

Exynos 4412

RISC Microprocessor

**Public Version
September 2012**

User's Manual

SAMSUNG ELECTRONICS RESERVES THE RIGHT TO CHANGE PRODUCTS, INFORMATION AND SPECIFICATIONS WITHOUT NOTICE.

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind.

This document and all information discussed herein remain the sole and exclusive property of Samsung Electronics. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or otherwise.

Samsung products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.

For updates or additional information about Samsung products, contact your nearest Samsung office.

All brand names, trademarks and registered trademarks belong to their respective owners.

© 2012 Samsung Electronics Co., Ltd. All rights reserved.

Important Notice

Samsung Electronics Co. Ltd. ("Samsung") reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Customers are responsible for their own products and applications. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding

any information provided in this publication. Customer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

WARNING No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung. This publication is intended for use by designated recipients only. This publication contains confidential information (including trade secrets) of Samsung protected by Competition Law, Trade Secrets Protection Act and other related laws, and therefore may not be, in part or in whole, directly or indirectly publicized, distributed, photocopied or used (including in a posting on the Internet where unspecified access is possible) by any unauthorized third party. Samsung reserves its right to take any and all measures both in equity and law available to it and claim full damages against any party that misappropriates Samsung's trade secrets and/or confidential information.

警告 本文件仅向经韩国三星电子株式会社授权的人员提供，其内容含有商业秘密保护相关法规规定并受其保护的三星电子株式会社商业秘密，任何直接或间接非法向第三人披露、传播、复制或允许第三人使用该文件全部或部分内容的行为（包括在互联网等公开媒介刊登该商业秘密而可能导致不特定第三人获取相关信息的行为）皆为法律严格禁止。此等违法行为一经发现，三星电子株式会社有权根据相关法规对其采取法律措施，包括但不限于提出损害赔偿请求。

Copyright © 2012 Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711

Contact Us: mobilesol.cs@samsung.com

Home Page: <http://www.samsungsemi.com>

Trademarks

All brand names, trademarks and registered trademarks belong to their respective owners.

- Exynos, Exynos4412, FlexOneNAND, and OneNAND are trademarks of Samsung Electronics.
- ARM, Jazelle, TrustZone, and Thumb are registered trademarks of ARM Limited.
- Cortex, ETM, ETB, Coresight, ISA, and Neon are trademarks of ARM Limited.
- Java is a trademark of Sun Microsystems, Inc.
- SD is a registered trademark of Toshiba Corporation.
- MMC and eMMC are trademarks of MultiMediaCard Association.
- JTAG is a registered trademark of JTAG Technologies, Inc.
- Synopsys is a registered trademark of Synopsys, Inc.
- I2S is a trademark of Phillips Electronics.
- I2C is a trademark of Phillips Semiconductor Corp.
- MIPI and Slimbus are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

All other trademarks used in this publication are the property of their respective owners.

Chip Handling Guide

Precaution against Electrostatic Discharge

When using semiconductor devices, ensure that the environment is protected against static electricity:

1. Wear antistatic clothes and use earth band.
2. All objects that are in direct contact with devices must be made up of materials that do not produce static electricity.
3. Ensure that the equipment and work table are earthed.
4. Use ionizer to remove electron charge.

Contamination

Do not use semiconductor products in an environment exposed to dust or dirt adhesion.

Temperature/Humidity

Semiconductor devices are sensitive to:

- Environment
- Temperature
- Humidity

High temperature or humidity deteriorates the characteristics of semiconductor devices. Therefore, do not store or use semiconductor devices in such conditions.

Mechanical Shock

Do not apply excessive mechanical shock or force on semiconductor devices.

Chemical

Do not expose semiconductor devices to chemicals because exposure to chemicals leads to reactions that deteriorate the characteristics of the devices.

Light Protection

In non-Epoxy Molding Compound (EMC) package, do not expose semiconductor IC to bright light. Exposure to bright light causes malfunctioning of the devices. However, a few special products that utilize light or with security functions are exempted from this guide.

Radioactive, Cosmic and X-ray

Radioactive substances, cosmic ray, or X-ray may influence semiconductor devices. These substances or rays may cause a soft error during a device operation. Therefore, ensure to shield the semiconductor devices under environment that may be exposed to radioactive substances, cosmic ray, or X-ray.

EMS (Electromagnetic Susceptibility)

Strong electromagnetic wave or magnetic field may affect the characteristic of semiconductor devices during the operation under insufficient PCB circuit design for Electromagnetic Susceptibility (EMS).

Revision History

| Revision No. | Date | Description | Author(s) |
|--------------|---------------|-------------------|--------------|
| 1.00 | Sep. 17, 2012 | • Initial version | Chongkun Lee |

Table of Contents

| | |
|--|------------|
| 1 PRODUCT OVERVIEW | 1-1 |
| 1.1 Introduction | 1-1 |
| 1.2 Features | 1-2 |
| 1.2.1 Multi-Core Processing Unit..... | 1-4 |
| 1.2.2 Memory Subsystem..... | 1-5 |
| 1.2.3 Multimedia | 1-6 |
| 1.2.4 Audio Subsystem..... | 1-8 |
| 1.2.5 Image Signal Processing Subsystem | 1-8 |
| 1.2.6 Connectivity | 1-9 |
| 1.2.7 System Peripheral | 1-11 |
| 1.3 Conventions | 1-13 |
| 1.3.1 Register RW Conventions | 1-13 |
| 1.3.2 Register Value Conventions | 1-13 |
| 2 MEMORY MAP | 2-1 |
| 2.1 Overview | 2-1 |
| 2.2 SFR Base Address | 2-2 |
| 3 CHIP ID..... | 3-1 |
| 3.1 Overview | 3-1 |
| 3.2 Register Description..... | 3-2 |
| 3.2.1 Register Map Summary | 3-2 |
| 4 GENERAL PURPOSE INPUT/OUTPUT (GPIO) CONTROL | 4-1 |
| 4.1 Overview | 4-1 |
| 4.2 Features | 4-3 |
| 4.2.1 Input/Output Description..... | 4-3 |
| 4.3 Register Description..... | 4-5 |
| 4.3.1 Registers Summary | 4-5 |
| 4.3.2 Part 1 | 4-20 |
| 4.3.3 Part 2 | 4-124 |
| 4.3.4 Part 3 | 4-289 |
| 4.3.5 Part 4 | 4-298 |
| 5 CLOCK MANAGEMENT UNIT | 5-1 |
| 5.1 Overview | 5-1 |
| 5.2 Clock Declaration | 5-2 |
| 5.2.1 Clocks from Clock Pads | 5-2 |
| 5.2.2 Clocks from CMU..... | 5-3 |
| 5.3 Clock Relationship | 5-4 |
| 5.3.1 Recommended PLL PMS Value for APLL and MPLL | 5-6 |
| 5.3.2 Recommended PLL PMS Value for EPLL | 5-7 |
| 5.3.3 Recommended PLL PMS Value for VPLL | 5-8 |
| 5.4 Clock Generation | 5-9 |
| 5.5 Clock Configuration Procedure | 5-14 |
| 5.5.1 Clock Gating | 5-15 |

| | |
|---|-------------|
| 5.5.2 Clock Diving..... | 5-15 |
| 5.6 Special Clock Description | 5-16 |
| 5.6.1 Special Clock Table | 5-16 |
| 5.7 CLKOUT | 5-19 |
| 5.8 I/O Description | 5-22 |
| 5.9 Register Description..... | 5-23 |
| 5.9.1 Register Map Summary..... | 5-25 |
| 6 INTERRUPT CONTROLLER | 6-1 |
| 6.1 Overview | 6-1 |
| 6.2 Features | 6-2 |
| 6.2.1 Security Extensions Support | 6-2 |
| 6.2.2 Implementation-Specific Configurable Features | 6-3 |
| 6.3 Interrupt Source | 6-4 |
| 6.3.1 Interrupt Sources Connection..... | 6-4 |
| 6.3.2 GIC Interrupt Table | 6-5 |
| 6.4 Functional Overview | 6-13 |
| 6.5 Register Description..... | 6-14 |
| 6.5.1 Register Map Summary..... | 6-14 |
| 7 INTERRUPT COMBINER | 7-1 |
| 7.1 Overview | 7-1 |
| 7.2 Features | 7-1 |
| 7.3 Functional Description | 7-2 |
| 7.3.1 Block Diagram | 7-2 |
| 7.4 Interrupt Sources..... | 7-3 |
| 7.4.1 Interrupt Combiner..... | 7-3 |
| 7.5 Functional Description | 7-8 |
| 7.6 Register Description..... | 7-9 |
| 7.6.1 Register Map Summary..... | 7-9 |
| 7.6.2 Interrupt Combiner..... | 7-10 |
| 8 DIRECT MEMORY ACCESS CONTROLLER (DMAC) | 8-1 |
| 8.1 Overview | 8-1 |
| 8.2 Features | 8-2 |
| 8.3 Register Description..... | 8-5 |
| 8.3.1 Register Map Summary..... | 8-5 |
| 8.4 Instruction..... | 8-14 |
| 9 SRAM CONTROLLER..... | 9-1 |
| 9.1 Overview | 9-1 |
| 9.2 Features | 9-1 |
| 9.3 Block Diagram..... | 9-1 |
| 9.4 Functional Description | 9-2 |
| 9.4.1 nWAIT Pin Operation..... | 9-2 |
| 9.4.2 Programmable Access Cycle | 9-3 |
| 9.5 I/O Description | 9-4 |
| 9.6 Register Description..... | 9-5 |
| 9.6.1 Register Map Summary..... | 9-5 |
| 10 NAND FLASH CONTROLLER | 10-1 |

| | |
|--|-------|
| 10.1 Overview | 10-1 |
| 10.2 Features | 10-1 |
| 10.3 Functional Description | 10-2 |
| 10.3.1 Block Diagram | 10-2 |
| 10.3.2 NAND Flash Memory Timing | 10-3 |
| 10.4 Software Mode | 10-4 |
| 10.4.1 Data Register Configuration | 10-4 |
| 10.4.2 1/4/8/12/16-bit ECC | 10-5 |
| 10.4.3 2048 Byte 1-bit ECC Parity Code Assignment Table | 10-6 |
| 10.4.4 32 Byte 1-bit ECC Parity Code Assignment Table | 10-6 |
| 10.4.5 1-bit ECC Module Features | 10-6 |
| 10.4.6 1-bit ECC Programming Guide | 10-7 |
| 10.4.7 4-bit ECC Programming Guide (ENCODING) | 10-8 |
| 10.4.8 4-bit ECC Programming Guide (DECODING) | 10-9 |
| 10.4.9 8/12/16-bit ECC Programming Guide (ENCODING) | 10-10 |
| 10.4.10 8/12/16-bit ECC Programming Guide (DECODING) | 10-11 |
| 10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC | 10-12 |
| 10.4.12 Lock Scheme for Data Protection | 10-13 |
| 10.5 Programming Constraints | 10-14 |
| 10.6 I/O Description | 10-14 |
| 10.7 Register Description | 10-15 |
| 10.7.1 Register Map Summary | 10-15 |
| 10.7.2 NAND Flash Interface and 1/4-bit ECC Registers | 10-17 |
| 10.7.3 ECC Registers for 8, 12 and 16-bit ECC | 10-29 |

11 PULSE WIDTH MODULATION TIMER **11-1**

| | |
|---|-------|
| 11.1 Overview | 11-1 |
| 11.2 Features | 11-4 |
| 11.3 PWM Operation | 11-5 |
| 11.3.1 Prescaler and Divider | 11-5 |
| 11.3.2 Basic Timer Operation | 11-6 |
| 11.3.3 Auto-Reload and Double Buffering | 11-8 |
| 11.3.4 Timer Operation Example | 11-9 |
| 11.3.5 Initialize Timer (Setting Manual-Up Data and Inverter) | 11-10 |
| 11.3.6 PWM | 11-10 |
| 11.3.7 During Current ISR. (Interrupt Service Routine) Output Level Control | 11-11 |
| 11.3.8 Dead Zone Generator | 11-12 |
| 11.4 I/O Description | 11-13 |
| 11.5 Register Description | 11-14 |
| 11.5.1 Register Map Summary | 11-14 |

12 WATCHDOG TIMER..... **12-1**

| | |
|---|------|
| 12.1 Overview | 12-1 |
| 12.2 Features | 12-1 |
| 12.3 Functional Description | 12-2 |
| 12.3.1 WDT Operation | 12-2 |
| 12.3.2 WTDAT and WTCNT | 12-3 |
| 12.3.3 WDT Start | 12-3 |
| 12.3.4 Consideration of Debugging Environment | 12-3 |
| 12.4 Register Description | 12-4 |
| 12.4.1 Register Map Summary | 12-4 |

| | |
|---|-------------|
| 13 UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER | 13-1 |
| 13.1 Overview | 13-1 |
| 13.2 Features | 13-0 |
| 13.3 UART Description | 13-1 |
| 13.3.1 Data Transmission..... | 13-2 |
| 13.3.2 Data Reception | 13-2 |
| 13.3.3 AFC..... | 13-3 |
| 13.3.4 Example of Non AFC (Controlling nRTS and nCTS by Software) | 13-4 |
| 13.3.5 Trigger Level of Tx/Rx FIFO and DMA Burst Size in DMA Mode | 13-4 |
| 13.3.6 RS-232C Interface | 13-4 |
| 13.3.7 Interrupt/DMA Request Generation | 13-5 |
| 13.3.8 UART Error Status FIFO | 13-7 |
| 13.4 UART Input Clock Description | 13-10 |
| 13.5 I/O Description | 13-11 |
| 13.6 Register Description..... | 13-12 |
| 13.6.1 Register Map Summary..... | 13-12 |
| 14 INTER-INTEGRATED CIRCUIT..... | 14-1 |
| 14.1 Overview | 14-1 |
| 14.2 Features | 14-2 |
| 14.3 Functional Description | 14-2 |
| 14.3.1 Block Diagram | 14-2 |
| 14.4 I2C-Bus Interface Operation | 14-3 |
| 14.4.1 Start and Stop Conditions..... | 14-4 |
| 14.4.2 Data Transfer Format | 14-5 |
| 14.4.3 ACK Signal Transmission..... | 14-6 |
| 14.4.4 Read-Write Operation..... | 14-7 |
| 14.4.5 Bus Arbitration Procedures..... | 14-7 |
| 14.4.6 Abort Conditions | 14-7 |
| 14.4.7 Configuring I2C-Bus | 14-7 |
| 14.4.8 Flowcharts of Operations in Each Mode | 14-8 |
| 14.5 I/O Description | 14-12 |
| 14.6 Register Description..... | 14-13 |
| 14.6.1 Register Map Summary..... | 14-13 |
| 15 SERIAL PERIPHERAL INTERFACE..... | 15-1 |
| 15.1 Overview | 15-1 |
| 15.2 Features | 15-1 |
| 15.2.1 Operation of SPI | 15-2 |
| 15.3 SPI Input Clock Description | 15-5 |
| 15.4 IO Description | 15-6 |
| 15.5 Register Description..... | 15-7 |
| 15.5.1 Register Map Summary..... | 15-7 |
| 16 DISPLAY CONTROLLER | 16-1 |
| 16.1 Overview | 16-1 |
| 16.2 Features | 16-2 |
| 16.3 Functional Description | 16-4 |
| 16.3.1 Brief Description | 16-4 |
| 16.3.2 Data Flow..... | 16-5 |
| 16.3.3 Overview of the Color Data | 16-8 |

| | |
|---|--------|
| 16.3.4 Palette Usage | 16-23 |
| 16.3.5 Window Blending | 16-26 |
| 16.3.6 VTIME Controller Operation | 16-35 |
| 16.3.7 Virtual Display | 16-39 |
| 16.3.8 RGB Interface Specification | 16-40 |
| 16.3.9 LCD Indirect i80 System Interface..... | 16-43 |
| 16.4 I/O Description | 16-47 |
| 16.5 Register Description..... | 16-48 |
| 16.5.1 Register Map Summary..... | 16-49 |
| 16.5.2 Palette Memory..... | 16-56 |
| 16.5.3 Control Register..... | 16-57 |
| 16.5.4 Gamma Lookup Table | 16-131 |
| 16.5.5 Shadow Windows Control | 16-134 |
| 16.5.6 Palette Ram | 16-136 |

17 KEYPAD INTERFACE.....17-1

| | |
|--|-------|
| 17.1 Overview | 17-1 |
| 17.2 Debouncing Filter..... | 17-3 |
| 17.3 Filter Clock | 17-3 |
| 17.4 Wakeup Source..... | 17-3 |
| 17.5 Keypad Scanning Procedure for Software Scan | 17-4 |
| 17.6 Keypad Scanning Procedure for Hardware Scan | 17-9 |
| 17.7 I/O Description | 17-10 |
| 17.8 Register Description..... | 17-12 |
| 17.8.1 Register Map Summary..... | 17-12 |

18 ADC18-1

| | |
|-----------------------------------|------|
| 18.1 Overview | 18-1 |
| 18.2 Features | 18-1 |
| 18.3 Functional Description | 18-1 |
| 18.3.1 Block Diagram | 18-1 |
| 18.3.2 ADC Selection | 18-2 |
| 18.3.3 A/D Conversion Time..... | 18-2 |
| 18.3.4 ADC Conversion Mode | 18-2 |
| 18.3.5 Standby Mode..... | 18-3 |
| 18.4 ADC Input Clock Diagram..... | 18-4 |
| 18.5 I/O Descriptions..... | 18-5 |
| 18.6 Register Description..... | 18-6 |
| 18.6.1 Register Map Summary..... | 18-6 |

List of Figures

| Figure Number | Title | Page Number |
|--------------------------|---|------------------------|
| Figure 4-1 | GPIO Block Diagram | 4-4 |
| Figure 5-1 | Exynos 4412 Clock Generation Circuit (CPU, BUS, DRAM, ISP Clocks) | 5-11 |
| Figure 5-2 | Exynos 4412 Clock Generation Circuit (Special Clocks)..... | 5-13 |
| Figure 5-3 | Exynos 4412 CLKOUT Control Logic | 5-19 |
| Figure 5-4 | Exynos 4412 Clock Controller Address Map | 5-24 |
| Figure 6-1 | Interrupt Sources Connection | 6-4 |
| Figure 7-1 | Block Diagram of Interrupt Combiner | 7-2 |
| Figure 8-1 | Two DMA Tops | 8-1 |
| Figure 9-1 | Block Diagram of SROMC Introduction | 9-1 |
| Figure 9-2 | SROMC nWAIT Timing Diagram | 9-2 |
| Figure 9-3 | SROMC Read Timing Diagram | 9-3 |
| Figure 9-4 | SROMC Write Timing Diagram..... | 9-3 |
| Figure 10-1 | NAND Flash Controller Block Diagram..... | 10-2 |
| Figure 10-2 | CLE and ALE Timing (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0) | 10-3 |
| Figure 10-3 | nWE and nRE Timing (TWRPH0 = 0, TWRPH1 = 0) | 10-3 |
| Figure 10-4 | Accessibility of NAND Area | 10-13 |
| Figure 11-1 | Simple Example of a PWM Cycle | 11-2 |
| Figure 11-2 | PWM TIMER Clock Tree Diagram..... | 11-3 |
| Figure 11-3 | Timer Operations | 11-6 |
| Figure 11-4 | Example of Double Buffering Feature | 11-8 |
| Figure 11-5 | Example of a Timer Operation..... | 11-9 |
| Figure 11-6 | Example of PWM | 11-10 |
| Figure 11-7 | Inverter On/Off | 11-11 |
| Figure 11-8 | Waveform when a Dead Zone Feature is Enabled..... | 11-12 |
| Figure 12-1 | Watchdog Timer Block Diagram | 12-2 |
| Figure 13-1 | Block Diagram of UART | 13-1 |
| Figure 13-2 | UART AFC Interface | 13-3 |
| Figure 13-3 | UART Receives the Five Characters Including Two Errors | 13-7 |
| Figure 13-4 | IrDA Function Block Diagram | 13-8 |
| Figure 13-5 | Serial I/O Frame Timing Diagram (Normal UART) | 13-8 |
| Figure 13-6 | Infra-Red Transmit Mode Frame Timing Diagram..... | 13-9 |
| Figure 13-7 | Infra-Red Receive Mode Frame Timing Diagram..... | 13-9 |
| Figure 13-8 | Input Clock Diagram for UART | 13-10 |
| Figure 13-9 | nCTS and Delta CTS Timing Diagram | 13-25 |
| Figure 13-10 | Block Diagram of UINTSP, UINTP, and UINTM..... | 13-30 |
| Figure 14-1 | I2C-Bus Block Diagram | 14-2 |

| | | |
|-----------------|---|-------|
| Figure 14-2 | Start and Stop Condition..... | 14-4 |
| Figure 14-3 | I2C-Bus Interface Data Format..... | 14-5 |
| Figure 14-4 | Data Transfer on the I2C-Bus..... | 14-5 |
| Figure 14-5 | Acknowledgement on the I2C-Bus | 14-6 |
| Figure 14-6 | Operations for Master/Transmitter Mode..... | 14-8 |
| Figure 14-7 | Operations for Master/Receiver Mode..... | 14-9 |
| Figure 14-8 | Operations for Slave/Transmitter Mode..... | 14-10 |
| Figure 14-9 | Operations for Slave/Receiver Mode..... | 14-11 |
| Figure 15-1 | SPI Transfer Format | 15-4 |
| Figure 15-2 | Input Clock Diagram for SPI | 15-5 |
| Figure 15-3 | Auto Chip Select Mode Waveform (CPOL = 0, CPHA = 0, CH_WIDTH = Byte) | 15-10 |
| Figure 16-1 | Block Diagram of Display Controller | 16-1 |
| Figure 16-2 | Block Diagram of the Data Flow | 16-6 |
| Figure 16-3 | Block Diagram of the Interface | 16-7 |
| Figure 16-4 | Memory Format of 25 BPP(A888) Display | 16-8 |
| Figure 16-5 | Memory Format of 32 BPP (8888) Display..... | 16-9 |
| Figure 16-6 | Memory Format of 24 BPP (A887) Display | 16-10 |
| Figure 16-7 | Memory Format of 24 BPP (888) Display | 16-11 |
| Figure 16-8 | Memory Format of 19 BPP (A666) Display | 16-12 |
| Figure 16-9 | Memory Format of 18 BPP (666) Display | 16-13 |
| Figure 16-10 | Memory Format of 16 BPP (A555) Display | 16-14 |
| Figure 16-11 | Memory Format of 16 BPP (1555) Display..... | 16-15 |
| Figure 16-12 | Memory Format of 16 BPP (565) Display..... | 16-16 |
| Figure 16-13 | 16 BPP (5:6:5) Display Types..... | 16-17 |
| Figure 16-14 | Memory Format of 13 BPP (A444) Display | 16-18 |
| Figure 16-15 | Memory Format of 8 BPP (A232) Display | 16-19 |
| Figure 16-16 | Memory Format of 8 BPP Display | 16-20 |
| Figure 16-17 | Memory Format of 4 BPP Display | 16-21 |
| Figure 16-18 | Memory Format of 2 BPP Display | 16-22 |
| Figure 16-19 | 32 BPP (8:8:8:8) Palette Data Format..... | 16-23 |
| Figure 16-20 | 25 BPP (A: 8:8:8) Palette Data Format | 16-24 |
| Figure 16-21 | 19 BPP (A: 6:6:6) Palette Data Format | 16-24 |
| Figure 16-22 | 16 BPP (A: 5:5:5) Palette Data Format | 16-25 |
| Figure 16-23 | Blending Equation..... | 16-28 |
| Figure 16-24 | Blending Diagram | 16-30 |
| Figure 16-25 | Blending Factor Decision | 16-31 |
| Figure 16-26 | Color-Key Function Configurations..... | 16-32 |
| Figure 16-27 | Blending and Color-Key Function | 16-33 |
| Figure 16-28 | Blending Decision Diagram | 16-34 |
| Figure 16-29 | Example of Scrolling in Virtual Display | 16-39 |
| Figure 16-30 | LCD RGB Interface Timing | 16-42 |
| Figure 16-31 | Indirect i80 System Interface Write Cycle Timing..... | 16-45 |
| Figure 17-1 | Key Matrix Interface External Connection Guide..... | 17-2 |
| Figure 17-2 | Internal Debouncing Filter Operation..... | 17-3 |
| Figure 17-3 | Keypad Scanning Procedure | 17-4 |
| Figure 17-4 | Keypad Scanning Procedure II | 17-5 |
| Figure 17-5 | Keypad Scanning Procedure III | 17-6 |
| Figure 17-6 | Keypad Scanning Procedure when the Two-key Pressed with Different Row | 17-7 |
| Figure 17-7 | Keypad I/F Block Diagram | 17-8 |

| | | |
|-------------|--|------|
| Figure 18-1 | ADC Top/Bottom Offset Error Diagram | 18-1 |
| Figure 18-2 | ADC Functional Block Diagram | 18-1 |
| Figure 18-3 | ADC Selection..... | 18-2 |
| Figure 18-4 | Input Clock Diagram for ADC & Touch Screen Interface | 18-4 |

List of Tables

| Table Number | Title | Page Number |
|-----------------|--|----------------|
| Table 5-1 | Operating Frequencies in Exynos 4412..... | 5-1 |
| Table 5-2 | APLL and MPLL PMS Value..... | 5-6 |
| Table 5-3 | EPLL PMS Value | 5-7 |
| Table 5-4 | VPLL PMS Value | 5-8 |
| Table 5-5 | Special Clocks in Exynos 4412..... | 5-16 |
| Table 5-6 | I/O Clocks in Exynos 4412..... | 5-18 |
| Table 5-7 | CLKOUT Input Clock Selection Information | 5-20 |
| Table 5-8 | I/O Description | 5-22 |
| Table 6-1 | GIC Configuration Values | 6-3 |
| Table 6-2 | GIC Interrupt Table (SPI[127:0])..... | 6-5 |
| Table 6-3 | GIC Interrupt Table (PPI[15:0])..... | 6-12 |
| Table 7-1 | Interrupt Groups of Interrupt Combiner..... | 7-3 |
| Table 8-1 | Features of DMA Controller | 8-2 |
| Table 8-2 | DMA Request Mapping Table..... | 8-2 |
| Table 11-1 | Minimum and Maximum Resolution Based on Prescaler and Clock Divider Values..... | 11-5 |
| Table 13-1 | Interrupts in Connection with FIFO | 13-6 |
| Table 16-1 | Relation 16 BPP between VCLK and CLKVAL (TFT, Frequency of Video Clock Source = 60 MHz) 16-35 | |
| Table 16-2 | RGB Interface Signals of Display Controller | 16-40 |
| Table 16-3 | LCD Indirect i80 System Interface Signals of Display Controller..... | 16-43 |
| Table 16-4 | Timing Reference Code (XY Definition)..... | 16-46 |
| Table 16-5 | I/O Signals of Display Controller | 16-47 |
| Table 17-1 | Keypad Inaterface I/O Description | 17-10 |

List of Examples

| Example Number | Title | Page Number |
|---------------------------|---------------------------------------|------------------------|
| Example 16-1 | Application..... | 16-26 |
| Example 16-2 | Total Five Windows..... | 16-26 |
| Example 16-3 | Blending Equation..... | 16-27 |
| Example 16-4 | Default Blending Equation | 16-28 |
| Example 16-5 | Window Blending Factor Decision | 16-30 |
| Example 16-6 | Hue Equation | 16-119 |
| Example 16-7 | Coefficient Decision | 16-119 |

List of Conventions

Register RW Access Type Conventions

| Type | Definition | Description |
|------|--------------|---|
| R | Read Only | The application has permission to read the Register field. Writes to read-only fields have no effect. |
| W | Write Only | The application has permission to write in the Register field. |
| RW | Read & Write | The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0. |

Register Value Conventions

| Expression | Description |
|------------------|--|
| x | Undefined bit |
| X | Undefined multiple bits |
| ? | Undefined, but depends on the device or pin status |
| Device dependent | The value depends on the device |
| Pin value | The value depends on the pin status |

Reset Value Conventions

| Expression | Description |
|------------|---------------------------|
| 0 | Clears the register field |
| 1 | Sets the register field |
| x | Don't care condition |

Warning: Some bits of control registers are driven by hardware or write operation only. As a result the indicated reset value and the read value after reset might be different.

1 Product Overview

1.1 Introduction

Exynos 4412 is a 32-bit RISC cost-effective, low power, performance optimized and Coretex-A9 Quad Core based micro-processor solution for smart phone applications.

The memory system has dedicated DRAM ports and Static Memory port. The dedicated DRAM ports support LPDDR2 interface for high bandwidth. Static Memory Port supports NOR Flash and ROM type external memory and components.

To reduce the total system cost and enhance the overall functionality, Exynos 4412 includes many hardware peripherals, such as TFT 24-bit true color LCD controller, Camera Interface, MIPI DSI, CSI-2, System Manager for power management, MIPI HSI, four UARTs, 24-channel DMA, Timers, General I/O Ports, three I2S, S/PDIF, eight IIC-BUS interface, three HS-SPI, USB Host 2.0, USB 2.0 Device operating at high speed (480 Mbps), two USB HSIC, four SD Host and high-speed Multimedia Card Interface, and four PLLs for clock generation.

1.2 Features

The features of Exynos 4412 are:

- ARM Cortex-A9 based Quad CPU Subsystem with NEON
 - 32/32/32 KB I/D Cache, 1 MB L2 Cache
 - Operating frequency up to 1.4 GHz
- 128-bit/64-bit Multi-layer bus architecture
 - Core-D domain for ARM Cortex-A9 Quad, CoreSight, and external memory interface
- Operating frequency up to 200 MHz
 - Global D- domain mainly for multimedia components and external storage interfaces
- Operating frequency up to 100 MHz
 - Core-P, Global-P domain mainly for other system component, such as system peripherals, peripheral DMAs, connectivity IPs and Audio interfaces.
- Operating frequency up to 100 MHz
 - Audio domain for low power audio play
- Advanced power management for mobile applications
- 64 KB ROM for secure booting and 256 KB RAM for security function
- 8-bit ITU 601/656 Camera Interface
- 2D Graphics Acceleration support.
- 1/2/4/ 8bpp Palletized or 8/16/24bpp Non-Palletized Color TFT recommend up to WXGA resolution
- HDMI interface support for NTSC and PAL mode with image enhancer
- MIPI-DSI and MIPI-CSI interface support
- One AC-97 audio codec interface and 3-channel PCM serial audio interface
- Three 24-bit I2S interface support
- One TX only S/PDIF interface support for digital audio
- Eight I2C interface support
- Three SPI support
- Four UART supports three Mbps ports for Bluetooth 2.0
- On-chip USB 2.0 Device supports high-speed (480 Mbps, on-chip transceiver)
- On-chip USB 2.0 Host support
- Two on-chip USB HSIC
- Four SD/ SDIO/HS-MMC interface support

- 24-channel DMA controller (8 channels for Memory-to-memory DMA, 16 channels for Peripheral DMA)
- Supports 14 × 8 key matrix
- Configurable GPIOs
- Real time clock, PLL, timer with PWM, and watch dog timer
- Multi-core timer support for accurate tick time in power down mode (except sleep mode)
- Memory Subsystem
 - Asynchronous SRAM/ ROM/NOR interface with x8 or x16 data bus
 - NAND interface with x8 data bus
 - LPDDR2 interface (800 Mbps/pin DDR)

1.2.1 Multi-Core Processing Unit

The features of main microprocessors are:

- The ARM Cortex-A9 MPCore (quad core) processor integrates the proven and highly successful ARM MPCore technology along with further enhancements to simplify and broaden the adoption of multi-core solutions.
- With the ability to scale in speed from 200 MHz to 1.4 GHz, the ARM Cortex-A9 MPCore quad processor meets the requirements of power-optimized mobile devices, which require operation in low power and performance-optimized consumer applications.
- Other features of ARM Cortex-A9 MPCore quad core processor are:
 - Thumb-2 technology for greater performance, energy efficiency, and code density
 - NEON™ signal processing extensions
 - Jazelle RCT Java-acceleration technology
 - TrustZone technology for secure transactions and DRM
 - Floating-Point unit for significant acceleration for both single and double precision scalar Floating-Point operations
 - Optimized L1 caches for performance and power
 - Integrated 1 MB L2 Cache using standard compiled RAMs
 - Program Trace Macrocell and CoreSight
- Generic Interrupt Controller
 - Supports three interrupt types
 - Software Generated Interrupt (SGI)
 - Private Peripheral Interrupt (PPI)
 - Shared Peripheral Interrupt (SPI)
 - Programmable interrupts that enable to set the
 - Security state for an interrupt
 - Priority level of an interrupt
 - Enabling or disabling of an interrupt
 - Processors that receive an interrupt
 - Enhanced security features

1.2.2 Memory Subsystem

The features of memory subsystem are:

- High bandwidth Memory Matrix subsystem
- Two independent external memory ports:
 - 1x16 Static Hybrid Memory port
 - 2x32 DRAM port
- Matrix architecture increases the overall bandwidth with simultaneous access capability:
 - SRAM/ROM/NOR Interface
 - x8 or x16 data bus
 - Addresses range support: 23-bit
 - Supports asynchronous interface
 - Supports byte and half-word access
 - NAND Interface
 - Supports industry standard NAND interface
 - x8 data bus
 - LPDDR2 interface
 - x32 data bus up to 800 Mbps/pin
 - 1.2 V interface voltage
 - Density support up to 4-Gb per port (2CS)

1.2.3 Multimedia

The features of multimedia are:

- Camera Interface
 - Multiple input support
 - ITU-R BT 601/656 mode
 - DMA (AXI 64-bit interface) mode
 - MIPI (CSI) mode
 - Direct FIFO mode (from LCDC)
 - Multiple output support
 - DMA (AXI 64-bit interface) mode
 - Direct FIFO mode (to LCDC)
 - Digital Zoom In (DZI) capability
 - Multiple camera input support
 - Programmable polarity of video sync signals
 - Image mirror and rotation (X-axis mirror, Y-axis mirror, 90°, 180°, and 270° rotation)
 - Various image formats generation
 - Capture frame control support
 - Image effect support
- JPEG Codec supports:
 - Supported format of compression
 - Input raw image: YCbCr4:2:2 or RGB 565
 - Output JPEG file: Baseline JPEG of YCbCr4:2:2 or YCbCr4:2:0
 - General-purpose color-space converter
- 2D Graphic Engine supports:
 - BitBLT
 - Window clipping, 90°/180°/270°/Rotation, X Flip/Y Flip
 - Totally 4-operand raster operation (ROP4)
 - Alpha blending (user-specified constant alpha value/per-pixel alpha value)
 - 8/16/24/32-bpp. Packed 24-bpp color format, Premultiplied/Non-premultiplied alpha format
 - 1 bpp/4 bpp/8 bpp/16 bpp/32 bpp Mask format, YCbCr format

- Digital TV Interface supports:
 - High-Definition Multimedia Interface (HDMI) 1.4 a
 - Up to 1080 p 60 Hz and 8-channel/112 kHz/24-bit audio
 - 480 p, 576 p, 720 p, 1080i (cannot support 480i)
 - HDCP V1.1
 - 3D support
- Rotator
 - Supported image format: YCbCr422 (Interleave), YCbCr420 (Non-interleave), and RGB565 and RGB888 (unpacked)
 - Supported rotate degree: 90, 180, 270, flip vertical, and flip horizontal
- Video processor: The video processor supports:
 - BOB/2D-IPC mode
 - Production of YCbCr 4: 4: 4 output to help the mixer blend video and graphics
 - 1/4X to 16X vertical scaling with 4-tap/16-phase polyphase filter
 - 1/4X to 16X horizontal scaling with 8-tap/16-phase polyphase filter
 - Pan and scan, Letterbox, and NTSC/PAL conversion using scaling
 - Flexible scaled video positioning within display area
 - 1/16 pixel resolution Pan and Scan modes
 - Flexible post video processing
 - Color saturation, brightness/contrast enhancement, edge enhancement
 - Color space conversion between BT.601 and BT.709
 - Video input source size up to 1920 × 1080
- Video Mixer
 - The Video Mixer supports:
 - Overlapping and blending input video and graphic layers
 - 480p, 576p, 720p, and 1080i/p display size
 - Four layers (1 video layer, 2 graphic layer, and 1 background layer)
- TFT-LCD Interface
 - The TFT-LCD Interface supports:
 - 24/18/16-bpp parallel RGB Interface LCD
 - 8/6 bpp serial RGB Interface
 - Dual i80 Interface LCD
 - 1/2/4/8 bpp Palletized or 8/16/24-bpp Non-Palletized Color TFT
 - Typical actual screen size: 1080 × 1024, 1024 × 768, 800 × 480, 640 × 480, 320 × 240, 160 × 160, and so on
 - Virtual image up to 16M pixel (4K pixel × 4K pixel)
 - Five Window Layers for PIP or OSD
 - Real-time overlay plane multiplexing
 - Programmable OSD window positioning
 - 16-level alpha blending

1.2.4 Audio Subsystem

The features of audio subsystem are:

- Reconfigurable Processor (RP) progresses audio processing
- Low power audio subsystem
 - 5.1 channel I2S with 32-bit-width 64-depth FIFO
 - 128 KB audio play output buffer
 - Hardware mixer mixes primary and secondary sounds

1.2.5 Image Signal Processing Subsystem

The features of ISP subsystem are:

- Dual camera input
- Image signal processing
- Dynamic range correction
- Face detection

1.2.6 Connectivity

The features of connectivity are:

- PCM Audio Interface supports:
 - 16-bit mono audio interface
 - Master mode only
 - 3-port PCM interface
- AC97 Audio Interface supports:
 - Independent channels for stereo PCM In, stereo PCM Out, and mono MIC In
 - 16-bit stereo (2-channel) audio
 - Variable sampling rate AC97 Codec interface (48 kHz and below)
 - AC97 full specification
- SPDIF Interface (TX only) supports:
 - Linear PCM up to 24-bit per sample support
 - 2x24-bit buffers that are alternately filled with data
- I2S Bus Interface supports:
 - Three I2S-bus for audio-codec interface with DMA-based operation
 - Serial and 8/16/24-bit per channel data transfers
 - I2S, MSB-justified, and LSB-justified data format
 - PCM 5.1 channel
 - Various bit clock frequency and codec clock frequency support
 - 16, 24, 32, and 48fs of bit clock frequency
 - 256, 384, 512, and 768fs of codec clock
 - One port for 5.1 channel I2S (in audio subsystem) and two ports for 2-channel I2S
- I2C Bus Interface supports:
 - Eight Multi-Master IIC-Bus
 - Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbps in the standard mode
 - Up to 400 Kbps in the fast mode
- MIPI-Slim bus Interface supports:
 - 6 ports. Each port has 16 entry FIFO with 32-bit width
- UART supports:
 - Four UART with DMA-based or interrupt-based operation
 - 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/ receive
 - Rx/Tx independent 256nbyte FIFO for UART0, 64 byte FIFO for UART1, and 16 byte FIFO for UART2/3/4
 - Programmable baud rate
 - IrDA 1.0 SIR (115.2 Kbps) mode
 - Loop back mode for testing
 - Non-integer clock divides in Baud clock generation

- USB 2.0 Device supports:
 - Complies to USB 2.0 Specification (Revision 1.0a) High-speed up to 480 Mbps
 - On-chip USB transceiver
- USB Host 2.0 supports:
 - With the USB Host 2.0
 - High-speed up to 480 Mbps
 - On-chip USB transceiver
- HS-MMC/SDIO Interface supports:
 - Multimedia Card Protocol version 4.3 compatible (HS-MMC)
 - SD Memory Card Protocol version 2.0 compatible
 - DMA based or interrupt based operation
 - 128 word FIFO for Tx/Rx
 - Four ports HS-MMC or four ports SDIO
- SPI Interface supports:
 - With three Serial Peripheral Interface Protocol version 2.11
 - Rx/Tx independent 64-Word FIFO for SPI0 and 16-Word FIFO for SPI1
 - DMA-based or interrupt-based operation
- GPIO.

