TX FIFO is empty. 2. The value written in NO_CHARS_FOR_TX register equals the number of bytes written to the TX FIFO since the register was updated. NOTE: There may be characters in the unpack buffer or in the shift register. Cleared by issuing GENERAL_COMMAND = RESET_TX_READY_IRQ (UART_DM_CR register)
6	CURRENT_CTS	This bit indicates the current state of the CTS input, high ('1') meaning peer is not ready for data reception, low ('0') - peer is ready. The toggling of this bit does not generate an interrupt.
5	DELTA_CTS	This bit, when set (1), indicates that the CTS input has changed states. Cleared by issuing CHANNEL_COMMAND = RESET_CTS_N (UART_DM_CR register).
4	RXLEV	This bit is set when a word is loaded into the receive FIFO that brings the total number of words in the FIFO above the programmed watermark level in the FIFO watermark register (UART_DM_RFWR). This bit is cleared after enough words have been read to bring the level equal to or below the programmed watermark level.
3	RXSTALE	This bit indicates that an RX-transfer has finished. Cleared by issuing CHANNEL_COMMAND = RESET_STALE_IRQ (UART_DM_CR register).
2	RXBREAK_CHANGE	This bit is set (1) if the start or end of a break condition is detected. Cleared by issuing CHANNEL_COMMAND = RESET_BREAK_CHANGE_IRQ (UART_DM_CR register). A detected break is defined as an all-zeros character with an invalid stop bit AND LOW PARITY. The end of a break is defined as a mark condition (1) at least 1/2 bit width. A detected break occupies one position in the Rx FIFO.
1	RXHUNT	This bit is set (1) when an incoming character matches the value programmed into the UART_DM_HCR register. Cleared by issuing CHANNEL_COMMAND = RESET_ERROR_STATUS (UART_DM_CR register).
0	TXLEV	This bit is set (1) when a word which is transferred from the transmit FIFO to the transmit shift register brings the total number of words in the FIFO below or equal to the programmed watermark level in the UART_DM_TFWR register. This bit is cleared (0) after enough words have been written to the FIFO to bring the level above the programmed watermark level. Since this bit does not have a Clear command, during interrupt handling this bit must be masked (using the matching bit in UART_DM_IMR) until writing additional data for transmission is required. After data is written, this interrupt-bit should be unmasked.

**0x0C1B00B8 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000

The UART\_DM\_IRDA register enables the IRDA function, selects a loopback, and allows inversion of RX and TX data. This register also controls the IRDA transceiver that optionally interfaces between the UART and the DP\_RX\_DATA and DP\_TX\_DATA pins.

The UART proper is a standard RX/TX configuration that converts the serial pin input and output lines to 8-bit data for the microprocessor. The configuration contains receive and transmit blocks, a control block, a bit rate generator for RX and TX, and FIFO interfaces with the microprocessor data bus. The IRDA register provides the means of switching IRDA circuits into the serial I/O lines and the IRDA function is normally switched out. The IRDA modulator/demodulator may be enabled or disabled under microprocessor control. When enabled, the RX and TX data paths have individual inversion select bits, so external true or inverted drivers may be used in the signal path.

\* The register is generated only when IRDA\_IFC\_GEN generic equals 1.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA**

Bits	Name	Description
4	MEDIUM_RATE_EN	- Set (1) for 1/4 bit-time pulse length (Medium rate) - Clear (0) for 3/16 bit-time pulse length (Low rate)
3	IRDA_LOOPBACK	- This bit loops the IRDA modulated TX line back into the IRDA RX line. The normal UART loopback mode must be off in order to see this effect.
2	INVERT_IRDA_TX	This bit inverts the polarity of the IRDA modulated signal on the DP_TX_DATA pin. - Set (1) this bit for an inverted polarity. - Clear (0) this bit for a non-inverted polarity.
1	INVERT_IRDA_RX	This bit inverts the polarity of the IRDA received signal on the DP_RX_DATA pin. - Set (1) this bit for inverted the polarity. - Clear (0) this bit for non-inverted polarity.
0	IRDA_EN	- Set (1) this bit to enable the IRDA transceiver. - Clear (0) this bit to bypass the IRDA transceiver and ignore the other bits of this register.

**0x0C1B00BC PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TOTAL\_SNAP****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TOTAL\_SNAP**

Bits	Name	Description
23:0	RX_TOTAL_BYTES	<ul style="list-style-type: none"> <li>•RX_TOTAL_SNAP register holds the number of characters received since the end of last Rx transfer. It is updated at the end of an Rx transfer.</li> <li>Rx transfer ends when one of the conditions is met: <ul style="list-style-type: none"> <li>- The number of characters which were received since the end of the previous transfer equals the value which was written to DMRX at 'Transfer initialization'.</li> <li>- A stale event occurred (flush operation already performed if was needed)</li> </ul> </li> </ul>

**0x0C1B00C0 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_WWT\_TIMEOUT****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_WWT\_TIMEOUT**

Bits	Name	Description
25	WWT_CYCLEREENABLE	When 1, a new received character will re-load the WWT counter instead of disabling it
24:0	WWT_TIMEOUT	timeout value for WWT mechanism. Minimal value should be 10 , otherwise the WWT_IRQ will be received even if two characters will follow each other

**0x0C1B00C4 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL**

Bits	Name	Description
23	UART_IRDA_CLK_CGC_OPEN	Forces UART IRDA-clock (uart_tx_clk) CGC to open when set.
22	UART_SIM_CLK_CGC_OPEN	Forces UART SIM-clock (uart_tx_clk) CGC to open when set.
21	UART_RX_CLK_CGC_OPEN	Forces UART RX-clock (uart_tx_clk) CGC to open when set.
20	UART_TX_CLK_CGC_OPEN	Forces UART TX-clock (uart_tx_clk) CGC to open when set.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL (cont.)**

Bits	Name	Description
14	AHB_RX_BAM_CLK_CGC_OPEN	Forces AHB RX-BAM-clock (hrx_bam_clk) CGC to open when set.
13	AHB_TX_BAM_CLK_CGC_OPEN	Forces AHB TX-BAM-clock (htx_bam_clk) CGC to open when set.
10	AHB_RX_CLK_CGC_OPEN	Forces AHB RX-clock (hrx_clk) CGC to open when set.
9	AHB_TX_CLK_CGC_OPEN	Forces AHB TX-clock (htx_clk) CGC to open when set.
8	AHB_WR_CLK_CGC_OPEN	Forces AHB write-clock (hwr_clk) CGC to open when set.
5	RX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to RX-DISABLE command/RX-RESET command only when set.
4	TX_ENABLE_CGC_OPT	CGCs will gate AHB & UART clocks due to TX-DISABLE command/TX-RESET command only when set.
0	AHB_CLK_CGC_CLOSE	Setting this bit to '1' will completely gate AHB-clock to the core. Only allowed register acces during the duration this bit is set is the access to reset this bit. All other register accesses are not allowed and might result in errors. When set to '0', AHB clock is not gated at the root of the core and can be gated by the other gating features.

**0x0C1B00C8 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BCR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BCR**

Bits	Name	Description
6	IGNORE_CR_PROT_VIOL	Relevant only when command protection is disabled. If during on-going command a new command is issued the new command is disregarded. If the bit is disabled legacy behavior is used
5	RX_DMRX_1BYTE_RES_EN	Allows DMRX values other than a multiple of 16 if set.
4	RX_STALE_IRQ_DMRX_EQUAL	If set, Stale irq will be asserted when (DMRX==valid_char_count) and stale is not enabled.
2	RX_DMRX_LOW_EN	DMRX lower than valid_char_cnt logic is enabled if bit is set
1	STALE_IRQ_EMPTY	Stale interrupt when RX-FIFO is empty will fire if bit is set. This relates to the issue that was when state when SW reads all the characters from the fifo and timeout expired. In that case without this bit enable interrupt will not be fired
0	TX_BREAK_DISABLE	TX-pin value when transmitter disabled while in break state is '0' if set, else '1'

**0x0C1B00CC PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL**

Bits	Name	Description
2	RX_DMRX_CYCLIC_EN	If set - Auto-Re-Activated RX-transfer will be re-activated with a DMRX value from the DMRX register, making a DMRX-transfer-end-event possible. Otherwise, Auto-Re-Activated RX-transfer cannot end with a DMRX event unless its value is written to the DMRX register throughout the transfer.
1	RX_TRANS_AUTO_RE_ACTIVE	RX-transfer will be automatically re-activated after last data of previous transfer was read - if set. Otherwise, transfers are only activated by writing to DMRX register.
0	RX_STALE_AUTO_RE_EN	RX-Stale automatic re-enable - on if set.

**0x0C1B00D4 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS**

Bits	Name	Description
29:28	TX_COMP_FSM	Value of TX-COMP FSM state : '00' for empty, '01' non-empty, '10' for empty-check, '11' for empty-check-fail.
26:24	RX_PACK_FSM	Value of RX-packing FSM state : '000' for idle, '001' for fifo_write, '010' for wait_ack, '011' for flush_check, '100' for flushing, '101' for flush_done.
21:20	RX_TRANS_FSM	Value of RX-transfer FSM state : '00' for inactive, '01' for active, '10' for active_dmrx, '11' for active_flush.
18:16	TX_TRANS_FSM	Value of TX-transfer FSM state : '000' for idle, '001' for bam_idle, '010' for active, '011' for inactive, '100' for last_trap, '101' for nwd.
14:12	RX_PRO_TRANS_END_FSM	Value of RX_PRO_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char_dphase, '101' for mess_only_dphase.
10:8	RX_PRO_ACTIVE_FSM	Value of RX_PRO_ACTIVE FSM state : '000' for disabled, '001' for isactive, '010' for wait_check_1, '011' for wait_check_1, '100' for pf_pending, '101' for wait_update_1, '110' for wait_update_2, '111' for inactive?

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS (cont.)**

Bits	Name	Description
6:4	TX_CON_TRANS_END_FS_M	Value of TX_CON_TRANS_END FSM state : '000' for disabled, '001' for idle, '010' for 2char_dphase, '011' for 1char_waiting, '100' for 1char dphase, '101' for mess_only_dphase, '110' for normal_dphase.
0	RX_TRANSFER_ACTIVE	1 is RX transfer is currently active, 0 if inactive.

**0x0C1B00DC PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_GENERICS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_GENERICS**

Bits	Name	Description
7	GENERIC_BAM_IFC	Value of 'BAM_IFC_GEN' generic.
6	GENERIC_DM_IFC	Value of 'DM_IFC_GEN' generic.
5	GENERIC_IRDA_IFC	Value of 'IRDA_IFC_GEN' generic
4	GENERIC_SIM_GLUE	Value of 'SIM_GLUE_GEN' generic
3:0	GENERIC_RAM_ADDR_WIDHTH	Value of 'RAM_ADDR_WIDTH' generic.

**0x0C1B0100 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

In order to support bursts of 4, 0x70-0x7C address space is reserved for the UART\_DM TX FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF**

Bits	Name	Description
31:0	UART_TF	The UART_TF register is the UART transmit FIFO register. This register is used to access the channel transmit FIFO. If the FIFO is full at the time of writing, the characters are lost and an interrupt is generated.

**0x0C1B0104 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_2****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_2**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0108 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_3****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_3**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B010C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_4****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_4**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0110 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_5****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_5**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0114 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_6****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_6**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0118 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_7****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_7**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B011C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_8****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_8**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0120 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_9****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_9**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0124 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_10**

**Type:** W  
**Clock:** AHB\_CLK  
**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_10**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0128 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_11**

**Type:** W  
**Clock:** AHB\_CLK  
**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_11**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B012C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_12**

**Type:** W  
**Clock:** AHB\_CLK  
**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_12**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0130 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_13**

**Type:** W  
**Clock:** AHB\_CLK  
**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_13**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0134 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_14****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_14**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0138 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_15****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_15**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B013C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_16****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_16**

Bits	Name	Description
31:0	UART_TF	UART_DM_TF register.

**0x0C1B0140 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined

The UART\_DM\_RF register is the UART receive FIFO register, which is used to access the channel receive FIFO. In order to support bursts of 16, 0x140-0x17C address space is reserved for the UART\_DM\_RX FIFO.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF**

Bits	Name	Description
31:0	UART_RF	This register returns the next word in the receive FIFO. After the read is completed, the FIFO read pointer is updated to make the next word available.

**0x0C1B0144 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_2****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_2**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0148 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_3****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_3**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B014C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_4****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_4**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0150 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_5****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_5**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0154 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_6****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_6**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0158 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_7****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_7**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B015C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_8****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_8**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0160 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_9****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_9**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0164 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_10****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_10**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0168 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_11****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_11**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B016C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_12****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_12**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0170 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_13****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_13**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0174 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_14****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_14**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0178 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_15****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_15**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B017C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_16****Type:** R**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_16**

Bits	Name	Description
31:0	UART_RF	UART_DM_RF register.

**0x0C1B0184 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_CMD****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

Command register for UIM-controller (linked with this UART\_DM). Writing to this register requires waiting until UIM-WRITE is done (indication by status-bit or interrupt) before attempting an additional write/read one of UIM\_CFG/UIM\_CMD registers.

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_CMD**

Bits	Name	Description
1	RECOVER_FROM_HW_DEACTIVATION	Writing '1' to this bit will cause the UIM-io-ctrl to exit override state and reflect UIM signals from UART_DM to the UIM card. Setting both bits of this register to '1' will cause neither of the commands to be executed.
0	INITIATE_HW_DEACTIVATION	Initiates a HW deactivation sequence when writing '1' to this bit. This executes the sequence on the uim_data, uim_clk & uim_rst_n lines and alerts the PMIC to cut the power-supply to the card afterwards. Setting both bits of this register to '1' will cause neither of the commands to be executed.

**0x0C1B0188 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000002**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS**

Bits	Name	Description
2	UIM_IO_WRITE_IN_PROGRESS	Asserted after UIM-write (UIM_CFG/UIM_CMD register write) while write is being propagated into the UIM-IO-controller. An additional read/write to UIM_CFG/UIM_CMD register should be done while high. De-asserted when write finishes propagating.
1	UIM_DEACTIVATION_STATUS	Indicates whether a HW deactivation sequence was performed and sequence recovery hasn't been executed yet. This bit is set when the sequence is initiated either by a HW trigger (batt_alarm, card-removal) or a SW trigger ('initiate_shutdown_sequence' bit in CMD register). Cleared after executing a 'recover_from_hw_deactivation' command.
0	CARD_PRESENCE	Indication on card presence in slot. High when card present, low otherwise.

**0x0C1B018C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Asserted when UIM-write had finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
3	HW_SEQUENCE_FINISH	Asserted when UIM-controller's HW-deactivation sequence has been initiated and finished. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
2	BATT_ALARM	Set to high when battery-alarm indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
1	UIM_CARD_INSERTION	Set to high when card insertion indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.
0	UIM_CARD_REMOVAL	Set to high when card removal indication received. Cleared by writing '1' to matching field in UART_DM_UIM_IRQ_CLR.

**0x0C1B0190 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR****Type:** R**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	See UART_DM_UIM_IRQ_ISR for field descriptions.
3	HW_SEQUENCE_FINISH	See UART_DM_UIM_IRQ_ISR for field descriptions.
2	BATT_ALARM	See UART_DM_UIM_IRQ_ISR for field descriptions.
1	UIM_CARD_INSERTION	See UART_DM_UIM_IRQ_ISR for field descriptions.
0	UIM_CARD_REMOVAL	See UART_DM_UIM_IRQ_ISR for field descriptions.

**0x0C1B0194 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
3	HW_SEQUENCE_FINISH	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
2	BATT_ALARM	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
1	UIM_CARD_INSERTION	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.
0	UIM_CARD_REMOVAL	Clears matching bit in UART_DM_UIM_IRQ_ISR/MISR.

**0x0C1B0198 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR****Type:** RW**Clock:** AHB\_CLK**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
3	HW_SEQUENCE_FINISH	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
2	BATT_ALARM	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
1	UIM_CARD_INSERTION	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.
0	UIM_CARD_REMOVAL	When set, matching bit in UART_DM_UIM_IRQ_MISR will reflect the matching bit in UART_DM_UIM_IRQ_ISR. Otherwise, the matching bit in UART_DM_UIM_IRQ_MISR will be a constant '0'.

**0x0C1B019C PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to set the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1B01A0 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR**

Bits	Name	Description
4	UIM_IO_WRITE_DONE	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
3	HW_SEQUENCE_FINISH	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
2	BATT_ALARM	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
1	UIM_CARD_INSERTION	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.
0	UIM_CARD_REMOVAL	Write '1' to clear the matching bit in UART_DM_UIM_IRQ_IMR.

**0x0C1B00E0 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR\_CLR****Type:** W**Clock:** AHB\_CLK**Reset State:** Undefined

New added interrupt clear register

**PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR\_CLR**

Bits	Name	Description
17	NO_FINISH_CMD_VIOL	Clears NO_FINISH_CMD_VIOL bit in UART_DM_ISR

**0x0C1B5000 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- a. The contents of this register should only be changed when in the RESET\_STATE.
- b. The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup wil not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C1B5004 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C1B5008 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.
- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.

c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_IO\_MODES**

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C1B500C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C1B5014 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is sceduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C1B5018 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C1B501C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C1B5020 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E N	If set, enables input over run error generation.

**0x0C1B5028 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C1B5100 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C1B5104 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C1B510C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C1B5110+PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc, c=[0..15]  
0x4\*c****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C1B5150 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C1B5154 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C1B5200 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input

transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

#### **0x0C1B5204 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_INPUT\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

#### **0x0C1B5208 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO

(BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

#### **0x0C1B520C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

#### **0x0C1B5214 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFO\_WORD\_CNT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C1B5218+PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C1B5300 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_CONFIG**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C1B5304 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C1B5308 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

#### **0x0C1B530C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C1B5310 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C1B5314 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C1B5318 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C1B531C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C1B5330 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C1B5334 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C1B5338 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>.

**0x0C1B5400 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clock edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C1B5404 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C1B5408 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C1B540C PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C1B5410 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C1B5414 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C1B6000 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- The contents of this register should only be changed when in the RESET\_STATE.
- The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup wil not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C1B6004 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C1B6008 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.
- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.

c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_IO\_MODES**

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C1B600C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C1B6014 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is sceduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C1B6018 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C1B601C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C1B6020 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR _EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_ EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E N	If set, enables input over run error generation.

**0x0C1B6028 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C1B6100 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C1B6104 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C1B610C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C1B6110+PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc, c=[0..15]  
0x4\*c****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C1B6150 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_WRITE\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C1B6154 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_WRITE\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C1B6200 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input

transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

#### **0x0C1B6204 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_INPUT\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

#### **0x0C1B6208 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO

(BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

#### **0x0C1B620C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

#### **0x0C1B6214 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFO\_WORD\_CNT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C1B6218+PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFOC, c=[0..15]**  
**0x4\*c**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFOC**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C1B6300 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_CONFIG**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C1B6304 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C1B6308 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

#### **0x0C1B630C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C1B6310 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C1B6314 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C1B6318 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C1B631C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C1B6330 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C1B6334 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C1B6338 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>. Unused bits may be garbage from previous word in case N!=32

**0x0C1B6400 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clock edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C1B6404 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C1B6408 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C1B640C PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C1B6410 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C1B6414 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C1B7000 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- a. The contents of this register should only be changed when in the RESET\_STATE.
- b. The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamical.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup wil not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C1B7004 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C1B7008 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.
- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.

c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_IO\_MODES**

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C1B700C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C1B7014 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is sceduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C1B7018 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C1B701C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C1B7020 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR_EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E N	If set, enables input over run error generation.

**0x0C1B7028 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C1B7100 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C1B7104 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C1B710C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C1B7110+PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc, c=[0..15]  
0x4\*c**

**Type:** W  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C1B7150 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C1B7154 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_WRITE\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C1B7200 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input

transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

#### **0x0C1B7204 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_INPUT\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

#### **0x0C1B7208 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO

(BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

#### **0x0C1B720C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_READ\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

#### **0x0C1B7214 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C1B7218+PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C1B7300 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_CONFIG**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C1B7304 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C1B7308 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

#### **0x0C1B730C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C1B7310 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C1B7314 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C1B7318 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C1B731C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C1B7330 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C1B7334 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C1B7338 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>. Unused bits may be garbage from previous word in case N!=32

**0x0C1B7400 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns =~ MAX(Tsu;DAT)) the clkok edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C1B7404 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C1B7408 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C1B740C PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C1B7410 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C1B7414 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

**0x0C1B8000 PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

- The contents of this register should only be changed when in the RESET\_STATE.
- The value written to the QUP output FIFO, is shifted left toward the MSB before passing it to the mini-core as follows:
  - o N equals 8 or less - shift 24
  - o N equals 16 to 9 - shift 16
  - o N equals 24 to 17 - shift 8
  - o N equals 32 to 25 - no shift

**PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_CONFIG**

Bits	Name	Description
17	DIS_INBUF_FLAG_FIX	when set to 1 disable logic fix to basi_trans_end_pro logic in qup_input_buffer. fix basi protocol viaolation, increment basi_valid_data when basi_trans_end_pro asserted.
16	EN_EXT_OUT_FLAG	<p>Enable Extended OUTPUT/INPUT_SERVICE_FLAG interrupt generation.</p> <p>a) For SPI:</p> <ol style="list-style-type: none"> <li>1. The shifting needs to complete</li> <li>2. Clock stopped if CLK_ALWAYS_ON=0,</li> <li>3. CS, if used, de-asserted</li> </ol> <p>b) For I2C,</p> <ol style="list-style-type: none"> <li>1. The ACK needs to complete</li> <li>2. STOP condition done generated (Attention need to be put in synchronizing the last transaction with STOP tag )</li> <li>3. clock and data wires return to high condition</li> </ol> <p>When this bit is clear, the legacy behavior is applicable. Under legacy, the DONE_FLAG is set when :</p> <ol style="list-style-type: none"> <li>1. MX_*_COUNT is reached</li> <li>2. The output FIFO is empty</li> <li>3. Last bit was sent on the link</li> </ol> <p>Under legacy, the flag interrupt is generated before the last transfer is complete on the external interface.</p>
15	CORE_EXTRA_CLK_ON_EN	Enable additional transfer based qup_core_clk gating for power save. When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on.
14	FIFO_CLK_GATE_EN	When set to 0, fifo clock is mostly on - legacy mode. When set to 1, fifo clock is turned off dynamically.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_CONFIG (cont.)**

Bits	Name	Description
13	CORE_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
12	APP_CLK_ON_EN	When set to 0, clock on is turned on unconditionally. When set to 1, dynamic clock on is turned on and driven high conditionally.
11:8	MINI_CORE	value: 0000 null core value: 0001 SPI core value: 0010 I2C master controller value: 0011 reserved value: 0100 SPI Slave value: 0101 reserved value: 0110 reserved value: 0111 reserved
7	NO_INPUT	The QUP input FIFO is always empty. The input service interrupt and INPUT_SERVICE_FLAG bit are never set.
6	NO_OUTPUT	The QUP output FIFO is always empty. The output service flag and OUTPUT_SERVICE_FLAG bit are never set. qup_data_out is still driven when SPI_CS#_N is asserted. The setting for NO_TRI_STATE still applies.
5	QUP_HREADY_CTRL	When set to 0, qup wil not stall the AHB bus (legacy mode). When set to 1, qup may stall the AHB bus until register access is done.
4:0	N	Number of logical bits N in the mini-core that constitutes a single transfer. The value zero indicates N equals one. The value of all ones indicates N equals 32. The value in this register must be 3 (i.e., N is 4 or higher in order for the STATE field to be set to RUN_STATE).

**0x0C1B8004 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_STATE****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000001C**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_STATE**

Bits	Name	Type	Description
8	SPI_S_GEN	R	Read only. Reflects that QUP has instance of spi slave mini core.
6	I2C_FLUSH	RW	Flusing an i2c transfer requires using version 2 tags. FLUSH should be used in BAM mode only. Setting this bit to 1 will flush all tags and tag related data besides EOT until FLUSH STOP tag is encountered. Setting this bit to 0 has no impact. Reading this bit returns the flush operation status - 0 = ongoing, 1 = done.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_STATE (cont.)**

Bits	Name	Type	Description
5	WAIT_FOR_EOT	RW	Only applicable when moving to RUN_STATE using command descriptor in SPI mode with NO_OUTPUT. Setting this bit to 1 will stall the done_toggle_com indication until basi_trans_end_pro is asserted. This will prevent the next command descriptor to start prematurely.
4	I2C_MAST_GEN	R	Read only. Reflects the RTL generic setting for GEN_I2C_MASTER_MINI_CORE.
3	SPI_GEN	R	Read only. Reflects the RTL generic setting for GEN_SPI_MINI_CORE.
2	STATE_VALID	R	Read only. If and only if set to one, writes to the STATE field is allowed or reads from the STATE field is valid. Writes to this bit is ignored.
1:0	STATE	RW	When clear (00), the mini-core and related logic is held in RESET_STATE. When set to '01', the mini-core and related logic is released from reset and enters the RUN_STATE. When set to '11', the mini-core and related logic enters the PAUSE_STATE at the next appropriate point in time. Writing (10) to this field clears these two bits. For PAUSE_STATE to RESET_STATE transition, two writes of (10) are required for the transition to complete.

**0x0C1B8008 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_IO\_MODES****Type:** RW**Clock:** crif\_clk**Reset State:** 0x000000A5

Unless otherwise stated, register bits written return the value when read.

NOTE:s:

a. 'Packing' occurs as follows:

- N equals 8 or less - pack four values into each QUP input FIFO word.
- N equals 16 to 9 - pack two values into each QUP input FIFO word.
- N equals 32 to 17 - no packing. Each mini-core value is moved to an QUP input FIFO word.

b. 'Un-Packing' occurs as follows:

- N equals 8 or less - un-pack four values from each QUP output FIFO word.
- N equals 16 to 9 - un-pack two values from each QUP output FIFO word.
- N equals 32 to 17 - no packing. Each QUP output FIFO value is moved to the mini-core.

c. INPUT\_MODE and OUTPUT\_MODE should be both in BAM\_Mode or both in non BAM\_Mode.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_IO\_MODES**

Bits	Name	Type	Description
16	OUTPUT_BIT_SHIFT_EN	RW	If set, enables the QUP output FIFO block to do bit shifting on the output data.
15	PACK_EN	RW	Indicates data values from the mini-core are to be packed before placement in the QUP input FIFO.
14	UNPACK_EN	RW	Indicates data values taken from the QUP output FIFO should be unpacked before passing to the mini-core.
13:12	INPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_INPUT = 0, then only FIFO_Mode is available)
11:10	OUTPUT_MODE	RW	Value 00: FIFO_Mode Value 01: Block_Mode Value 10: Reserved Value 11: BAM_Mode (NOTE: if the RTL generic BLOCK_SIZE_OUTPUT = 0, then only FIFO_Mode is available)
9:7	INPUT_FIFO_SIZE	R	Read only, actual value set by RTL generic. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
6:5	INPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL generic. Indicates the block size associated with Block_Mode for input. Value 00: 4 Bytes (FIFO_Mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved
4:2	OUTPUT_FIFO_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 000: 2X BLOCK_SIZE Value 001: 4X BLOCK_SIZE Value 010: 8X BLOCK_SIZE Value 011: 16X BLOCK_SIZE
1:0	OUTPUT_BLOCK_SIZE	R	Read only. Actual value set by RTL GENERIC. Value 00: 04 Bytes (FIFO mode only) Value 01: 16 Bytes Value 10: 32 Bytes Value 11: Reserved

**0x0C1B800C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_SW\_RESET****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000

A write to this register resets the entire QUP core and all mini-cores. The internal registers are brought back to their reset values. Reading of this register returns zero. The AHB clock domain reset will stay asserted until either the spi\_clk or i2c\_clk domains are reset. It is prohibited to write to any QUP register during this period and failing to do so will cause an ERROR response on the AHB bus. In order to avoid this SW should poll the QUP\_STATE[STATE\_VALID] bit until it is asserted

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_SW\_RESET**

Bits	Name	Description
1:0	QUP_SW_RESET	Writing 2'b01 - performs single core clock pulse reset Writing 2'b10 - performs long pulse reset Other values are reserved and will not cause reset.

**0x0C1B8014 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_TRANSFER\_CANCEL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_TRANSFER\_CANCEL**

Bits	Name	Type	Description
15:8	TRANSFER_CANCEL_ID	RW	This field include the id of the transfer that is sceduled for cancellation. This field is valid when asserting TRANSFER_CANCEL bit. A value of 0xFF will cancel the current transfer regardless of its LOCAL_ID. Canceling an i2c transfer requires using version 2 tags.
7	TRANSFER_CANCEL	W	Setting this bit to 1 will cause the HW to go to pause_state when the LOCAL_ID = TRANSFER_CANCEL_ID. This would allow flushing the rest of the transfer in order to implement cancel API. Canceling an i2c transfer requires using version 2 tags.

**0x0C1B8018 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATIONAL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000000C0**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATIONAL**

Bits	Name	Description
15	NWD	Notify When Done (NWD). Read only status bit which when set indicates a NWD acknowledgement is outstanding. This bit is set by QUP hardware when NWD is signaled from BAM. This bit is cleared by QUP hardware when the NWD request is acknowledged by assertion of done_toggle. This bit is always unconditionally cleared in RESET_STATE.
14	DONE_TOGGLE	Read only status bit which provides the current state of the side-band done_toggle signal sent to BAM. At each NWD acknowledgement, this bit toggles.
13	IN_BLOCK_READ_REQ	Read only. When set by hardware, indicates QUP input FIFO has BLOCK_SIZE_INPUT amount of data ready for reading. Valid only in Block_Mode.
12	OUT_BLOCK_WRITE_REQ	Read only. When set by hardware, indicates QUP output FIFO needs BLOCK_SIZE_OUTPUT amount of data to be written. Valid only in Block_Mode.
9	INPUT_SERVICE_FLAG	When set by hardware, indicates QUP input FIFO has an outstanding input service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
8	OUTPUT_SERVICE_FLAG	When set by hardware, indicates QUP output FIFO has an outstanding output service request. At the point in time this bit is set, the hardware also asserts qup_irq. Writing a 'zero' to this bit does nothing. Writing a 'one' to this bit clears it and acknowledges software has or will read the data. Valid in all modes , recommended to be masked in BAM mode
7	INPUT_FIFO_FULL	Read only. When set, indicates the input FIFO is full and can accept no more data from the QUP mini-core.
6	OUTPUT_FIFO_FULL	Read only. When set, indicates the output FIFO is full and can accept no more CRIF writes.
5	INPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the input FIFO has at least one value in it to be read. When clear, indicates the input FIFO is empty.
4	OUTPUT_FIFO_NOT_EMPTY	Read only. When set, indicates the output FIFO has at least one value in it to be shifted out. When clear, indicates the output FIFO is empty.

**0x0C1B801C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR	Indicates the output FIFO was full when a CRIF write was attempted to the FIFO. The write is discarded.
4	INPUT_UNDER_RUN_ERR	Indicates the input FIFO was empty when a CRIF read was attempted to the FIFO. The read value returns is indeterminate.
3	OUTPUT_UNDER_RUN_ER R	Indicates the output FIFO was empty when an output shift operation required a value.
2	INPUT_OVER_RUN_ERR	Indicates the input FIFO was full when an input shift operation provided a new value. When this happens, the new value is discarded.

**0x0C1B8020 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0000003C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of qup\_irq and the setting of the corresponding error flag in the QUP\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN**

Bits	Name	Description
5	OUTPUT_OVER_RUN_ERR _EN	If set, enables output over run error generation.
4	INPUT_UNDER_RUN_ERR_ EN	If set, enables input under run error generation.
3	OUTPUT_UNDER_RUN_ER R_EN	If set, enables output under run error generation.
2	INPUT_OVER_RUN_ERR_E N	If set, enables input over run error generation.

**0x0C1B8028 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATIONAL\_MASK****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register masks several QUP\_OPERATIONAL flags from creating an interrupt .

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATIONAL\_MASK**

Bits	Name	Description
9	INPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.
8	OUTPUT_SERVICE_MASK	If set, this flag in QUP_OPERATIONAL does not cause an interrupt but provides status only.

**0x0C1B8100 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode. The counter decrements for each output transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE doe not effect the count.

NOTE:s:

- a. Allows the total number of Mini Core transfers to be less than an exact multiple of OUTPUT\_BLOCK\_SIZE. Any additional outputs are discarded.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_OUTPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_OUTPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_OUTPUT_COUNT	RW	Number of writes of size N to the mini-core per RUN_STATE. A value of zero indicates the output count function is not enabled for use.

**0x0C1B8104 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUTPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_OUTPUT_CNT_CURRENT	Current value of QUP_MX_OUTPUT_COUNT.

**0x0C1B810C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

This register holds the number of words in the output FIFO at a given time. NOTE: the fields in this register are dynamically updated hence when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	OUTPUT_FIFO_WORD_CNT	Number of words in the output FIFO.

**0x0C1B8110+PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc, c=[0..15]  
0x4\*c**

**Type:** W  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

NOTE: that consecutive writes to this address keeps filling up the output FIFO. The max address to address the output FIFO is 0x14C.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc**

Bits	Name	Description
31:0	OUTPUT	Value to be shifted out.

**0x0C1B8150 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

What the QUP\_MX\_OUTPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_OUTPUT\_COUNT register case, the SW should not program the QUP\_MX\_WRITE\_COUNT value to be more than the 'actual depth' of the FIFO (BLOCK\_SIZE\_OUTPUT \* FIFO\_SIZE\_OUTPUT).

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_WRITE_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_WRITE_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance.
15:0	MX_WRITE_COUNT	RW	The number of 'writes' of size N. This is used only if the core is in FIFO_Mode.

**0x0C1B8154 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_WRITE\_CNT\_CURRENT****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_WRITE\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_WRITE_CNT_CURRENT	Current value of QUP_MX_WRITE_COUNT counter.

**0x0C1B8200 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read. This register only applies to Block\_Mode and Bam\_Mode (for non-balanced SPI). The counter decrements for each input

transfer when the STATE field is moved from the RESET\_STATE to the RUN\_STATE. The PAUSE\_STATE does not affect the count.

NOTE:s:

- a. Allows the number of shift register transfers to be less than an exact multiple of INPUT\_BLOCK\_SIZE. When count reached, remainder of INPUT\_BLOCK\_SIZE is filled with zeroes.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_INPUT_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_INPUT_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_INPUT_COUNT	RW	Number of reads of size N from the mini-core per RUN_STATE. A value of zero indicates the input count function is not enabled for use.

#### **0x0C1B8204 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_INPUT\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_INPUT\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_INPUT_CNT_CURRENT	Current value of QUP_MX_INPUT_COUNT.

#### **0x0C1B8208 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_READ\_COUNT**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x00000000

What the QUP\_MX\_INPUT\_COUNT register means to Block\_Mode and Bam\_Mode, this register means the same to FIFO\_mode. If this register is non-zero, then the qup\_input\_service\_irq is asserted after shifting in the number of shifts specified by this register.

Unlike the QUP\_MX\_INPUT\_COUNT register case, the SW should not program the QUP\_MX\_READ\_COUNT value to be more than the 'actual depth' of the FIFO

(BLOCK\_SIZE\_INPUT \* FIFO\_SIZE\_INPUT). If the SPI mini-core and slave operation is enabled, then an InputOverRun error will result.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_MX\_READ\_COUNT**

Bits	Name	Type	Description
31	MX_CONFIG_DURING_RUN	W	Write only value. When reset to zero the MX_READ_COUNT value only applies in reset_state (legacy behavior). When set to one the MX_READ_COUNT value will also apply in run_state and allow writing the transaction data in a few portions when not knowing the transaction size in advance. Reconfiguration during run is only allowed after the last portion has ended (i.e. MAX_INPUT_DONE_FLAG was asserted)
15:0	MX_READ_COUNT	RW	The number of 'reads' of size N. This is used only if the core is in FIFO_Mode.

#### **0x0C1B820C PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_MX\_READ\_CNT\_CURRENT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_MX\_READ\_CNT\_CURRENT**

Bits	Name	Description
15:0	MX_READ_CNT_CURRENT	Current value of QUP_MX_READ_COUNT counter.

#### **0x0C1B8214 PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_INPUT\_FIFO\_WORD\_CNT**

**Type:** R

**Clock:** crif\_clk

**Reset State:** 0x00000000

This register holds the number of words in the input FIFO at a given time. NOTE: the fields in this register are dynamically updated. Hence, when this register is read during a FIFO operation, the value of this register could have been altered by the time SW gets the read value.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUPI3\_QUPI\_INPUT\_FIFO\_WORD\_CNT**

Bits	Name	Description
8:0	INPUT_FIFO_WORD_CNT	Number of words in the input FIFO.

**0x0C1B8218+PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_INPUT\_FIFOc, c=[0..15]**  
**0x4\*c**

**Type:** R  
**Clock:** crif\_clk  
**Reset State:** Undefined

Consecutive reads to this address reads the input FIFO contents on a FIFO basis. The max address to read the input FIFO is 0x254.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_INPUT\_FIFOc**

Bits	Name	Description
31:0	INPUT	Value shifted in.

**0x0C1B8300 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_CONFIG**

**Type:** RW  
**Clock:** crif\_clk  
**Reset State:** 0x00000000

Unless otherwise stated, register bits written return the value when read.

The contents of this register should only be changed when in the RESET\_STATE. Both NO\_OUTPUT and NO\_INPUT set to zero provides default full duplex operation. Setting both these bits to one is not a legal operational state.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_CONFIG**

Bits	Name	Type	Description
10	HS_MODE	RW	When set, SPI HS_MODE is enabled. When clear, SPI HS_MODE is disabled.
9	INPUT_FIRST	RW	When set, INPUT FIRST SPI protocol is used. When clear, OUTPUT FIRST SPI protocol is used.
8	LOOP_BACK	RW	Loopback is only valid in non-HS mode. Always clear for normal operation. If set, loop back on SPI_DATA_MO_SI under MASTER operation or SPI_DATA_MI_SO under SLAVE operation.
5	SLAVE_OPERATION	R	This register is writable only when the hardware signal spi_slave_en is asserted. When set, the SPI is configured for SLAVE operation. Zero indicates MASTER operation.

**0x0C1B8304 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_IO\_CONTROL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The contents of this register should only be changed when in the RESET\_STATE.

Unless otherwise stated, register bits written return the value when read.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_IO\_CONTROL**

Bits	Name	Description
11	FORCE_CS	When this bit is set, the chip select is asserted unconditionally with no relationship to QUP_STATE or transaction state.
10	CLK_IDLE_HIGH	When set to 1, spi_clk will be high when spi is idle (spi_cs is not asserted). When cleared to 0, spi_clk will be low when spi is idle (spi_cs is not asserted).
9	CLK_ALWAYS_ON	When MASTER operation is used, run SPI_CLK during IDLE.
8	MX_CS_MODE	If this bit is set, then for a given RUN state, the associated chip select will be asserted for the first N-bit transfer and will be kept asserted till the last programmed transfer. Applies only in MASTER mode. The number of transfers is determined by SPI_MX_OUTPUT_COUNT and/or SPI_MX_INPUT_COUNT registers.
7:4	CS_N_POLARITY	These four bits control the polarity of four SPI_CS#_N respectively. Setting any of this bit to '1', makes the associated SPI_CS#_N active HIGH. This field is a 'don't care' in SLAVE operation.
3:2	CS_SELECT	When MASTER operation is used, controls the assertion of SPI_CS#_N pins. Selects SPI_CS_N if 00, SPI_CS1_N if 01, SPI_CS2_N if 10, or SPI_CS3_N if 11. The deselected ChipSelects will be driven to inactive state dictated by the field CS_N_POLARITY. This field is a 'don't care' in SLAVE operation.
1	TRISTATE_CS	When set, drives Z on all SPI_CS#_N lines. When clear, enables normal functionality on these lines. This bit can be used to support deployment of the core as one of the masters in a multi-master configuration.
0	NO_TRI_STATE	When set, spi_data_out is not taken tri-state when SPI_CS#_N is de-asserted. This bit is normally set for MASTER operation but may be optionally left cleared.

**0x0C1B8308 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

All bits in this register are set by hardware and remain set until cleared by software. Writing a 'one' to a bit clears it while writing a 'zero' leaves the bit unchanged.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE	This bit is set (1) when the SPI controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
2	TRANSFER_CANCEL_ID_MATCH	This bit is set (1) when the SPI controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.
1	SPI_SLV_CLK_OVER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was greater than the programmed value of N. When this happens, only the first N bits are passed to the SPI input FIFO and the output shift value is held at zero.
0	SPI_SLV_CLK_UNDER_RUN_ERR	Number of SPI_CLK ticks which occurred while SPI_CS#_N was asserted was less than the programmed value of N. When this occurs, the bits which were received are passed to the SPI input FIFO and the CURRENT output bit is forced to zero.

#### **0x0C1B830C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN**

**Type:** RW

**Clock:** crif\_clk

**Reset State:** 0x0000000C

Each bit in this register controls the setting of a corresponding error flag. If set, each bit enables generation of spi\_error\_irq and the setting of the corresponding error flag in the SPI\_ERROR\_FLAGS register for the specified error case. At reset, all error enable bits are set to '1'.

#### **PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN**

Bits	Name	Description
3	TRANSFER_CANCEL_DONE_EN	If set, enables generating the transfer cancel done indication.
2	TRANSFER_CANCEL_ID_MATCH_EN	If set, enables generating the transfer cancel ID match indication.
1	SPI_SLV_CLK_OVER_RUN_ERR_EN	If set, enables clock over run error generation.
0	SPI_SLV_CLK_UNDER_RUN_ERR_EN	If set, enables clock under run error generation.

**0x0C1B8310 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

This register holds the de-assertion wait time of SPI\_CS#\_N between consecutive N-bit transfers in MASTER operation. The wait time is specified in number of cycles of cc\_spi\_master\_clk.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT**

Bits	Name	Description
5:0	DEASSERT_WAIT	Number cc_spi_master_clk ticks associated with the deasserted time of SPI_CS#_N. Only applies to MASTER operation. For SLAVE operation, this field is a 'don't care'. A value of zero indicates SPI_CS#_N remains de-asserted for exactly one clock tick. A value of one, indicates two ticks, etc. All ones indicates 64 ticks.

**0x0C1B8314 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID**

Bits	Name	Description
15:8	EXTENDED_ID	SW may use this field to specify the current sub-transfer ID to be read when TRANSFER_CANCEL_ID_MATCH is asserted. This value has no impact on HW.
7:0	LOCAL_ID	SW specifies the current transfer ID. HW will use this value to match with TRANSFER_CANCEL_ID.

**0x0C1B8318 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_COMMAND****Type:** W**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command.

**0x0C1B831C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_STATUS**

Bits	Name	Description
2:0	CANCEL_FSM_STATE	This 3-bit field informs the microprocessor of the state of the SPI cancel logic.