1.2.7 System Peripheral

The features of system peripheral are:

- Real Time Clock
 - Full clock features: sec, min, hour, date, day, month, and year
 - 32.768 kHz operation
 - Alarm interrupt
 - Time-tick interrupt
- PLL
 - Four on-chip PLLs and APLL/MPLL/EPLL/VPLL
 - APLL generates ARM core and MSYS clocks
 - MPLL generates a system bus clock and special clocks
 - EPLL generates special clocks
 - VPLL generates clocks for video interface
- Keypad
 - 14 × 8 Key Matrix support
 - Provides internal de-bounce filter
- Timer with Pulse Width Modulation
 - Five channel 32-bit internal timer with interrupt-based operation
 - Three channel 32-bit Timer with PWM
 - Programmable duty cycle, frequency, and polarity
 - Dead-zone generation
 - Supports external clock source
- Multi-Core timer
 - 64-bit global timer with four independent count comparators
 - Two 31-bit local timers
- It can change interrupt interval without stopping reference tick timer DMA:
 - Micro-code programming based DMA
 - The specific instruction set provides flexibility to program DMA transfers
 - Supports linked list DMA function
 - Supports three enhanced built-in DMA with eight channels per DMA, so the total number of channels it supports are 32
 - Supports one Memory-to-memory type optimized DMA and two Peripheral-to-memory type optimized DMA
 - M2M DMA supports up to 16 burst and P2M DMA supports up to 8 burst
- Watch Dog Timer
 - 16-bit watch dog timer

- Thermal Management Unit (TMU)
- Power Management
 - Clock-gating control for components
 - Various low power modes are available, such as Idle, Stop, Deep Stop, Deep Idle, and Sleep modes
 - Wake up sources in sleep mode are:
 - External interrupts
 - RTC alarm
 - Tick timer
 - Key interface
 - Wake up sources of Stop and Deep Stop mode are:
 - MMC
 - Touch screen interface
 - System timer
 - Entire wake up sources of Sleep mode
 - Wake up sources of Deep Idle mode are:
 - 5.1 channel I2S
 - Wake up source of Stop mode

1.3 Conventions

1.3.1 Register RW Conventions

| Symbol | Definition | Description |
|--------|-------------------------|--|
| R | Read Only | The application has permission to read the register field. Writes to read-only fields have no effect. |
| W | Write Only | The application has permission to write in the Register field. |
| RW | Read and Write | The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0. |
| R/WC | Read and Write to clear | The application has permission to read and write in the register field. The application clears this field by writing 1'b1. A register write of 1'b0 has no effect on this field. |
| R/WS | Read and Write to set | The application has permission to read and write in the register field. The application sets this field by writing 1'b1. A register write of 1'b0 has no effect on this field. |

1.3.2 Register Value Conventions

| Expression | Description |
|------------------|--|
| x | Undefined bit |
| X | Undefined multiple bits |
| ? | Undefined but depends on the device, or pin status |
| Device dependent | The value depends on the device |
| Pin value | The value depends on the pin status |

2 Memory Map

2.1 Overview

This section describes the base address of region.

| Base Address | Limit Address | Size | Description |
|--------------|---------------|--------|---|
| 0x0000_0000 | 0x0001_0000 | 64 KB | iROM |
| 0x0200_0000 | 0x0201_0000 | 64 KB | iROM (mirror of 0x0 to 0x10000) |
| 0x0202_0000 | 0x0206_0000 | 256 KB | iRAM |
| 0x0300_0000 | 0x0302_0000 | 128 KB | Data memory or general purpose of Samsung Reconfigurable Processor SRP. |
| 0x0302_0000 | 0x0303_0000 | 64 KB | I-cache or general purpose of SRP. |
| 0x0303_0000 | 0x0303_9000 | 36 KB | Configuration memory (write only) of SRP |
| 0x0381_0000 | 0x0383_0000 | – | AudioSS's SFR region |
| 0x0400_0000 | 0x0500_0000 | 16 MB | Bank0 of Static Read Only Memory Controller (SMC) (16-bit only) |
| 0x0500_0000 | 0x0600_0000 | 16 MB | Bank1 of SMC |
| 0x0600_0000 | 0x0700_0000 | 16 MB | Bank2 of SMC |
| 0x0700_0000 | 0x0800_0000 | 16 MB | Bank3 of SMC |
| 0x0800_0000 | 0x0C00_0000 | 64 MB | Reserved |
| 0x0C00_0000 | 0x0CD0_0000 | – | Reserved |
| 0x0CE0_0000 | 0x0D00_0000 | – | SFR region of Nand Flash Controller (NFCON) |
| 0x1000_0000 | 0x1400_0000 | – | SFR region |
| 0x4000_0000 | 0xA000_0000 | 1.5 GB | Memory of Dynamic Memory Controller (DMC)-0 |
| 0xA000_0000 | 0x0000_0000 | 1.5 GB | Memory of DMC-1 |

2.2 SFR Base Address

This section describes the base address of SFR.

| Base Address | IP |
|--------------|-------------------------------|
| 0x1000_0000 | CHIPID |
| 0x1001_0000 | SYSREG |
| 0x1002_0000 | Power Management Unit (PMU) |
| 0x1003_0000 | CMU_TOP_PART |
| 0x1004_0000 | CMU_CORE_ISP_PART |
| 0x1005_0000 | Multi Core Timer (MCT) |
| 0x1006_0000 | Watch Dog Timer (WDT) |
| 0x1007_0000 | Real Time Clock (RTC) |
| 0x100A_0000 | KEYIF |
| 0x100B_0000 | HDMI_CEC |
| 0x100C_0000 | Thermal Management Unit (TMU) |
| 0x1010_0000 | SECKEY |
| 0x1011_0000 | TZPC0 |
| 0x1012_0000 | TZPC1 |
| 0x1013_0000 | TZPC2 |
| 0x1014_0000 | TZPC3 |
| 0x1015_0000 | TZPC4 |
| 0x1016_0000 | TZPC5 |
| 0x1044_0000 | Int_combiner |
| 0x1048_0000 | GIC_controller |
| 0x1049_0000 | GIC_distributor |
| 0x1060_0000 | DMC0 |
| 0x1061_0000 | DMC1 |
| 0x106A_0000 | PPMU_DMC_L |
| 0x106B_0000 | PPMU_DMC_R |
| 0x106C_0000 | PPMU_CPU |
| 0x1070_0000 | TZASC_LR |
| 0x1071_0000 | TZASC_LW |
| 0x1072_0000 | TZASC_RR |
| 0x1073_0000 | TZASC_RW |
| 0x1080_0000 | G2D_ACP |
| 0x1083_0000 | Security Sub System (SSS) |
| 0x1088_0000 | Coresight |
| 0x1089_0000 | Coresight |

| Base Address | IP |
|--------------|-----------------|
| 0x108B_0000 | Coresight |
| 0x10A4_0000 | SMMUG2D_ACP |
| 0x10A5_0000 | SMMUSSS |
| 0x1100_0000 | GPIO_right |
| 0x1140_0000 | GPIO_left |
| 0x1180_0000 | FIMC0 |
| 0x1181_0000 | FIMC1 |
| 0x1182_0000 | FIMC2 |
| 0x1183_0000 | FIMC3 |
| 0x1184_0000 | JPEG |
| 0x1188_0000 | MIPI_CSI0 |
| 0x1189_0000 | MIPI_CSI1 |
| 0x11A2_0000 | SMMUFIMC0 |
| 0x11A3_0000 | SMMUFIMC1 |
| 0x11A4_0000 | SMMUFIMC2 |
| 0x11A5_0000 | SMMUFIMC3 |
| 0x11A6_0000 | SMMUJPEG |
| 0x11C0_0000 | FIMD0 |
| 0x11C8_0000 | MIPI_DSI0 |
| 0x11E2_0000 | SMMUFIMD0 |
| 0x1200_0000 | FIMC_ISP |
| 0x1201_0000 | FIMC_DRC_TOP |
| 0x1204_0000 | FIMC_FD_TOP |
| 0x1211_0000 | MPWM_ISP |
| 0x1213_0000 | I2C0_ISP |
| 0x1214_0000 | I2C1_ISP |
| 0x1215_0000 | MTCADC_ISP |
| 0x1216_0000 | PWM_ISP |
| 0x1217_0000 | WDT_ISP |
| 0x1218_0000 | MCUCTL_ISP |
| 0x1219_0000 | UART_ISP |
| 0x121A_0000 | SPI0_ISP |
| 0x121B_0000 | SPI1_ISP |
| 0x121E_0000 | GIC_C_ISP |
| 0x121F_0000 | GIC_D_ISP |
| 0x1226_0000 | sysMMU_FIMC-ISP |
| 0x1227_0000 | sysMMU_FIMC-DRC |

| Base Address | IP |
|--------------|----------------------------------|
| 0x122A_0000 | sysMMU_FIMC-FD |
| 0x122B_0000 | sysMMU_ISPCPU |
| 0x1239_0000 | FIMC_LITE0 |
| 0x123A_0000 | FIMC_LITE1 |
| 0x123B_0000 | sysMMU_FIMC-LITE0 |
| 0x123C_0000 | sysMMU_FIMC-LITE1 |
| 0x1248_0000 | USBDEV0 |
| 0x1249_0000 | USBDEV0 |
| 0x124A_0000 | USBDEV0 |
| 0x124B_0000 | USBDEV0 |
| 0x1250_0000 | Transport Stream Interface (TSI) |
| 0x1251_0000 | SDMMC0 |
| 0x1252_0000 | SDMMC1 |
| 0x1253_0000 | SDMMC2 |
| 0x1254_0000 | SDMMC3 |
| 0x1255_0000 | SDMMC4 |
| 0x1256_0000 | MIPI_HSI |
| 0x1257_0000 | SROMC |
| 0x1258_0000 | USBHOST0 |
| 0x1259_0000 | USBHOST1 |
| 0x125B_0000 | USBOTG1 |
| 0x1268_0000 | PDMA0 |
| 0x1269_0000 | PDMA1 |
| 0x126C_0000 | General ADC |
| 0x1281_0000 | Rotator |
| 0x1284_0000 | sMDMA |
| 0x1285_0000 | nsMDMA |
| 0x12A3_0000 | SMMURotator |
| 0x12A4_0000 | SMMUMDMA |
| 0x12C0_0000 | Video Processor (VP) |
| 0x12C1_0000 | Mixer |
| 0x12D0_0000 | HDMI0 |
| 0x12D1_0000 | HDMI1 |
| 0x12D2_0000 | HDMI2 |
| 0x12D3_0000 | HDMI3 |
| 0x12D4_0000 | HDMI4 |
| 0x12D5_0000 | HDMI5 |

| Base Address | IP |
|--------------|---|
| 0x12D6_0000 | HDMI6 |
| 0x12E2_0000 | SMMUTV |
| 0x1300_0000 | 3D Graphic Accelerator (G3D) |
| 0x1322_0000 | PPMU_3D |
| 0x1340_0000 | Multi Format Codec (MFC) |
| 0x1362_0000 | SMMUMFC_L |
| 0x1363_0000 | SMMUMFC_R |
| 0x1366_0000 | PPMU_MFC_L |
| 0x1367_0000 | PPMU_MFC_R |
| 0x1380_0000 | Universal Asynchronous Receiver And Transmitter0 (UART) |
| 0x1381_0000 | UART1 |
| 0x1382_0000 | UART2 |
| 0x1383_0000 | UART3 |
| 0x1384_0000 | UART4 |
| 0x1386_0000 | Inter-Integrated Circuit0 (I2C) |
| 0x1387_0000 | I2C1 |
| 0x1388_0000 | I2C2 |
| 0x1389_0000 | I2C3 |
| 0x138A_0000 | I2C4 |
| 0x138B_0000 | I2C5 |
| 0x138C_0000 | I2C6 |
| 0x138D_0000 | I2C7 |
| 0x138E_0000 | I2CHDMI |
| 0x1392_0000 | Serial Peripheral Interface0 (SPI) |
| 0x1393_0000 | SPI1 |
| 0x1394_0000 | SPI2 |
| 0x1396_0000 | I2S1 |
| 0x1397_0000 | I2S2 |
| 0x1398_0000 | PCM1 |
| 0x1399_0000 | PCM2 |
| 0x139A_0000 | AC97 |
| 0x139B_0000 | SPDIF |
| 0x139D_0000 | PWMTimer |

3 Chip ID

3.1 Overview

The Exynos 4412 includes a Chip ID block for the Software (SW) that sends and receives Advanced Peripheral Bus (APB) interface signals to the bus system.

3.2 Register Description

3.2.1 Register Map Summary

- Base Address: 0x1000_0000

| Register | Offset | Description | Reset Value |
|------------|--------|---|-------------|
| PRO_ID | 0x0000 | Product information (ID, package, and revision) | 0xE441_2XXX |
| PACKAGE_ID | 0x0004 | Package information (POP type and package) | 0xXXXX_XXXX |

3.2.1.1 PRO_ID

- Base Address: 0x1000_0000
- Address = Base Address + 0x0000, Reset Value = 0xE441_2XXX

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|----------------------|---------------------|
| Product ID | [31:12] | R | Product ID | 0x4412 |
| RSVD | [11:10] | R | Reserved | 0x0 |
| Package | [9:8] | R | Package Information | Exynos 4412: 0x2 |
| MainRev | [7:4] | R | Main Revision Number | 0x1 |
| SubRev | [3:0] | R | Sub Revision Number | 0x1 |

NOTE: PRO_ID register[31:0] depends on the e-fuse ROM value. As power on sequence is progressing, it loads the e-fuse ROM values to the registers. It can read the loaded current e-fuse ROM values. An e-fuse ROM has main and sub revision numbers.

3.2.1.2 PACKAGE_ID

- Base Address: 0x1000_0000
- Address = Base Address + 0x0004, Reset Value = 0xXXXX_XXXX

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|--|-------------|
| Package ID | [31:0] | R | Package information (POP type and package) | 0xXXXX_XXXX |

NOTE: PACKAGE_ID register[31:0] depends on the e-fuse ROM value.

4 General Purpose Input/Output (GPIO) Control

This chapter describes the General Purpose Input/Output (GPIO).

4.1 Overview

Exynos 4412 contains 304 multi-functional input/output port pins and 164 memory port pins. There are 37 general port groups and two memory port groups. They are:

- GPA0, GPA1: 14 in/out ports-3xUART with flow control, UART without flow control, and/or 2xI2C
- GPB: 8 in/out ports-2xSPI and/or 2xI2C and/ or IEM
- GPC0, GPC1: 10 in/out ports-2xI2S, and/or 2xPCM, and/or AC97, SPDIF, I2C, and/or SPI
- GPD0, GPD1: 8 in/out ports-PWM, 2xI2C, and/ or LCD I/F, MIPI
- GPM0, GPM1, GPM2, GPM3, GPM4: 35 in/out ports-CAM I/F, and/ or TS I/F, HSI, and/ or Trace I/F
- GPF0, GPF1, GPF2, GPF3: 30 in/out ports-LCD I/F
- GPJ0, GPJ1: 13 in/out ports-CAM I/F
- GPK0, GPK1, GPK2, GPK3: 28 in/out ports-4xMMC (4-bit MMC), and/or 2xMMC (8-bit MMC)), and/or GPS debugging I/F
- GPL0, GPL1: 11 in/out ports-GPS I/F
- GPL2: 8 in/out ports-GPS debugging I/F or Key pad I/F
- GPX0, GPX1, GPX2, GPX3: 32 in/out ports-External wake-up, and/or Key pad I/F

NOTE: These are in ALIVE region.

- GPZ: 7 in/out ports-low Power I2S and/or PCM
- GPY0, GPY1, GPY2: 16 in/out ports-Control signals of EBI (SROM, NF, One NAND)
- GPY3, GPY4, GPY5, GPY6: 32 in/out memory ports-EBI (For more information about EBI configuration, refer to Chapter 5, and 6)
- MP1_0-MP1_9: 78 DRAM1 ports

NOTE: GPIO registers does not control these ports.

- MP2_0-MP2_9: 78 DRAM2 ports

NOTE: GPIO registers does not control these ports.

- ETC0, ETC1, ETC6: 18 in/out ETC ports-JTAG, SLIMBUS, RESET, CLOCK
- ETC7, ETC8 : 4 clock port for C2C

Warning: When you do not use or connect port to an input pin without Pull-up/Pull-down then do not leave a port in Input Pull-up/Pull-down disable state. It may cause unexpected state and leakage current. Disable Pull-up/Pull-down when you use port as output function.

4.2 Features

The features of GPIO include:

- Controls 172 External Interrupts
- Controls 32 External Wake-up Interrupts
- 252 multi-functional input/output ports
- Controls pin states in Sleep Mode except GPX0, GPX1, GPX2, and GPX3 (GPX* pins are alive-pads)

4.2.1 Input/Output Description

This section includes:

- General Purpose Input/Output Block Diagram
- Register Description

4.2.1.1 General Purpose Input/Output Block Diagram

GPIO consists of two parts,

- alive-part
- off-part

In Alive-part, you should supply power on sleep mode, but in off-part, it is not same. Therefore, registers in alive-part keep their values during sleep mode.

[Figure 4-1](#) illustrates the block diagram of GPIO.

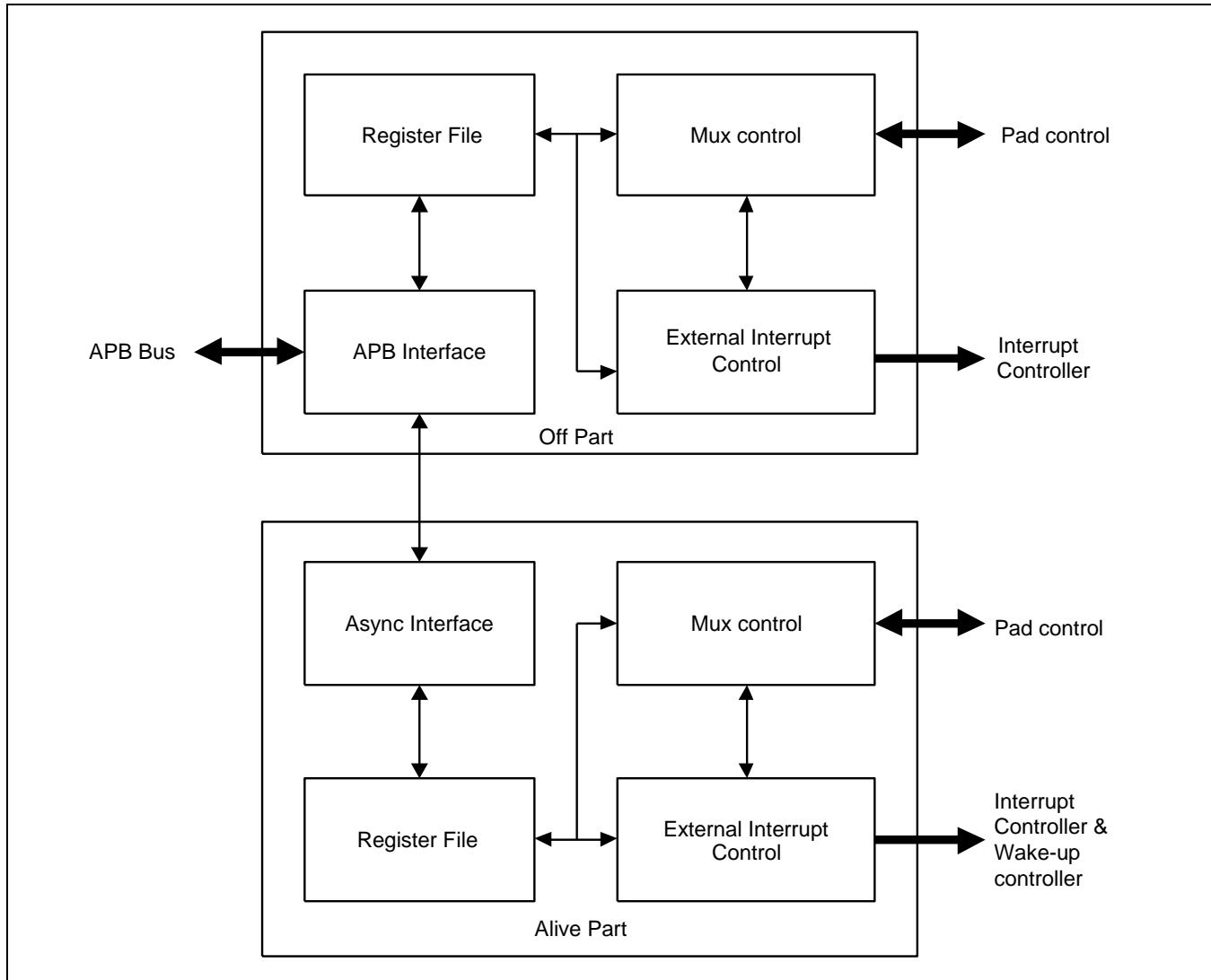


Figure 4-1 GPIO Block Diagram

4.3 Register Description

4.3.1 Registers Summary

- Base Address: 0x1140_0000 GPIO模块的地址

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| GPA0CON | 0x0000 | Port group GPA0 configuration register | 0x0000_0000 |
| GPA0DAT | 0x0004 | Port group GPA0 data register | 0x00 |
| GPA0PUD | 0x0008 | Port group GPA0 pull-up/pull-down register | 0x5555 |
| GPA0DRV | 0x000C | Port group GPA0 drive strength control register | 0x00_0000 |
| GPA0CONPDN | 0x0010 | Port group GPA0 power down mode configuration register | 0x0000 |
| GPA0PUDPDN | 0x0014 | Port group GPA0 power down mode pull-up/pull-down register | 0x0000 |
| GPA1CON | 0x0020 | Port group GPA1 configuration register | 0x0000_0000 |
| GPA1DAT | 0x0024 | Port group GPA1 data register | 0x00 |
| GPA1PUD | 0x0028 | Port group GPA1 pull-up/pull-down register | 0x0555 |
| GPA1DRV | 0x002C | Port group GPA1 drive strength control register | 0x00_0000 |
| GPA1CONPDN | 0x0030 | Port group GPA1 power down mode configuration register | 0x0000 |
| GPA1PUDPDN | 0x0034 | Port group GPA1 power down mode pull-up/pull-down register | 0x0000 |
| GPBCON | 0x0040 | Port group GPB configuration register | 0x0000_0000 |
| GPBDAT | 0x0044 | Port group GPB data register | 0x00 |
| GPBPUD | 0x0048 | Port group GPB pull-up/pull-down register | 0x5555 |
| GPBDRV | 0x004C | Port group GPB drive strength control register | 0x00_0000 |
| GPBCONPDN | 0x0050 | Port group GPB power down mode configuration register | 0x0000 |
| GPBPUDPDN | 0x0054 | Port group GPB power down mode pull-up/pull-down register | 0x0000 |
| GPC0CON | 0x0060 | Port group GPC0 configuration register | 0x0000_0000 |
| GPC0DAT | 0x0064 | Port group GPC0 data register | 0x00 |
| GPC0PUD | 0x0068 | Port group GPC0 Pull-up/pull-down register | 0x0155 |
| GPC0DRV | 0x006C | Port group GPC0 drive strength control register | 0x00_0000 |
| GPC0CONPDN | 0x0070 | Port group GPC0 power down mode configuration register | 0x0000 |
| GPC0PUDPDN | 0x0074 | Port group GPC0 power down mode pull-up/pull-down register | 0x0000 |
| GPC1CON | 0x0080 | Port group GPC1 configuration register | 0x0000_0000 |
| GPC1DAT | 0x0084 | Port group GPC1 data register | 0x00 |
| GPC1PUD | 0x0088 | Port group GPC1 pull-up/pull-down register | 0x0155 |
| GPC1DRV | 0x008C | Port group GPC1 drive strength control register | 0x00_0000 |
| GPC1CONPDN | 0x0090 | Port group GPC1 power down mode configuration register | 0x0000 |
| GPC1PUDPDN | 0x0094 | Port group GPC1 power down mode pull-up/pull-down | 0x0000 |

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| | | register | |
| GPD0CON | 0x00A0 | Port group GPD0 configuration register | 0x0000_0000 |
| GPD0DAT | 0x00A4 | Port group GPD0 data register | 0x00 |
| GPD0PUD | 0x00A8 | Port group GPD0 pull-up/pull-down register | 0x0055 |
| GPD0DRV | 0x00AC | Port group GPD0 drive strength control register | 0x00_0000 |
| GPD0CONPDN | 0x00B0 | Port group GPD0 power down mode configuration register | 0x0000 |
| GPD0PUDPDN | 0x00B4 | Port group GPD0 power down mode pull-up/pull-down register | 0x0000 |
| GPD1CON | 0x00C0 | Port group GPD1 configuration register | 0x0000_0000 |
| GPD1DAT | 0x00C4 | Port group GPD1 data register | 0x00 |
| GPD1PUD | 0x00C8 | Port group GPD1 Pull-up/pull-down register | 0x0055 |
| GPD1DRV | 0x00CC | Port group GPD1 drive strength control register | 0x00_0000 |
| GPD1CONPDN | 0x00D0 | Port group GPD1 power down mode configuration register | 0x0000 |
| GPD1PUDPDN | 0x00D4 | Port group GPD1 power down mode pull-up/pull-down register | 0x0000 |
| GPF0CON | 0x0180 | Port group GPF0 configuration register | 0x0000_0000 |
| GPF0DAT | 0x0184 | Port group GPF0 data register | 0x00 |
| GPF0PUD | 0x0188 | Port group GPF0 pull-up/pull-down register | 0x5555 |
| GPF0DRV | 0x018C | Port group GPF0 drive strength control register | 0x00_0000 |
| GPF0CONPDN | 0x0190 | Port group GPF0 power down mode configuration register | 0x0000 |
| GPF0PUDPDN | 0x0194 | Port group GPF0 power down mode pull-up/pull-down register | 0x0000 |
| GPF1CON | 0x01A0 | Port group GPF1 configuration register | 0x0000_0000 |
| GPF1DAT | 0x01A4 | Port group GPF1 data register | 0x00 |
| GPF1PUD | 0x01A8 | Port group GPF1 pull-up/pull-down register | 0x5555 |
| GPF1DRV | 0x01AC | Port group GPF1 drive strength control register | 0x00_0000 |
| GPF1CONPDN | 0x01B0 | Port group GPF1 power down mode configuration register | 0x0000 |
| GPF1PUDPDN | 0x01B4 | Port group GPF1 power down mode pull-up/pull-down register | 0x0000 |
| GPF2CON | 0x01C0 | Port group GPF2 configuration register | 0x0000_0000 |
| GPF2DAT | 0x01C4 | Port group GPF2 data register | 0x00 |
| GPF2PUD | 0x01C8 | Port group GPF2 pull-up/pull-down register | 0x5555 |
| GPF2DRV | 0x01CC | Port group GPF2 drive strength control register | 0x00_0000 |
| GPF2CONPDN | 0x01D0 | Port group GPF2 power down mode configuration register | 0x0000 |
| GPF2PUDPDN | 0x01D4 | Port group GPF2 power down mode pull-up/pull-down register | 0x0000 |
| GPF3CON | 0x01E0 | Port group GPF3 configuration register | 0x0000_0000 |
| GPF3DAT | 0x01E4 | Port group GPF3 data register | 0x00 |

| Register | Offset | Description | Reset Value |
|------------------|--------|---|-------------|
| GPF3PUD | 0x01E8 | Port group GPF3 pull-up/pull-down register | 0x0555 |
| GPF3DRV | 0x01EC | Port group GPF3 drive strength control register | 0x00_0000 |
| GPF3CONPDN | 0x01F0 | Port group GPF3 power down mode configuration register | 0x0000 |
| GPF3PUDPDN | 0x01F4 | Port group GPF3 power down mode pull-up/pull-down register | 0x0000 |
| ETC1PUD | 0x0228 | Port group ETC1 pull-up/pull-down register | 0x0005 |
| ETC1DRV | 0x022C | Port group ETC1 drive strength control register | 0x00_0000 |
| GPJ0CON | 0x0240 | Port group GPJ0 configuration register | 0x0000_0000 |
| GPJ0DAT | 0x0244 | Port group GPJ0 data register | 0x00 |
| GPJ0PUD | 0x0248 | Port group GPJ0 pull-up/pull-down register | 0x5555 |
| GPJ0DRV | 0x024C | Port group GPJ0 drive strength control register | 0x00_0000 |
| GPJ0CONPDN | 0x0250 | Port group GPJ0 power down mode configuration register | 0x0000 |
| GPJ0PUDPDN | 0x0254 | Port group GPJ0 power down mode pull-up/pull-down register | 0x0000 |
| GPJ1CON | 0x0260 | Port group GPJ1 configuration register | 0x0000_0000 |
| GPJ1DAT | 0x0264 | Port group GPJ1 data register | 0x00 |
| GPJ1PUD | 0x0268 | Port group GPJ1 pull-up/pull-down register | 0x0155 |
| GPJ1DRV | 0x026C | Port group GPJ1 drive strength control register | 0x00_0000 |
| GPJ1CONPDN | 0x0270 | Port group GPJ1 power down mode configuration register | 0x0000 |
| GPJ1PUDPDN | 0x0274 | Port group GPJ1 power down mode pull-up/pull-down register | 0x0000 |
| EXT_INT1_CON | 0x0700 | External interrupt EXT_INT1 configuration register | 0x0000_0000 |
| EXT_INT2_CON | 0x0704 | External interrupt EXT_INT2 configuration register | 0x0000_0000 |
| EXT_INT3_CON | 0x0708 | External interrupt EXT_INT3 configuration register | 0x0000_0000 |
| EXT_INT4_CON | 0x070C | External interrupt EXT_INT4 configuration register | 0x0000_0000 |
| EXT_INT5_CON | 0x0710 | External interrupt EXT_INT5 configuration register | 0x0000_0000 |
| EXT_INT6_CON | 0x0714 | External interrupt EXT_INT6 configuration register | 0x0000_0000 |
| EXT_INT7_CON | 0x0718 | External interrupt EXT_INT7 configuration register | 0x0000_0000 |
| EXT_INT13_CON | 0x0730 | External interrupt EXT_INT13 configuration register | 0x0000_0000 |
| EXT_INT14_CON | 0x0734 | External interrupt EXT_INT14 configuration register | 0x0000_0000 |
| EXT_INT15_CON | 0x0738 | External interrupt EXT_INT15 configuration register | 0x0000_0000 |
| EXT_INT16_CON | 0x073C | External interrupt EXT_INT16 configuration register | 0x0000_0000 |
| EXT_INT21_CON | 0x0740 | External interrupt EXT_INT21 configuration register | 0x0000_0000 |
| EXT_INT22_CON | 0x0744 | External interrupt EXT_INT22 configuration register | 0x0000_0000 |
| EXT_INT1_FLTCON0 | 0x0800 | External interrupt EXT_INT1 filter configuration register 0 | 0x0000_0000 |
| EXT_INT1_FLTCON1 | 0x0804 | External interrupt EXT_INT1 filter configuration register 1 | 0x0000_0000 |
| EXT_INT2_FLTCON0 | 0x0808 | External interrupt EXT_INT2 filter configuration register 0 | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------|--------|--|-------------|
| EXT_INT2_FLTCON1 | 0x080C | External interrupt EXT_INT2 filter configuration register 1 | 0x0000_0000 |
| EXT_INT3_FLTCON0 | 0x0810 | External interrupt EXT_INT3 filter configuration register 0 | 0x0000_0000 |
| EXT_INT3_FLTCON1 | 0x0814 | External interrupt EXT_INT3 filter configuration register 1 | 0x0000_0000 |
| EXT_INT4_FLTCON0 | 0x0818 | External interrupt EXT_INT4 filter configuration register 0 | 0x0000_0000 |
| EXT_INT4_FLTCON1 | 0x081C | External interrupt EXT_INT4 filter configuration register 1 | 0x0000_0000 |
| EXT_INT5_FLTCON0 | 0x0820 | External interrupt EXT_INT5 filter configuration register 0 | 0x0000_0000 |
| EXT_INT5_FLTCON1 | 0x0824 | External interrupt EXT_INT5 filter configuration register 1 | 0x0000_0000 |
| EXT_INT6_FLTCON0 | 0x0828 | External interrupt EXT_INT6 filter configuration register 0 | 0x0000_0000 |
| EXT_INT6_FLTCON1 | 0x082C | External interrupt EXT_INT6 filter configuration register 1 | 0x0000_0000 |
| EXT_INT7_FLTCON0 | 0x0830 | External interrupt EXT_INT7 filter configuration register 0 | 0x0000_0000 |
| EXT_INT7_FLTCON1 | 0x0834 | External interrupt EXT_INT7 filter configuration register 1 | 0x0000_0000 |
| EXT_INT13_FLTCON0 | 0x0860 | External interrupt EXT_INT13 filter configuration register 0 | 0x0000_0000 |
| EXT_INT13_FLTCON1 | 0x0864 | External interrupt EXT_INT13 filter configuration register 1 | 0x0000_0000 |
| EXT_INT14_FLTCON0 | 0x0868 | External interrupt EXT_INT14 filter configuration register 0 | 0x0000_0000 |
| EXT_INT14_FLTCON1 | 0x086C | External interrupt EXT_INT14 filter configuration register 1 | 0x0000_0000 |
| EXT_INT15_FLTCON0 | 0x0870 | External interrupt EXT_INT15 filter configuration register 0 | 0x0000_0000 |
| EXT_INT15_FLTCON1 | 0x0874 | External interrupt EXT_INT15 filter configuration register 1 | 0x0000_0000 |
| EXT_INT16_FLTCON0 | 0x0878 | External interrupt EXT_INT16 filter configuration register 0 | 0x0000_0000 |
| EXT_INT16_FLTCON1 | 0x087C | External interrupt EXT_INT16 filter configuration register 1 | 0x0000_0000 |
| EXT_INT21_FLTCON0 | 0x0880 | External interrupt EXT_INT21 filter configuration register 0 | 0x0000_0000 |
| EXT_INT21_FLTCON1 | 0x0884 | External interrupt EXT_INT21 filter configuration register 1 | 0x0000_0000 |
| EXT_INT22_FLTCON0 | 0x0888 | External interrupt EXT_INT22 filter configuration register 0 | 0x0000_0000 |
| EXT_INT22_FLTCON1 | 0x088C | External interrupt EXT_INT22 filter configuration register 1 | 0x0000_0000 |
| EXT_INT1_MASK | 0x0900 | External interrupt EXT_INT1 mask register | 0x0000_00FF |
| EXT_INT2_MASK | 0x0904 | External interrupt EXT_INT2 mask register | 0x0000_003F |
| EXT_INT3_MASK | 0x0908 | External interrupt EXT_INT3 mask register | 0x0000_00FF |
| EXT_INT4_MASK | 0x090C | External interrupt EXT_INT4 mask register | 0x0000_001F |
| EXT_INT5_MASK | 0x0910 | External interrupt EXT_INT5 mask register | 0x0000_001F |
| EXT_INT6_MASK | 0x0914 | External interrupt EXT_INT6 mask register | 0x0000_000F |
| EXT_INT7_MASK | 0x0918 | External interrupt EXT_INT7 mask register | 0x0000_000F |
| EXT_INT13_MASK | 0x0930 | External interrupt EXT_INT13 mask register | 0x0000_00FF |
| EXT_INT14_MASK | 0x0934 | External interrupt EXT_INT14 mask register | 0x0000_00FF |
| EXT_INT15_MASK | 0x0938 | External interrupt EXT_INT15 mask register | 0x0000_00FF |
| EXT_INT16_MASK | 0x093C | External interrupt EXT_INT16 mask register | 0x0000_003F |
| EXT_INT21_MASK | 0x0940 | External interrupt EXT_INT21 mask register | 0x0000_00FF |
| EXT_INT22_MASK | 0x0944 | External interrupt EXT_INT22 mask register | 0x0000_001F |
| EXT_INT1_PEND | 0xA00 | External interrupt EXT_INT1 pending register | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------------|--------|--|-------------|
| EXT_INT2_PEND | 0x0A04 | External interrupt EXT_INT2 pending register | 0x0000_0000 |
| EXT_INT3_PEND | 0x0A08 | External interrupt EXT_INT3 pending register | 0x0000_0000 |
| EXT_INT4_PEND | 0x0A0C | External interrupt EXT_INT4 pending register | 0x0000_0000 |
| EXT_INT5_PEND | 0x0A10 | External interrupt EXT_INT5 pending register | 0x0000_0000 |
| EXT_INT6_PEND | 0x0A14 | External interrupt EXT_INT6 pending register | 0x0000_0000 |
| EXT_INT7_PEND | 0x0A18 | External interrupt EXT_INT7 pending register | 0x0000_0000 |
| EXT_INT13_PEND | 0x0A30 | External interrupt EXT_INT13 pending register | 0x0000_0000 |
| EXT_INT14_PEND | 0x0A34 | External interrupt EXT_INT14 pending register | 0x0000_0000 |
| EXT_INT15_PEND | 0x0A38 | External interrupt EXT_INT15 pending register | 0x0000_0000 |
| EXT_INT16_PEND | 0x0A3C | External interrupt EXT_INT16 pending register | 0x0000_0000 |
| EXT_INT21_PEND | 0x0A40 | External interrupt EXT_INT21 pending register | 0x0000_0000 |
| EXT_INT22_PEND | 0x0A44 | External interrupt EXT_INT22 pending register | 0x0000_0000 |
| EXT_INT_SERVICE_XB | 0xB08 | Current service register | 0x0000_0000 |
| EXT_INT_SERVICE_PEND_XB | 0xB0C | Current service pending register | 0x0000_0000 |
| EXT_INT_GRPFIXPRI_XB | 0xB10 | External interrupt group fixed priority control register | 0x0000_0000 |
| EXT_INT1_FIXPRI | 0xB14 | External interrupt 1 fixed priority control register | 0x0000_0000 |
| EXT_INT2_FIXPRI | 0xB18 | External interrupt 2 fixed priority control register | 0x0000_0000 |
| EXT_INT3_FIXPRI | 0xB1C | External interrupt 3 fixed priority control register | 0x0000_0000 |
| EXT_INT4_FIXPRI | 0xB20 | External interrupt 4 fixed priority control register | 0x0000_0000 |
| EXT_INT5_FIXPRI | 0xB24 | External interrupt 5 fixed priority control register | 0x0000_0000 |
| EXT_INT6_FIXPRI | 0xB28 | External interrupt 6 fixed priority control register | 0x0000_0000 |
| EXT_INT7_FIXPRI | 0xB2C | External interrupt 7 fixed priority control register | 0x0000_0000 |
| EXT_INT13_FIXPRI | 0xB44 | External interrupt 13 fixed priority control register | 0x0000_0000 |
| EXT_INT14_FIXPRI | 0xB48 | External interrupt 14 fixed priority control register | 0x0000_0000 |
| EXT_INT15_FIXPRI | 0xB4C | External interrupt 15 fixed priority control register | 0x0000_0000 |
| EXT_INT16_FIXPRI | 0xB50 | External interrupt 16 fixed priority control register | 0x0000_0000 |
| EXT_INT21_FIXPRI | 0xB54 | External interrupt 21 fixed priority control register | 0x0000_0000 |
| EXT_INT22_FIXPRI | 0xB58 | External interrupt 22 fixed priority control register | 0x0000_0000 |
| PDNEN | 0xF80 | Power down mode pad configure register | 0x00 |

- Base Address: 0x1100_0000

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| GPK0CON | 0x0040 | Port group GPK0 configuration register | 0x0000_0000 |
| GPK0DAT | 0x0044 | Port group GPK0 data register | 0x00 |
| GPK0PUD | 0x0048 | Port group GPK0 pull-up/pull-down register | 0x1555 |
| GPK0DRV | 0x004C | Port group GPK0 drive strength control register | 0x00_2AAA |
| GPK0CONPDN | 0x0050 | Port group GPK0 power down mode configuration register | 0x0000 |
| GPK0PUDPDN | 0x0054 | Port group GPK0 power down mode pull-up/pull-down register | 0x0000 |
| GPK1CON | 0x0060 | Port group GPK1 configuration register | 0x0000_0000 |
| GPK1DAT | 0x0064 | Port group GPK1 data register | 0x00 |
| GPK1PUD | 0x0068 | Port group GPK1 pull-up/pull-down register | 0x1555 |
| GPK1DRV | 0x006C | Port group GPK1 drive strength control register | 0x00_0000 |
| GPK1CONPDN | 0x0070 | Port group GPK1 power down mode configuration register | 0x0000 |
| GPK1PUDPDN | 0x0074 | Port group GPK1 power down mode pull-up/pull-down register | 0x0000 |
| GPK2CON | 0x0080 | Port group GPK2 configuration register | 0x0000_0000 |
| GPK2DAT | 0x0084 | Port group GPK2 data register | 0x00 |
| GPK2PUD | 0x0088 | Port group GPK2 pull-up/pull-down register | 0x1555 |
| GPK2DRV | 0x008C | Port group GPK2 drive strength control register | 0x00_0000 |
| GPK2CONPDN | 0x0090 | Port group GPK2 power down mode configuration register | 0x0000 |
| GPK2PUDPDN | 0x0094 | Port group GPK2 power down mode pull-up/pull-down register | 0x0000 |
| GPK3CON | 0x00A0 | Port group GPK3 configuration register | 0x0000_0000 |
| GPK3DAT | 0x00A4 | Port group GPK3 data register | 0x00 |
| GPK3PUD | 0x00A8 | Port group GPK3 pull-up/pull-down register | 0x1555 |
| GPK3DRV | 0x00AC | Port group GPK3 drive strength control register | 0x00_0000 |
| GPK3CONPDN | 0x00B0 | Port group GPK3 power down mode configuration register | 0x0000 |
| GPK3PUDPDN | 0x00B4 | Port group GPK3 power down mode pull-up/pull-down register | 0x0000 |
| GPL0CON | 0x00C0 | Port group GPL0 configuration register | 0x0000_0000 |
| GPL0DAT | 0x00C4 | Port group GPL0 data register | 0x00 |
| GPL0PUD | 0x00C8 | Port group GPL0 pull-up/pull-down register | 0x1555 |
| GPL0DRV | 0x00CC | Port group GPL0 drive strength control register | 0x00_0000 |
| GPL0CONPDN | 0x00D0 | Port group GPL0 power down mode configuration register | 0x0000 |
| GPL0PUDPDN | 0x00D4 | Port group GPL0 power down mode pull-up/pull-down register | 0x0000 |
| GPL1CON | 0x00E0 | Port group GPL1 configuration register | 0x0000_0000 |
| GPL1DAT | 0x00E4 | Port group GPL1 data register | 0x00 |

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| GPL1PUD | 0x00E8 | Port group GPL1 pull-up/pull-down register | 0x0005 |
| GPL1DRV | 0x00EC | Port group GPL1 drive strength control register | 0x00_0000 |
| GPL1CONPDN | 0x00F0 | Port group GPL1 power down mode configuration register | 0x0000 |
| GPL1PUDPDN | 0x00F4 | Port group GPL1 power down mode pull-up/pull-down register | 0x0000 |
| GPL2CON | 0x0100 | Port group GPL2 configuration register | 0x0000_0000 |
| GPL2DAT | 0x0104 | Port group GPL2 data register | 0x00 |
| GPL2PUD | 0x0108 | Port group GPL2 pull-up/pull-down register | 0x5555 |
| GPL2DRV | 0x010C | Port group GPL2 drive strength control register | 0x00_0000 |
| GPL2CONPDN | 0x0110 | Port group GPL2 power down mode configuration register | 0x0000 |
| GPL2PUDPDN | 0x0114 | Port group GPL2 power down mode pull-up/pull-down register | 0x0000 |
| GPY0CON | 0x0120 | Port group GPY0 configuration register | 0x0000_0000 |
| GPY0DAT | 0x0124 | Port group GPY0 data register | 0x00 |
| GPY0PUD | 0x0128 | Port group GPY0 pull-up/pull-down register | 0x0FFF |
| GPY0DRV | 0x012C | Port group GPY0 drive strength control register | 0x00_0AAA |
| GPY0CONPDN | 0x0130 | Port group GPY0 power down mode configuration register | 0x0000 |
| GPY0PUDPDN | 0x0134 | Port group GPY0 power down mode pull-up/pull-down register | 0x0000 |
| GPY1CON | 0x0140 | Port group GPY1 configuration register | 0x0000_0000 |
| GPY1DAT | 0x0144 | Port group GPY1 data register | 0x00 |
| GPY1PUD | 0x0148 | Port group GPY1 pull-up/pull-down register | 0x00FF |
| GPY1DRV | 0x014C | Port group GPY1 drive strength control register | 0x00_00AA |
| GPY1CONPDN | 0x0150 | Port group GPY1 power down mode configuration register | 0x0000 |
| GPY1PUDPDN | 0x0154 | Port group GPY1 power down mode pull-up/pull-down register | 0x0000 |
| GPY2CON | 0x0160 | Port group GPY2 configuration register | 0x0000_0000 |
| GPY2DAT | 0x0164 | Port group GPY2 data register | 0x00 |
| GPY2PUD | 0x0168 | Port group GPY2 pull-up/pull-down register | 0x0FFF |
| GPY2DRV | 0x016C | Port group GPY2 drive strength control register | 0x00_0AAA |
| GPY2CONPDN | 0x0170 | Port group GPY2 power down mode configuration register | 0x0000 |
| GPY2PUDPDN | 0x0174 | Port group GPY2 power down mode pull-up/pull-down register | 0x0000 |
| GPY3CON | 0x0180 | Port group GPY3 configuration register | 0x0000_0000 |
| GPY3DAT | 0x0184 | Port group GPY3 data register | 0x00 |
| GPY3PUD | 0x0188 | Port group GPY3 pull-up/pull-down register | 0x5555 |
| GPY3DRV | 0x018C | Port group GPY3 drive strength control register | 0x00_AAAA |
| GPY3CONPDN | 0x0190 | Port group GPY3 power down mode configuration register | 0x0000 |

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| GPY3PUDPDN | 0x0194 | Port group GPY3 power down mode pull-up/pull-down register | 0x0000 |
| GPY4CON | 0x01A0 | Port group GPY4 configuration register | 0x0000_0000 |
| GPY4DAT | 0x01A4 | Port group GPY4 data register | 0x00 |
| GPY4PUD | 0x01A8 | Port group GPY4 pull-up/pull-down register | 0x5555 |
| GPY4DRV | 0x01AC | Port group GPY4 drive strength control register | 0x00_AAAA |
| GPY4CONPDN | 0x01B0 | Port group GPY4 power down mode configuration register | 0x0000 |
| GPY4PUDPDN | 0x01B4 | Port group GPY4 power down mode pull-up/pull-down register | 0x0000 |
| GPY5CON | 0x01C0 | Port group GPY5 configuration register | 0x0000_0000 |
| GPY5DAT | 0x01C4 | Port group GPY5 data register | 0x00 |
| GPY5PUD | 0x01C8 | Port group GPY5 pull-up/pull-down register | 0x5555 |
| GPY5DRV | 0x01CC | Port group GPY5 drive strength control register | 0x00_AAAA |
| GPY5CONPDN | 0x01D0 | Port group GPY5 power down mode configuration register | 0x0000 |
| GPY5PUDPDN | 0x01D4 | Port group GPY5 power down mode pull-up/pull-down register | 0x0000 |
| GPY6CON | 0x01E0 | Port group GPY6 configuration register | 0x0000_0000 |
| GPY6DAT | 0x01E4 | Port group GPY6 data register | 0x00 |
| GPY6PUD | 0x01E8 | Port group GPY6 pull-up/pull-down register | 0x5555 |
| GPY6DRV | 0x01EC | Port group GPY6 drive strength control register | 0x00_AAAA |
| GPY6CONPDN | 0x01F0 | Port group GPY6 power down mode configuration register | 0x0000 |
| GPY6PUDPDN | 0x01F4 | Port group GPY6 power down mode pull-up/pull-down register | 0x0000 |
| ETC0PUD | 0x0208 | Port group ETC0 pull-up/pull-down register | 0x0400 |
| ETC0DRV | 0x020C | Port group ETC0 drive strength control register | 0x00_0000 |
| ETC6PUD | 0x0228 | Port group ETC6 pull-up/pull-down register | 0xC000 |
| ETC6DRV | 0x022C | Port group ETC6 drive strength control register | 0x00_0000 |
| GPM0CON | 0x0260 | Port group GPM0 configuration register | 0x0000_0000 |
| GPM0DAT | 0x0264 | Port group GPM0 data register | 0x00 |
| GPM0PUD | 0x0268 | Port group GPM0 pull-up/pull-down register | 0x5555 |
| GPM0DRV | 0x026C | Port group GPM0 drive strength control register | 0x00_0000 |
| GPM0CONPDN | 0x0270 | Port group GPM0 power down mode configuration register | 0x0000 |
| GPM0PUDPDN | 0x0274 | Port group GPM0 power down mode pull-up/pull-down register | 0x0000 |
| GPM1CON | 0x0280 | Port group GPM1 configuration register | 0x0000_0000 |
| GPM1DAT | 0x0284 | Port group GPM1 data register | 0x00 |
| GPM1PUD | 0x0288 | Port group GPM1 pull-up/pull-down register | 0x1555 |
| GPM1DRV | 0x028C | Port group GPM1 drive strength control register | 0x00_0000 |

| Register | Offset | Description | Reset Value |
|-------------------|--------|--|-------------|
| GPM1CONPDN | 0x0290 | Port group GPM1 power down mode configuration register | 0x0000 |
| GPM1PUDPDN | 0x0294 | Port group GPM1 power down mode pull-up/pull-down register | 0x0000 |
| GPM2CON | 0x02A0 | Port group GPM2 configuration register | 0x0000_0000 |
| GPM2DAT | 0x02A4 | Port group GPM2 data register | 0x00 |
| GPM2PUD | 0x02A8 | Port group GPM2 pull-up/pull-down register | 0x0155 |
| GPM2DRV | 0x02AC | Port group GPM2 drive strength control register | 0x00_0000 |
| GPM2CONPDN | 0x02B0 | Port group GPM2 power down mode configuration register | 0x0000 |
| GPM2PUDPDN | 0x02B4 | Port group GPM2 power down mode pull-up/pull-down register | 0x0000 |
| GPM3CON | 0x02C0 | Port group GPM3 configuration register | 0x0000_0000 |
| GPM3DAT | 0x02C4 | Port group GPM3 data register | 0x00 |
| GPM3PUD | 0x02C8 | Port group GPM3 pull-up/pull-down register | 0x5555 |
| GPM3DRV | 0x02CC | Port group GPM3 drive strength control register | 0x00_0000 |
| GPM3CONPDN | 0x02D0 | Port group GPM3 power down mode configuration register | 0x0000 |
| GPM3PUDPDN | 0x02D4 | Port group GPM3 power down mode pull-up/pull-down register | 0x0000 |
| GPM4CON | 0x02E0 | Port group GPM4 configuration register | 0x0000_0000 |
| GPM4DAT | 0x02E4 | Port group GPM4 data register | 0x00 |
| GPM4PUD | 0x02E8 | Port group GPM4 pull-up/pull-down register | 0x5555 |
| GPM4DRV | 0x02EC | Port group GPM4 drive strength control register | 0x00_0000 |
| GPM4CONPDN | 0x02F0 | Port group GPM4 power down mode configuration register | 0x0000 |
| GPM4PUDPDN | 0x02F4 | Port group GPM4 power down mode pull-up/pull-down register | 0x0000 |
| EXT_INT23_CON | 0x0708 | External interrupt EXT_INT23 configuration register | 0x0000_0000 |
| EXT_INT24_CON | 0x070C | External interrupt EXT_INT24 configuration register | 0x0000_0000 |
| EXT_INT25_CON | 0x0710 | External interrupt EXT_INT25 configuration register | 0x0000_0000 |
| EXT_INT26_CON | 0x0714 | External interrupt EXT_INT26 configuration register | 0x0000_0000 |
| EXT_INT27_CON | 0x0718 | External interrupt EXT_INT27 configuration register | 0x0000_0000 |
| EXT_INT28_CON | 0x071C | External interrupt EXT_INT28 configuration register | 0x0000_0000 |
| EXT_INT29_CON | 0x0720 | External interrupt EXT_INT29 configuration register | 0x0000_0000 |
| EXT_INT8_CON | 0x0724 | External interrupt EXT_INT8 configuration register | 0x0000_0000 |
| EXT_INT9_CON | 0x0728 | External interrupt EXT_INT9 configuration register | 0x0000_0000 |
| EXT_INT10_CON | 0x072C | External interrupt EXT_INT10 configuration register | 0x0000_0000 |
| EXT_INT11_CON | 0x0730 | External interrupt EXT_INT11 configuration register | 0x0000_0000 |
| EXT_INT12_CON | 0x0734 | External interrupt EXT_INT12 configuration register | 0x0000_0000 |
| EXT_INT23_FLTCON0 | 0x0810 | External interrupt EXT_INT23 filter configuration register 0 | 0x0000_0000 |
| EXT_INT23_FLTCON1 | 0x0814 | External interrupt EXT_INT23 filter configuration register 1 | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------|--------|--|-------------|
| EXT_INT24_FLTCON0 | 0x0818 | External interrupt EXT_INT24 filter configuration register 0 | 0x0000_0000 |
| EXT_INT24_FLTCON1 | 0x081C | External interrupt EXT_INT24 filter configuration register 1 | 0x0000_0000 |
| EXT_INT25_FLTCON0 | 0x0820 | External interrupt EXT_INT25 filter configuration register 0 | 0x0000_0000 |
| EXT_INT25_FLTCON1 | 0x0824 | External interrupt EXT_INT25 filter configuration register 1 | 0x0000_0000 |
| EXT_INT26_FLTCON0 | 0x0828 | External interrupt EXT_INT26 filter configuration register 0 | 0x0000_0000 |
| EXT_INT26_FLTCON1 | 0x082C | External interrupt EXT_INT26 filter configuration register 1 | 0x0000_0000 |
| EXT_INT27_FLTCON0 | 0x0830 | External interrupt EXT_INT27 filter configuration register 0 | 0x0000_0000 |
| EXT_INT27_FLTCON1 | 0x0834 | External interrupt EXT_INT27 filter configuration register 1 | 0x0000_0000 |
| EXT_INT28_FLTCON0 | 0x0838 | External interrupt EXT_INT28 filter configuration register 0 | 0x0000_0000 |
| EXT_INT28_FLTCON1 | 0x083C | External interrupt EXT_INT28 filter configuration register 1 | 0x0000_0000 |
| EXT_INT29_FLTCON0 | 0x0840 | External interrupt EXT_INT29 filter configuration register 0 | 0x0000_0000 |
| EXT_INT29_FLTCON1 | 0x0844 | External interrupt EXT_int29 filter configuration register 1 | 0x0000_0000 |
| EXT_INT8_FLTCON0 | 0x0848 | External interrupt EXT_INT8 filter configuration register 0 | 0x0000_0000 |
| EXT_INT8_FLTCON1 | 0x084C | External interrupt EXT_INT8 filter configuration register 1 | 0x0000_0000 |
| EXT_INT9_FLTCON0 | 0x0850 | External interrupt EXT_INT9 filter configuration register 0 | 0x0000_0000 |
| EXT_INT9_FLTCON1 | 0x0854 | External interrupt EXT_INT9 filter configuration register 1 | 0x0000_0000 |
| EXT_INT10_FLTCON0 | 0x0858 | External interrupt EXT_INT10 filter configuration register 0 | 0x0000_0000 |
| EXT_INT10_FLTCON1 | 0x085C | External interrupt EXT_INT10 filter configuration register 1 | 0x0000_0000 |
| EXT_INT11_FLTCON0 | 0x0860 | External interrupt EXT_INT11 filter configuration register 0 | 0x0000_0000 |
| EXT_INT11_FLTCON1 | 0x0864 | External interrupt EXT_INT11 filter configuration register 1 | 0x0000_0000 |
| EXT_INT12_FLTCON0 | 0x0868 | External interrupt EXT_INT12 filter configuration register 0 | 0x0000_0000 |
| EXT_INT12_FLTCON1 | 0x086C | External interrupt EXT_INT12 filter configuration register 1 | 0x0000_0000 |
| EXT_INT23_MASK | 0x0908 | External interrupt EXT_INT23 mask register | 0x0000_007F |
| EXT_INT24_MASK | 0x090C | External interrupt EXT_INT24 mask register | 0x0000_007F |
| EXT_INT25_MASK | 0x0910 | External interrupt EXT_INT25 mask register | 0x0000_007F |
| EXT_INT26_MASK | 0x0914 | External interrupt EXT_INT26 mask register | 0x0000_007F |
| EXT_INT27_MASK | 0x0918 | External interrupt EXT_INT27 mask register | 0x0000_007F |
| EXT_INT28_MASK | 0x091C | External interrupt EXT_INT28 mask register | 0x0000_0003 |
| EXT_INT29_MASK | 0x0920 | External interrupt EXT_INT29 mask register | 0x0000_00FF |
| EXT_INT8_MASK | 0x0924 | External interrupt EXT_INT8 mask register | 0x0000_00FF |
| EXT_INT9_MASK | 0x0928 | External interrupt EXT_INT9 mask register | 0x0000_007F |
| EXT_INT10_MASK | 0x092C | External interrupt EXT_INT10 mask register | 0x0000_001F |
| EXT_INT11_MASK | 0x0930 | External interrupt EXT_INT11 mask register | 0x0000_00FF |
| EXT_INT12_MASK | 0x0934 | External interrupt EXT_INT12 mask register | 0x0000_00FF |
| EXT_INT23_PEND | 0x0A08 | External interrupt EXT_INT23 pending register | 0x0000_0000 |
| EXT_INT24_PEND | 0x0A0C | External interrupt EXT_INT24 pending register | 0x0000_0000 |
| EXT_INT25_PEND | 0x0A10 | External interrupt EXT_INT25 pending register | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------------|--------|--|-------------|
| EXT_INT26_PEND | 0x0A14 | External interrupt EXT_INT26 pending register | 0x0000_0000 |
| EXT_INT27_PEND | 0x0A18 | External interrupt EXT_INT27 pending register | 0x0000_0000 |
| EXT_INT28_PEND | 0x0A1C | External interrupt EXT_INT28 pending register | 0x0000_0000 |
| EXT_INT29_PEND | 0x0A20 | External interrupt EXT_INT29 pending register | 0x0000_0000 |
| EXT_INT8_PEND | 0x0A24 | External interrupt EXT_INT8 pending register | 0x0000_0000 |
| EXT_INT9_PEND | 0x0A28 | External interrupt EXT_INT9 pending register | 0x0000_0000 |
| EXT_INT10_PEND | 0x0A2C | External interrupt EXT_INT10 pending register | 0x0000_0000 |
| EXT_INT11_PEND | 0x0A30 | External interrupt EXT_INT11 pending register | 0x0000_0000 |
| EXT_INT12_PEND | 0x0A34 | External interrupt EXT_INT12 pending register | 0x0000_0000 |
| EXT_INT_SERVICE_XA | 0x0B08 | Current service register | 0x0000_0000 |
| EXT_INT_SERVICE_PEND_XA | 0x0B0C | Current service pending register | 0x0000_0000 |
| EXT_INT_GRPFIXPRI_XA | 0x0B10 | External interrupt group fixed priority control register | 0x0000_0000 |
| EXT_INT23_FIXPRI | 0x0B1C | External interrupt 23 fixed priority control register | 0x0000_0000 |
| EXT_INT24_FIXPRI | 0x0B20 | External interrupt 24 fixed priority control register | 0x0000_0000 |
| EXT_INT25_FIXPRI | 0x0B24 | External interrupt 25 fixed priority control register | 0x0000_0000 |
| EXT_INT26_FIXPRI | 0x0B28 | External interrupt 26 fixed priority control register | 0x0000_0000 |
| EXT_INT27_FIXPRI | 0x0B2C | External interrupt 27 fixed priority control register | 0x0000_0000 |
| EXT_INT28_FIXPRI | 0x0B30 | External interrupt 28 fixed priority control register | 0x0000_0000 |
| EXT_INT29_FIXPRI | 0x0B34 | External interrupt 29 fixed priority control register | 0x0000_0000 |
| EXT_INT8_FIXPRI | 0x0B38 | External interrupt 8 fixed priority control register | 0x0000_0000 |
| EXT_INT9_FIXPRI | 0x0B3C | External interrupt 9 fixed priority control register | 0x0000_0000 |
| EXT_INT10_FIXPRI | 0x0B40 | External interrupt 10 fixed priority control register | 0x0000_0000 |
| EXT_INT11_FIXPRI | 0x0B44 | External interrupt 11 fixed priority control register | 0x0000_0000 |
| EXT_INT12_FIXPRI | 0x0B48 | External interrupt 12 fixed priority control register | 0x0000_0000 |
| GPX0CON | 0x0C00 | Port group GPX0 configuration register | 0x0000_0000 |
| GPX0DAT | 0x0C04 | Port group GPX0 data register | 0x00 |
| GPX0PUD | 0x0C08 | Port group GPX0 pull-up/pull-down register | 0x5555 |
| GPX0DRV | 0x0C0C | Port group GPX0 drive strength control register | 0x00_0000 |
| GPX1CON | 0x0C20 | Port group GPX1 configuration register | 0x0000_0000 |
| GPX1DAT | 0x0C24 | Port group GPX1 data register | 0x00 |
| GPX1PUD | 0x0C28 | Port group GPX1 pull-up/pull-down register | 0x5555 |
| GPX1DRV | 0x0C2C | Port group GPX1 drive strength control register | 0x00_0000 |
| GPX2CON | 0x0C40 | Port group GPX2 configuration register | 0x0000_0000 |
| GPX2DAT | 0x0C44 | Port group GPX2 data register | 0x00 |

| Register | Offset | Description | Reset Value |
|-------------------|--------|--|-------------|
| GPX2PUD | 0x0C48 | Port group GPX2 pull-up/pull-down register | 0x5555 |
| GPX2DRV | 0x0C4C | Port group GPX2 drive strength control register | 0x00_0000 |
| GPX3CON | 0x0C60 | Port group GPX3 configuration register | 0x0000_0000 |
| GPX3DAT | 0x0C64 | Port group GPX3 data register | 0x00 |
| GPX3PUD | 0x0C68 | Port group GPX3 pull-up/pull-down register | 0x5555 |
| GPX3DRV | 0x0C6C | Port group GPX3 drive strength control register | 0x00_0000 |
| EXT_INT40_CON | 0x0E00 | External interrupt EXT_INT40 configuration register | 0x0000_0000 |
| EXT_INT41_CON | 0x0E04 | External interrupt EXT_INT41 configuration register | 0x0000_0000 |
| EXT_INT42_CON | 0x0E08 | External interrupt EXT_INT42 configuration register | 0x0000_0000 |
| EXT_INT43_CON | 0x0E0C | External interrupt EXT_INT43 configuration register | 0x0000_0000 |
| EXT_INT40_FLTCON0 | 0x0E80 | External Interrupt EXT_INT40 filter configuration register 0 | 0x8080_8080 |
| EXT_INT40_FLTCON1 | 0x0E84 | External interrupt EXT_INT40 filter configuration register 1 | 0x8080_8080 |
| EXT_INT41_FLTCON0 | 0x0E88 | External interrupt EXT_INT41 filter configuration register 0 | 0x8080_8080 |
| EXT_INT41_FLTCON1 | 0x0E8C | External interrupt EXT_INT41 filter configuration register 1 | 0x8080_8080 |
| EXT_INT42_FLTCON0 | 0x0E90 | External interrupt EXT_INT42 filter configuration register 0 | 0x8080_8080 |
| EXT_INT42_FLTCON1 | 0x0E94 | External interrupt EXT_INT42 filter configuration register 1 | 0x8080_8080 |
| EXT_INT43_FLTCON0 | 0x0E98 | External interrupt EXT_INT43 filter configuration register 0 | 0x8080_8080 |
| EXT_INT43_FLTCON1 | 0x0E9C | External interrupt EXT_INT43 filter configuration register 1 | 0x8080_8080 |
| EXT_INT40_MASK | 0x0F00 | External interrupt EXT_INT40 mask register | 0x0000_00FF |
| EXT_INT41_MASK | 0x0F04 | External interrupt EXT_INT41 mask register | 0x0000_00FF |
| EXT_INT42_MASK | 0x0F08 | External interrupt EXT_INT42 mask register | 0x0000_00FF |
| EXT_INT43_MASK | 0x0F0C | External interrupt EXT_INT43 mask register | 0x0000_00FF |
| EXT_INT40_PEND | 0x0F40 | External interrupt EXT_INT40 pending register | 0x0000_0000 |
| EXT_INT41_PEND | 0x0F44 | External interrupt EXT_INT41 pending register | 0x0000_0000 |
| EXT_INT42_PEND | 0x0F48 | External interrupt EXT_INT42 pending register | 0x0000_0000 |
| EXT_INT43_PEND | 0x0F4C | External interrupt EXT_INT43 pending register | 0x0000_0000 |
| PDNEN | 0x0F80 | Power down mode pad configure register | 0x00 |

- Base Address: 0x0386_0000

| Register | Offset | Description | Reset Value |
|-------------------------|--------|--|-------------|
| GPZCON | 0x0000 | Port group GPIO group Z (GPZ) configuration register | 0x0000_0000 |
| GPZDAT | 0x0004 | Port group GPZ data register | 0x00 |
| GPZPUD | 0x0008 | Port group GPZ pull-up/pull-down register | 0x1555 |
| GPZDRV | 0x000C | Port group GPZ drive strength control register | 0x00_0000 |
| GPZCONPDN | 0x0010 | Port group GPZ power down mode configuration register | 0x0000 |
| GPZPUDPDN | 0x0014 | Port group GPZ power down mode pull-up/pull-down register | 0x0000 |
| EXT_INT50_CON | 0x0700 | External interrupt EXT_INT50 configuration register | 0x0000_0000 |
| EXT_INT50_FLTCON0 | 0x0800 | External interrupt EXT_INT50 filter configuration register 0 | 0x0000_0000 |
| EXT_INT50_FLTCON1 | 0x0804 | External interrupt EXT_INT50 filter configuration register 1 | 0x0000_0000 |
| EXT_INT50_MASK | 0x0900 | External interrupt EXT_INT50 mask register | 0x0000_007F |
| EXT_INT50_PEND | 0x0A00 | External interrupt EXT_INT50 pending register | 0x0000_0000 |
| EXT_INT_SERVICE_XD | 0x0B08 | Current service register | 0x0000_0000 |
| EXT_INT_SERVICE_PEND_XD | 0x0B0C | Current service pending register | 0x0000_0000 |
| EXT_INT_GRPFPXPRI_XD | 0x0B10 | External interrupt group fixed priority control register | 0x0000_0000 |
| EXT_INT50_FIXPRI | 0x0B14 | External interrupt 50 fixed priority control register | 0x0000_0000 |
| PDNEN | 0x0F80 | Power down mode pad configure register | 0x00 |

- Base Address: 0x106E_0000

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| GPV0CON | 0x0000 | Port group GPV0 configuration register | 0x0000_0000 |
| GPV0DAT | 0x0004 | Port group GPV0 data register | 0x00 |
| GPV0PUD | 0x0008 | Port group GPV0 pull-up/pull-down register | 0x5555 |
| GPV0DRV | 0x000C | Port group GPV0 drive strength control register | 0x00_0000 |
| GPV0CONPDN | 0x0010 | Port group GPV0 power down mode configuration register | 0x0000 |
| GPV0PUDPDN | 0x0014 | Port group GPV0 power down mode pull-up/pull-down register | 0x0000 |
| GPV1CON | 0x0020 | Port group GPV1 configuration register | 0x0000_0000 |
| GPV1DAT | 0x0024 | Port group GPV1 data register | 0x00 |
| GPV1PUD | 0x0028 | Port group GPV1 pull-up/pull-down register | 0x5555 |
| GPV1DRV | 0x002C | Port group GPV1 drive strength control register | 0x00_0000 |
| GPV1CONPDN | 0x0030 | Port group GPV1 power down mode configuration register | 0x0000 |
| GPV1PUDPDN | 0x0034 | Port group GPV1 power down mode pull-up/pull-down register | 0x0000 |

| Register | Offset | Description | Reset Value |
|-------------------|--------|--|-------------|
| ETC7PUD | 0x0048 | Port group ETC7 pull-up/pull-down register | 0x0005 |
| ETC7DRV | 0x004C | Port group ETC7 drive strength control register | 0x00_0000 |
| GPV2CON | 0x0060 | Port group GPV2 configuration register | 0x0000_0000 |
| GPV2DAT | 0x0064 | Port group GPV2 data register | 0x00 |
| GPV2PUD | 0x0068 | Port group GPV2 pull-up/pull-down register | 0x5555 |
| GPV2DRV | 0x006C | Port group GPV2 drive strength control register | 0x00_0000 |
| GPV2CONPDN | 0x0070 | Port group GPV2 power down mode configuration register | 0x0000 |
| GPV2PUDPDN | 0x0074 | Port group GPV2 power down mode pull-up/pull-down register | 0x0000 |
| GPV3CON | 0x0080 | Port group GPV3 configuration register | 0x0000_0000 |
| GPV3DAT | 0x0084 | Port group GPV3 data register | 0x00 |
| GPV3PUD | 0x0088 | Port group GPV3 pull-up/pull-down register | 0x5555 |
| GPV3DRV | 0x008C | Port group GPV3 drive strength control register | 0x00_0000 |
| GPV3CONPDN | 0x0090 | Port group GPV3 power down mode configuration register | 0x0000 |
| GPV3PUDPDN | 0x0094 | Port group GPV3 power down mode pull-up/pull-down register | 0x0000 |
| ETC8PUD | 0x00A8 | Port group ETC8 pull-up/pull-down register | 0x0005 |
| ETC8DRV | 0x00AC | Port group ETC8 drive strength control register | 0x00_0000 |
| GPV4CON | 0x00C0 | Port group GPV4 configuration register | 0x0000_0000 |
| GPV4DAT | 0x00C4 | Port group GPV4 data register | 0x00 |
| GPV4PUD | 0x00C8 | Port group GPV4 pull-up/pull-down register | 0x0005 |
| GPV4DRV | 0x00CC | Port group GPV4 drive strength control register | 0x00_0000 |
| GPV4CONPDN | 0x00D0 | Port group GPV4 power down mode configuration register | 0x0000 |
| GPV4PUDPDN | 0x00D4 | Port group GPV4 power down mode pull-up/pull-down register | 0x0000 |
| EXT_INT30_CON | 0x0700 | External interrupt EXT_INT30 configuration register | 0x0000_0000 |
| EXT_INT31_CON | 0x0704 | External interrupt EXT_INT31 configuration register | 0x0000_0000 |
| EXT_INT32_CON | 0x0708 | External interrupt EXT_INT32 configuration register | 0x0000_0000 |
| EXT_INT33_CON | 0x070C | External interrupt EXT_INT33 configuration register | 0x0000_0000 |
| EXT_INT34_CON | 0x0710 | External interrupt EXT_INT34 configuration register | 0x0000_0000 |
| EXT_INT30_FLTCON0 | 0x0800 | External interrupt EXT_INT30 filter configuration register 0 | 0x0000_0000 |
| EXT_INT30_FLTCON1 | 0x0804 | External interrupt EXT_INT30 filter configuration register 1 | 0x0000_0000 |
| EXT_INT31_FLTCON0 | 0x0808 | External interrupt EXT_INT31 filter configuration register 0 | 0x0000_0000 |
| EXT_INT31_FLTCON1 | 0x080C | External interrupt EXT_INT31 filter configuration register 1 | 0x0000_0000 |
| EXT_INT32_FLTCON0 | 0x0810 | External interrupt EXT_INT32 filter configuration register 0 | 0x0000_0000 |
| EXT_INT32_FLTCON1 | 0x0814 | External interrupt EXT_INT32 filter configuration register 1 | 0x0000_0000 |
| EXT_INT33_FLTCON0 | 0x0818 | External interrupt EXT_INT33 filter configuration register 0 | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------------|--------|--|-------------|
| EXT_INT33_FLTCON1 | 0x081C | External interrupt EXT_INT33 filter configuration register 1 | 0x0000_0000 |
| EXT_INT34_FLTCON0 | 0x0820 | External interrupt EXT_INT34 filter configuration register 0 | 0x0000_0000 |
| EXT_INT34_FLTCON1 | 0x0824 | External interrupt EXT_INT34 filter configuration register 1 | 0x0000_0000 |
| EXT_INT30_MASK | 0x0900 | External interrupt EXT_INT30 mask register | 0x0000_00FF |
| EXT_INT31_MASK | 0x0904 | External interrupt EXT_INT31 mask register | 0x0000_00FF |
| EXT_INT32_MASK | 0x0908 | External interrupt EXT_INT32 mask register | 0x0000_00FF |
| EXT_INT33_MASK | 0x090C | External interrupt EXT_INT33 mask register | 0x0000_00FF |
| EXT_INT34_MASK | 0x0910 | External interrupt EXT_INT34 mask register | 0x0000_0003 |
| EXT_INT30_PEND | 0x0A00 | External interrupt EXT_INT30 pending register | 0x0000_0000 |
| EXT_INT31_PEND | 0x0A04 | External interrupt EXT_INT31 pending register | 0x0000_0000 |
| EXT_INT32_PEND | 0x0A08 | External interrupt EXT_INT32 pending register | 0x0000_0000 |
| EXT_INT33_PEND | 0x0A0C | External interrupt EXT_INT33 pending register | 0x0000_0000 |
| EXT_INT34_PEND | 0x0A10 | External interrupt EXT_INT34 pending register | 0x0000_0000 |
| EXT_INT_SERVICE_XC | 0x0B08 | Current service register | 0x0000_0000 |
| EXT_INT_SERVICE_PEND_XC | 0x0B0C | Current service pending register | 0x0000_0000 |
| EXT_INT_GRPFIXPRI_XC | 0x0B10 | External interrupt group fixed priority control register | 0x0000_0000 |
| EXT_INT30_FIXPRI | 0x0B14 | External interrupt 30 fixed priority control register | 0x0000_0000 |
| EXT_INT31_FIXPRI | 0x0B18 | External interrupt 31 fixed priority control register | 0x0000_0000 |
| EXT_INT32_FIXPRI | 0x0B1C | External interrupt 32 fixed priority control register | 0x0000_0000 |
| EXT_INT33_FIXPRI | 0x0B20 | External interrupt 33 fixed priority control register | 0x0000_0000 |
| EXT_INT34_FIXPRI | 0x0B24 | External interrupt 34 fixed priority control register | 0x0000_0000 |
| PDNEN | 0x0F80 | Power down mode pad configure register | 0x00 |

4.3.2 Part 1

For the following SFRs, Sets the value does not take effect immediately. It takes at least 800 APB clocks for the value to take effect after the SFR is actually changed: The SFRs are:

GPA0PUD, GPA0DRV, GPA1PUD, GPA1DRV, GPBPUD, GPBDRV, GPC0PUD, GPC0DRV, GPC1PUD, GPC1DRV, GPD0PUD, GPD0DRV, GPD1PUD, GPD1DRV, GPF0PUD, GPF0DRV, GPF1PUD, GPF1DRV, GPF2PUD, GPF2DRV, GPF3PUD, GPF3DRV, GPJ0PUD, GPJ0DRV, GPJ1PUD, GPJ1DRV.

4.3.2.1 GPA0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPA0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_1_RTSn 0x3 = I2C_2_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT1[7] | 0x00 |
| GPA0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_1_CTSn 0x3 = I2C_2_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT1[6] | 0x00 |
| GPA0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_1_TXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[5] | 0x00 |
| GPA0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_1_RXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[4] | 0x00 |
| GPA0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_0_RTSn 0x3 to 0xE = Reserved 0xF = EXT_INT1[3] | 0x00 |
| GPA0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_0_CTSn 0x3 to 0xE = Reserved 0xF = EXT_INT1[2] | 0x00 |
| GPA0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_0_TXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[1] | 0x00 |
| GPA0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_0_RXD 0x3 to 0xE = Reserved 0xF = EXT_INT1[0] | 0x00 |

4.3.2.2 GPA0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPA0DAT[7:0] | [7:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.3 GPA0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0008, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPA0PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.4 GPA0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPA0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.5 GPA0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPA0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.6 GPA0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPA0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.7 GPA1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPA1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_3_TXD 0x3 = Reserved 0x4 = UART_AUDIO_TXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[5] | 0x00 |
| GPA1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_3_RXD 0x3 = Reserved 0x4 = UART_AUDIO_RXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[4] | 0x00 |
| GPA1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_2_RTsn 0x3 = I2C_3_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT2[3] | 0x00 |
| GPA1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_2_CTSn 0x3 = I2C_3_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT2[2] | 0x00 |
| GPA1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_2_TXD 0x3 = Reserved 0x4 = UART_AUDIO_TXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[1] | 0x00 |
| GPA1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = UART_2_RXD 0x3 = Reserved 0x4 = UART_AUDIO_RXD 0x5 to 0xE = Reserved 0xF = EXT_INT2[0] | 0x00 |

4.3.2.8 GPA1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPA1DAT[5:0] | [5:0] | RWX | When you configure port as input port, then corresponding bit is the pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.9 GPA1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0028, Reset Value = 0x0555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPA1PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0555 |

4.3.2.10 GPA1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x002C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------------------------|------|--|-------------|
| GPA1DRV [n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.11 GPA1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPA1[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.12 GPA1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPA1[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.13 GPBCON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| GPBCON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_1_MOSI 0x3 to 0xE = Reserved 0xF = EXT_INT3[7] | 0x00 |
| GPBCON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_1_MISO 0x3 to 0xE = Reserved 0xF = EXT_INT3[6] | 0x00 |
| GPBCON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_1_nSS 0x3 = Reserved 0x4 = IEM_SPWI 0x5 to 0xE = Reserved 0xF = EXT_INT3[5] | 0x00 |
| GPBCON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_1_CLK 0x3 = Reserved 0x4 = IEM_SCLK 0x5 to 0xE = Reserved 0xF = EXT_INT3[4] | 0x00 |
| GPBCON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_0_MOSI 0x3 = I2C_5_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT3[3] | 0x00 |
| GPBCON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_0_MISO 0x3 = I2C_5_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT3[2] | 0x00 |
| GPBCON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SPI_0_nSS 0x3 = I2C_4_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT3[1] | 0x00 |
| GPBCON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|--|-------------|
| | | | 0x2 = SPI_0_CLK 0x3 = I2C_4_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT3[0] | |

4.3.2.14 GPBDAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-------------|-------|------|--|-------------|
| GPBDAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.15 GPBPUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0048, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|-----------|---------------------------|------|--|-------------|
| GPBPUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.16 GPBDRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------------------------|------|---|-------------|
| GPBDRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4 | 0x0000 |

4.3.2.17 GPBCONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|--------|---------------------------|------|---|-------------|
| GPB[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.18 GPBPUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|--------|---------------------------|------|---|-------------|
| GPB[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up | 0x00 |

4.3.2.19 GPC0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPC0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_1_SDO 0x3 = PCM_1_SOUT 0x4 = AC97SDO 0x5 to 0xE = Reserved 0xF = EXT_INT4[4] | 0x00 |
| GPC0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_1_SDI 0x3 = PCM_1_SIN 0x4 = AC97SDI 0x5 to 0xE = Reserved 0xF = EXT_INT4[3] | 0x00 |
| GPC0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_1_LRCK 0x3 = PCM_1_FSYNC 0x4 = AC97SYNC 0x5 to 0xE = Reserved 0xF = EXT_INT4[2] | 0x00 |
| GPC0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_1_CDCLK 0x3 = PCM_1_EXTCLK 0x4 = AC97RESETn 0x5 to 0xE = Reserved 0xF = EXT_INT4[1] | 0x00 |
| GPC0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_1_SCLK 0x3 = PCM_1_SCLK 0x4 = AC97BITCLK 0x5 to 0xE = Reserved 0xF = EXT_INT4[0] | 0x00 |

4.3.2.20 GPC0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPC0DAT[4:0] | [4:0] | RWX | When you configure as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.21 GPC0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0068, Reset Value = 0x0155

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPC0PUD[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0155 |

4.3.2.22 GPC0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPC0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.23 GPC0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPC0[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.24 GPC0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPC0[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.25 GPC1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPC1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_2_SDO 0x3 = PCM_2_SOUT 0x4 = I2C_6_SCL 0x5 = SPI_2_MOSI 0x6 to 0xE = Reserved 0xF = EXT_INT5[4] | 0x00 |
| GPC1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_2_SDI 0x3 = PCM_2_SIN 0x4 = I2C_6_SDA 0x5 = SPI_2_MISO 0x6 to 0xE = Reserved 0xF = EXT_INT5[3] | 0x00 |
| GPC1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_2_LRCK 0x3 = PCM_2_FSYNC 0x4 = Reserved 0x5 = SPI_2_nSS 0x6 to 0xE = Reserved 0xF = EXT_INT5[2] | 0x00 |
| GPC1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_2_CDCLK 0x3 = PCM_2_EXTCLK 0x4 = SPDIF_EXTCLK 0x5 = SPI_2_CLK 0x6 to 0xE = Reserved 0xF = EXT_INT5[1] | 0x00 |
| GPC1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_2_SCLK 0x3 = PCM_2_SCLK 0x4 = SPDIF_0_OUT 0x5 to 0xE = Reserved 0xF = EXT_INT5[0] | 0x00 |

4.3.2.26 GPC1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPC1DAT[4:0] | [4:0] | RWX | When you configure port as input port, corresponding bit is pin state. When configuring as output port, pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.27 GPC1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0088, Reset Value = 0x0155

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPC1PUD[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0155 |

4.3.2.28 GPC1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPC1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.29 GPC1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPC1[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.30 GPC1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPC1[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.31 GPD0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPD0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = TOUT_3 0x3 = I2C_7_SCL 0x4 to 0xE = Reserved 0xF = EXT_INT6[3] | 0x00 |
| GPD0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = TOUT_2 0x3 = I2C_7_SDA 0x4 to 0xE = Reserved 0xF = EXT_INT6[2] | 0x00 |
| GPD0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = TOUT_1 0x3 = LCD_PWM 0x4 to 0xE = Reserved 0xF = EXT_INT6[1] | 0x00 |
| GPD0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = TOUT_0 0x3 = LCD_FRM 0x4 to 0xE = Reserved 0xF = EXT_INT6[0] | 0x00 |

4.3.2.32 GPD0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPD0DAT[3:0] | [3:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.33 GPD0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0055

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPD0PUD[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0055 |

4.3.2.34 GPD0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPD0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.35 GPD0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00B0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPD0[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.36 GPD0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00B4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPD0[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.37 GPD1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPD1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = I2C_1_SCL 0x3 = MIPI1_ESC_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[3] | 0x00 |
| GPD1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = I2C_1_SDA 0x3 = MIPI1_BYTE_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[2] | 0x00 |
| GPD1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = I2C_0_SCL 0x3 = MIPI0_ESC_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[1] | 0x00 |
| GPD1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = I2C_0_SDA 0x3 = MIPI0_BYTE_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT7[0] | 0x00 |

4.3.2.38 GPD1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPD1DAT[3:0] | [3:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.39 GPD1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0055

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPD1PUD[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up | 0x0055 |

4.3.2.40 GPD1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPD1DRV[n] | [23:16] | RW | Reserved (should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.41 GPD1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPD1[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.42 GPD1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPD1[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.43 GPF0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPF0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT13[7] | 0x00 |
| GPF0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT13[6] | 0x00 |
| GPF0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT13[5] | 0x00 |
| GPF0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT13[4] | 0x00 |
| GPF0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VCLK 0x3 to 0xE = Reserved 0xF = EXT_INT13[3] | 0x00 |
| GPF0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output, 0x2 = LCD_VDEN, 0x3 to 0xE = Reserved, 0xF = EXT_INT13[2] | 0x00 |
| GPF0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT13[1] | 0x00 |
| GPF0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_HSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT13[0] | 0x00 |

4.3.2.44 GPF0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0184, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPF0DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.45 GPF0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0188, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF0PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.46 GPF0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x018C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.47 GPF0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPF0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.48 GPF0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0194, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPF0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.49 GPF1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPF1CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT14[7] | 0x00 |
| GPF1CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT14[6] | 0x00 |
| GPF1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT14[5] | 0x00 |
| GPF1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT14[4] | 0x00 |
| GPF1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT14[3] | 0x00 |
| GPF1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT14[2] | 0x00 |
| GPF1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT14[1] | 0x00 |
| GPF1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT14[0] | 0x00 |

4.3.2.50 GPF1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPF1DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.51 GPF1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01A8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF1PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.52 GPF1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01AC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF1DRV[n] | [23:16] | RW | Reserved (should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.53 GPF1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01B0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPF1[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0, 0x1 = Outputs 1, 0x2 = Input, 0x3 = Previous state | 0x00 |

4.3.2.54 GPF1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01B4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPF1[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up | 0x00 |

4.3.2.55 GPF2CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPF2CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[19] 0x3 to 0xE = Reserved 0xF = EXT_INT15[7] | 0x00 |
| GPF2CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[18] 0x3 to 0xE = Reserved 0xF = EXT_INT15[6] | 0x00 |
| GPF2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[17] 0x3 to 0xE = Reserved 0xF = EXT_INT15[5] | 0x00 |
| GPF2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[16] 0x3 to 0xE = Reserved 0xF = EXT_INT15[4] | 0x00 |
| GPF2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT15[3] | 0x00 |
| GPF2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT15[2] | 0x00 |
| GPF2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output, 0x2 = LCD_VD[13], 0x3 to 0xE = Reserved 0xF = EXT_INT15[1] | 0x00 |
| GPF2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[12] 0x3 to 0xE = Reserved, 0xF = EXT_INT15[0] | 0x00 |

4.3.2.56 GPF2DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPF2DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.57 GPF2PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01C8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF2PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.58 GPF2DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01CC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.59 GPF2CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPF2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.60 GPF2PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPF2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.61 GPF3CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPF3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SYS_OE 0x3 to 0xE = Reserved 0xF = EXT_INT16[5] | 0x00 |
| GPF3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = VSYNC_LDI 0x3 to 0xE = Reserved 0xF = EXT_INT16[4] | 0x00 |
| GPF3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[23] 0x3 to 0xE = Reserved 0xF = EXT_INT16[3] | 0x00 |
| GPF3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[22] 0x3 to 0xE = Reserved 0xF = EXT_INT16[2] | 0x00 |
| GPF3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[21] 0x3 to 0xE = Reserved 0xF = EXT_INT16[1] | 0x00 |
| GPF3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = LCD_VD[20] 0x3 to 0xE = Reserved 0xF = EXT_INT16[0] | 0x00 |

4.3.2.62 GPF3DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPF3DAT[5:0] | [5:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.63 GPF3PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x01E8, Reset Value = 0x0555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF3PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0555 |

4.3.2.64 GPF3DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x01EC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPF3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.65 GPF3CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01F0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPF3[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.66 GPF3PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPF3[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.67 ETC1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0228, Reset Value = 0x0005

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC1PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0005 |

ETC1PUD[1:0] controls XsbusDATA.