**0x0C1B8330 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt status. writing 1 will clear the status

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS**

Bits	Name	Description
6	CS_N_ERXT	Early RX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
5	RX_OVERFLOW_NO_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when not waiting to EOT - Causes error interrupt
4	RX_OVERFLOW_WAIT_EOT	Indication that RX tried to write data to internal buffer but the buffer was full when waiting to EOT
3	TX_UNDERFLOW	Indication that TX tried to read data from internal buffer but the buffer was empty - Causes error interrupt
2	CS_N_ETXT	Early TX termination due to chip select assertion - Causes error interrupt depending on PAUSE_ON_ERR_DIS
1	CS_N_DEASSERT	chip select de-assertion
0	CS_N_ASSERT	chip select assertion

**0x0C1B8334 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000

SPI SLAVE Interrupt Enable.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN**

Bits	Name	Description
6	CS_N_ERXT_EN	Enable for CS_N_RTXT
5	RX_OVERFLOW_NO_EOT_EN	Enable for RX_OVERFLOW_NO_EOT
4	RX_OVERFLOW_WAIT_EOT_EN	Enable for RX_OVERFLOW_WAIT_EOT
3	TX_UNDERFLOW_EN	Enable for TX_UNDERFLOW
2	CS_N_ETXT_EN	Enable for CS_N_ETXT
1	CS_N_DEASSERT_EN	Enable for CS_N_DEASSERT
0	CS_N_ASSERT_EN	Enable for CS_N_ASSERT

**0x0C1B8338 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_CFG****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_CFG**

Bits	Name	Description
31:8	NOT_INUSE	Reserved bits in register for future use.
7	SLAVE_AUTO_PAUSE_EOT	Setting this bit to 1 will cause SPI Slave Mini Core to enter pause state after EOT.
6:5		
4	SLAVE_DIS_RESET_ST	When set SPI Slave Mini Core will not perform sync reset in its own reset state. core will move from transit state to reset state with no reset.
3	RX_UNBALANCED_MASK	When set QUP Input FIFO will ignore unbalanced condition in case of RX only
2	SPI_S_CGC_EN	When set internal CGC for SPI_CLK_IN inside SPI_SLAVE will be always on
1	PAUSE_ON_ERR_DIS	When RX_OVERFLOW_NO_EOT or TX_UNDERFLOW or CS_N_ETXT occurs mini_core will enter pause state. setting the bit to 1 will disable moving to pause upon CS_N_ETXT
0	RX_N_SHIFT	When set RX side will push the data to QUP fifo as follows: <N, unused bits>. Default format <unused bits, N>. Unused bits may be garbage from previous word in case N!=32

**0x0C1B8400 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

The I2C\_CLK\_CTL register is a read/write register that controls clock divider values.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL**

Bits	Name	Description
28	SCL_EXT_FORCE_LOW	When set to 0, SCL generation maintains legacy behavior. When set to 1, SCL state machine will go to the FORCED_LOW_STATE if force_low is asserted near (within 5 5 i2c_clk cycles = 5 * 52ns = 260ns ≈ MAX(Tsu;DAT)) the clock edge.
27:26	SDA_NOISE_REJECTION	Allows adding extra sampling levels on SDA to reject short low pulses. This value specifies how many TCXO cycles of logic low on SDA would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
25:24	SCL_NOISE_REJECTION	Allows adding extra sampling levels on SCL to reject short low pulses. This value specifies how many TCXO cycles of logic low on SCL would be considered as valid logic low. 0x0 - legacy mode, 0x01 - one cycle wide low pulse is rejected, 0x2 - two cycles wide low pulse is rejected, 0x3 - three cycles wide low pulse is rejected
23:16	HIGH_TIME_DIVIDER_VALUE	Allows setting SCL duty cycle to non 50%. If this value is zero than legacy mode is used.
7:0	FS_DIVIDER_VALUE	The value in this register represents the clock period multiplier in fast/standard (FS) mode. Minimum value is 0x7. When HIGH_TIME_DIVIDER_VALUE=0: I2C_FS_CLK = I2C_CLK/(2*(FS_DIVIDER_VALUE+3)) When HIGH_TIME_DIVIDER_VALUE!=0: I2C_FS_CLK = I2C_CLK/(FS_DIVIDER_VALUE+HIGH_TIME_DIVIDER_VALUE+6)

**0x0C1B8404 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_STATUS****Type:** RW**Clock:** crif\_clk**Reset State:** 0x0C000000

The I2C\_STATUS is a status register. Writing any value clears the status bits.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_STATUS**

Bits	Name	Type	Description
27	I2C_SCL	R	Logic state of I2C bus serial clock wire.
26	I2C_SDA	R	Logic state of I2C bus serial data wire.
25	INVALID_READ_SEQ	RW	Interrupt source. This bit is set (1) when a MI_REC tag does not follow a START tag for an I2C READ.
24	INVALID_READ_ADDR	RW	Interrupt source. In version 1 tags this bit is set (1) when the I2C controller is trying to receive data from a non-existent I2C slave (address). In version 2 tags this bit is set (1) when the I2C controller is trying to access a non-existent I2C slave (address).
23	INVALID_TAG	RW	Interrupt source. This bit is set (1) when the I2C controller is trying to process data from the output FIFO that is improperly tagged.
9	BUS_MASTER	R	This bit is set (1) when the I2C controller is the present bus master.
8	BUS_ACTIVE	R	This bit is set (1) when the bus is in use by this, or any other controller.
7:6	FAILED	RW	Interrupt source. This 2-bit field contains the failure information of the present I2C transfers. If transmitting, failed[1] contains the information regarding the byte that has been transmitted. Failed[0] contains the information of the byte awaiting transmission if there is a byte pipelined. If no byte is pipelined, ignore this bit. If receiving, failed[1] contains the information of the byte received and stored in the buffer, and failed[0] should be ignored. For example: Value 00: Byte n transmitted successfully, byte n+1 to begin transmission Value 01: Byte n transmitted successfully, byte n+1 errored: type of error indicated in other STATUS bits. (queued invalid write, for example) Value 10: Byte n errored: type of error indicated in other STATUS bits, byte n+1 to begin transmission (byte n+1 would have to be a valid address byte for this condition to occur) Value 11: Byte n errored: type of error indicated in other STATUS bits, byte n+1 discarded as well (if the byte was a data byte destined for a NACKed address, data is useless, therefore discarded)
5	INVALID_WRITE	RW	Interrupt source. This bit is set (1) when software writes data to the I2C_DATA register that should be flagged as an address but is not.
4	ARB_LOST	RW	Interrupt source. This bit is set (1) when the controller loses arbitration for the bus. If this bit gets set (1) during the transmission, all data has been lost, and the microprocessor must re-request its transmission.
3	PACKET_NACKED	RW	Interrupt source.

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_STATUS (cont.)**

Bits	Name	Type	Description
2	BUS_ERROR	RW	Interrupt source. This bit is set (1) when an unexpected START or STOP condition is detected. This returns the controller to its reset state.
1	TRANSFER_CANCEL_DONE	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller has completed canceling the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID).
0	TRANSFER_CANCEL_ID_MATCH	RW	Only applicable when using version 2 tags. Interrupt source. This bit is set (1) when the I2C controller is ready for cancel operation: it has found the transfer that was scheduled for cancellation (LOCAL_ID = TRANSFER_CANCEL_ID) and the logic is done with previous activity.

**0x0C1B8408 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG**

Bits	Name	Description
3	BUSY_INDICATION_SELECT	When set to 0, clock_ctrl FSM maintains legacy behavior. When set to 1, busy logic will monitor the bus and indicate if there is a valid cycle (start-to-stop). clk_ctrl FSM will use it to detect other master transaction and move to NOT_MASTER_STATE without generating bus error
2	SDA_DELAYED_DETECTION	When set to 0, SDA fall/rise detection maintains legacy behavior. When set to 1, it will delay detection by 3 clocks and will compensate for noise reject high period stretching
1	LOW_PERIOD_NOISE_REJECT_EN	When set to 0, noise reject maintains legacy behavior. When set to 1, noise reject on SCL/SDA low level is enabled. Allows adding extra sampling levels on SCL/SDA to reject short low/high pulses. use SCL/SDA_NOISE_REJECTION register to select noise width rejection (in clock cycles)
0		

**0x0C1B840C PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_CLEAR****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000

**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_CLEAR**

Bits	Name	Description
0	CLEAR	This command should be given only when the I2C mincore is idle. When in doubt SW can use SW_RESET to reset the minicore. When set, an I2C 'Bus Clear' executed per the I2C standard. A bus clear consists of nine I2C clock ticks with data wire left not-driven. Has the effect of clearing any slave which has lost read sync. After the clear is complete, the hardware sets the CLEAR bit to zero. The bit can not be cleared by software. After a bus clear, both clock and data lines should be high. If not, an external slave has hung the bus by holding down one or both lines. Reset the slave to clear if possible. Any slave reset mechanism is beyond the scope of the I2C mini core.

**0x0C1B8410 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID****Type:** R**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID**

Bits	Name	Description
7:0	LOCAL_ID	Read only value capturing the ID byte of the last NOP LOCAL tag. Only applicable when using V2 tags

**0x0C1B8414 PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND****Type:** RW**Clock:** crif\_clk**Reset State:** 0x00000000**PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND**

Bits	Name	Description
0	RESET_CANCEL_FSM	This command should be given only when the CANCEL_FSM_STATE is in CANCEL_PENDING_STATE or PAUSE_WAIT_STATE and when the core is in PAUSE. The main use case for this bit is to allow withdrawing a cancel command in case the transfer was flushed (QUP_STATE[I2C_FLUSH] was set) due to I2C NACK.

# 10 Camera subsystem registers

---

**0x0002482C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000000

CSI\_COMMON\_CTRL11

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11**

Bits	Name	Description
7:0	CSI_COMMON_CTRL11	IRQ_MASK[7:0]

**0x00024830 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12**

**Type:** RW

**Clock:** WCLK

**Reset State:** 0x00000000

CSI\_COMMON\_CTRL12

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12**

Bits	Name	Description
7:0	CSI_COMMON_CTRL12	IRQ_MASK[15:8]

**0x00024834 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL13

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13**

Bits	Name	Description
7:0	CSI_COMMON_CTRL13	IRQ_MASK[23:16]

**0x00024838 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL14

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14**

Bits	Name	Description
7:0	CSI_COMMON_CTRL14	IRQ_MASK[31:24]

**0x0002483C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL15

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15**

Bits	Name	Description
7:0	CSI_COMMON_CTRL15	IRQ_MASK[39:32]

**0x00024840 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL16

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16**

Bits	Name	Description
7:0	CSI_COMMON_CTRL16	IRQ_MASK[47:40]

**0x00024844 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL17

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17**

Bits	Name	Description
7:0	CSI_COMMON_CTRL17	IRQ_MASK[55:48]

**0x00024848 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL18

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18**

Bits	Name	Description
7:0	CSI_COMMON_CTRL18	IRQ_MASK[63:56]

**0x0002484C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL19

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19**

Bits	Name	Description
7:0	CSI_COMMON_CTRL19	IRQ_MASK[71:64]

**0x00024850 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL20

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20**

Bits	Name	Description
7:0	CSI_COMMON_CTRL20	IRQ_MASK[79:72]

**0x00024854 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL21

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21**

Bits	Name	Description
7:0	CSI_COMMON_CTRL21	IRQ_MASK[87:80]

**0x00024858 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL22

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22**

Bits	Name	Description
7:0	CSI_COMMON_CTRL22	IRQ_CLEAR[7:0]

**0x0002485C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL23

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23**

Bits	Name	Description
7:0	CSI_COMMON_CTRL23	IRQ_CLEAR[15:8]

**0x00024860 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL24

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24**

Bits	Name	Description
7:0	CSI_COMMON_CTRL24	IRQ_CLEAR[23:16]

**0x00024864 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL25

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25**

Bits	Name	Description
7:0	CSI_COMMON_CTRL25	IRQ_CLEAR[31:24]

**0x00024868 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL26

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26**

Bits	Name	Description
7:0	CSI_COMMON_CTRL26	IRQ_CLEAR[39:32]

**0x0002486C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL27

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27**

Bits	Name	Description
7:0	CSI_COMMON_CTRL27	IRQ_CLEAR[47:40]

**0x00024870 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL28

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28**

Bits	Name	Description
7:0	CSI_COMMON_CTRL28	IRQ_CLEAR[55:48]

**0x00024874 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL29

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29**

Bits	Name	Description
7:0	CSI_COMMON_CTRL29	IRQ_CLEAR[63:56]

**0x00024878 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL30

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30**

Bits	Name	Description
7:0	CSI_COMMON_CTRL30	IRQ_CLEAR[71:64]

**0x0002487C MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL31

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31**

Bits	Name	Description
7:0	CSI_COMMON_CTRL31	IRQ_CLEAR[79:72]

**0x00024880 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL32

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32**

Bits	Name	Description
7:0	CSI_COMMON_CTRL32	IRQ_CLEAR[87:80]

**0x000248B0 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS0

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0**

Bits	Name	Description
7	CSI_COMMON_STATUS0_7	IRQ_LN0_ULPM_EXIT; Lane 0 ULPM exit
6	CSI_COMMON_STATUS0_6	IRQ_LN0_ULPM_ENTRY; Lane 0 ULPM entry
5	CSI_COMMON_STATUS0_5	IRQ_LN0_ULPM_ERR; Lane 0 ULPM error
4	CSI_COMMON_STATUS0_4	IRQ_LN0_ERR_SOT_SYNC; Lane 0 SYNC error correction
3	CSI_COMMON_STATUS0_3	IRQ_LN0_ERR_SOT; Lane 0 SYNC error detection
2	CSI_COMMON_STATUS0_2	IRQ_LN0_CTL_ERR; Lane 0 control error

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0 (cont.)**

Bits	Name	Description
1	CSI_COMMON_STATUS0_1	IRQ_LN0_CMD_ERR; Lane 0 command error
0	CSI_COMMON_STATUS0_0	IRQ_LN0_FIFO_OFLOW; Lane 0 FIFO over flow

**0x000248B4 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS1

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1**

Bits	Name	Description
7	CSI_COMMON_STATUS1_7	IRQ_LN1_HS_ENTRY_ERROR
6	CSI_COMMON_STATUS1_6	IRQ_LN1_HS_EXIT_ERROR
5	CSI_COMMON_STATUS1_5	IRQ_LN1_ESCAPE_ENTRY_ERROR
4	CSI_COMMON_STATUS1_4	IRQ_LN1_ESCAPE_COMMAND_ERROR
3	CSI_COMMON_STATUS1_3	IRQ_LN1_ESCAPE_EXIT_ERROR
2	CSI_COMMON_STATUS1_2	IRQ_LN1_SYNC_BIT_ERROR
1	CSI_COMMON_STATUS1_1	IRQ_LN1_FIFO_OFLOW
0	CSI_COMMON_STATUS1_0	IRQ_LN0_RXSYNC

**0x000248B8 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS2

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2**

Bits	Name	Description
7	CSI_COMMON_STATUS2_7	IRQ_LN2_CTL_ERR; Lane 2 control error
6	CSI_COMMON_STATUS2_6	IRQ_LN2_CMD_ERR; Lane 2 command error
5	CSI_COMMON_STATUS2_5	IRQ_LN2_FIFO_OFLOW; Lane 2 FIFO over flow
4	CSI_COMMON_STATUS2_4	IRQ_LN1_FIRST_SYNC_FLAG

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2 (cont.)**

Bits	Name	Description
3	CSI_COMMON_STATUS2_3	IRQ_LN1_SECOND_SYNC_FLAG
2	CSI_COMMON_STATUS2_2	IRQ_LN1_DEMAPPING_ERROR
1	CSI_COMMON_STATUS2_1	IRQ_LN1_ULPS_ENTRY
0	CSI_COMMON_STATUS2_0	IRQ_LN1_MISSING_POST

**0x000248BC MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS3

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3**

Bits	Name	Description
7	CSI_COMMON_STATUS3_7	IRQ_LN3_SYNC_BIT_ERROR
6	CSI_COMMON_STATUS3_6	IRQ_LN3_FIFO_OFLOW
5	CSI_COMMON_STATUS3_5	IRQ_LN2_RXSYNC
4	CSI_COMMON_STATUS3_4	IRQ_LN2_ULPM_EXIT; Lane 2 ULPM exit
3	CSI_COMMON_STATUS3_3	IRQ_LN2_ULPM_ENTRY; Lane 2 ULPM entry
2	CSI_COMMON_STATUS3_2	IRQ_LN2_ULPM_ERR; Lane 2 ULPM error
1	CSI_COMMON_STATUS3_1	IRQ_LN2_ERR_SOT_SYNC; Lane 2 SYNC error correction
0	CSI_COMMON_STATUS3_0	IRQ_LN2_ERR_SOT; Lane 2 SYNC error detection

**0x000248C0 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS4

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4**

Bits	Name	Description
7	CSI_COMMON_STATUS4_7	IRQ_LN3_DEMAPPING_ERROR
6	CSI_COMMON_STATUS4_6	IRQ_LN3_ULPS_ENTRY

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4 (cont.)**

Bits	Name	Description
5	CSI_COMMON_STATUS4_5	IRQ_LN3_MISSING_POST
4	CSI_COMMON_STATUS4_4	IRQ_LN3_HS_ENTRY_ERROR
3	CSI_COMMON_STATUS4_3	IRQ_LN3_HS_EXIT_ERROR
2	CSI_COMMON_STATUS4_2	IRQ_LN3_ESCAPE_ENTRY_ERROR
1	CSI_COMMON_STATUS4_1	IRQ_LN3_ESCAPE_COMMAND_ERROR
0	CSI_COMMON_STATUS4_0	IRQ_LN3_ESCAPE_EXIT_ERRO

**0x000248C4 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS5

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5**

Bits	Name	Description
7	CSI_COMMON_STATUS5_7	IRQ_LN4_ERR_SOT_SYNC; Lane 4 SYNC error correction
6	CSI_COMMON_STATUS5_6	IRQ_LN4_ERR_SOT; Lane 4 SYNC error detection
5	CSI_COMMON_STATUS5_5	IRQ_LN4_CTL_ERR; Lane 4 control error
4	CSI_COMMON_STATUS5_4	IRQ_LN4_CMD_ERR; Lane 4 command error
3	CSI_COMMON_STATUS5_3	IRQ_LN4_FIFO_OFLOW; Lane 4 FIFO over flow
2	CSI_COMMON_STATUS5_2	IRQ_LN4_ULPM_EXIT; Lane 4 ULPM exit
1	CSI_COMMON_STATUS5_1	IRQ_LN3_FIRST_SYNC_FLAG
0	CSI_COMMON_STATUS5_0	IRQ_LN3_SECOND_SYNC_FLAG

**0x000248C8 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS6

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6**

Bits	Name	Description
7	CSI_COMMON_STATUS6_7	IRQ_LN5_ESCAPE_ENTRY_ERROR
6	CSI_COMMON_STATUS6_6	IRQ_LN5_ESCAPE_COMMAND_ERROR
5	CSI_COMMON_STATUS6_5	IRQ_LN5_ESCAPE_EXIT_ERROR
4	CSI_COMMON_STATUS6_4	IRQ_LN5_SYNC_BIT_ERROR
3	CSI_COMMON_STATUS6_3	IRQ_LN5_FIFO_OFLOW
2	CSI_COMMON_STATUS6_2	IRQ_LN4_RXSYNC
1	CSI_COMMON_STATUS6_1	IRQ_LN4_ULPM_ENTRY; Lane 4 ULPM entry
0	CSI_COMMON_STATUS6_0	IRQ_LN4_ULPM_ERR; Lane 4 ULPM error

**0x000248CC MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS7

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7**

Bits	Name	Description
7	CSI_COMMON_STATUS7_7	IRQ_LN6_FIFO_OFLOW; Lane 6 FIFO over flow
6	CSI_COMMON_STATUS7_6	IRQ_LN5_FIRST_SYNC_FLAG
5	CSI_COMMON_STATUS7_5	IRQ_LN5_SECOND_SYNC_FLAG
4	CSI_COMMON_STATUS7_4	IRQ_LN5_DEMAPPING_ERROR
3	CSI_COMMON_STATUS7_3	IRQ_LN5_ULPS_ENTRY
2	CSI_COMMON_STATUS7_2	IRQ_LN5_MISSING_POST
1	CSI_COMMON_STATUS7_1	IRQ_LN5_HS_ENTRY_ERROR
0	CSI_COMMON_STATUS7_0	IRQ_LN5_HS_EXIT_ERROR

**0x000248D0 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS8

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8**

Bits	Name	Description
7	CSI_COMMON_STATUS8_7	IRQ_LN6_RXSYNC
6	CSI_COMMON_STATUS8_6	IRQ_LN6_ULPM_EXIT; Lane 6 ULPM exit
5	CSI_COMMON_STATUS8_5	IRQ_LN6_ULPM_ENTRY; Lane 6 ULPM entry
4	CSI_COMMON_STATUS8_4	IRQ_LN6_ULPM_ERR; Lane 6 ULPM error
3	CSI_COMMON_STATUS8_3	IRQ_LN6_ERR_SOT_SYNC; Lane 6 SYNC error correction
2	CSI_COMMON_STATUS8_2	IRQ_LN6_ERR_SOT; Lane 6 SYNC error detection
1	CSI_COMMON_STATUS8_1	IRQ_LN6_CTL_ERR; Lane 6 control error
0	CSI_COMMON_STATUS8_0	IRQ_LN6_CMD_ERR; Lane 6 command error

**0x000248D4 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS9

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9**

Bits	Name	Description
7	CSI_COMMON_STATUS9_7	IRQ_LNCK_CLK_STOP
6	CSI_COMMON_STATUS9_6	IRQ_LNCK_CLK_START
5	CSI_COMMON_STATUS9_5	IRQ_LN6_CLK_STOP
4	CSI_COMMON_STATUS9_4	IRQ_LN6_CLK_START
3	CSI_COMMON_STATUS9_3	IRQ_LN4_CLK_STOP
2	CSI_COMMON_STATUS9_2	IRQ_LN4_CLK_START
1	CSI_COMMON_STATUS9_1	IRQ_LN2_CLK_STOP
0	CSI_COMMON_STATUS9_0	IRQ_LN2_CLK_START

**0x000248D8 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS10

**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10**

Bits	Name	Description
7	CSI_COMMON_STATUS10_7	IRQ_LNCK_ULPM_EXIT; Lane CKULPM exit
6	CSI_COMMON_STATUS10_6	IRQ_LNCK_ULPM_ENTRY; Lane CK ULPM entry
5	CSI_COMMON_STATUS10_5	IRQ_LNCK_ULPM_ERR; Lane CK ULPM error
4	CSI_COMMON_STATUS10_4	IRQ_LNCK_ERR_SOT_SYNC; Lane CK SYNC error correction
3	CSI_COMMON_STATUS10_3	IRQ_LNCK_ERR_SOT; Lane CK SYNC error detection
2	CSI_COMMON_STATUS10_2	IRQ_LNCK_CTL_ERR; Lane CK control error
1	CSI_COMMON_STATUS10_1	IRQ_LNCK_CMD_ERR; Lane CK command error
0	CSI_COMMON_STATUS10_0	IRQ_LNCK_FIFO_OFLOW; Lane CK FIFO over flow

**0x0002582C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL11

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11**

Bits	Name	Description
7:0	CSI_COMMON_CTRL11	IRQ_MASK[7:0]

**0x00025830 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL12

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12**

Bits	Name	Description
7:0	CSI_COMMON_CTRL12	IRQ_MASK[15:8]

**0x00025834 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL13

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13**

Bits	Name	Description
7:0	CSI_COMMON_CTRL13	IRQ_MASK[23:16]

**0x00025838 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL14

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14**

Bits	Name	Description
7:0	CSI_COMMON_CTRL14	IRQ_MASK[31:24]

**0x0002583C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL15

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15**

Bits	Name	Description
7:0	CSI_COMMON_CTRL15	IRQ_MASK[39:32]

**0x00025840 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL16

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16**

Bits	Name	Description
7:0	CSI_COMMON_CTRL16	IRQ_MASK[47:40]

**0x00025844 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL17

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17**

Bits	Name	Description
7:0	CSI_COMMON_CTRL17	IRQ_MASK[55:48]

**0x00025848 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL18

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18**

Bits	Name	Description
7:0	CSI_COMMON_CTRL18	IRQ_MASK[63:56]

**0x0002584C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL19

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19**

Bits	Name	Description
7:0	CSI_COMMON_CTRL19	IRQ_MASK[71:64]

**0x00025850 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL20

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20**

Bits	Name	Description
7:0	CSI_COMMON_CTRL20	IRQ_MASK[79:72]

**0x00025854 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL21

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21**

Bits	Name	Description
7:0	CSI_COMMON_CTRL21	IRQ_MASK[87:80]

**0x00025858 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL22

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22**

Bits	Name	Description
7:0	CSI_COMMON_CTRL22	IRQ_CLEAR[7:0]

**0x0002585C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL23

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23**

Bits	Name	Description
7:0	CSI_COMMON_CTRL23	IRQ_CLEAR[15:8]

**0x00025860 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL24

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24**

Bits	Name	Description
7:0	CSI_COMMON_CTRL24	IRQ_CLEAR[23:16]

**0x00025864 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL25

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25**

Bits	Name	Description
7:0	CSI_COMMON_CTRL25	IRQ_CLEAR[31:24]

**0x00025868 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL26

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26**

Bits	Name	Description
7:0	CSI_COMMON_CTRL26	IRQ_CLEAR[39:32]

**0x0002586C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL27

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27**

Bits	Name	Description
7:0	CSI_COMMON_CTRL27	IRQ_CLEAR[47:40]

**0x00025870 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL28

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28**

Bits	Name	Description
7:0	CSI_COMMON_CTRL28	IRQ_CLEAR[55:48]

**0x00025874 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL29

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29**

Bits	Name	Description
7:0	CSI_COMMON_CTRL29	IRQ_CLEAR[63:56]

**0x00025878 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL30

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30**

Bits	Name	Description
7:0	CSI_COMMON_CTRL30	IRQ_CLEAR[71:64]

**0x0002587C MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL31

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31**

Bits	Name	Description
7:0	CSI_COMMON_CTRL31	IRQ_CLEAR[79:72]

**0x00025880 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL32

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32**

Bits	Name	Description
7:0	CSI_COMMON_CTRL32	IRQ_CLEAR[87:80]

**0x000258B0 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS0

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0**

Bits	Name	Description
7	CSI_COMMON_STATUS0_7	IRQ_LN0_ULPM_EXIT; Lane 0 ULPM exit
6	CSI_COMMON_STATUS0_6	IRQ_LN0_ULPM_ENTRY; Lane 0 ULPM entry
5	CSI_COMMON_STATUS0_5	IRQ_LN0_ULPM_ERR; Lane 0 ULPM error
4	CSI_COMMON_STATUS0_4	IRQ_LN0_ERR_SOT_SYNC; Lane 0 SYNC error correction
3	CSI_COMMON_STATUS0_3	IRQ_LN0_ERR_SOT; Lane 0 SYNC error detection
2	CSI_COMMON_STATUS0_2	IRQ_LN0_CTL_ERR; Lane 0 control error

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0 (cont.)**

Bits	Name	Description
1	CSI_COMMON_STATUS0_1	IRQ_LN0_CMD_ERR; Lane 0 command error
0	CSI_COMMON_STATUS0_0	IRQ_LN0_FIFO_OFLOW; Lane 0 FIFO over flow

**0x000258B4 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS1

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1**

Bits	Name	Description
7	CSI_COMMON_STATUS1_7	IRQ_LN1_HS_ENTRY_ERROR
6	CSI_COMMON_STATUS1_6	IRQ_LN1_HS_EXIT_ERROR
5	CSI_COMMON_STATUS1_5	IRQ_LN1_ESCAPE_ENTRY_ERROR
4	CSI_COMMON_STATUS1_4	IRQ_LN1_ESCAPE_COMMAND_ERROR
3	CSI_COMMON_STATUS1_3	IRQ_LN1_ESCAPE_EXIT_ERROR
2	CSI_COMMON_STATUS1_2	IRQ_LN1_SYNC_BIT_ERROR
1	CSI_COMMON_STATUS1_1	IRQ_LN1_FIFO_OFLOW
0	CSI_COMMON_STATUS1_0	IRQ_LN0_RXSYNC

**0x000258B8 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS2

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2**

Bits	Name	Description
7	CSI_COMMON_STATUS2_7	IRQ_LN2_CTL_ERR; Lane 2 control error
6	CSI_COMMON_STATUS2_6	IRQ_LN2_CMD_ERR; Lane 2 command error
5	CSI_COMMON_STATUS2_5	IRQ_LN2_FIFO_OFLOW; Lane 2 FIFO over flow
4	CSI_COMMON_STATUS2_4	IRQ_LN1_FIRST_SYNC_FLAG

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2 (cont.)**

Bits	Name	Description
3	CSI_COMMON_STATUS2_3	IRQ_LN1_SECOND_SYNC_FLAG
2	CSI_COMMON_STATUS2_2	IRQ_LN1_DEMAPPING_ERROR
1	CSI_COMMON_STATUS2_1	IRQ_LN1_ULPS_ENTRY
0	CSI_COMMON_STATUS2_0	IRQ_LN1_MISSING_POST

**0x000258BC MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS3

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3**

Bits	Name	Description
7	CSI_COMMON_STATUS3_7	IRQ_LN3_SYNC_BIT_ERROR
6	CSI_COMMON_STATUS3_6	IRQ_LN3_FIFO_OFLOW
5	CSI_COMMON_STATUS3_5	IRQ_LN2_RXSYNC
4	CSI_COMMON_STATUS3_4	IRQ_LN2_ULPM_EXIT; Lane 2 ULPM exit
3	CSI_COMMON_STATUS3_3	IRQ_LN2_ULPM_ENTRY; Lane 2 ULPM entry
2	CSI_COMMON_STATUS3_2	IRQ_LN2_ULPM_ERR; Lane 2 ULPM error
1	CSI_COMMON_STATUS3_1	IRQ_LN2_ERR_SOT_SYNC; Lane 2 SYNC error correction
0	CSI_COMMON_STATUS3_0	IRQ_LN2_ERR_SOT; Lane 2 SYNC error detection

**0x000258C0 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS4

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4**

Bits	Name	Description
7	CSI_COMMON_STATUS4_7	IRQ_LN3_DEMAPPING_ERROR
6	CSI_COMMON_STATUS4_6	IRQ_LN3_ULPS_ENTRY

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4 (cont.)**

Bits	Name	Description
5	CSI_COMMON_STATUS4_5	IRQ_LN3_MISSING_POST
4	CSI_COMMON_STATUS4_4	IRQ_LN3_HS_ENTRY_ERROR
3	CSI_COMMON_STATUS4_3	IRQ_LN3_HS_EXIT_ERROR
2	CSI_COMMON_STATUS4_2	IRQ_LN3_ESCAPE_ENTRY_ERROR
1	CSI_COMMON_STATUS4_1	IRQ_LN3_ESCAPE_COMMAND_ERROR
0	CSI_COMMON_STATUS4_0	IRQ_LN3_ESCAPE_EXIT_ERRO

**0x000258C4 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS5

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5**

Bits	Name	Description
7	CSI_COMMON_STATUS5_7	IRQ_LN4_ERR_SOT_SYNC; Lane 4 SYNC error correction
6	CSI_COMMON_STATUS5_6	IRQ_LN4_ERR_SOT; Lane 4 SYNC error detection
5	CSI_COMMON_STATUS5_5	IRQ_LN4_CTL_ERR; Lane 4 control error
4	CSI_COMMON_STATUS5_4	IRQ_LN4_CMD_ERR; Lane 4 command error
3	CSI_COMMON_STATUS5_3	IRQ_LN4_FIFO_OFLOW; Lane 4 FIFO over flow
2	CSI_COMMON_STATUS5_2	IRQ_LN4_ULPM_EXIT; Lane 4 ULPM exit
1	CSI_COMMON_STATUS5_1	IRQ_LN3_FIRST_SYNC_FLAG
0	CSI_COMMON_STATUS5_0	IRQ_LN3_SECOND_SYNC_FLAG

**0x000258C8 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS6

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6**

Bits	Name	Description
7	CSI_COMMON_STATUS6_7	IRQ_LN5_ESCAPE_ENTRY_ERROR
6	CSI_COMMON_STATUS6_6	IRQ_LN5_ESCAPE_COMMAND_ERROR
5	CSI_COMMON_STATUS6_5	IRQ_LN5_ESCAPE_EXIT_ERROR
4	CSI_COMMON_STATUS6_4	IRQ_LN5_SYNC_BIT_ERROR
3	CSI_COMMON_STATUS6_3	IRQ_LN5_FIFO_OFLOW
2	CSI_COMMON_STATUS6_2	IRQ_LN4_RXSYNC
1	CSI_COMMON_STATUS6_1	IRQ_LN4_ULPM_ENTRY; Lane 4 ULPM entry
0	CSI_COMMON_STATUS6_0	IRQ_LN4_ULPM_ERR; Lane 4 ULPM error

**0x000258CC MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS7

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7**

Bits	Name	Description
7	CSI_COMMON_STATUS7_7	IRQ_LN6_FIFO_OFLOW; Lane 6 FIFO over flow
6	CSI_COMMON_STATUS7_6	IRQ_LN5_FIRST_SYNC_FLAG
5	CSI_COMMON_STATUS7_5	IRQ_LN5_SECOND_SYNC_FLAG
4	CSI_COMMON_STATUS7_4	IRQ_LN5_DEMAPPING_ERROR
3	CSI_COMMON_STATUS7_3	IRQ_LN5_ULPS_ENTRY
2	CSI_COMMON_STATUS7_2	IRQ_LN5_MISSING_POST
1	CSI_COMMON_STATUS7_1	IRQ_LN5_HS_ENTRY_ERROR
0	CSI_COMMON_STATUS7_0	IRQ_LN5_HS_EXIT_ERROR

**0x000258D0 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS8

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8**

Bits	Name	Description
7	CSI_COMMON_STATUS8_7	IRQ_LN6_RXSYNC
6	CSI_COMMON_STATUS8_6	IRQ_LN6_ULPM_EXIT; Lane 6 ULPM exit
5	CSI_COMMON_STATUS8_5	IRQ_LN6_ULPM_ENTRY; Lane 6 ULPM entry
4	CSI_COMMON_STATUS8_4	IRQ_LN6_ULPM_ERR; Lane 6 ULPM error
3	CSI_COMMON_STATUS8_3	IRQ_LN6_ERR_SOT_SYNC; Lane 6 SYNC error correction
2	CSI_COMMON_STATUS8_2	IRQ_LN6_ERR_SOT; Lane 6 SYNC error detection
1	CSI_COMMON_STATUS8_1	IRQ_LN6_CTL_ERR; Lane 6 control error
0	CSI_COMMON_STATUS8_0	IRQ_LN6_CMD_ERR; Lane 6 command error

**0x000258D4 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS9

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9**

Bits	Name	Description
7	CSI_COMMON_STATUS9_7	IRQ_LNCK_CLK_STOP
6	CSI_COMMON_STATUS9_6	IRQ_LNCK_CLK_START
5	CSI_COMMON_STATUS9_5	IRQ_LN6_CLK_STOP
4	CSI_COMMON_STATUS9_4	IRQ_LN6_CLK_START
3	CSI_COMMON_STATUS9_3	IRQ_LN4_CLK_STOP
2	CSI_COMMON_STATUS9_2	IRQ_LN4_CLK_START
1	CSI_COMMON_STATUS9_1	IRQ_LN2_CLK_STOP
0	CSI_COMMON_STATUS9_0	IRQ_LN2_CLK_START

**0x000258D8 MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS10

**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10**

Bits	Name	Description
7	CSI_COMMON_STATUS10_7	IRQ_LNCK_ULPM_EXIT; Lane CKULPM exit
6	CSI_COMMON_STATUS10_6	IRQ_LNCK_ULPM_ENTRY; Lane CK ULPM entry
5	CSI_COMMON_STATUS10_5	IRQ_LNCK_ULPM_ERR; Lane CK ULPM error
4	CSI_COMMON_STATUS10_4	IRQ_LNCK_ERR_SOT_SYNC; Lane CK SYNC error correction
3	CSI_COMMON_STATUS10_3	IRQ_LNCK_ERR_SOT; Lane CK SYNC error detection
2	CSI_COMMON_STATUS10_2	IRQ_LNCK_CTL_ERR; Lane CK control error
1	CSI_COMMON_STATUS10_1	IRQ_LNCK_CMD_ERR; Lane CK command error
0	CSI_COMMON_STATUS10_0	IRQ_LNCK_FIFO_OFLOW; Lane CK FIFO over flow

**0x0002682C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL11

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11**

Bits	Name	Description
7:0	CSI_COMMON_CTRL11	IRQ_MASK[7:0]

**0x00026830 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL12

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12**

Bits	Name	Description
7:0	CSI_COMMON_CTRL12	IRQ_MASK[15:8]

**0x00026834 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL13

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13**

Bits	Name	Description
7:0	CSI_COMMON_CTRL13	IRQ_MASK[23:16]

**0x00026838 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL14

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14**

Bits	Name	Description
7:0	CSI_COMMON_CTRL14	IRQ_MASK[31:24]

**0x0002683C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL15

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15**

Bits	Name	Description
7:0	CSI_COMMON_CTRL15	IRQ_MASK[39:32]

**0x00026840 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL16

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16**

Bits	Name	Description
7:0	CSI_COMMON_CTRL16	IRQ_MASK[47:40]

**0x00026844 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL17

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17**

Bits	Name	Description
7:0	CSI_COMMON_CTRL17	IRQ_MASK[55:48]

**0x00026848 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL18

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18**

Bits	Name	Description
7:0	CSI_COMMON_CTRL18	IRQ_MASK[63:56]

**0x0002684C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL19

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19**

Bits	Name	Description
7:0	CSI_COMMON_CTRL19	IRQ_MASK[71:64]

**0x00026850 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL20

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20**

Bits	Name	Description
7:0	CSI_COMMON_CTRL20	IRQ_MASK[79:72]

**0x00026854 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL21

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21**

Bits	Name	Description
7:0	CSI_COMMON_CTRL21	IRQ_MASK[87:80]

**0x00026858 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL22

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22**

Bits	Name	Description
7:0	CSI_COMMON_CTRL22	IRQ_CLEAR[7:0]

**0x0002685C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL23

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23**

Bits	Name	Description
7:0	CSI_COMMON_CTRL23	IRQ_CLEAR[15:8]

**0x00026860 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL24

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24**

Bits	Name	Description
7:0	CSI_COMMON_CTRL24	IRQ_CLEAR[23:16]

**0x00026864 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL25

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25**

Bits	Name	Description
7:0	CSI_COMMON_CTRL25	IRQ_CLEAR[31:24]

**0x00026868 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL26

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26**

Bits	Name	Description
7:0	CSI_COMMON_CTRL26	IRQ_CLEAR[39:32]

**0x0002686C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL27

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27**

Bits	Name	Description
7:0	CSI_COMMON_CTRL27	IRQ_CLEAR[47:40]

**0x00026870 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL28

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28**

Bits	Name	Description
7:0	CSI_COMMON_CTRL28	IRQ_CLEAR[55:48]

**0x00026874 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL29

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29**

Bits	Name	Description
7:0	CSI_COMMON_CTRL29	IRQ_CLEAR[63:56]

**0x00026878 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL30

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30**

Bits	Name	Description
7:0	CSI_COMMON_CTRL30	IRQ_CLEAR[71:64]

**0x0002687C MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL31

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31**

Bits	Name	Description
7:0	CSI_COMMON_CTRL31	IRQ_CLEAR[79:72]

**0x00026880 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32****Type:** RW**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_CTRL32

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32**

Bits	Name	Description
7:0	CSI_COMMON_CTRL32	IRQ_CLEAR[87:80]

**0x000268B0 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS0

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0**

Bits	Name	Description
7	CSI_COMMON_STATUS0_7	IRQ_LN0_ULPM_EXIT; Lane 0 ULPM exit
6	CSI_COMMON_STATUS0_6	IRQ_LN0_ULPM_ENTRY; Lane 0 ULPM entry
5	CSI_COMMON_STATUS0_5	IRQ_LN0_ULPM_ERR; Lane 0 ULPM error
4	CSI_COMMON_STATUS0_4	IRQ_LN0_ERR_SOT_SYNC; Lane 0 SYNC error correction
3	CSI_COMMON_STATUS0_3	IRQ_LN0_ERR_SOT; Lane 0 SYNC error detection
2	CSI_COMMON_STATUS0_2	IRQ_LN0_CTL_ERR; Lane 0 control error

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS0 (cont.)**

Bits	Name	Description
1	CSI_COMMON_STATUS0_1	IRQ_LN0_CMD_ERR; Lane 0 command error
0	CSI_COMMON_STATUS0_0	IRQ_LN0_FIFO_OFLOW; Lane 0 FIFO over flow

**0x000268B4 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS1

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS1**

Bits	Name	Description
7	CSI_COMMON_STATUS1_7	IRQ_LN1_HS_ENTRY_ERROR
6	CSI_COMMON_STATUS1_6	IRQ_LN1_HS_EXIT_ERROR
5	CSI_COMMON_STATUS1_5	IRQ_LN1_ESCAPE_ENTRY_ERROR
4	CSI_COMMON_STATUS1_4	IRQ_LN1_ESCAPE_COMMAND_ERROR
3	CSI_COMMON_STATUS1_3	IRQ_LN1_ESCAPE_EXIT_ERROR
2	CSI_COMMON_STATUS1_2	IRQ_LN1_SYNC_BIT_ERROR
1	CSI_COMMON_STATUS1_1	IRQ_LN1_FIFO_OFLOW
0	CSI_COMMON_STATUS1_0	IRQ_LN0_RXSYNC

**0x000268B8 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS2

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2**

Bits	Name	Description
7	CSI_COMMON_STATUS2_7	IRQ_LN2_CTL_ERR; Lane 2 control error
6	CSI_COMMON_STATUS2_6	IRQ_LN2_CMD_ERR; Lane 2 command error
5	CSI_COMMON_STATUS2_5	IRQ_LN2_FIFO_OFLOW; Lane 2 FIFO over flow
4	CSI_COMMON_STATUS2_4	IRQ_LN1_FIRST_SYNC_FLAG

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS2 (cont.)**

Bits	Name	Description
3	CSI_COMMON_STATUS2_3	IRQ_LN1_SECOND_SYNC_FLAG
2	CSI_COMMON_STATUS2_2	IRQ_LN1_DEMAPPING_ERROR
1	CSI_COMMON_STATUS2_1	IRQ_LN1_ULPS_ENTRY
0	CSI_COMMON_STATUS2_0	IRQ_LN1_MISSING_POST

**0x000268BC MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS3

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS3**

Bits	Name	Description
7	CSI_COMMON_STATUS3_7	IRQ_LN3_SYNC_BIT_ERROR
6	CSI_COMMON_STATUS3_6	IRQ_LN3_FIFO_OFLOW
5	CSI_COMMON_STATUS3_5	IRQ_LN2_RXSYNC
4	CSI_COMMON_STATUS3_4	IRQ_LN2_ULPM_EXIT; Lane 2 ULPM exit
3	CSI_COMMON_STATUS3_3	IRQ_LN2_ULPM_ENTRY; Lane 2 ULPM entry
2	CSI_COMMON_STATUS3_2	IRQ_LN2_ULPM_ERR; Lane 2 ULPM error
1	CSI_COMMON_STATUS3_1	IRQ_LN2_ERR_SOT_SYNC; Lane 2 SYNC error correction
0	CSI_COMMON_STATUS3_0	IRQ_LN2_ERR_SOT; Lane 2 SYNC error detection