ETC1PUD[3:2] controls XsbusCLK.

4.3.2.68 ETC1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x022C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| ETC1DRV[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

ETC1PUD[1:0] controls XsbusDATA.

ETC1PUD[3:2] controls XsbusCLK.

4.3.2.69 GPJ0CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0240, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPJ0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[4] 0x3 to 0xE = Reserved 0xF = EXT_INT21[7] | 0x00 |
| GPJ0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[3] 0x3 to 0xE = Reserved 0xF = EXT_INT21[6] | 0x00 |
| GPJ0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[2] 0x3 to 0xE = Reserved 0xF = EXT_INT21[5] | 0x00 |
| GPJ0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[1] 0x3 to 0xE = Reserved 0xF = EXT_INT21[4] | 0x00 |
| GPJ0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[0] 0x3 to 0xE = Reserved 0xF = EXT_INT21[3] | 0x00 |
| GPJ0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_HREF 0x3 to 0xE = Reserved 0xF = EXT_INT21[2] | 0x00 |
| GPJ0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_VSYNC 0x3 to 0xE = Reserved 0xF = EXT_INT21[1] | 0x00 |
| GPJ0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_PCLK 0x3 to 0xE = Reserved 0xF = EXT_INT21[0] | 0x00 |

4.3.2.70 GPJ0DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0244, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPJ0DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.71 GPJ0PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0248, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPJ0PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.2.72 GPJ0DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x024C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPJ0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4 | 0x0000 |

4.3.2.73 GPJ0CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0250, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPJ0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.74 GPJ0PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0254, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPJ0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.75 GPJ1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPJ1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_FIELD 0x3 to 0xE = Reserved 0xF = EXT_INT22[4] | 0x00 |
| GPJ1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_CLKOUT 0x3 to 0xE = Reserved 0xF = EXT_INT22[3] | 0x00 |
| GPJ1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[7] 0x3 to 0xE = Reserved 0xF = EXT_INT22[2] | 0x00 |
| GPJ1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[6] 0x3 to 0xE = Reserved 0xF = EXT_INT22[1] | 0x00 |
| GPJ1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_A_DATA[5] 0x3 to 0xE = Reserved 0xF = EXT_INT22[0] | 0x00 |

4.3.2.76 GPJ1DAT

- Base Address: 0x1140_0000
- Address = Base Address + 0x0264, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPJ1DAT[4:0] | [4:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.2.77 GPJ1PUD

- Base Address: 0x1140_0000
- Address = Base Address + 0x0268, Reset Value = 0x0155

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPJ1PUD[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0155 |

4.3.2.78 GPJ1DRV

- Base Address: 0x1140_0000
- Address = Base Address + 0x026C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPJ1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.2.79 GPJ1CONPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0270, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPJ1[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.2.80 GPJ1PUDPDN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0274, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPJ1[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.2.81 EXT_INT1CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT1_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT1[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT1_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT1[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT1_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT1[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT1_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT1[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT1_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT1[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| EXT_INT1_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT1[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT1_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT1[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT1_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT1[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.82 EXT_INT2CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT2_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT2[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT2_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT2[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT2_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT2[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT2_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT2[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT2_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT2[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|-------|------|---|-------------|
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT2_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT2[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.83 EXT_INT3CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT3_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT3[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT3_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT3[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT3_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT3[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT3_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT3[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT3_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT3[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| EXT_INT3_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT3[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT3_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT3[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT3_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT3[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.84 EXT_INT4CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:20] | – | Reserved | 0x000 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT4_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT4[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT4_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT4[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT4_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT4[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT4_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT4[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT4_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT4[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.85 EXT_INT5CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:20] | – | Reserved | 0x000 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT5_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT5[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT5_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT5[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT5_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT5[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT5_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT5[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT5_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT5[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.86 EXT_INT6CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0x0000 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT6_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT6[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT6_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT6[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT6_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT6[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT6_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT6[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.87 EXT_INT7CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0718, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0x0000 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT7_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT7[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT7_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT7[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT7_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT7[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT7_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT7[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.88 EXT_INT13CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0730, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| EXT_INT13_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT13[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT13_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT13[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT13_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT13[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT13_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT13[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT13_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT13[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT13_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT13[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT13_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT13[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT13_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT13[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.89 EXT_INT14CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0734, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| EXT_INT14_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT14[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT14_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT14[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT14_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT14[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT14_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT14[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT14_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT14[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT14_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT14[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT14_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT14[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT14_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT14[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.90 EXT_INT15CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0738, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| EXT_INT15_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT15[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT15_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT15[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT15_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT15[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT15_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT15[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT15_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT15[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT15_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT15[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT15_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT15[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT15_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT15[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.91 EXT_INT16CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x073C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT16_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT16[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT16_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT16[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT16_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT16[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT16_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT16[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT16_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT16[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT16_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT16[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.92 EXT_INT21CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0740, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| EXT_INT21_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT21[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT21_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT21[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT21_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT21[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT21_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT21[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT21_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT21[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT21_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT21[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT21_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT21[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT21_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT21[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.93 EXT_INT22CON

- Base Address: 0x1140_0000
- Address = Base Address + 0x0744, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:20] | – | Reserved | 0x000 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT22_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT22[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT22_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT22[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT22_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT22[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT22_CON[1] | [6:4] | W | Sets signaling method of EXT_INT22[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT22_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT22[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.2.94 EXT_INT1_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN1[3] | [31] | RW | Filter Enable for EXT_INT1[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[3] | [30:24] | RW | Filtering width of EXT_INT1[3] | 0x00 |
| FLTEN1[2] | [23] | RW | Filter Enable for EXT_INT1[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[2] | [22:16] | RW | Filtering width of EXT_INT1[2] | 0x00 |
| FLTEN1[1] | [15] | RW | Filter Enable for EXT_INT1[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[1] | [14:8] | RW | Filtering width of EXT_INT1[1] | 0x00 |
| FLTEN1[0] | [7] | RW | Filter Enable for EXT_INT1[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[0] | [6:0] | RW | Filtering width of EXT_INT1[0] | 0x00 |

4.3.2.95 EXT_INT1_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN1[7] | [31] | RW | Filter Enable for EXT_INT1[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[7] | [30:24] | RW | Filtering width of EXT_INT1[7] | 0x00 |
| FLTEN1[6] | [23] | RW | Filter Enable for EXT_INT1[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[6] | [22:16] | RW | Filtering width of EXT_INT1[6] | 0x00 |
| FLTEN1[5] | [15] | RW | Filter Enable for EXT_INT1[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[5] | [14:8] | RW | Filtering width of EXT_INT1[5] | 0x00 |
| FLTEN1[4] | [7] | RW | Filter Enable for EXT_INT1[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH1[4] | [6:0] | RW | Filtering width of EXT_INT1[4] | 0x00 |

4.3.2.96 EXT_INT2_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN2[3] | [31] | RW | Filter Enable for EXT_INT2[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[3] | [30:24] | RW | Filtering width of EXT_INT2[3] | 0x00 |
| FLTEN2[2] | [23] | RW | Filter Enable for EXT_INT2[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[2] | [22:16] | RW | Filtering width of EXT_INT2[2] | 0x00 |
| FLTEN2[1] | [15] | RW | Filter Enable for EXT_INT2[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[1] | [14:8] | RW | Filtering width of EXT_INT2[1] | 0x00 |
| FLTEN2[0] | [7] | RW | Filter Enable for EXT_INT2[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[0] | [6:0] | RW | Filtering width of EXT_INT2[0] | 0x00 |

4.3.2.97 EXT_INT2_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:16] | – | Reserved | 0x0000 |
| FLTEN2[5] | [15] | RW | Filter Enable for EXT_INT2[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[5] | [14:8] | RW | Filtering width of EXT_INT2[5] | 0x00 |
| FLTEN2[4] | [7] | RW | Filter Enable for EXT_INT2[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH2[4] | [6:0] | RW | Filtering width of EXT_INT2[4] | 0x00 |

4.3.2.98 EXT_INT3_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN3[3] | [31] | RW | Filter Enable for EXT_INT3[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[3] | [30:24] | RW | Filtering width of EXT_INT3[3] | 0x00 |
| FLTEN3[2] | [23] | RW | Filter Enable for EXT_INT3[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[2] | [22:16] | RW | Filtering width of EXT_INT3[2] | 0x00 |
| FLTEN3[1] | [15] | RW | Filter Enable for EXT_INT3[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[1] | [14:8] | RW | Filtering width of EXT_INT3[1] | 0x00 |
| FLTEN3[0] | [7] | RW | Filter Enable for EXT_INT3[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[0] | [6:0] | RW | Filtering width of EXT_INT3[0] | 0x00 |

4.3.2.99 EXT_INT3_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN3[7] | [31] | RW | Filter Enable for EXT_INT3[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[7] | [30:24] | RW | Filtering width of EXT_INT3[7] | 0x00 |
| FLTEN3[6] | [23] | RW | Filter Enable for EXT_INT3[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[6] | [22:16] | RW | Filtering width of EXT_INT3[6] | 0x00 |
| FLTEN3[5] | [15] | RW | Filter Enable for EXT_INT3[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[5] | [14:8] | RW | Filtering width of EXT_INT3[5] | 0x00 |
| FLTEN3[4] | [7] | RW | Filter Enable for EXT_INT3[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH3[4] | [6:0] | RW | Filtering width of EXT_INT3[4] | 0x00 |

4.3.2.100 EXT_INT4_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN4[3] | [31] | RW | Filter Enable for EXT_INT4[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH4[3] | [30:24] | RW | Filtering width of EXT_INT4[3] | 0x00 |
| FLTEN4[2] | [23] | RW | Filter Enable for EXT_INT4[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH4[2] | [22:16] | RW | Filtering width of EXT_INT4[2] | 0x00 |
| FLTEN4[1] | [15] | RW | Filter Enable for EXT_INT4[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH4[1] | [14:8] | RW | Filtering width of EXT_INT4[1] | 0x00 |
| FLTEN4[0] | [7] | RW | Filter Enable for EXT_INT4[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH4[0] | [6:0] | RW | Filtering width of EXT_INT4[0] | 0x00 |

4.3.2.101 EXT_INT4_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|--|-------------|
| RSVD | [31:8] | – | Reserved | 0x0000000 |
| FLTEN4[4] | [7] | RW | Filter Enable for EXT_INT4[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH4[4] | [6:0] | RW | Filtering width of EXT_INT4[4] | 0x00 |

4.3.2.102 EXT_INT5_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN5[3] | [31] | RW | Filter Enable for EXT_INT5[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH5[3] | [30:24] | RW | Filtering width of EXT_INT5[3] | 0x00 |
| FLTEN5[2] | [23] | RW | Filter Enable for EXT_INT5[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH5[2] | [22:16] | RW | Filtering width of EXT_INT5[2] | 0x00 |
| FLTEN5[1] | [15] | RW | Filter Enable for EXT_INT5[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH5[1] | [14:8] | RW | Filtering width of EXT_INT5[1] | 0x00 |
| FLTEN5[0] | [7] | RW | Filter Enable for EXT_INT5[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH5[0] | [6:0] | RW | Filtering width of EXT_INT5[0] | 0x00 |

4.3.2.103 EXT_INT5_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|--|-------------|
| RSVD | [31:8] | – | Reserved | 0x0000000 |
| FLTEN5[4] | [7] | RW | Filter Enable for EXT_INT5[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH5[4] | [6:0] | RW | Filtering width of EXT_INT5[4] | 0x00 |

4.3.2.104 EXT_INT6_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN6[3] | [31] | RW | Filter Enable for EXT_INT6[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH6[3] | [30:24] | RW | Filtering width of EXT_INT6[3] | 0x00 |
| FLTEN6[2] | [23] | RW | Filter Enable for EXT_INT6[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH6[2] | [22:16] | RW | Filtering width of EXT_INT6[2] | 0x00 |
| FLTEN6[1] | [15] | RW | Filter Enable for EXT_INT6[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH6[1] | [14:8] | RW | Filtering width of EXT_INT6[1] | 0x00 |
| FLTEN6[0] | [7] | RW | Filter Enable for EXT_INT6[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH6[0] | [6:0] | RW | Filtering width of EXT_INT6[0] | 0x00 |

4.3.2.105 EXT_INT6_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|-------------|-------------|
| RSVD | [31:0] | – | Reserved | 0x00000000 |

4.3.2.106 EXT_INT7_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| FLTEN7[3] | [31] | RW | Filter Enable for EXT_INT7[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH7[3] | [30:24] | RW | Filtering width of EXT_INT7[3] | 0x00 |
| FLTEN7[2] | [23] | RW | Filter Enable for EXT_INT7[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH7[2] | [22:16] | RW | Filtering width of EXT_INT7[2] | 0x00 |
| FLTEN7[1] | [15] | RW | Filter Enable for EXT_INT7[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH7[1] | [14:8] | RW | Filtering width of EXT_INT7[1] | 0x00 |
| FLTEN7[0] | [7] | RW | Filter Enable for EXT_INT7[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH7[0] | [6:0] | RW | Filtering width of EXT_INT7[0] | 0x00 |

4.3.2.107 EXT_INT7_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|-------------|-------------|
| RSVD | [31:0] | – | Reserved | 0x00000000 |

4.3.2.108 EXT_INT13_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN13[3] | [31] | RW | Filter Enable for EXT_INT13[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[3] | [30:24] | RW | Filtering width of EXT_INT13[3] | 0x00 |
| FLTEN13[2] | [23] | RW | Filter Enable for EXT_INT13[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[2] | [22:16] | RW | Filtering width of EXT_INT13[2] | 0x00 |
| FLTEN13[1] | [15] | RW | Filter Enable for EXT_INT13[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[1] | [14:8] | RW | Filtering width of EXT_INT13[1] | 0x00 |
| FLTEN13[0] | [7] | RW | Filter Enable for EXT_INT13[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[0] | [6:0] | RW | Filtering width of EXT_INT13[0] | 0x00 |

4.3.2.109 EXT_INT13_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN13[7] | [31] | RW | Filter Enable for EXT_INT13[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[7] | [30:24] | RW | Filtering width of EXT_INT13[7] | 0x00 |
| FLTEN13[6] | [23] | RW | Filter Enable for EXT_INT13[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[6] | [22:16] | RW | Filtering width of EXT_INT13[6] | 0x00 |
| FLTEN13[5] | [15] | RW | Filter Enable for EXT_INT13[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[5] | [14:8] | RW | Filtering width of EXT_INT13[5] | 0x00 |
| FLTEN13[4] | [7] | RW | Filter Enable for EXT_INT13[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH13[4] | [6:0] | RW | Filtering width of EXT_INT13[4] | 0x00 |

4.3.2.110 EXT_INT14_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN14[3] | [31] | RW | Filter Enable for EXT_INT14[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[3] | [30:24] | RW | Filtering width of EXT_INT14[3] | 0x00 |
| FLTEN14[2] | [23] | RW | Filter Enable for EXT_INT14[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[2] | [22:16] | RW | Filtering width of EXT_INT14[2] | 0x00 |
| FLTEN14[1] | [15] | RW | Filter Enable for EXT_INT14[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[1] | [14:8] | RW | Filtering width of EXT_INT14[1] | 0x00 |
| FLTEN14[0] | [7] | RW | Filter Enable for EXT_INT14[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[0] | [6:0] | RW | Filtering width of EXT_INT14[0] | 0x00 |

4.3.2.111 EXT_INT14_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN14[7] | [31] | RW | Filter Enable for EXT_INT14[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[7] | [30:24] | RW | Filtering width of EXT_INT14[7] | 0x00 |
| FLTEN14[6] | [23] | RW | Filter Enable for EXT_INT14[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[6] | [22:16] | RW | Filtering width of EXT_INT14[6] | 0x00 |
| FLTEN14[5] | [15] | RW | Filter Enable for EXT_INT14[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[5] | [14:8] | RW | Filtering width of EXT_INT14[5] | 0x00 |
| FLTEN14[4] | [7] | RW | Filter Enable for EXT_INT14[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH14[4] | [6:0] | RW | Filtering width of EXT_INT14[4] | 0x00 |

4.3.2.112 EXT_INT15_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0870, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN15[3] | [31] | RW | Filter Enable for EXT_INT15[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[3] | [30:24] | RW | Filtering width of EXT_INT15[3] | 0x00 |
| FLTEN15[2] | [23] | RW | Filter Enable for EXT_INT15[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[2] | [22:16] | RW | Filtering width of EXT_INT15[2] | 0x00 |
| FLTEN15[1] | [15] | RW | Filter Enable for EXT_INT15[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[1] | [14:8] | RW | Filtering width of EXT_INT15[1] | 0x00 |
| FLTEN15[0] | [7] | RW | Filter Enable for EXT_INT15[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[0] | [6:0] | RW | Filtering width of EXT_INT15[0] | 0x00 |

4.3.2.113 EXT_INT15_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0874, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN15[7] | [31] | RW | Filter Enable for EXT_INT15[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[7] | [30:24] | RW | Filtering width of EXT_INT15[7] | 0x00 |
| FLTEN15[6] | [23] | RW | Filter Enable for EXT_INT15[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[6] | [22:16] | RW | Filtering width of EXT_INT15[6] | 0x00 |
| FLTEN15[5] | [15] | RW | Filter Enable for EXT_INT15[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[5] | [14:8] | RW | Filtering width of EXT_INT15[5] | 0x00 |
| FLTEN15[4] | [7] | RW | Filter Enable for EXT_INT15[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH15[4] | [6:0] | RW | Filtering width of EXT_INT15[4] | 0x00 |

4.3.2.114 EXT_INT16_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0878, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN16[3] | [31] | RW | Filter Enable for EXT_INT16[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[3] | [30:24] | RW | Filtering width of EXT_INT16[3] | 0x00 |
| FLTEN16[2] | [23] | RW | Filter Enable for EXT_INT16[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[2] | [22:16] | RW | Filtering width of EXT_INT16[2] | 0x00 |
| FLTEN16[1] | [15] | RW | Filter Enable for EXT_INT16[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[1] | [14:8] | RW | Filtering width of EXT_INT16[1] | 0x00 |
| FLTEN16[0] | [7] | RW | Filter Enable for EXT_INT16[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[0] | [6:0] | RW | Filtering width of EXT_INT16[0] | 0x00 |

4.3.2.115 EXT_INT16_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x087C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0x0000 |
| FLTEN16[5] | [15] | RW | Filter Enable for EXT_INT16[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[5] | [14:8] | RW | Filtering width of EXT_INT16[5] | 0x00 |
| FLTEN16[4] | [7] | RW | Filter Enable for EXT_INT16[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH16[4] | [6:0] | RW | Filtering width of EXT_INT16[4] | 0x00 |

4.3.2.116 EXT_INT21_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0880, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN17[3] | [31] | RW | Filter Enable for EXT_INT21[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[3] | [30:24] | RW | Filtering width of EXT_INT21[3] | 0x00 |
| FLTEN17[2] | [23] | RW | Filter Enable for EXT_INT21[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[2] | [22:16] | RW | Filtering width of EXT_INT21[2] | 0x00 |
| FLTEN17[1] | [15] | RW | Filter Enable for EXT_INT21[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[1] | [14:8] | RW | Filtering width of EXT_INT21[1] | 0x00 |
| FLTEN17[0] | [7] | RW | Filter Enable for EXT_INT21[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[0] | [6:0] | RW | Filtering width of EXT_INT21[0] | 0x00 |

4.3.2.117 EXT_INT21_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x0884, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN17[7] | [31] | RW | Filter Enable for EXT_INT21[7] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[7] | [30:24] | RW | Filtering width of EXT_INT21[7] | 0x00 |
| FLTEN17[6] | [23] | RW | Filter Enable for EXT_INT21[6] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[6] | [22:16] | RW | Filtering width of EXT_INT21[6] | 0x00 |
| FLTEN17[5] | [15] | RW | Filter Enable for EXT_INT21[5] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[5] | [14:8] | RW | Filtering width of EXT_INT21[5] | 0x00 |
| FLTEN17[4] | [7] | RW | Filter Enable for EXT_INT21[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH17[4] | [6:0] | RW | Filtering width of EXT_INT21[4] | 0x00 |

4.3.2.118 EXT_INT22_FLTCON0

- Base Address: 0x1140_0000
- Address = Base Address + 0x0888, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN18[3] | [31] | RW | Filter Enable for EXT_INT22[3] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH18[3] | [30:24] | RW | Filtering width of EXT_INT22[3] | 0x00 |
| FLTEN18[2] | [23] | RW | Filter Enable for EXT_INT22[2] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH18[2] | [22:16] | RW | Filtering width of EXT_INT22[2] | 0x00 |
| FLTEN18[1] | [15] | RW | Filter Enable for EXT_INT22[1] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH18[1] | [14:8] | RW | Filtering width of EXT_INT22[1] | 0x00 |
| FLTEN18[0] | [7] | RW | Filter Enable for EXT_INT22[0] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH18[0] | [6:0] | RW | Filtering width of EXT_INT22[0] | 0x00 |

4.3.2.119 EXT_INT22_FLTCON1

- Base Address: 0x1140_0000
- Address = Base Address + 0x088C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:8] | – | Reserved | 0x0000000 |
| FLTEN18[4] | [7] | RW | Filter Enable for EXT_INT22[4] 0x0 = Disables filter 0x1 = Enables filter | 0x0 |
| FLTWIDTH18[4] | [6:0] | RW | Filtering width of EXT_INT22[4] | 0x00 |

4.3.2.120 EXT_INT1_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | – | Reserved | 0x000000 |
| EXT_INT1_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT1_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.121 EXT_INT2_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0904, Reset Value = 0x0000_003F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:6] | – | Reserved | 0x00000000 |
| EXT_INT2_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT2_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT2_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT2_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT2_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT2_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.122 EXT_INT3_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | – | Reserved | 0x000000 |
| EXT_INT3_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT3_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.123 EXT_INT4_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_001F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:5] | – | Reserved | 0x00000000 |
| EXT_INT4_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT4_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT4_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT4_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT4_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.124 EXT_INT5_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_001F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:5] | – | Reserved | 0x00000000 |
| EXT_INT5_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT5_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT5_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT5_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT5_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.125 EXT_INT6_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0914, Reset Value = 0x0000_000F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:4] | – | Reserved | 0x00000000 |
| EXT_INT6_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT6_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT6_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT6_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.126 EXT_INT7_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0918, Reset Value = 0x0000_000F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:4] | – | Reserved | 0x00000000 |
| EXT_INT7_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT7_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT7_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT7_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.127 EXT_INT13_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0930, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT13_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT13_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.128 EXT_INT14_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0934, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT14_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT14_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.129 EXT_INT15_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0938, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT15_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT15_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.130 EXT_INT16_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x093C, Reset Value = 0x0000_003F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:6] | — | Reserved | 0x00000000 |
| EXT_INT16_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT16_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT16_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT16_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT16_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT16_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.131 EXT_INT21_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0940, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT21_MASK[7] | [7] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[6] | [6] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[5] | [5] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT21_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.132 EXT_INT22_MASK

- Base Address: 0x1140_0000
- Address = Base Address + 0x0944, Reset Value = 0x0000_001F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT22_MASK[4] | [4] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT22_MASK[3] | [3] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT22_MASK[2] | [2] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT22_MASK[1] | [1] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |
| EXT_INT22_MASK[0] | [0] | RW | 0x0 = Enables interrupt 0x1 = Masked | 0x1 |

4.3.2.133 EXT_INT1_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x0000000 |
| EXT_INT1_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT1_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.134 EXT_INT2_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:6] | — | Reserved | 0x00000000 |
| EXT_INT2_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT2_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT2_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT2_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT2_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT2_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.135 EXT_INT3_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT3_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT3_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.136 EXT_INT4_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT4_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT4_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT4_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT4_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT4_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.137 EXT_INT5_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT5_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT5_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT5_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT5_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT5_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.138 EXT_INT6_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A14, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:4] | — | Reserved | 0x00000000 |
| EXT_INT6_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT6_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT6_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT6_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.139 EXT_INT7_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A18, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:4] | — | Reserved | 0x00000000 |
| EXT_INT7_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT7_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT7_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT7_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.140 EXT_INT13_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0xA30, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT13_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT13_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.141 EXT_INT14_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0xA34, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT14_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT14_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.142 EXT_INT15_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A38, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x0000000 |
| EXT_INT15_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT15_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.143 EXT_INT16_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A3C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:6] | — | Reserved | 0x00000000 |
| EXT_INT16_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT16_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT16_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT16_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT16_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT16_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.144 EXT_INT21_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A40, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x0000000 |
| EXT_INT21_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT21_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.145 EXT_INT22_PEND

- Base Address: 0x1140_0000
- Address = Base Address + 0x0A44, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT22_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT22_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT22_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT22_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT22_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.2.146 EXT_INT_SERVICE_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0xB08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|--|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_Group_Num | [7:3] | RW | EXT_INT Service group number 0x1 = EXT_INT1 0x2 = EXT_INT2 0x3 = EXT_INT3 0x4 = EXT_INT4 0x5 = EXT_INT5 0x6 = EXT_INT6 0x7 = EXT_INT7 0x8 = EXT_INT13 0x9 = EXT_INT14 0xA = EXT_INT15 0xB = EXT_INT16 0xC = EXT_INT21 0xD = EXT_INT22 | 0x00 |
| SVC_Num | [2:0] | RW | Interrupt number to be serviced | 0x0 |

4.3.2.147 EXT_INT_SERVICE_PEND_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0xB0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_PEND | [7:0] | RW | 0x0 = Not occur 0x1 = Interrupt occurs | 0x00 |

4.3.2.148 EXT_INT_GRPFIXPRI_XB

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| RSVD | [31:4] | — | Reserved | 0x00000000 |
| Highest_GRP_NUM | [3:0] | RW | <p>When fixed group priority mode = 0 to 12, then group number should be of the highest priority.</p> <p>0x0 = EXT_INT1 0x1 = EXT_INT2 0x2 = EXT_INT3 0x3 = EXT_INT4 0x4 = EXT_INT5 0x5 = EXT_INT6 0x6 = EXT_INT7 0x7 = EXT_INT13 0x8 = EXT_INT14 0x9 = EXT_INT15 0xA = EXT_INT16 0xB = EXT_INT21 0xC = EXT_INT22</p> | 0x00 |

4.3.2.149 EXT_INT1_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B14, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT1) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.150 EXT_INT2_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B18, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT2) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.151 EXT_INT3_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B1C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT3) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.152 EXT_INT4_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT4) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.153 EXT_INT5_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0xB24, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT5) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.154 EXT_INT6_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0xB28, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 5 (EXT_INT6) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.155 EXT_INT7_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0xB2C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 6 (EXT_INT7) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.156 EXT_INT13_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B44, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 7 (EXT_INT13) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.157 EXT_INT14_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B48, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 8 (EXT_INT14) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.158 EXT_INT15_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B4C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 9 (EXT_INT15) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.159 EXT_INT16_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B50, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 10 (EXT_INT16) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.160 EXT_INT21_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B54, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 11 (EXT_INT21) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.161 EXT_INT22_FIXPRI

- Base Address: 0x1140_0000
- Address = Base Address + 0x0B58, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 12 (EXT_INT22) when fixed priority mode: 0 to 7 | 0x0 |

4.3.2.162 PDNEN

- Base Address: 0x1140_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|--|-------------|
| RSVD | [7:2] | – | Reserved | 0x00 |
| PDNEN_CFG | [1] | RW | 0 = Automatically by power down mode 1 = by PDNEN bit | 0x0 |
| PDNEN | [0] | RW | Power down mode pad state enable register. 0 = PADs Controlled by normal mode 1 = PADs Controlled by Power Down mode control registers This bit is set to "1" automatically when system enters into Power down mode and can be cleared by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" | 0x0 |

4.3.3 Part 2

For the following SFRs, Sets the value does not take effect immediately. It takes at least 800 APB clocks for the value to take effect after the SFR is actually changed. The SFRs are:

GPK0PUD, GPK0DRV, GPK1PUD, GPK1DRV, GPK2PUD, GPK2DRV, GPK3PUD, GPK3DRV, GPL0PUD, GPL0DRV, GPL1PUD, GPL1DRV, GPL2PUD, GPL2DRV, GPY0PUD, GPY0DRV, GPY1PUD, GPY1DRV, GPY2PUD, GPY2DRV, GPY3PUD, GPY3DRV, GPY4PUD, GPY4DRV, GPY5PUD, GPY5DRV, GPY6PUD, GPY6DRV, GPM0PUD, GPM0DRV, GPM1PUD, GPM1DRV, GPM2PUD, GPM2DRV, GPM3PUD, GPM3DRV, GPM4PUD, GPM4DRV.

4.3.3.1 GPK0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPK0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[3] 0x3 = SD_4_DATA[3] 0x4 to 0xE = Reserved 0xF = EXT_INT23[6] | 0x00 |
| GPK0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[2] 0x3 = SD_4_DATA[2] 0x4 to 0xE = Reserved 0xF = EXT_INT23[5] | 0x00 |
| GPK0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[1] 0x3 = SD_4_DATA[1] 0x4 to 0xE = Reserved 0xF = EXT_INT23[4] | 0x00 |
| GPK0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_DATA[0] 0x3 = SD_4_DATA[0] 0x4 to 0xE = Reserved 0xF = EXT_INT23[3] | 0x00 |
| GPK0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_CDn 0x3 = SD_4_CDn 0x4 = GNSS_GPIO[8] 0x5 to 0xE = Reserved 0xF = EXT_INT23[2] | 0x00 |
| GPK0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_CMD 0x3 = SD_4_CMD 0x4 to 0xE = Reserved 0xF = EXT_INT23[1] | 0x00 |
| GPK0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_0_CLK 0x3 = SD_4_CLK 0x4 to 0xE = Reserved 0xF = EXT_INT23[0] | 0x00 |

4.3.3.2 GPK0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0044, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPK0DAT[6:0] | [6:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.3 GPK0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0048, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK0PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.4 GPK0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_2AAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x2AAA |

4.3.3.5 GPK0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPK0[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.6 GPK0PUPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPK0[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.7 GPK1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPK1CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[3] 0x3 = SD_0_DATA[7] 0x4 = SD_4_DATA[7] 0x5 to 0xE = Reserved 0xF = EXT_INT24[6] | 0x00 |
| GPK1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[2] 0x3 = SD_0_DATA[6] 0x4 = SD_4_DATA[6] 0x5 to 0xE = Reserved 0xF = EXT_INT24[5] | 0x00 |
| GPK1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[1] 0x3 = SD_0_DATA[5] 0x4 = SD_4_DATA[5] 0x5 to 0xE = Reserved 0xF = EXT_INT24[4] | 0x00 |
| GPK1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_DATA[0] 0x3 = SD_0_DATA[4] 0x4 = SD_4_DATA[4] 0x5 to 0xE = Reserved 0xF = EXT_INT24[3] | 0x00 |
| GPK1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_CDn 0x3 = GNSS_GPIO[9] 0x4 = SD_4_nRESET_OUT 0x5 to 0xE = Reserved 0xF = EXT_INT24[2] | 0x00 |
| GPK1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT24[1] | 0x00 |
| GPK1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_1_CLK 0x3 to 0xE = Reserved | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|--------------------|-------------|
| | | | 0xF = EXT_INT24[0] | |

4.3.3.8 GPK1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPK1DAT[6:0] | [6:0] | RWX | When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.9 GPK1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0068, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK1PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.10 GPK1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.11 GPK1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPK1[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.12 GPK1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPK1[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.13 GPK2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPK2CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[3] 0x3 to 0xE = Reserved 0xF = EXT_INT25[6] | 0x00 |
| GPK2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[2] 0x3 to 0xE = Reserved 0xF = EXT_INT25[5] | 0x00 |
| GPK2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[1] 0x3 to 0xE = Reserved 0xF = EXT_INT25[4] | 0x00 |
| GPK2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_DATA[0] 0x3 to 0xE = Reserved 0xF = EXT_INT25[3] | 0x00 |
| GPK2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_CDn 0x3 = GNSS_GPIO[10] 0x4 to 0xE = Reserved 0xF = EXT_INT25[2] | 0x00 |
| GPK2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT25[1] | 0x00 |
| GPK2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_2_CLK 0x3 to 0xE = Reserved 0xF = EXT_INT25[0] | 0x00 |

4.3.3.14 GPK2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPK2DAT[6:0] | [6:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.15 GPK2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0088, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK2PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.16 GPK2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.17 GPK2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPK2[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.18 GPK2PUPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPK2[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Enables Reserved 0x3 = Pull-up | 0x00 |

4.3.3.19 GPK3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPK3CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[3] 0x3 = SD_2_DATA[7] 0x4 to 0xE = Reserved 0xF = EXT_INT26[6] | 0x00 |
| GPK3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[2] 0x3 = SD_2_DATA[6] 0x4 to 0xE = Reserved 0xF = EXT_INT26[5] | 0x00 |
| GPK3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[1] 0x3 = SD_2_DATA[5] 0x4 to 0xE = Reserved 0xF = EXT_INT26[4] | 0x00 |
| GPK3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_DATA[0] 0x3 = SD_2_DATA[4] 0x4 to 0xE = Reserved 0xF = EXT_INT26[3] | 0x00 |
| GPK3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_CDn 0x3 = GNSS_GPIO[11] 0x4 to 0xE = Reserved 0xF = EXT_INT26[2] | 0x00 |
| GPK3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_CMD 0x3 to 0xE = Reserved 0xF = EXT_INT26[1] | 0x00 |
| GPK3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = SD_3_CLK 0x3 to 0xE = Reserved 0xF = EXT_INT26[0] | 0x00 |

4.3.3.20 GPK3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPK3DAT[6:0] | [6:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.21 GPK3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00A8, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK3PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.22 GPK3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPK3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.23 GPK3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00B0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPK3[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.24 GPK3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00B4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPK3[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.25 GPL0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPL0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_RF_RSTN 0x3 to 0xE = Reserved 0xF = EXT_INT27[6] | 0x00 |
| GPL0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 to 0xE = Reserved 0xF = EXT_INT27[5] | 0x00 |
| GPL0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_QMAG 0x3 to 0xE = Reserved 0xF = EXT_INT27[4] | 0x00 |
| GPL0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_QSIGN 0x3 to 0xE = Reserved 0xF = EXT_INT27[3] | 0x00 |
| GPL0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_IMAG 0x3 to 0xE = Reserved 0xF = EXT_INT27[2] | 0x00 |
| GPL0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_ISIGN 0x3 to 0xE = Reserved 0xF = EXT_INT27[1] | 0x00 |
| GPL0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_SYNC 0x3 to 0xE = Reserved 0xF = EXT_INT27[0] | 0x00 |

4.3.3.26 GPL0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPL0DAT[6:0] | [6:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.27 GPL0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00C8, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPL0PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.28 GPL0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPL0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.29 GPL0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPL0[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.30 GPL0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPL0[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.31 GPL1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| GPL1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_SDA 0x3 to 0xE = Reserved 0xF = EXT_INT28[1] | 0x00 |
| GPL1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_SCL 0x3 to 0xE = Reserved 0xF = EXT_INT28[0] | 0x00 |

4.3.3.32 GPL1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPL1DAT[1:0] | [1:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.33 GPL1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x00E8, Reset Value = 0x0005

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPL1PUD[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0005 |

4.3.3.34 GPL1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x00EC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPL1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.35 GPL1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPL1[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.36 GPL1PUPDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x00F4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPL1[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.37 GPL2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPL2CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[7] 0x3 = KP_COL[7] 0x4 to 0xE = Reserved 0xF = EXT_INT29[7] | 0x00 |
| GPL2CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[6] 0x3 = KP_COL[6] 0x4 to 0xE = Reserved 0xF = EXT_INT29[6] | 0x00 |
| GPL2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[5] 0x3 = KP_COL[5] 0x4 to 0xE = Reserved 0xF = EXT_INT29[5] | 0x00 |
| GPL2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[4] 0x3 = KP_COL[4] 0x4 to 0xE = Reserved 0xF = EXT_INT29[4] | 0x00 |
| GPL2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[3] 0x3 = KP_COL[3] 0x4 to 0xE = Reserved 0xF = EXT_INT29[3] | 0x00 |
| GPL2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[2] 0x3 = KP_COL[2] 0x4 to 0xE = Reserved 0xF = EXT_INT29[2] | 0x00 |
| GPL2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = GNSS_GPIO[1] 0x3 = KP_COL[1] 0x4 to 0xE = Reserved 0xF = EXT_INT29[1] | 0x00 |
| GPL2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|--|-------------|
| | | | 0x2 = GNSS_GPIO[0] 0x3 = KP_COL[0] 0x4 to 0xE = Reserved 0xF = EXT_INT29[0] | |

4.3.3.38 GPL2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0104, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPL2DAT[7:0] | [7:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.39 GPL2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0108, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPL2PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up | 0x5555 |

4.3.3.40 GPL2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x010C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPL2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.41 GPL2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPL2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.42 GPL2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0114, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPL2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.43 GPY0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0120, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPY0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_WEn 0x4 to 0xF = Reserved | 0x00 |
| GPY0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_OEn 0x4 to 0xF = Reserved | 0x00 |
| GPY0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = SROM_CSn[3] 0x3 = NF_CSn[1] 0x4 = Reserved 0x5 = OND_CSn[1] 0x4 to 0xF = Reserved | 0x00 |
| GPY0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SROM_CSn[2] 0x3 = NF_CSn[0] 0x4 = Reserved 0x5 = OND_CSn[0] 0x4 to 0xF = Reserved | 0x00 |
| GPY0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = SROM_CSn[1] 0x3 = NF_CSn[3] 0x4 to 0xF = Reserved | 0x00 |
| GPY0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = SROM_CSn[0] 0x3 = NF_CSn[2] 0x4 to 0xF = Reserved | 0x00 |

4.3.3.44 GPY0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0124, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPY0DAT[5:0] | [5:0] | RWX | When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.45 GPY0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0128, Reset Value = 0x0FFF

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY0PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0FFF |

4.3.3.46 GPY0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x012C, Reset Value = 0x00_0AAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0AAA |

4.3.3.47 GPY0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0130, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY0[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.48 GPY0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0134, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY0[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.49 GPY1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0140, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPY1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA_RDn 0x4 to 0xF = Reserved | 0x00 |
| GPY1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = SROM_WAITn 0x4 to 0xF = Reserved | 0x00 |
| GPY1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_BEn[1] 0x4 to 0xF = Reserved | 0x00 |
| GPY1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_BEn[0] 0x4 to 0xF = Reserved | 0x00 |

4.3.3.50 GPY1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0144, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPY1DAT[3:0] | [3:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.51 GPY1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0148, Reset Value = 0x00FF

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPY1PUD[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up | 0x00FF |

4.3.3.52 GPY1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x014C, Reset Value = 0x00_00AA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x00AA |

4.3.3.53 GPY1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0150, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY1[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.54 GPY1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0154, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY1[n] | [2n + 1:2n] n = 0 to 3 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.55 GPY2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0160, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPY2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_RnB[3] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_RnB[2] 0x3= Reserved 0x4= Reserved 0x5 = OND_RPn 0x6 to 0xE = Reserved 0xF = - | 0x00 |
| GPY2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_RnB[1] 0x3= Reserved 0x4= Reserved 0x5 = OND_INT[1] 0x6 to 0xE = Reserved 0xF = - | 0x00 |
| GPY2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_RnB[0] 0x3= Reserved 0x4= Reserved 0x5 = OND_INT[0] 0x6 to 0xE = Reserved 0xF = - | 0x00 |
| GPY2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_ALE 0x3= Reserved 0x4= Reserved 0x5 = OND_SMCLK 0x6 to 0xE = Reserved 0xF = - | 0x00 |
| GPY2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = NF_CLE 0x3= Reserved 0x4= Reserved 0x5 = OND_ADDRVALID 0x6 to 0xE = Reserved | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|-------------|-------------|
| | | | 0xF = - | |

4.3.3.56 GPY2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0164, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPY2DAT[5:0] | [5:0] | RWX | When you configure port as input port, the corresponding bit is the pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.57 GPY2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0168, Reset Value = 0x0FFF

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY2PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0 FFF |

4.3.3.58 GPY2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x016C, Reset Value = 0x00_0AAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0AAA |

4.3.3.59 GPY2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0170, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY2[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.60 GPY2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0174, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY2[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.61 GPY3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPY3CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[7] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output, 0x2 = EBI_ADDR[6] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[5] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[4] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[3] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[2] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[1] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[0] 0x3 to 0xE = Reserved 0xF = - | 0x00 |

4.3.3.62 GPY3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0184, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPY3DAT[7:0] | [7:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port, the pin state should be same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.63 GPY3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0188, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY3PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.64 GPY3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x018C, Reset Value = 0x00_AAAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0xAAAA |

4.3.3.65 GPY3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.66 GPY3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0194, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.67 GPY4CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPY4CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[15] 0x3= Reserved 0x4 = XhsiCAREADY 0x5 to 0xE = Reserved, 0xF = – | 0x00 |
| GPY4CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[14] 0x3= Reserved 0x4 = XhsiACFLAG 0x5 to 0xE = Reserved, 0xF = – | 0x00 |
| GPY4CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[13] 0x3= Reserved 0x4 = XhsiACDATA 0x5 to 0xE = Reserved 0xF = – | 0x00 |
| GPY4CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[12] 0x3= Reserved 0x4 = XhsiACWAKE 0x5 to 0xE = Reserved 0xF = – | 0x00 |
| GPY4CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[11] 0x3= Reserved 0x4 = XhsiACREADY 0x5 to 0xE = Reserved 0xF = – | 0x00 |
| GPY4CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[10] 0x3= Reserved 0x4 = XhsiCAFLAG 0x5 to 0xE = Reserved 0xF = – | 0x00 |
| GPY4CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|---|-------------|
| | | | 0x2 = EBI_ADDR[9] 0x3= Reserved 0x4 = XhsiCADATA 0x5 to 0xE = Reserved 0xF = - | |
| GPY4CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_ADDR[8] 0x3= Reserved 0x4 = XhsiCAWAKE 0x5 to 0xE = Reserved 0xF = - | 0x00 |

4.3.3.68 GPY4DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPY4DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.69 GPY4PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01A8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY4PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.70 GPY4DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01AC, Reset Value = 0x00_AAAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY4DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0xAAAA |

4.3.3.71 GPY4CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01B0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY4[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.72 GPY4PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01B4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY4[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.73 GPY5CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPY5CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[7] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[6] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[5] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[4] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[3] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[2] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[1] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY5CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[0] 0x3 to 0xE = Reserved 0xF = - | 0x00 |

4.3.3.74 GPY5DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|--|-------------|
| GPY5DAT[7:0] | [7:0] | RWX | When you configure port as input port, then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.75 GPY5PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01C8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPY5PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up | 0x5555 |

4.3.3.76 GPY5DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01CC, Reset Value = 0x00_AAAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY5DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0xAAAA |

4.3.3.77 GPY5CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY5[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.78 GPY5PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY5[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Disables Pull-up | 0x00 |

4.3.3.79 GPY6CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPY6CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[15] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[14] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[13] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[12] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[11] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[10] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[9] 0x3 to 0xE = Reserved 0xF = - | 0x00 |
| GPY6CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = EBI_DATA[8] 0x3 to 0xE = Reserved 0xF = - | 0x00 |

4.3.3.80 GPY6DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPY6DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.81 GPY6PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x01E8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY6PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.82 GPY6DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x01EC, Reset Value = 0x00_AAAA

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPY6DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0xAAAA |

4.3.3.83 GPY6CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01F0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPY6[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.84 GPY6PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x01F4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPY6[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.85 ETC0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0208, Reset Value = 0x0400

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC0PUD[n] | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0400 |

ETC0PUD[1:0] controls XjTRSTn.

ETC0PUD[3:2] controls XjTMS.

ETC0PUD[5:4] controls XjTCK.

ETC0PUD[7:6] controls XjTDI.

ETC0PUD[9:8] controls XjTDO.

ETC0PUD[11:10] controls XjDBGSEL.

4.3.3.86 ETC0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x020C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 5 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

ETC0DRV[1:0] controls XjTRSTn.

ETC0DRV[3:2] controls XjTMS.

ETC0DRV[5:4] controls XjTCK.

ETC0DRV[7:6] controls XjTDI.

ETC0DRV[9:8] controls XjTDO.

ETC0DRV[11:10] controls XjDBGSEL.

4.3.3.87 ETC6PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0228, Reset Value = 0xC000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC6PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0xC000 |

ETC6PUD[1:0] controls XnRESET.

ETC6PUD[3:2] controls XCLKOUT.

ETC6PUD[5:4] controls XnRSTOUT.

ETC6PUD[9:8] controls XRTCCLKO.

ETC6PUD[11:10] controls XuotgDRVVBUS.

ETC6PUD[13:12] controls XuhostPWREN.

ETC6PUD[15:14] controls XuhostOVERCUR.

4.3.3.88 ETC6DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x022C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC6DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

ETC6DRV[1:0] controls XnRESET.

ETC6DRV[3:2] controls XCLKOUT.

ETC6DRV[5:4] controls XnRSTOUT.

ETC6DRV[7:6] controls XnWRESET.

ETC6DRV[9:8] controls XRTCCLKO.

ETC6DRV[11:10] controls XuotgDRVVBUS.

ETC6DRV[13:12] controls XuhostPWREN.

ETC6DRV[15:14] controls XuhostOVERCUR.

4.3.3.89 GPM0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPM0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[6] 0x4 = XhsiCAFLAG 0x5 = TraceData[6] 0x6 to 0xE = Reserved 0xF = EXT_INT8[7] | 0x00 |
| GPM0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[5] 0x4 = XhsiCADATA 0x5 = TraceData[5] 0x6 to 0xE = Reserved 0xF = EXT_INT8[6] | 0x00 |
| GPM0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output, 0x2 = Reserved 0x3 = CAM_B_DATA[4], 0x4 = XhsiCAWAKE, 0x5 = TraceData[4], 0x6 to 0xE = Reserved, 0xF = EXT_INT8[5] | 0x00 |
| GPM0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[3] 0x4 = TS_ERROR 0x5 = TraceData[3] 0x6 to 0xE = Reserved 0xF = EXT_INT8[4] | 0x00 |
| GPM0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[2] 0x4 = TS_DATA 0x5 = TraceData[2] 0x6 to 0xE = Reserved 0xF = EXT_INT8[3] | 0x00 |
| GPM0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[1] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|---|-------------|
| | | | 0x4 = TS_VAL 0x5 = TraceData[1] 0x6 to 0xE = Reserved 0xF = EXT_INT8[2] | |
| GPM0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[0] 0x4 = TS_SYNC 0x5 = TraceData[0] 0x6 to 0xE = Reserved 0xF = EXT_INT8[1] | 0x00 |
| GPM0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_PCLK 0x4 = TS_CLK 0x5 = TraceClk 0x6 to 0xE = Reserved 0xF = EXT_INT8[0] | 0x00 |

4.3.3.90 GPM0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0264, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPM0DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.91 GPM0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0268, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM0PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.92 GPM0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x026C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.93 GPM0CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0270, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPM0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.94 GPM0PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0274, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPM0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.95 GPM1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0280, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPM1CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[13] 0x3 = Reserved 0x4 = Reserved 0x5 = TraceData[12] 0x6 to 0xE = Reserved 0xF = EXT_INT9[6] | 0x00 |
| GPM1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[12] 0x3 = Reserved 0x4 = Reserved 0x5 = TraceData[11] 0x6 to 0xE = Reserved 0xF = EXT_INT9[5] | 0x00 |
| GPM1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[11] 0x3 = Reserved 0x4 = XhsiCAREADY 0x5 = TraceData[10] 0x6 to 0xE = Reserved 0xF = EXT_INT9[4] | 0x00 |
| GPM1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[10] 0x3 = Reserved 0x4 = XhsiACFLAG 0x5 = TraceData[9] 0x6 to 0xE = Reserved 0xF = EXT_INT9[3] | 0x00 |
| GPM1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[9] 0x3 = Reserved 0x4 = XhsiACDATA 0x5 = TraceData[8] 0x6 to 0xE = Reserved 0xF = EXT_INT9[2] | 0x00 |
| GPM1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_BAY_RGB[8] 0x3 = CAM_B_FIELD | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|---|-------------|
| | | | 0x4 = XhsiACWAKE 0x5 = TraceCtl 0x6 to 0xE = Reserved 0xF = EXT_INT9[1] | |
| GPM1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_DATA[7] 0x4 = XhsiACREADY 0x5 = TraceData[7] 0x6 to 0xE = Reserved 0xF = EXT_INT9[0] | 0x00 |

4.3.3.96 GPM1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0284, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPM1DAT[6:0] | [6:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.97 GPM1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0288, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM1PUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.3.98 GPM1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x028C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.99 GPM1CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0290, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPM1[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.100 GPM1PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0294, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPM1[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.101 GPM2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPM2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[1] 0x3 = MPWM2_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT10[4] | 0x00 |
| GPM2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[0] 0x3 = MPWM1_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT10[3] | 0x00 |
| GPM2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_CLKOUT 0x4 = Reserved 0x5 = TraceData[15] 0x6 to 0xE = Reserved 0xF = EXT_INT10[2] | 0x00 |
| GPM2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_HREF 0x4 = Reserved 0x5 = TraceData[14] 0x6 to 0xE = Reserved 0xF = EXT_INT10[1] | 0x00 |
| GPM2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = CAM_B_VSYNC 0x4 = Reserved 0x5 = TraceData[13] 0x6 to 0xE = Reserved 0xF = EXT_INT10[0] | 0x00 |

4.3.3.102 GPM2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPM2DAT[4:0] | [4:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.103 GPM2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02A8, Reset Value = 0x0155

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM2PUD[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0155 |

4.3.3.104 GPM2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02AC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.105 GPM2CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02B0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPM2[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.106 GPM2PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02B4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPM2[n] | [2n + 1:2n] n = 0 to 4 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.107 GPM3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPM3CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[9] 0x3 = RXD_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[7] | 0x00 |
| GPM3CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[8] 0x3 = nCTS_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[6] | 0x00 |
| GPM3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[7] 0x3 = TXD_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[5] | 0x00 |
| GPM3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[6] 0x3 = nRTS_UART_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[4] | 0x00 |
| GPM3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[5] 0x3 = MPWM6_OUT_ISP 0x4 = CAM_SPI1_MOSI 0x5 to 0xE = Reserved 0xF = EXT_INT11[3] | 0x00 |
| GPM3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[4] 0x3 = MPWM5_OUT_ISP 0x4 = CAM_SPI1_MISO 0x5 to 0xE = Reserved 0xF = EXT_INT11[2] | 0x00 |
| GPM3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[3] 0x3 = MPWM4_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[1] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| GPM3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_GPIO[2] 0x3 = MPWM3_OUT_ISP 0x4 to 0xE = Reserved 0xF = EXT_INT11[0] | 0x00 |

4.3.3.108 GPM3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPM3DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.109 GPM3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02C8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM3PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.110 GPM3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02CC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.111 GPM3CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPM3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.112 GPM3PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPM3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.113 GPM4CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPM4CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_SPI_MOSI 0x3 = CAM_GPIO[17] 0x4 to 0xE = Reserved 0xF = EXT_INT12[7] | 0x00 |
| GPM4CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_SPI_MISO 0x3 = CAM_GPIO[16] 0x4 to 0xE = Reserved 0xF = EXT_INT12[6] | 0x00 |
| GPM4CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_SPI_nSS 0x3 = CAM_GPIO[15] 0x4 to 0xE = Reserved 0xF = EXT_INT12[5] | 0x00 |
| GPM4CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_SPI_CLK 0x3 = CAM_GPIO[14] 0x4 to 0xE = Reserved 0xF = EXT_INT12[4] | 0x00 |
| GPM4CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_I2C1_SDA 0x3 = CAM_GPIO[13] 0x4 = CAM_SPI1_nSS 0x5 to 0xE = Reserved 0xF = EXT_INT12[3] | 0x00 |
| GPM4CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_I2C1_SCL 0x3 = CAM_GPIO[12] 0x4 = CAM_SPI1_CLK 0x5 to 0xE = Reserved 0xF = EXT_INT12[2] | 0x00 |
| GPM4CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_I2C0_SDA 0x3 = CAM_GPIO[11] 0x4 to 0xE = Reserved 0xF = EXT_INT12[1] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| GPM4CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = CAM_I2C0_SCL 0x3 = CAM_GPIO[10] 0x4 to 0xE = Reserved 0xF = EXT_INT12[0] | 0x00 |

4.3.3.114 GPM4DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPM4DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.115 GPM4PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x02E8, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM4PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.116 GPM4DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x02EC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPM4DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.117 GPM4CONPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02F0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPM4[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.3.118 GPM4PUDPDN

- Base Address: 0x1100_0000
- Address = Base Address + 0x02F4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPM4[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.3.119 EXT_INT23CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT23_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT23[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT23_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT23[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT23_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT23[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT23_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT23[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT23_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT23[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT23_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT23[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT23_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT23[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.120 EXT_INT24CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT24_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT24[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT24_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT24[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT24_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT24[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT24_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT24[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT24_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT24[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT24_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT24[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT24_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT24[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.121 EXT_INT25CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT25_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT25[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT25_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT25[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT25_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT25[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT25_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT25[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT25_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT25[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|---|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT25_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT25[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT25_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT25[0] 0x0 = Low level 0x1 = High level, 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.122 EXT_INT26CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0714, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT26_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT26[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT26_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT26[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT26_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT26[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT26_CON[3] | [14:12] | W | Sets signaling method of EXT_INT26[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT26_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT26[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT26_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT26[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT26_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT26[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.123 EXT_INT27CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0718, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT27_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT27[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT27_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT27[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT27_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT27[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT27_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT27[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT27_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT27[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT27_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT27[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT27_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT27[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.124 EXT_INT28CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x071C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:8] | – | Reserved | 0x000000 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT28_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT28[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT28_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT28[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.125 EXT_INT29CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0720, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT29_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT29[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT29_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT29[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT29_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT29[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT29_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT29[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT29_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT29[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT29_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT29[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT29_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT29[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT29_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT29[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.126 EXT_INT8CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0724, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT8_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT8[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT8_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT8[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT8_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT8[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT8_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT8[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT8_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT8[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| EXT_INT8_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT8[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT8_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT8[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT8_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT8[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.127 EXT_INT9CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0728, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| RSVD | [27] | – | Reserved | 0x0 |
| EXT_INT9_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT9[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | – | Reserved | 0x0 |
| EXT_INT9_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT9[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT9_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT9[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT9_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT9[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT9_CON[2] | [10:8] | W | Sets signaling method of EXT_INT9[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|-------|------|---|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT9_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT9[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT9_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT9[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.128 EXT_INT10CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x072C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:20] | – | Reserved | 0x000 |
| RSVD | [19] | – | Reserved | 0x0 |
| EXT_INT10_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT10[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | – | Reserved | 0x0 |
| EXT_INT10_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT10[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | – | Reserved | 0x0 |
| EXT_INT10_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT10[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | – | Reserved | 0x0 |
| EXT_INT10_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT10[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| EXT_INT10_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT10[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers falling edge 0x3 = Triggers rising edge 0x4 = Triggers both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.129 EXT_INT11CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0730, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT11_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT11[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT11_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT11[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT11_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT11[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT11_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT11[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT11_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT11[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Rising edge triggered 0x4 = Both edge triggered 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT11_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT11[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT11_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT11[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT11_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT11[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.130 EXT_INT12CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0734, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT12_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT12[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT12_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT12[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT12_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT12[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT12_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT12[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT12_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT12[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT12_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT12[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT12_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT12[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT12_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT12[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.131 EXT_INT23_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN3[3] | [31] | RW | Filter Enable for EXT_INT23[3] 0x0 = Disables Filter 0x1 = Enabled Filter | 0x0 |
| FLTWIDTH3[3] | [30:24] | RW | Filtering width of EXT_INT23[3] | 0x00 |
| FLTEN3[2] | [23] | RW | Filter Enable for EXT_INT23[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[2] | [22:16] | RW | Filtering width of EXT_INT23[2] | 0x00 |
| FLTEN3[1] | [15] | RW | Filter Enable for EXT_INT23[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[1] | [14:8] | RW | Filtering width of EXT_INT23[1] | 0x00 |
| FLTEN3[0] | [7] | RW | Filter Enable for EXT_INT23[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[0] | [6:0] | RW | Filtering width of EXT_INT23[0] | 0x00 |

4.3.3.132 EXT_INT23_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN3[6] | [23] | RW | Filter Enable for EXT_INT23[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[6] | [22:16] | RW | Filtering width of EXT_INT23[6] | 0x00 |
| FLTEN3[5] | [15] | RW | Filter Enable for EXT_INT23[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[5] | [14:8] | RW | Filtering width of EXT_INT23[5] | 0x00 |
| FLTEN3[4] | [7] | RW | Filter Enable for EXT_INT23[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[4] | [6:0] | RW | Filtering width of EXT_INT23[4] | 0x00 |

4.3.3.133 EXT_INT24_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN4[3] | [31] | RW | Filter Enable for EXT_INT24[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[3] | [30:24] | RW | Filtering width of EXT_INT24[3] | 0x00 |
| FLTEN4[2] | [23] | RW | Filter Enable for EXT_INT24[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[2] | [22:16] | RW | Filtering width of EXT_INT24[2] | 0x00 |
| FLTEN4[1] | [15] | RW | Filter Enable for EXT_INT24[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[1] | [14:8] | RW | Filtering width of EXT_INT24[1] | 0x00 |
| FLTEN4[0] | [7] | RW | Filter Enable for EXT_INT24[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[0] | [6:0] | RW | Filtering width of EXT_INT24[0] | 0x00 |

4.3.3.134 EXT_INT24_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN4[6] | [23] | RW | Filter Enable for EXT_INT24[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[6] | [22:16] | RW | Filtering width of EXT_INT24[6] | 0x00 |
| FLTEN4[5] | [15] | RW | Filter Enable for EXT_INT24[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[5] | [14:8] | RW | Filtering width of EXT_INT24[5] | 0x00 |
| FLTEN4[4] | [7] | RW | Filter Enable for EXT_INT24[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[4] | [6:0] | RW | Filtering width of EXT_INT24[4] | 0x00 |

4.3.3.135 EXT_INT25_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN5[3] | [31] | RW | Filter Enable for EXT_INT25[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[3] | [30:24] | RW | Filtering width of EXT_INT25[3] | 0x00 |
| FLTEN5[2] | [23] | RW | Filter Enable for EXT_INT25[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[2] | [22:16] | RW | Filtering width of EXT_INT25[2] | 0x00 |
| FLTEN5[1] | [15] | RW | Filter Enable for EXT_INT25[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[1] | [14:8] | RW | Filtering width of EXT_INT25[1] | 0x00 |
| FLTEN5[0] | [7] | RW | Filter Enable for EXT_INT25[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[0] | [6:0] | RW | Filtering width of EXT_INT25[0] | 0x00 |

4.3.3.136 EXT_INT25_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN5[6] | [23] | RW | Filter Enable for EXT_INT25[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[6] | [22:16] | RW | Filtering width of EXT_INT25[6] | 0x00 |
| FLTEN5[5] | [15] | RW | Filter Enable for EXT_INT25[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[5] | [14:8] | RW | Filtering width of EXT_INT25[5] | 0x00 |
| FLTEN5[4] | [7] | RW | Filter Enable for EXT_INT25[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[4] | [6:0] | RW | Filtering width of EXT_INT25[4] | 0x00 |

4.3.3.137 EXT_INT26_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0828, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN6[3] | [31] | RW | Filter Enable for EXT_INT26[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[3] | [30:24] | RW | Filtering width of EXT_INT26[3] | 0x00 |
| FLTEN6[2] | [23] | RW | Filter Enable for EXT_INT26[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[2] | [22:16] | RW | Filtering width of EXT_INT26[2] | 0x00 |
| FLTEN6[1] | [15] | RW | Filter Enable for EXT_INT26[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[1] | [14:8] | RW | Filtering width of EXT_INT26[1] | 0x00 |
| FLTEN6[0] | [7] | RW | Filter Enable for EXT_INT26[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[0] | [6:0] | RW | Filtering width of EXT_INT26[0] | 0x00 |

4.3.3.138 EXT_INT26_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x082C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN6[6] | [23] | RW | Filter Enable for EXT_INT26[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[6] | [22:16] | RW | Filtering width of EXT_INT26[6] | 0x00 |
| FLTEN6[5] | [15] | RW | Filter Enable for EXT_INT26[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[5] | [14:8] | RW | Filtering width of EXT_INT26[5] | 0x00 |
| FLTEN6[4] | [7] | RW | Filter Enable for EXT_INT26[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH6[4] | [6:0] | RW | Filtering width of EXT_INT26[4] | 0x00 |

4.3.3.139 EXT_INT27_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0830, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN7[3] | [31] | RW | Filter Enable for EXT_INT27[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[3] | [30:24] | RW | Filtering width of EXT_INT27[3] | 0x00 |
| FLTEN7[2] | [23] | RW | Filter Enable for EXT_INT27[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[2] | [22:16] | RW | Filtering width of EXT_INT27[2] | 0x00 |
| FLTEN7[1] | [15] | RW | Filter Enable for EXT_INT27[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[1] | [14:8] | RW | Filtering width of EXT_INT27[1] | 0x00 |
| FLTEN7[0] | [7] | RW | Filter Enable for EXT_INT27[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[0] | [6:0] | RW | Filtering width of EXT_INT27[0] | 0x00 |

4.3.3.140 EXT_INT27_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0834, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN7[6] | [23] | RW | Filter Enable for EXT_INT27[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[6] | [22:16] | RW | Filtering width of EXT_INT27[6] | 0x00 |
| FLTEN7[5] | [15] | RW | Filter Enable for EXT_INT27[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[5] | [14:8] | RW | Filtering width of EXT_INT27[5] | 0x00 |
| FLTEN7[4] | [7] | RW | Filter Enable for EXT_INT27[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH7[4] | [6:0] | RW | Filtering width of EXT_INT27[4] | 0x00 |

4.3.3.141 EXT_INT28_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0838, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0 |
| FLTEN8[1] | [15] | RW | Filter Enable for EXT_INT28[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH8[1] | [14:8] | RW | Filtering width of EXT_INT28[1] | 0x00 |
| FLTEN8[0] | [7] | RW | Filter Enable for EXT_INT28[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH8[0] | [6:0] | RW | Filtering width of EXT_INT28[0] | 0x00 |

4.3.3.142 EXT_INT28_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x083C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|-------------|-------------|
| RSVD | [31:0] | – | Reserved | 0x00000000 |

4.3.3.143 EXT_INT29_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0840, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN9[3] | [31] | RW | Filter Enable for EXT_INT29[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[3] | [30:24] | RW | Filtering width of EXT_INT29[3] | 0x00 |
| FLTEN9[2] | [23] | RW | Filter Enable for EXT_INT29[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[2] | [22:16] | RW | Filtering width of EXT_INT29[2] | 0x00 |
| FLTEN9[1] | [15] | RW | Filter Enable for EXT_INT29[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[1] | [14:8] | RW | Filtering width of EXT_INT29[1] | 0x00 |
| FLTEN9[0] | [7] | RW | Filter Enable for EXT_INT29[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[0] | [6:0] | RW | Filtering width of EXT_INT29[0] | 0x00 |

4.3.3.144 EXT_INT29_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0844, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN9[7] | [31] | RW | Filter Enable for EXT_INT29[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[7] | [30:24] | RW | Filtering width of EXT_INT29[7] | 0x00 |
| FLTEN9[6] | [23] | RW | Filter Enable for EXT_INT29[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[6] | [22:16] | RW | Filtering width of EXT_INT29[6] | 0x00 |
| FLTEN9[5] | [15] | RW | Filter Enable for EXT_INT29[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[5] | [14:8] | RW | Filtering width of EXT_INT29[5] | 0x00 |
| FLTEN9[4] | [7] | RW | Filter Enable for EXT_INT29[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH9[4] | [6:0] | RW | Filtering width of EXT_INT29[4] | 0x00 |

4.3.3.145 EXT_INT8_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0848, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN10[3] | [31] | RW | Filter Enable for EXT_INT8[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[3] | [30:24] | RW | Filtering width of EXT_INT8[3] | 0x00 |
| FLTEN10[2] | [23] | RW | Filter Enable for EXT_INT8[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[2] | [22:16] | RW | Filtering width of EXT_INT8[2] | 0x00 |
| FLTEN10[1] | [15] | RW | Filter Enable for EXT_INT8[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[1] | [14:8] | RW | Filtering width of EXT_INT8[1] | 0x00 |
| FLTEN10[0] | [7] | RW | Filter Enable for EXT_INT8[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[0] | [6:0] | RW | Filtering width of EXT_INT8[0] | 0x00 |

4.3.3.146 EXT_INT8_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x084C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN10[7] | [31] | RW | Filter Enable for EXT_INT8[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[7] | [30:24] | RW | Filtering width of EXT_INT8[7] | 0x00 |
| FLTEN10[6] | [23] | RW | Filter Enable for EXT_INT8[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[6] | [22:16] | RW | Filtering width of EXT_INT8[6] | 0x00 |
| FLTEN10[5] | [15] | RW | Filter Enable for EXT_INT8[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[5] | [14:8] | RW | Filtering width of EXT_INT8[5] | 0x00 |
| FLTEN10[4] | [7] | RW | Filter Enable for EXT_INT8[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH10[4] | [6:0] | RW | Filtering width of EXT_INT8[4] | 0x00 |

4.3.3.147 EXT_INT9_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0850, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN11[3] | [31] | RW | Filter Enable for EXT_INT9[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[3] | [30:24] | RW | Filtering width of EXT_INT9[3] | 0x00 |
| FLTEN11[2] | [23] | RW | Filter Enable for EXT_INT9[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[2] | [22:16] | RW | Filtering width of EXT_INT9[2] | 0x00 |
| FLTEN11[1] | [15] | RW | Filter Enable for EXT_INT9[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[1] | [14:8] | RW | Filtering width of EXT_INT9[1] | 0x00 |
| FLTEN11[0] | [7] | RW | Filter Enable for EXT_INT9[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[0] | [6:0] | RW | Filtering width of EXT_INT9[0] | 0x00 |

4.3.3.148 EXT_INT9_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0854, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN11[6] | [23] | RW | Filter Enable for EXT_INT9[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[6] | [22:16] | RW | Filtering width of EXT_INT9[6] | 0x00 |
| FLTEN11[5] | [15] | RW | Filter Enable for EXT_INT9[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[5] | [14:8] | RW | Filtering width of EXT_INT9[5] | 0x00 |
| FLTEN11[4] | [7] | RW | Filter Enable for EXT_INT9[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH11[4] | [6:0] | RW | Filtering width of EXT_INT9[4] | 0x00 |

4.3.3.149 EXT_INT10_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0858, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN12[3] | [31] | RW | Filter Enable for EXT_INT10[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH12[3] | [30:24] | RW | Filtering width of EXT_INT10[3] | 0x00 |
| FLTEN12[2] | [23] | RW | Filter Enable for EXT_INT10[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH12[2] | [22:16] | RW | Filtering width of EXT_INT10[2] | 0x00 |
| FLTEN12[1] | [15] | RW | Filter Enable for EXT_INT10[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH12[1] | [14:8] | RW | Filtering width of EXT_INT10[1] | 0x00 |
| FLTEN12[0] | [7] | RW | Filter Enable for EXT_INT10[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH12[0] | [6:0] | RW | Filtering width of EXT_INT10[0] | 0x00 |

4.3.3.150 EXT_INT10_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x085C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:8] | – | Reserved | 0x0000000 |
| FLTEN12[4] | [7] | RW | Filter Enable for EXT_INT10[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH12[4] | [6:0] | RW | Filtering width of EXT_INT10[4] | 0x00 |

4.3.3.151 EXT_INT11_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0860, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN13[3] | [31] | RW | Filter Enable for EXT_INT11[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[3] | [30:24] | RW | Filtering width of EXT_INT11[3] | 0x00 |
| FLTEN13[2] | [23] | RW | Filter Enable for EXT_INT11[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[2] | [22:16] | RW | Filtering width of EXT_INT11[2] | 0x00 |
| FLTEN13[1] | [15] | RW | Filter Enable for EXT_INT11[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[1] | [14:8] | RW | Filtering width of EXT_INT11[1] | 0x00 |
| FLTEN13[0] | [7] | RW | Filter Enable for EXT_INT11[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[0] | [6:0] | RW | Filtering width of EXT_INT11[0] | 0x00 |

4.3.3.152 EXT_INT11_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0864, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN13[7] | [31] | RW | Filter Enable For EXT_INT11[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[7] | [30:24] | RW | Filtering width of EXT_INT11[7] | 0x00 |
| FLTEN13[6] | [23] | RW | Filter Enable for EXT_INT11[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[6] | [22:16] | RW | Filtering width of EXT_INT11[6] | 0x00 |
| FLTEN13[5] | [15] | RW | Filter Enable for EXT_INT11[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[5] | [14:8] | RW | Filtering width of EXT_INT11[5] | 0x00 |
| FLTEN13[4] | [7] | RW | Filter Enable for EXT_INT11[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH13[4] | [6:0] | RW | Filtering width of EXT_INT11[4] | 0x00 |

4.3.3.153 EXT_INT12_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0868, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN14[3] | [31] | RW | Filter Enable for EXT_INT12[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[3] | [30:24] | RW | Filtering width of EXT_INT12[3] | 0x00 |
| FLTEN14[2] | [23] | RW | Filter Enable for EXT_INT12[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[2] | [22:16] | RW | Filtering width of EXT_INT12[2] | 0x00 |
| FLTEN14[1] | [15] | RW | Filter Enable for EXT_INT12[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[1] | [14:8] | RW | Filtering width of EXT_INT12[1] | 0x00 |
| FLTEN14[0] | [7] | RW | Filter Enable for EXT_INT12[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[0] | [6:0] | RW | Filtering width of EXT_INT12[0] | 0x00 |

4.3.3.154 EXT_INT12_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x086C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| FLTEN14[7] | [31] | RW | Filter Enable for EXT_INT12[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[7] | [30:24] | RW | Filtering width of EXT_INT12[7] | 0x00 |
| FLTEN14[6] | [23] | RW | Filter Enable for EXT_INT12[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[6] | [22:16] | RW | Filtering width of EXT_INT12[6] | 0x00 |
| FLTEN14[5] | [15] | RW | Filter Enable for EXT_INT12[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[5] | [14:8] | RW | Filtering width of EXT_INT12[5] | 0x00 |
| FLTEN14[4] | [7] | RW | Filter Enable for EXT_INT12[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH14[4] | [6:0] | RW | Filtering width of EXT_INT12[4] | 0x00 |

4.3.3.155 EXT_INT23_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT23_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT23_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.156 EXT_INT24_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT24_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT24_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.157 EXT_INT25_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT25_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT25_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.158 EXT_INT26_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0914, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT26_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT26_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.159 EXT_INT27_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0918, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | – | Reserved | 0x00000000 |
| EXT_INT27_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT27_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.160 EXT_INT28_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x091C, Reset Value = 0x0000_0003

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:2] | – | Reserved | 0x00000000 |
| EXT_INT28_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT28_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.161 EXT_INT29_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0920, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT29_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT29_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.162 EXT_INT8_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0924, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT8_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT8_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.163 EXT_INT9_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0928, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT9_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT9_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.164 EXT_INT10_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x092C, Reset Value = 0x0000_001F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT10_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT10_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT10_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT10_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT10_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.165 EXT_INT11_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0930, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT11_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT11_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.166 EXT_INT12_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0934, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT12_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT12_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.167 EXT_INT23_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT23_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT23_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.168 EXT_INT24_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT24_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT24_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT24_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT24_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT24_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT24_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = interrupt occurs | 0x0 |
| EXT_INT24_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.169 EXT_INT25_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT25_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT25_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.170 EXT_INT26_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A14, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT26_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT26_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.171 EXT_INT27_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A18, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT27_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT27_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.172 EXT_INT28_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A1C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:2] | — | Reserved | 0x00000000 |
| EXT_INT28_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT28_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.173 EXT_INT29_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0xA20, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT29_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT29_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.174 EXT_INT8_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0A24, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT8_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = interrupt occurs | 0x0 |
| EXT_INT8_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT8_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.175 EXT_INT9_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0xA28, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT9_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT9_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.176 EXT_INT10_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0xA2C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:5] | — | Reserved | 0x00000000 |
| EXT_INT10_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT10_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT10_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT10_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT10_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.177 EXT_INT11_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0xA30, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT11_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT11_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.178 EXT_INT12_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0xA34, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT12_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |
| EXT_INT12_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt occurs | 0x0 |

4.3.3.179 EXT_INT_SERVICE_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|--|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_Group_Num | [7:3] | RW | EXT_INT Service group number 0x1 = EXT_INT23 0x2 = EXT_INT24 0x3 = EXT_INT25 0x4 = EXT_INT26 0x5 = EXT_INT27 0x6 = EXT_INT28 0x7 = EXT_INT29 0x8 = EXT_INT8 0x9 = EXT_INT9 0xA = EXT_INT10 0xB = EXT_INT11 0xC = EXT_INT12 | 0x00 |
| SVC_Num | [2:0] | RW | Interrupt number to be serviced | 0x0 |

4.3.3.180 EXT_INT_SERVICE_PEND_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_PEND | [7:0] | RW | 0x0 = Not occur 0x1 = Interrupt occurs | 0x00 |

4.3.3.181 EXT_INT_GRPFIXPRI_XA

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| RSVD | [31:4] | - | Reserved | 0x00000000 |
| Highest_GRP_NUM | [3:0] | RW | <p>When fixed group priority mode = 0 to 11, then group number should be of the highest priority.</p> <p>0x0 = EXT_INT23 0x1 = EXT_INT24 0x2 = EXT_INT25 0x3 = EXT_INT26 0x4 = EXT_INT27 0x5 = EXT_INT28 0x6 = EXT_INT29 0x7 = EXT_INT8 0x8 = EXT_INT9 0x9 = EXT_INT10 0xA = EXT_INT11 0xB = EXT_INT12</p> | 0x00 |

4.3.3.182 EXT_INT23_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B1C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT23) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.183 EXT_INT24_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT24) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.184 EXT_INT25_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B24, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT25) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.185 EXT_INT26_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B28, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT26) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.186 EXT_INT27_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B2C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT27) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.187 EXT_INT28_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B30, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 5 (EXT_INT28) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.188 EXT_INT29_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B34, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 6 (EXT_INT29) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.189 EXT_INT8_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B38, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 7 (EXT_INT8) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.190 EXT_INT9_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B3C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 8 (EXT_INT9) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.191 EXT_INT10_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B40, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 9 (EXT_INT10) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.192 EXT_INT11_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B44, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 10 (EXT_INT11) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.193 EXT_INT12_FIXPRI

- Base Address: 0x1100_0000
- Address = Base Address + 0x0B48, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 11 (EXT_INT12) when fixed priority mode: 0 to 7 | 0x0 |

4.3.3.194 GPX0CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPX0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[3] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[7] | 0x00 |
| GPX0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[2] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[6] | 0x00 |
| GPX0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = Reserved 0x4 = Reserved 0x5 = ALV_DBG[1] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[5] | 0x00 |
| GPX0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TRSTn 0x4 = GNSS_TRSTn 0x5 = ALV_DBG[0] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[4] | 0x00 |
| GPX0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TDO 0x4 = GNSS_TDO 0x5 = ALV_TDO 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[3] | 0x00 |
| GPX0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TDI | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| | | | 0x4 = GNSS_TDI 0x5 = ALV_TDI 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[2] | |
| GPX0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TMS 0x4 = GNSS_TMS 0x5 = ALV_TMS 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[1] | 0x00 |
| GPX0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = AUD_TCK 0x4 = GNSS_TCK 0x5 = ALV_TCK 0x6 to 0xE = Reserved 0xF = WAKEUP_INT0[0] | 0x00 |

4.3.3.195 GPX0DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C04, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPX0DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.196 GPX0PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C08, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX0PUD[n] | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.197 GPX0DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C0C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.198 GPX1CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C20, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPX1CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[7] 0x4 = Reserved 0x5 = ALV_DBG[11] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[7] | 0x00 |
| GPX1CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[6] 0x4 = Reserved 0x5 = ALV_DBG[10] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[6] | 0x00 |
| GPX1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[5] 0x4 = Reserved 0x5 = ALV_DBG[9] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[5] | 0x00 |
| GPX1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[4] 0x4 = Reserved 0x5 = ALV_DBG[8] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[4] | 0x00 |
| GPX1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[3] 0x4 = Reserved 0x5 = ALV_DBG[7] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[3] | 0x00 |
| GPX1CON[2] | [11:8] | RW | 0x0 = Input, 0x1 = Output, 0x2 = Reserved 0x3 = KP_COL[2] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|---|-------------|
| | | | 0x4 = Reserved 0x5 = ALV_DBG[6] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[2] | |
| GPX1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[1] 0x4 = Reserved 0x5 = ALV_DBG[5] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[1] | 0x00 |
| GPX1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_COL[0] 0x4 = Reserved 0x5 = ALV_DBG[4] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT1[0] | 0x00 |