**0x000268C0 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS4

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4**

Bits	Name	Description
7	CSI_COMMON_STATUS4_7	IRQ_LN3_DEMAPPING_ERROR
6	CSI_COMMON_STATUS4_6	IRQ_LN3_ULPS_ENTRY

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS4 (cont.)**

Bits	Name	Description
5	CSI_COMMON_STATUS4_5	IRQ_LN3_MISSING_POST
4	CSI_COMMON_STATUS4_4	IRQ_LN3_HS_ENTRY_ERROR
3	CSI_COMMON_STATUS4_3	IRQ_LN3_HS_EXIT_ERROR
2	CSI_COMMON_STATUS4_2	IRQ_LN3_ESCAPE_ENTRY_ERROR
1	CSI_COMMON_STATUS4_1	IRQ_LN3_ESCAPE_COMMAND_ERROR
0	CSI_COMMON_STATUS4_0	IRQ_LN3_ESCAPE_EXIT_ERRO

**0x000268C4 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS5

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS5**

Bits	Name	Description
7	CSI_COMMON_STATUS5_7	IRQ_LN4_ERR_SOT_SYNC; Lane 4 SYNC error correction
6	CSI_COMMON_STATUS5_6	IRQ_LN4_ERR_SOT; Lane 4 SYNC error detection
5	CSI_COMMON_STATUS5_5	IRQ_LN4_CTL_ERR; Lane 4 control error
4	CSI_COMMON_STATUS5_4	IRQ_LN4_CMD_ERR; Lane 4 command error
3	CSI_COMMON_STATUS5_3	IRQ_LN4_FIFO_OFLOW; Lane 4 FIFO over flow
2	CSI_COMMON_STATUS5_2	IRQ_LN4_ULPM_EXIT; Lane 4 ULPM exit
1	CSI_COMMON_STATUS5_1	IRQ_LN3_FIRST_SYNC_FLAG
0	CSI_COMMON_STATUS5_0	IRQ_LN3_SECOND_SYNC_FLAG

**0x000268C8 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS6

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS6**

Bits	Name	Description
7	CSI_COMMON_STATUS6_7	IRQ_LN5_ESCAPE_ENTRY_ERROR
6	CSI_COMMON_STATUS6_6	IRQ_LN5_ESCAPE_COMMAND_ERROR
5	CSI_COMMON_STATUS6_5	IRQ_LN5_ESCAPE_EXIT_ERROR
4	CSI_COMMON_STATUS6_4	IRQ_LN5_SYNC_BIT_ERROR
3	CSI_COMMON_STATUS6_3	IRQ_LN5_FIFO_OFLOW
2	CSI_COMMON_STATUS6_2	IRQ_LN4_RXSYNC
1	CSI_COMMON_STATUS6_1	IRQ_LN4_ULPM_ENTRY; Lane 4 ULPM entry
0	CSI_COMMON_STATUS6_0	IRQ_LN4_ULPM_ERR; Lane 4 ULPM error

**0x000268CC MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS7

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS7**

Bits	Name	Description
7	CSI_COMMON_STATUS7_7	IRQ_LN6_FIFO_OFLOW; Lane 6 FIFO over flow
6	CSI_COMMON_STATUS7_6	IRQ_LN5_FIRST_SYNC_FLAG
5	CSI_COMMON_STATUS7_5	IRQ_LN5_SECOND_SYNC_FLAG
4	CSI_COMMON_STATUS7_4	IRQ_LN5_DEMAPPING_ERROR
3	CSI_COMMON_STATUS7_3	IRQ_LN5_ULPS_ENTRY
2	CSI_COMMON_STATUS7_2	IRQ_LN5_MISSING_POST
1	CSI_COMMON_STATUS7_1	IRQ_LN5_HS_ENTRY_ERROR
0	CSI_COMMON_STATUS7_0	IRQ_LN5_HS_EXIT_ERROR

**0x000268D0 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS8

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS8**

Bits	Name	Description
7	CSI_COMMON_STATUS8_7	IRQ_LN6_RXSYNC
6	CSI_COMMON_STATUS8_6	IRQ_LN6_ULPM_EXIT; Lane 6 ULPM exit
5	CSI_COMMON_STATUS8_5	IRQ_LN6_ULPM_ENTRY; Lane 6 ULPM entry
4	CSI_COMMON_STATUS8_4	IRQ_LN6_ULPM_ERR; Lane 6 ULPM error
3	CSI_COMMON_STATUS8_3	IRQ_LN6_ERR_SOT_SYNC; Lane 6 SYNC error correction
2	CSI_COMMON_STATUS8_2	IRQ_LN6_ERR_SOT; Lane 6 SYNC error detection
1	CSI_COMMON_STATUS8_1	IRQ_LN6_CTL_ERR; Lane 6 control error
0	CSI_COMMON_STATUS8_0	IRQ_LN6_CMD_ERR; Lane 6 command error

**0x000268D4 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS9

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS9**

Bits	Name	Description
7	CSI_COMMON_STATUS9_7	IRQ_LNCK_CLK_STOP
6	CSI_COMMON_STATUS9_6	IRQ_LNCK_CLK_START
5	CSI_COMMON_STATUS9_5	IRQ_LN6_CLK_STOP
4	CSI_COMMON_STATUS9_4	IRQ_LN6_CLK_START
3	CSI_COMMON_STATUS9_3	IRQ_LN4_CLK_STOP
2	CSI_COMMON_STATUS9_2	IRQ_LN4_CLK_START
1	CSI_COMMON_STATUS9_1	IRQ_LN2_CLK_STOP
0	CSI_COMMON_STATUS9_0	IRQ_LN2_CLK_START

**0x000268D8 MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10****Type:** R**Clock:** WCLK**Reset State:** 0x00000000

CSI\_COMMON\_STATUS10

**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STATUS10**

Bits	Name	Description
7	CSI_COMMON_STATUS10_7	IRQ_LNCK_ULPM_EXIT; Lane CKULPM exit
6	CSI_COMMON_STATUS10_6	IRQ_LNCK_ULPM_ENTRY; Lane CK ULPM entry
5	CSI_COMMON_STATUS10_5	IRQ_LNCK_ULPM_ERR; Lane CK ULPM error
4	CSI_COMMON_STATUS10_4	IRQ_LNCK_ERR_SOT_SYNC; Lane CK SYNC error correction
3	CSI_COMMON_STATUS10_3	IRQ_LNCK_ERR_SOT; Lane CK SYNC error detection
2	CSI_COMMON_STATUS10_2	IRQ_LNCK_CTL_ERR; Lane CK control error
1	CSI_COMMON_STATUS10_1	IRQ_LNCK_CMD_ERR; Lane CK command error
0	CSI_COMMON_STATUS10_0	IRQ_LNCK_FIFO_OFLOW; Lane CK FIFO over flow

**0x00230068 MMSS\_A\_CSID\_0\_CSID\_IRQ\_MASK****Type:** RW**Clock:** cc\_ahb\_clk**Reset State:** 0x00000800

This SW register contains the CSID core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the CSID core from notifying the system processors.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_MASK**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFL OW	0x0: DISABLE 0x1: ENABLE
30	ERROR_6PP_FIFO_OVERFL OW	0x0: DISABLE 0x1: ENABLE
29	ERROR_UNBOUNDED_FR AME	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_MASK (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
28	ERROR_STREAM_UNDERFL OW	0x0: DISABLE 0x1: ENABLE
27	ERROR_UNMAPPED_VC_D T	0x0: DISABLE 0x1: ENABLE
26	ERROR_MULTIMAPPED_V C_DT	0x0: DISABLE 0x1: ENABLE
25	ERROR_ECC	0x0: DISABLE 0x1: ENABLE
24	ERROR_CRC	0x0: DISABLE 0x1: ENABLE
23	ERROR_PHY_DL3_FIFO_O VERFLOW	0x0: DISABLE 0x1: ENABLE
22	ERROR_PHY_DL2_FIFO_O VERFLOW	0x0: DISABLE 0x1: ENABLE
21	ERROR_PHY_DL1_FIFO_O VERFLOW	0x0: DISABLE 0x1: ENABLE
20	ERROR_PHY_DL0_FIFO_O VERFLOW	0x0: DISABLE 0x1: ENABLE
19	ERROR_DL2_FIFO_OV ERFL OW	0x0: DISABLE 0x1: ENABLE
18	ERROR_DL1_FIFO_OV ERFL OW	0x0: DISABLE 0x1: ENABLE
17	ERROR_DL0_FIFO_OV ERFL OW	0x0: DISABLE 0x1: ENABLE
16	WARNING_ECC	0x0: DISABLE 0x1: ENABLE
15	ERROR_3P_PH_CRC	0x0: DISABLE 0x1: ENABLE
14	ERROR_3P_SOT_RECEP TION	0x0: DISABLE 0x1: ENABLE
13	ERROR_DL3_FIFO_OV ERFL OW	0x0: DISABLE 0x1: ENABLE
12	INFO_3P_PKT_HDR_CAPT URED	0x0: DISABLE 0x1: ENABLE
11	INFO_RST_DONE	0x0: DISABLE 0x1: ENABLE
10	INFO_TG_DONE	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_MASK (cont.)**

Bits	Name	Description
9	INFO_SHORT_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
8	INFO_LONG_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
7	INFO_PHY_DL3_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
6	INFO_PHY_DL2_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
5	INFO_PHY_DL1_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
4	INFO_PHY_DL0_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
3	INFO_PHY_DL3_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
2	INFO_PHY_DL2_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
1	INFO_PHY_DL1_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
0	INFO_PHY_DL0_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE

**0x00230064 MMSS\_A\_CSID\_0\_CSID\_IRQ\_CLEAR\_CMD****Type:** W**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register allows the user to clear CSID IRQs.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_CLEAR\_CMD**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFLOW	0x1: CLEAR
30	ERROR_6PP_FIFO_OVERFLOW	0x1: CLEAR
29	ERROR_UNBOUNDED_FRAME	0x1: CLEAR
28	ERROR_STREAM_UNDERFLOW	0x1: CLEAR

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
27	ERROR_UNMAPPED_VC_DT	0x1: CLEAR
26	ERROR_MULTIMAPPED_VC_DT	0x1: CLEAR
25	ERROR_ECC	0x1: CLEAR
24	ERROR_CRC	0x1: CLEAR
23	ERROR_PHY_DL3_FIFO_OVFLOW	0x1: CLEAR
22	ERROR_PHY_DL2_FIFO_OVFLOW	0x1: CLEAR
21	ERROR_PHY_DL1_FIFO_OVFLOW	0x1: CLEAR
20	ERROR_PHY_DL0_FIFO_OVFLOW	0x1: CLEAR
19	ERROR_DL2_FIFO_OVERFLOW	0x1: CLEAR
18	ERROR_DL1_FIFO_OVERFLOW	0x1: CLEAR
17	ERROR_DL0_FIFO_OVERFLOW	0x1: CLEAR
16	WARNING_ECC	0x1: CLEAR
15	ERROR_3P_PH_CRC	0x1: CLEAR
14	ERROR_3P_SOT_RECEPTION	0x1: CLEAR
13	ERROR_DL3_FIFO_OVERFLOW	0x1: CLEAR
12	INFO_3P_PKT_HDR_CAPTURED	0x1: CLEAR
11	INFO_RST_DONE	0x1: CLEAR
10	INFO_TG_DONE	0x1: CLEAR
9	INFO_SHORT_PKT_CAPTURED	0x1: CLEAR
8	INFO_LONG_PKT_CAPTURED	0x1: CLEAR
7	INFO_PHY_DL3_SOT_CAPTURED	0x1: CLEAR
6	INFO_PHY_DL2_SOT_CAPTURED	0x1: CLEAR

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

Bits	Name	Description
5	INFO_PHY_DL1_SOT_CAPTURED	0x1: CLEAR
4	INFO_PHY_DL0_SOT_CAPTURED	0x1: CLEAR
3	INFO_PHY_DL3_EOT_CAPTURED	0x1: CLEAR
2	INFO_PHY_DL2_EOT_CAPTURED	0x1: CLEAR
1	INFO_PHY_DL1_EOT_CAPTURED	0x1: CLEAR
0	INFO_PHY_DL0_EOT_CAPTURED	0x1: CLEAR

**0x0023006C MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the CSID core IRQ status.

Note that masking an IRQ will only prevent the CSID from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS**

Bits	Name	Description
29	ERROR_UNBOUNDED_FRAME	This IRQ fires when one of the following events occur: 1. FS (Frame Start) for VC n, <0 or more packets that are not FE for VC n>, FS for VC n. 2. FE (Frame End) for VC n, <0 or more packets that are not FE for VC n>, FE for VC n.
28	ERROR_STREAM_UNDERFLOW	This IRQ fires when the CSID core receives less bytes of payload in a long packet than specified in the long packet's header (word count). This is a fatal error which requires the user to reset the CSID core before continuing.
27	ERROR_UNMAPPED_VCDT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is not mapped to a CID. The header of the first error packet will be captured in CSID_CAPTURED_UNMAPPED_LONG_PKT_HDR.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
26	ERROR_MULTIMAPPED_V_C_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is mapped to more than one CID. The header of the erroneous long packet will be captured in CSID_CAPTURED_MMAPPED_LONG_PKT_HDR.
25	ERROR_ECC	This IRQ fires when one of the following events occur: 1. A corrupted short packet which cannot be recovered is received. 2. A corrupted long packet header which cannot be recovered is received.
24	ERROR_CRC	This IRQ fires when the CSID core receives a long packet and calculates a CRC that does not match the transmitted (expected) CRC. A CRC mismatch indicates the payload is corrupted.
23	ERROR_PHY_DL3_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
22	ERROR_PHY_DL2_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
21	ERROR_PHY_DL1_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
20	ERROR_PHY_DL0_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
19	ERROR_DL2_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
18	ERROR_DL1_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
17	ERROR_DL0_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
16	WARNING_ECC	This IRQ fires when the CSID core receives a short packet or a long packet header with 1 bit of corruption which is properly corrected by the core.
15	ERROR_3P_PH_CRC	This IRQ fires when all the 3-phase packet headers received are corrupted and CRC mismatches are produced.
14	ERROR_3P_SOT_RECEPTION	This IRQ fires when 1 or multiple trios receive less than 2 SOTs during a 3-phase packet transmission.
13	ERROR_DL3_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
12	INFO_3P_PKT_HDR_CAPTURED	This IRQ fires when the CSID core has captured the header of the first 3-phase packet matching the VC/DT specified in CSID_3PHASE_CTRL_0::PKT_CAPTURE_VC_DT. The captured packet header is present in CSID_CAPTURED_3P_PKT_HDR.

**MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
11	INFO_RST_DONE	This IRQ fires when the CSID core completes a software reset initiated by writing to CSID_RST_CMD.
9	INFO_SHORT_PKT_CAPTURED	This IRQ fires when the CSID core has captured the first short packet matching the VC/DT specified in CSID_CORE_CTRL::SHORT_PKT_CAPTURE_VC. The captured packet is present in CSID_CAPTURED_SHORT_PKT.
8	INFO_LONG_PKT_CAPTURED	This IRQ fires when the CSID core has captured the header of the first long packet matching the VC/DT specified in CSID_CORE_CTRL::LONG_PKT_CAPTURE_VC_DT. The captured packet header and footer are present in CSID_CAPTURED_LONG_PKT_HDR and CSID_CAPTURED_LONG_PKT_FTR respectively.
7	INFO_PHY_DL3_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
6	INFO_PHY_DL2_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 2.
5	INFO_PHY_DL1_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 1.
4	INFO_PHY_DL0_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 0.
3	INFO_PHY_DL3_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
2	INFO_PHY_DL2_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 2.
1	INFO_PHY_DL1_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 1.
0	INFO_PHY_DL0_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 0.

**0x00230070 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_UNMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230074 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_MULTIMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230078 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_SHORT\_PKT****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured short packet. The packet that is captured is controlled by CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_EN and CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_VC

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_SHORT\_PKT**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured short packet.
29:24	DT	The DT (Data Type) of the captured short packet.
23:8	FRAME_LINE_COUNT	For FS/FE packets, this is the frame number. For LS/LE packets, this is the line number.
7:0	ECC	The ECC of the captured short packet.

**0x0023007C MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230080 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_FTR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet footer. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT.

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_FTR**

Bits	Name	Description
31:16	CALCULATED_CRC	The CRC calculated by the CSID core based on the data received. This value should match the expected CRC.
15:0	EXPECTED_CRC	The CRC sent by the transmitter at the end of the packet.

**0x00230084 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_3P\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured 3-phase packet header. The packet that is captured is controlled by CSID\_3PHASE\_CTRL\_0::PKT\_CAPTURE\_EN.

The correct packet header will be captured in priority. In case all packet headers are corrupted, the second packet header on the last data trio is captured.

**MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_3P\_PKT\_HDR**

Bits	Name	Description
23:22	VC	The VC (Virtual Channel) of the captured 3-phase packet.
21:16	DT	The DT (Data Type) of the captured 3-phase packet.
15:0	WC	The WC (Word Count) of the captured 3-phase packet.

**0x00230468 MMSS\_A\_CSID\_1\_CSID\_IRQ\_MASK****Type:** RW**Clock:** cc\_ahb\_clk**Reset State:** 0x00000800

This SW register contains the CSID core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the CSID core from notifying the system processor(s).

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_MASK**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFL OW	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_MASK (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
29	ERROR_UNBOUNDED_FRAME	0x0: DISABLE 0x1: ENABLE
28	ERROR_STREAM_UNDERFLOW	0x0: DISABLE 0x1: ENABLE
27	ERROR_UNMAPPED_VCDT	0x0: DISABLE 0x1: ENABLE
26	ERROR_MULTIMAPPED_VC_DT	0x0: DISABLE 0x1: ENABLE
25	ERROR_ECC	0x0: DISABLE 0x1: ENABLE
24	ERROR_CRC	0x0: DISABLE 0x1: ENABLE
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
19	ERROR_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
18	ERROR_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
17	ERROR_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
16	WARNING_ECC	0x0: DISABLE 0x1: ENABLE
15	ERROR_3P_PH_CRC	0x0: DISABLE 0x1: ENABLE
14	ERROR_3P_SOT_RXCEPTION	0x0: DISABLE 0x1: ENABLE
13	ERROR_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
12	INFO_3P_PKT_HDR_CAPTURED	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_MASK (cont.)**

Bits	Name	Description
11	INFO_RST_DONE	0x0: DISABLE 0x1: ENABLE
10	INFO_TG_DONE	0x0: DISABLE 0x1: ENABLE
9	INFO_SHORT_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
8	INFO_LONG_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
7	INFO_PHY_DL3_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
6	INFO_PHY_DL2_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
5	INFO_PHY_DL1_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
4	INFO_PHY_DL0_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
3	INFO_PHY_DL3_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
2	INFO_PHY_DL2_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
1	INFO_PHY_DL1_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
0	INFO_PHY_DL0_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE

**0x00230464 MMSS\_A\_CSID\_1\_CSID\_IRQ\_CLEAR\_CMD****Type:** W**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register allows the user to clear CSID IRQs.

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_CLEAR\_CMD**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFLOW	0x1: CLEAR

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x1: CLEAR
29	ERROR_UNBOUNDED_FRAME	0x1: CLEAR
28	ERROR_STREAM_UNDERFLOW	0x1: CLEAR
27	ERROR_UNMAPPED_VC_DT	0x1: CLEAR
26	ERROR_MULTIMAPPED_VC_DT	0x1: CLEAR
25	ERROR_ECC	0x1: CLEAR
24	ERROR_CRC	0x1: CLEAR
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x1: CLEAR
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x1: CLEAR
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x1: CLEAR
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x1: CLEAR
19	ERROR_DL2_FIFO_OVERFLOW	0x1: CLEAR
18	ERROR_DL1_FIFO_OVERFLOW	0x1: CLEAR
17	ERROR_DL0_FIFO_OVERFLOW	0x1: CLEAR
16	WARNING_ECC	0x1: CLEAR
15	ERROR_3P_PH_CRC	0x1: CLEAR
14	ERROR_3P_SOT_RECEPTION	0x1: CLEAR
13	ERROR_DL3_FIFO_OVERFLOW	0x1: CLEAR
12	INFO_3P_PKT_HDR_CAPTURED	0x1: CLEAR
11	INFO_RST_DONE	0x1: CLEAR
10	INFO_TG_DONE	0x1: CLEAR
9	INFO_SHORT_PKT_CAPTURED	0x1: CLEAR

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

Bits	Name	Description
8	INFO_LONG_PKT_CAPTURED	0x1: CLEAR
7	INFO_PHY_DL3_SOT_Captured	0x1: CLEAR
6	INFO_PHY_DL2_SOT_Captured	0x1: CLEAR
5	INFO_PHY_DL1_SOT_Captured	0x1: CLEAR
4	INFO_PHY_DL0_SOT_Captured	0x1: CLEAR
3	INFO_PHY_DL3_EOT_Captured	0x1: CLEAR
2	INFO_PHY_DL2_EOT_Captured	0x1: CLEAR
1	INFO_PHY_DL1_EOT_Captured	0x1: CLEAR
0	INFO_PHY_DL0_EOT_Captured	0x1: CLEAR

**0x0023046C MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the CSID core IRQ status.

Note that masking an IRQ will only prevent the CSID from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS**

Bits	Name	Description
29	ERROR_UNBOUNDED_FRAME	This IRQ fires when one of the following events occur: 1. FS (Frame Start) for VC n, <0 or more packets that are not FE for VC n>, FS for VC n. 2. FE (Frame End) for VC n, <0 or more packets that are not FE for VC n>, FE for VC n.
28	ERROR_STREAM_UNDERFLOW	This IRQ fires when the CSID core receives less bytes of payload in a long packet than specified in the long packet's header (word count). This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
27	ERROR_UNMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is not mapped to a CID. The header of the first error packet will be captured in CSID_CAPTURED_UNMAPPED_LONG_PKT_HDR.
26	ERROR_MULTIMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is mapped to more than one CID. The header of the erroneous long packet will be captured in CSID_CAPTURED_MMAPPED_LONG_PKT_HDR.
25	ERROR_ECC	This IRQ fires when one of the following events occur: 1. A corrupted short packet which cannot be recovered is received. 2. A corrupted long packet header which cannot be recovered is received.
24	ERROR_CRC	This IRQ fires when the CSID core receives a long packet and calculates a CRC that does not match the transmitted (expected) CRC. A CRC mismatch indicates the payload is corrupted.
23	ERROR_PHY_DL3_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
22	ERROR_PHY_DL2_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
21	ERROR_PHY_DL1_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
20	ERROR_PHY_DL0_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
19	ERROR_DL2_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
18	ERROR_DL1_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
17	ERROR_DL0_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
16	WARNING_ECC	This IRQ fires when the CSID core receives a short packet or a long packet header with 1 bit of corruption which is properly corrected by the core.
15	ERROR_3P_PH_CRC	This IRQ fires when all the 3-phase packet headers received are corrupted and CRC mismatches are produced.
14	ERROR_3P_SOT_RECEPTION	This IRQ fires when 1 or multiple trios receive less than 2 SOTs during a 3-phase packet transmission.
13	ERROR_DL3_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
12	INFO_3P_PKT_HDR_CAPTURED	This IRQ fires when the CSID core has captured the header of the first 3-phase packet matching the VC/DT specified in CSID_3PHASE_CTRL_0::PKT_CAPTURE_VC_DT. The captured packet header is present in CSID_CAPTURED_3P_PKT_HDR.

**MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
11	INFO_RST_DONE	This IRQ fires when the CSID core completes a software reset initiated by writing to CSID_RST_CMD.
9	INFO_SHORT_PKT_CAPTURED	This IRQ fires when the CSID core has captured the first short packet matching the VC/DT specified in CSID_CORE_CTRL::SHORT_PKT_CAPTURE_VC. The captured packet is present in CSID_CAPTURED_SHORT_PKT.
8	INFO_LONG_PKT_CAPTURED	This IRQ fires when the CSID core has captured the header of the first long packet matching the VC/DT specified in CSID_CORE_CTRL::LONG_PKT_CAPTURE_VC_DT. The captured packet header and footer are present in CSID_CAPTURED_LONG_PKT_HDR and CSID_CAPTURED_LONG_PKT_FTR respectively.
7	INFO_PHY_DL3_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
6	INFO_PHY_DL2_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 2.
5	INFO_PHY_DL1_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 1.
4	INFO_PHY_DL0_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 0.
3	INFO_PHY_DL3_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
2	INFO_PHY_DL2_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 2.
1	INFO_PHY_DL1_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 1.
0	INFO_PHY_DL0_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 0.

**0x00230470 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_UNMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230474 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_MULTIMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230478 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_SHORT\_PKT****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured short packet. The packet that is captured is controlled by CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_EN and CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_VC

#### **MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_SHORT\_PKT**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured short packet.
29:24	DT	The DT (Data Type) of the captured short packet.
23:8	FRAME_LINE_COUNT	For FS/FE packets, this is the frame number. For LS/LE packets, this is the line number.
7:0	ECC	The ECC of the captured short packet.

#### **0x0023047C MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_HDR**

**Type:** R

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register contains the captured long packet header. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT

#### **MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

#### **0x00230480 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_FTR**

**Type:** R

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register contains the captured long packet footer. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT.

**MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_FTR**

Bits	Name	Description
31:16	CALCULATED_CRC	The CRC calculated by the CSID core based on the data received. This value should match the expected CRC.
15:0	EXPECTED_CRC	The CRC sent by the transmitter at the end of the packet.

**0x00230484 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_3P\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured 3-phase packet header. The packet that is captured is controlled by CSID\_3PHASE\_CTRL\_0::PKT\_CAPTURE\_EN.

The correct packet header will be captured in priority. In case all packet headers are corrupted, the second packet header on the last data trio is captured.

**MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_3P\_PKT\_HDR**

Bits	Name	Description
23:22	VC	The VC (Virtual Channel) of the captured 3-phase packet.
21:16	DT	The DT (Data Type) of the captured 3-phase packet.
15:0	WC	The WC (Word Count) of the captured 3-phase packet.

**0x00230868 MMSS\_A\_CSID\_2\_CSID\_IRQ\_MASK****Type:** RW**Clock:** cc\_ahb\_clk**Reset State:** 0x00000800

This SW register contains the CSID core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the CSID core from notifying the system processor(s).

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_MASK**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFL OW	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_MASK (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
29	ERROR_UNBOUNDED_FRAME	0x0: DISABLE 0x1: ENABLE
28	ERROR_STREAM_UNDERFLOW	0x0: DISABLE 0x1: ENABLE
27	ERROR_UNMAPPED_VCDT	0x0: DISABLE 0x1: ENABLE
26	ERROR_MULTIMAPPED_VC_DT	0x0: DISABLE 0x1: ENABLE
25	ERROR_ECC	0x0: DISABLE 0x1: ENABLE
24	ERROR_CRC	0x0: DISABLE 0x1: ENABLE
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
19	ERROR_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
18	ERROR_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
17	ERROR_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
16	WARNING_ECC	0x0: DISABLE 0x1: ENABLE
15	ERROR_3P_PH_CRC	0x0: DISABLE 0x1: ENABLE
14	ERROR_3P_SOT_RXCEPTION	0x0: DISABLE 0x1: ENABLE
13	ERROR_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
12	INFO_3P_PKT_HDR_CAPTURED	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_MASK (cont.)**

Bits	Name	Description
11	INFO_RST_DONE	0x0: DISABLE 0x1: ENABLE
10	INFO_TG_DONE	0x0: DISABLE 0x1: ENABLE
9	INFO_SHORT_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
8	INFO_LONG_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
7	INFO_PHY_DL3_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
6	INFO_PHY_DL2_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
5	INFO_PHY_DL1_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
4	INFO_PHY_DL0_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
3	INFO_PHY_DL3_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
2	INFO_PHY_DL2_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
1	INFO_PHY_DL1_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
0	INFO_PHY_DL0_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE

**0x00230864 MMSS\_A\_CSID\_2\_CSID\_IRQ\_CLEAR\_CMD****Type:** W**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register allows the user to clear CSID IRQs.

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_CLEAR\_CMD**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFLOW	0x1: CLEAR

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x1: CLEAR
29	ERROR_UNBOUNDED_FRAME	0x1: CLEAR
28	ERROR_STREAM_UNDERFLOW	0x1: CLEAR
27	ERROR_UNMAPPED_VC_DT	0x1: CLEAR
26	ERROR_MULTIMAPPED_VC_DT	0x1: CLEAR
25	ERROR_ECC	0x1: CLEAR
24	ERROR_CRC	0x1: CLEAR
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x1: CLEAR
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x1: CLEAR
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x1: CLEAR
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x1: CLEAR
19	ERROR_DL2_FIFO_OVERFLOW	0x1: CLEAR
18	ERROR_DL1_FIFO_OVERFLOW	0x1: CLEAR
17	ERROR_DL0_FIFO_OVERFLOW	0x1: CLEAR
16	WARNING_ECC	0x1: CLEAR
15	ERROR_3P_PH_CRC	0x1: CLEAR
14	ERROR_3P_SOT_RECEPTION	0x1: CLEAR
13	ERROR_DL3_FIFO_OVERFLOW	0x1: CLEAR
12	INFO_3P_PKT_HDR_CAPTURED	0x1: CLEAR
11	INFO_RST_DONE	0x1: CLEAR
10	INFO_TG_DONE	0x1: CLEAR
9	INFO_SHORT_PKT_CAPTURED	0x1: CLEAR

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

Bits	Name	Description
8	INFO_LONG_PKT_CAPTURED	0x1: CLEAR
7	INFO_PHY_DL3_SOT_Captured	0x1: CLEAR
6	INFO_PHY_DL2_SOT_Captured	0x1: CLEAR
5	INFO_PHY_DL1_SOT_Captured	0x1: CLEAR
4	INFO_PHY_DL0_SOT_Captured	0x1: CLEAR
3	INFO_PHY_DL3_EOT_Captured	0x1: CLEAR
2	INFO_PHY_DL2_EOT_Captured	0x1: CLEAR
1	INFO_PHY_DL1_EOT_Captured	0x1: CLEAR
0	INFO_PHY_DL0_EOT_Captured	0x1: CLEAR

**0x0023086C MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the CSID core IRQ status.

Note that masking an IRQ will only prevent the CSID from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS**

Bits	Name	Description
29	ERROR_UNBOUNDED_FRAME	This IRQ fires when one of the following events occur: 1. FS (Frame Start) for VC n, <0 or more packets that are not FE for VC n>, FS for VC n. 2. FE (Frame End) for VC n, <0 or more packets that are not FE for VC n>, FE for VC n.
28	ERROR_STREAM_UNDERFLOW	This IRQ fires when the CSID core receives less bytes of payload in a long packet than specified in the long packet's header (word count). This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
27	ERROR_UNMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is not mapped to a CID. The header of the first error packet will be captured in CSID_CAPTURED_UNMAPPED_LONG_PKT_HDR.
26	ERROR_MULTIMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is mapped to more than one CID. The header of the erroneous long packet will be captured in CSID_CAPTURED_MMAPPED_LONG_PKT_HDR.
25	ERROR_ECC	This IRQ fires when one of the following events occur: 1. A corrupted short packet which cannot be recovered is received. 2. A corrupted long packet header which cannot be recovered is received.
24	ERROR_CRC	This IRQ fires when the CSID core receives a long packet and calculates a CRC that does not match the transmitted (expected) CRC. A CRC mismatch indicates the payload is corrupted.
23	ERROR_PHY_DL3_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
22	ERROR_PHY_DL2_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
21	ERROR_PHY_DL1_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
20	ERROR_PHY_DL0_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
19	ERROR_DL2_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
18	ERROR_DL1_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
17	ERROR_DL0_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
16	WARNING_ECC	This IRQ fires when the CSID core receives a short packet or a long packet header with 1 bit of corruption which is properly corrected by the core.
15	ERROR_3P_PH_CRC	This IRQ fires when all the 3-phase packet headers received are corrupted and CRC mismatches are produced.
14	ERROR_3P_SOT_RECEPTION	This IRQ fires when 1 or multiple trios receive less than 2 SOTs during a 3-phase packet transmission.
13	ERROR_DL3_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
12	INFO_3P_PKT_HDR_CAPTURED	This IRQ fires when the CSID core has captured the header of the first 3-phase packet matching the VC/DT specified in CSID_3PHASE_CTRL_0::PKT_CAPTURE_VC_DT. The captured packet header is present in CSID_CAPTURED_3P_PKT_HDR.

**MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
11	INFO_RST_DONE	This IRQ fires when the CSID core completes a software reset initiated by writing to CSID_RST_CMD.
9	INFO_SHORT_PKT_CAPTURED	This IRQ fires when the CSID core has captured the first short packet matching the VC/DT specified in CSID_CORE_CTRL::SHORT_PKT_CAPTURE_VC. The captured packet is present in CSID_CAPTURED_SHORT_PKT.
8	INFO_LONG_PKT_CAPTURED	This IRQ fires when the CSID core has captured the header of the first long packet matching the VC/DT specified in CSID_CORE_CTRL::LONG_PKT_CAPTURE_VC_DT. The captured packet header and footer are present in CSID_CAPTURED_LONG_PKT_HDR and CSID_CAPTURED_LONG_PKT_FTR respectively.
7	INFO_PHY_DL3_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
6	INFO_PHY_DL2_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 2.
5	INFO_PHY_DL1_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 1.
4	INFO_PHY_DL0_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 0.
3	INFO_PHY_DL3_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
2	INFO_PHY_DL2_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 2.
1	INFO_PHY_DL1_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 1.
0	INFO_PHY_DL0_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 0.

**0x00230870 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_UNMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230874 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_MULTIMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230878 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_SHORT\_PKT****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured short packet. The packet that is captured is controlled by CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_EN and CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_VC

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_SHORT\_PKT**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured short packet.
29:24	DT	The DT (Data Type) of the captured short packet.
23:8	FRAME_LINE_COUNT	For FS/FE packets, this is the frame number. For LS/LE packets, this is the line number.
7:0	ECC	The ECC of the captured short packet.

**0x0023087C MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230880 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_FTR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet footer. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT.

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_FTR**

Bits	Name	Description
31:16	CALCULATED_CRC	The CRC calculated by the CSID core based on the data received. This value should match the expected CRC.
15:0	EXPECTED_CRC	The CRC sent by the transmitter at the end of the packet.

**0x00230884 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_3P\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured 3-phase packet header. The packet that is captured is controlled by CSID\_3PHASE\_CTRL\_0::PKT\_CAPTURE\_EN.

The correct packet header will be captured in priority. In case all packet headers are corrupted, the second packet header on the last data trio is captured.

**MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_3P\_PKT\_HDR**

Bits	Name	Description
23:22	VC	The VC (Virtual Channel) of the captured 3-phase packet.
21:16	DT	The DT (Data Type) of the captured 3-phase packet.
15:0	WC	The WC (Word Count) of the captured 3-phase packet.

**0x00230C68 MMSS\_A\_CSID\_3\_CSID\_IRQ\_MASK****Type:** RW**Clock:** cc\_ahb\_clk**Reset State:** 0x00000800

This SW register contains the CSID core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the CSID core from notifying the system processor(s).

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_MASK**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFL OW	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_MASK (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
29	ERROR_UNBOUNDED_FRAME	0x0: DISABLE 0x1: ENABLE
28	ERROR_STREAM_UNDERFLOW	0x0: DISABLE 0x1: ENABLE
27	ERROR_UNMAPPED_VCDT	0x0: DISABLE 0x1: ENABLE
26	ERROR_MULTIMAPPED_VC_DT	0x0: DISABLE 0x1: ENABLE
25	ERROR_ECC	0x0: DISABLE 0x1: ENABLE
24	ERROR_CRC	0x0: DISABLE 0x1: ENABLE
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
19	ERROR_DL2_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
18	ERROR_DL1_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
17	ERROR_DL0_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
16	WARNING_ECC	0x0: DISABLE 0x1: ENABLE
15	ERROR_3P_PH_CRC	0x0: DISABLE 0x1: ENABLE
14	ERROR_3P_SOT_RXCEPTION	0x0: DISABLE 0x1: ENABLE
13	ERROR_DL3_FIFO_OVERFLOW	0x0: DISABLE 0x1: ENABLE
12	INFO_3P_PKT_HDR_CAPTURED	0x0: DISABLE 0x1: ENABLE

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_MASK (cont.)**

Bits	Name	Description
11	INFO_RST_DONE	0x0: DISABLE 0x1: ENABLE
10	INFO_TG_DONE	0x0: DISABLE 0x1: ENABLE
9	INFO_SHORT_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
8	INFO_LONG_PKT_CAPTURED	0x0: DISABLE 0x1: ENABLE
7	INFO_PHY_DL3_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
6	INFO_PHY_DL2_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
5	INFO_PHY_DL1_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
4	INFO_PHY_DL0_EOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
3	INFO_PHY_DL3_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
2	INFO_PHY_DL2_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
1	INFO_PHY_DL1_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE
0	INFO_PHY_DL0_SOT_CAPTURED	0x0: DISABLE 0x1: ENABLE

**0x00230C64 MMSS\_A\_CSID\_3\_CSID\_IRQ\_CLEAR\_CMD****Type:** W**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register allows the user to clear CSID IRQs.

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_CLEAR\_CMD**

Bits	Name	Description
31	ERROR_TG_FIFO_OVERFLOW	0x1: CLEAR

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
30	ERROR_6PP_FIFO_OVERFLOW	0x1: CLEAR
29	ERROR_UNBOUNDED_FRAME	0x1: CLEAR
28	ERROR_STREAM_UNDERFLOW	0x1: CLEAR
27	ERROR_UNMAPPED_VC_DT	0x1: CLEAR
26	ERROR_MULTIMAPPED_VC_DT	0x1: CLEAR
25	ERROR_ECC	0x1: CLEAR
24	ERROR_CRC	0x1: CLEAR
23	ERROR_PHY_DL3_FIFO_OVERFLOW	0x1: CLEAR
22	ERROR_PHY_DL2_FIFO_OVERFLOW	0x1: CLEAR
21	ERROR_PHY_DL1_FIFO_OVERFLOW	0x1: CLEAR
20	ERROR_PHY_DL0_FIFO_OVERFLOW	0x1: CLEAR
19	ERROR_DL2_FIFO_OVERFLOW	0x1: CLEAR
18	ERROR_DL1_FIFO_OVERFLOW	0x1: CLEAR
17	ERROR_DL0_FIFO_OVERFLOW	0x1: CLEAR
16	WARNING_ECC	0x1: CLEAR
15	ERROR_3P_PH_CRC	0x1: CLEAR
14	ERROR_3P_SOT_RECEPTION	0x1: CLEAR
13	ERROR_DL3_FIFO_OVERFLOW	0x1: CLEAR
12	INFO_3P_PKT_HDR_CAPTURED	0x1: CLEAR
11	INFO_RST_DONE	0x1: CLEAR
10	INFO_TG_DONE	0x1: CLEAR
9	INFO_SHORT_PKT_CAPTURED	0x1: CLEAR

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_CLEAR\_CMD (cont.)**

Bits	Name	Description
8	INFO_LONG_PKT_CAPTURED	0x1: CLEAR
7	INFO_PHY_DL3_SOT_Captured	0x1: CLEAR
6	INFO_PHY_DL2_SOT_Captured	0x1: CLEAR
5	INFO_PHY_DL1_SOT_Captured	0x1: CLEAR
4	INFO_PHY_DL0_SOT_Captured	0x1: CLEAR
3	INFO_PHY_DL3_EOT_Captured	0x1: CLEAR
2	INFO_PHY_DL2_EOT_Captured	0x1: CLEAR
1	INFO_PHY_DL1_EOT_Captured	0x1: CLEAR
0	INFO_PHY_DL0_EOT_Captured	0x1: CLEAR

**0x00230C6C MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the CSID core IRQ status.

Note that masking an IRQ will only prevent the CSID from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS**

Bits	Name	Description
29	ERROR_UNBOUNDED_FRAME	This IRQ fires when one of the following events occur: 1. FS (Frame Start) for VC n, <0 or more packets that are not FE for VC n>, FS for VC n. 2. FE (Frame End) for VC n, <0 or more packets that are not FE for VC n>, FE for VC n.
28	ERROR_STREAM_UNDERFLOW	This IRQ fires when the CSID core receives less bytes of payload in a long packet than specified in the long packet's header (word count). This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
27	ERROR_UNMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is not mapped to a CID. The header of the first error packet will be captured in CSID_CAPTURED_UNMAPPED_LONG_PKT_HDR.
26	ERROR_MULTIMAPPED_VC_DT	This IRQ fires when the CSID core receives a long packet with a VC/DT combination that is mapped to more than one CID. The header of the erroneous long packet will be captured in CSID_CAPTURED_MMAPPED_LONG_PKT_HDR.
25	ERROR_ECC	This IRQ fires when one of the following events occur: 1. A corrupted short packet which cannot be recovered is received. 2. A corrupted long packet header which cannot be recovered is received.
24	ERROR_CRC	This IRQ fires when the CSID core receives a long packet and calculates a CRC that does not match the transmitted (expected) CRC. A CRC mismatch indicates the payload is corrupted.
23	ERROR_PHY_DL3_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
22	ERROR_PHY_DL2_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
21	ERROR_PHY_DL1_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.
20	ERROR_PHY_DL0_FIFO_OVERFLOW	This IRQ fires when the input FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDR clock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset the CSID core before continuing.

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS (cont.)**

Bits	Name	Description
19	ERROR_DL2_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL2 (Data Lane 2) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
18	ERROR_DL1_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL1 (Data Lane 1) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
17	ERROR_DL0_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL0 (Data Lane 0) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
16	WARNING_ECC	This IRQ fires when the CSID core receives a short packet or a long packet header with 1 bit of corruption which is properly corrected by the core.
15	ERROR_3P_PH_CRC	This IRQ fires when all the 3-phase packet headers received are corrupted and CRC mismatches are produced.
14	ERROR_3P_SOT_RECEPTION	This IRQ fires when 1 or multiple trios receive less than 2 SOTs during a 3-phase packet transmission.
13	ERROR_DL3_FIFO_OVERFLOW	This IRQ fires when the input async FIFO on the PHY DL3 (Data Lane 3) interface overflows. This is usually caused by one of the following: 1. cc_csi_clk is running too slow relative to the camera sensor DDRclock. 2. There is too much skew between data lanes coming from the PHY.  This is a fatal error which requires the user to reset to CSID core before continuing.
12	INFO_3P_PKT_HDR_CAPTURED	This IRQ fires when the CSID core has captured the header of the first 3-phase packet matching the VC/DT specified in CSID_3PHASE_CTRL_0::PKT_CAPTURE_VC_DT. The captured packet header is present in CSID_CAPTURED_3P_PKT_HDR.

**MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
11	INFO_RST_DONE	This IRQ fires when the CSID core completes a software reset initiated by writing to CSID_RST_CMD.
9	INFO_SHORT_PKT_CAPTURED	This IRQ fires when the CSID core has captured the first short packet matching the VC/DT specified in CSID_CORE_CTRL::SHORT_PKT_CAPTURE_VC. The captured packet is present in CSID_CAPTURED_SHORT_PKT.
8	INFO_LONG_PKT_CAPTURED	This IRQ fires when the CSID core has captured the header of the first long packet matching the VC/DT specified in CSID_CORE_CTRL::LONG_PKT_CAPTURE_VC_DT. The captured packet header and footer are present in CSID_CAPTURED_LONG_PKT_HDR and CSID_CAPTURED_LONG_PKT_FTR respectively.
7	INFO_PHY_DL3_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
6	INFO_PHY_DL2_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 2.
5	INFO_PHY_DL1_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 1.
4	INFO_PHY_DL0_SOT_Captured	This IRQ fires when the CSID core receives an SOT (Start of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing SOT on 3-phase PHY trio 0.
3	INFO_PHY_DL3_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 3 (Data Lane 3). When 3-phase is enabled, this IRQ is set to 0.
2	INFO_PHY_DL2_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 2 (Data Lane 2). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 2.
1	INFO_PHY_DL1_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 1 (Data Lane 1). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 1.
0	INFO_PHY_DL0_EOT_Captured	This IRQ fires when the CSID core receives an EOT (End of Transmission) on PHY DL 0 (Data Lane 0). When 3-phase is enabled, this IRQ is used for capturing EOT on 3-phase PHY trio 0.

**0x00230C70 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_UNMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_UNMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230C74 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header when the ERROR\_MULTIMAPPED\_VC\_DT IRQ fires.

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_MMAPPED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230C78 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_SHORT\_PKT****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured short packet. The packet that is captured is controlled by CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_EN and CSID\_CORE\_CTRL::SHORT\_PKT\_CAPTURE\_VC

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_SHORT\_PKT**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured short packet.
29:24	DT	The DT (Data Type) of the captured short packet.
23:8	FRAME_LINE_COUNT	For FS/FE packets, this is the frame number. For LS/LE packets, this is the line number.
7:0	ECC	The ECC of the captured short packet.

**0x00230C7C MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet header. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_HDR**

Bits	Name	Description
31:30	VC	The VC (Virtual Channel) of the captured long packet.
29:24	DT	The DT (Data Type) of the captured long packet.
23:8	WC	The WC (Word Count) of the captured long packet.
7:0	ECC	The ECC of the captured short packet.

**0x00230C80 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_FTR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured long packet footer. The packet that is captured is controlled by CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_EN, CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_VC and CSID\_CORE\_CTRL::LONG\_PKT\_CAPTURE\_DT.

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_FTR**

Bits	Name	Description
31:16	CALCULATED_CRC	The CRC calculated by the CSID core based on the data received. This value should match the expected CRC.
15:0	EXPECTED_CRC	The CRC sent by the transmitter at the end of the packet.

**0x00230C84 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_3P\_PKT\_HDR****Type:** R**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This SW register contains the captured 3-phase packet header. The packet that is captured is controlled by CSID\_3PHASE\_CTRL\_0::PKT\_CAPTURE\_EN.

The correct packet header will be captured in priority. In case all packet headers are corrupted, the second packet header on the last data trio is captured.

**MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_3P\_PKT\_HDR**

Bits	Name	Description
23:22	VC	The VC (Virtual Channel) of the captured 3-phase packet.
21:16	DT	The DT (Data Type) of the captured 3-phase packet.
15:0	WC	The WC (Word Count) of the captured 3-phase packet.

**R0x00231208 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_0, m=[0..1]  
+0x200\*m**
**Type:** RW**Clock:** cc\_ahb\_clk**Reset State:** 0x08000000

This SW register contains the ISPIF core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the ISPIF core from notifying the system processor(s).

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_0**

Bits	Name	Description
27:0	MASK	<p>This mask is used to enable/disable an interrupt source for the ISPIF interrupt to the ARM.</p> <p>When a bit is set (1), the interrupt source is enabled.</p> <p>When clear (0), the interrupt source is disabled</p> <pre>ispif_irq_src(0) &lt;= pix_intf_0_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_0_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_0_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_0_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_0_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_0_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_0_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_0_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_0_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_0_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_0_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_0_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_0_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_0_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_0_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_0_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_0_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_0_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_0_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_0_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_0_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_0_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_0_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_0_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_0_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_0_overflow_irq; ispif_irq_src(26) &lt;= reserved; ispif_irq_src(27) &lt;= rst_done_irq; ispif_irq_src(31:28) &lt;= reserved;</pre>

**0x0023120C+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_1, m=[0..1]  
0x200\*m**

**Type:** RW

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register contains the ISPIF core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the ISPIF core from notifying the system processor(s).

### MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_1

Bits	Name	Description
25:0	MASK_1	<p>This mask is used to enable/disable an interrupt source for the ISPIF interrupt to the ARM.</p> <p>When a bit is set (1), the interrupt source is enabled.</p> <p>When clear (0), the interrupt source is disabled</p> <pre>ispif_irq_src(0) &lt;= pix_intf_1_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_1_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_1_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_1_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_1_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_1_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_1_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_1_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_1_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_1_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_1_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_1_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_1_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_1_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_1_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_1_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_1_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_1_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_1_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_1_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_1_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_1_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_1_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_1_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_1_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_1_overflow_irq; ispif_irq_src(31:26) &lt;= reserved;</pre>

### 0x00231210+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_2, m=[0..1]

0x200\*m

**Type:** RW

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register contains the ISPIF core IRQ mask. The IRQ mask is used to configure which events the system processor(s) will handle (unmasked IRQ; setting the IRQ mask bit to 0x1) and ignore (masked IRQ; setting the IRQ mask bit to 0x0).

Note that a masked IRQ will still assert in this SW register if the corresponding event occurs. The mask only prevents the ISPIF core from notifying the system processor(s).

### MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_2

Bits	Name	Description
20:0	MASK_2	<p>This mask is used to enable/disable an interrupt source for the ISPIF interrupt to the ARM.</p> <p>When a bit is set (1), the interrupt source is enabled.</p> <p>When clear (0), the interrupt source is disabled</p> <pre>ispif_irq_src(0) &lt;= raw_intf_2_vc_0_sof_irq; ispif_irq_src(1) &lt;= raw_intf_2_vc_0_eof_irq; ispif_irq_src(2) &lt;= raw_intf_2_vc_0_eol_irq; ispif_irq_src(3) &lt;= raw_intf_2_vc_1_sof_irq; ispif_irq_src(4) &lt;= raw_intf_2_vc_1_eof_irq; ispif_irq_src(5) &lt;= raw_intf_2_vc_1_eol_irq; ispif_irq_src(6) &lt;= raw_intf_2_vc_2_sof_irq; ispif_irq_src(7) &lt;= raw_intf_2_vc_2_eof_irq; ispif_irq_src(8) &lt;= raw_intf_2_vc_2_eol_irq; ispif_irq_src(9) &lt;= raw_intf_2_vc_3_sof_irq; ispif_irq_src(10) &lt;= raw_intf_2_vc_3_eof_irq; ispif_irq_src(11) &lt;= raw_intf_2_vc_3_eol_irq; ispif_irq_src(12) &lt;= raw_intf_2_overflow_irq; ispif_irq_src(13) &lt;= 3d_sof_irq; ispif_irq_src(14) &lt;= 3d_eof_irq; ispif_irq_src(15) &lt;= 3d_eol_irq; ispif_irq_src(16) &lt;= 3d_l_r_sof_mismatch_err_irq; ispif_irq_src(17) &lt;= 3d_l_r_eof_mismatch_err_irq; ispif_irq_src(18) &lt;= 3d_l_r_sol_mismatch_err_irq; ispif_irq_src(19) &lt;= 3d_l_r_eol_mismatch_err_irq; ispif_irq_src(20) &lt;= 3d_deskew_size_latched; ispif_irq_src(31:21) &lt;= reserved;</pre>

**0x00231230+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_0, m=[0..1]**  
**0x200\*m**

**Type:** RW  
**Clock:** cc\_ahb\_clk  
**Reset State:** 0x00000000

This SW register allows the SW to select the interrupt bits that needs to be cleared in IRQ\_STATUS\_[n] register. The actual clear happens after issuing the global clear command from the IRQ\_GLOBAL\_CLEAR\_CMD register

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_0**

Bits	Name	Description
27:0	CLEAR	<p>This register must be programmed first before it is used to clear an interrupt source.</p> <ul style="list-style-type: none"> <li>- Setting (1) a bit clears the interrupt status in IRQ_STATUS after the global clear command is applied</li> <li>- Setting (0) a bit retains the corresponding status bit unchanged.</li> </ul> <pre>ispif_irq_src(0) &lt;= pix_intf_0_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_0_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_0_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_0_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_0_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_0_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_0_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_0_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_0_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_0_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_0_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_0_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_0_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_0_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_0_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_0_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_0_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_0_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_0_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_0_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_0_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_0_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_0_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_0_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_0_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_0_overflow_irq; ispif_irq_src(26) &lt;= reserved; ispif_irq_src(27) &lt;= rst_done_irq; ispif_irq_src(31:28) &lt;= reserved;</pre>

**0x00231234+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_1, m=[0..1]**  
**0x200\*m**

**Type:** RW

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register allows the SW to select the interrupt bits that needs to be cleared in IRQ\_STATUS\_1 register. The actual clear happens after issuing the global clear command from the IRQ\_GLOBAL\_CLEAR\_CMD register

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_1**

Bits	Name	Description
25:0	CLEAR_1	<p>This register must be programmed first before it is used to clear an interrupt source.</p> <ul style="list-style-type: none"> <li>- Setting (1) a bit clears the interrupt status in IRQ_STATUS_1 after the global clear command is applied</li> <li>- Setting (0) a bit retains the corresponding status bit unchanged.</li> </ul> <pre>ispif_irq_src(0) &lt;= pix_intf_1_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_1_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_1_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_1_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_1_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_1_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_1_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_1_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_1_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_1_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_1_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_1_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_1_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_1_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_1_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_1_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_1_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_1_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_1_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_1_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_1_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_1_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_1_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_1_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_1_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_1_overflow_irq; ispif_irq_src(31:26) &lt;= reserved;</pre>

**0x00231238+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_2, m=[0..1]**  
**0x200\*m**

**Type:** RW  
**Clock:** cc\_ahb\_clk  
**Reset State:** 0x00000000

This SW register allows the SW to select the interrupt bits that needs to be cleared in IRQ\_STATUS\_2 register. The actual clear happens after issuing the global clear command from the IRQ\_GLOBAL\_CLEAR\_CMD register

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_2**

Bits	Name	Description
20:0	CLEAR_2	<p>This register must be programmed first before it is used to clear an interrupt source.</p> <ul style="list-style-type: none"> <li>- Setting (1) a bit clears the interrupt status in IRQ_STATUS_2 after the global clear command is applied</li> <li>- Setting (0) a bit retains the corresponding status bit unchanged.</li> </ul> <pre>ispif_irq_src(0) &lt;= raw_intf_2_vc_0_sof_irq; ispif_irq_src(1) &lt;= raw_intf_2_vc_0_eof_irq; ispif_irq_src(2) &lt;= raw_intf_2_vc_0_eol_irq; ispif_irq_src(3) &lt;= raw_intf_2_vc_1_sof_irq; ispif_irq_src(4) &lt;= raw_intf_2_vc_1_eof_irq; ispif_irq_src(5) &lt;= raw_intf_2_vc_1_eol_irq; ispif_irq_src(6) &lt;= raw_intf_2_vc_2_sof_irq; ispif_irq_src(7) &lt;= raw_intf_2_vc_2_eof_irq; ispif_irq_src(8) &lt;= raw_intf_2_vc_2_eol_irq; ispif_irq_src(9) &lt;= raw_intf_2_vc_3_sof_irq; ispif_irq_src(10) &lt;= raw_intf_2_vc_3_eof_irq; ispif_irq_src(11) &lt;= raw_intf_2_vc_3_eol_irq; ispif_irq_src(12) &lt;= raw_intf_2_overflow_irq; ispif_irq_src(13) &lt;= 3d_sof_irq; ispif_irq_src(14) &lt;= 3d_eof_irq; ispif_irq_src(15) &lt;= 3d_eol_irq; ispif_irq_src(16) &lt;= 3d_l_r_sof_mismatch_err_irq; ispif_irq_src(17) &lt;= 3d_l_r_eof_mismatch_err_irq; ispif_irq_src(18) &lt;= 3d_l_r_sol_mismatch_err_irq; ispif_irq_src(19) &lt;= 3d_l_r_eol_mismatch_err_irq; ispif_irq_src(20) &lt;= 3d_deskew_size_latched; ispif_irq_src(31:21) &lt;= reserved;</pre>

**0x0023101C MMSS\_A\_ISPIF\_IRQ\_GLOBAL\_CLEAR\_CMD****Type:** W**Clock:** cc\_ahb\_clk**Reset State:** 0x00000000

This register is used to clear all the irq bits specified in IRQ\_CLEAR, IRQ\_CLEAR\_1 and IRQ\_CLEAR\_2 registers

**MMSS\_A\_ISPIF\_IRQ\_GLOBAL\_CLEAR\_CMD**

Bits	Name	Description
0	GLOBAL_CLEAR	<ul style="list-style-type: none"> <li>- Setting (1) the bit clears the interrupt status in VFE_[m]_IRQ_STATUS_[n] simultaneously.</li> <li>- Setting (0) the bit has no effect.</li> </ul>

**0x0023121C+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_0, m=[0..1]  
0x200\*m**

**Type:** R

**Clock:** cc\_ahb\_clk

**Reset State:** 0x00000000

This SW register contains the ISPIF core IRQ status.

Note that masking an IRQ will only prevent the ISPIF from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_0**

Bits	Name	Description
27:0	STATUS	<p>This register shows the raw status of the interrupts that were triggered.</p> <pre> ispif_irq_src(0) &lt;= pix_intf_0_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_0_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_0_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_0_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_0_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_0_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_0_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_0_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_0_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_0_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_0_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_0_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_0_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_0_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_0_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_0_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_0_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_0_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_0_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_0_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_0_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_0_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_0_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_0_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_0_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_0_overflow_irq; ispif_irq_src(26) &lt;= reserved; ispif_irq_src(27) &lt;= rst_done_irq; ispif_irq_src(31:28) &lt;= reserved;</pre>

**0x00231220+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_1, m=[0..1]**  
**0x200\*m**

**Type:** R  
**Clock:** cc\_ahb\_clk  
**Reset State:** 0x00000000

This SW register contains the ISPIF core IRQ status.

Note that masking an IRQ will only prevent the ISPIF from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

**MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_1**

Bits	Name	Description
25:0	STATUS_1	<p>This register shows the raw status of the interrupts that were triggered.</p> <pre> ispif_irq_src(0) &lt;= pix_intf_1_vc_0_sof_irq; ispif_irq_src(1) &lt;= pix_intf_1_vc_0_eof_irq; ispif_irq_src(2) &lt;= pix_intf_1_vc_0_eol_irq; ispif_irq_src(3) &lt;= pix_intf_1_vc_1_sof_irq; ispif_irq_src(4) &lt;= pix_intf_1_vc_1_eof_irq; ispif_irq_src(5) &lt;= pix_intf_1_vc_1_eol_irq; ispif_irq_src(6) &lt;= pix_intf_1_vc_2_sof_irq; ispif_irq_src(7) &lt;= pix_intf_1_vc_2_eof_irq; ispif_irq_src(8) &lt;= pix_intf_1_vc_2_eol_irq; ispif_irq_src(9) &lt;= pix_intf_1_vc_3_sof_irq; ispif_irq_src(10) &lt;= pix_intf_1_vc_3_eof_irq; ispif_irq_src(11) &lt;= pix_intf_1_vc_3_eol_irq; ispif_irq_src(12) &lt;= pix_intf_1_overflow_irq; ispif_irq_src(13) &lt;= raw_intf_1_vc_0_sof_irq; ispif_irq_src(14) &lt;= raw_intf_1_vc_0_eof_irq; ispif_irq_src(15) &lt;= raw_intf_1_vc_0_eol_irq; ispif_irq_src(16) &lt;= raw_intf_1_vc_1_sof_irq; ispif_irq_src(17) &lt;= raw_intf_1_vc_1_eof_irq; ispif_irq_src(18) &lt;= raw_intf_1_vc_1_eol_irq; ispif_irq_src(19) &lt;= raw_intf_1_vc_2_sof_irq; ispif_irq_src(20) &lt;= raw_intf_1_vc_2_eof_irq; ispif_irq_src(21) &lt;= raw_intf_1_vc_2_eol_irq; ispif_irq_src(22) &lt;= raw_intf_1_vc_3_sof_irq; ispif_irq_src(23) &lt;= raw_intf_1_vc_3_eof_irq; ispif_irq_src(24) &lt;= raw_intf_1_vc_3_eol_irq; ispif_irq_src(25) &lt;= raw_intf_1_overflow_irq; ispif_irq_src(31:26) &lt;= reserved;</pre>

**0x00231224+ MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_2, m=[0..1]**  
**0x200\*m**

Type: R

Clock: cc\_ahb\_clk

Reset State: 0x00000000

This SW register contains the ISPIF core IRQ status.

Note that masking an IRQ will only prevent the ISPIF from notifying the processor of the corresponding event. The corresponding IRQ bit in this register will still be set to 0x1.

### MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_2

Bits	Name	Description
20:0	STATUS_2	<p>This register shows the raw status of the interrupts that were triggered.</p> <pre> ispif_irq_src(0) &lt;= raw_intf_2_vc_0_sof_irq; ispif_irq_src(1) &lt;= raw_intf_2_vc_0_eof_irq; ispif_irq_src(2) &lt;= raw_intf_2_vc_0_eol_irq; ispif_irq_src(3) &lt;= raw_intf_2_vc_1_sof_irq; ispif_irq_src(4) &lt;= raw_intf_2_vc_1_eof_irq; ispif_irq_src(5) &lt;= raw_intf_2_vc_1_eol_irq; ispif_irq_src(6) &lt;= raw_intf_2_vc_2_sof_irq; ispif_irq_src(7) &lt;= raw_intf_2_vc_2_eof_irq; ispif_irq_src(8) &lt;= raw_intf_2_vc_2_eol_irq; ispif_irq_src(9) &lt;= raw_intf_2_vc_3_sof_irq; ispif_irq_src(10) &lt;= raw_intf_2_vc_3_eof_irq; ispif_irq_src(11) &lt;= raw_intf_2_vc_3_eol_irq; ispif_irq_src(12) &lt;= raw_intf_2_overflow_irq; ispif_irq_src(13) &lt;= 3d_sof_irq; ispif_irq_src(14) &lt;= 3d_eof_irq; ispif_irq_src(15) &lt;= 3d_eol_irq; ispif_irq_src(16) &lt;= 3d_l_r_sof_mismatch_err_irq; ispif_irq_src(17) &lt;= 3d_l_r_eof_mismatch_err_irq; ispif_irq_src(18) &lt;= 3d_l_r_sol_mismatch_err_irq; ispif_irq_src(19) &lt;= 3d_l_r_eol_mismatch_err_irq; ispif_irq_src(20) &lt;= 3d_deskew_size_latched; ispif_irq_src(31:21) &lt;= reserved; </pre>

**0x0021005C MMSS\_A\_VFE\_0\_IRQ\_MASK\_0****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x80000000**MMSS\_A\_VFE\_0\_IRQ\_MASK\_0**

Bits	Name	Description
31:0	MASK	<p>This mask is used to enable/disable an interrupt source for the VFE interrupt to the ARM.</p> <ul style="list-style-type: none"> <li>* When a bit is set (1), the interrupt source is enabled.</li> <li>* When clear (0), the interrupt source is disabled.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00210060 MMSS\_A\_VFE\_0\_IRQ\_MASK\_1****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_0\_IRQ\_MASK\_1**

Bits	Name	Description
31:0	MASK	<p>This mask is used to enable/disable an interrupt source for the VFE interrupt to the ARM.</p> <ul style="list-style-type: none"> <li>* When a bit is set (1), the interrupt source is enabled.</li> <li>* When clear (0), the interrupt source is disabled.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_error_irq; vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer BUS_OPERATION_STATUS register for the precise OVERWRITE_IRQ. vfe_irq_src(2) &lt;= spare_0_irq; vfe_irq_src(3) &lt;= spare_1_irq; vfe_irq_src(4) &lt;= bus_error_irq; vfe_irq_src(5) &lt;= spare_2_irq; vfe_irq_src(6) &lt;= spare_3_irq; vfe_irq_src(7) &lt;= violation_irq; vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq; vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq; vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq; vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq; vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq; vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq; vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq; vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq; vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq; vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq; vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq; vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq; vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq; vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq; vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq; vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq; vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq; vfe_irq_src(25) &lt;= dsp_error_irq; vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq; vfe_irq_src(27) &lt;= spare_4_irq; vfe_irq_src(28) &lt;= diag_irq; vfe_irq_src(29) &lt;= rdi0_sof_irq; vfe_irq_src(30) &lt;= rdi1_sof_irq;</pre>
		vfe_irq_src(31) <= rdi2_sof_irq;

**0x00210064 MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_0****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_0**

Bits	Name	Description
31:0	CLEAR	<p>This register must be programmed first before it used to clear an interrupt source.</p> <p>* Setting (1) a bit clears the interrupt status in IRQ_STATUS_0 after the global clear command is applied.</p> <p>* Setting (0) a bit retains the corresponding status bit unchanged.</p> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00210068 MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_1****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_1**

Bits	Name	Description
31:0	CLEAR	<p>This register must be programmed first before it used to clear an interrupt source.</p> <ul style="list-style-type: none"> <li>* Setting (1) a bit clears the interrupt status in IRQ_STATUS_1 after the global clear command is applied.</li> <li>* Setting (0) a bit retains the corresponding status bit unchanged.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_error_irq; vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer BUS_OPERATION_STATUS register for the precise OVERWRITE_IRQ. vfe_irq_src(2) &lt;= spare_0_irq; vfe_irq_src(3) &lt;= spare_1_irq; vfe_irq_src(4) &lt;= bus_error_irq; vfe_irq_src(5) &lt;= spare_2_irq; vfe_irq_src(6) &lt;= spare_3_irq; vfe_irq_src(7) &lt;= violation_irq; vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq; vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq; vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq; vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq; vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq; vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq; vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq; vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq; vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq; vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq; vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq; vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq; vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq; vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq; vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq; vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq; vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq; vfe_irq_src(25) &lt;= dsp_error_irq; vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq; vfe_irq_src(27) &lt;= spare_4_irq; vfe_irq_src(28) &lt;= diag_irq; vfe_irq_src(29) &lt;= rdi0_sof_irq;</pre>
		<pre>vfe_irq_src(30) &lt;= rdi1_sof_irq; vfe_irq_src(31) &lt;= rdi2_sof_irq;</pre>

**0x0021006C MMSS\_A\_VFE\_0\_IRQ\_STATUS\_0****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_0\_IRQ\_STATUS\_0**

Bits	Name	Description
31:0	STATUS	<p>This register shows the raw status of the interrupts that were triggered.</p> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00210070 MMSS\_A\_VFE\_0\_IRQ\_STATUS\_1****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_0\_IRQ\_STATUS\_1**

Bits	Name	Description
31:0	STATUS	<p>This register shows the raw status of the interrupts that were triggered.</p> <p>vfe_irq_src(0) &lt;= camif_error_irq;  vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer  BUS_OPERATION_STATUS register for the precise  OVERWRITE_IRQ.  vfe_irq_src(2) &lt;= spare_0_irq;  vfe_irq_src(3) &lt;= spare_1_irq;  vfe_irq_src(4) &lt;= bus_error_irq;  vfe_irq_src(5) &lt;= spare_2_irq;  vfe_irq_src(6) &lt;= spare_3_irq;  vfe_irq_src(7) &lt;= violation_irq;  vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq;  vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq;  vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq;  vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq;  vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq;  vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq;  vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq;  vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq;  vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq;  vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq;  vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq;  vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq;  vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq;  vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq;  vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq;  vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq;  vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq;  vfe_irq_src(25) &lt;= dsp_error_irq;  vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq;  vfe_irq_src(27) &lt;= spare_4_irq;  vfe_irq_src(28) &lt;= diag_irq;  vfe_irq_src(29) &lt;= rdi0_sof_irq;  vfe_irq_src(30) &lt;= rdi1_sof_irq;  vfe_irq_src(31) &lt;= rdi2_sof_irq;</p>

**0x0021007C MMSS\_A\_VFE\_0\_VIOLATION\_STATUS****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000

When violation IRQ is asserted, software user is encouraged to read this register to determine the violation module in the image processing pipeline. This field is asserted when the processing module's incoming frame timing strobes are misaligned due to module or sensor misconfiguration or violation of minimum horizontal blanking interval (32 cycles) and vertical blanking interval (16 lines).

Qualcomm  
2019-09-18 18:26:30 PDT  
xiaoyongjie15@huqin.com

**MMSS\_A\_VFE\_0\_VIOLATION\_STATUS**

Bits	Name	Description
5:0	STATUS	<p>This register value determines the first violation module in the image processing pipeline. A value of zero means no violation.</p> <p>violation(1) = camif_violation  violation(2) = dsp_violation  violation(3) = pedestal_violation  violation(4) = black_violation  violation(5) = demux_violation  violation(6) = hdr_recon_violation  violation(7) = hdr_mac_violation  violation(8) = bpc_abf_violation  violation(9) = pdaf_violation  violation(10) = black_level_violation  violation(11) = rolloff_violation  violation(12) = gic_violation  violation(13) = demo_violation  violation(14) = cac_snr_violation  violation(15) = color_correct_violation  violation(16) = gtm_violation  violation(17) = rgb_lut_violation  violation(18) = ltm_violation  violation(19) = reserved  violation(20) = chroma_enhan_violation  violation(21) = chroma_suppress_violation  violation(22) = skin_enhan_violation  violation(23) = color_xform_enc_violation  violation(24) = color_xform_view_violation  violation(25) = color_xform_vid_violation  violation(26) = scale_enc_y_violation  violation(27) = scale_enc_cbcr_violation  violation(28) = scale_view_y_violation  violation(29) = scale_view_cbcr_violation  violation(30) = scale_vid_y_violation  violation(31) = scale_vid_cbcr_violation  violation(32) = crop_enc_y_violation  violation(33) = crop_enc_cbcr_violation  violation(34) = crop_view_y_violation  violation(35) = crop_view_cbcr_violation  violation(36) = crop_vid_y_violation  violation(37) = crop_vid_cbcr_violation  violation(38) = realign_buf_y_violation  violation(39) = realign_buf_cb_violation  violation(40) = realign_buf_cr_violation</p>

**0x0021405C MMSS\_A\_VFE\_1\_IRQ\_MASK\_0****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x80000000**MMSS\_A\_VFE\_1\_IRQ\_MASK\_0**

Bits	Name	Description
31:0	MASK	<p>This mask is used to enable/disable an interrupt source for the VFE interrupt to the ARM.</p> <ul style="list-style-type: none"> <li>* When a bit is set (1), the interrupt source is enabled.</li> <li>* When clear (0), the interrupt source is disabled.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00214060 MMSS\_A\_VFE\_1\_IRQ\_MASK\_1****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_1\_IRQ\_MASK\_1**

Bits	Name	Description
31:0	MASK	<p>This mask is used to enable/disable an interrupt source for the VFE interrupt to the ARM.</p> <ul style="list-style-type: none"> <li>* When a bit is set (1), the interrupt source is enabled.</li> <li>* When clear (0), the interrupt source is disabled.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_error_irq; vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer BUS_OPERATION_STATUS register for the precise OVERWRITE_IRQ. vfe_irq_src(2) &lt;= spare_0_irq; vfe_irq_src(3) &lt;= spare_1_irq; vfe_irq_src(4) &lt;= bus_error_irq; vfe_irq_src(5) &lt;= spare_2_irq; vfe_irq_src(6) &lt;= spare_3_irq; vfe_irq_src(7) &lt;= violation_irq; vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq; vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq; vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq; vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq; vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq; vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq; vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq; vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq; vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq; vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq; vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq; vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq; vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq; vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq; vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq; vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq; vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq; vfe_irq_src(25) &lt;= dsp_error_irq; vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq; vfe_irq_src(27) &lt;= spare_4_irq; vfe_irq_src(28) &lt;= diag_irq; vfe_irq_src(29) &lt;= rdi0_sof_irq; vfe_irq_src(30) &lt;= rdi1_sof_irq;</pre>
		vfe_irq_src(31) <= rdi2_sof_irq;

**0x00214064 MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_0****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_0**

Bits	Name	Description
31:0	CLEAR	<p>This register must be programmed first before it used to clear an interrupt source.</p> <p>* Setting (1) a bit clears the interrupt status in IRQ_STATUS_0 after the global clear command is applied.</p> <p>* Setting (0) a bit retains the corresponding status bit unchanged.</p> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00214068 MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_1****Type:** RW**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_1**

Bits	Name	Description
31:0	CLEAR	<p>This register must be programmed first before it used to clear an interrupt source.</p> <ul style="list-style-type: none"> <li>* Setting (1) a bit clears the interrupt status in IRQ_STATUS_1 after the global clear command is applied.</li> <li>* Setting (0) a bit retains the corresponding status bit unchanged.</li> </ul> <pre>vfe_irq_src(0) &lt;= camif_error_irq; vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer BUS_OPERATION_STATUS register for the precise OVERWRITE_IRQ. vfe_irq_src(2) &lt;= spare_0_irq; vfe_irq_src(3) &lt;= spare_1_irq; vfe_irq_src(4) &lt;= bus_error_irq; vfe_irq_src(5) &lt;= spare_2_irq; vfe_irq_src(6) &lt;= spare_3_irq; vfe_irq_src(7) &lt;= violation_irq; vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq; vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq; vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq; vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq; vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq; vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq; vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq; vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq; vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq; vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq; vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq; vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq; vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq; vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq; vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq; vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq; vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq; vfe_irq_src(25) &lt;= dsp_error_irq; vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq; vfe_irq_src(27) &lt;= spare_4_irq; vfe_irq_src(28) &lt;= diag_irq; vfe_irq_src(29) &lt;= rdi0_sof_irq;</pre>
		<pre>vfe_irq_src(30) &lt;= rdi1_sof_irq; vfe_irq_src(31) &lt;= rdi2_sof_irq;</pre>

**0x0021406C MMSS\_A\_VFE\_1\_IRQ\_STATUS\_0****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_1\_IRQ\_STATUS\_0**

Bits	Name	Description
31:0	STATUS	<p>This register shows the raw status of the interrupts that were triggered.</p> <pre>vfe_irq_src(0) &lt;= camif_sof_irq; vfe_irq_src(1) &lt;= camif_eof_irq; vfe_irq_src(2) &lt;= camif_epoch0_irq; vfe_irq_src(3) &lt;= camif_epoch1_irq; vfe_irq_src(4) &lt;= reg_update_irq; vfe_irq_src(5) &lt;= rdi0_reg_update_irq; vfe_irq_src(6) &lt;= rdi1_reg_update_irq; vfe_irq_src(7) &lt;= rdi2_reg_update_irq; vfe_irq_src(8) &lt;= image_master_0_ping_pong_irq; vfe_irq_src(9) &lt;= image_master_1_ping_pong_irq; vfe_irq_src(10) &lt;= image_master_2_ping_pong_irq; vfe_irq_src(11) &lt;= image_master_3_ping_pong_irq; vfe_irq_src(12) &lt;= image_master_4_ping_pong_irq; vfe_irq_src(13) &lt;= image_master_5_ping_pong_irq; vfe_irq_src(14) &lt;= image_master_6_ping_pong_irq; vfe_irq_src(15) &lt;= stats_aec_bg_ping_pong_irq; vfe_irq_src(16) &lt;= stats_hdr_be_ping_pong_irq; vfe_irq_src(17) &lt;= stats_awb_bg_ping_pong_irq; vfe_irq_src(18) &lt;= stats_baf_early_done_irq; vfe_irq_src(19) &lt;= stats_hdr_bhists_ping_pong_irq; vfe_irq_src(20) &lt;= stats_rs_ping_pong_irq; vfe_irq_src(21) &lt;= stats_cs_ping_pong_irq; vfe_irq_src(22) &lt;= stats_ihist_ping_pong_irq; vfe_irq_src(23) &lt;= stats_skin_bhists_ping_pong_irq; vfe_irq_src(24) &lt;= rd_ping_pong_irq; vfe_irq_src(25) &lt;= image_composite_done_0_irq; vfe_irq_src(26) &lt;= image_composite_done_1_irq; vfe_irq_src(27) &lt;= image_composite_done_2_irq; vfe_irq_src(28) &lt;= image_composite_done_3_irq; vfe_irq_src(29) &lt;= stats_composite_done_0_irq; vfe_irq_src(30) &lt;= stats_composite_done_1_irq; vfe_irq_src(31) &lt;= reset_ack_irq;</pre>

**0x00214070 MMSS\_A\_VFE\_1\_IRQ\_STATUS\_1****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000**MMSS\_A\_VFE\_1\_IRQ\_STATUS\_1**

Bits	Name	Description
31:0	STATUS	<p>This register shows the raw status of the interrupts that were triggered.</p> <p>vfe_irq_src(0) &lt;= camif_error_irq;  vfe_irq_src(1) &lt;= stats_overwrite_irq; Refer  BUS_OPERATION_STATUS register for the precise  OVERWRITE_IRQ.  vfe_irq_src(2) &lt;= spare_0_irq;  vfe_irq_src(3) &lt;= spare_1_irq;  vfe_irq_src(4) &lt;= bus_error_irq;  vfe_irq_src(5) &lt;= spare_2_irq;  vfe_irq_src(6) &lt;= spare_3_irq;  vfe_irq_src(7) &lt;= violation_irq;  vfe_irq_src(8) &lt;= bus_bdg_halt_ack_irq;  vfe_irq_src(9) &lt;= image_master_0_bus_overflow_irq;  vfe_irq_src(10) &lt;= image_master_1_bus_overflow_irq;  vfe_irq_src(11) &lt;= image_master_2_bus_overflow_irq;  vfe_irq_src(12) &lt;= image_master_3_bus_overflow_irq;  vfe_irq_src(13) &lt;= image_master_4_bus_overflow_irq;  vfe_irq_src(14) &lt;= image_master_5_bus_overflow_irq;  vfe_irq_src(15) &lt;= image_master_6_bus_overflow_irq;  vfe_irq_src(16) &lt;= stats_hdr_be_bus_overflow_irq;  vfe_irq_src(17) &lt;= stats_awb_bg_bus_overflow_irq;  vfe_irq_src(18) &lt;= stats_baf_bus_overflow_irq;  vfe_irq_src(19) &lt;= stats_hdr_bhst_bus_overflow_irq;  vfe_irq_src(20) &lt;= stats_rs_bus_overflow_irq;  vfe_irq_src(21) &lt;= stats_cs_bus_overflow_irq;  vfe_irq_src(22) &lt;= stats_ihist_bus_overflow_irq;  vfe_irq_src(23) &lt;= stats_skin_bhst_bus_overflow_irq;  vfe_irq_src(24) &lt;= stats_aec_bg_bus_overflow_irq;  vfe_irq_src(25) &lt;= dsp_error_irq;  vfe_irq_src(26) &lt;= stats_baf_ping_pong_irq;  vfe_irq_src(27) &lt;= spare_4_irq;  vfe_irq_src(28) &lt;= diag_irq;  vfe_irq_src(29) &lt;= rdi0_sof_irq;  vfe_irq_src(30) &lt;= rdi1_sof_irq;  vfe_irq_src(31) &lt;= rdi2_sof_irq;</p>

**0x0021407C MMSS\_A\_VFE\_1\_VIOLATION\_STATUS****Type:** R**Clock:** VFE\_CLK**Reset State:** 0x00000000

When violation IRQ is asserted, software user is encouraged to read this register to determine the violation module in the image processing pipeline. This field is asserted when the processing module's incoming frame timing strobes are misaligned due to module or sensor misconfiguration or violation of minimum horizontal blanking interval (32 cycles) and vertical blanking interval (16 lines).

Qualcomm  
2019-09-18 18:26:30 PDT  
xiaoyongjie15@huqin.com

**MMSS\_A\_VFE\_1\_VIOLATION\_STATUS**

Bits	Name	Description
5:0	STATUS	<p>This register value determines the first violation module in the image processing pipeline. A value of zero means no violation.</p> <p>violation(1) = camif_violation  violation(2) = dsp_violation  violation(3) = pedestal_violation  violation(4) = black_violation  violation(5) = demux_violation  violation(6) = hdr_recon_violation  violation(7) = hdr_mac_violation  violation(8) = bpc_abf_violation  violation(9) = pdaf_violation  violation(10) = black_level_violation  violation(11) = rolloff_violation  violation(12) = gic_violation  violation(13) = demo_violation  violation(14) = cac_snr_violation  violation(15) = color_correct_violation  violation(16) = gtm_violation  violation(17) = rgb_lut_violation  violation(18) = ltm_violation  violation(19) = reserved  violation(20) = chroma_enhan_violation  violation(21) = chroma_suppress_violation  violation(22) = skin_enhan_violation  violation(23) = color_xform_enc_violation  violation(24) = color_xform_view_violation  violation(25) = color_xform_vid_violation  violation(26) = scale_enc_y_violation  violation(27) = scale_enc_cbcr_violation  violation(28) = scale_view_y_violation  violation(29) = scale_view_cbcr_violation  violation(30) = scale_vid_y_violation  violation(31) = scale_vid_cbcr_violation  violation(32) = crop_enc_y_violation  violation(33) = crop_enc_cbcr_violation  violation(34) = crop_view_y_violation  violation(35) = crop_view_cbcr_violation  violation(36) = crop_vid_y_violation  violation(37) = crop_vid_cbcr_violation  violation(38) = realign_buf_y_violation  violation(39) = realign_buf_cb_violation  violation(40) = realign_buf_cr_violation</p>

# 11 Low power audio subsystem registers

---

**0x00000000 TLMM\_GPIO\_ISLAND\_CFG0**

**Type:** RW

**Clock:** CSR\_WR\_CLK

**Reset State:** 0x000000C1

**TLMM\_GPIO\_ISLAND\_CFG0**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: TS_AUX_GPIO 0x1: LPI_SPI_1_CS1_N 0x2: SYNC_OUT 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00000004 TLMM\_GPIO\_ISLAND\_IN\_OUT0****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT0**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00001000 TLMM\_GPIO\_ISLAND\_CFG1****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG1**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: LPI_PWR_EN_GPIO
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00001004 TLMM\_GPIO\_ISLAND\_IN\_OUT1****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000

**TLMM\_GPIO\_ISLAND\_IN\_OUT1**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00002000 TLMM\_GPIO\_ISLAND\_CFG2****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG2**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_I2C_3_SDA 0x2: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00002004 TLMM\_GPIO\_ISLAND\_IN\_OUT2****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000

**TLMM\_GPIO\_ISLAND\_IN\_OUT2**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00002010 TLMM\_GPIO\_ISLAND\_CFG3****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG3**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_I2C_3_SCL 0x2: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00002014 TLMM\_GPIO\_ISLAND\_IN\_OUT3****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT3**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00003000 TLMM\_GPIO\_ISLAND\_CFG4****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG4**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_I2C_2_SDA 0x2: LPI_SPI_2_CS_N 0x3: LPI_SPI_1_CS2_N 0x4: LPI_TER_I2S_CLK 0x5: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00003004 TLMM\_GPIO\_ISLAND\_IN\_OUT4****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT4**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00003010 TLMM\_GPIO\_ISLAND\_CFG5****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG5**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_I2C_2_SCL 0x2: LPI_SPI_2_CLK 0x3: LPI_SPI_1_CS3_N 0x4: LPI_TER_I2S_WS 0x5: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00003014 TLMM\_GPIO\_ISLAND\_IN\_OUT5****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT5**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00004000 TLMM\_GPIO\_ISLAND\_CFG6****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG6**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_3_TX 0x2: LPI_SPI_2_MOSI 0x3: LPI_SPI_1_CS2_N 0x4: LPI_TER_I2S_DATA0 0x5: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00004004 TLMM\_GPIO\_ISLAND\_IN\_OUT6****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT6**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00004010 TLMM\_GPIO\_ISLAND\_CFG7****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG7**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_3_RX 0x2: LPI_SPI_2_MISO 0x3: LPI_SPI_1_CS3_N 0x4: LPI_TER_I2S_DATA1 0x5: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00004014 TLMM\_GPIO\_ISLAND\_IN\_OUT7****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT7**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005000 TLMM\_GPIO\_ISLAND\_CFG8****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG8**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_CS_N 0x2: LPI_I2C_1_SDA 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005004 TLMM\_GPIO\_ISLAND\_IN\_OUT8****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT8**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005010 TLMM\_GPIO\_ISLAND\_CFG9****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG9**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_CLK 0x2: LPI_I2C_1_SCL 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005014 TLMM\_GPIO\_ISLAND\_IN\_OUT9****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT9**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005020 TLMM\_GPIO\_ISLAND\_CFG10****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG10**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_MOSI 0x2: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005024 TLMM\_GPIO\_ISLAND\_IN\_OUT10****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT10**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005030 TLMM\_GPIO\_ISLAND\_CFG11****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG11**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_MISO 0x2: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005034 TLMM\_GPIO\_ISLAND\_IN\_OUT11****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT11**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005040 TLMM\_GPIO\_ISLAND\_CFG16****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG16**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_CS2_N 0x2: LPI_MCLK1 0x3: LPI_QUA_I2S_CLK 0x4: LPI_MCLK0 0x5: LPI_SPI_3_CLK 0x6: LPI_UART_4_TX
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005044 TLMM\_GPIO\_ISLAND\_IN\_OUT16****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT16**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00005050 TLMM\_GPIO\_ISLAND\_CFG17****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG17**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_SPI_1_CS3_N 0x2: LPI_MCLK1 0x3: LPI_QUA_I2S_WS 0x4: LPI_MCLK0 0x5: LPI_SPI_3_CS_N 0x6: LPI_UART_4_RX
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00005054 TLMM\_GPIO\_ISLAND\_IN\_OUT17****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT17**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00006000 TLMM\_GPIO\_ISLAND\_CFG12****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG12**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_1_TX 0x2: LPI_SPI_3_CLK 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00006004 TLMM\_GPIO\_ISLAND\_IN\_OUT12****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT12**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00006010 TLMM\_GPIO\_ISLAND\_CFG13****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG13**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_1_RX 0x2: LPI_SPI_3_CS_N 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00006014 TLMM\_GPIO\_ISLAND\_IN\_OUT13****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT13**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00007000 TLMM\_GPIO\_ISLAND\_CFG14****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG14**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_2_TX 0x2: LPI_SPI_3_MOSI 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00007004 TLMM\_GPIO\_ISLAND\_IN\_OUT14****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT14**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00007010 TLMM\_GPIO\_ISLAND\_CFG15****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG15**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_UART_2_RX 0x2: LPI_SPI_3_MISO 0x3: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00007014 TLMM\_GPIO\_ISLAND\_IN\_OUT15****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT15**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008000 TLMM\_GPIO\_ISLAND\_CFG18****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG18**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_CLK 0x2: LPI_MCLK0 0x3: LPI_QUA_I2S_CLK 0x4: RESERVED
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008004 TLMM\_GPIO\_ISLAND\_IN\_OUT18****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT18**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008010 TLMM\_GPIO\_ISLAND\_CFG19****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG19**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_SYNC 0x2: LPI_AUD_SB_CLK 0x3: LPI_QUA_I2S_WS
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008014 TLMM\_GPIO\_ISLAND\_IN\_OUT19****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT19**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008020 TLMM\_GPIO\_ISLAND\_CFG20****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG20**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_TX0 0x2: LPI_AUD_SB_DATA0 0x3: LPI_QUA_I2S_DATA0
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008024 TLMM\_GPIO\_ISLAND\_IN\_OUT20****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT20**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008030 TLMM\_GPIO\_ISLAND\_CFG21****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG21**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_RX0 0x2: LPI_AUD_SB_DATA1 0x3: LPI_QUA_I2S_DATA1
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008034 TLMM\_GPIO\_ISLAND\_IN\_OUT21****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT21**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008040 TLMM\_GPIO\_ISLAND\_CFG22****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG22**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_RX0_COMP 0x2: LPI_AUD_CDC_INT1 0x3: LPI_QUA_I2S_DATA2
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008044 TLMM\_GPIO\_ISLAND\_IN\_OUT22****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT22**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008050 TLMM\_GPIO\_ISLAND\_CFG23****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG23**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM0_RX1 0x2: LPI_AUD_CDC_INT2 0x3: LPI_QUA_I2S_DATA3
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008054 TLMM\_GPIO\_ISLAND\_IN\_OUT23****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT23**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008060 TLMM\_GPIO\_ISLAND\_CFG24****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG24**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_RX1_COMP 0x2: LPI_AUD_CDC_RSTN
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008064 TLMM\_GPIO\_ISLAND\_IN\_OUT24****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT24**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00008070 TLMM\_GPIO\_ISLAND\_CFG25****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG25**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_CDC_PDM_RX2
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00008074 TLMM\_GPIO\_ISLAND\_IN\_OUT25****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT25**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00009000 TLMM\_GPIO\_ISLAND\_CFG26****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG26**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_DMIC1_CLK
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00009004 TLMM\_GPIO\_ISLAND\_IN\_OUT26****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT26**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x00009010 TLMM\_GPIO\_ISLAND\_CFG27****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG27**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_DMIC1_DATA
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x00009014 TLMM\_GPIO\_ISLAND\_IN\_OUT27****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT27**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0000A000 TLMM\_GPIO\_ISLAND\_CFG28****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG28**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_DMIC2_CLK 0x2: LPI_QUA_I2S_CLK 0x3: LPI_MCLK1 0x4: LPI_SPI_3_MOSI 0x5: LPI_UART_4_TX
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0000A004 TLMM\_GPIO\_ISLAND\_IN\_OUT28****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT28**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0000A010 TLMM\_GPIO\_ISLAND\_CFG29****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG29**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_DMIC2_DATA 0x2: LPI_QUA_I2S_WS 0x3: LPI_SPI_3_MISO 0x4: LPI_UART_4_RX
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0000A014 TLMM\_GPIO\_ISLAND\_IN\_OUT29****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT29**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0000B000 TLMM\_GPIO\_ISLAND\_CFG30****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG30**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_QCA_SB_CLK 0x2: LPI_QUA_I2S_DATA0
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0000B004 TLMM\_GPIO\_ISLAND\_IN\_OUT30****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x00000000**TLMM\_GPIO\_ISLAND\_IN\_OUT30**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0000B010 TLMM\_GPIO\_ISLAND\_CFG31****Type:** RW**Clock:** CSR\_WR\_CLK**Reset State:** 0x000000C1**TLMM\_GPIO\_ISLAND\_CFG31**