4.3.3.199 GPX1DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C24, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPX1DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.200 GPX1PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C28, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX1PUD[n] | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.201 GPX1DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C2C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.202 GPX2CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C40, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPX2CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[7] 0x4 = Reserved 0x5 = ALV_DBG[19] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[7] | 0x00 |
| GPX2CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[6] 0x4 = Reserved 0x5 = ALV_DBG[18] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[6] | 0x00 |
| GPX2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[5] 0x4 = Reserved 0x5 = ALV_DBG[17] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[5] | 0x00 |
| GPX2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[4] 0x4 = Reserved 0x5 = ALV_DBG[16] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[4] | 0x00 |
| GPX2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[3] 0x4 = Reserved 0x5 = ALV_DBG[15] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[3] | 0x00 |
| GPX2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[2] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| | | | 0x4 = Reserved 0x5 = ALV_DBG[14] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[2] | |
| GPX2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[1] 0x4 = Reserved 0x5 = ALV_DBG[13] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[1] | 0x00 |
| GPX2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[0] 0x4 = Reserved 0x5 = ALV_DBG[12] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT2[0] | 0x00 |

4.3.3.203 GPX2DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C44, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPX2DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.204 GPX2PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C48, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX2PUD[n] | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.205 GPX2DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C4C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] N = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.206 GPX3CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C60, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPX3CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = HDMI_HPD 0x4 = Reserved 0x5 = ALV_DBG[27] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[7] | 0x00 |
| GPX3CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = HDMI_CEC 0x4 = Reserved 0x5 = ALV_DBG[26] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[6] | 0x00 |
| GPX3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[13] 0x4 = Reserved 0x5 = ALV_DBG[25] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[5] | 0x00 |
| GPX3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[12] 0x4 = Reserved 0x5 = ALV_DBG[24] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[4] | 0x00 |
| GPX3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[11] 0x4 = Reserved 0x5 = ALV_DBG[23] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[3] | 0x00 |
| GPX3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[10] | 0x00 |

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| | | | 0x4 = Reserved 0x5 = ALV_DBG[22] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[2] | |
| GPX3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[9] 0x4 = Reserved 0x5 = ALV_DBG[21] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[1] | 0x00 |
| GPX3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = Reserved 0x3 = KP_ROW[8] 0x4 = Reserved 0x5 = ALV_DBG[20] 0x6 to 0xE = Reserved 0xF = WAKEUP_INT3[0] | 0x00 |

4.3.3.207 GPX3DAT

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C64, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPX3DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.3.208 GPX3PUD

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C68, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX3PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.3.209 GPX3DRV

- Base Address: 0x1100_0000
- Address = Base Address + 0x0C6C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPX3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.3.210 EXT_INT40CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT40_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT40[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT40_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT40[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT40_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT40[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT40_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT40[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT40_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT40[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT40_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT40[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT40_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT40[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT40_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT40[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.211 EXT_INT41CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT41_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT41[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT41_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT41[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT41_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT41[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT41_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT41[4] 0x0 = Low level 0x1 = High level 0x2 = Falling edge triggered 0x3 = Rising edge triggered 0x4 = Both edge triggered 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT41_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT41[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT41_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT41[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT41_CON[1] | [6:4] | W | Sets signaling method of EXT_INT41[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT41_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT41[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.212 EXT_INT42CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT42_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT42[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT42_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT42[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT42_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT42[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT42_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT42[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT42_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT42[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT42_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT42[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT42_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT42[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT42_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT42[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.213 EXT_INT43CON

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT43_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT43[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT43_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT43[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT43_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT43[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT43_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT43[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT43_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT43[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT43_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT43[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT43_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT43[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT43_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT43[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.3.214 EXT_INT40_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E80, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN15[3] | [31] | RW | Filter Enable for EXT_INT40[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[3] | [30] | RW | Filter Selection for EXT_INT40[3] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[3] | [29:24] | RW | Filtering width of EXT_INT40[3] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[2] | [23] | RW | Filter Enable for EXT_INT40[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[2] | [22] | RW | Filter Selection for EXT_INT40[2] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[2] | [21:16] | RW | Filtering width of EXT_INT40[2] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[1] | [15] | RW | Filter Enable for EXT_INT40[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[1] | [14] | RW | Filter Selection for EXT_INT40[1] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[1] | [13:8] | RW | Filtering width of EXT_INT40[1] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[0] | [7] | RW | Filter Enable for EXT_INT40[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[0] | [6] | RW | Filter Selection for EXT_INT40[0] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[0] | [5:0] | RW | Filtering width of EXT_INT40[0] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |

4.3.3.215 EXT_INT40_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E84, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN15[7] | [31] | RW | Filter Enable for EXT_INT40[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[7] | [30] | RW | Filter Selection for EXT_INT40[7] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[7] | [29:24] | RW | Filtering width of EXT_INT40[7] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[6] | [23] | RW | Filter Enable for EXT_INT40[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[6] | [22] | RW | Filter Selection for EXT_INT40[6] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[6] | [21:16] | RW | Filtering width of EXT_INT40[6] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[5] | [15] | RW | Filter Enable for EXT_INT40[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[5] | [14] | RW | Filter Selection for EXT_INT40[5] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[5] | [13:8] | RW | Filtering width of EXT_INT40[5] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |
| FLTEN15[4] | [7] | RW | Filter Enable for EXT_INT40[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL15[4] | [6] | RW | Filter Selection for EXT_INT40[4] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH15[4] | [5:0] | RW | Filtering width of EXT_INT40[4] This value is valid when FLTSEL15 (of EXT_INT40) is 0x1. | 0x00 |

4.3.3.216 EXT_INT41_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E88, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN16[3] | [31] | RW | Filter Enable for EXT_INT41[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[3] | [30] | RW | Filter Selection for EXT_INT41[3] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[3] | [29:24] | RW | Filtering width of EXT_INT41[3] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[2] | [23] | RW | Filter Enable for EXT_INT41[2] 0x0 = Disables Filter 0x1 = Enables | 0x1 |
| FLTSEL16[2] | [22] | RW | Filter Selection for EXT_INT41[2] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[2] | [21:16] | RW | Filtering width of EXT_INT41[2] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[1] | [15] | RW | Filter Enable for EXT_INT41[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[1] | [14] | RW | Filter Selection for EXT_INT41[1] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[1] | [13:8] | RW | Filtering width of EXT_INT41[1] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[0] | [7] | RW | Filter Enable for EXT_INT41[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[0] | [6] | RW | Filter Selection for EXT_INT41[0] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[0] | [5:0] | RW | Filtering width of EXT_INT41[0] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |

4.3.3.217 EXT_INT41_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E8C, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN16[7] | [31] | RW | Filter Enable for EXT_INT41[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[7] | [30] | RW | Filter Selection for EXT_INT41[7] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[7] | [29:24] | RW | Filtering width of EXT_INT41[7] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[6] | [23] | RW | Filter Enable for EXT_INT41[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[6] | [22] | RW | Filter Selection for EXT_INT41[6] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[6] | [21:16] | RW | Filtering width of EXT_INT41[6] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[5] | [15] | RW | Filter Enable for EXT_INT41[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[5] | [14] | RW | Filter Selection for EXT_INT41[5] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[5] | [13:8] | RW | Filtering width of EXT_INT41[5] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |
| FLTEN16[4] | [7] | RW | Filter Enable for EXT_INT41[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL16[4] | [6] | RW | Filter Selection for EXT_INT41[4] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH16[4] | [5:0] | RW | Filtering width of EXT_INT41[4] This value is valid when FLTSEL16 (of EXT_INT41) is 0x1. | 0x00 |

4.3.3.218 EXT_INT42_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E90, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN17[3] | [31] | RW | Filter Enable for EXT_INT42[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[3] | [30] | RW | Filter Selection for EXT_INT42[3] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[3] | [29:24] | RW | Filtering width of EXT_INT42[3] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[2] | [23] | RW | Filter Enable for EXT_INT42[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[2] | [22] | RW | Filter Selection for EXT_INT42[2] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[2] | [21:16] | RW | Filtering width of EXT_INT42[2] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[1] | [15] | RW | Filter Enable for EXT_INT42[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[1] | [14] | RW | Filter Selection for EXT_INT42[1] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[1] | [13:8] | RW | Filtering width of EXT_INT42[1] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[0] | [7] | RW | Filter Enable for EXT_INT42[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[0] | [6] | RW | Filter Selection for EXT_INT42[0] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[0] | [5:0] | RW | Filtering width of EXT_INT42[0] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |

4.3.3.219 EXT_INT42_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E94, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN17[7] | [31] | RW | Filter Enable for EXT_INT42[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[7] | [30] | RW | Filter Selection for EXT_INT42[7] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[7] | [29:24] | RW | Filtering width of EXT_INT42[7] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[6] | [23] | RW | Filter Enable for EXT_INT42[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[6] | [22] | RW | Filter Selection for EXT_INT42[6] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[6] | [21:16] | RW | Filtering width of EXT_INT42[6] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[5] | [15] | RW | Filter Enable for EXT_INT42[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[5] | [14] | RW | Filter Selection for EXT_INT42[5] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[5] | [13:8] | RW | Filtering width of EXT_INT42[5] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |
| FLTEN17[4] | [7] | RW | Filter Enable for EXT_INT42[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL17[4] | [6] | RW | Filter Selection for EXT_INT42[4] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH17[4] | [5:0] | RW | Filtering width of EXT_INT42[4] This value is valid when FLTSEL17 (of EXT_INT42) is 0x1. | 0x00 |

4.3.3.220 EXT_INT43_FLTCON0

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E98, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN18[3] | [31] | RW | Filter Enable for EXT_INT43[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[3] | [30] | RW | Filter Selection for EXT_INT43[3] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[3] | [29:24] | RW | Filtering width of EXT_INT43[3] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[2] | [23] | RW | Filter Enable for EXT_INT43[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[2] | [22] | RW | Filter Selection for EXT_INT43[2] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[2] | [21:16] | RW | Filtering width of EXT_INT43[2] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[1] | [15] | RW | Filter Enable for EXT_INT43[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[1] | [14] | RW | Filter Selection for EXT_INT43[1] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[1] | [13:8] | RW | Filtering width of EXT_INT43[1] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[0] | [7] | RW | Filter Enable for EXT_INT43[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[0] | [6] | RW | Filter Selection for EXT_INT43[0] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[0] | [5:0] | RW | Filtering width of EXT_INT43[0] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |

4.3.3.221 EXT_INT43_FLTCON1

- Base Address: 0x1100_0000
- Address = Base Address + 0x0E9C, Reset Value = 0x8080_8080

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| FLTEN18[7] | [31] | RW | Filter Enable for EXT_INT43[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[7] | [30] | RW | Filter Selection for EXT_INT43[7] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[7] | [29:24] | RW | Filtering width of EXT_INT43[7] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[6] | [23] | RW | Filter Enable for EXT_INT43[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[6] | [22] | RW | Filter Selection for EXT_INT43[6] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[6] | [21:16] | RW | Filtering width of EXT_INT43[6] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[5] | [15] | RW | Filter Enable for EXT_INT43[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[5] | [14] | RW | Filter Selection for EXT_INT43[5] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[5] | [13:8] | RW | Filtering width of EXT_INT43[5] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |
| FLTEN18[4] | [7] | RW | Filter Enable for EXT_INT43[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x1 |
| FLTSEL18[4] | [6] | RW | Filter Selection for EXT_INT43[4] 0x0 = Delays filter 0x1 = Digital filter (clock count) | 0x0 |
| FLTWIDTH18[4] | [5:0] | RW | Filtering width of EXT_INT43[4] This value is valid when FLTSEL18 (of EXT_INT43) is 0x1. | 0x00 |

4.3.3.222 EXT_INT40_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F00, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT40_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT40_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.223 EXT_INT41_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F04, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT41_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT41_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.224 EXT_INT42_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F08, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT42_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT42_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.225 EXT_INT43_MASK

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F0C, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT43_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT43_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.3.226 EXT_INT40_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F40, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT40_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT40_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.3.227 EXT_INT41_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F44, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT41_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT41_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.3.228 EXT_INT42_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F48, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT42_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT42_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.3.229 EXT_INT43_PEND

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F4C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT43_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT43_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.3.230 PDNEN

- Base Address: 0x1100_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|--|-------------|
| RSVD | [7:2] | — | Reserved | 0x00 |
| PDNEN_CFG | [1] | RW | 0 = Automatically by power down mode 1 = By PDNEN bit | 0x0 |
| PDNEN | [0] | RW | Power down mode pad state enable register. 0 = PADs Controlled by normal mode This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" 1 = PADs Controlled by Power Down mode control registers | 0x0 |

4.3.4 Part 3

4.3.4.1 GPZCON

- Base Address: 0x0386_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| GPZCON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[2] 0x3 = ST_INT 0x4 to 0xE = Reserved 0xF = EXT_INT50[6] | 0x00 |
| GPZCON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[1] 0x3 = ST_TICK 0x4 to 0xE = Reserved 0xF = EXT_INT50[5] | 0x00 |
| GPZCON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_SDO[0] 0x3 = PCM_0_SOUT 0x4 to 0xE = Reserved 0xF = EXT_INT50[4] | 0x00 |
| GPZCON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_SD1 0x3 = PCM_0_SIN 0x4 to 0xE = Reserved 0xF = EXT_INT50[3] | 0x00 |
| GPZCON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_LRCK 0x3 = PCM_0_FSYNC 0x4 to 0xE = Reserved 0xF = EXT_INT50[2] | 0x00 |
| GPZCON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_CDCLK 0x3 = PCM_0_EXTCLK 0x4 to 0xE = Reserved 0xF = EXT_INT50[1] | 0x00 |
| GPZCON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = I2S_0_SCLK 0x3 = PCM_0_SCLK 0x4 to 0xE = Reserved 0xF = EXT_INT50[0] | 0x00 |

4.3.4.2 GPZDAT

- Base Address: 0x0386_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-------------|-------|------|--|-------------|
| GPZDAT[6:0] | [6:0] | RWX | When you configure port as input port then corresponding bit is pin state. While configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.4.3 GPZPUD

- Base Address: 0x0386_0000
- Address = Base Address + 0x0008, Reset Value = 0x1555

| Name | Bit | Type | Description | Reset Value |
|-----------|---------------------------|------|--|-------------|
| GPZPUD[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x1555 |

4.3.4.4 GPZDRV

- Base Address: 0x0386_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------------------------|------|--|-------------|
| GPZDRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.4.5 GPZCONPDN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|--------|---------------------------|------|---|-------------|
| GPZ[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.4.6 GPZPUDPDN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|--------|---------------------------|------|--|-------------|
| GPZ[n] | [2n + 1:2n] n = 0 to 6 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.4.7 EXT_INT50CON

- Base Address: 0x0386_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | — | Reserved | 0x0 |
| RSVD | [27] | — | Reserved | 0x0 |
| EXT_INT50_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT50[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | — | Reserved | 0x0 |
| EXT_INT50_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT50[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | — | Reserved | 0x0 |
| EXT_INT50_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT50[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | — | Reserved | 0x0 |
| EXT_INT50_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT50[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | — | Reserved | 0x0 |
| EXT_INT50_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT50[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-------|------|--|-------------|
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT50_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT50[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT50_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT50[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.4.8 EXT_INT50_FLTCON0

- Base Address: 0x0386_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN1[3] | [31] | RW | Filter Enable for EXT_INT50[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[3] | [30:24] | RW | Filtering width of EXT_INT50[3] | 0x00 |
| FLTEN1[2] | [23] | RW | Filter Enable for EXT_INT50[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[2] | [22:16] | RW | Filtering width of EXT_INT50[2] | 0x00 |
| FLTEN1[1] | [15] | RW | Filter Enable for EXT_INT50[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[1] | [14:8] | RW | Filtering width of EXT_INT50[1] | 0x00 |
| FLTEN1[0] | [7] | RW | Filter Enable for EXT_INT50[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[0] | [6:0] | RW | Filtering width of EXT_INT50[0] | 0x00 |

4.3.4.9 EXT_INT50_FLTCON1

- Base Address: 0x0386_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| FLTEN1[6] | [23] | RW | Filter Enable for EXT_INT50[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[6] | [22:16] | RW | Filtering width of EXT_INT50[6] | 0x00 |
| FLTEN1[5] | [15] | RW | Filter Enable for EXT_INT50[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[5] | [14:8] | RW | Filtering width of EXT_INT50[5] | 0x00 |
| FLTEN1[4] | [7] | RW | Filter Enable for EXT_INT50[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[4] | [6:0] | RW | Filtering width of EXT_INT50[4] | 0x00 |

4.3.4.10 EXT_INT50_MASK

- Base Address: 0x0386_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_007F

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT50_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT50_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.4.11 EXT_INT50_PEND

- Base Address: 0x0386_0000
- Address = Base Address + 0xA00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0x00000000 |
| EXT_INT50_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT50_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT50_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT50_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = interrupt Occurs | 0x0 |
| EXT_INT50_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = interrupt Occurs | 0x0 |
| EXT_INT50_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = interrupt Occurs | 0x0 |
| EXT_INT50_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.4.12 EXT_INT_SERVICE_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0xB08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_Group_Num | [7:3] | RW | EXT_INT Service group number 0x1 = EXT_INT50 | 0x00 |
| SVC_Num | [2:0] | RW | Interrupt number to be serviced | 0x0 |

4.3.4.13 EXT_INT_SERVICE_PEND_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0xB0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_PEND | [7:0] | RW | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x00 |

4.3.4.14 EXT_INT_GRPFIXPRI_XD

- Base Address: 0x0386_0000
- Address = Base Address + 0xB10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| RSVD | [31:4] | – | Reserved | 0x00000000 |
| Highest_GRP_NUM | [3:0] | RW | When fixed group priority mode = 0, then group number should be of the highest priority. 0x0 = EXT_INT50 | 0x00 |

4.3.4.15 EXT_INT50_FIXPRI

- Base Address: 0x0386_0000
- Address = Base Address + 0x0B14, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | – | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT50) when fixed priority mode: 0 to 7 | 0x0 |

4.3.4.16 PDNEN

- Base Address: 0x0386_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|--|-------------|
| RSVD | [7:2] | – | Reserved | 0x00 |
| PDNEN_CFG | [1] | RW | 0 = Automatically by power down mode 1 = By PDNEN bit | 0x0 |
| PDNEN | [0] | RW | Power down mode pad state enable register. 0 = PADs Controlled by normal mode This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" 1 = PADs Controlled by Power Down mode control registers | 0x0 |

4.3.5 Part 4

4.3.5.1 GPV0CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPV0CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT30[7] | 0x00 |
| GPV0CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT30[6] | 0x00 |
| GPV0CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT30[5] | 0x00 |
| GPV0CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT30[4] | 0x00 |
| GPV0CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT30[3] | 0x00 |
| GPV0CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT30[2] | 0x00 |
| GPV0CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT30[1] | 0x00 |
| GPV0CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT30[0] | 0x00 |

4.3.5.2 GPV0DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPV0DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.5.3 GPV0PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0008, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV0PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.5.4 GPV0DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x000C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV0DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.5.5 GPV0CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPV0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.5.6 GPV0PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPV0[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.5.7 GPV1CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPV1CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT31[7] | 0x00 |
| GPV1CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT31[6] | 0x00 |
| GPV1CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[13] 0x3 to 0xE = Reserved 0xF = EXT_INT31[5] | 0x00 |
| GPV1CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[12] 0x3 to 0xE = Reserved 0xF = EXT_INT31[4] | 0x00 |
| GPV1CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT31[3] | 0x00 |
| GPV1CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT31[2] | 0x00 |
| GPV1CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT31[1] | 0x00 |
| GPV1CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_RXD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT31[0] | 0x00 |

4.3.5.8 GPV1DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0024, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPV1DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.5.9 GPV1PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0028, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV1PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.5.10 GPV1DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x002C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV1DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.5.11 GPV1CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPV1[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.5.12 GPV1PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPV1[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.5.13 ETC7PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0048, Reset Value = 0x0005

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC7PUD[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0005 |

ETC7PUD[1:0] controls Xc2cRXCLK[0].

ETC7PUD[3:2] controls Xc2cRXCLK[1].

4.3.5.14 ETC7DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x004C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC7DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

ETC7DRV[1:0] controls Xc2cRXCLK[0].

ETC7DRV[3:2] controls Xc2cRXCLK[1].

4.3.5.15 GPV2CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| GPV2CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[7] 0x3 to 0xE = Reserved 0xF = EXT_INT32[7] | 0x00 |
| GPV2CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[6] 0x3 to 0xE = Reserved 0xF = EXT_INT32[6] | 0x00 |
| GPV2CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[5] 0x3 to 0xE = Reserved 0xF = EXT_INT32[5] | 0x00 |
| GPV2CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[4] 0x3 to 0xE = Reserved 0xF = EXT_INT32[4] | 0x00 |
| GPV2CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[3] 0x3 to 0xE = Reserved 0xF = EXT_INT32[3] | 0x00 |
| GPV2CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[2] 0x3 to 0xE = Reserved 0xF = EXT_INT32[2] | 0x00 |
| GPV2CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[1] 0x3 to 0xE = Reserved 0xF = EXT_INT32[1] | 0x00 |
| GPV2CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[0] 0x3 to 0xE = Reserved 0xF = EXT_INT32[0] | 0x00 |

4.3.5.16 GPV2DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0064, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPV2DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.5.17 GPV2PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0068, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV2PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x5555 |

4.3.5.18 GPV2DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x006C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV2DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.5.19 GPV2CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPV2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.5.20 GPV2PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPV2[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.5.21 GPV3CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| GPV3CON[7] | [31:28] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[15] 0x3 to 0xE = Reserved 0xF = EXT_INT33[7] | 0x00 |
| GPV3CON[6] | [27:24] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[14] 0x3 to 0xE = Reserved 0xF = EXT_INT33[6] | 0x00 |
| GPV3CON[5] | [23:20] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[13] 0x3 to 0xE = Reserved 0xF = EXT_INT33[5] | 0x00 |
| GPV3CON[4] | [19:16] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[12] 0x3 to 0xE = Reserved 0xF = EXT_INT33[4] | 0x00 |
| GPV3CON[3] | [15:12] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[11] 0x3 to 0xE = Reserved 0xF = EXT_INT33[3] | 0x00 |
| GPV3CON[2] | [11:8] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[10] 0x3 to 0xE = Reserved 0xF = EXT_INT33[2] | 0x00 |
| GPV3CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[9] 0x3 to 0xE = Reserved 0xF = EXT_INT33[1] | 0x00 |
| GPV3CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_TXD[8] 0x3 to 0xE = Reserved 0xF = EXT_INT33[0] | 0x00 |

4.3.5.22 GPV3DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x0084, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPV3DAT[7:0] | [7:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.5.23 GPV3PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x0088, Reset Value = 0x5555

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|---|-------------|
| GPV3PUD[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved, 0x3 = Enables Pull-up | 0x5555 |

4.3.5.24 GPV3DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x008C, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV3DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.5.25 GPV3CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPV3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.5.26 GPV3PUDPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0094, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPV3[n] | [2n + 1:2n] n = 0 to 7 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.5.27 ETC8PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x00A8, Reset Value = 0x0005

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC8PUD[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0005 |

ETC8PUD[1:0] controls Xc2cTXCLK[0].

ETC8PUD[3:2] controls Xc2cTXCLK[1].

4.3.5.28 ETC8DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x00AC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| ETC8DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

ETC8DRV[1:0] controls Xc2cTXCLK[0].

ETC8DRV[3:2] controls Xc2cTXCLK[1].

4.3.5.29 GPV4CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|-------|------|--|-------------|
| GPV4CON[1] | [7:4] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_WKREQOUT 0x3 to 0xE = Reserved 0xF = EXT_INT34[1] | 0x00 |
| GPV4CON[0] | [3:0] | RW | 0x0 = Input 0x1 = Output 0x2 = C2C_WKREQIN 0x3 to 0xE = Reserved 0xF = EXT_INT34[0] | 0x00 |

4.3.5.30 GPV4DAT

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C4, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|--------------|-------|------|---|-------------|
| GPV4DAT[1:0] | [1:0] | RWX | When you configure port as input port then corresponding bit is pin state. When configuring as output port then pin state should be same as corresponding bit. When the port is configured as functional pin, the undefined value will be read. | 0x00 |

4.3.5.31 GPV4PUD

- Base Address: 0x106E_0000
- Address = Base Address + 0x00C8, Reset Value = 0x0005

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV4PUD[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x0005 |

4.3.5.32 GPV4DRV

- Base Address: 0x106E_0000
- Address = Base Address + 0x00CC, Reset Value = 0x00_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------------------------|------|--|-------------|
| GPV4DRV[n] | [23:16] | RW | Reserved (Should be zero) | 0x00 |
| | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = 1x 0x2 = 2x 0x1 = 3x 0x3 = 4x | 0x0000 |

4.3.5.33 GPV4CONPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|---|-------------|
| GPV4[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Outputs 0 0x1 = Outputs 1 0x2 = Input 0x3 = Previous state | 0x00 |

4.3.5.34 GPV4PUPDN

- Base Address: 0x106E_0000
- Address = Base Address + 0x00D4, Reset Value = 0x0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------------------------|------|--|-------------|
| GPV4[n] | [2n + 1:2n] n = 0 to 1 | RW | 0x0 = Disables Pull-up/Pull-down 0x1 = Enables Pull-down 0x2 = Reserved 0x3 = Enables Pull-up | 0x00 |

4.3.5.35 EXT_INT30CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0700, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT30_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT30[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT30_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT30[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT30_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT30[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT30_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT30[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT30_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT30[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT30_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT30[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT30_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT30[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT30_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT30[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.5.36 EXT_INT31CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0704, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT31_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT31[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT31_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT31[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT31_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT31[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT31_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT31[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT31_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT31[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT31_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT31[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT31_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT31[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT31_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT31[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.5.37 EXT_INT32CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0708, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT32_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT32[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT32_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT32[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT32_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT32[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT32_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT32[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT32_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT32[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT32_CON[2] | [10:8] | RW | Sets the signaling method of EXT_INT32[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT32_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT32[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT32_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT32[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.5.38 EXT_INT33CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x070C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| EXT_INT33_CON[7] | [30:28] | RW | Sets signaling method of EXT_INT33[7] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| EXT_INT33_CON[6] | [26:24] | RW | Sets signaling method of EXT_INT33[6] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| EXT_INT33_CON[5] | [22:20] | RW | Sets signaling method of EXT_INT33[5] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| EXT_INT33_CON[4] | [18:16] | RW | Sets signaling method of EXT_INT33[4] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| EXT_INT33_CON[3] | [14:12] | RW | Sets signaling method of EXT_INT33[3] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| EXT_INT33_CON[2] | [10:8] | RW | Sets signaling method of EXT_INT33[2] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| EXT_INT33_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT33[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| EXT_INT33_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT33[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.5.39 EXT_INT34CON

- Base Address: 0x106E_0000
- Address = Base Address + 0x0710, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|--|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| RSVD | [7] | — | Reserved | 0x0 |
| EXT_INT34_CON[1] | [6:4] | RW | Sets signaling method of EXT_INT34[1] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |
| RSVD | [3] | — | Reserved | 0x0 |
| EXT_INT34_CON[0] | [2:0] | RW | Sets signaling method of EXT_INT34[0] 0x0 = Low level 0x1 = High level 0x2 = Triggers Falling edge 0x3 = Triggers Rising edge 0x4 = Triggers Both edge 0x5 to 0x7 = Reserved | 0x0 |

4.3.5.40 EXT_INT30_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0800, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN1[3] | [31] | RW | Filter Enable for EXT_INT30[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[3] | [30:24] | RW | Filtering width of EXT_INT30[3] | 0x00 |
| FLTEN1[2] | [23] | RW | Filter Enable for EXT_INT30[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[2] | [22:16] | RW | Filtering width of EXT_INT30[2] | 0x00 |
| FLTEN1[1] | [15] | RW | Filter Enable for EXT_INT30[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[1] | [14:8] | RW | Filtering width of EXT_INT30[1] | 0x00 |
| FLTEN1[0] | [7] | RW | Filter Enable for EXT_INT30[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[0] | [6:0] | RW | Filtering width of EXT_INT30[0] | 0x00 |

4.3.5.41 EXT_INT30_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0804, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN1[7] | [31] | RW | Filter Enable for EXT_INT30[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[7] | [30:24] | RW | Filtering width of EXT_INT30[7] | 0x00 |
| FLTEN1[6] | [23] | RW | Filter Enable for EXT_INT30[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[6] | [22:16] | RW | Filtering width of EXT_INT30[6] | 0x00 |
| FLTEN1[5] | [15] | RW | Filter Enable for EXT_INT30[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[5] | [14:8] | RW | Filtering width of EXT_INT30[5] | 0x00 |
| FLTEN1[4] | [7] | RW | Filter Enable for EXT_INT30[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH1[4] | [6:0] | RW | Filtering width of EXT_INT30[4] | 0x00 |

4.3.5.42 EXT_INT31_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0808, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN2[3] | [31] | RW | Filter Enable for EXT_INT31[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[3] | [30:24] | RW | Filtering width of EXT_INT31[3] | 0x00 |
| FLTEN2[2] | [23] | RW | Filter Enable for EXT_INT31[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[2] | [22:16] | RW | Filtering width of EXT_INT31[2] | 0x00 |
| FLTEN2[1] | [15] | RW | Filter Enable for EXT_INT31[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[1] | [14:8] | RW | Filtering width of EXT_INT31[1] | 0x00 |
| FLTEN2[0] | [7] | RW | Filter Enable for EXT_INT31[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[0] | [6:0] | RW | Filtering width of EXT_INT31[0] | 0x00 |

4.3.5.43 EXT_INT31_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x080C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN2[7] | [31] | RW | Filter Enable for EXT_INT31[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[7] | [30:24] | RW | Filtering width of EXT_INT31[7] | 0x00 |
| FLTEN2[6] | [23] | RW | Filter Enable for EXT_INT31[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[6] | [22:16] | RW | Filtering width of EXT_INT31[6] | 0x00 |
| FLTEN2[5] | [15] | RW | Filter Enable for EXT_INT31[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[5] | [14:8] | RW | Filtering width of EXT_INT31[5] | 0x00 |
| FLTEN2[4] | [7] | RW | Filter Enable for EXT_INT31[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH2[4] | [6:0] | RW | Filtering width of EXT_INT31[4] | 0x00 |

4.3.5.44 EXT_INT32_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0810, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN3[3] | [31] | RW | Filter Enable for EXT_INT32[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[3] | [30:24] | RW | Filtering width of EXT_INT32[3] | 0x00 |
| FLTEN3[2] | [23] | RW | Filter Enable for EXT_INT32[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[2] | [22:16] | RW | Filtering width of EXT_INT32[2] | 0x00 |
| FLTEN3[1] | [15] | RW | Filter Enable for EXT_INT32[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[1] | [14:8] | RW | Filtering width of EXT_INT32[1] | 0x00 |
| FLTEN3[0] | [7] | RW | Filter Enable for EXT_INT32[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[0] | [6:0] | RW | Filtering width of EXT_INT32[0] | 0x00 |

4.3.5.45 EXT_INT32_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0814, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN3[7] | [31] | RW | Filter Enable for EXT_INT32[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[7] | [30:24] | RW | Filtering width of EXT_INT32[7] | 0x00 |
| FLTEN3[6] | [23] | RW | Filter Enable for EXT_INT32[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[6] | [22:16] | RW | Filtering width of EXT_INT32[6] | 0x00 |
| FLTEN3[5] | [15] | RW | Filter Enable for EXT_INT32[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[5] | [14:8] | RW | Filtering width of EXT_INT32[5] | 0x00 |
| FLTEN3[4] | [7] | RW | Filter Enable for EXT_INT32[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH3[4] | [6:0] | RW | Filtering width of EXT_INT32[4] | 0x00 |

4.3.5.46 EXT_INT33_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0818, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN4[3] | [31] | RW | Filter Enable for EXT_INT33[3] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[3] | [30:24] | RW | Filtering width of EXT_INT33[3] | 0x00 |
| FLTEN4[2] | [23] | RW | Filter Enable for EXT_INT33[2] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[2] | [22:16] | RW | Filtering width of EXT_INT33[2] | 0x00 |
| FLTEN4[1] | [15] | RW | Filter Enable for EXT_INT33[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[1] | [14:8] | RW | Filtering width of EXT_INT33[1] | 0x00 |
| FLTEN4[0] | [7] | RW | Filter Enable for EXT_INT33[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[0] | [6:0] | RW | Filtering width of EXT_INT33[0] | 0x00 |

4.3.5.47 EXT_INT33_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x081C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| FLTEN4[7] | [31] | RW | Filter Enable for EXT_INT33[7] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[7] | [30:24] | RW | Filtering width of EXT_INT33[7] | 0x00 |
| FLTEN4[6] | [23] | RW | Filter Enable for EXT_INT33[6] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[6] | [22:16] | RW | Filtering width of EXT_INT33[6] | 0x00 |
| FLTEN4[5] | [15] | RW | Filter Enable for EXT_INT33[5] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[5] | [14:8] | RW | Filtering width of EXT_INT33[5] | 0x00 |
| FLTEN4[4] | [7] | RW | Filter Enable for EXT_INT33[4] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH4[4] | [6:0] | RW | Filtering width of EXT_INT33[4] | 0x00 |

4.3.5.48 EXT_INT34_FLTCON0

- Base Address: 0x106E_0000
- Address = Base Address + 0x0820, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0 |
| FLTEN5[1] | [15] | RW | Filter Enable for EXT_INT34[1] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[1] | [14:8] | RW | Filtering width of EXT_INT34[1] | 0x00 |
| FLTEN5[0] | [7] | RW | Filter Enable for EXT_INT34[0] 0x0 = Disables Filter 0x1 = Enables Filter | 0x0 |
| FLTWIDTH5[0] | [6:0] | RW | Filtering width of EXT_INT34[0] | 0x00 |

4.3.5.49 EXT_INT34_FLTCON1

- Base Address: 0x106E_0000
- Address = Base Address + 0x0824, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|-------------|-------------|
| RSVD | [31:0] | – | Reserved | 0x00000000 |

4.3.5.50 EXT_INT30_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0900, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT30_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT30_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.5.51 EXT_INT31_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0904, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT31_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT31_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.5.52 EXT_INT32_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0908, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT32_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT32_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.5.53 EXT_INT33_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x090C, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT33_MASK[7] | [7] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[6] | [6] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[5] | [5] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[4] | [4] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[3] | [3] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[2] | [2] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT33_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.5.54 EXT_INT34_MASK

- Base Address: 0x106E_0000
- Address = Base Address + 0x0910, Reset Value = 0x0000_0003

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:2] | — | Reserved | 0x00000000 |
| EXT_INT34_MASK[1] | [1] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |
| EXT_INT34_MASK[0] | [0] | RW | 0x0 = Enables Interrupt 0x1 = Masked | 0x1 |

4.3.5.55 EXT_INT30_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x000000 |
| EXT_INT30_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT30_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.5.56 EXT_INT31_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT31_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT31_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.5.57 EXT_INT32_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x000000 |
| EXT_INT32_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT32_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.5.58 EXT_INT33_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0x0000000 |
| EXT_INT33_PEND[7] | [7] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[6] | [6] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[5] | [5] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[4] | [4] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[3] | [3] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[2] | [2] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT33_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.5.59 EXT_INT34_PEND

- Base Address: 0x106E_0000
- Address = Base Address + 0x0A10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:2] | — | Reserved | 0x00000000 |
| EXT_INT34_PEND[1] | [1] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |
| EXT_INT34_PEND[0] | [0] | RWX | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x0 |

4.3.5.60 EXT_INT_SERVICE_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0xB08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_Group_Num | [7:3] | RW | EXT_INT Service group number 0x1 = EXT_INT30 0x2 = EXT_INT31 0x3 = EXT_INT32 0x4 = EXT_INT33 0x5 = EXT_INT34 | 0x00 |
| SVC_Num | [2:0] | RW | Interrupt number to be serviced | 0x0 |

4.3.5.61 EXT_INT_SERVICE_PEND_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0xB0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:8] | RW | Reserved | 0x00000000 |
| SVC_PEND | [7:0] | RW | 0x0 = Not occur 0x1 = Interrupt Occurs | 0x00 |

4.3.5.62 EXT_INT_GRPFIXPRI_XC

- Base Address: 0x106E_0000
- Address = Base Address + 0xB10, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|--|-------------|
| RSVD | [31:4] | - | Reserved | 0x00000000 |
| Highest_GRP_NUM | [3:0] | RW | Group number of the highest priority when fixed group priority mode: 0 to 4 0x0 = EXT_INT30 0x1 = EXT_INT31 0x2 = EXT_INT32 0x3 = EXT_INT33 0x4 = EXT_INT34 | 0x00 |

4.3.5.63 EXT_INT30_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0xB14, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 0 (EXT_INT30) when fixed priority mode: 0 to 7 | 0x0 |

4.3.5.64 EXT_INT31_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0xB18, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 1 (EXT_INT31) when fixed priority mode: 0 to 7 | 0x0 |

4.3.5.65 EXT_INT32_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0xB1C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 2 (EXT_INT32) when fixed priority mode: 0 to 7 | 0x0 |

4.3.5.66 EXT_INT33_FIXPRI

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B20, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 3 (EXT_INT33) when fixed priority mode: 0 to 7 | 0x0 |

4.3.5.67 EXT_INT34_FIXPR

- Base Address: 0x106E_0000
- Address = Base Address + 0x0B24, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x00000000 |
| Highest_EINT_NUM | [2:0] | RW | Interrupt number of the highest priority in External Interrupt Group 4 (EXT_INT34) when fixed priority mode: 0 to 7 | 0x0 |

4.3.5.68 PDNEN

- Base Address: 0x106E_0000
- Address = Base Address + 0x0F80, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|--|-------------|
| RSVD | [7:2] | – | Reserved | 0x00 |
| PDNEN_CFG | [1] | RW | 0 = Automatically by power down mode 1 = by PDNEN bit | 0x0 |
| PDNEN | [0] | RW | Power down mode pad state enable register. 0 = PADs Controlled by normal mode 1 = PADs Controlled by Power Down mode control registers This bit is set to "1" automatically when system enters into Power down mode and clears by writing "0" to this bit or cold reset. After wake up from Power down mode, this bit maintains value "1" until writing "0" | 0x0 |

5 Clock Management Unit

5.1 Overview

This chapter describes the Clock Management Units (CMUs) of Exynos 4412. CMUs control Phase Locked Loops (PLLs) and generate system clocks for CPU, buses, and function clocks for individual IPs in Exynos 4412. They also communicate with the power management unit (PMU) in order to stop clocks before entering certain low power mode to reduce power consumption by minimizing clock toggling.

Table 5-1 describes the typical operating frequencies for each function block in Exynos 4412.

Table 5-1 Operating Frequencies in Exynos 4412

| Function Block | Description | Typical Operating Frequency |
|----------------|---|--------------------------------|
| CPU | Cortex-A9 MPCore It is a Quad Core processor. | 200 MHz to 1.4 GHz |
| | CoreSight | 200 MHz/100 MHz |
| DMC | DMC, 2D Graphics Engine | 400 MHz (up to 200MHz for G2D) |
| SSS | Security Sub-System | 200 MHz |
| LEFTBUS | Data Bus/Peripheral Bus | 200 MHz/100 MHz |
| RIGHTBUS | Data Bus/Peripheral Bus | 200 MHz/100 MHz |
| G3D | 3D Graphics Engine | 440 MHz |
| MFC | Multi-format Codec | 200 MHz |
| IMAGE | Rotator, MDMA | 200 MHz |
| LCD0 | FIMD0, MIE0, MIPI DS10 | 160 MHz |
| ISP | ISP | 160 MHz |
| CAM | FIMC0, FIMC1, FIMC2, FIMC3 JPEG | 160 MHz |
| TV | VP, MIXER, TVENC | 160 MHz |
| FSYS | USB, PCIe, SDMMC, TSI, OneNANDC, SROMC, PDMA0, PDMA1, NFCON, MIPI-HIS, ADC | 133 MHz |
| GPS | GPS | 133 MHz |
| MAUDIO | AudioSS, iROM, iRAM | 192 MHz |
| PERI-L | UART, I2C, SPI, I2S, PCM, SPDIF, PWM, I2CHDMI, Slimbus | 100 MHz |
| PERI-R | CHIPID, SYSREG, PMU/CMU/TMU Bus I/F, MCTimer, WDT, RTC, KEYIF, SECKEY, TZPC | 100 MHz |

NOTE: Refer to Audio Subsystem Chapter for more details on MAUDIO block clocks.