Bits	Name	Description
15:11	GPIO_MISC_CFG	
10	GPIO_HIHYS_EN	
9	GPIO_OE	
8:6	GPIO_HDRIVE	0x0: DRV_2_MA 0x1: DRV_4_MA 0x2: DRV_6_MA 0x3: DRV_8_MA 0x4: DRV_10_MA 0x5: DRV_12_MA 0x6: DRV_14_MA 0x7: DRV_16_MA
5:2	GPIO_FUNC_SEL	0x0: GPIO 0x1: LPI_QCA_SB_DATA0 0x2: LPI_QUA_I2S_DATA1
1:0	GPIO_PULL	0x0: NO_PULL 0x1: PULL_DOWN 0x2: KEEPER 0x3: PULL_UP

**0x0000B014 TLMM\_GPIO\_ISLAND\_IN\_OUT31**

Type: RW

Clock: CSR\_WR\_CLK

Reset State: 0x00000000

**TLMM\_GPIO\_ISLAND\_IN\_OUT31**

Bits	Name	Type	Description
1	GPIO_OUT	RW	
0	GPIO_IN	R	

**0x0000F000 TLMM\_GPIO\_ISLAND\_CODEC\_RST**

Type: RW

Reset State: 0x00000000

**TLMM\_GPIO\_ISLAND\_CODEC\_RST**

Bits	Name	Description
0	CODEC_RST	

**0x00001000+ LPAIF\_I2S\_CTLa, a=[0..4]****0x1000\*a**

Type: RW

Clock: CC\_LPAIF\_HCLK

Reset State: 0x000F0004

**LPAIF\_I2S\_CTLa**

Bits	Name	Description
31	RESET	0x0: DISABLE 0x1: ENABLE
22	EN_LONG_RATE	Allow for frame size to be longer than I2S sample bitwidth, BIT_WIDTH 0x0: DISABLE 0x1: ENABLE
21:16	LONG_RATE	Used as WS rate when EN_LONG_RATE=1

**LPAIF\_I2S\_CTLa (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
15	LOOPBACK	<p>SW: RW, HW: R Loopback control bit. 0x0: DISABLE (DISABLE) 0x1: ENABLE (ENABLE (only applicable to a=1=codec_mic, a=2=mi2s, &amp; a=4=sec mic. If ENABLE is set for a=1, codec speaker is looped back to both wires of codec mic. For a=2, mi2s is looped back on itself. For a=4 sec speaker is looped backed to sec mic))</p>
14	SPKR_EN	<p>SW: RW, HW: R Enable for the speaker direction operation (out of the MSM). I2S interface will start transmitting serial data when enabled and trying to access a read DMA channel. Set the enable after the other control bits (speaker related, bit_width, ws_src) are configured. Disable before changing the other bits. 0x0: DISABLE (DISABLE) 0x1: ENABLE (ENABLE)</p>
13:10	SPKR_MODE	<p>SW: RW, HW: R Controls which I2S lines are used (if supported by HW) and multichannel operation 0x0: NONE (NONE) 0x1: SD0 (SD0 (data sent to SD0 only- stereo or mono)) 0x2: SD1 (SD1 (data sent to SD1 only- stereo or mono)) 0x3: SD2 (SD2 (data sent to SD2 only- stereo or mono)) 0x4: SD3 (SD3 (data sent to SD3 only- stereo or mono)) 0x5: QUAD01 (QUAD01 (SD0 &amp; SD1 = quad channel mode)) 0x6: QUAD23 (QUAD23 (SD2 &amp; SD3 = quad channel mode)) 0x7: ENUM_6CH (6CH (SD0, SD1 &amp; SD2 = 6 channel mode)) 0x8: ENUM_8CH (8CH (SD0, SD1, SD2 &amp; SD3 = 8 channel mode))</p>

**LPAIF\_I2S\_CTLa (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
9	SPKR_MONO	<p>SW: RW, HW: R            In mono mode, the speaker path of I2S (out of MSM) will take each mono sample from memory and transmit it during both the right and left portions of the word select. Only applicable when a single I2S line is used in SPKR_MODE.</p> <p>0x0: STEREO            (STEREO)            0x1: MONO            (MONO)</p>
8	MIC_EN	<p>SW: RW, HW: R            Enable for the microphone direction operation (into the MSM). I2S interface will start capturing data when enabled and trying to access a write DMA channel. Set the enable after the other control bits (mic related, bit_width, ws_src) are configured. Disable before changing the other bits.</p> <p>0x0: DISABLE            (DISABLE)            0x1: ENABLE            (ENABLE)</p>
7:4	MIC_MODE	<p>SW: RW, HW: R            Controls which I2S lines are used (if supported by HW) and multichannel operation</p> <p>0x0: NONE            (NONE)            0x1: SD0            (SD0 (data taken from SD0 only- stereo or mono))            0x2: SD1            (SD1 (data taken from SD1 only- stereo or mono))            0x3: SD2            (SD2 (data taken from SD2 only- stereo or mono))            0x4: SD3            (SD3 (data taken from SD3 only- stereo or mono))            0x5: QUAD01            (QUAD01 (SD0 &amp; SD1 = quad channel mode))            0x6: QUAD23            (QUAD23 (SD2 &amp; SD3 = quad channel mode))            0x7: ENUM_6CH            (6CH (SD0, SD1 &amp; SD2 = 6 channel mode))            0x8: ENUM_8CH            (8CH (SD0, SD1, SD2 &amp; SD3 = 8 channel mode))</p>

**LPAIF\_I2S\_CTLa (cont.)**

Bits	Name	Description
3	MIC_MONO	SW: RW, HW: R Controls whether only the left word is stored in memory (mono) or the left and right words are stored (stereo). In mono mode, the data in the right portion of WS is ignored. Only applicable when a single I2S line is used in MIC_MODE. 0x0: STEREO (STEREO) 0x1: MONO (MONO)
2	WS_SRC	SW: RW, HW: R Controls the source of the word select, internally generated or taken from outside. Does not affect the clock. Master/Slave bit clock mode is controlled in LPASS clk_ctl. 0x0: INT (INT (master mode)) 0x1: EXT (EXT (slave mode))
1:0	BIT_WIDTH	SW: RW, HW: R Controls the bit width of the I2S samples 0x0: ENUM_16 (16) 0x1: ENUM_24 (24) 0x2: ENUM_32 (32) 0x3: ENUM_25 (25)

**0x00020000+ LPAIF\_INT\_I2S\_CTLa, a=[0..6]  
0x1000\*a**

**Type:** RW  
**Clock:** CC\_LPAIF\_HCLK  
**Reset State:** 0x000F0004

Registers for Internal MI2S interfaces going to internal codec.

**LPAIF\_INT\_I2S\_CTLa**

Bits	Name	Description
31	RESET	0x0: DISABLE 0x1: ENABLE

**LPAIF\_INT\_I2S\_CTLa (cont.)**

Bits	Name	Description
22	EN_LONG_RATE	Allow for frame size to be longer than I2S sample bitwidth, BIT_WIDTH 0x0: DISABLE 0x1: ENABLE
21:16	LONG_RATE	Used as WS rate when EN_LONG_RATE=1
15	LOOPBACK	SW: RW, HW: R Loopback control bit. 0x0: DISABLE (DISABLE) 0x1: ENABLE (ENABLE (only applicable to a=1=codec_mic, a=2=mi2s, & a=4=sec mic. If ENABLE is set for a=1, codec speaker is looped back to both wires of codec mic. For a=2, mi2s is looped back on itself. For a=4 sec speaker is looped backed to sec mic))
14	SPKR_EN	SW: RW, HW: R Enable for the speaker direction operation (out of the MSM). I2S interface will start transmitting serial data when enabled and trying to access a read DMA channel. Set the enable after the other control bits (speaker related, bit_width, ws_src) are configured. Disable before changing the other bits. 0x0: DISABLE (DISABLE) 0x1: ENABLE (ENABLE)
13:10	SPKR_MODE	SW: RW, HW: R Controls which I2S lines are used (if supported by HW) and multichannel operation 0x0: NONE (NONE) 0x1: SD0 (SD0 (data sent to SD0 only- stereo or mono)) 0x2: SD1 (SD1 (data sent to SD1 only- stereo or mono)) 0x3: SD2 (SD2 (data sent to SD2 only- stereo or mono)) 0x4: SD3 (SD3 (data sent to SD3 only- stereo or mono)) 0x5: QUAD01 (QUAD01 (SD0 & SD1 = quad channel mode)) 0x6: QUAD23 (QUAD23 (SD2 & SD3 = quad channel mode)) 0x7: ENUM_6CH (6CH (SD0, SD1 & SD2 = 6 channel mode)) 0x8: ENUM_8CH (8CH (SD0, SD1, SD2 & SD3 = 8 channel mode))

**LPAIF\_INT\_I2S\_CTLa (cont.)**

Bits	Name	Description
9	SPKR_MONO	<p>SW: RW, HW: R            In mono mode, the speaker path of I2S (out of MSM) will take each mono sample from memory and transmit it during both the right and left portions of the word select. Only applicable when a single I2S line is used in SPKR_MODE.</p> <p>0x0: STEREO            (STEREO)            0x1: MONO            (MONO)</p>
8	MIC_EN	<p>SW: RW, HW: R            Enable for the microphone direction operation (into the MSM). I2S interface will start capturing data when enabled and trying to access a write DMA channel. Set the enable after the other control bits (mic related, bit_width, ws_src) are configured. Disable before changing the other bits.</p> <p>0x0: DISABLE            (DISABLE)            0x1: ENABLE            (ENABLE)</p>
7:4	MIC_MODE	<p>SW: RW, HW: R            Controls which I2S lines are used (if supported by HW) and multichannel operation</p> <p>0x0: NONE            (NONE)            0x1: SD0            (SD0 (data taken from SD0 only- stereo or mono))            0x2: SD1            (SD1 (data taken from SD1 only- stereo or mono))            0x3: SD2            (SD2 (data taken from SD2 only- stereo or mono))            0x4: SD3            (SD3 (data taken from SD3 only- stereo or mono))            0x5: QUAD01            (QUAD01 (SD0 &amp; SD1 = quad channel mode))            0x6: QUAD23            (QUAD23 (SD2 &amp; SD3 = quad channel mode))            0x7: ENUM_6CH            (6CH (SD0, SD1 &amp; SD2 = 6 channel mode))            0x8: ENUM_8CH            (8CH (SD0, SD1, SD2 &amp; SD3 = 8 channel mode))</p>

**LPAIF\_INT\_I2S\_CTLa (cont.)**

Bits	Name	Description
3	MIC_MONO	SW: RW, HW: R Controls whether only the left word is stored in memory (mono) or the left and right words are stored (stereo). In mono mode, the data in the right portion of WS is ignored. Only applicable when a single I2S line is used in MIC_MODE. 0x0: STEREO (STEREO) 0x1: MONO (MONO)
2	WS_SRC	SW: RW, HW: R Controls the source of the word select, internally generated or taken from outside. Does not affect the clock. Master/Slave bit clock mode is controlled in LPASS clk_ctl. 0x0: INT (INT (master mode)) 0x1: EXT (EXT (slave mode))
1:0	BIT_WIDTH	SW: RW, HW: R Controls the bit width of the I2S samples 0x0: ENUM_16 (16) 0x1: ENUM_24 (24) 0x2: ENUM_32 (32) 0x3: ENUM_25 (25)

## 12 LPASS digital codec core registers

---

### 0x152C1308 LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CTL

Type: RW

Clock: rif\_clk

Reset State: 0x00000002

Reset Name: rif\_async\_reset\_n

Page A register

#### LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CTL

Bits	Name	Description
5	RESET	
4	CLK_EN	
3:0	PCM_RATE	0x0: F_8K (-) 0x2: F_24K (-) 0x4: F_48K (-)

### 0x152C130C LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CFG0

Type: RW

Clock: rif\_clk

Reset State: 0x00000000

Reset Name: rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CFG0**

Bits	Name	Description
1	SPKPROT_GN_MX	Specifies digital gain 0x0: DISABLE (-) 0x1: ENABLE (-)
0	TX_PH_EQU_MX	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C1318 LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000002**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CTL**

Bits	Name	Description
5	RESET	
4	CLK_EN	
3:0	PCM_RATE	0x0: F_8K (-) 0x2: F_24K (-) 0x4: F_48K (-)

**0x152C131C LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CFG0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CFG0**

Bits	Name	Description
1	SPKPROT_GN_MX	Specifies digital gain 0x0: DISABLE (-) 0x1: ENABLE (-)
0	TX_PH_EQU_MX	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C1328 LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000002**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CTL**

Bits	Name	Description
5	RESET	
4	CLK_EN	
3:0	PCM_RATE	0x0: F_8K (-) 0x2: F_24K (-) 0x4: F_48K (-)

**0x152C132C LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CFG0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CFG0**

Bits	Name	Description
1	SPKPROT_GN_MX	Specifies digital gain 0x0: DISABLE (-) 0x1: ENABLE (-)
0	TX_PH_EQU_MX	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C1338 LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000002**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CTL**

Bits	Name	Description
5	RESET	
4	CLK_EN	
3:0	PCM_RATE	0x0: F_8K (-) 0x2: F_24K (-) 0x4: F_48K (-)

**0x152C133C LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CFG0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page A register

**LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CFG0**

Bits	Name	Description
1	SPKPROT_GN_MX	Specifies digital gain 0x0: DISABLE (-) 0x1: ENABLE (-)
0	TX_PH_EQU_MX	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C11A4 LPASS\_CDC\_RX7\_RX\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Controls Rx primary path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_CTL**

Bits	Name	Description
6	RESET	Rx data path reset 0x0: DISABLE (-) 0x1: ENABLE (-)
5	CLK_EN	Rx path clock enable 0x0: DISABLE (-) 0x1: ENABLE (-)
4	PGA_MUTE_EN	Enables Rx primary path PGA mute 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_CTL (cont.)**

Bits	Name	Description
3:0	PCM_RATE	Rx primary path input sample rate. 0x0: F_8K (-) 0x1: F_16K (-) 0x3: F_32K (-) 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-) 0x7: F_384K (-) 0x8: F_44P1K_CADENCE (-)

**0x152C11A8 LPASS\_CDC\_RX7\_RX\_PATH\_CFG0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_CFG0**

Bits	Name	Description
7	SPLINE_SRC_EN	Rx Path will gate inputs to and outputs from SPLINE SRC 0x0: DISABLE (-) 0x1: ENABLE (-)
4	ANC_EN	Rx path will gate inputs to and outputs from ANC block. This field is also routed to CLSH for ANC data gating. 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_CFG0 (cont.)**

Bits	Name	Description
3	DLY_ZN_EN	Enables DLY_Zn 0x0: DISABLE (-) 0x1: ENABLE (-)
2	HD2_EN	Enables HD2. 0x0: DISABLE (-) 0x1: ENABLE (-)
1	CMP_EN	Enables compander. 0x0: DISABLE (-) 0x1: ENABLE (-)
0	PH_EQ_EN	Enables Linear phase 0x0: MINIMUM_PHASE (-) 0x1: LINEAR_PHASE (-)

**0x152C11AC LPASS\_CDC\_RX7\_RX\_PATH\_CFG1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000044**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_CFG1**

Bits	Name	Description
6	HPF_PBGC_EN	Enables HPF Passband Gain Cancellation Feature 0x0: DISABLE (-) 0x1: ENABLE (-)
5	RX_LPF_MX_SEL	Choose between including LPF or NOT 0x0: WO_LPF (-) 0x1: WITH_LPF (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_CFG1 (cont.)**

Bits	Name	Description
4	SIDETONE_EN	Rx path will gate inputs to and outputs from sidetone SRC 0x0: DISABLE (-) 0x1: ENABLE (-)
3	SPKR_RATE	specifies spkr path rate 4.8MHz/6.144MHz or 2.4Mhz/3.072MHz 0x0: FS_4P8_6P144 (-) 0x1: FS_2P4_3P072 (-)
2	HPF_EN	Enables Rx primary path HPF 0x0: DISABLE (-) 0x1: ENABLE (-)
1	VBAT_EN	Data path gating for Vbatt. 0x0: DISABLE (-) 0x1: ENABLE (-)
0	SMART_BST_EN	Data path gating for SmartBoost 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C11B0 LPASS\_CDC\_RX7\_RX\_PATH\_CFG2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x0000008F**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_CFG2**

Bits	Name	Description
7:5	ECHO_REF_PCM_RATE	Echo Reference Sample Rate 0x0: F_8K (-) 0x1: F_16K (-) 0x3: F_32K (-) 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-)
4	RX_US_MX_SEL	Choose between the path before or after mixing to downsample. 0x0: POST_MIXING (-) 0x1: PRE_MIXING (-)
3:2	HPF_CUT_OFF_FREQ_384	Specifies the cutoff frequency for the DC blocker inside the Rx chain for 384KHZ Sample Rate only. 0x0: CF_NEG_3DB_8HZ (-) 0x1: CF_NEG_3DB_150HZ (-) 0x2: CF_NEG_3DB_300HZ (-) 0x3: CF_NEG_3DB_0P96HZ (-)
1:0	HPF_CUT_OFF_FREQ	Specifies the cutoff frequency for the DC blocker inside the Rx chain. 0x0: CF_NEG_3DB_4HZ (-) 0x1: CF_NEG_3DB_75HZ (-) 0x2: CF_NEG_3DB_150HZ (-) 0x3: CF_NEG_3DB_0P48HZ (-)

**0x152C11B4 LPASS\_CDC\_RX7\_RX\_VOL\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Rx Main Path Vol Cntl Gain Register. Page B register

**LPASS\_CDC\_RX7\_RX\_VOL\_CTL**

Bits	Name	Description
7:0	PGA_GAIN	<p>Specifies the volume controller gain. The value is a two complement number. The supported gain range is from -84 dB to 40 dB (in increments of 0.5 dB). This GAIN field specifies the integer portion of the gain. For example:</p> <p>0x02 = +2 dB      0x01 = +1 dB      0x00 = 0 dB      0xFF = -1 dB      0xFE = -2 dB</p> <p>The HALF_DB bit field from the CDC_RXn_B6_CTL, n=[1..7] register adds an optional 0.5 dB to the integer portion of the gain. For example, to specify 5 dB gain, program GAIN=0x5 and set HALF_DB to 0; to specify 5.5 dB gain, program GAIN=0x5 and set HALF_DB to 1; to specify -0.5 dB gain, program GAIN=0xFF and set HALF_DB to 1.</p>

**0x152C11B8 LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Controls Rx mixing path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CTL**

Bits	Name	Description
5	MIX_CLK_EN	This clk_en will gate off all the clocks, fs for the mixing path. The input clk will still be the main Rx clk. When enabled, CLK_EN (main path) should also be enabled.
4	MIX_PGA_MUTE_EN	Enables Rx mixing path PGA mute 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CTL (cont.)**

Bits	Name	Description
3:0	MIX_PCM_RATE	Rx mixing path input sample rate 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-)

**0x152C11BC LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CFG****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000001E**Reset Name:** rif\_async\_reset\_n

Configures Mixing path. Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CFG**

Bits	Name	Description
4	MIX_HPF_PBGC_EN	Enables mixing path HPF Passband Gain Cancellation Feature 0x0: DISABLE (-) 0x1: ENABLE (-)
3:2	MIX_HPF_CUT_OFF_FREQ	Specifies the cutoff frequency for the DC blocker inside the Rx chain. 0x0: CF_NEG_3DB_4HZ (-) 0x1: CF_NEG_3DB_75HZ (-) 0x2: CF_NEG_3DB_150HZ (-) 0x3: CF_NEG_3DB_0P48HZ (-)
1	MIX_HPF_EN	Enables Rx Mixing Path HPF 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CFG (cont.)**

Bits	Name	Description
0	MIX_PH_EQ_EN	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C11C0 LPASS\_CDC\_RX7\_RX\_VOL\_MIX\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Rx Mixing Path Vol Cntl Gain Register, Page B register

**LPASS\_CDC\_RX7\_RX\_VOL\_MIX\_CTL**

Bits	Name	Description
7:0	MIX_PGA_GAIN	Specifies the volume controller gain. The value is a two complement number. The supported gain range is from -84 dB to 40 dB (in increments of 0.5 dB). This GAIN field specifies the integer portion of the gain. For example: 0x02 = +2 dB 0x01 = +1 dB 0x00 = 0 dB 0xFF = -1 dB 0xFE = -2 dB  The HALF_DB bit field from the CDC_RXn_B6_CTL, n=[1..7] register adds an optional 0.5 dB to the integer portion of the gain. For example, to specify 5 dB gain, program GAIN=0x5 and set HALF_DB to 0; to specify 5.5 dB gain, program GAIN=0x5 and set HALF_DB to 1; to specify -0.5 dB gain, program GAIN=0xFF and set HALF_DB to 1.

**0x152C11C4 LPASS\_CDC\_RX7\_RX\_PATH\_SEC0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_SEC0**

Bits	Name	Description
2	EN_MOD_DITHER	When set (1), random data from LFSR is passed as input to DSM. When reset (0), zero is passed as input to DSM. 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C11C8 LPASS\_CDC\_RX7\_RX\_PATH\_SEC1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000008**Reset Name:** rif\_async\_reset\_n

Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_SEC1**

Bits	Name	Type	Description
7	LFSR_RESET	RW	Soft reset for the LFSR block inside DSM. When active high, resets the LFSR generator with the default value (31'h7FFFFFFF).
3	HPF_ZERO_GATE	RW	Specifies whether to gate the HPF differentiator output to zero. When the value is 0x0, the HPF differentiator output is gated to zero. When the value is 0x1, the HPF differentiator output is allowed to pass through ungated. 0x0: GATE (-) 0x1: NO_GATE (-)
2	PGA_GAIN_UPD_STATUS	R	Specifies if the gain update has been completed. Automatically resets to IN_PROCESS when the gain is updated. When the gain has been applied, changes state to COMPLETE. Read-only bit. 0x0: IN_PROCESS (-) 0x1: COMPLETE (-)
1	PGA_MODE	RW	Specifies the mode in which gain is updated. 0x0: AUTOMATIC (-) 0x1: MANUAL (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_SEC1 (cont.)**

Bits	Name	Type	Description
0	PGA_HALF_DB	RW	

**0x152C11CC LPASS\_CDC\_RX7\_RX\_PATH\_SEC2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_SEC2**

Bits	Name	Description
7:0	PGA_TIMER	Specifies the volume controller timeout period in Fs cycles. At a 48 kHz sampling rate (Fs), the maximum timeout is 1.37s ((256*TIMER_VAL+255)/Fs). 11111111: (256*255+255)/48k = 1.365s; (256*255+255)/8k = 8.19s 10000000: (256*128+255)/48k = 0.69s; (256*128+255)/8k = 4.13s 01000000: (256*64+255)/48k = 0.347s; (256*64+255)/8k = 2.08s 00000000: Immediate gain update without timer expiration and zero crossing detection. Software control mode.

**0x152C11E4 LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000008**Reset Name:** rif\_async\_reset\_n**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC0**

Bits	Name	Type	Description
3	MIX_HPF_ZERO_GATE	RW	Specifies whether to gate the HPF differentiator output to zero. When the value is 0x0, the HPF differentiator output is gated to zero. When the value is 0x1, the HPF differentiator output is allowed to pass through ungated. 0x0: GATE (-) 0x1: NO_GATE (-)

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC0 (cont.)**

Bits	Name	Type	Description
2	MIX_PGA_GAIN_UPD_STATUS	R	Specifies if the gain update has been completed. Automatically resets to IN_PROCESS when the gain is updated. When the gain has been applied, changes state to COMPLETE. Read-only bit. 0x0: IN_PROCESS (-) 0x1: COMPLETE (-)
1	MIX_PGA_MODE	RW	Specifies the mode in which gain is updated. 0x0: AUTOMATIC (-) 0x1: MANUAL (-)
0	MIX_PGA_HALF_DB	RW	Specifies the addition of a half dB to the original gain. The integer portion of the gain is specified in the GAIN bit field of the RX_VOL_CTL register. When HALF_DB is set (1), and if the actual gain is 1, a gain of 1.5 dB is applied. When HALF_DB is set (1), and if the actual gain is -1, a gain of -0.5 dB is applied. 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C11E8 LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page B register

**LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC1**

Bits	Name	Description
7:0	MIX_PGA_TIMER	Specifies the volume controller timeout period in Fs cycles. At a 48 kHz sampling rate (Fs), the maximum timeout is 1.37s ((256*TIMER_VAL+255)/Fs). 11111111: (256*255+255)/48k = 1.365s; (256*255+255)/8k = 8.19s 10000000: (256*128+255)/48k = 0.69s; (256*128+255)/8k = 4.13s 01000000: (256*64+255)/48k = 0.347s; (256*64+255)/8k = 2.08s 00000000: Immediate gain update without timer expiration and zero crossing detection. Software control mode.

**0x152C1384 LPASS\_CDC\_RX8\_RX\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Controls Rx primary path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_CTL**

Bits	Name	Description
6	RESET	Rx data path reset 0x0: DISABLE (-) 0x1: ENABLE (-)
5	CLK_EN	Rx path clock enable 0x0: DISABLE (-) 0x1: ENABLE (-)
4	PGA_MUTE_EN	Enables Rx primary path PGA mute 0x0: DISABLE (-) 0x1: ENABLE (-)
3:0	PCM_RATE	Rx primary path input sample rate 0x0: F_8K (-) 0x1: F_16K (-) 0x3: F_32K (-) 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-) 0x7: F_384K (-) 0x8: F_44P1K_CADENCE (-)

**0x152C1388 LPASS\_CDC\_RX8\_RX\_PATH\_CFG0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_CFG0**

Bits	Name	Description
7	SPLINE_SRC_EN	Rx Path will gate inputs to and outputs from SPLINE SRC 0x0: DISABLE (-) 0x1: ENABLE (-)
3	DLY_ZN_EN	Enables DLY_Zn 0x0: DISABLE (-) 0x1: ENABLE (-)
2	HD2_EN	Enables HD2. 0x0: DISABLE (-) 0x1: ENABLE (-)
1	CMP_EN	Enables compander. 0x0: DISABLE (-) 0x1: ENABLE (-)
0	PH_EQ_EN	Enables Linear phase 0x0: MINIMUM_PHASE (-) 0x1: LINEAR_PHASE (-)

**0x152C138C LPASS\_CDC\_RX8\_RX\_PATH\_CFG1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000044**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_CFG1**

Bits	Name	Description
6	HPF_PBGC_EN	Enables HPF Passband Gain Cancellation Feature 0x0: DISABLE (-) 0x1: ENABLE (-)
5	RX_LPF_MX_SEL	Choose between including LPF or NOT 0x0: WO_LPF (-) 0x1: WITH_LPF (-)
3	SPKR_RATE	specifies spkr path rate 4.8MHz/6.144MHz or 2.4Mhz/3.072MHz 0x0: FS_4P8_6P144 (-) 0x1: FS_2P4_3P072 (-)
2	HPF_EN	Enables Rx primary path HPF 0x0: DISABLE (-) 0x1: ENABLE (-)
1	VBAT_EN	Data path gating for Vbatt. 0x0: DISABLE (-) 0x1: ENABLE (-)
0	SMART_BST_EN	Data path gating for SmartBoost 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C1390 LPASS\_CDC\_RX8\_RX\_PATH\_CFG2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x0000008F**Reset Name:** rif\_async\_reset\_n

Configures Rx path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_CFG2**

Bits	Name	Description
7:5	ECHO_REF_PCM_RATE	Echo Reference Sample Rate 0x0: F_8K (-) 0x1: F_16K (-) 0x3: F_32K (-) 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-)
4	RX_US_MX_SEL	Choose between the path before or after mixing to downsample. 0x0: POST_MIXING (-) 0x1: PRE_MIXING (-)
3:2	HPF_CUT_OFF_FREQ_384	Specifies the cutoff frequency for the DC blocker inside the Rx chain for 384KHZ Sample Rate only. 0x0: CF_NEG_3DB_8HZ (-) 0x1: CF_NEG_3DB_150HZ (-) 0x2: CF_NEG_3DB_300HZ (-) 0x3: CF_NEG_3DB_0P96HZ (-)
1:0	HPF_CUT_OFF_FREQ	Specifies the cutoff frequency for the DC blocker inside the Rx chain. 0x0: CF_NEG_3DB_4HZ (-) 0x1: CF_NEG_3DB_75HZ (-) 0x2: CF_NEG_3DB_150HZ (-) 0x3: CF_NEG_3DB_0P48HZ (-)

**0x152C1394 LPASS\_CDC\_RX8\_RX\_VOL\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Rx Main Path Vol Cntl Gain Register. Page B register.

**LPASS\_CDC\_RX8\_RX\_VOL\_CTL**

Bits	Name	Description
7:0	PGA_GAIN	<p>Specifies the volume controller gain. The value is a two complement number. The supported gain range is from -84 dB to 40 dB (in increments of 0.5 dB). This GAIN field specifies the integer portion of the gain. For example:</p> <p>0x02 = +2 dB      0x01 = +1 dB      0x00 = 0 dB      0xFF = -1 dB      0xFE = -2 dB</p> <p>The HALF_DB bit field from the CDC_RXn_B6_CTL, n=[1..7] register adds an optional 0.5 dB to the integer portion of the gain. For example, to specify 5 dB gain, program GAIN=0x5 and set HALF_DB to 0; to specify 5.5 dB gain, program GAIN=0x5 and set HALF_DB to 1; to specify -0.5 dB gain, program GAIN=0xFF and set HALF_DB to 1.</p>

**0x152C1398 LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Controls Rx mixing path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CTL**

Bits	Name	Description
5	MIX_CLK_EN	This clk_en will gate off all the clocks, fs for the mixing path. The input clk will still be the main Rx clk. When enabled, CLK_EN (main path) should also be enabled.
4	MIX_PGA_MUTE_EN	Enables Rx mixing path PGA mute 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CTL (cont.)**

Bits	Name	Description
3:0	MIX_PCM_RATE	Rx mixing path input sample rate 0x4: F_48K (-) 0x5: F_96K (-) 0x6: F_192K (-)

**0x152C139C LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CFG****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000001E**Reset Name:** rif\_async\_reset\_n

Configures Mixing path. Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CFG**

Bits	Name	Description
4	MIX_HPF_PBGC_EN	Enables mixing path HPF Passband Gain Cancellation Feature 0x0: DISABLE (-) 0x1: ENABLE (-)
3:2	MIX_HPF_CUT_OFF_FREQ	Specifies the cutoff frequency for the DC blocker inside the Rx chain. 0x0: CF_NEG_3DB_4HZ (-) 0x1: CF_NEG_3DB_75HZ (-) 0x2: CF_NEG_3DB_150HZ (-) 0x3: CF_NEG_3DB_0P48HZ (-)
1	MIX_HPF_EN	Enables Rx mixing path HPF 0x0: DISABLE (-) 0x1: ENABLE (-)

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CFG (cont.)**

Bits	Name	Description
0	MIX_PH_EQ_EN	Enables Linear phase 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C13A0 LPASS\_CDC\_RX8\_RX\_VOL\_MIX\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Enter register level description here. Page B register.

**LPASS\_CDC\_RX8\_RX\_VOL\_MIX\_CTL**

Bits	Name	Description
7:0	MIX_PGA_GAIN	Specifies the volume controller gain. The value is a two complement number. The supported gain range is from -84 dB to 40 dB (in increments of 0.5 dB). This GAIN field specifies the integer portion of the gain. For example: 0x02 = +2 dB 0x01 = +1 dB 0x00 = 0 dB 0xFF = -1 dB 0xFE = -2 dB  The HALF_DB bit field from the CDC_RXn_B6_CTL, n=[1..7] register adds an optional 0.5 dB to the integer portion of the gain. For example, to specify 5 dB gain, program GAIN=0x5 and set HALF_DB to 0; to specify 5.5 dB gain, program GAIN=0x5 and set HALF_DB to 1; to specify -0.5 dB gain, program GAIN=0xFF and set HALF_DB to 1.

**0x152C13A4 LPASS\_CDC\_RX8\_RX\_PATH\_SEC0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000004**Reset Name:** rif\_async\_reset\_n

Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_SEC0**

Bits	Name	Description
2	EN_MOD_DITHER	When set (1), random data from LFSR is passed as input to DSM. When reset (0), zero is passed as input to DSM. 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C13A8 LPASS\_CDC\_RX8\_RX\_PATH\_SEC1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000008**Reset Name:** rif\_async\_reset\_n

Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_SEC1**

Bits	Name	Type	Description
7	LFSR_RESET	RW	Soft reset for the LFSR block inside DSM. When active high, resets the LFSR generator with the default value (31'h7FFFFFFF).
3	HPF_ZERO_GATE	RW	Specifies whether to gate the HPF differentiator output to zero. When the value is 0x0, the HPF differentiator output is gated to zero. When the value is 0x1, the HPF differentiator output is allowed to pass through ungated. 0x0: GATE (-) 0x1: NO_GATE (-)
2	PGA_GAIN_UPD_STATUS	R	Specifies if the gain update has been completed. Automatically resets to IN_PROCESS when the gain is updated. When the gain has been applied, changes state to COMPLETE. Read-only bit. 0x0: IN_PROCESS (-) 0x1: COMPLETE (-)
1	PGA_MODE	RW	Specifies the mode in which gain is updated. 0x0: AUTOMATIC (-) 0x1: MANUAL (-)

**LPASS\_CDC\_RX8\_RX\_PATH\_SEC1 (cont.)**

Bits	Name	Type	Description
0	PGA_HALF_DB	RW	<p>Specifies the addition of a half dB to the original gain. The integer portion of the gain is specified in the GAIN bit field of the RX_VOL_CTL register. When HALF_DB is set (1), and if the actual gain is 1, a gain of 1.5 dB is applied. When HALF_DB is set (1), and if the actual gain is -1, a gain of -0.5 dB is applied.</p> <p>0x0: DISABLE (-) 0x1: ENABLE (-)</p>

**0x152C13AC LPASS\_CDC\_RX8\_RX\_PATH\_SEC2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_SEC2**

Bits	Name	Description
7:0	PGA_TIMER	<p>Specifies the volume controller timeout period in Fs cycles. At a 48 kHz sampling rate (Fs), the maximum timeout is 1.37s ((256*TIMER_VAL+255)/Fs).</p> <p>11111111: (256*255+255)/48k = 1.365s; (256*255+255)/8k = 8.19s      10000000: (256*128+255)/48k = 0.69s; (256*128+255)/8k = 4.13s      01000000: (256*64+255)/48k = 0.347s; (256*64+255)/8k = 2.08s      00000000: Immediate gain update without timer expiration and zero crossing detection. Software control mode.</p>

**0x152C13C4 LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC0****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC0**

Bits	Name	Type	Description
3	MIX_HPF_ZERO_GATE	RW	Specifies whether to gate the HPF differentiator output to zero. When the value is 0x0, the HPF differentiator output is gated to zero. When the value is 0x1, the HPF differentiator output is allowed to pass through ungated. 0x0: GATE (-) 0x1: NO_GATE (-)
2	MIX_PGA_GAIN_UPD_STATUS	R	Specifies if the gain update has been completed. Automatically resets to IN_PROCESS when the gain is updated. When the gain has been applied, changes state to COMPLETE. Read-only bit. 0x0: IN_PROCESS (-) 0x1: COMPLETE (-)
1	MIX_PGA_MODE	RW	Specifies the mode in which gain is updated. 0x0: AUTOMATIC (-) 0x1: MANUAL (-)
0	MIX_PGA_HALF_DB	RW	Specifies the addition of a half dB to the original gain. The integer portion of the gain is specified in the GAIN bit field of the RX_VOL_CTL register. When HALF_DB is set (1), and if the actual gain is 1, a gain of 1.5 dB is applied. When HALF_DB is set (1), and if the actual gain is -1, a gain of -0.5 dB is applied. 0x0: DISABLE (-) 0x1: ENABLE (-)

**0x152C13C8 LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page B register.

**LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC1**

Bits	Name	Description
7:0	MIX_PGA_TIMER	Specifies the volume controller timeout period in Fs cycles. At a 48 kHz sampling rate (Fs), the maximum timeout is 1.37s ((256*TIMER_VAL+255)/Fs). 11111111: (256*255+255)/48k = 1.365s; (256*255+255)/8k = 8.19s 10000000: (256*128+255)/48k = 0.69s; (256*128+255)/8k = 4.13s 01000000: (256*64+255)/48k = 0.347s; (256*64+255)/8k = 2.08s 00000000: Immediate gain update without timer expiration and zero crossing detection. Software control mode.