5.2 Clock Declaration

The top-level clocks in Exynos 4412 are:

- Clocks from clock pads, namely, XRTCXTI, XXTI, and XUSBXTI.
- Clocks from CMUs
For instance, ARMCLK, ACLK, HCLK, and SCLK
ARMCLK specifies clock for Cortex-A9 MPCore (up to 800 MHz @ 1.0 V, 1 GHz @ 1.1 V).
ACKL, HCLK, PCLK specify bus clocks.
SCLK (Special clock) specifies all clocks except bus clocks and processor core clock.
- Clocks from USB PHY
- Clocks from HDMI_PHY
- Clocks from GPIO pads

5.2.1 Clocks from Clock Pads

The clock pads derive the clocks. They are:

- **XRTCXTI:** Specifies the clock generated from the crystal pad of 32.768 KHz with XRTCXTI and XRTCXTO pins. XRTCXTI and XRTCXTO are the two pins of crystal pad. RTC uses this clock as a source to the real-time clock. It requires a parallel resistance of 10 MΩ between the XUSBXTI and XUSBXTO pins.
- **XXTI:** Specifies the clock from external oscillator with XXTI pins. The input frequency ranges from 12 to 50 MHz. When XXTI is not used, it should be pulled-down.
- **XUSBXTI:** Specifies the clock from crystal pad with XUSBXTI and XUSBXTO pins. XUSBXTI and XUSBXTO use wide-range OSC pads. This clock is supplied to the USB PHY and the phase locked loops, namely, APLL, MPPLL, VPPLL, and EPPLL. Refer to Chapter 36 USB HOST and Chapter 37 USB DEVICE, for more information. We recommend using a 24 MHz crystal as the iROM design is based on the 24 MHz input clock. It requires parallel resistance of 5 MΩ between the XUSBXTI and XUSBXTO pins.

5.2.2 Clocks from CMU

CMUs generate internal clocks with intermediate frequencies using from clocks from the clock pads. They are:

- Clock pads, namely, XRTCXTI, XXTI, and XUSBXTI
- Four PLLs, namely, APLL, MPLL, EPLL, and VPLL
- USB PHY and HDMI PHY

Some of these clocks are selected, pre-scaled, and provided to the corresponding modules.

We recommend using 24 MHz input clock source for APLL, MPLL, EPLL, and VPLL.

The components to generate internal clocks are:

- APLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz.
- MPLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz.
- EPLL uses FINPLL as input to generate frequencies from 22 to 1400 MHz. This PLL generates a 192 MHz clock for the Audio Sub-system. It divides EPLL output to generate 24 MHz SLIMbus clock.
- VPLL uses FINPLL or SCLK_HDMI24M as input to generate frequencies from 22 to 1400 MHz. This PLL generates 54 MHz video clock or G3D clock.
- USB Device PHY uses XUSBXTI to generate frequencies of 30 and 48 MHz.
- HDMI PHY uses XUSBXTI to generate 54 MHz.

In typical Exynos 4412 applications,

- Cortex-A9 MPCore, CoreSight, and HPM use APLL.
- DRAM, system bus clocks, and other peripheral clocks like audio IPs, and SPI use MPLL and EPLL.
- Video clock uses VPLL.
- G3D uses MPLL or VPLL as input clock source.

Clock controllers allow bypassing of PLLs for low frequency clock. They also provide clock gating to each block, thereby reducing power consumption.

5.3 Clock Relationship

The clock relationship between various clocks are:

- CPU_BLK clocks
 - freq (ARMCLK) = freq (MOUTCORE)/n, where n = 1 to 16
 - freq (ACLK_COREM0) = freq (ARMCLK)/n, where n = 1 to 8
 - freq (ACLK_COREM1) = freq (ARMCLK)/n, where n = 1 to 8
 - freq (PERIPHCLK) = freq (ARMCLK)/n, where n = 1 to 8
 - freq (ATCLK) = freq (MOUTCORE)/n, where n = 1 to 8
 - freq (PCLK_DBG) = freq (ATCLK)/n, where n = 1 to 8
- DMC_BLK clocks
 - freq (SCLK_DMC) = freq (MOUTDMC_BUS)/n, where n = 1 to 8
 - freq (ACLK_DMCD) = freq (SCLK_DMC)/n, where n = 1 to 8
 - freq (ACLK_DMCP) = freq (ACLK_DMCD)/n, where n = 1 to 8
 - freq (ACLK_ACP) = freq (MOUTDMC_BUS)/n, where n = 1 to 8
 - freq (PCLK_ACP) = freq (ACLK_ACP)/n, where n = 1 to 8
 - freq (SCLK_C2C) = freq (MOUTC2C)/n, where n = 1 to 8
 - freq (ACLK_C2C) = freq (SCLK_C2C)/n, where n = 1 to 8
- LEFTBUS_BLK clocks
 - freq (ACLK_GDL) = freq (MOUTGDL)/n, where n = 1 to 8
 - freq (ACLK_GPL) = freq (ACLK_GDL)/n, where n = 1 to 8
- RIGHTBUS_BLK clocks
 - freq (ACLK_GDR) = freq (MOUTGDR)/n, where n = 1 to 8
 - freq (ACLK_GPR) = freq (ACLK_GDR)/n, where n = 1 to 8
- CMU_TOP clocks
 - freq (ACLK_400_MCUISP) = freq (MOUTACLK_400_mcuisp)/n, where n = 1 to 8
 - freq (ACLK_200) = freq (MOUTACLK_200)/n, where n = 1 to 8
 - freq (ACLK_100) = freq (MOUTACLK_100)/n, where n = 1 to 16
 - freq (ACLK_160) = freq (MOUTACLK_160)/n, where n = 1 to 8
 - freq (ACLK_133) = freq (MOUTACLK_133)/n, where n = 1 to 8
 - freq (SCLK_ONENAND) = freq (MOUTONENAND)/n, where n = 1 to 8
- MAUDIO_BLK clocks
 - freq (RP_CLK) = freq (MOUTASS)/n, where n = 1 to 16
 - freq (BUS_CLK) = freq (MOUTRP)/n, where n = 1 to 16

NOTE: [Figure 5-3](#) of Chapter 39 Audio Subsystem illustrates the clock names including iROM/iRAM and clock tree diagram of MAUDIO_BLK.

Caution: Ensure that the ratio between the SCLK_DMC and ACLK_DMCD frequency should be 2:1 or 1:1 always. Do not change this ratio during the running state of DMC. You should also ensure that the ratio between the SCLK_C2C and ACLK_C2C frequency should be 2:1. You should not change this ratio during the running state of C2C.

The values for high-performance operation are:

- freq (ARMCLK) = 1400 MHz
- freq (ACLK_COREM0) = 350 MHz
- freq (ACLK_COREM1) = 188 MHz
- freq (PERIPHCLK) = 1400 MHz
- freq (ATCLK) = 214 MHz
- freq (PCLK_DBG) = 107 MHz
- freq (SCLK_DMC) = 400 MHz
- freq (ACLK_DMCD) = 200 MHz
- freq (ACLK_DMCP) = 100 MHz
- freq (ACLK_ACP) = 200 MHz
- freq (PCLK_ACP) = 100 MHz
- freq (SCLK_C2C) = 400 MHz
- freq (ACLK_C2C) = 200 MHz
- freq (ACLK_GDL) = 200 MHz
- freq (ACLK_GPL) = 100 MHz
- freq (ACLK_GDR) = 200 MHz
- freq (ACLK_GPR) = 100 MHz
- freq (ACLK_400_MCUISP) = 400 MHz
- freq (ACLK_200) = 160 MHz
- freq (ACLK_100) = 100 MHz
- freq (ACLK_160) = 160 MHz
- freq (ACLK_133) = 133 MHz
- freq (SCLK_ONENAND) = 160 MHz

The PLL operations are:

- APLL mainly drives the CPU_BLK clocks. It generates frequencies up to 1.4 GHz with a duty ratio of 49:51. APLL also generates DMC_BLK, LEFTBUS_BLK, RIGHTBUS_BLK, and CMU_TOP clocks as supplement of MPLL.
- MPLL mainly drives the DMC_BLK, LEFTBUS_BLK, RIGHTBUS_BLK, and CMU_TOP clocks. It generates frequencies up to 1 GHz with a duty ratio of 49:51. MPLL also generates CPU_BLK clocks when it blocks APLL for locking during the Dynamic Voltage Frequency Scaling (DVFS).
- EPLL mainly generates an audio clock.
- VPLL mainly generates video system operating clock of 54 MHz, or a G3D clock, or 440 MHz clock at 1.1 V.

5.3.1 Recommended PLL PMS Value for APLL and MPLL

Table 5-2 describes the recommended PLL PMS value for APLL and MPLL.

Table 5-2 APLL and MPLL PMS Value

| F _{IN} (MHz) | Target F _{OUT} (MHz) | P | M | S | F _{OUT} (MHz) |
|-----------------------|-------------------------------|---|-----|---|------------------------|
| 24 | 200 | 3 | 100 | 2 | 200 |
| 24 | 300 | 4 | 200 | 2 | 300 |
| 24 | 400 | 3 | 100 | 1 | 400 |
| 24 | 500 | 3 | 125 | 1 | 500 |
| 24 | 600 | 4 | 200 | 1 | 600 |
| 24 | 700 | 3 | 175 | 1 | 700 |
| 24 | 800 | 3 | 100 | 0 | 800 |
| 24 | 900 | 4 | 150 | 0 | 900 |
| 24 | 1000 | 3 | 125 | 0 | 1000 |
| 24 | 1100 | 6 | 275 | 0 | 1100 |
| 24 | 1200 | 4 | 200 | 0 | 1200 |
| 24 | 1300 | 6 | 325 | 0 | 1300 |
| 24 | 1400 | 3 | 175 | 0 | 1400 |

NOTE: Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table.
If you have to use other values, please contact us.

5.3.2 Recommended PLL PMS Value for EPLL

Table 5-3 describes the recommended PLL PMS value for EPLL.

Table 5-3 EPLL PMS Value

| F _{IN} (MHz) | Target F _{OUT} (MHz) | P | M | S | K | F _{OUT} (MHz) |
|-----------------------|-------------------------------|---|-----|---|-------|------------------------|
| 24 | 90 | 2 | 60 | 3 | 0 | 90 |
| 24 | 180 | 2 | 60 | 2 | 0 | 180 |
| 24 | 180.6 | 3 | 90 | 2 | 19661 | 108.6 |
| 24 | 180.6336 | 3 | 90 | 2 | 20762 | 180.6336 |
| 24 | 192 | 2 | 64 | 2 | 0 | 192 |
| 24 | 200 | 3 | 100 | 2 | 0 | 200 |
| 24 | 400 | 3 | 100 | 1 | 0 | 400 |
| 24 | 408 | 2 | 68 | 1 | 0 | 408 |
| 24 | 416 | 3 | 104 | 1 | 0 | 416 |

NOTE:

1. K value description "Positive value (Negative value)":
Positive values is that you should write to EPLLCON/VPLLCON register.
Negative value is that you can calculate PLL output frequency with it.
2. Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table.
If you have to use other values, please contact us.

5.3.3 Recommended PLL PMS Value for VPLL

Table 5-4 describes the recommended PLL PMS value for VPLL.

Table 5-4 VPLL PMS Value

| F _{IN} (MHz) | Target F _{OUT} (MHz) | P | M | S | K | MFR | MRR | SSCG_EN |
|-----------------------|-------------------------------|---|-----|---|---|-----|-----|---------|
| 24 | 100 | 3 | 100 | 3 | 0 | – | – | 0 |
| | 160 | 3 | 160 | 3 | 0 | – | – | 0 |
| | 266 | 3 | 133 | 2 | 0 | – | – | 0 |
| | 350 | 3 | 175 | 2 | 0 | – | – | 0 |
| | 440 | 3 | 110 | 1 | 0 | – | – | 0 |

NOTE:

1. Although there is an equation for choosing PMS values, we strongly recommend only the values in the above table.
If you have to use other values, please contact us.
2. K value description "Positive value (Negative value)":
Positive values is that you should write to EPLLCON/VPLLCON register.
Negative value is that you can calculate PLL output frequency with it.

5.4 Clock Generation

[Figure 5-1](#) and [Figure 5-2](#), illustrates the block diagram of the clock generation logic. The clock generator consists of an external crystal clock that is connected to the oscillation amplifier. The PLL converts the incoming low frequency to a high frequency clock that is required by the Exynos 4412. The clock generator also includes a built-in logic to stabilize the clock frequency for each system reset. The clock requires a specified time for stabilization.

[Figure 5-1](#) and [Figure 5-2](#) illustrates the two types of clock MUX. Clock MUX in grey color represents glitch-free clock MUX that is free of glitches while changing the clock selection. Clock MUX in white color represents non-glitch-free clock MUX that can suffer from glitches while changing the clock sources. You have to be careful while using each clock MUX.

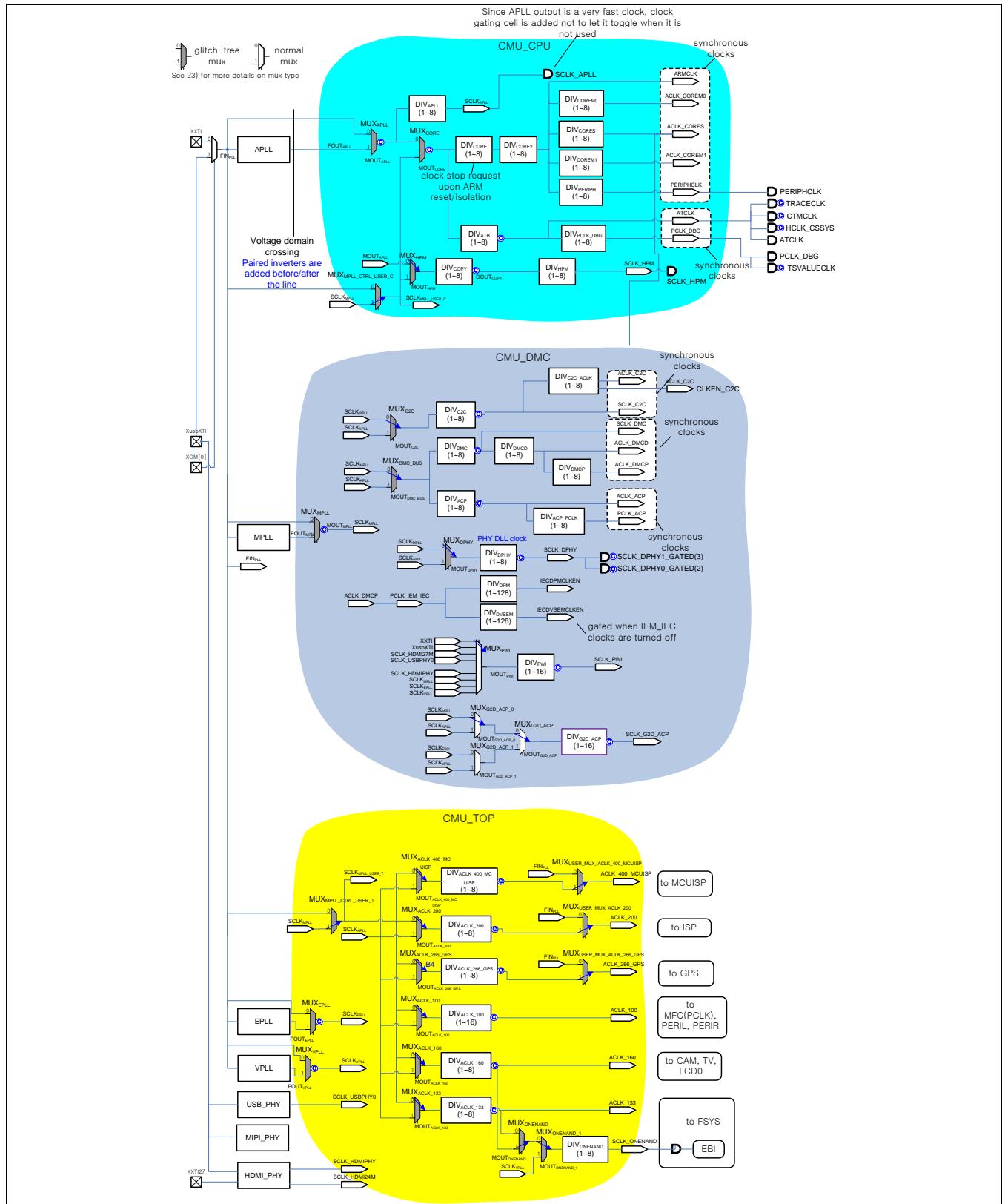
For glitch-free MUX, you should ensure that all clock sources are running while changing the clock selection. If not, it implies that the clock selection process is not complete and it results in clock output having unknown states. The clock MUX status registers are identified with a keyword that starts with CLK_MUX_STAT

For non-glitch-free clock MUX, glitches may occur while changing the clock selection. To prevent glitch signals, we recommend disabling the output of a non-glitch-free MUX before any change of clock selection. After completing the clock change, you can re-enable the output of the non-glitch-free clock MUX. This is done to ensure that there are no glitches resulting due to the clock change selection. The outputs of non-glitch-free MUXES are masked by the clock source mask control registers. The clock source mask control registers are identified with a keyword that starts with CLK_SRC_MASK.

[Figure 5-1](#) and [Figure 5-2](#) illustrates a clock divider that indicates possible dividing value in parentheses. The dividing values can be changed by clock divider registers during run-time. Some clock dividers have only one dividing value and you are not allowed to change the dividing value.

[Figure 5-1](#) illustrates the Exynos 4412 Clock Generation Circuit (CPU, BUS, DRAM, and ISP Clocks) diagram.

[Figure 5-2](#) illustrates the Exynos 4412 Clock Generation Circuit (Special Clocks) diagram.



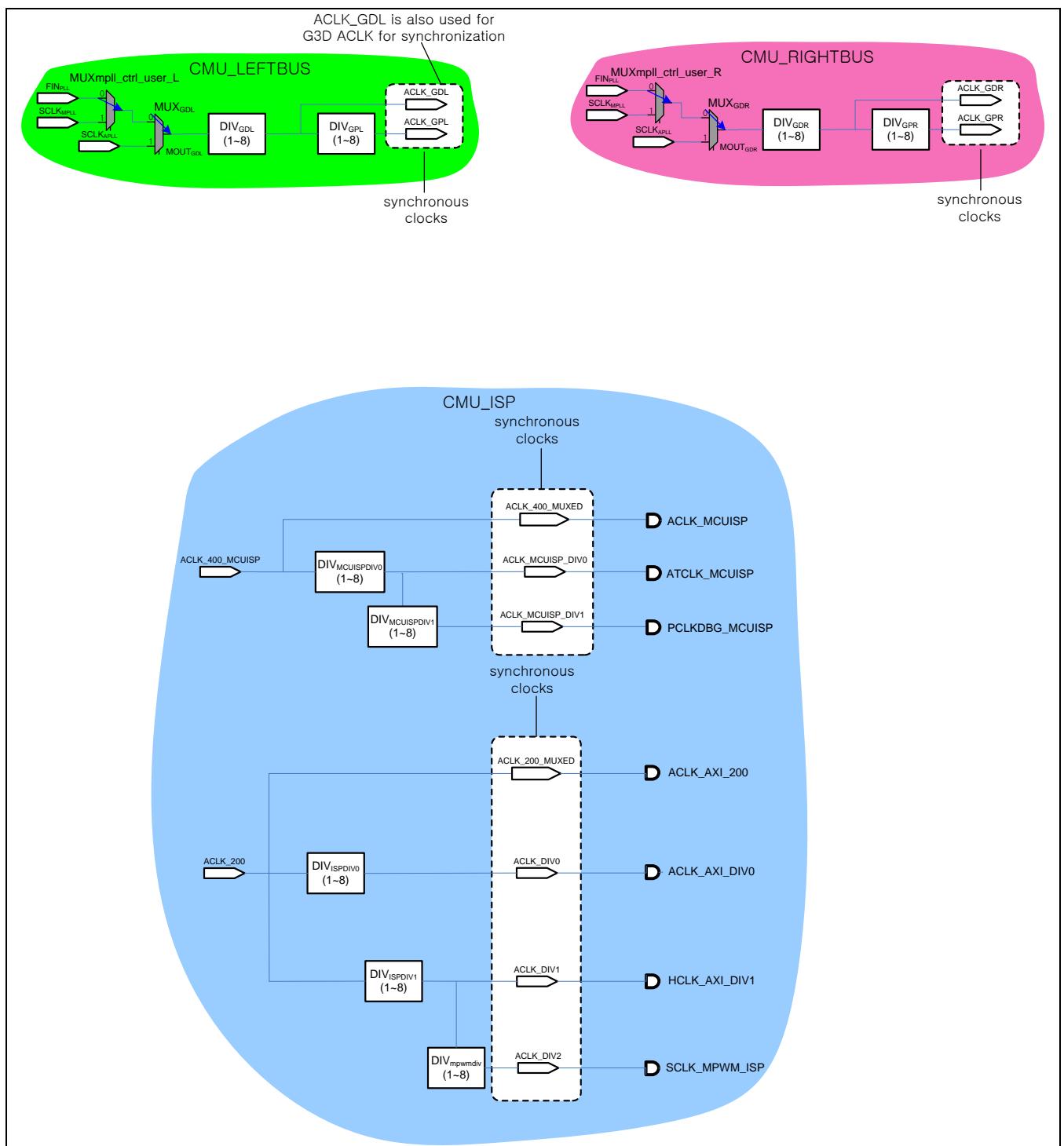
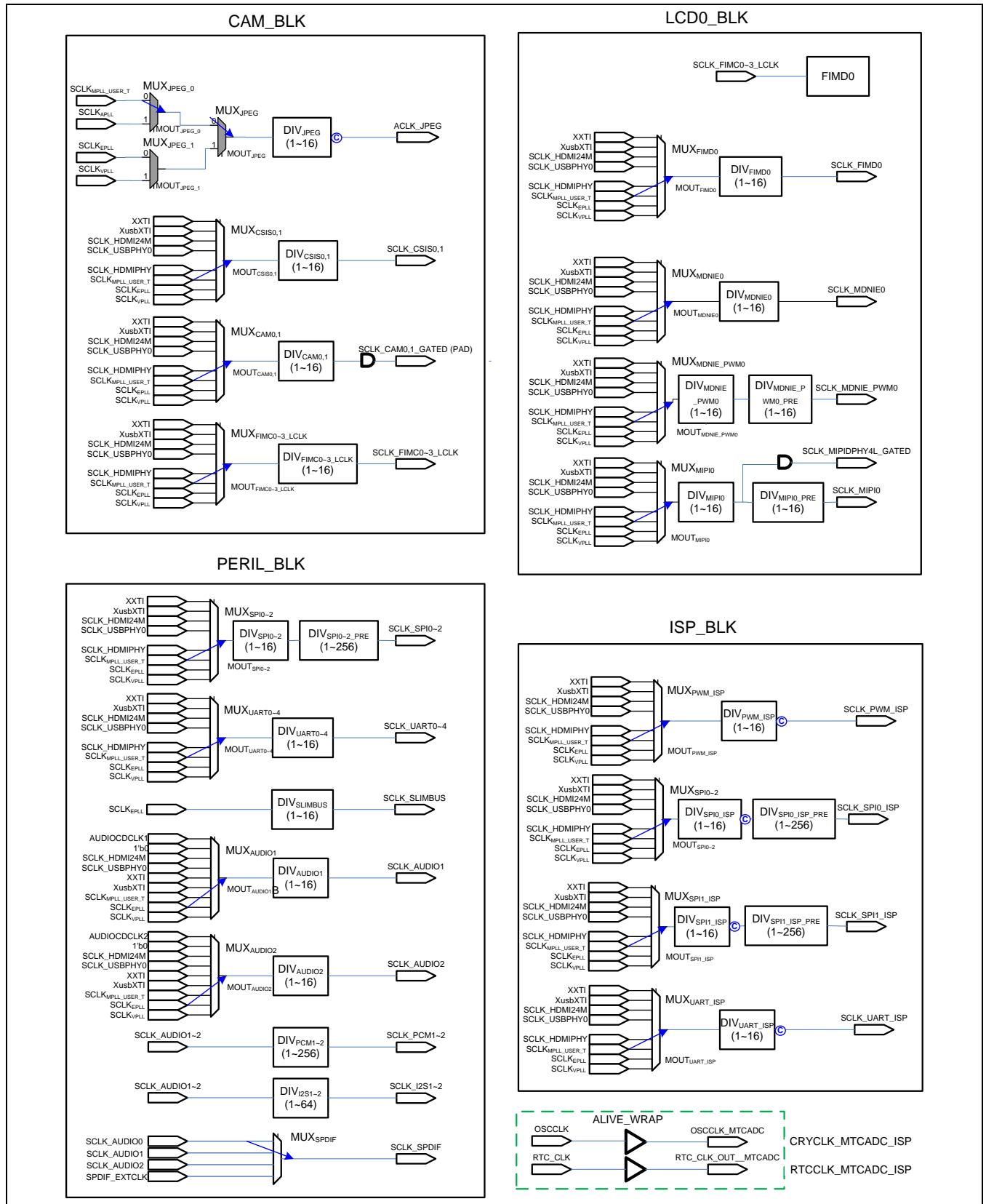


Figure 5-1 Exynos 4412 Clock Generation Circuit (CPU, BUS, DRAM, ISP Clocks)



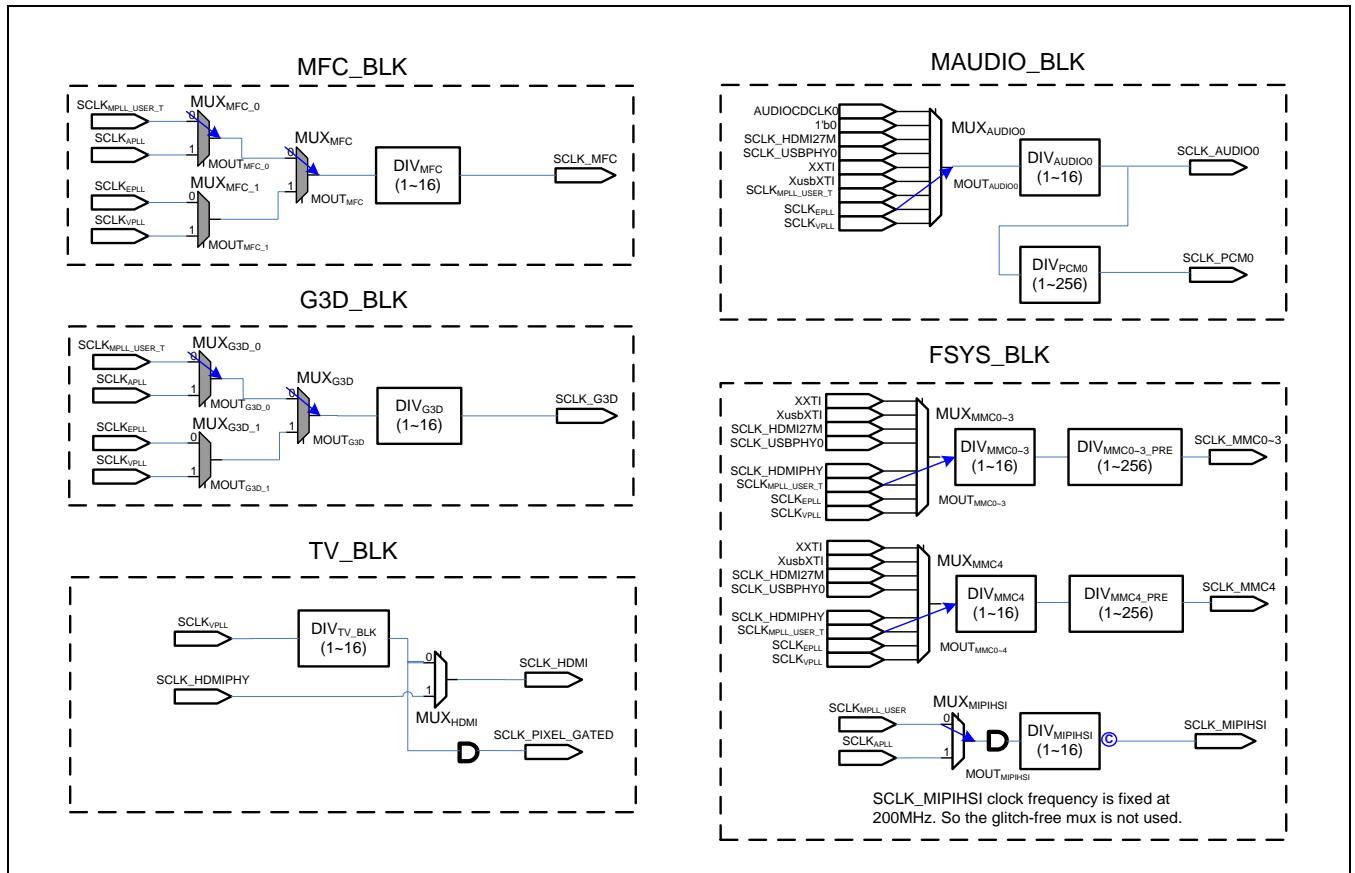


Figure 5-2 Exynos 4412 Clock Generation Circuit (Special Clocks)

NOTE: The SCLKuser_mppll In [Figure 5-2](#) means SCLKuser_mppll_T.

Caution: In [Figure 5-1](#) and [Figure 5-2](#), the MUX's with grey color are glitch-free. For glitch-free clock MUX, ensure that all clock sources are running while changing the clock selection. For clock dividers, ensure that input clock is running while changing the divider value.

5.5 Clock Configuration Procedure

The rules for changing the clock configuration are:

- All inputs of a glitch-free MUX should run.
- When a PLL is turned OFF, you should not select the output of PLL.

The basic SFR configuration requires change in system clock divider values that are:

- CLK_DIV_CPU0[31:0] = target value0
- CLK_DIV_DMC0[31:0] = target value1
- CLK_DIV_TOP[31:0] = target value2
- CLK_DIV_LEFTBUS[31:0] = target value3
- CLK_DIV_RIGHTBUS[31:0] = target value4

Change the divider values for special clocks by setting CLK_DIV_XXX SFRs in CMU_TOP

- CLK_DIV_XXX[31:0] = target value

The following sequence shows turn on PLL procedure.

```

Change PLL PMS values
Set PMS values;
    // Set PDIV, MDIV, and SDIV values (Refer to (A, M, E, V) PLL_CON0 SFRs)
Change other PLL control values
(A, M, E, V)PLL_CON1[31:0]      = target value;
    // Set K, AFC, MRR, MFR values if necessary (Refer to (A, M, E, V)PLL_CON1 SFRs)
Turn on a PLL
(A, M, E, V)PLL_CON0[31]      = 1;
    // Turn on a PLL (Refer to (A, M, E, V)PLL_CON0 SFRs)

wait_lock_time;      // Wait until the PLL is locked

MUX_(A, M, E, V)PLL_SEL      = 1;
    // Select the PLL output clock instead of input reference clock,
    after PLL output clock is stabilized.
    (Refer to CLK_SRC_CPU SFR for APLL and MPLL, CLK_SRC_TOP0 for EPLL and VPLL)
Once a PLL is turned on, do not turn it off.

```

5.5.1 Clock Gating

Exynos 4412 can disable the clock operation of each IP, if it does not require. This reduces the dynamic power consumption.

The two types of clock gating control register to disable or enable clock operations are:

- Clock gating control register for function blocks
- Clock gating control register for IP

The two clock gating control registers are ANDed to generate the final clock gating enable signal. As a result, if it turns OFF either of the two registers filed, then the resulting clock will stop. For example, to stop the clocks provided to the Mixer module, you should set the CLK_MIXER field in CLK_GATE_IP_TV register to 0 or CLK_TV field in CLK_GATE_BLOCK register to 0. For latter case, all clocks in TV block, not only MIXER clocks, are turned off.

Caution: Ensure that the software does not access the IPs whose clock is gated, as it may cause system failure.

5.5.2 Clock Diving

Whenever clock divider control register is changed, it is recommended to check clock divider status registers before using the new clock output. This guarantees the corresponding divider finishes changing to a new dividing value before its output is used by other modules.

5.6 Special Clock Description

Special Clock Description section describes special clock in Exynos 4412.

5.6.1 Special Clock Table

[Table 5-5](#) describes the special clocks in Exynos 4412.

Table 5-5 Special Clocks in Exynos 4412

| Name | Description | Range | Source |
|--|---|---|---|
| SCLK_ONENAND | ONENAND operating clock | 160 MHz | ACLK_160, ACLK_133 |
| SCLK_G3D | G3D core operating clock | 440 MHz | SCLK _{APLL} , SCLK _{MPLL} , SCLK _{EPLL} , SCLK _{VPLL} |
| SCLK_G2D | G2D core operating clock | 200 MHz | SCLK _{APLL} , SCLK _{MPLL} , SCLK _{EPLL} , SCLK _{VPLL} |
| SCLK_MFC | MFC core operating clock | 200 MHz | SCLK _{APLL} , SCLK _{MPLL} , SCLK _{EPLL} , SCLK _{VPLL} |
| SCLK_CAM0, SCLK_CAM1 | Reference clock for external CAM device | Range varies in accordance to the CAM specifications. | All possible clock sources |
| SCLK_CSIS0, SCLK_CSIS1 | CSIS operating clock | 160 MHz | All possible clock sources |
| SCLK_FIMC_LCLK0, SCLK_FIMC_LCLK1, SCLK_FIMC_LCLK2, SCLK_FIMC_LCLK3 | FIMC core operating clock | 160 MHz | All possible clock sources |
| SCLK_FIMD0 | FIMD operating clock | 100 MHz | All possible clock sources |
| SCLK_MDNIE0 | MDNIE operating clock | 100 MHz | All possible clock sources |
| SCLK_MDNIE_PWM0 | MDNIE PWM clock | 100 MHz | All possible clock sources |
| SCLK_MIPI0 | MIPI DSIM clock | 100 MHz | All possible clock sources |
| SCLK_MIPIDPHY4L | MIPI DPHY 4 Lane clock | 800 MHz | All possible clock sources |
| SCLK_HDMI | HDMI LINK clock | 148.5 MHz | All possible clock sources |
| SCLK_PIXEL | HDMI PIXEL clock | 148.5 MHz | All possible clock sources |
| SCLK_SPDIF | SPDIF operating clock | 83 MHz | SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2 |
| SCLK_MMC0, SCLK_MMC1, SCLK_MMC2, SCLK_MMC3, SCLK_MMC4 | HSMMC operating clock | 50 MHz | All possible clock sources |
| SCLK_USBPHY0 | USB device clock | 48 MHz | USB Device PHY clock out |

| Name | Description | Range | Source |
|--|--|-------------|--|
| SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2 | AUDIO operating clock (I2S) | 100 MHz | All possible clock sources, AUDIOCDCLKx |
| SCLK_PCM0, SCLK_PCM1, SCLK_PCM2 | AUDIO operating clock (PCM) | 5 MHz | SCLK_AUDIO0, SCLK_AUDIO1, SCLK_AUDIO2 |
| SCLK_PWI | IEM APC operating clock | 6 to 30 MHz | All possible clock sources |
| SCLK_KEY | KEY I/F or TSADC filter clock (fixed clock) | 24 MHz | XXTI, XUSBXTI |
| SCLK_SPI0, SCLK_SPI1, SCLK_SPI2 | SPI operating clock | 100 MHz | All possible clock sources |
| SCLK_UART0, SCLK_UART1, SCLK_UART2, SCLK_UART3, SCLK_UART4 | UART operating clock | 200 MHz | All possible clock sources |
| SCLK_SLIMBUS | SLIMBUS clock | 25 MHz | SCLK_EPLL |
| ACLK_JPEG | JPEG core operating clock | 160 MHz | SCLK _{APLL} , SCLK _{MPLL} , SCLK _{EPLL} , SCLK _{VPLL} |
| SCLK_PWM_ISP | PWM_ISP operating clock | 66 MHz | All possible clock sources |
| SCLK_SPI0_ISP | SPI0_ISP operating clock | 100 MHz | All possible clock sources |
| SCLK_SPI1_ISP | SPI1_ISP operating clock | 100 MHz | All possible clock sources |
| SCLK_UART_ISP | UART_ISP operating clock | 66 MHz | All possible clock sources |
| SCLK_MIPIHSI | MIPIHSI core operating clock | 200 MHz | SCLK _{APLL} , SCLK _{MPLL} , |

- All possible clock sources include XXTI, XUSBXTI, SCLK_HDMI24M, SCLK_USBPHY, SCLK_HDMIPHY, SCLK_{MPLL}, SCLK_{EPLL}, and SCLK_{VPLL}.
- XXTI and XUSBXTI refer to external crystal.
- SCLK_USBPHY refers to USB PHY 48 MHz output clock.
- SCLK_HDMI24M refers to HDMI PHY (24 MHz reference clock for XUSBXTI) output.
- SCLK_HDMIPHY refers to HDMI PHY (PIXEL_CLKO) output clock.
- SCLK_{MPLL}, SCLK_{EPLL}, and SCLK_{VPLL} refer to the output clock of MPLL, EPLL, and VPLL, respectively.

[Table 5-6](#) describes the I/O clocks in Exynos 4412.

Table 5-6 I/O Clocks in Exynos 4412

| Name | I/O | Pad | GPIO Function | Range | Description |
|--|-------|---|---|------------|-------------------|
| IOCLK_AC97 | Input | Xi2s1SCLK | Func2: AC97BITCLK | 12.288 MHz | AC97 Bit Clock |
| IOCLK_I2S0, IOCLK_I2S1, IOCLK_I2S2 | Input | Xi2s0CDCLK Xi2s1CDCLK Xpcm2EXTCLK | Func0: I2S_0_CDCLK Func0: I2S_1_CDCLK Func2: I2S_2_CDCLK | 83.4 MHz | I2S CODEC Clock |
| IOCLK_PCM0, IOCLK_PCM1, IOCLK_PCM2 | Input | Xi2s0CDCLK Xi2s1CDCLK Xpcm2EXTCLK | Func1: PCM_0_EXTCLK Func1: PCM_1_EXTCLK Func0: PCM_2_EXTCLK | 83.4 MHz | PCM CODEC Clock |
| IOCLK_SPDIF | Input | Xpcm2EXTCLK | Func1: SPDIF_EXTCLK | 36.864 MHz | SPDIF Input Clock |

5.7 CLKOUT

You can use the XCLKOUT port to monitor certain clocks in Exynos 4412. The six CMUs in Exynos 4412 contain the CLKOUT control logic. If necessary, you can select and divide one of the clocks in the CMU. It generates CLKOUT signal from each CMU and feeds this into the power management unit. It is then muxed with CLKOUT signals and XXTI, XUSBXTI, RTC_TICK_SRC, and RTCCLK.

[Figure 5-3](#) illustrates the CLKOUT control logic in Exynos 4412 .

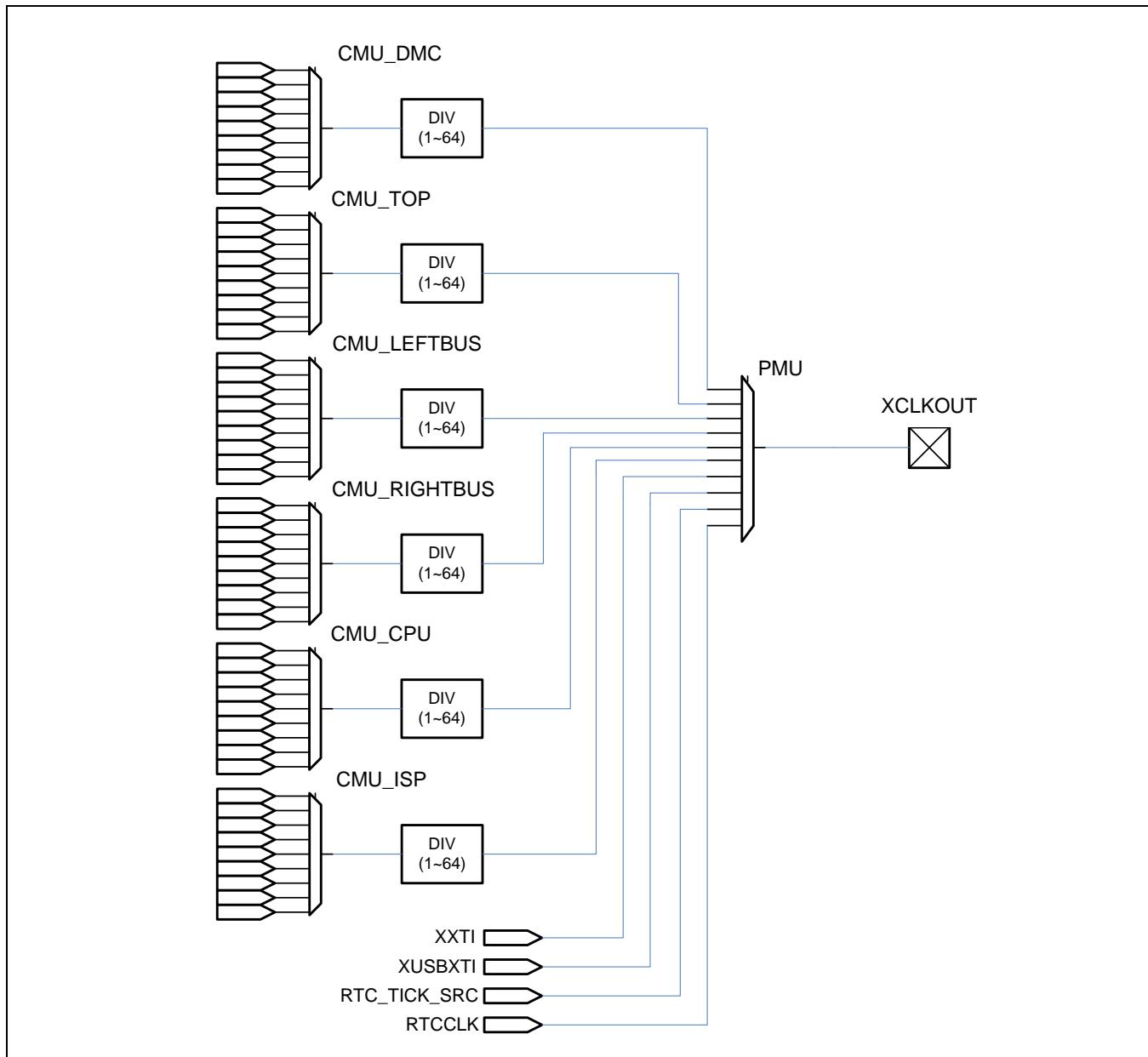


Figure 5-3 Exynos 4412 CLKOUT Control Logic

[Table 5-7](#) describes the CLKOUT input clock selection information.