**0x152C1064 LPASS\_CDC\_BOOST0\_BOOST\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST0\_BOOST\_PATH\_CTL**

Bits	Name	Description
5	RESET	
4	CLK_EN	

**0x152C1068 LPASS\_CDC\_BOOST0\_BOOST\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x000000B2**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST0\_BOOST\_CTL**

Bits	Name	Description
7	BST_TRGR_EN	The boost trigger enable bit enables or disables the block from triggering on signal levels above the thresholds. Disable the block if you do not want it to react to the audio level, or if you want the output to always be zero 0x0: DISABLE (-) 0x1: ENABLE (-)
6:4	BST_SPKR_GAIN	Set this to match the speaker amp gain being used by the external boosted speaker amp. This is necessary to get accurate thresholds 0x0: B_SUB9DB (-) 0x1: B_9DB (-) 0x2: B_10P5DB (-) 0x3: B_12DB (-) 0x4: B_13P5DB (-) 0x5: B_15DB (-) 0x6: B_16P5DB (-) 0x7: B_18DB (-)
3:2	BST_STATE_MAX	These control bits can impose a max state of the cdc_boost output. When set the output will never go above the given state. 0x0: NO_MAX_STATE (-) 0x1: MAX_STATE_1 (-) 0x2: MAX_STATE_2 (-)
1:0	BST_HLD_TIME	Controls the auto-boost hold time. 0x0: B_25MS (-) 0x1: B_75MS (-) 0x2: B_125MS (-) 0x3: B_500MS (-)

**0x152C106C LPASS\_CDC\_BOOST0\_BOOST\_CFG1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST0\_BOOST\_CFG1**

Bits	Name	Description
5:3	BST_BVTH1_OFFSET	Allows some adjustment(offset) of the thresholds. This one is for the first (lowest) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively. 0x0: B_0MV (-) 0x1: B_200MV (-) 0x2: B_400MV (-) 0x3: B_600MV (-) 0x4: B_NEG_0MV (-) 0x5: B_NEG_200MV (-) 0x6: B_NEG_400MV (-) 0x7: B_NEG_600MV (-)

**LPASS\_CDC\_BOOST0\_BOOST\_CFG1 (cont.)**

<b>Bits</b>	<b>Name</b>	<b>Description</b>
2:0	BST_BVTH2_OFFSET	<p>Allows some adjustment(offset) of the thresholds. This one is for the second (middle) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively.</p> <p>0x0: B_0MV            (-)            0x1: B_200MV            (-)            0x2: B_400MV            (-)            0x3: B_600MV            (-)            0x4: B_NEG_0MV            (-)            0x5: B_NEG_200MV            (-)            0x6: B_NEG_400MV            (-)            0x7: B_NEG_600MV            (-)</p>

**0x152C1070 LPASS\_CDC\_BOOST0\_BOOST\_CFG2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST0\_BOOST\_CFG2**

Bits	Name	Description
4:2	BST_BVTH3_OFFSET	Allows some adjustment(offset) of the thresholds. This one is for the third (highest) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively. 0x0: B_0MV (-) 0x1: B_200MV (-) 0x2: B_400MV (-) 0x3: B_600MV (-) 0x4: B_NEG_0MV (-) 0x5: B_NEG_200MV (-) 0x6: B_NEG_400MV (-) 0x7: B_NEG_600MV (-)
1:0	BST_MODE	Control bits that can over-ride the auto-boost control function and force it into particular states. If you want to force it to 00 always, disable the block. 0x0: ACTIVE_MODE (-) 0x1: FORCE_CDC_BOOST_STATE1 (-) 0x2: FORCE_CDC_BOOST_STATE2 (-) 0x3: FORCE_CDC_BOOST_STATE3 (-)

**0x152C1084 LPASS\_CDC\_BOOST1\_BOOST\_PATH\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST1\_BOOST\_PATH\_CTL**

Bits	Name	Description
5	RESET	
4	CLK_EN	

**0x152C1088 LPASS\_CDC\_BOOST1\_BOOST\_CTL****Type:** RW**Clock:** rif\_clk**Reset State:** 0x000000B2**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST1\_BOOST\_CTL**

Bits	Name	Description
7	BST_TRGR_EN	The boost trigger enable bit enables or disables the block from triggering on signal levels above the thresholds. Disable the block if you do not want it to react to the audio level, or if you want the output to always be zero 0x0: DISABLE (-) 0x1: ENABLE (-)
6:4	BST_SPKR_GAIN	Set this to match the speaker amp gain being used by the external boosted speaker amp. This is necessary to get accurate thresholds 0x0: B_SUB9DB (-) 0x1: B_9DB (-) 0x2: B_10P5DB (-) 0x3: B_12DB (-) 0x4: B_13P5DB (-) 0x5: B_15DB (-) 0x6: B_16P5DB (-) 0x7: B_18DB (-)

**LPASS\_CDC\_BOOST1\_BOOST\_CTL (cont.)**

Bits	Name	Description
3:2	BST_STATE_MAX	These control bits can impose a max state of the cdc_boost output. When set the output will never go above the given state. 0x0: NO_MAX_STATE (-) 0x1: MAX_STATE_1 (-) 0x2: MAX_STATE_2 (-)
1:0	BST_HLD_TIME	Controls the auto-boost hold time. 0x0: B_25MS (-) 0x1: B_75MS (-) 0x2: B_125MS (-) 0x3: B_500MS (-)

**0x152C108C LPASS\_CDC\_BOOST1\_BOOST\_CFG1****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST1\_BOOST\_CFG1**

Bits	Name	Description
5:3	BST_BVTH1_OFFSET	<p>Allows some adjustment(offset) of the thresholds. This one is for the first (lowest) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively.</p> <p>0x0: B_0MV            (-)            0x1: B_200MV            (-)            0x2: B_400MV            (-)            0x3: B_600MV            (-)            0x4: B_NEG_0MV            (-)            0x5: B_NEG_200MV            (-)            0x6: B_NEG_400MV            (-)            0x7: B_NEG_600MV            (-)</p>
2:0	BST_BVTH2_OFFSET	<p>Allows some adjustment(offset) of the thresholds. This one is for the second (middle) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively.</p> <p>0x0: B_0MV            (-)            0x1: B_200MV            (-)            0x2: B_400MV            (-)            0x3: B_600MV            (-)            0x4: B_NEG_0MV            (-)            0x5: B_NEG_200MV            (-)            0x6: B_NEG_400MV            (-)            0x7: B_NEG_600MV            (-)</p>

**0x152C1090 LPASS\_CDC\_BOOST1\_BOOST\_CFG2****Type:** RW**Clock:** rif\_clk**Reset State:** 0x00000000**Reset Name:** rif\_async\_reset\_n

Page C register

**LPASS\_CDC\_BOOST1\_BOOST\_CFG2**

Bits	Name	Description
4:2	BST_BVTH3_OFFSET	<p>Allows some adjustment(offset) of the thresholds. This one is for the third (highest) of the three thresholds. The MSB is the sign bit, and the 2LSBS are the magnitude where you can select 200mV, 400mV and 600mV respectively.</p> <p>0x0: B_0MV            (-)            0x1: B_200MV            (-)            0x2: B_400MV            (-)            0x3: B_600MV            (-)            0x4: B_NEG_0MV            (-)            0x5: B_NEG_200MV            (-)            0x6: B_NEG_400MV            (-)            0x7: B_NEG_600MV            (-)</p>
1:0	BST_MODE	<p>Control bits that can over-ride the auto-boost control function and force it into particular states. If you want to force it to 00 always, disable the block.</p> <p>0x0: ACTIVE_MODE            (-)            0x1: FORCE_CDC_BOOST_STATE1            (-)            0x2: FORCE_CDC_BOOST_STATE2            (-)            0x3: FORCE_CDC_BOOST_STATE3            (-)</p>

# Index of registers

ANTI\_ROLLBACK\_1\_0, 31  
ANTI\_ROLLBACK\_1\_1, 31  
ANTI\_ROLLBACK\_2\_0, 32  
ANTI\_ROLLBACK\_2\_1, 32  
ANTI\_ROLLBACK\_3\_0, 33  
ANTI\_ROLLBACK\_3\_1, 33  
ANTI\_ROLLBACK\_4\_0, 34  
ANTI\_ROLLBACK\_4\_1, 34  
BOOT\_CONFIG, 15  
FEC\_EAR, 9  
FEC\_ESR, 8  
GPUCC\_CXO\_CBCR, 336  
GPUCC\_FREQUENCY\_LIMITER\_IRQ\_CLEAR, 331  
GPUCC\_FREQUENCY\_LIMITER\_IRQ\_MASK, 330  
GPUCC\_FREQUENCY\_LIMITER\_IRQ\_STATUS, 330  
GPUCC\_GDS\_HW\_CTL\_DVM\_HALT\_REQ\_SW, 328  
GPUCC\_GDS\_HW\_CTL\_DVM\_NIU\_STATUS, 329  
GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_REQ\_SW, 327  
GPUCC\_GDS\_HW\_CTL\_SMMU\_HALT\_STATUS, 328  
GPUCC\_GDS\_HW\_CTRL\_IRQ\_CLEAR, 351  
GPUCC\_GDS\_HW\_CTRL\_IRQ\_MASK, 351  
GPUCC\_GDS\_HW\_CTRL\_IRQ\_STATUS, 351  
GPUCC\_GFX\_HM\_BHS\_STRENGTH\_CTRL, 347  
GPUCC\_GFX\_RAC\_BHS\_STRENGTH\_CTRL, 348  
GPUCC\_GFX\_SPTP\_BHS\_STRENGTH\_CTRL, 348  
GPUCC\_GFX3D\_AON\_DIV\_CTRL, 347  
GPUCC\_GFX3D\_CBCR, 342  
GPUCC\_GFX3D\_CFG\_RCGR, 337  
GPUCC\_GFX3D\_CMD\_RCGR, 336  
GPUCC\_GFX3D\_ZMEAS\_CONFIG, 327  
GPUCC\_GPU\_CX\_BCR, 331  
GPUCC\_GPU\_CX\_GDS\_HW\_CTRL, 335  
GPUCC\_GPU\_CX\_GDSC\_EN\_ACK\_CTL, 329  
GPUCC\_GPU\_CX\_GDSCR, 332  
GPUCC\_GPU\_GX\_BCR, 338  
GPUCC\_GPU\_GX\_GDSC\_EN\_ACK\_CTL, 330  
GPUCC\_GPU\_GX\_GDSCR, 339  
GPUCC\_GX\_DOMAIN\_MISC, 326  
GPUCC\_GX\_DOMAIN\_MISC2, 326  
GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_CLK, 349  
GPUCC\_HLOS1\_VOTE\_GPU\_SMMU\_GDS, 350  
GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_CLK, 350  
GPUCC\_HLOS2\_VOTE\_GPU\_SMMU\_GDS, 350  
GPUCC\_HYP\_VOTE\_GPU\_SMMU\_CLK, 348  
GPUCC\_HYP\_VOTE\_GPU\_SMMU\_GDS, 349  
GPUCC\_IDLE\_REQ\_FSM\_STATE, 331  
GPUCC\_RBBMTIMER\_CBCR, 346  
GPUCC\_RBBMTIMER\_CFG\_RCGR, 345  
GPUCC\_RBBMTIMER\_CMD\_RCGR, 344  
GPUCC\_RCG\_SRC\_ACTIVE\_CTL, 325  
GPUCC\_RCG\_SRC\_ACTIVE\_STATUS, 325  
HW\_KEY\_STATUS, 9  
IMAGE\_ENCRIPT\_KEY1\_0, 23  
IMAGE\_ENCRIPT\_KEY1\_1, 24  
IMAGE\_ENCRIPT\_KEY1\_2, 24  
IMAGE\_ENCRIPT\_KEY1\_3, 24  
JTAG\_ID, 22  
LPAIF\_I2S\_CTLa, a=[0..4], 1264  
LPAIF\_INT\_I2S\_CTLa, a=[0..6], 1267  
LPASS\_CDC\_BOOST0\_BOOST\_CFG1, 1298  
LPASS\_CDC\_BOOST0\_BOOST\_CFG2, 1299  
LPASS\_CDC\_BOOST0\_BOOST\_CTL, 1296  
LPASS\_CDC\_BOOST0\_BOOST\_PATH\_CTL, 1298  
LPASS\_CDC\_BOOST1\_BOOST\_CFG1, 1302  
LPASS\_CDC\_BOOST1\_BOOST\_CFG2, 1304  
LPASS\_CDC\_BOOST1\_BOOST\_CTL, 1301  
LPASS\_CDC\_BOOST1\_BOOST\_PATH\_CTL, 1300  
LPASS\_CDC\_RX7\_RX\_PATH\_CFG0, 1276  
LPASS\_CDC\_RX7\_RX\_PATH\_CFG1, 1277  
LPASS\_CDC\_RX7\_RX\_PATH\_CFG2, 1278  
LPASS\_CDC\_RX7\_RX\_PATH\_CTL, 1275  
LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CFG, 1281  
LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_CTL, 1280  
LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC0, 1284  
LPASS\_CDC\_RX7\_RX\_PATH\_MIX\_SEC1, 1285  
LPASS\_CDC\_RX7\_RX\_PATH\_SEC0, 1282  
LPASS\_CDC\_RX7\_RX\_PATH\_SEC1, 1283  
LPASS\_CDC\_RX7\_RX\_PATH\_SEC2, 1284  
LPASS\_CDC\_RX7\_RX\_VOL\_CTL, 1280  
LPASS\_CDC\_RX7\_RX\_VOL\_MIX\_CTL, 1282  
LPASS\_CDC\_RX8\_RX\_PATH\_CFG0, 1287  
LPASS\_CDC\_RX8\_RX\_PATH\_CFG1, 1287  
LPASS\_CDC\_RX8\_RX\_PATH\_CFG2, 1288  
LPASS\_CDC\_RX8\_RX\_PATH\_CTL, 1286  
LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CFG, 1291  
LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_CTL, 1290  
LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC0, 1294  
LPASS\_CDC\_RX8\_RX\_PATH\_MIX\_SEC1, 1295

LPASS\_CDC\_RX8\_RX\_PATH\_SEC0, 1292  
 LPASS\_CDC\_RX8\_RX\_PATH\_SEC1, 1293  
 LPASS\_CDC\_RX8\_RX\_PATH\_SEC2, 1294  
 LPASS\_CDC\_RX8\_RX\_VOL\_CTL, 1290  
 LPASS\_CDC\_RX8\_RX\_VOL\_MIX\_CTL, 1292  
 LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CFG0, 1272  
 LPASS\_CDC\_TX10\_SPKR\_PROT\_PATH\_CTL, 1272  
 LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CFG0, 1273  
 LPASS\_CDC\_TX11\_SPKR\_PROT\_PATH\_CTL, 1273  
 LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CFG0, 1274  
 LPASS\_CDC\_TX12\_SPKR\_PROT\_PATH\_CTL, 1274  
 LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CFG0, 1271  
 LPASS\_CDC\_TX9\_SPKR\_PROT\_PATH\_CTL, 1271  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_3P\_PKT\_HDR, 1176  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_FTR, 1175  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_LONG\_PKT\_HDR,  
 1175  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_MMAPPED\_LONG\_P-  
 KT\_HDR, 1174  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_SHORT\_PKT, 1174  
 MMSS\_A\_CSID\_0\_CSID\_CAPTURED\_UN-  
 MAPPED\_LONG\_PKT\_HDR, 1174  
 MMSS\_A\_CSID\_0\_CSID\_IRQ\_CLEAR\_CMD, 1168  
 MMSS\_A\_CSID\_0\_CSID\_IRQ\_MASK, 1166  
 MMSS\_A\_CSID\_0\_CSID\_IRQ\_STATUS, 1170  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_3P\_PKT\_HDR, 1186  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_FTR, 1185  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_LONG\_PKT\_HDR,  
 1185  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_MMAPPED\_LONG\_P-  
 KT\_HDR, 1184  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_SHORT\_PKT, 1184  
 MMSS\_A\_CSID\_1\_CSID\_CAPTURED\_UN-  
 MAPPED\_LONG\_PKT\_HDR, 1184  
 MMSS\_A\_CSID\_1\_CSID\_IRQ\_CLEAR\_CMD, 1178  
 MMSS\_A\_CSID\_1\_CSID\_IRQ\_MASK, 1176  
 MMSS\_A\_CSID\_1\_CSID\_IRQ\_STATUS, 1180  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_3P\_PKT\_HDR, 1196  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_FTR, 1195  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_LONG\_PKT\_HDR,  
 1195  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_MMAPPED\_LONG\_P-  
 KT\_HDR, 1194  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_SHORT\_PKT, 1194  
 MMSS\_A\_CSID\_2\_CSID\_CAPTURED\_UN-  
 MAPPED\_LONG\_PKT\_HDR, 1194  
 MMSS\_A\_CSID\_2\_CSID\_IRQ\_CLEAR\_CMD, 1188  
 MMSS\_A\_CSID\_2\_CSID\_IRQ\_MASK, 1186  
 MMSS\_A\_CSID\_2\_CSID\_IRQ\_STATUS, 1190  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_3P\_PKT\_HDR, 1206  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_FTR, 1205  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_LONG\_PKT\_HDR,  
 1205  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_MMAPPED\_LONG\_P-  
 KT\_HDR, 1204  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_SHORT\_PKT, 1204  
 MMSS\_A\_CSID\_3\_CSID\_CAPTURED\_UN-  
 MAPPED\_LONG\_PKT\_HDR, 1204  
 MMSS\_A\_CSID\_3\_CSID\_IRQ\_CLEAR\_CMD, 1198  
 MMSS\_A\_CSID\_3\_CSID\_IRQ\_MASK, 1196  
 MMSS\_A\_CSID\_3\_CSID\_IRQ\_STATUS, 1200  
 MMSS\_A\_ISPIF\_IRQ\_GLOBAL\_CLEAR\_CMD, 1212  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_0, m=[0..1], 1209  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_1, m=[0..1], 1210  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_CLEAR\_2, m=[0..1], 1211  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_0, m=[0..1], 1206  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_1, m=[0..1], 1207  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_MASK\_2, m=[0..1], 1208  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_0, m=[0..1], 1213  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_1, m=[0..1], 1214  
 MMSS\_A\_ISPIF\_VFE\_m\_IRQ\_STATUS\_2, m=[0..1], 1215  
 MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_0, 1218  
 MMSS\_A\_VFE\_0\_IRQ\_CLEAR\_1, 1219  
 MMSS\_A\_VFE\_0\_IRQ\_MASK\_0, 1216  
 MMSS\_A\_VFE\_0\_IRQ\_MASK\_1, 1217  
 MMSS\_A\_VFE\_0\_IRQ\_STATUS\_0, 1220  
 MMSS\_A\_VFE\_0\_IRQ\_STATUS\_1, 1221  
 MMSS\_A\_VFE\_0\_VIOLATION\_STATUS, 1222  
 MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_0, 1226  
 MMSS\_A\_VFE\_1\_IRQ\_CLEAR\_1, 1227  
 MMSS\_A\_VFE\_1\_IRQ\_MASK\_0, 1224  
 MMSS\_A\_VFE\_1\_IRQ\_MASK\_1, 1225  
 MMSS\_A\_VFE\_1\_IRQ\_STATUS\_0, 1228  
 MMSS\_A\_VFE\_1\_IRQ\_STATUS\_1, 1229  
 MMSS\_A\_VFE\_1\_VIOLATION\_STATUS, 1230  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11,  
 1127  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12,  
 1127  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13,  
 1128  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14,  
 1128  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15,  
 1128  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16,  
 1129  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17,  
 1129  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18,  
 1129  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19,  
 1130  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20,  
 1130  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21,  
 1130  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22,  
 1131  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23,  
 1131  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24,  
 1131  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25,  
 1132  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26,  
 1132  
 MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27,

1132  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28,**  
 1133  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29,**  
 1133  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30,**  
 1133  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31,**  
 1134  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32,**  
 1134  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS0,** 1134  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS1,** 1135  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS10,** 1139  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS2,** 1135  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS3,** 1136  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS4,** 1136  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TU5,** 1137  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS6,** 1137  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS7,** 1138  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS8,** 1138  
**MMSS\_CSI\_PHY\_0\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS9,** 1139  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11,**  
 1140  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12,**  
 1140  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13,**  
 1141  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14,**  
 1141  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15,**  
 1141  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16,**  
 1142  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17,**  
 1142  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18,**  
 1142  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19,**  
 1143  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20,**  
 1143  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21,**  
 1143  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22,**  
 1144  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23,**  
 1144  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24,**  
 1144  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25,**  
 1145  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26,**  
 1145  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27,**  
 1145  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28,**  
 1146  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29,**  
 1146  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30,**  
 1146  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31,**  
 1147  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32,**  
 1147  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS0,** 1147  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS1,** 1148  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS10,** 1152  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS2,** 1148  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS3,** 1149  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS4,** 1149  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS5,** 1150  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS6,** 1150  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS7,** 1151  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS8,** 1151  
**MMSS\_CSI\_PHY\_1\_CSIPHY\_CMN\_CSI\_COMMON\_STA-**  
**TUS9,** 1152  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL11,**  
 1153  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL12,**  
 1153  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL13,**  
 1154  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL14,**  
 1154  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL15,**  
 1154  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL16,**  
 1155  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL17,**  
 1155  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL18,**  
 1155  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL19,**  
 1156  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL20,**

1156  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL21**,  
 1156  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL22**,  
 1157  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL23**,  
 1157  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL24**,  
 1157  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL25**,  
 1158  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL26**,  
 1158  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL27**,  
 1158  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL28**,  
 1159  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL29**,  
 1159  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL30**,  
 1159  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL31**,  
 1160  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_CTRL32**,  
 1160  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS0**, 1160  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS1**, 1161  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS10**, 1165  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS2**, 1161  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS3**, 1162  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS4**, 1162  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS5**, 1163  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS6**, 1163  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS7**, 1164  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS8**, 1164  
**MMSS\_CSI\_PHY\_2\_CSIPHY\_CMN\_CSI\_COMMON\_STA-TUS9**, 1165  
**MMSS\_DP\_ACTIVE\_HCTL**, 429  
**MMSS\_DP\_ACTIVE\_HOR\_VER**, 381  
**MMSS\_DP\_ACTIVE\_V\_END\_F0**, 428  
**MMSS\_DP\_ACTIVE\_V\_END\_F1**, 428  
**MMSS\_DP\_ACTIVE\_V\_START\_F0**, 427  
**MMSS\_DP\_ACTIVE\_V\_START\_F1**, 428  
**MMSS\_DP\_AUDIO\_ACR\_CTRL**, 397  
**MMSS\_DP\_AUDIO\_CFG**, 396  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_0**, 404  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_1**, 404  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_2**, 404  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_3**, 405  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_4**, 405  
**MMSS\_DP\_AUDIO\_COPYMANAGEMENT\_5**, 406  
**MMSS\_DP\_AUDIO\_CTRL\_RESET**, 397  
**MMSS\_DP\_AUDIO\_INFOFRAME\_0**, 408  
**MMSS\_DP\_AUDIO\_INFOFRAME\_1**, 409  
**MMSS\_DP\_AUDIO\_INFOFRAME\_2**, 409  
**MMSS\_DP\_AUDIO\_INFOFRAME\_3**, 409  
**MMSS\_DP\_AUDIO\_INFOFRAME\_4**, 410  
**MMSS\_DP\_AUDIO\_INFOFRAME\_5**, 410  
**MMSS\_DP\_AUDIO\_INFOFRAME\_6**, 410  
**MMSS\_DP\_AUDIO\_INFOFRAME\_7**, 411  
**MMSS\_DP\_AUDIO\_INFOFRAME\_8**, 411  
**MMSS\_DP\_AUDIO\_ISRC\_0**, 406  
**MMSS\_DP\_AUDIO\_ISRC\_1**, 406  
**MMSS\_DP\_AUDIO\_ISRC\_2**, 407  
**MMSS\_DP\_AUDIO\_ISRC\_3**, 407  
**MMSS\_DP\_AUDIO\_ISRC\_4**, 407  
**MMSS\_DP\_AUDIO\_ISRC\_5**, 408  
**MMSS\_DP\_AUDIO\_PKT\_CTRL2**, 396  
**MMSS\_DP\_AUDIO\_STREAM\_0**, 399  
**MMSS\_DP\_AUDIO\_STREAM\_1**, 400  
**MMSS\_DP\_AUDIO\_TIMESTAMP\_0**, 398  
**MMSS\_DP\_AUDIO\_TIMESTAMP\_1**, 399  
**MMSS\_DP\_AUDIO\_TIMING\_GEN**, 386  
**MMSS\_DP\_AUDIO\_TIMING\_HBR\_32**, 386  
**MMSS\_DP\_AUDIO\_TIMING\_HBR\_44**, 387  
**MMSS\_DP\_AUDIO\_TIMING\_HBR\_48**, 388  
**MMSS\_DP\_AUDIO\_TIMING\_RBR\_32**, 386  
**MMSS\_DP\_AUDIO\_TIMING\_RBR\_44**, 387  
**MMSS\_DP\_AUDIO\_TIMING\_RBR\_48**, 387  
**MMSS\_DP\_AUX\_CTRL**, 373  
**MMSS\_DP\_AUX\_DATA**, 373  
**MMSS\_DP\_AUX\_LIMITS**, 375  
**MMSS\_DP\_AUX\_MISR\_CTRL**, 376  
**MMSS\_DP\_AUX\_MISR\_RX\_VALUE**, 376  
**MMSS\_DP\_AUX\_MISR\_TX\_VALUE**, 376  
**MMSS\_DP\_AUX\_STATUS**, 375  
**MMSS\_DP\_AUX\_TRANS\_CTRL**, 374  
**MMSS\_DP\_CLK\_ACTIVE**, 365  
**MMSS\_DP\_CLK\_CTRL**, 365  
**MMSS\_DP\_COMPRESSION\_MODE\_CTRL**, 395  
**MMSS\_DP\_CONFIGURATION\_CTRL**, 378  
**MMSS\_DP\_DB\_CTRL**, 385  
**MMSS\_DP\_DISPLAY\_HCTL**, 429  
**MMSS\_DP\_DISPLAY\_V\_END\_F0**, 427  
**MMSS\_DP\_DISPLAY\_V\_END\_F1**, 427  
**MMSS\_DP\_DISPLAY\_V\_START\_F0**, 426  
**MMSS\_DP\_DISPLAY\_V\_START\_F1**, 426  
**MMSS\_DP\_DP\_HPD\_CTRL**, 370  
**MMSS\_DP\_DP\_HPD\_EVENT\_TIME\_0**, 372  
**MMSS\_DP\_DP\_HPD\_EVENT\_TIME\_1**, 372  
**MMSS\_DP\_DP\_HPD\_INT\_ACK**, 371  
**MMSS\_DP\_DP\_HPD\_INT\_MASK**, 371  
**MMSS\_DP\_DP\_HPD\_INT\_STATUS**, 370  
**MMSS\_DP\_DP\_HPD\_REF\_TIMER**, 372  
**MMSS\_DP\_EXTENSION\_0**, 400  
**MMSS\_DP\_EXTENSION\_1**, 400  
**MMSS\_DP\_EXTENSION\_2**, 401  
**MMSS\_DP\_EXTENSION\_3**, 401

MMSS\_DP\_EXTENSION\_4, 401  
 MMSS\_DP\_EXTENSION\_5, 402  
 MMSS\_DP\_EXTENSION\_6, 402  
 MMSS\_DP\_EXTENSION\_7, 403  
 MMSS\_DP\_EXTENSION\_8, 403  
 MMSS\_DP\_EXTENSION\_9, 403  
 MMSS\_DP\_GENERIC0\_0, 415  
 MMSS\_DP\_GENERIC0\_1, 416  
 MMSS\_DP\_GENERIC0\_2, 416  
 MMSS\_DP\_GENERIC0\_3, 416  
 MMSS\_DP\_GENERIC0\_4, 417  
 MMSS\_DP\_GENERIC0\_5, 417  
 MMSS\_DP\_GENERIC0\_6, 418  
 MMSS\_DP\_GENERIC0\_7, 418  
 MMSS\_DP\_GENERIC0\_8, 418  
 MMSS\_DP\_GENERIC0\_9, 419  
 MMSS\_DP\_GENERIC1\_0, 419  
 MMSS\_DP\_GENERIC1\_1, 419  
 MMSS\_DP\_GENERIC1\_2, 420  
 MMSS\_DP\_GENERIC1\_3, 420  
 MMSS\_DP\_GENERIC1\_4, 421  
 MMSS\_DP\_GENERIC1\_5, 421  
 MMSS\_DP\_GENERIC1\_6, 421  
 MMSS\_DP\_GENERIC1\_7, 422  
 MMSS\_DP\_GENERIC1\_8, 422  
 MMSS\_DP\_GENERIC1\_9, 422  
 MMSS\_DP\_HBR2\_COMPLIANCE\_SCRAMBLER\_RESET, 385  
 MMSS\_DP\_HSYNC\_CTL, 424  
 MMSS\_DP\_HSYNC\_SKEW, 429  
 MMSS\_DP\_HSYNC\_VSYNC\_WIDTH\_POLARITY, 380  
 MMSS\_DP\_INTERRUPT\_STATUS\_2, 367  
 MMSS\_DP\_INTERRUPT\_STATUS\_3, 369  
 MMSS\_DP\_INTERRUPT\_STATUS, 366  
 MMSS\_DP\_INTF\_CONFIG, 423  
 MMSS\_DP\_LOGICAL2PHYSICAL\_LANE\_MAPPING, 383  
 MMSS\_DP\_MAINLINK\_CTRL, 377  
 MMSS\_DP\_MAINLINK\_LEVELS\_2, 384  
 MMSS\_DP\_MAINLINK\_LEVELS, 384  
 MMSS\_DP\_MISC1\_MISC0, 381  
 MMSS\_DP\_MISR\_CTRL, 388  
 MMSS\_DP\_MISR\_FILLER, 390  
 MMSS\_DP\_MISR\_VALUE\_LANE0, 389  
 MMSS\_DP\_MISR\_VALUE\_LANE1, 389  
 MMSS\_DP\_MISR\_VALUE\_LANE2, 389  
 MMSS\_DP\_MISR\_VALUE\_LANE3, 389  
 MMSS\_DP\_MISR\_WRONG\_FILLERS\_1, 390  
 MMSS\_DP\_MISR\_WRONG\_FILLERS\_2, 391  
 MMSS\_DP\_MISR40\_CTRL, 391  
 MMSS\_DP\_MISR40\_SIG\_LANE\_00, 391  
 MMSS\_DP\_MISR40\_SIG\_LANE\_01, 392  
 MMSS\_DP\_MISR40\_SIG\_LANE\_10, 392  
 MMSS\_DP\_MISR40\_SIG\_LANE\_11, 392  
 MMSS\_DP\_MISR40\_SIG\_LANE\_20, 393  
 MMSS\_DP\_MISR40\_SIG\_LANE\_21, 393  
 MMSS\_DP\_MISR40\_SIG\_LANE\_30, 393  
 MMSS\_DP\_MISR40\_SIG\_LANE\_31, 393  
 MMSS\_DP\_PHY\_CTRL, 364  
 MMSS\_DP\_POLARITY\_CTL, 430  
 MMSS\_DP\_RGB\_MAPPING, 430  
 MMSS\_DP\_SDPCFG, 397  
 MMSS\_DP\_SDPCFG2, 398  
 MMSS\_DP\_SOFTWARE\_MVID, 379  
 MMSS\_DP\_SOFTWARE\_NVID, 379  
 MMSS\_DP\_START\_HOR\_VER\_FROM\_SYNC, 380  
 MMSS\_DP\_STATE\_CTRL, 377  
 MMSS\_DP\_SW\_RESET, 364  
 MMSS\_DP\_TIMEOUT\_COUNT, 374  
 MMSS\_DP\_TIMING\_ENGINE\_EN, 423  
 MMSS\_DP\_TOTAL\_HOR\_VER, 380  
 MMSS\_DP\_TU, 384  
 MMSS\_DP\_VALID\_BOUNDARY\_2, 382  
 MMSS\_DP\_VALID\_BOUNDARY, 381  
 MMSS\_DP\_VSC\_DB0\_PB4, 395  
 MMSS\_DP\_VSC\_HB0\_PB0\_HB1\_PB1, 394  
 MMSS\_DP\_VSC\_HB2\_PB2\_HB3\_PB3, 394  
 MMSS\_DP\_VSCEXT\_0, 412  
 MMSS\_DP\_VSCEXT\_1, 412  
 MMSS\_DP\_VSCEXT\_2, 412  
 MMSS\_DP\_VSCEXT\_3, 413  
 MMSS\_DP\_VSCEXT\_4, 413  
 MMSS\_DP\_VSCEXT\_5, 413  
 MMSS\_DP\_VSCEXT\_6, 414  
 MMSS\_DP\_VSCEXT\_7, 414  
 MMSS\_DP\_VSCEXT\_8, 415  
 MMSS\_DP\_VSCEXT\_9, 415  
 MMSS\_DP\_VSYNC\_PERIOD\_F0, 424  
 MMSS\_DP\_VSYNC\_PERIOD\_F1, 425  
 MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F0, 425  
 MMSS\_DP\_VSYNC\_PULSE\_WIDTH\_F1, 426  
 MMSS\_MDSS\_HW\_INTR\_STATUS, 353  
 MMSS\_MDSS\_HW\_MDSS\_CLK\_CTRL, 358  
 MMSS\_MDSS\_HW\_MDSS\_CLK\_STATUS, 359  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_0, 353  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_1, 354  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_10, 356  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_11, 357  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_12, 357  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_13, 357  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_14, 357  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_15, 358  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_2, 354  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_3, 354  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_4, 355  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_5, 355  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_6, 355  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_7, 355  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_8, 356  
 MMSS\_MDSS\_HW\_MDSS\_SCRATCH\_REGISTER\_9, 356  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_0, 359  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_1, 360  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_10, 362  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_11, 362  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_12, 363  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_13, 363  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_14, 363  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_15, 364  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_2, 360  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_3, 360

MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_4, 360  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_5, 361  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_6, 361  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_7, 361  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_8, 362  
 MMSS\_MDSS\_HW\_MDSS\_SEMAPHORE\_REGISTER\_9, 362  
 MMSS\_MDSS\_HW\_MDSS\_VBIF\_UWC\_CLK\_CTRL, 358  
 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE0, 487  
 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE1, 487  
 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE2, 488  
 MMSS\_ROT\_QDSS\_QDSS\_ATB\_DATA\_ENABLE3, 488  
 MMSS\_ROT\_QDSS\_QDSS\_CLK\_CTRL, 488  
 MMSS\_ROT\_QDSS\_QDSS\_CLK\_STATUS, 489  
 MMSS\_ROT\_QDSS\_QDSS\_PULSE\_TRIGGER, 489  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_0, 480  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_1, 481  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_2, 481  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_HI\_3, 482  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_0, 480  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_1, 480  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_2, 481  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_LO\_3, 482  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_BLOCK\_STATUS,  
 486  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_FSM\_STATE, 487  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_CLEAR, 485  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_EN, 485  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INT\_STATUS, 484  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_INVALID\_CM-  
 D\_RAM\_OFFSET, 486  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_OP\_MODE, 478  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_STATUS,  
 479  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_0\_SUBMIT,  
 478  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_STATUS,  
 480  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_QUEUE\_1\_SUBMIT,  
 479  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL,  
 482  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_CTL2,  
 482  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_-  
 LOAD\_VALUE, 483  
 MMSS\_ROT\_REGDMA\_CSR\_REGDMA\_WD\_TIMER\_STA-  
 TUS\_VALUE, 484  
 MMSS\_ROT\_REGDMA\_RAM\_REGDMA\_CMD\_RAM\_i,  
 i=[0..2047], 478  
 MMSS\_ROT\_ROTTOP\_CLK\_CTRL, 431  
 MMSS\_ROT\_ROTTOP\_CLK\_STATUS, 431  
 MMSS\_ROT\_ROTTOP\_DNSC, 435  
 MMSS\_ROT\_ROTTOP\_INTR\_CLEAR, 434  
 MMSS\_ROT\_ROTTOP\_INTR\_EN, 433  
 MMSS\_ROT\_ROTTOP\_INTR\_STATUS, 434  
 MMSS\_ROT\_ROTTOP\_OP\_MODE, 435  
 MMSS\_ROT\_ROTTOP\_ROT\_NEWSROI\_PRIOR\_TO\_START,  
 432  
 MMSS\_ROT\_ROTTOP\_START\_CTRL, 434  
 MMSS\_ROT\_ROTTOP\_STATUS, 435  
 MMSS\_ROT\_ROTTOP\_SW\_RESET\_CTRL, 432  
 MMSS\_ROT\_ROTTOP\_SW\_RESET\_OVERRIDE, 433  
 MMSS\_ROT\_ROTTOP\_SW\_RESET, 432  
 MMSS\_ROT\_SSPP\_CDP\_CNTL, 459  
 MMSS\_ROT\_SSPP\_CREQ\_LUT, 448  
 MMSS\_ROT\_SSPP\_CURRENT\_SRC0\_ADDR, 451  
 MMSS\_ROT\_SSPP\_CURRENT\_SRC1\_ADDR, 451  
 MMSS\_ROT\_SSPP\_CURRENT\_SRC2\_ADDR, 452  
 MMSS\_ROT\_SSPP\_CURRENT\_SRC3\_ADDR, 452  
 MMSS\_ROT\_SSPP\_DANGER\_LUT, 447  
 MMSS\_ROT\_SSPP\_FETCH\_CONFIG, 445  
 MMSS\_ROT\_SSPP\_FILL\_LEVELS, 452  
 MMSS\_ROT\_SSPP\_MULTI\_REC\_OP\_MODE, 464  
 MMSS\_ROT\_SSPP\_OUT\_SIZE\_REC1, 463  
 MMSS\_ROT\_SSPP\_OUT\_SIZE, 437  
 MMSS\_ROT\_SSPP\_OUT\_XY\_REC1, 463  
 MMSS\_ROT\_SSPP\_OUT\_XY, 438  
 MMSS\_ROT\_SSPP\_QOS\_CTRL, 449  
 MMSS\_ROT\_SSPP\_SAFE\_LUT, 447  
 MMSS\_ROT\_SSPP\_SRC\_ADDR\_SW\_STATUS, 450  
 MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR\_REC1, 468  
 MMSS\_ROT\_SSPP\_SRC\_CONSTANT\_COLOR, 445  
 MMSS\_ROT\_SSPP\_SRC\_FORMAT\_REC1, 465  
 MMSS\_ROT\_SSPP\_SRC\_FORMAT, 441  
 MMSS\_ROT\_SSPP\_SRC\_OP\_MODE\_REC1, 466  
 MMSS\_ROT\_SSPP\_SRC\_OP\_MODE, 443  
 MMSS\_ROT\_SSPP\_SRC\_SIZE\_REC1, 464  
 MMSS\_ROT\_SSPP\_SRC\_SIZE, 436  
 MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN\_REC1, 466  
 MMSS\_ROT\_SSPP\_SRC\_UNPACK\_PATTERN, 443  
 MMSS\_ROT\_SSPP\_SRC\_XY\_REC1, 463  
 MMSS\_ROT\_SSPP\_SRC\_XY, 437  
 MMSS\_ROT\_SSPP\_SRC\_YSTRIDE0, 440  
 MMSS\_ROT\_SSPP\_SRC\_YSTRIDE1, 440  
 MMSS\_ROT\_SSPP\_SRC0\_ADDR, 438  
 MMSS\_ROT\_SSPP\_SRC1\_ADDR, 438  
 MMSS\_ROT\_SSPP\_SRC2\_ADDR, 439  
 MMSS\_ROT\_SSPP\_SRC3\_ADDR, 439  
 MMSS\_ROT\_SSPP\_STATUS, 453  
 MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C0C3, 461  
 MMSS\_ROT\_SSPP\_SW\_CROP\_H\_C1C2, 462  
 MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C0C3, 460  
 MMSS\_ROT\_SSPP\_SW\_CROP\_W\_C1C2, 461  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_LR, 454  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_REQ\_PIXELS, 455  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C0\_TB, 455  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_LR, 456  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_REQ\_PIXELS, 457  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C1C2\_TB, 456  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_LR, 457  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_REQ\_PIXELS, 458  
 MMSS\_ROT\_SSPP\_SW\_PIX\_EXT\_C3\_TB, 458  
 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_PREFILL, 462  
 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER\_REC1\_PREFILL, 462  
 MMSS\_ROT\_SSPP\_TRAFFIC\_SHAPER, 459  
 MMSS\_ROT\_SSPP\_UWC\_ERROR\_STATUS, 460  
 MMSS\_ROT\_SSPP\_UNPACK\_BLK\_COUNT, 454  
 MMSS\_ROT\_SSPP\_UNPACK\_LINE\_COUNT, 454

MMSS\_ROT\_SSPP\_VC1\_RANGE, 446  
 MMSS\_ROT\_WB\_CDP\_CNTL, 476  
 MMSS\_ROT\_WB\_DST\_ADDR\_SW\_STATUS, 475  
 MMSS\_ROT\_WB\_DST\_ALPHA\_X\_VALUE, 475  
 MMSS\_ROT\_WB\_DST\_FORMAT, 468  
 MMSS\_ROT\_WB\_DST\_OP\_MODE, 470  
 MMSS\_ROT\_WB\_DST\_PACK\_PATTERN, 470  
 MMSS\_ROT\_WB\_DST\_WRITE\_CONFIG, 474  
 MMSS\_ROT\_WB\_DST\_YSTRIDE0, 473  
 MMSS\_ROT\_WB\_DST\_YSTRIDE1, 473  
 MMSS\_ROT\_WB\_DST0\_ADDR, 471  
 MMSS\_ROT\_WB\_DST1\_ADDR, 471  
 MMSS\_ROT\_WB\_DST2\_ADDR, 472  
 MMSS\_ROT\_WB\_DST3\_ADDR, 472  
 MMSS\_ROT\_WB\_OUT\_IMG\_SIZE, 477  
 MMSS\_ROT\_WB\_OUT\_SIZE, 475  
 MMSS\_ROT\_WB\_OUT\_XY, 477  
 MMSS\_ROT\_WB\_STATUS, 476  
 MMSS\_ROT\_WB\_TRAFFIC\_SHAPER\_WR\_CLIENT, 473  
 MMSS\_ROT\_WB\_UBWC\_ERROR\_STATUS, 477  
 OEM\_CONFIG1, 12  
 OEM\_CONFIG2, 13  
 OEM\_CONFIG3, 14  
 OEM\_CONFIG4, 14  
 OEM\_CONFIG5, 15  
 OEM\_ID, 23  
 OEM\_IMAGE\_ENCR\_KEYn, n=[0..3], 23  
 OVERRIDE\_0, 18  
 OVERRIDE\_1, 18  
 OVERRIDE\_2, 19  
 OVERRIDE\_3, 20  
 OVERRIDE\_4, 21  
 PERIPH\_SS\_AHB2PHY\_BROADCAST\_ADDRESS\_SPACE\_n, n=[0..255], 620  
 PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_LOWER, 618  
 PERIPH\_SS\_AHB2PHY\_BROADCAST\_EN\_CFG\_UPPER, 619  
 PERIPH\_SS\_AHB2PHY\_TOP\_CFG, 619  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER\_R\_ADDR\_LSB, 742  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER\_R\_ADDR\_MSB, 743  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER\_R\_ADDR, 741  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER\_R\_CTRLs, 741  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER\_R\_DATA, 742  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CNFG\_BITS\_2, 738  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_CTRL, 733  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD, 735  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_CLR, 736  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_EN, 737  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn, n=[0..3], 744  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_S-  
 RCS\_MSK\_EEn, n=[0..3], 745  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSK, 746  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UN-

MASKED\_EEn, n=[0..3], 745  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UN-MASKED, 747  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_SRCS, 746  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_IRQ\_STTS, 735  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_NUM\_PIPES, 740  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..11], 764  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn, n=[0..11], 756  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_CTRLn, n=[0..11], 748  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_AD-DR LSBn, n=[0..11], 762  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_AD-DR MSBn, n=[0..11], 763  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_AD-DRn, n=[0..11], 762  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_AD-DR LSBn, n=[0..11], 760  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_AD-DR MSBn, n=[0..11], 761  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_AD-DRn, n=[0..11], 760  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n, n=[0..11], 767  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_AD-DR LSBn, n=[0..11], 758  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_AD-DR MSBn, n=[0..11], 759  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_AD-DRn, n=[0..11], 758  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..11], 763  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn, n=[0..11], 759  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn, n=[0..11], 761  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_HALTN, n=[0..11], 750  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn, n=[0..11], 752  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn, n=[0..11], 753  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn, n=[0..11], 751  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn, n=[0..11], 755  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..11], 764  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_T\_3\_LSBn, n=[0..11], 765  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_T\_3\_MSBN, n=[0..11], 766  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..11], 765  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..11], 766  
 PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n,

n=[0..11], 767  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n,  
n=[0..11], 768  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_RSTn, n=[0..11], 750  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n,  
n=[0..11], 768  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_SW\_OFSTS<sub>n</sub>,  
n=[0..11], 757  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRL<sub>n</sub>,  
n=[0..11], 754  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TIMER<sub>n</sub>, n=[0..11],  
754  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_P\_TRUST\_REG<sub>n</sub>,  
n=[0..11], 747  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn,  
n=[0..3], 746  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_REVISION, 739  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER\_CTRL, 737  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TIMER, 737  
PERIPH\_SS\_BLSP1\_BLSP\_BAM\_BAM\_TRUST\_REG, 743  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_-  
CLEAR, 862  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL,  
859  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_COM-  
MAND, 863  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG, 862  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID,  
863  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_I2C\_MASTER\_STATUS, 860  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_CONFIG, 841  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN,  
847  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS, 847  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FI-  
FO\_WORD\_CNT, 853  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc,  
c=[0..15], 853  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_IO\_MODES, 843  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_IN-  
PUT\_CNT\_CURRENT, 851  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT,  
851  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUT-  
PUT\_CNT\_CURRENT, 849  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_OUT-  
PUT\_COUNT, 848  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_-  
READ\_CNT\_CURRENT, 852  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT,  
852  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_-  
\_WRITE\_CNT\_CURRENT, 850  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT,  
850  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATION-  
AL\_MASK, 848  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OPERATIONAL, 846  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FI-

FO\_WORD\_CNT, 849  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc,  
c=[0..15], 850  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_STATE, 842  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_SW\_RESET, 845  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_QUP\_TRANSFER\_CAN-  
CEL, 845  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_CONFIG, 853  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT, 856  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN,  
855  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS, 855  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_IO\_CONTROL, 854  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_COM-  
MAND, 857  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID,  
856  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_MASTER\_STATUS, 857  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_CFG, 858  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN, 858  
PERIPH\_SS\_BLSP1\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS,  
857  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_-  
CLEAR, 884  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL,  
882  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_COM-  
MAND, 885  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG, 884  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID,  
885  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_I2C\_MASTER\_STATUS, 882  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_CONFIG, 863  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN,  
870  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS, 869  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FI-  
FO\_WORD\_CNT, 875  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_INPUT\_FIFOc,  
c=[0..15], 876  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_IO\_MODES, 866  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_IN-  
PUT\_CNT\_CURRENT, 874  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT,  
873  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUT-  
PUT\_CNT\_CURRENT, 871  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_OUT-  
PUT\_COUNT, 871  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_-  
READ\_CNT\_CURRENT, 875  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT,  
874  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_-  
\_WRITE\_CNT\_CURRENT, 873  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_MX\_WRITE\_COUNT,  
872  
PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATION-  
AL\_MASK, 870

PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OPERATIONAL, 868  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT, 872  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc, c=[0..15], 872  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_STATE, 865  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_SW\_RESET, 867  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_QUP\_TRANSFER\_CAN-CEL, 868  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_CONFIG, 876  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT, 879  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN, 878  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS, 877  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_IO\_CONTROL, 877  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND, 879  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID, 879  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_MASTER\_STATUS, 880  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_CFG, 881  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN, 880  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS, 880  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR, 907  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL, 904  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND, 908  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG, 907  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID, 908  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_I2C\_MASTER\_STATUS, 905  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_CONFIG, 886  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN, 892  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS, 892  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT, 898  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc, c=[0..15], 898  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_IO\_MODES, 888  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_IN-PUT\_CNT\_CURRENT, 896  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT, 896  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUT-PUT\_CNT\_CURRENT, 894  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_OUT-PUT\_COUNT, 893  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX-\_READ\_CNT\_CURRENT, 897  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT, 897  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_CNT\_CURRENT, 895  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT, 895

PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATION-AL\_MASK, 893  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OPERATIONAL, 891  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFO\_WORD\_CNT, 894  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc, c=[0..15], 895  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_STATE, 887  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_SW\_RESET, 890  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_QUP\_TRANSFER\_CAN-CEL, 890  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_CONFIG, 898  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT, 901  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN, 901  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS, 900  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_IO\_CONTROL, 899  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_COMMAND, 902  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID, 902  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_MASTER\_STATUS, 902  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_CFG, 904  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN, 903  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS, 903  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_CLEAR, 930  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL, 927  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND, 931  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG, 930  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID, 931  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_I2C\_MASTER\_STATUS, 928  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_CONFIG, 909  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN, 915  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS, 915  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFO\_WORD\_CNT, 921  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_INPUT\_FIFOc, c=[0..15], 921  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_IO\_MODES, 911  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_IN-PUT\_CNT\_CURRENT, 919  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT, 919  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUT-PUT\_CNT\_CURRENT, 917  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_OUT-PUT\_COUNT, 916  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX-\_READ\_CNT\_CURRENT, 920  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_READ\_COUNT, 920  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_CNT\_CURRENT, 918

PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT, 918  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATION-AL\_MASK, 916  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OPERATIONAL, 914  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFO\_WORD\_CNT, 917  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc, c=[0..15], 918  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_STATE, 910  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_SW\_RESET, 913  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_QUP\_TRANSFER\_CAN-CEL, 913  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_CONFIG, 921  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT, 924  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN, 924  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS, 923  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_IO\_CONTROL, 922  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_COMMAND, 925  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID, 925  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_MASTER\_STATUS, 925  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_CFG, 927  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN, 926  
 PERIPH\_SS\_BLSP1\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS, 926  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BADR, 776  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_BCR, 788  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL, 787  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR\_MISR - DEPRECATED, 771  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CR, 781  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR\_SR - DEPRECATED, 771  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_CSR, 779  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMEM, 775  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_DMRX, 774  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS, 789  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_GENERICS, 790  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_HCR, 773  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR\_ISR - DEPRECATED, 772  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IMR, 782  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IPR, 772  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TO\_TAL\_SNAP\_DEPRECATED, 774  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_IRDA, 786  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR\_CLR, 803  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_ISR, 783  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR\_VAL, 777  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MISR, 782  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR1, 769  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_MR2, 769  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_NO\_CHARS \_FOR\_TX, 775  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_10, 797  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_11, 797  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_12, 797  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_13, 798  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_14, 798  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_15, 798  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_16, 798  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_2, 795  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_3, 795  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_4, 795  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_5, 796  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_6, 796  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_7, 796  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_8, 796  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF\_9, 797  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RF, 794  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RFWR, 773  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX\_TO-TAL\_SNAP, 786  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RX-TRANS\_CTRL, 789  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_RXFS, 776  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SIM\_CFG, 777  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_SR, 780  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_10, 793  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_11, 793  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_12, 793  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_13, 793  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_14, 794  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_15, 794  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_16, 794  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_2, 791  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_3, 791  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_4, 791  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_5, 791  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_6, 792  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_7, 792  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_8, 792  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF\_9, 792  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TF, 790  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TFWR, 772  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_TXFS, 776  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CFG, 799  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_CMD, 800  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STA-TUS, 800  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ - CLR, 801  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IM-R\_CLR, 803  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IM-R\_SET, 802  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IM-R\_IMR, 802  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR, 801

PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR, 801  
 PERIPH\_SS\_BLSP1\_BLSP\_UART0\_UART\_DM\_WWT\_TIME-OUT, 787  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BADR, 811  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_BCR, 825  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL, 824  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR\_MISR - DEPRECATED, 806  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CR, 817  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR\_SR - DEPRECATED, 806  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_CSR, 814  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMEN, 810  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_DMRX, 809  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS, 826  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_GENERICS, 827  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_HCR, 808  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR\_ISR - DEPRECATED, 807  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IMR, 819  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IPR, 807  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TO\_TAL\_SNAP\_DEPRECATED, 809  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_IRDA, 823  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR\_CLR, 840  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_ISR, 820  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_MODE, 812  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_RESET, 812  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR\_VAL, 813  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MISR, 819  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR1, 804  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_MR2, 804  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX, 810  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_10, 834  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_11, 834  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_12, 834  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_13, 835  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_14, 835  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_15, 835  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_16, 835  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_2, 832  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_3, 832  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_4, 832  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_5, 833  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_6, 833  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_7, 833  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_8, 833  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF\_9, 834  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RF, 831  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RFWR, 808  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TO-TAL\_SNAP, 823  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL, 826  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_RXFS, 811  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SIM\_CFG, 813  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_SR, 815  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_10, 830  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_11, 830  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_12, 830  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_13, 830  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_14, 831  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_15, 831  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_16, 831  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_2, 828  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_3, 828  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_4, 828  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_5, 828  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_6, 829  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_7, 829  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_8, 829  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF\_9, 829  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TF, 827  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TFWR, 807  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_TXFS, 811  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CFG, 836  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_CMD, 837  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS, 837  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR, 838  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR, 840  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET, 839  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_IMR, 839  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR, 838  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR, 838  
 PERIPH\_SS\_BLSP1\_BLSP\_UART1\_UART\_DM\_WWT\_TIME-OUT, 824  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER-R\_ADDR\_LSB, 941  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER-R\_ADDR\_MSB, 942  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER-R\_ADDR, 940  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER-R\_CTRL, 940  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_AHB\_MASTER\_ER-R\_DATA, 941  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CNFQ\_BITS\_2, 937  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_CTRL, 932  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_DESC\_CNT\_TRSHLD, 934  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_CLR, 935  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_EN, 936

PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_EEn, n=[0..3], 943  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_MSkn, n=[0..3], 944  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED\_EEn, n=[0..3], 944  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS\_UNMASKED, 946  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_SRCS, 945  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_IRQ\_STTS, 934  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_NUM\_PIPES, 939  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_AU\_PSM\_CNTXT\_1\_n, n=[0..11], 963  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CNSMR\_SDBNDn, n=[0..11], 955  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_CTRLn, n=[0..11], 947  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR LSBn, n=[0..11], 961  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDR\_MSBn, n=[0..11], 962  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DATA\_FIFO\_ADDRn, n=[0..11], 961  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR LSBn, n=[0..11], 959  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDR\_MSBn, n=[0..11], 960  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DESC\_FIFO\_ADDRn, n=[0..11], 959  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_DF\_CNTXT\_n, n=[0..11], 966  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR LSBn, n=[0..11], 957  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDR\_MSBn, n=[0..11], 958  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_DEST\_ADDRn, n=[0..11], 957  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_GEN\_TRSHLDn, n=[0..11], 962  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_EVNT\_REGn, n=[0..11], 958  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_FIFO\_SIZESn, n=[0..11], 960  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_HALTn, n=[0..11], 949  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_CLRn, n=[0..11], 951  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_ENn, n=[0..11], 952  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_IRQ\_STTSn, n=[0..11], 950  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PRDCR\_SDBNDn, n=[0..11], 954  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..11], 963  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_LSBn, n=[0..11], 964  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_3\_MSBn, n=[0..11], 965  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..11], 965  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..11], 966  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RETR\_CNTXT\_n, n=[0..11], 967  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_RSTn, n=[0..11], 949  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SI\_CNTXT\_n, n=[0..11], 967  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_SW\_OFSTSn, n=[0..11], 956  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMER\_CTRLn, n=[0..11], 953  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TIMERn, n=[0..11], 953  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_P\_TRUST\_REGn, n=[0..11], 946  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_PIPE\_ATTR\_EEn, n=[0..3], 945  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_REVISION, 938  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER\_CTRL, 936  
 PERIPH\_SS\_BLSP2\_BLSP\_BAM\_BAM\_TIMER, 936  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_BUS\_CLEAR, 1059  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CLK\_CTL, 1057  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_COMMAND, 1060  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_CONFIG, 1059  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_LOCAL\_ID, 1060  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_I2C\_MASTER\_STATUS, 1057  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_CONFIG, 1039  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS\_EN, 1045  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_ERROR\_FLAGS, 1045  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFO\_CNT, 1050  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_INPUT\_FIFOc, c=[0..15], 1051  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_IO\_MODES, 1041  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_IN\_PUT\_CNT\_CURRENT, 1049  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_INPUT\_COUNT, 1048  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUT\_PUT\_CNT\_CURRENT, 1047  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_OUT\_PUT\_COUNT, 1046  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_CNT\_CURRENT, 1050  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_READ\_COUNT, 1049

PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_CNT\_CURRENT, 1048  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_MX\_WRITE\_COUNT, 1048  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATION\_AL\_MASK, 1046  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OPERATIONAL, 1044  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFO\_WORD\_CNT, 1047  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_OUTPUT\_FIFOc, c=[0..15], 1047  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_STATE, 1040  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_SW\_RESET, 1043  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_QUP\_TRANSFER\_CANCEL, 1043  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_CONFIG, 1051  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_DEASSERT\_WAIT, 1054  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS\_EN, 1053  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_ERROR\_FLAGS, 1052  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_IO\_CONTROL, 1052  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_COMMAND, 1054  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_LOCAL\_ID, 1054  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_MASTER\_STATUS, 1055  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_CFG, 1056  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_EN, 1055  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP0\_SPI\_SLAVE\_IRQ\_STATUS, 1055  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_BUS\_CLEAR, 1081  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CLK\_CTL, 1079  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_COMMAND, 1082  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_CONFIG, 1081  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_LOCAL\_ID, 1082  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_I2C\_MASTER\_STATUS, 1079  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_CONFIG, 1061  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS\_EN, 1067  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_ERROR\_FLAGS, 1067  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFO\_WORD\_CNT, 1072  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_INPUT\_FIFOc, c=[0..15], 1073  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_IO\_MODES, 1063  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_IN\_PUT\_CNT\_CURRENT, 1071  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_INPUT\_COUNT, 1070  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUT\_PUT\_CNT\_CURRENT, 1069

PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_OUT\_PUT\_COUNT, 1068  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_CNT\_CURRENT, 1072  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_READ\_COUNT, 1071  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_MX\_WRITE\_CNT\_CURRENT, 1070  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATION\_AL\_MASK, 1068  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OPERATIONAL, 1066  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFO\_WORD\_CNT, 1069  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_OUTPUT\_FIFOc, c=[0..15], 1069  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_STATE, 1062  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_SW\_RESET, 1065  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_QUP\_TRANSFER\_CANCEL, 1065  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_CONFIG, 1073  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_DEASSERT\_WAIT, 1076  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS\_EN, 1075  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_ERROR\_FLAGS, 1074  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_IO\_CONTROL, 1074  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_COMMAND, 1076  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_LOCAL\_ID, 1076  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_MASTER\_STATUS, 1077  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_CFG, 1078  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_EN, 1077  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP1\_SPI\_SLAVE\_IRQ\_STATUS, 1077  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_BUS\_CLEAR, 1103  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CLK\_CTL, 1101  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_COMMAND, 1104  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_CONFIG, 1103  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_LOCAL\_ID, 1104  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_I2C\_MASTER\_STATUS, 1101  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_CONFIG, 1083  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS\_EN, 1089  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_ERROR\_FLAGS, 1089  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFO\_WORD\_CNT, 1094  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_INPUT\_FIFOc, c=[0..15], 1095  
 PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_IO\_MODES, 1085

PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_IN-  
PUT\_CNT\_CURRENT, 1093  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_INPUT\_COUNT,  
1092  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUT-  
PUT\_CNT\_CURRENT, 1091  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_OUT-  
PUT\_COUNT, 1090  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX-  
\_READ\_CNT\_CURRENT, 1094  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_READ\_COUNT,  
1093  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX-  
\_WRITE\_CNT\_CURRENT, 1092  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_MX\_WRITE\_COUNT,  
1092  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATION-  
AL\_MASK, 1090  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OPERATIONAL, 1088  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FI-  
FO\_WORD\_CNT, 1091  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_OUTPUT\_FIFOc,  
c=[0..15], 1091  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_STATE, 1084  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_SW\_RESET, 1087  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_QUP\_TRANSFER\_CAN-  
CEL, 1087  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_CONFIG, 1095  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_DEASSERT\_WAIT,  
1098  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS\_EN,  
1097  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_ERROR\_FLAGS, 1096  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_IO\_CONTROL, 1096  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_COM-  
MAND, 1098  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_LOCAL\_ID,  
1098  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_MASTER\_STATUS,  
1099  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_CFG, 1100  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_EN, 1099  
PERIPH\_SS\_BLSP2\_BLSP\_QUP2\_SPI\_SLAVE\_IRQ\_STATUS,  
1099  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_BUS\_-  
CLEAR, 1125  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CLK\_CTL,  
1123  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_COMMAND,  
1126  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_CONFIG,  
1125  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_LOCAL\_ID,  
1126  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_I2C\_MASTER\_STATUS,  
1123  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_CONFIG, 1105  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS\_EN,  
1111  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_ERROR\_FLAGS, 1111  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_INPUT\_FI-  
FO\_WORD\_CNT, 1116  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_INPUT\_FIFOc,  
c=[0..15], 1117  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_IO\_MODES, 1107  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_IN-  
PUT\_CNT\_CURRENT, 1115  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_INPUT\_COUNT,  
1114  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUT-  
PUT\_CNT\_CURRENT, 1113  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_OUT-  
PUT\_COUNT, 1112  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX-  
\_READ\_CNT\_CURRENT, 1116  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_READ\_COUNT,  
1115  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX-  
\_WRITE\_CNT\_CURRENT, 1114  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_MX\_WRITE\_COUNT,  
1114  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATION-  
AL\_MASK, 1112  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OPERATIONAL, 1110  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FI-  
FO\_WORD\_CNT, 1113  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_OUTPUT\_FIFOc,  
c=[0..15], 1113  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_STATE, 1106  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_SW\_RESET, 1109  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_QUP\_TRANSFER\_CAN-  
CEL, 1109  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_CONFIG, 1117  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_DEASSERT\_WAIT,  
1120  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS\_EN,  
1119  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_ERROR\_FLAGS, 1118  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_IO\_CONTROL, 1118  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_COM-  
MAND, 1120  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_LOCAL\_ID,  
1120  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_MASTER\_STATUS,  
1121  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_CFG, 1122  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_EN, 1121  
PERIPH\_SS\_BLSP2\_BLSP\_QUP3\_SPI\_SLAVE\_IRQ\_STATUS,  
1121  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BADR, 975  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_BCR, 989  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CLK\_CTRL,  
988  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR\_MISR\_-  
DEPRECATED, 970  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CR, 982  
PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR\_SR\_-  
DEPRECATED, 970

PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_CSR, 979  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMEN, 974  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_DMRX, 973  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_FSM\_STATUS, 990  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_GENERICS, 990  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_HCR, 972  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR\_ISR - DEPRECATED, 971  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IMR, 983  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IPR, 971  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA\_RX\_TO\_TAL\_SNAP\_DEPRECATED, 973  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_IRDA, 986  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR\_CLR, 1003  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_ISR, 984  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_RESET, 976  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR\_VAL, 976  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MISR, 982  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR1, 968  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_MR2, 968  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_NO\_CHARS\_FOR\_TX, 974  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_10, 997  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_11, 998  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_12, 998  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_13, 998  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_14, 998  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_15, 999  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_16, 999  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_2, 995  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_3, 995  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_4, 996  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_5, 996  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_6, 996  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_7, 997  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_8, 997  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF\_9, 997  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RF, 995  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RFWR, 972  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TO\_TAL\_SNAP, 987  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RX\_TRANS\_CTRL, 989  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_RXFS, 975  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SIM\_CFG, 978  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_SR, 980  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_10, 993  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_11, 993  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_12, 994  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_13, 994  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_14, 994  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_15, 994  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_16, 995  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_2, 991  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_3, 991  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_4, 992  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_5, 992  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_6, 992  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_7, 992  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_8, 993  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_9, 993  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_2 - DEPRECATED, 977  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_3 - DEPRECATED, 977  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_4 - DEPRECATED, 978  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF\_RF\_DEPRECATED, 977  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TF, 991  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TFWR, 971  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_TXFS, 975  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_CMD, 999  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IO\_STATUS, 1000  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_CLR, 1001  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_CLR, 1002  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR\_SET, 1002  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_IMR, 1001  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_ISR, 1000  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_UIM\_IRQ\_MISR, 1001  
 PERIPH\_SS\_BLSP2\_BLSP\_UART0\_UART\_DM\_WWT\_TIMEOUT, 987  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BADR, 1010  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_BCR, 1024  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CLK\_CTRL, 1023  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR\_MISR\_DEPRECATED, 1006  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CR, 1017  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR\_SR\_DEPRECATED, 1005  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_CSR, 1015  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMEN, 1009  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_DMRX, 1008  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_FSM\_STATUS, 1025  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_GENERICS, 1026  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_HCR, 1008  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR\_ISR\_DEPRECATED, 1006  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IMR, 1018  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IPR, 1006  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA\_RX\_TO\_TAL\_SNAP\_DEPRECATED, 1009

PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_IRDA, 1022  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR\_CLR, 1038  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_ISR, 1019  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_MODE, 1011  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_RESET, 1012  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR\_VAL, 1012  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MISR, 1018  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR1, 1003  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_MR2, 1004  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_NO\_CHARS\_FOR\_TX, 1010  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_10, 1033  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_11, 1033  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_12, 1033  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_13, 1034  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_14, 1034  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_15, 1034  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_16, 1034  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_2, 1031  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_3, 1031  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_4, 1031  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_5, 1032  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_6, 1032  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_7, 1032  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_8, 1032  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF\_9, 1033  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RF, 1030  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RFWR, 1007  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TOTAL\_SNAP, 1022  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RX\_TRANS\_CTRL, 1025  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_RXFS, 1011  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SIM\_CFG, 1014  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_SR, 1016  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_10, 1029  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_11, 1029  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_12, 1029  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_13, 1029  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_14, 1030  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_15, 1030  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_16, 1030  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_2, 1027  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_3, 1027  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_4, 1027  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_5, 1027  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_6, 1028  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_7, 1028  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_8, 1028  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_9, 1028  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_2\_DEPRECATED, 1013  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_3\_DEPRECATED, 1013  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_4\_DEPRECATED, 1013  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF\_RF\_DEPRECATED, 1012  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TF, 1026  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TFWR, 1007  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_TXFS, 1010  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_CMD, 1035  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IO\_STATUS, 1035  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_CLR, 1036  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_CLR, 1038  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR\_SET, 1037  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_IMR, 1037  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_ISR, 1036  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_UIM\_IRQ\_MISR, 1036  
 PERIPH\_SS\_BLSP2\_BLSP\_UART1\_UART\_DM\_WWT\_TIMEOUT, 1023  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG0, 600  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG1, 600  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG2, 601  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG3, 601  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG4, 602  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG5, 602  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG6, 602  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG7, 603  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG8, 603  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_CFG9, 604  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_CLEAR, 605  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_MASK, 604  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_AUX\_INTERRUPT\_STATUS, 608  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG\_1, 598  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_CFG, 598  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_CTRL, 608  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MISR\_STATUS, 609  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_MODE, 599  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_PD\_CTL, 599  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID0, 597  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID1, 597  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID2, 597  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_REVISION\_ID3, 597  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_STATUS, 608  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS000, 609  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS001, 609  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS010, 610  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS011, 610  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS100, 610  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS101, 611  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS110, 611

PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_MISR\_STATUS111, 611  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_LANE\_CTL, 606  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE0, 606  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX0\_TX1\_PRBS\_SEED\_BYTE1, 607  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS000, 611  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS001, 612  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS010, 612  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS011, 612  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS100, 613  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS101, 613  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS110, 613  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX1\_MISR\_STATUS111, 613  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS000, 614  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS001, 614  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS010, 614  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS011, 615  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS100, 615  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS101, 615  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS110, 615  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_MISR\_STATUS111, 616  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_LANE\_CTL, 607  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE0, 607  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX2\_TX3\_PRBS\_SEED\_BYTE1, 607  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS000, 616  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS001, 616  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS010, 617  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS011, 617  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS100, 617  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS101, 617  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS110, 618  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_TX3\_MISR\_STATUS111, 618  
 PERIPH\_SS\_DP\_PHY\_DP\_PHY\_VCO\_DIV, 606  
 PERIPH\_SS\_SDC1\_SDCC\_DATA\_COUNT, 655  
 PERIPH\_SS\_SDC1\_SDCC\_FIFO\_STATUS, 659  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CAPABILITIES, 661  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_PENDING\_TASKS, 668  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CARD\_QUEUE\_STATUS, 668  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_ARGUMENT, 673  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CMD\_RESP\_INDEX, 672  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONFIGURATION, 661  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_CONTROL, 662  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_DCMD\_RESPONSE, 671  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_COALESCING, 665  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_SIGNAL\_EN, 664  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS\_EN, 664  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_INT\_STATUS, 663  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_RESP\_MODE\_ERR\_MASK, 671

PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_1, 670  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_SEND\_STATUS\_CONFIG\_2, 670  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_CLEAR, 669  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_COMPLETE\_INT, 667  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DESC\_PROC\_ERROR, 669  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_DOORBELL, 667  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TASK\_ERR\_INFO, 672  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_LSB, 666  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_TDL\_BASE\_ADDR\_MSB, 666  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VEN\_DOR\_SPECIFIC\_CFG, 674  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VEN\_DOR\_SPECIFIC\_CFG2, 675  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VEN\_DOR\_SPECIFIC\_COUNTERS, 677  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CAPABILITIES, 676  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_CMDQ\_VS\_CMD\_INDICES, 676  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_FIFO\_ALTN, n=[0..255], 660  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_ENABLE, 673  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_NONCQ\_ICE\_INT\_STATUS, 673  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_0\_2, 621  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_10\_12, 623  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_14\_16, 623  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_18\_1A, 623  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_1C\_1E, 624  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_20\_22, 624  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_24\_26, 624  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_28\_2A, 625  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_2C\_2E, 626  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_30\_32, 627  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_34\_36, 629  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_38\_3A, 631  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_3C\_3E, 632  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_4\_6, 621  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_40\_42, 633  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_44\_46, 634  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_48\_4A, 635  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_50\_52, 636  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_54\_56, 637  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_58\_5A, 637  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_5C\_5E, 638  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_60\_62, 638  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_64\_66, 639  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_68\_6A, 639  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_6C\_6E, 640  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_8\_A, 622  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_C\_E, 622  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DDR\_CONFIG, 653

PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_2, 652  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG\_3, 652  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_CONFIG, 641  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_DLL\_STATUS, 643  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_REG\_E0\_E2, 640  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX\_I\_AR\_MON, 655  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX\_I\_AW\_MON, 653  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX\_I\_B\_MON, 654  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX\_I\_R\_MON, 655  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX\_I\_W\_MON, 654  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0, 646  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1, 647  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_DR200\_CFG, 648  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2, 644  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4, 651  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL, 660  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_SIG\_EN, 657  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_STS\_EN, 656  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_STS, 656  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG, 650  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG, 650  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG, 649  
 PERIPH\_SS\_SDC1\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0, 646  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_BYPASS\_STATUS, 685  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CONTROL, 677  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_16, r=[0..31], 686  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_17, r=[0..31], 687  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_CRYPTO CFG\_r\_n, r=[0..31], n=[0..15], 685  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_FUSE\_SETTING, 678  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_CLR, 682  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_MASK, 682  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_GENERAL\_ERR\_STTS, 681  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_CLR, 680  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ER-R\_MASK, 679  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_INVALID\_CCFG\_ERR\_STTS, 679  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_CLR, r=[0..31], 688  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_LUT\_r\_KEY\_STTS, r=[0..31], 688  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_RESET, 677  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_CONTROL, 690  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_CLR, 689  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_SEC\_IRQ\_MASK, 689  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYN-DROME1, 683  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM1\_ERROR\_SYN-DROME2, 684  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYN-DROME1, 684  
 PERIPH\_SS\_SDC1\_SDCC\_ICE\_STREAM2\_ERROR\_SYN-DROME2, 685  
 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_ACK\_TIMER, 658  
 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT\_DATA\_TIMER, 658  
 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_BOOT, 658  
 PERIPH\_SS\_SDC1\_SDCC\_SDCC\_GENERICS, 659  
 PERIPH\_SS\_SDC2\_SDCC\_DATA\_COUNT, 727  
 PERIPH\_SS\_SDC2\_SDCC\_FIFO\_STATUS, 731  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_FIFO\_ALTN, n=[0..255], 732  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_0\_2, 690  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_10\_12, 692  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_14\_16, 693  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_18\_1A, 693  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_1C\_1E, 693  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_20\_22, 694  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_24\_26, 694  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_28\_2A, 695  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_2C\_2E, 696  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_30\_32, 697  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_34\_36, 699  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_38\_3A, 700  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_3C\_3E, 702  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_4\_6, 691  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_40\_42, 703  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_44\_46, 704  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_48\_4A, 705  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_50\_52, 705  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_54\_56, 707  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_58\_5A, 707  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_5C\_5E, 707  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_60\_62, 708  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_64\_66, 708  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_68\_6A, 709  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_6C\_6E, 709  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_8\_A, 691  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_C\_E, 692  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DDR\_CONFIG, 725  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_2, 723  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG\_3, 724  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_CONFIG, 711  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_DLL\_STATUS, 713  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_E0\_E2, 710  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_REG\_FC\_FE, 711  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX-I\_AR\_MON, 726  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX-

I\_AW\_MON, 725  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX-I\_B\_MON, 726  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX-I\_R\_MON, 727  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_AX-I\_W\_MON, 726  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES0, 716  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_CAPABILITIES1, 717  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_DR200\_CFG, 718  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC2, 714  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC3, 721  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_FUNC4, 722  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_ICE\_CTRL, 732  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_SIG\_EN, 729  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_STS\_EN, 728  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VEN-DOR\_SPECIFIC\_INT\_STS, 728  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_CLEAR\_REG, 720  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_MASK\_REG, 720  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VENDOR\_SPECIFIC\_PWRCTL\_STATUS\_REG, 719  
 PERIPH\_SS\_SDC2\_SDCC\_HC\_VS\_ADMA\_ERR\_ADDR0, 716  
 PERIPH\_SS\_SDC2\_SDCC\_BOOT\_ACK\_TIMER, 730  
 PERIPH\_SS\_SDC2\_SDCC\_BOOT\_DATA\_TIMER, 730  
 PERIPH\_SS\_SDC2\_SDCC\_BOOT, 729  
 PERIPH\_SS\_SDC2\_SDCC\_SDCC\_GENERICS, 731  
 PERIPH\_SS\_USB3PHY\_PCS\_ALFPS\_DEGLITCH\_VAL, 575  
 PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_CM\_DLY, 575  
 PERIPH\_SS\_USB3PHY\_PCS\_ARCVR\_DTCT\_EN\_PERIOD, 575  
 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL, 574  
 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_CTRL2, 593  
 PERIPH\_SS\_USB3PHY\_PCS\_AUTONOMOUS\_MODE\_STATUS, 590  
 PERIPH\_SS\_USB3PHY\_PCS\_BEACON\_2\_IDLE\_TIME\_L, 567  
 PERIPH\_SS\_USB3PHY\_PCS\_CMN\_ACK\_OUT\_SEL, 590  
 PERIPH\_SS\_USB3PHY\_PCS\_COM\_CLK\_SWITCH\_CTRL, 583  
 PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_CONTROL, 580  
 PERIPH\_SS\_USB3PHY\_PCS\_COM\_RESET\_STATUS, 586  
 PERIPH\_SS\_USB3PHY\_PCS\_COMMAS\_POS\_STATUS, 596  
 PERIPH\_SS\_USB3PHY\_PCS\_DEC\_ERROR\_COUNT\_STATUS, 596  
 PERIPH\_SS\_USB3PHY\_PCS\_ELECIDLE\_DLY\_SEL, 583  
 PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_CNTRL, 590

PERIPH\_SS\_USB3PHY\_PCS\_ENDPOINT\_REFCLK\_DRIVE, 559  
 PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_H, 591  
 PERIPH\_SS\_USB3PHY\_PCS\_EPCLK\_DLY\_COUNT\_VAL\_L, 591  
 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT\_CTRL, 581  
 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT0, 582  
 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT1, 582  
 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT2, 582  
 PERIPH\_SS\_USB3PHY\_PCS\_FIXED\_PAT3, 583  
 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_H\_TOL, 573  
 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNT\_VAL\_L, 572  
 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL1, 571  
 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_CNTRL2, 572  
 PERIPH\_SS\_USB3PHY\_PCS\_FLL\_MAN\_CODE, 573  
 PERIPH\_SS\_USB3PHY\_PCS\_IDAC\_CAL\_CNTRL, 589  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL1, 577  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL2, 578  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_MX\_CTRL3, 578  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL1, 576  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL2, 576  
 PERIPH\_SS\_USB3PHY\_PCS\_INSIG\_SW\_CTRL3, 577  
 PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK LSB, 592  
 PERIPH\_SS\_USB3PHY\_PCS\_L1SS\_WAKEUP\_DLY\_TIME\_AUXCLK\_MSB, 592  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_DE-T\_HIGH\_COUNT\_VAL, 569  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_PER\_TIMER\_VAL, 594  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_CLEAR, 574  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_RXTERM\_IRQ\_SOURCE\_STATUS, 584  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX\_ECSTART\_EQT-LOCK, 569  
 PERIPH\_SS\_USB3PHY\_PCS\_LFPS\_TX-END\_CNT\_P2U3\_START, 570  
 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG1, 565  
 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG2, 566  
 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG3, 566  
 PERIPH\_SS\_USB3PHY\_PCS\_LOCK\_DETECT\_CONFIG4, 594  
 PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUX-CLK\_MSB, 588  
 PERIPH\_SS\_USB3PHY\_PCS\_LP\_WAKEUP\_DLY\_TIME\_AUX-CLK, 569  
 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_PLACEHOLDER\_STA-TUS, 552  
 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_CTRL, 551  
 PERIPH\_SS\_USB3PHY\_PCS\_MISC\_TYPEC\_STATUS, 551  
 PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_ACTIONS, 588  
 PERIPH\_SS\_USB3PHY\_PCS\_OSC\_DTCT\_STATUS, 586  
 PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_MX\_CTRL1, 579  
 PERIPH\_SS\_USB3PHY\_PCS\_OUTSIG\_SW\_CTRL1, 579  
 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS, 584  
 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS2, 585  
 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS3, 585  
 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4\_CLEAR, 596  
 PERIPH\_SS\_USB3PHY\_PCS\_PCS\_STATUS4, 595

PERIPH\_SS\_USB3PHY\_PCS\_POWER\_DOWN\_CONTROL,  
 553  
 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG1, 560  
 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG2, 562  
 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG3, 563  
 PERIPH\_SS\_USB3PHY\_PCS\_POWER\_STATE\_CONFIG4, 564  
 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY0, 580  
 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_POLY1, 580  
 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED0, 581  
 PERIPH\_SS\_USB3PHY\_PCS\_PRBS\_SEED1, 581  
 PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_-  
 TIME\_AUXCLK, 568  
 PERIPH\_SS\_USB3PHY\_PCS\_PWRUP\_RESET\_DLY\_-  
 TIME\_SYSCLK, 568  
 PERIPH\_SS\_USB3PHY\_PCS\_RATE\_SLEW\_CNTRL, 560  
 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_H, 565  
 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_P1U2\_L, 564  
 PERIPH\_SS\_USB3PHY\_PCS\_RCVR\_DTCT\_DLY\_U3\_L, 565  
 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID0, 587  
 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID1, 587  
 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID2, 587  
 PERIPH\_SS\_USB3PHY\_PCS\_REVISION\_ID3, 588  
 PERIPH\_SS\_USB3PHY\_PCS\_RX\_IDLE\_DTCT\_CNTRL, 560  
 PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_DTCT\_CNTRL,  
 595  
 PERIPH\_SS\_USB3PHY\_PCS\_RX\_SIGDET\_LVL, 592  
 PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_RUN\_TIME,  
 571  
 PERIPH\_SS\_USB3PHY\_PCS\_RXEQTRAINING\_WAIT\_TIME,  
 570  
 PERIPH\_SS\_USB3PHY\_PCS\_RXTERMINATION\_DLY\_SEL,  
 593  
 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_CNTRL, 589  
 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_LOW\_2\_IDLE\_TIME,  
 567  
 PERIPH\_SS\_USB3PHY\_PCS\_SIGDET\_STARTUP\_TIM-  
 ER\_VAL, 594  
 PERIPH\_SS\_USB3PHY\_PCS\_SW\_RESET, 552  
 PERIPH\_SS\_USB3PHY\_PCS\_TSYNC\_RSYNC\_TIME, 567  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_LS, 559  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V0, 556  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V1, 556  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V2, 557  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V3, 558  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M3P5DB\_V4, 558  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_LS, 559  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V0, 555  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V1, 556  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V2, 557  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V3, 557  
 PERIPH\_SS\_USB3PHY\_PCS\_TXDEEMPH\_M6DB\_V4, 558  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_LS, 555  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V0, 553  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V1, 553  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V2, 554  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V3, 554  
 PERIPH\_SS\_USB3PHY\_PCS\_TXMGN\_V4, 554  
 PERIPH\_SS\_USB3PHY\_PCS\_TXONESZE-  
 ROS\_RUN\_LENGTH, 571

PK\_HASH0\_0, 25  
 PK\_HASH0\_1, 25  
 PK\_HASH0\_2, 25  
 PK\_HASH0\_3, 26  
 PK\_HASH0\_4, 26  
 PK\_HASH0\_5, 26  
 PK\_HASH0\_6, 27  
 PK\_HASH0\_7, 27  
 PK\_HASH1\_0, 27  
 PK\_HASH1\_1, 28  
 PK\_HASH1\_2, 28  
 PK\_HASH1\_3, 28  
 PK\_HASH1\_4, 29  
 PK\_HASH1\_5, 29  
 PK\_HASH1\_6, 29  
 PK\_HASH1\_7, 30  
 QFPROM0\_MATCH\_STATUS, 7  
 QFPROM1\_MATCH\_STATUS, 7  
 RESET\_JDR\_STATUS, 11  
 SECURE\_BOOTn, n=[1..14], 17  
 SERIAL\_NUM, 22  
 SPMI\_CFG\_REG, 491  
 SPMI\_CGC\_CTRL, 492  
 SPMI\_CMPL\_EN\_REG, 498  
 SPMI\_HW\_SW\_EVENTS\_BITWISE\_REGj, j=[0..7], 498  
 SPMI\_HW\_SW\_EVENTS\_CTRL, 498  
 SPMI\_MAPPING\_TABLE\_REGk, k=[0..510], 490  
 SPMI\_MID\_REG, 491  
 SPMI\_MWB\_ENABLE\_REG, 493  
 SPMI\_PERIPHm\_2OWNER\_TABLE\_REG, m=[0..511], 490  
 SPMI\_PIC\_ACC\_ENABLEn, n=[0..511], 499  
 SPMI\_PIC\_IRQ\_CLEARn, n=[0..511], 500  
 SPMI\_PIC\_IRQ\_STATUSn, n=[0..511], 500  
 SPMI\_PIC\_OWNERm\_ACC\_STATUSn, m=[0..7], n=[0..15], 499  
 SPMI\_PROTOCOL\_IRQ\_CLEAR, 495  
 SPMI\_PROTOCOL\_IRQ\_EN\_CLEAR, 497  
 SPMI\_PROTOCOL\_IRQ\_EN\_SET, 496  
 SPMI\_PROTOCOL\_IRQ\_ENABLE, 494  
 SPMI\_PROTOCOL\_IRQ\_STATUS, 493  
 SPMI\_SEC\_DISABLE\_REG, 492  
 SW\_ROT\_CONFIG, 30  
 SW\_ROT\_STICKY\_BIT, 30  
 TLMM\_BTFM\_SLIMBUS\_CTL, 320  
 TLMM\_CENTER\_CLK\_GATE\_EN, 237  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_CDSP, n=[0..1],  
 241  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_GSS, n=[0..1], 240  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_HMSS, n=[0..7],  
 240  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_LPA\_DSP,  
 n=[0..5], 239  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_RPM, n=[0..0], 240  
 TLMM\_CENTER\_DIR\_CONN\_INTRn\_CFG\_SENSORS,  
 n=[0..1], 239  
 TLMM\_CENTER\_IE\_CTRL\_DISABLE, 237  
 TLMM\_CENTER\_MPM\_WAKEUP\_INT\_EN\_0, 236  
 TLMM\_DDR\_RESET\_N\_HDRV\_PULL\_CTL, 321  
 TLMM\_ETM\_MODE, 318  
 TLMM\_GPIO\_CFG0, 47

TLMM\_GPIO\_CFG1, 49  
TLMM\_GPIO\_CFG10, 248  
TLMM\_GPIO\_CFG100, 170  
TLMM\_GPIO\_CFG101, 173  
TLMM\_GPIO\_CFG102, 175  
TLMM\_GPIO\_CFG103, 177  
TLMM\_GPIO\_CFG104, 179  
TLMM\_GPIO\_CFG105, 182  
TLMM\_GPIO\_CFG106, 184  
TLMM\_GPIO\_CFG107, 186  
TLMM\_GPIO\_CFG108, 188  
TLMM\_GPIO\_CFG109, 191  
TLMM\_GPIO\_CFG11, 250  
TLMM\_GPIO\_CFG110, 193  
TLMM\_GPIO\_CFG111, 195  
TLMM\_GPIO\_CFG112, 197  
TLMM\_GPIO\_CFG113, 200  
TLMM\_GPIO\_CFG12, 252  
TLMM\_GPIO\_CFG13, 254  
TLMM\_GPIO\_CFG14, 257  
TLMM\_GPIO\_CFG15, 259  
TLMM\_GPIO\_CFG16, 209  
TLMM\_GPIO\_CFG17, 211  
TLMM\_GPIO\_CFG18, 213  
TLMM\_GPIO\_CFG19, 216  
TLMM\_GPIO\_CFG2, 51  
TLMM\_GPIO\_CFG20, 62  
TLMM\_GPIO\_CFG21, 65  
TLMM\_GPIO\_CFG22, 218  
TLMM\_GPIO\_CFG23, 220  
TLMM\_GPIO\_CFG24, 261  
TLMM\_GPIO\_CFG25, 263  
TLMM\_GPIO\_CFG26, 266  
TLMM\_GPIO\_CFG27, 268  
TLMM\_GPIO\_CFG28, 222  
TLMM\_GPIO\_CFG29, 225  
TLMM\_GPIO\_CFG3, 53  
TLMM\_GPIO\_CFG30, 227  
TLMM\_GPIO\_CFG31, 229  
TLMM\_GPIO\_CFG32, 67  
TLMM\_GPIO\_CFG33, 69  
TLMM\_GPIO\_CFG34, 71  
TLMM\_GPIO\_CFG35, 74  
TLMM\_GPIO\_CFG36, 76  
TLMM\_GPIO\_CFG37, 78  
TLMM\_GPIO\_CFG38, 80  
TLMM\_GPIO\_CFG39, 83  
TLMM\_GPIO\_CFG4, 241  
TLMM\_GPIO\_CFG40, 85  
TLMM\_GPIO\_CFG41, 87  
TLMM\_GPIO\_CFG42, 89  
TLMM\_GPIO\_CFG43, 92  
TLMM\_GPIO\_CFG44, 94  
TLMM\_GPIO\_CFG45, 96  
TLMM\_GPIO\_CFG46, 98  
TLMM\_GPIO\_CFG47, 101  
TLMM\_GPIO\_CFG48, 103  
TLMM\_GPIO\_CFG49, 105  
TLMM\_GPIO\_CFG5, 56  
TLMM\_GPIO\_CFG50, 107  
TLMM\_GPIO\_CFG51, 110  
TLMM\_GPIO\_CFG52, 112  
TLMM\_GPIO\_CFG53, 270  
TLMM\_GPIO\_CFG54, 272  
TLMM\_GPIO\_CFG55, 114  
TLMM\_GPIO\_CFG56, 116  
TLMM\_GPIO\_CFG57, 119  
TLMM\_GPIO\_CFG58, 121  
TLMM\_GPIO\_CFG59, 275  
TLMM\_GPIO\_CFG6, 58  
TLMM\_GPIO\_CFG60, 277  
TLMM\_GPIO\_CFG61, 279  
TLMM\_GPIO\_CFG62, 281  
TLMM\_GPIO\_CFG63, 284  
TLMM\_GPIO\_CFG64, 123  
TLMM\_GPIO\_CFG65, 125  
TLMM\_GPIO\_CFG66, 286  
TLMM\_GPIO\_CFG67, 288  
TLMM\_GPIO\_CFG68, 290  
TLMM\_GPIO\_CFG69, 293  
TLMM\_GPIO\_CFG7, 60  
TLMM\_GPIO\_CFG70, 295  
TLMM\_GPIO\_CFG71, 297  
TLMM\_GPIO\_CFG72, 299  
TLMM\_GPIO\_CFG73, 302  
TLMM\_GPIO\_CFG74, 304  
TLMM\_GPIO\_CFG75, 306  
TLMM\_GPIO\_CFG76, 308  
TLMM\_GPIO\_CFG77, 311  
TLMM\_GPIO\_CFG78, 313  
TLMM\_GPIO\_CFG79, 128  
TLMM\_GPIO\_CFG8, 243  
TLMM\_GPIO\_CFG80, 130  
TLMM\_GPIO\_CFG81, 231  
TLMM\_GPIO\_CFG82, 234  
TLMM\_GPIO\_CFG83, 132  
TLMM\_GPIO\_CFG84, 134  
TLMM\_GPIO\_CFG85, 137  
TLMM\_GPIO\_CFG86, 139  
TLMM\_GPIO\_CFG87, 141  
TLMM\_GPIO\_CFG88, 143  
TLMM\_GPIO\_CFG89, 146  
TLMM\_GPIO\_CFG9, 245  
TLMM\_GPIO\_CFG90, 148  
TLMM\_GPIO\_CFG91, 150  
TLMM\_GPIO\_CFG92, 152  
TLMM\_GPIO\_CFG93, 155  
TLMM\_GPIO\_CFG94, 157  
TLMM\_GPIO\_CFG95, 159  
TLMM\_GPIO\_CFG96, 161  
TLMM\_GPIO\_CFG97, 164  
TLMM\_GPIO\_CFG98, 166  
TLMM\_GPIO\_CFG99, 168  
TLMM\_GPIO\_GS\_I2C\_CTL, 238  
TLMM\_GPIO\_ID\_STATUS0, 49  
TLMM\_GPIO\_ID\_STATUS1, 51  
TLMM\_GPIO\_ID\_STATUS10, 250  
TLMM\_GPIO\_ID\_STATUS100, 172

TLMM\_GPIO\_ID\_STATUS101, 175  
TLMM\_GPIO\_ID\_STATUS102, 177  
TLMM\_GPIO\_ID\_STATUS103, 179  
TLMM\_GPIO\_ID\_STATUS104, 181  
TLMM\_GPIO\_ID\_STATUS105, 184  
TLMM\_GPIO\_ID\_STATUS106, 186  
TLMM\_GPIO\_ID\_STATUS107, 188  
TLMM\_GPIO\_ID\_STATUS108, 190  
TLMM\_GPIO\_ID\_STATUS109, 193  
TLMM\_GPIO\_ID\_STATUS11, 252  
TLMM\_GPIO\_ID\_STATUS110, 195  
TLMM\_GPIO\_ID\_STATUS111, 197  
TLMM\_GPIO\_ID\_STATUS112, 199  
TLMM\_GPIO\_ID\_STATUS113, 202  
TLMM\_GPIO\_ID\_STATUS12, 254  
TLMM\_GPIO\_ID\_STATUS13, 256  
TLMM\_GPIO\_ID\_STATUS14, 259  
TLMM\_GPIO\_ID\_STATUS15, 261  
TLMM\_GPIO\_ID\_STATUS16, 211  
TLMM\_GPIO\_ID\_STATUS17, 213  
TLMM\_GPIO\_ID\_STATUS18, 215  
TLMM\_GPIO\_ID\_STATUS19, 218  
TLMM\_GPIO\_ID\_STATUS2, 53  
TLMM\_GPIO\_ID\_STATUS20, 64  
TLMM\_GPIO\_ID\_STATUS21, 67  
TLMM\_GPIO\_ID\_STATUS22, 220  
TLMM\_GPIO\_ID\_STATUS23, 222  
TLMM\_GPIO\_ID\_STATUS24, 263  
TLMM\_GPIO\_ID\_STATUS25, 265  
TLMM\_GPIO\_ID\_STATUS26, 268  
TLMM\_GPIO\_ID\_STATUS27, 270  
TLMM\_GPIO\_ID\_STATUS28, 224  
TLMM\_GPIO\_ID\_STATUS29, 227  
TLMM\_GPIO\_ID\_STATUS3, 55  
TLMM\_GPIO\_ID\_STATUS30, 229  
TLMM\_GPIO\_ID\_STATUS31, 231  
TLMM\_GPIO\_ID\_STATUS32, 69  
TLMM\_GPIO\_ID\_STATUS33, 71  
TLMM\_GPIO\_ID\_STATUS34, 73  
TLMM\_GPIO\_ID\_STATUS35, 76  
TLMM\_GPIO\_ID\_STATUS36, 78  
TLMM\_GPIO\_ID\_STATUS37, 80  
TLMM\_GPIO\_ID\_STATUS38, 82  
TLMM\_GPIO\_ID\_STATUS39, 85  
TLMM\_GPIO\_ID\_STATUS4, 243  
TLMM\_GPIO\_ID\_STATUS40, 87  
TLMM\_GPIO\_ID\_STATUS41, 89  
TLMM\_GPIO\_ID\_STATUS42, 91  
TLMM\_GPIO\_ID\_STATUS43, 94  
TLMM\_GPIO\_ID\_STATUS44, 96  
TLMM\_GPIO\_ID\_STATUS45, 98  
TLMM\_GPIO\_ID\_STATUS46, 100  
TLMM\_GPIO\_ID\_STATUS47, 103  
TLMM\_GPIO\_ID\_STATUS48, 105  
TLMM\_GPIO\_ID\_STATUS49, 107  
TLMM\_GPIO\_ID\_STATUS5, 58  
TLMM\_GPIO\_ID\_STATUS50, 109  
TLMM\_GPIO\_ID\_STATUS51, 112  
TLMM\_GPIO\_ID\_STATUS52, 114  
TLMM\_GPIO\_ID\_STATUS53, 272  
TLMM\_GPIO\_ID\_STATUS54, 274  
TLMM\_GPIO\_ID\_STATUS55, 116  
TLMM\_GPIO\_ID\_STATUS56, 118  
TLMM\_GPIO\_ID\_STATUS57, 121  
TLMM\_GPIO\_ID\_STATUS58, 123  
TLMM\_GPIO\_ID\_STATUS59, 277  
TLMM\_GPIO\_ID\_STATUS6, 60  
TLMM\_GPIO\_ID\_STATUS60, 279  
TLMM\_GPIO\_ID\_STATUS61, 281  
TLMM\_GPIO\_ID\_STATUS62, 283  
TLMM\_GPIO\_ID\_STATUS63, 286  
TLMM\_GPIO\_ID\_STATUS64, 125  
TLMM\_GPIO\_ID\_STATUS65, 127  
TLMM\_GPIO\_ID\_STATUS66, 288  
TLMM\_GPIO\_ID\_STATUS67, 290  
TLMM\_GPIO\_ID\_STATUS68, 292  
TLMM\_GPIO\_ID\_STATUS69, 295  
TLMM\_GPIO\_ID\_STATUS7, 62  
TLMM\_GPIO\_ID\_STATUS70, 297  
TLMM\_GPIO\_ID\_STATUS71, 299  
TLMM\_GPIO\_ID\_STATUS72, 301  
TLMM\_GPIO\_ID\_STATUS73, 304  
TLMM\_GPIO\_ID\_STATUS74, 306  
TLMM\_GPIO\_ID\_STATUS75, 308  
TLMM\_GPIO\_ID\_STATUS76, 310  
TLMM\_GPIO\_ID\_STATUS77, 313  
TLMM\_GPIO\_ID\_STATUS78, 315  
TLMM\_GPIO\_ID\_STATUS79, 130  
TLMM\_GPIO\_ID\_STATUS8, 245  
TLMM\_GPIO\_ID\_STATUS80, 132  
TLMM\_GPIO\_ID\_STATUS81, 233  
TLMM\_GPIO\_ID\_STATUS82, 236  
TLMM\_GPIO\_ID\_STATUS83, 134  
TLMM\_GPIO\_ID\_STATUS84, 136  
TLMM\_GPIO\_ID\_STATUS85, 139  
TLMM\_GPIO\_ID\_STATUS86, 141  
TLMM\_GPIO\_ID\_STATUS87, 143  
TLMM\_GPIO\_ID\_STATUS88, 145  
TLMM\_GPIO\_ID\_STATUS89, 148  
TLMM\_GPIO\_ID\_STATUS9, 247  
TLMM\_GPIO\_ID\_STATUS90, 150  
TLMM\_GPIO\_ID\_STATUS91, 152  
TLMM\_GPIO\_ID\_STATUS92, 154  
TLMM\_GPIO\_ID\_STATUS93, 157  
TLMM\_GPIO\_ID\_STATUS94, 159  
TLMM\_GPIO\_ID\_STATUS95, 161  
TLMM\_GPIO\_ID\_STATUS96, 163  
TLMM\_GPIO\_ID\_STATUS97, 166  
TLMM\_GPIO\_ID\_STATUS98, 168  
TLMM\_GPIO\_ID\_STATUS99, 170  
TLMM\_GPIO\_IN\_OUT0, 47  
TLMM\_GPIO\_IN\_OUT1, 49  
TLMM\_GPIO\_IN\_OUT10, 248  
TLMM\_GPIO\_IN\_OUT100, 171  
TLMM\_GPIO\_IN\_OUT101, 173  
TLMM\_GPIO\_IN\_OUT102, 175  
TLMM\_GPIO\_IN\_OUT103, 178  
TLMM\_GPIO\_IN\_OUT104, 180

TLMM\_GPIO\_IN\_OUT105, 182  
 TLMM\_GPIO\_IN\_OUT106, 184  
 TLMM\_GPIO\_IN\_OUT107, 187  
 TLMM\_GPIO\_IN\_OUT108, 189  
 TLMM\_GPIO\_IN\_OUT109, 191  
 TLMM\_GPIO\_IN\_OUT11, 250  
 TLMM\_GPIO\_IN\_OUT110, 193  
 TLMM\_GPIO\_IN\_OUT111, 196  
 TLMM\_GPIO\_IN\_OUT112, 198  
 TLMM\_GPIO\_IN\_OUT113, 200  
 TLMM\_GPIO\_IN\_OUT12, 253  
 TLMM\_GPIO\_IN\_OUT13, 255  
 TLMM\_GPIO\_IN\_OUT14, 257  
 TLMM\_GPIO\_IN\_OUT15, 259  
 TLMM\_GPIO\_IN\_OUT16, 210  
 TLMM\_GPIO\_IN\_OUT17, 212  
 TLMM\_GPIO\_IN\_OUT18, 214  
 TLMM\_GPIO\_IN\_OUT19, 216  
 TLMM\_GPIO\_IN\_OUT2, 52  
 TLMM\_GPIO\_IN\_OUT20, 63  
 TLMM\_GPIO\_IN\_OUT21, 65  
 TLMM\_GPIO\_IN\_OUT22, 218  
 TLMM\_GPIO\_IN\_OUT23, 221  
 TLMM\_GPIO\_IN\_OUT24, 262  
 TLMM\_GPIO\_IN\_OUT25, 264  
 TLMM\_GPIO\_IN\_OUT26, 266  
 TLMM\_GPIO\_IN\_OUT27, 268  
 TLMM\_GPIO\_IN\_OUT28, 223  
 TLMM\_GPIO\_IN\_OUT29, 225  
 TLMM\_GPIO\_IN\_OUT3, 54  
 TLMM\_GPIO\_IN\_OUT30, 227  
 TLMM\_GPIO\_IN\_OUT31, 230  
 TLMM\_GPIO\_IN\_OUT32, 67  
 TLMM\_GPIO\_IN\_OUT33, 70  
 TLMM\_GPIO\_IN\_OUT34, 72  
 TLMM\_GPIO\_IN\_OUT35, 74  
 TLMM\_GPIO\_IN\_OUT36, 76  
 TLMM\_GPIO\_IN\_OUT37, 79  
 TLMM\_GPIO\_IN\_OUT38, 81  
 TLMM\_GPIO\_IN\_OUT39, 83  
 TLMM\_GPIO\_IN\_OUT4, 242  
 TLMM\_GPIO\_IN\_OUT40, 85  
 TLMM\_GPIO\_IN\_OUT41, 88  
 TLMM\_GPIO\_IN\_OUT42, 90  
 TLMM\_GPIO\_IN\_OUT43, 92  
 TLMM\_GPIO\_IN\_OUT44, 94  
 TLMM\_GPIO\_IN\_OUT45, 97  
 TLMM\_GPIO\_IN\_OUT46, 99  
 TLMM\_GPIO\_IN\_OUT47, 101  
 TLMM\_GPIO\_IN\_OUT48, 103  
 TLMM\_GPIO\_IN\_OUT49, 106  
 TLMM\_GPIO\_IN\_OUT5, 56  
 TLMM\_GPIO\_IN\_OUT50, 108  
 TLMM\_GPIO\_IN\_OUT51, 110  
 TLMM\_GPIO\_IN\_OUT52, 112  
 TLMM\_GPIO\_IN\_OUT53, 271  
 TLMM\_GPIO\_IN\_OUT54, 273  
 TLMM\_GPIO\_IN\_OUT55, 115  
 TLMM\_GPIO\_IN\_OUT56, 117  
 TLMM\_GPIO\_IN\_OUT57, 119  
 TLMM\_GPIO\_IN\_OUT58, 121  
 TLMM\_GPIO\_IN\_OUT59, 275  
 TLMM\_GPIO\_IN\_OUT6, 58  
 TLMM\_GPIO\_IN\_OUT60, 277  
 TLMM\_GPIO\_IN\_OUT61, 280  
 TLMM\_GPIO\_IN\_OUT62, 282  
 TLMM\_GPIO\_IN\_OUT63, 284  
 TLMM\_GPIO\_IN\_OUT64, 124  
 TLMM\_GPIO\_IN\_OUT65, 126  
 TLMM\_GPIO\_IN\_OUT66, 286  
 TLMM\_GPIO\_IN\_OUT67, 289  
 TLMM\_GPIO\_IN\_OUT68, 291  
 TLMM\_GPIO\_IN\_OUT69, 293  
 TLMM\_GPIO\_IN\_OUT7, 61  
 TLMM\_GPIO\_IN\_OUT70, 295  
 TLMM\_GPIO\_IN\_OUT71, 298  
 TLMM\_GPIO\_IN\_OUT72, 300  
 TLMM\_GPIO\_IN\_OUT73, 302  
 TLMM\_GPIO\_IN\_OUT74, 304  
 TLMM\_GPIO\_IN\_OUT75, 307  
 TLMM\_GPIO\_IN\_OUT76, 309  
 TLMM\_GPIO\_IN\_OUT77, 311  
 TLMM\_GPIO\_IN\_OUT78, 313  
 TLMM\_GPIO\_IN\_OUT79, 128  
 TLMM\_GPIO\_IN\_OUT8, 244  
 TLMM\_GPIO\_IN\_OUT80, 130  
 TLMM\_GPIO\_IN\_OUT81, 232  
 TLMM\_GPIO\_IN\_OUT82, 234  
 TLMM\_GPIO\_IN\_OUT83, 133  
 TLMM\_GPIO\_IN\_OUT84, 135  
 TLMM\_GPIO\_IN\_OUT85, 137  
 TLMM\_GPIO\_IN\_OUT86, 139  
 TLMM\_GPIO\_IN\_OUT87, 142  
 TLMM\_GPIO\_IN\_OUT88, 144  
 TLMM\_GPIO\_IN\_OUT89, 146  
 TLMM\_GPIO\_IN\_OUT9, 246  
 TLMM\_GPIO\_IN\_OUT90, 148  
 TLMM\_GPIO\_IN\_OUT91, 151  
 TLMM\_GPIO\_IN\_OUT92, 153  
 TLMM\_GPIO\_IN\_OUT93, 155  
 TLMM\_GPIO\_IN\_OUT94, 157  
 TLMM\_GPIO\_IN\_OUT95, 160  
 TLMM\_GPIO\_IN\_OUT96, 162  
 TLMM\_GPIO\_IN\_OUT97, 164  
 TLMM\_GPIO\_IN\_OUT98, 166  
 TLMM\_GPIO\_IN\_OUT99, 169  
 TLMM\_GPIO\_INTR\_CFG0, 48  
 TLMM\_GPIO\_INTR\_CFG1, 50  
 TLMM\_GPIO\_INTR\_CFG10, 249  
 TLMM\_GPIO\_INTR\_CFG100, 171  
 TLMM\_GPIO\_INTR\_CFG101, 174  
 TLMM\_GPIO\_INTR\_CFG102, 176  
 TLMM\_GPIO\_INTR\_CFG103, 178  
 TLMM\_GPIO\_INTR\_CFG104, 180  
 TLMM\_GPIO\_INTR\_CFG105, 183  
 TLMM\_GPIO\_INTR\_CFG106, 185  
 TLMM\_GPIO\_INTR\_CFG107, 187  
 TLMM\_GPIO\_INTR\_CFG108, 189

TLMM\_GPIO\_INTR\_CFG109, 192  
 TLMM\_GPIO\_INTR\_CFG11, 251  
 TLMM\_GPIO\_INTR\_CFG110, 194  
 TLMM\_GPIO\_INTR\_CFG111, 196  
 TLMM\_GPIO\_INTR\_CFG112, 198  
 TLMM\_GPIO\_INTR\_CFG113, 201  
 TLMM\_GPIO\_INTR\_CFG12, 253  
 TLMM\_GPIO\_INTR\_CFG13, 255  
 TLMM\_GPIO\_INTR\_CFG14, 258  
 TLMM\_GPIO\_INTR\_CFG15, 260  
 TLMM\_GPIO\_INTR\_CFG16, 210  
 TLMM\_GPIO\_INTR\_CFG17, 212  
 TLMM\_GPIO\_INTR\_CFG18, 214  
 TLMM\_GPIO\_INTR\_CFG19, 217  
 TLMM\_GPIO\_INTR\_CFG2, 52  
 TLMM\_GPIO\_INTR\_CFG20, 63  
 TLMM\_GPIO\_INTR\_CFG21, 66  
 TLMM\_GPIO\_INTR\_CFG22, 219  
 TLMM\_GPIO\_INTR\_CFG23, 221  
 TLMM\_GPIO\_INTR\_CFG24, 262  
 TLMM\_GPIO\_INTR\_CFG25, 264  
 TLMM\_GPIO\_INTR\_CFG26, 267  
 TLMM\_GPIO\_INTR\_CFG27, 269  
 TLMM\_GPIO\_INTR\_CFG28, 223  
 TLMM\_GPIO\_INTR\_CFG29, 226  
 TLMM\_GPIO\_INTR\_CFG3, 54  
 TLMM\_GPIO\_INTR\_CFG30, 228  
 TLMM\_GPIO\_INTR\_CFG31, 230  
 TLMM\_GPIO\_INTR\_CFG32, 68  
 TLMM\_GPIO\_INTR\_CFG33, 70  
 TLMM\_GPIO\_INTR\_CFG34, 72  
 TLMM\_GPIO\_INTR\_CFG35, 75  
 TLMM\_GPIO\_INTR\_CFG36, 77  
 TLMM\_GPIO\_INTR\_CFG37, 79  
 TLMM\_GPIO\_INTR\_CFG38, 81  
 TLMM\_GPIO\_INTR\_CFG39, 84  
 TLMM\_GPIO\_INTR\_CFG4, 242  
 TLMM\_GPIO\_INTR\_CFG40, 86  
 TLMM\_GPIO\_INTR\_CFG41, 88  
 TLMM\_GPIO\_INTR\_CFG42, 90  
 TLMM\_GPIO\_INTR\_CFG43, 93  
 TLMM\_GPIO\_INTR\_CFG44, 95  
 TLMM\_GPIO\_INTR\_CFG45, 97  
 TLMM\_GPIO\_INTR\_CFG46, 99  
 TLMM\_GPIO\_INTR\_CFG47, 102  
 TLMM\_GPIO\_INTR\_CFG48, 104  
 TLMM\_GPIO\_INTR\_CFG49, 106  
 TLMM\_GPIO\_INTR\_CFG5, 57  
 TLMM\_GPIO\_INTR\_CFG50, 108  
 TLMM\_GPIO\_INTR\_CFG51, 111  
 TLMM\_GPIO\_INTR\_CFG52, 113  
 TLMM\_GPIO\_INTR\_CFG53, 271  
 TLMM\_GPIO\_INTR\_CFG54, 273  
 TLMM\_GPIO\_INTR\_CFG55, 115  
 TLMM\_GPIO\_INTR\_CFG56, 117  
 TLMM\_GPIO\_INTR\_CFG57, 120  
 TLMM\_GPIO\_INTR\_CFG58, 122  
 TLMM\_GPIO\_INTR\_CFG59, 276  
 TLMM\_GPIO\_INTR\_CFG6, 59  
 TLMM\_GPIO\_INTR\_CFG60, 278  
 TLMM\_GPIO\_INTR\_CFG61, 280  
 TLMM\_GPIO\_INTR\_CFG62, 282  
 TLMM\_GPIO\_INTR\_CFG63, 285  
 TLMM\_GPIO\_INTR\_CFG64, 124  
 TLMM\_GPIO\_INTR\_CFG65, 126  
 TLMM\_GPIO\_INTR\_CFG66, 287  
 TLMM\_GPIO\_INTR\_CFG67, 289  
 TLMM\_GPIO\_INTR\_CFG68, 291  
 TLMM\_GPIO\_INTR\_CFG69, 294  
 TLMM\_GPIO\_INTR\_CFG7, 61  
 TLMM\_GPIO\_INTR\_CFG70, 296  
 TLMM\_GPIO\_INTR\_CFG71, 298  
 TLMM\_GPIO\_INTR\_CFG72, 300  
 TLMM\_GPIO\_INTR\_CFG73, 303  
 TLMM\_GPIO\_INTR\_CFG74, 305  
 TLMM\_GPIO\_INTR\_CFG75, 307  
 TLMM\_GPIO\_INTR\_CFG76, 309  
 TLMM\_GPIO\_INTR\_CFG77, 312  
 TLMM\_GPIO\_INTR\_CFG78, 314  
 TLMM\_GPIO\_INTR\_CFG79, 129  
 TLMM\_GPIO\_INTR\_CFG8, 244  
 TLMM\_GPIO\_INTR\_CFG80, 131  
 TLMM\_GPIO\_INTR\_CFG81, 232  
 TLMM\_GPIO\_INTR\_CFG82, 235  
 TLMM\_GPIO\_INTR\_CFG83, 133  
 TLMM\_GPIO\_INTR\_CFG84, 135  
 TLMM\_GPIO\_INTR\_CFG85, 138  
 TLMM\_GPIO\_INTR\_CFG86, 140  
 TLMM\_GPIO\_INTR\_CFG87, 142  
 TLMM\_GPIO\_INTR\_CFG88, 144  
 TLMM\_GPIO\_INTR\_CFG89, 147  
 TLMM\_GPIO\_INTR\_CFG9, 246  
 TLMM\_GPIO\_INTR\_CFG90, 149  
 TLMM\_GPIO\_INTR\_CFG91, 151  
 TLMM\_GPIO\_INTR\_CFG92, 153  
 TLMM\_GPIO\_INTR\_CFG93, 156  
 TLMM\_GPIO\_INTR\_CFG94, 158  
 TLMM\_GPIO\_INTR\_CFG95, 160  
 TLMM\_GPIO\_INTR\_CFG96, 162  
 TLMM\_GPIO\_INTR\_CFG97, 165  
 TLMM\_GPIO\_INTR\_CFG98, 167  
 TLMM\_GPIO\_INTR\_CFG99, 169  
 TLMM\_GPIO\_INTR\_STATUS0, 48  
 TLMM\_GPIO\_INTR\_STATUS1, 51  
 TLMM\_GPIO\_INTR\_STATUS10, 249  
 TLMM\_GPIO\_INTR\_STATUS100, 172  
 TLMM\_GPIO\_INTR\_STATUS101, 174  
 TLMM\_GPIO\_INTR\_STATUS102, 177  
 TLMM\_GPIO\_INTR\_STATUS103, 179  
 TLMM\_GPIO\_INTR\_STATUS104, 181  
 TLMM\_GPIO\_INTR\_STATUS105, 183  
 TLMM\_GPIO\_INTR\_STATUS106, 186  
 TLMM\_GPIO\_INTR\_STATUS107, 188  
 TLMM\_GPIO\_INTR\_STATUS108, 190  
 TLMM\_GPIO\_INTR\_STATUS109, 192  
 TLMM\_GPIO\_INTR\_STATUS11, 252  
 TLMM\_GPIO\_INTR\_STATUS110, 195  
 TLMM\_GPIO\_INTR\_STATUS111, 197

TLMM\_GPIO\_INTR\_STATUS112, 199  
 TLMM\_GPIO\_INTR\_STATUS113, 201  
 TLMM\_GPIO\_INTR\_STATUS12, 254  
 TLMM\_GPIO\_INTR\_STATUS13, 256  
 TLMM\_GPIO\_INTR\_STATUS14, 258  
 TLMM\_GPIO\_INTR\_STATUS15, 261  
 TLMM\_GPIO\_INTR\_STATUS16, 211  
 TLMM\_GPIO\_INTR\_STATUS17, 213  
 TLMM\_GPIO\_INTR\_STATUS18, 215  
 TLMM\_GPIO\_INTR\_STATUS19, 217  
 TLMM\_GPIO\_INTR\_STATUS2, 53  
 TLMM\_GPIO\_INTR\_STATUS20, 64  
 TLMM\_GPIO\_INTR\_STATUS21, 66  
 TLMM\_GPIO\_INTR\_STATUS22, 220  
 TLMM\_GPIO\_INTR\_STATUS23, 222  
 TLMM\_GPIO\_INTR\_STATUS24, 263  
 TLMM\_GPIO\_INTR\_STATUS25, 265  
 TLMM\_GPIO\_INTR\_STATUS26, 267  
 TLMM\_GPIO\_INTR\_STATUS27, 270  
 TLMM\_GPIO\_INTR\_STATUS28, 224  
 TLMM\_GPIO\_INTR\_STATUS29, 226  
 TLMM\_GPIO\_INTR\_STATUS3, 55  
 TLMM\_GPIO\_INTR\_STATUS30, 229  
 TLMM\_GPIO\_INTR\_STATUS31, 231  
 TLMM\_GPIO\_INTR\_STATUS32, 69  
 TLMM\_GPIO\_INTR\_STATUS33, 71  
 TLMM\_GPIO\_INTR\_STATUS34, 73  
 TLMM\_GPIO\_INTR\_STATUS35, 75  
 TLMM\_GPIO\_INTR\_STATUS36, 78  
 TLMM\_GPIO\_INTR\_STATUS37, 80  
 TLMM\_GPIO\_INTR\_STATUS38, 82  
 TLMM\_GPIO\_INTR\_STATUS39, 84  
 TLMM\_GPIO\_INTR\_STATUS4, 243  
 TLMM\_GPIO\_INTR\_STATUS40, 87  
 TLMM\_GPIO\_INTR\_STATUS41, 89  
 TLMM\_GPIO\_INTR\_STATUS42, 91  
 TLMM\_GPIO\_INTR\_STATUS43, 93  
 TLMM\_GPIO\_INTR\_STATUS44, 96  
 TLMM\_GPIO\_INTR\_STATUS45, 98  
 TLMM\_GPIO\_INTR\_STATUS46, 100  
 TLMM\_GPIO\_INTR\_STATUS47, 102  
 TLMM\_GPIO\_INTR\_STATUS48, 105  
 TLMM\_GPIO\_INTR\_STATUS49, 107  
 TLMM\_GPIO\_INTR\_STATUS5, 57  
 TLMM\_GPIO\_INTR\_STATUS50, 109  
 TLMM\_GPIO\_INTR\_STATUS51, 111  
 TLMM\_GPIO\_INTR\_STATUS52, 114  
 TLMM\_GPIO\_INTR\_STATUS53, 272  
 TLMM\_GPIO\_INTR\_STATUS54, 274  
 TLMM\_GPIO\_INTR\_STATUS55, 116  
 TLMM\_GPIO\_INTR\_STATUS56, 118  
 TLMM\_GPIO\_INTR\_STATUS57, 120  
 TLMM\_GPIO\_INTR\_STATUS58, 123  
 TLMM\_GPIO\_INTR\_STATUS59, 276  
 TLMM\_GPIO\_INTR\_STATUS6, 60  
 TLMM\_GPIO\_INTR\_STATUS60, 279  
 TLMM\_GPIO\_INTR\_STATUS61, 281  
 TLMM\_GPIO\_INTR\_STATUS62, 283  
 TLMM\_GPIO\_INTR\_STATUS63, 285  
 TLMM\_GPIO\_INTR\_STATUS64, 125  
 TLMM\_GPIO\_INTR\_STATUS65, 127  
 TLMM\_GPIO\_INTR\_STATUS66, 288  
 TLMM\_GPIO\_INTR\_STATUS67, 290  
 TLMM\_GPIO\_INTR\_STATUS68, 292  
 TLMM\_GPIO\_INTR\_STATUS69, 294  
 TLMM\_GPIO\_INTR\_STATUS7, 62  
 TLMM\_GPIO\_INTR\_STATUS70, 297  
 TLMM\_GPIO\_INTR\_STATUS71, 299  
 TLMM\_GPIO\_INTR\_STATUS72, 301  
 TLMM\_GPIO\_INTR\_STATUS73, 303  
 TLMM\_GPIO\_INTR\_STATUS74, 306  
 TLMM\_GPIO\_INTR\_STATUS75, 308  
 TLMM\_GPIO\_INTR\_STATUS76, 310  
 TLMM\_GPIO\_INTR\_STATUS77, 312  
 TLMM\_GPIO\_INTR\_STATUS78, 315  
 TLMM\_GPIO\_INTR\_STATUS79, 129  
 TLMM\_GPIO\_INTR\_STATUS8, 245  
 TLMM\_GPIO\_INTR\_STATUS80, 132  
 TLMM\_GPIO\_INTR\_STATUS81, 233  
 TLMM\_GPIO\_INTR\_STATUS82, 235  
 TLMM\_GPIO\_INTR\_STATUS83, 134  
 TLMM\_GPIO\_INTR\_STATUS84, 136  
 TLMM\_GPIO\_INTR\_STATUS85, 138  
 TLMM\_GPIO\_INTR\_STATUS86, 141  
 TLMM\_GPIO\_INTR\_STATUS87, 143  
 TLMM\_GPIO\_INTR\_STATUS88, 145  
 TLMM\_GPIO\_INTR\_STATUS89, 147  
 TLMM\_GPIO\_INTR\_STATUS9, 247  
 TLMM\_GPIO\_INTR\_STATUS90, 150  
 TLMM\_GPIO\_INTR\_STATUS91, 152  
 TLMM\_GPIO\_INTR\_STATUS92, 154  
 TLMM\_GPIO\_INTR\_STATUS93, 156  
 TLMM\_GPIO\_INTR\_STATUS94, 159  
 TLMM\_GPIO\_INTR\_STATUS95, 161  
 TLMM\_GPIO\_INTR\_STATUS96, 163  
 TLMM\_GPIO\_INTR\_STATUS97, 165  
 TLMM\_GPIO\_INTR\_STATUS98, 168  
 TLMM\_GPIO\_INTR\_STATUS99, 170  
 TLMM\_GPIO\_ISLAND\_CFG0, 1232  
 TLMM\_GPIO\_ISLAND\_CFG1, 1233  
 TLMM\_GPIO\_ISLAND\_CFG10, 1242  
 TLMM\_GPIO\_ISLAND\_CFG11, 1243  
 TLMM\_GPIO\_ISLAND\_CFG12, 1246  
 TLMM\_GPIO\_ISLAND\_CFG13, 1247  
 TLMM\_GPIO\_ISLAND\_CFG14, 1248  
 TLMM\_GPIO\_ISLAND\_CFG15, 1249  
 TLMM\_GPIO\_ISLAND\_CFG16, 1244  
 TLMM\_GPIO\_ISLAND\_CFG17, 1245  
 TLMM\_GPIO\_ISLAND\_CFG18, 1250  
 TLMM\_GPIO\_ISLAND\_CFG19, 1251  
 TLMM\_GPIO\_ISLAND\_CFG2, 1234  
 TLMM\_GPIO\_ISLAND\_CFG20, 1252  
 TLMM\_GPIO\_ISLAND\_CFG21, 1253  
 TLMM\_GPIO\_ISLAND\_CFG22, 1254  
 TLMM\_GPIO\_ISLAND\_CFG23, 1255  
 TLMM\_GPIO\_ISLAND\_CFG24, 1256  
 TLMM\_GPIO\_ISLAND\_CFG25, 1257  
 TLMM\_GPIO\_ISLAND\_CFG26, 1258

TLMM_GPIO_ISLAND_CFG27	1259	322
TLMM_GPIO_ISLAND_CFG28	1260	TLMM_NORTH_IE_CTRL_DISABLE, 317
TLMM_GPIO_ISLAND_CFG29	1261	TLMM_NORTH_MPM_WAKEUP_INT_EN_0, 315
TLMM_GPIO_ISLAND_CFG3	1235	TLMM_PHASE_FLAG_STATUS, 237
TLMM_GPIO_ISLAND_CFG30	1262	TLMM_PMIC_HDRV_PULL_CTL, 206
TLMM_GPIO_ISLAND_CFG31	1263	TLMM_RESIN_N_HIHYS_CTL, 207
TLMM_GPIO_ISLAND_CFG4	1236	TLMM_RESOUT_HDRV_CTL, 320
TLMM_GPIO_ISLAND_CFG5	1237	TLMM_RFFE_CTL, 205
TLMM_GPIO_ISLAND_CFG6	1238	TLMM_SDC1_HDRV_PULL_CTL, 318
TLMM_GPIO_ISLAND_CFG7	1239	TLMM_SDC2_HDRV_PULL_CTL, 319
TLMM_GPIO_ISLAND_CFG8	1240	TLMM SNDWIRE_SLIMBUS_CTL, 320
TLMM_GPIO_ISLAND_CFG9	1241	TLMM_SOUTH_CLK_GATE_EN, 204
TLMM_GPIO_ISLAND_CODEC_RST	1264	TLMM_SOUTH_DIR_CONN_INTRn_CFG_CDSP, n=[0..1], 209
TLMM_GPIO_ISLAND_IN_OUT0	1233	TLMM_SOUTH_DIR_CONN_INTRn_CFG_GSS, n=[0..1], 208
TLMM_GPIO_ISLAND_IN_OUT1	1233	TLMM_SOUTH_DIR_CONN_INTRn_CFG_HMSS, n=[0..7], 208
TLMM_GPIO_ISLAND_IN_OUT10	1243	TLMM_SOUTH_DIR_CONN_INTRn_CFG_LPA_DSP, n=[0..5], 207
TLMM_GPIO_ISLAND_IN_OUT11	1244	TLMM_SOUTH_DIR_CONN_INTRn_CFG_RPM, n=[0..0], 208
TLMM_GPIO_ISLAND_IN_OUT12	1247	TLMM_SOUTH_DIR_CONN_INTRn_CFG_SENSORS, n=[0..1], 207
TLMM_GPIO_ISLAND_IN_OUT13	1248	TLMM_SOUTH_IE_CTRL_DISABLE, 205
TLMM_GPIO_ISLAND_IN_OUT14	1249	TLMM_SOUTH_MPM_WAKEUP_INT_EN_0, 202
TLMM_GPIO_ISLAND_IN_OUT15	1250	TLMM_SOUTH_MPM_WAKEUP_INT_EN_1, 204
TLMM_GPIO_ISLAND_IN_OUT16	1245	TLMM_UFS_REF_CLK_CTL, 321
TLMM_GPIO_ISLAND_IN_OUT17	1246	TLMM_UFS_RESET_CTL, 321
TLMM_GPIO_ISLAND_IN_OUT18	1251	TLMM_UFS_RESET, 322
TLMM_GPIO_ISLAND_IN_OUT19	1252	USB30_BAM_AHB_MASTER_ERR_ADDR_LSB, 525
TLMM_GPIO_ISLAND_IN_OUT2	1234	USB30_BAM_AHB_MASTER_ERR_ADDR_MSB, 525
TLMM_GPIO_ISLAND_IN_OUT20	1253	USB30_BAM_AHB_MASTER_ERR_ADDR, 524
TLMM_GPIO_ISLAND_IN_OUT21	1254	USB30_BAM_AHB_MASTER_ERR_CTRLS, 523
TLMM_GPIO_ISLAND_IN_OUT22	1255	USB30_BAM_AHB_MASTER_ERR_DATA, 524
TLMM_GPIO_ISLAND_IN_OUT23	1256	USB30_BAM_CNG_BITS_2, 521
TLMM_GPIO_ISLAND_IN_OUT24	1257	USB30_BAM_CTRL, 516
TLMM_GPIO_ISLAND_IN_OUT25	1258	USB30_BAM_DESC_CNT_TRSHLD, 518
TLMM_GPIO_ISLAND_IN_OUT26	1259	USB30_BAM_IRQ_CLR, 519
TLMM_GPIO_ISLAND_IN_OUT27	1260	USB30_BAM_IRQ_EN, 519
TLMM_GPIO_ISLAND_IN_OUT28	1261	USB30_BAM_IRQ_SRCS_EEn, n=[0..3], 527
TLMM_GPIO_ISLAND_IN_OUT29	1262	USB30_BAM_IRQ_SRCS_MSK_EEn, n=[0..3], 527
TLMM_GPIO_ISLAND_IN_OUT3	1236	USB30_BAM_IRQ_SRCS_MSK, 529
TLMM_GPIO_ISLAND_IN_OUT30	1263	USB30_BAM_IRQ_SRCS_UNMASKED_EEn, n=[0..3], 528
TLMM_GPIO_ISLAND_IN_OUT31	1264	USB30_BAM_IRQ_SRCS_UNMASKED, 530
TLMM_GPIO_ISLAND_IN_OUT4	1237	USB30_BAM_IRQ_SRCS, 529
TLMM_GPIO_ISLAND_IN_OUT5	1238	USB30_BAM_NUM_PIPES, 523
TLMM_GPIO_ISLAND_IN_OUT6	1239	USB30_BAM_P_AU_PSM_CNTXT_1_n, n=[0..3], 546
TLMM_GPIO_ISLAND_IN_OUT7	1240	USB30_BAM_P_CNSMR_SDBNDn, n=[0..3], 539
TLMM_GPIO_ISLAND_IN_OUT8	1241	USB30_BAM_P_CTRLn, n=[0..3], 531
TLMM_GPIO_ISLAND_IN_OUT9	1242	USB30_BAM_P_DATA_FIFO_ADDR LSBn, n=[0..3], 545
TLMM_INT_JTAG_CTL	317	USB30_BAM_P_DATA_FIFO_ADDR_MSBn, n=[0..3], 545
TLMM_JTAG_HDRV_CTL	319	USB30_BAM_P_DATA_FIFO_ADDRn, n=[0..3], 544
TLMM_LPI_DIR_CONN_INTRn_CFG_APPS, n=[0..2]	239	USB30_BAM_P_DESC_FIFO_ADDR LSBn, n=[0..3], 543
TLMM_LPI_GPIO_INTR_CFGn, n=[0..2]	238	USB30_BAM_P_DESC_FIFO_ADDR_MSBn, n=[0..3], 543
TLMM_MODE_PULL_CTL	318	USB30_BAM_P_DESC_FIFO_ADDRn, n=[0..3], 543
TLMM_NORTH_CLK_GATE_EN	317	USB30_BAM_P_DESC_FIFO_ADDR_MSBN, n=[0..3], 543
TLMM_NORTH_DIR_CONN_INTRn_CFG_CDSP, n=[0..1]	324	USB30_BAM_P_DESC_FIFO_ADDRn, n=[0..3], 543
TLMM_NORTH_DIR_CONN_INTRn_CFG_GSS, n=[0..1]	323	USB30_BAM_P_DF_CNTXT_n, n=[0..3], 550
TLMM_NORTH_DIR_CONN_INTRn_CFG_HMSS, n=[0..7]	323	USB30_BAM_P_EVNT_DEST_ADDR LSBn, n=[0..3], 541
TLMM_NORTH_DIR_CONN_INTRn_CFG_LPA_DSP, n=[0..5]	322	USB30_BAM_P_EVNT_DEST_ADDR_MSBn, n=[0..3], 541
TLMM_NORTH_DIR_CONN_INTRn_CFG_RPM, n=[0..0]	323	USB30_BAM_P_EVNT_DEST_ADDRn, n=[0..3], 540
TLMM_NORTH_DIR_CONN_INTRn_CFG_SENSORS, n=[0..1]		USB30_BAM_P_EVNT_GEN_TRSHLDn, n=[0..3], 546
		USB30_BAM_P_EVNT_REGn, n=[0..3], 542
		USB30_BAM_P_FIFO_SIZESn, n=[0..3], 544

USB30\_BAM\_P\_HALTN, n=[0..3], 533  
 USB30\_BAM\_P\_IRQ\_CLRn, n=[0..3], 535  
 USB30\_BAM\_P\_IRQ\_ENn, n=[0..3], 536  
 USB30\_BAM\_P\_IRQ\_STTSn, n=[0..3], 534  
 USB30\_BAM\_P\_PRDCR\_SDBNDn, n=[0..3], 538  
 USB30\_BAM\_P\_PSM\_CNTXT\_2\_n, n=[0..3], 547  
 USB30\_BAM\_P\_PSM\_CNTXT\_3\_LSBn, n=[0..3], 548  
 USB30\_BAM\_P\_PSM\_CNTXT\_3\_MSBn, n=[0..3], 548  
 USB30\_BAM\_P\_PSM\_CNTXT\_3\_n, n=[0..3], 547  
 USB30\_BAM\_P\_PSM\_CNTXT\_4\_n, n=[0..3], 549  
 USB30\_BAM\_P\_PSM\_CNTXT\_5\_n, n=[0..3], 549  
 USB30\_BAM\_P\_RETR\_CNTXT\_n, n=[0..3], 550  
 USB30\_BAM\_P\_RSTn, n=[0..3], 533  
 USB30\_BAM\_P\_SI\_CNTXT\_n, n=[0..3], 551  
 USB30\_BAM\_P\_SW\_OFSTS, n=[0..3], 540  
 USB30\_BAM\_P\_TIMER\_CTRLn, n=[0..3], 537  
 USB30\_BAM\_P\_TIMERn, n=[0..3], 537  
 USB30\_BAM\_P\_TRUST\_REGn, n=[0..3], 530  
 USB30\_BAM\_PIPE\_ATTR\_EEn, n=[0..3], 528  
 USB30\_BAM\_REVISION, 521  
 USB30\_BAM\_TIMER\_CTRL, 520  
 USB30\_BAM\_TIMER, 520  
 USB30\_BAM\_TRUST\_REG, 526  
 USB30\_QSCRATCH\_ALT\_INTERRUPT\_EN, 504  
 USB30\_QSCRATCH\_CGCTL\_REG, 504  
 USB30\_QSCRATCH\_CHARGING\_DET\_OUTPUT, 504  
 USB30\_QSCRATCH\_CTRL\_REG, 501  
 USB30\_QSCRATCH\_FLADJ\_30MHZ\_REG, 508  
 USB30\_QSCRATCH\_GENERAL\_CFG, 502  
 USB30\_QSCRATCH\_HS\_PHY\_CTRL, 503  
 USB30\_QSCRATCH\_HS\_PHY\_IRQ\_STAT, 504  
 USB30\_QSCRATCH\_HW\_SW\_EVT\_CTRL\_REG, 507  
 USB30\_QSCRATCH\_M\_AR\_USER\_REG, 509  
 USB30\_QSCRATCH\_M\_AW\_USER\_REG, 508  
 USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_MASK, 507  
 USB30\_QSCRATCH\_PWR\_EVNT\_IRQ\_STAT, 505  
 USB30\_QSCRATCH\_QSCRTCH\_REG\_n, n=[0..4], 510  
 USB30\_QSCRATCH\_SNPS\_CORE\_CFG, 510  
 USB30\_QSCRATCH\_SS\_PHY\_CTRL, 505  
 USB30\_QSCRATCH\_SS\_QMP\_PHY\_CTRL, 510  
 USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Hn, n=[0..2],  
 514  
 USB30\_QSCRATCH\_USB30\_GSI\_DBL\_ADDR\_Ln, n=[0..2],  
 513  
 USB30\_QSCRATCH\_USB30\_GSI\_DEPCMD\_ADDR\_L\_I-  
 PA\_EPn, n=[0..1], 515  
 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_H, 513  
 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_ON\_ERR\_L, 513  
 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_H, 512  
 USB30\_QSCRATCH\_USB30\_GSI\_EVT\_POINTER\_L, 512  
 USB30\_QSCRATCH\_USB30\_GSI\_GENERAL\_CFG, 511  
 USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Hn,  
 n=[0..2], 514  
 USB30\_QSCRATCH\_USB30\_GSI\_RING\_BASE\_ADDR\_Ln,  
 n=[0..2], 514  
 USB30\_QSCRATCH\_USB30\_IMODn, n=[0..1], 515  
 USB30\_QSCRATCH\_USB30\_STS\_REG, 511  
 USB30\_QSCRATCH\_USB30\_USEC\_CNT, 516  
 USB30\_QSRAM\_REGS\_QSRAM\_REG\_n, n=[0..63], 501