Table 5-7 CLKOUT Input Clock Selection Information

| No. | CMU_CPU | CMU_DMC | CMU_TOP | CMU_RIGHTBUS | CMU_LEFTBUS | CMU_ISP | PMU |
|-----|-------------|-------------|--------------------|--------------|-------------|----------------|--------------|
| 0 | APLL_FOUT/2 | ACLK_DMCD | EPLL_FOUT | SCLK_MPLL/2 | SCLK_MPLL/2 | ACLK_MCUISP | CMU_DMC |
| 1 | Reserved | ACLK_DMCP | VPLL_FOUT | SCLK_APLL/2 | SCLK_APLL/2 | PCLKDBG_MCUISP | CMU_TOP |
| 2 | Reserved | ACLK_ACP | SCLK_HDMI24M | ACLK_GDR | ACLK_GDL | ACLK_DIV0 | CMU_LEFTBUS |
| 3 | Reserved | PCLK_ACP | SCLK_USBPHY0 | ACLK_GPR | ACLK_GPL | ACLK_DIV1 | CMU_RIGHTBUS |
| 4 | ARMCLK/2 | SCLK_DMC | Reserved | - | - | SCLK_MPWM_ISP | CMU_CPU |
| 5 | ACLK_COREM0 | SCLK_DPHY | SCLK_HDMIPHY | - | - | - | XXTI |
| 6 | ACLK_COREM1 | MPLL_FOUT/2 | AUDIOCDC_LK0 | - | - | - | XUSBXTI |
| 7 | ACLK_CORES | SCLK_PWI | AUDIOCDC_LK1 | - | - | - | RTC_TICK_SRC |
| 8 | ATCLK | - | AUDIOCDC_LK2 | - | - | - | RTCCLK |
| 9 | PERIPHCLK | SCLK_C2C | SPDIF_EXTCLK | - | - | - | - |
| 10 | PCLK_DBG | ACLK_C2C | ACLK_160 | - | - | - | - |
| 11 | SCLK_HPM | - | ACLK_133 | - | - | - | - |
| 12 | - | - | ACLK_200 | - | - | - | - |
| 13 | - | - | ACLK_100 | - | - | - | - |
| 14 | - | - | SCLK_MFC | - | - | - | - |
| 15 | - | - | SCLK_G3D | - | - | - | - |
| 16 | - | - | ACLK_400_MCUISP | - | - | - | - |
| 17 | - | - | CAM_A_PCLK | - | - | - | - |
| 18 | - | - | CAM_B_PCLK | - | - | - | - |
| 19 | - | - | S_RXBYTE_CLKHS0_2L | - | - | - | - |
| 20 | - | - | S_RXBYTE_CLKHS0_4L | - | - | - | - |
| 21 | - | - | RX_HALF_ | - | - | - | - |

| No. | CMU_CPU | CMU_DMC | CMU_TOP | CMU_RIGHTBUS | CMU_LEFTBUS | CMU_ISP | PMU |
|-----|---------|---------|------------------------|--------------|-------------|---------|-----|
| | | | BYTE_CLK_CSIS0 | | | | |
| 22 | - | - | RX_HALF_BYTE_CLK_CSIS1 | - | - | - | - |
| 23 | - | - | SCLK_JPEG | - | - | - | - |
| 24 | - | - | SCLK_PWM_ISP | - | - | - | - |
| 25 | - | - | SCLK_SPI0_ISP | - | - | - | - |
| 26 | - | - | SCLK_SPI1_ISP | - | - | - | - |
| 27 | - | - | SCLK_UART_ISP | - | - | - | - |
| 28 | - | - | SCLK_MIPIHSI | - | - | - | - |
| 29 | - | - | SCLK_HDMI | - | - | - | - |
| 30 | - | - | SCLK_FIMD0 | - | - | - | - |
| 31 | - | - | SCLK_PCM0 | - | - | - | - |

5.8 I/O Description

[Table 5-8](#) describes the I/O.

Table 5-8 I/O Description

| Signal | I/O | Description | Pad | Type |
|------------|--------------|--------------------------------------|-------------|-----------|
| XXTI | Input | External oscillator pad | XXTI | Dedicated |
| XUSBXTI | Input | Input pad for crystal | XUSBXTI | Dedicated |
| XUSBXTO | Output | Output pad for crystal | XUSBXTO | Dedicated |
| EPLLFILTER | Input/Output | Pad for EPLL loop filter capacitance | XEPLLFILTER | Dedicated |
| VPLLFILTER | Input/Output | Pad for VPLL loop filter capacitance | XVPLLFILTER | Dedicated |
| XCLKOUT | Output | Clock out pad | XCLKOUT | Dedicated |

5.9 Register Description

The clock controller controls PLLs and clock generation units. This section describes the usage of Special Functional Registers (SFRs) in the clock controller. Do not change any reserved area. Any change in the reserved area leads to an unexpected behavior.

The address map of Exynos 4412 clock controller consists of six CMUs. They are, CMU_LEFTBUS, CMU_RIGHTBUS, CMU_TOP, CMU_DMC, CMU_CPU, and CMU_ISP. Each CMU uses an address space of 16 KB for SFRs. The internal structure of address space for each CMU is similar for all CMUs.

The six categories into which the address space is divided are:

- Use 0x000 to 0x1FF for PLL control : PLL lock time and control
- Use 0x200 to 0x4FF for MUX control : MUX selection, output masking, and status
- Use 0x500 to 0x6FF for clock division : Divider ratio and status
- 0x700 to 0x8FF is reserved and you are not allowed to access the region.
- Use 0x900 to 0x9FF for clock gating control : Clock gating of IPs and function blocks
- Use 0xA00 to 0xAFF for CLKOUT : CLKOUT input clock selection and divider ratio

NOTE: The CLK_GATE_IP_XXX registers in the CMU_LEFTBUS and CMU_RIGHTBUS are located at 0x800. Additionally, some CMUs use addresses beyond 0xAFF for other functions such as CPU control functions in CMU_CPU. Refer to register description for more information.

In [Figure 5-4](#), XXX in the register name shall be replaced with the function block name, i.e., LEFTBUS, RIGHTBUS, TOP, CAM, TV, MFC, G3D, IMAGE, LCD0, LCD1, MAUDIO, FSYS, PERIL, and PERIR.

[Figure 5-4](#) illustrates the Exynos 4412 clock controller address map.

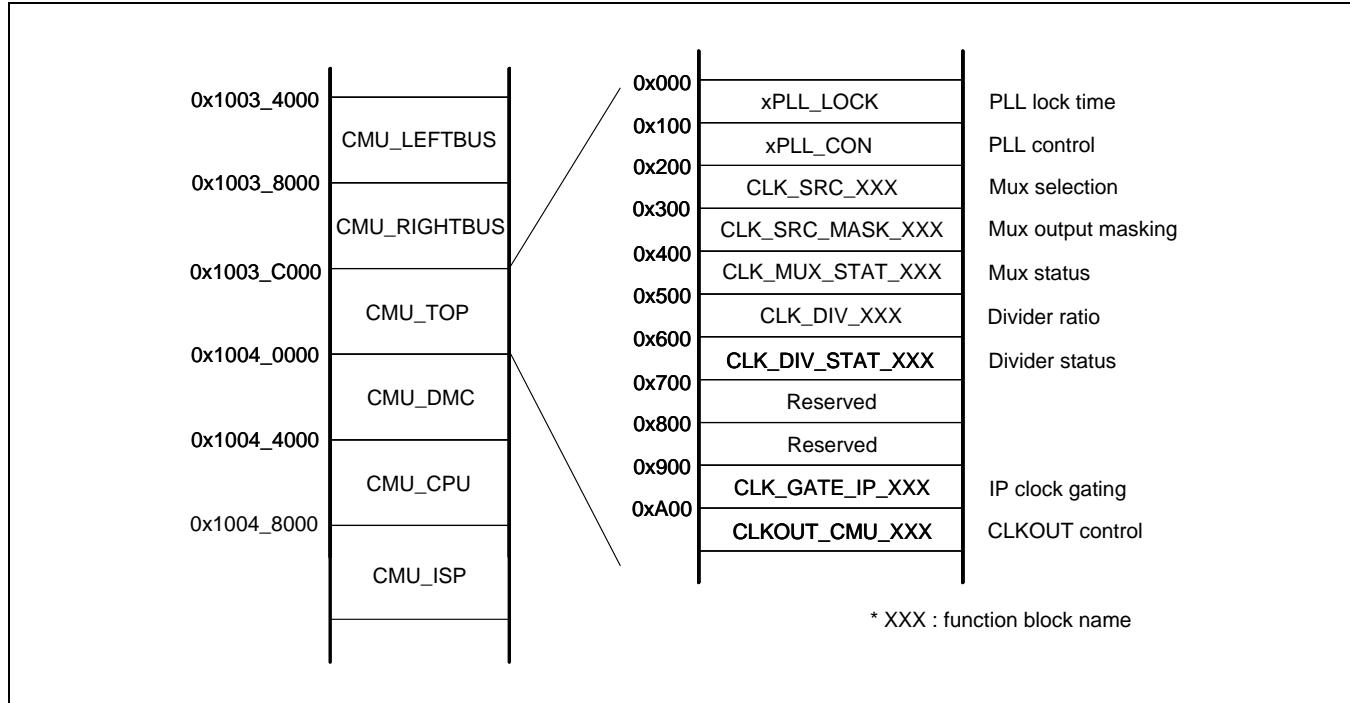


Figure 5-4 Exynos 4412 Clock Controller Address Map

5.9.1 Register Map Summary

- Base Address: 0x1003_0000

| Register | Offset | Description | Reset Value |
|-----------------------------|------------------|---|-------------|
| CLK_SRC_LEFTBUS | 0x4200 | Selects clock source for CMU_LEFTBUS | 0x0000_0000 |
| RSVD | 0x4204 to 0x43FF | Reserved | Undefined |
| CLK_MUX_STAT_LEFTBUS | 0x4400 | Clock MUX status for CMU_LEFTBUS | 0x0000_0011 |
| RSVD | 0x4404 to 0x44FF | Reserved | Undefined |
| CLK_DIV_LEFTBUS | 0x4500 | Sets clock divider ratio for CMU_LEFTBUS | 0x0000_0000 |
| RSVD | 0x4504 to 0x45FF | Reserved | Undefined |
| CLK_DIV_STAT_LEFTBUS | 0x4600 | Clock divider status for CMU_LEFTBUS | 0x0000_0000 |
| RSVD | 0x4604 to 0x47FC | Reserved | Undefined |
| CLK_GATE_IP_LEFTBUS | 0x4800 | Control IP clock gating for LEFTBUS_BLK | 0xFFFF_FFFF |
| RSVD | 0x4804 to 0x492C | Reserved | Undefined |
| CLK_GATE_IP_IMAGE | 0x4930 | Control IP clock gating for IMAGE_SS | 0xFFFF_FFFF |
| RSVD | 0x4934 to 0x49FC | Reserved | Undefined |
| CLKOUT_CMU_LEFTBUS | 0x4A00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_LEFTBUS_DIV_STAT | 0x4A04 | Clock divider status for CLKOUT | 0x0000_0000 |
| RSVD | 0x4A08 to 0x81FF | Reserved | Undefined |
| CLK_SRC_RIGHTBUS | 0x8200 | Selects clock source for CMU_RIGHTBUS | 0x0000_0000 |
| RSVD | 0x8204 to 0x83FF | Reserved | Undefined |
| CLK_MUX_STAT_RIGHTBUS | 0x8400 | Clock MUX status for CMU_RIGHTBUS | 0x0000_0011 |
| RSVD | 0x8404 to 0x84FF | Reserved | Undefined |
| CLK_DIV_RIGHTBUS | 0x8500 | Sets clock divider ratio for CMU_RIGHTBUS | 0x0000_0000 |
| RSVD | 0x8504 to 0x85FF | Reserved | Undefined |
| CLK_DIV_STAT_RIGHTBUS | 0x8600 | Clock divider status for CMU_RIGHTBUS | 0x0000_0000 |
| RSVD | 0x8604 to 0x87FC | Reserved | Undefined |
| CLK_GATE_IP_RIGHTBUS | 0x8800 | Control IP clock gating for RIGHTBUS_BLK | 0xFFFF_FFFF |
| RSVD | 0x8804 to | Reserved | Undefined |

| Register | Offset | Description | Reset Value |
|------------------------------|------------------|--|-------------|
| | 0x895C | | |
| CLK_GATE_IP_PERIR | 0x8960 | Controls IP clock gating for PERIR_S | 0xFFFF_FFFF |
| RSVD | 0x8964 to 0x89FC | Reserved | Undefined |
| CLKOUT_CMU_RIGHTBUS | 0x8A00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_RIGHTBUS_DIV_STAT | 0x8A04 | Clock divider status for CLKOUT | 0x0000_0000 |
| RSVD | 0x8A08 to 0xC00F | Reserved | Undefined |
| EPLL_LOCK | 0xC010 | Controls PLL locking period for EPLL | 0x0000_0FFF |
| RSVD | 0xC014 to 0xC01F | Reserved | Undefined |
| VPLL_LOCK | 0xC020 | Controls PLL locking period for VPLL | 0x0000_0FFF |
| RSVD | 0xC024 to 0xC10F | Reserved | Undefined |
| EPLL_CON0 | 0xC110 | Controls PLL output frequency for EPLL | 0x0060_0302 |
| EPLL_CON1 | 0xC114 | Controls PLL output frequency for EPLL | 0x6601_0000 |
| EPLL_CON2 | 0xC118 | Controls PLL output frequency for EPLL | 0x0000_0080 |
| RSVD | 0xC11C | Reserved | Undefined |
| VPLL_CON0 | 0xC120 | Controls PLL output frequency for VPLL | 0x006F_0302 |
| VPLL_CON1 | 0xC124 | Controls PLL output frequency for VPLL | 0x6601_6000 |
| VPLL_CON2 | 0xC128 | Controls PLL output frequency for VPLL | 0x0000_0080 |
| RSVD | 0xC12C to 0xC20C | Reserved | Undefined |
| CLK_SRC_TOP0 | 0xC210 | Selects clock source for CMU_TOP0 | 0x0000_0000 |
| CLK_SRC_TOP1 | 0xC214 | Selects clock source for CMU_TOP1 | 0x0000_0000 |
| RSVD | 0xC218 to 0xC21F | Reserved | Undefined |
| CLK_SRC_CAM0 | 0xC220 | Selects clock source for CAM_BLK | 0x1111_1111 |
| CLK_SRC_TV | 0xC224 | Selects clock source for TV_BLK | 0x0000_0000 |
| CLK_SRC_MFC | 0xC228 | Selects clock source for MFC_BLK | 0x0000_0000 |
| CLK_SRC_G3D | 0xC22C | Selects clock source for G3D_BLK | 0x0000_0000 |
| RSVD | 0xC230 | Reserved | Undefined |
| CLK_SRC_LCD | 0xC234 | Selects clock source for LCD_BLK | 0x0000_1111 |
| CLK_SRC_ISP | 0xC238 | Selects clock source for ISP_BLK | 0x0000_1111 |
| CLK_SRC_MAUDIO | 0xC23C | Selects clock source for AUDIO_BLK | 0x0000_0005 |
| CLK_SRC_FSYS | 0xC240 | Selects clock source for FSYS_BLK | 0x0001_1111 |
| RSVD | 0xC244 to 0xC24C | Reserved | Undefined |

| Register | Offset | Description | Reset Value |
|---------------------|------------------|---|-------------|
| CLK_SRC_PERIL0 | 0xC250 | Selects clock source for connectivity IPs | 0x0001_1111 |
| CLK_SRC_PERIL1 | 0xC254 | Selects clock source for connectivity IPs | 0x0111_0055 |
| CLK_SRC_CAM1 | 0xC258 | Selects clock source for CAM_BLK | 0x0000_0000 |
| RSVD | 0xC25C to 0xC31F | Reserved | Undefined |
| CLK_SRC_MASK_CAM0 | 0xC320 | Clock source mask for CAM_BLK | 0x1111_1111 |
| CLK_SRC_MASK_TV | 0xC324 | Clock source mask for TV_BLK | 0x0000_0111 |
| RSVD | 0xC328 to 0xC333 | Reserved | Undefined |
| CLK_SRC_MASK_LCD | 0xC334 | Clock source mask for LCD_BLK | 0x0000_1111 |
| CLK_SRC_MASK_ISP | 0xC338 | Clock source mask for ISP_BLK | 0x0000_1111 |
| CLK_SRC_MASK_MAUDIO | 0xC33C | Clock source mask for AUDIO_BLK | 0x0000_0001 |
| CLK_SRC_MASK_FSYS | 0xC340 | Clock source mask for FSYS_BLK | 0x0101_1111 |
| RSVD | 0xC344 to 0xC34F | Reserved | Undefined |
| CLK_SRC_MASK_PERIL0 | 0xC350 | Clock source mask for PERIL_BLK | 0x0001_1111 |
| CLK_SRC_MASK_PERIL1 | 0xC354 | Clock source mask for PERIL_BLK | 0x0111_0111 |
| RSVD | 0xC358 to 0xC40F | Reserved | Undefined |
| CLK_MUX_STAT_TOP0 | 0xC410 | Clock MUX status for CMU_TOP | 0x1111_1111 |
| CLK_MUX_STAT_TOP1 | 0xC414 | Clock MUX status for CMU_TOP | 0x0111_1110 |
| RSVD | 0xC418 to 0xC427 | Reserved | Undefined |
| CLK_MUX_STAT_MFC | 0xC428 | Clock MUX status for MFC_BLK | 0x0000_0111 |
| CLK_MUX_STAT_G3D | 0xC42C | Clock MUX status for G3D_BLK | 0x0000_0111 |
| RSVD | 0xC430 to 0xC454 | Reserved | Undefined |
| CLK_MUX_STAT_CAM1 | 0xC458 | Clock MUX status for CAM_BLK | 0x0000_0111 |
| RSVD | 0xC45C to 0xC50C | Reserved | Undefined |
| CLK_DIV_TOP | 0xC510 | Sets clock divider ratio for CMU_TOP | 0x0000_0000 |
| RSVD | 0xC514 to 0xC51F | Reserved | Undefined |
| CLK_DIV_CAM0 | 0xC520 | Sets clock divider ratio for CAM_BLK | 0x0000_0000 |
| CLK_DIV_TV | 0xC524 | Sets clock divider ratio for TV_BLK | 0x0000_0000 |
| CLK_DIV_MFC | 0xC528 | Sets clock divider ratio for MFC_BLK | 0x0000_0000 |
| CLK_DIV_G3D | 0xC52C | Sets clock divider ratio for G3D_BLK | 0x0000_0000 |
| RSVD | 0xC530 | Reserved | Undefined |
| CLK_DIV_LCD | 0xC534 | Sets clock divider ratio for LCD_BLK | 0x0070_0000 |

| Register | Offset | Description | Reset Value |
|---------------------|------------------|--|-------------|
| CLK_DIV_ISP | 0xC538 | Sets clock divider ratio for ISP_BLK | 0x0000_0000 |
| CLK_DIV_AUDIO | 0xC53C | Sets clock divider ratio for AUDIO_BLK | 0x0000_0000 |
| CLK_DIV_FSYS0 | 0xC540 | Sets clock divider ratio for FSYS_BLK | 0x00B0_0000 |
| CLK_DIV_FSYS1 | 0xC544 | Sets clock divider ratio for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_FSYS2 | 0xC548 | Sets clock divider ratio for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_FSYS3 | 0xC54C | Sets clock divider ratio for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_PERIL0 | 0xC550 | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_PERIL1 | 0xC554 | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_PERIL2 | 0xC558 | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_PERIL3 | 0xC55C | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_PERIL4 | 0xC560 | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_PERIL5 | 0xC564 | Sets clock divider ratio for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_CAM1 | 0xC568 | Sets clock divider ratio for CAM_BLK | 0x0000_0000 |
| RSVD | 0xC56C to 0xC57C | Reserved | Undefined |
| CLKDIV2_RATIO | 0xC580 | Sets PCLK divider ratio in FSYS, CAM, LCD, TV, and GPS block | 0x0111_1111 |
| RSVD | 0xC584 to 0xC60F | Reserved | Undefined |
| CLK_DIV_STAT_TOP | 0xC610 | Clock divider status for CMU_TOP | 0x0000_0000 |
| RSVD | 0xC614 to 0xC61F | Reserved | Undefined |
| CLK_DIV_STAT_CAM0 | 0xC620 | Clock divider status for CAM_BLK | 0x0000_0000 |
| CLK_DIV_STAT_TV | 0xC624 | Clock divider status for TV_BLK | 0x0000_0000 |
| CLK_DIV_STAT_MFC | 0xC628 | Clock divider status for MFC_BLK | 0x0000_0000 |
| CLK_DIV_STAT_G3D | 0xC62C | Clock divider status for G3D_BLK | 0x0000_0000 |
| RSVD | 0xC630 | Reserved | Undefined |
| CLK_DIV_STAT_LCD | 0xC634 | Clock divider status for LCD_BLK | 0x0000_0000 |
| CLK_DIV_STAT_ISP | 0xC638 | Clock divider status for ISP_BLK | 0x0000_0000 |
| CLK_DIV_STAT_AUDIO | 0xC63C | Clock divider status for AUDIO_BLK | 0x0000_0000 |
| CLK_DIV_STAT_FSYS0 | 0xC640 | Clock divider status for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_STAT_FSYS1 | 0xC644 | Clock divider status for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_STAT_FSYS2 | 0xC648 | Clock divider status for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_STAT_FSYS3 | 0xC64C | Clock divider status for FSYS_BLK | 0x0000_0000 |
| CLK_DIV_STAT_PERIL0 | 0xC650 | Clock divider status for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_STAT_PERIL1 | 0xC654 | Clock divider status for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_STAT_PERIL2 | 0xC658 | Clock divider status for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_STAT_PERIL3 | 0xC65C | Clock divider status for PERIL_BLK | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-------------------------|------------------|--|-------------|
| CLK_DIV_STAT_PERIL4 | 0xC660 | Clock divider status for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_STAT_PERIL5 | 0xC664 | Clock divider status for PERIL_BLK | 0x0000_0000 |
| CLK_DIV_STAT_CAM1 | 0xC668 | Clock divider status for CAM_BLK | 0x0000_0000 |
| RSVD | 0xC66C to 0xC67C | Reserved | Undefined |
| CLKDIV2_STAT | 0xC680 | PCLK divider status for FSYS, CAM, LCD, and TV block | 0x0000_0000 |
| RSVD | 0xC684 to 0xC740 | Reserved | Undefined |
| CLK_GATE_BUS_FSYS1 | 0xC744 | Control gating of AXI/AHB/APB clock for FSYS_BLK | 0xFFFF_FFFF |
| RSVD | 0xC748 to 0xC91F | Reserved | Undefined |
| CLK_GATE_IP_CAM | 0xC920 | Controls IP clock gating for CAM_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_TV | 0xC924 | Controls IP clock gating for TV_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_MFC | 0xC928 | Controls IP clock gating for MFC_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_G3D | 0xC92C | Controls IP clock gating for G3D_BLK | 0xFFFF_FFFF |
| RSVD | 0xC930 | Reserved | Undefined |
| CLK_GATE_IP_LCD | 0xC934 | Controls IP clock gating for LCD_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_ISP | 0xC938 | Controls IP clock gating for ISP_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_FSYS | 0xC940 | Controls IP clock gating for FSYS_BLK | 0xFFFF_FFFF |
| RSVD | 0xC944 to 0xC948 | Reserved | Undefined |
| CLK_GATE_IP_GPS | 0xC94C | Controls IP clock gating for GPS_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_PERIL | 0xC950 | Controls IP clock gating for PERIL_BLK | 0xFFFF_FFFF |
| RSVD | 0xC954 to 0xC96F | Reserved | Undefined |
| CLK_GATE_BLOCK | 0xC970 | Clock gating control block | 0xFFFF_FFFF |
| RSVD | 0xC974 to 0xC9FF | Reserved | Undefined |
| CLKOUT_CMU_TOP | 0xCA00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_TOP_DIV_STAT | 0xCA04 | Clock divider status for CLKOUT | 0x0000_0000 |
| RSVD | 0xCA08 to 0x0004 | Reserved | Undefined |

- Base Address 0x10040000

| Register | Offset | Description | Reset Value |
|-------------------------|------------------|---|-------------|
| MPLL_LOCK | 0x0008 | Controls PLL locking period for MPLL | 0x0000_0FFF |
| RSVD | 0x000C to 0x0104 | Reserved | Undefined |
| MPLL_CON0 | 0x0108 | Controls PLL output frequency for MPLL | 0x0064_0300 |
| MPLL_CON1 | 0x010C | Controls PLL AFC | 0x0080_3800 |
| RSVD | 0x0110 to 0x01FC | Reserved | Undefined |
| CLK_SRC_DMC | 0x0200 | Selects clock source for CMU_DMC | 0x0001_0000 |
| RSVD | 0x0204 to 0x02FF | Reserved | Undefined |
| CLK_SRC_MASK_DMC | 0x0300 | Clock source mask for DMC_BLK | 0x0001_0000 |
| RSVD | 0x0304 to 0x03FF | Reserved | Undefined |
| CLK_MUX_STAT_DMC | 0x0400 | Clock MUX status for CMU_DMC | 0x1110_1111 |
| RSVD | 0x0404 to 0x04FF | Reserved | Undefined |
| CLK_DIV_DMC0 | 0x0500 | Sets clock divider ratio for CMU_DMC | 0x0000_0000 |
| CLK_DIV_DMC1 | 0x0504 | Sets clock divider ratio for CMU_DMC | 0x0000_1000 |
| RSVD | 0x0508 to 0x05FF | Reserved | Undefined |
| CLK_DIV_STAT_DMC0 | 0x0600 | Clock divider status for CMU_DMC | 0x0000_0000 |
| CLK_DIV_STAT_DMC1 | 0x0604 | Clock divider status for CMU_DMC | 0x0000_0000 |
| RSVD | 0x0608 to 0x06FC | Reserved | Undefined |
| CLK_GATE_BUS_DMC0 | 0x0700 | Control gating of AXI clock for DMC_BLK | 0xFFFF_FFFF |
| CLK_GATE_BUS_DMC1 | 0x0704 | Control gating of APB clock for DMC_BLK | 0xFFFF_FFFF |
| RSVD | 0x0708 to 0x08FC | Reserved | Undefined |
| CLK_GATE_IP_DMC0 | 0x0900 | Control IP clock gating for DMC_BLK | 0xFFFF_FFFF |
| CLK_GATE_IP_DMC1 | 0x0904 | Control IP clock gating for DMC_BLK | 0xFFFF_FFFF |
| RSVD | 0x0908 to 0x09FF | Reserved | Undefined |
| CLKOUT_CMU_DMC | 0x0A00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_DMC_DIV_STAT | 0x0A04 | Clock divider status for CLKOUT | 0x0000_0000 |
| RSVD | 0x0A08 to 0x0FFF | Reserved | Undefined |
| DCGIDX_MAP0 | 0x1000 | DCG index map0 | 0xFFFF_FFFF |
| DCGIDX_MAP1 | 0x1004 | DCG index map1 | 0xFFFF_FFFF |

| Register | Offset | Description | Reset Value |
|------------------|------------------|--|-------------|
| DCGIDX_MAP2 | 0x1008 | DCG index map2 | 0xFFFF_FFFF |
| RSVD | 0x100C to 0x101F | Reserved | Undefined |
| DCGPERF_MAP0 | 0x1020 | DCG performance map0 | 0xFFFF_FFFF |
| DCGPERF_MAP1 | 0x1024 | DCG performance map1 | 0xFFFF_FFFF |
| RSVD | 0x1028 to 0x103F | Reserved | Undefined |
| DVCIDX_MAP | 0x1040 | DVC index map | 0x00FF_FFFF |
| RSVD | 0x1044 to 0x105F | Reserved | Undefined |
| FREQ_CPU | 0x1060 | Maximum frequency of CPU | 0x0000_0000 |
| FREQ_DPM | 0x1064 | Frequency of DPM | 0x0000_0000 |
| RSVD | 0x1068 to 0x107F | Reserved | Undefined |
| DVSEMCLOCK_EN | 0x1080 | DVS emulation clock enable | 0x0000_0000 |
| MAXPERF | 0x1084 | Maximum performance enable | 0x0000_0000 |
| RSVD | 0x1088 to 0x1090 | Reserved | Undefined |
| DMC_PAUSE_CTRL | 0x1094 | Pause function of DREX2 for DVFS | 0x0000_0000 |
| DDRPHY_LOCK_CTRL | 0x1098 | DDRPHY DLL lock control register when C2C is enabled | 0x0000_0000 |
| C2C_STATE | 0x109C | Current state of C2C SEC FSM | 0x0000_0000 |
| RSVD | 0x10A0 to 0x3FFC | Reserved | Undefined |
| APLL_LOCK | 0x4000 | Control PLL locking period for APPLL | 0x0000_0FFF |
| RSVD | 0x4004 to 0x40FC | Reserved | Undefined |
| APLL_CON0 | 0x4100 | Control PLL output frequency for APPLL | 0x0064_0300 |
| APLL_CON1 | 0x4104 | Control PLL AFC | 0x0080_3800 |
| RSVD | 0x4108 to 0x41FC | Reserved | Undefined |
| CLK_SRC_CPU | 0x4200 | Selects clock source for CMU_CPU | 0x0000_0000 |
| RSVD | 0x4204 to 0x43FF | Reserved | Undefined |
| CLK_MUX_STAT_CPU | 0x4400 | Clock MUX status for CMU_CPU | 0x0011_0101 |
| RSVD | 0x4404 to 0x44FF | Reserved | Undefined |
| CLK_DIV_CPU0 | 0x4500 | Sets clock divider ratio for CMU_CPU | 0x0000_0000 |
| CLK_DIV_CPU1 | 0x4504 | Sets clock divider ratio for CMU_CPU | 0x0000_0000 |
| RSVD | 0x4508 to | Reserved | Undefined |

| Register | Offset | Description | Reset Value |
|-------------------------|------------------|---|-------------|
| | 0x45FF | | |
| CLK_DIV_STAT_CPU0 | 0x4600 | Clock divider status for CMU_CPU | 0x0000_0000 |
| CLK_DIV_STAT_CPU1 | 0x4604 | Clock divider status for CMU_CPU | 0x0000_0000 |
| RSVD | 0x4608 to 0x48FF | Reserved | Undefined |
| CLK_GATE_IP_CPU | 0x4900 | Controls IP clock gating for CMU_CPU | 0xFFFF_FFFF |
| RSVD | 0x4904 to 0x49FF | Reserved | Undefined |
| CLKOUT_CMU_CPU | 0x4A00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_CPU_DIV_STAT | 0x4A04 | Clock divider status for CLKOUT | 0x0000_0000 |
| RSVD | 0x4A08 to 0x4FFF | Reserved | Undefined |
| ARMCLK_STOPCTRL | 0x5000 | ARM clock stop control register SCLK_APPL counts the number of cycles. | 0x0404_0404 |
| ATCLK_STOPCTRL | 0x5004 | ATCLK stop control register SCLK_APPL counts the number of cycles. | 0x0000_0404 |
| RSVD | 0x500C to 0x501C | Reserved | Undefined |
| PWR_CTRL | 0x5020 | Power control register | 0x0000_04FF |
| PWR_CTRL2 | 0x5024 | Power control register | 0x0000_0000 |
| RSVD | 0x5028 to 0x53FC | Reserved | Undefined |
| L2_STATUS | 0x5400 | L2 cache status register | 0x0000_0000 |
| RSVD | 0x5404 to 0x540C | Reserved | Undefined |
| CPU_STATUS | 0x5410 | Cortex-A9 processor status register | 0x0000_0000 |
| RSVD | 0x5414 to 0x541C | Reserved | Undefined |
| PTM_STATUS | 0x5420 | Program trace macrocell (PTM) status register | 0x0000_0000 |
| CLK_DIV_ISP0 | 0x8300 | Set clock divider ratio for CMU_ISP0 | 0x0000_0000 |
| CLK_DIV_ISP1 | 0x8304 | Set clock divider ratio for MPWM in CMU_ISP1 | 0x0000_0000 |
| CLK_DIV_STAT_ISP0 | 0x8400 | Clock divider status for CMU_ISP0 | 0x0000_0000 |
| CLK_DIV_STAT_ISP1 | 0x8404 | Clock divider status for MPWM in CMU_ISP1 | 0x0000_0000 |
| RSVD | 0x8408 to 0x87FC | Reserved | Undefined |
| CLK_GATE_IP_ISP0 | 0x8800 | Control IP clock gating for ISP_BLK register0 | 0xFFFF_FFFF |
| CLK_GATE_IP_ISP1 | 0x8804 | Control IP clock gating for ISP_BLK register1 | 0xFFFF_FFFF |

| Register | Offset | Description | Reset Value |
|-------------------------|------------------|---------------------------------|-------------|
| RSVD | 0x8808 to 0x89FC | Reserved | Undefined |
| CLKOUT_CMU_ISP | 0x8A00 | CLKOUT control register | 0x0001_0000 |
| CLKOUT_CMU_ISP_DIV_STAT | 0x8A04 | Clock divider status for CLKOUT | 0x0000_0000 |
| CMU_ISP_SPARE0 | 0x8B00 | CMU_ISP spare register0 | 0x0000_0000 |
| CMU_ISP_SPARE1 | 0x8B04 | CMU_ISP spare register1 | 0x0000_0000 |
| CMU_ISP_SPARE2 | 0x8B08 | CMU_ISP spare register2 | 0x0000_0000 |
| CMU_ISP_SPARE3 | 0x8B0C | CMU_ISP spare register3 | 0x0000_0000 |

The six address spaces that SFRs fall into are: SFRs with address 0x0_4000 to 0x0_7FFF-These special function registers control clock-related logics for LEFTBUS block. They control clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x0_8000 to 0x0_BFFF-These special function registers control clock-related logics for RIGHTBUS block. They control clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x0_C000 to 0x0_FFFF-These special function registers control clock-related logics for MFC, G3D, TV, LCD, ISP, CAM, FSYS, PERIL, and PERIR blocks. They control EPLL and VPLL, clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x1_0000 to 0x1_3FFF-These special function registers control clock-related logics for DMC block. They control MPLL, clock source selection, clock divider ratio, and clock gating.

SFRs with address 0x1_4000 to 0x1_7FFF-These special function registers control clock-related logics for CPU block. They control APLL, clock source selection, clock divider ratio and CPU-related logics.

SFRs with address 0x1_8000 to 0x1_BFFF-These special function registers control clock-related logics for ISP block. They control clock source selection, clock divider ratio, and clock gating.

5.9.1.1 CLK_SRC_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4200, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------|--------|------|--|-------------|
| RSVD | [31:5] | - | Reserved | 0x0 |
| MUX_MPLL_USER_SEL_L | [4] | RW | Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| MUX_GDL_SEL | [0] | RW | Controls MUXGDL 0 = SCLKMPPLL 1 = SCLKAPLL | 0x0 |

5.9.1.2 CLK_MUX_STAT_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4400, Reset Value = 0x0000_0011

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|---|-------------|
| RSVD | [31:7] | - | Reserved | 0x0 |
| MPLL_USER_SEL_L | [6:4] | R | Selection Signal Status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing. | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| GDL_SEL | [2:0] | R | Selection Signal Status of MUXGDL 001 = SCLKMPPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.3 CLK_DIV_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4500, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:7] | – | Reserved | 0x0 |
| GPL_RATIO | [6:4] | RW | DIVGPL Clock Divider Ratio ACLK_GPL = MOUTGPL/(GPL_RATIO + 1) | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| GDL_RATIO | [2:0] | RW | DIVGDL Clock Divider Ratio ACLK_GDL = MOUTGDL/(GDL_RATIO + 1) | 0x0 |

5.9.1.4 CLK_DIV_STAT_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4600, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| RSVD | [31:5] | – | Reserved | 0x0 |
| DIV_GPL | [4] | R | DIVGPL Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | – | Reserved | 0x0 |
| DIV_GDL | [0] | R | DIVGDL Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.5 CLK_GATE_IP_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4800, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|----------------|--------|------|--|-------------|
| RSVD | [31:7] | - | Reserved | 0x1FFFFFFF |
| CLK_ASYNC_G3D | [6] | RW | Gating all clocks for ASYNC_G3D 0 = Mask 1 = Pass | 0x1 |
| RSVD | [5] | - | Reserved | 0x1 |
| CLK_ASYNC_MFCL | [4] | RW | Gating all clocks for ASYNC_MFCL 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_TVX | [3] | RW | Gating all clocks for ASYNC_TVX 0 = Mask 1 = Pass | 0x1 |
| RSVD | [2] | - | Reserved | 0x1 |
| CLK_PPMULEFT | [1] | RW | Gating all clocks for PPMULEFT 0 = Mask 1 = Pass | 0x1 |
| CLK_GPIO_LEFT | [0] | RW | Gating all clocks for GPIO_LEFT 0 = Mask 1 = Pass | 0x1 |

5.9.1.6 CLK_GATE_IP_IMAGE

- Base Address: 0x1003_0000
- Address = Base Address + 0x4930, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:10] | – | Reserved | 0x3FFFFF |
| CLK_PPMUIMAGE | [9] | RW | Gating all clocks for PPMUIMAGE 0 = Mask 1 = Pass | 0x1 |
| RSVD | [8] | – | Reserved | 0x1 |
| RSVD | [7] | – | Reserved | 0x1 |
| RSVD | [6] | – | Reserved | 0x1 |
| CLK_SMMUUDMA | [5] | RW | Gating all clocks for SMMUUDMA 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUROTATOR | [4] | RW | Gating all clocks for SMMUROTATOR 0 = Mask 1 = Pass | 0x1 |
| RSVD | [3] | – | Reserved | 0x1 |
| CLK_MDMA | [2] | RW | Gating all clocks for MDMA 0 = Mask 1 = Pass | 0x1 |
| CLK_ROTATOR | [1] | RW | Gating all clocks for ROTATOR 0 = Mask 1 = Pass | 0x1 |
| RSVD | [0] | – | Reserved | 0x1 |

5.9.1.7 CLKOUT_CMU_LEFTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x4A00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX selection 00000 = SCLK_MPLL/2 00001 = SCLK_APLL/2 00010 = ACLK_GDL 00011 = ACLK_GPL | 0x0 |

5.9.1.8 CLKOUT_CMU_LEFTBUS_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0x4A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.9 CLK_SRC_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8200, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------|--------|------|---|-------------|
| RSVD | [31:5] | - | Reserved | 0x0 |
| MUX_MPLL_USER_SEL_R | [4] | RW | Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| MUX_GDR_SEL | [0] | RW | Controls MUXGDR 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |

5.9.1.10 CLK_MUX_STAT_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8400, Reset Value = 0x0000_0011

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|--|-------------|
| RSVD | [31:7] | - | Reserved | 0x0 |
| MPLL_USER_SEL_R | [6:4] | R | Selection Signal Status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | - | Reserved | 0x0 |
| GDR_SEL | [2:0] | R | Selection Signal Status of MUXGDR 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.11 CLK_DIV_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8500, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:7] | – | Reserved | 0x0 |
| GPR_RATIO | [6:4] | RW | DIVGPR Clock Divider Ratio ACLK_GPR = MOUTGPR/(GPR_RATIO + 1) | 0x0 |
| RSVD | [3] | – | Reserved | 0x0 |
| GDR_RATIO | [2:0] | RW | DIVGDR Clock Divider Ratio ACLK_GDR = MOUTGDR/(GDR_RATIO + 1) | 0x0 |

5.9.1.12 CLK_DIV_STAT_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8600, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| RSVD | [31:5] | – | Reserved | 0x0 |
| DIV_GPR | [4] | R | DIVGPR Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | – | Reserved | 0x0 |
| DIV_GDR | [0] | R | DIVGDR Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.13 CLK_GATE_IP_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8800, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------------|---------|------|---|-------------|
| RSVD | [31:10] | - | Reserved | 0x3FFFFF |
| CLK_ASYNC_ISPMX | [9] | RW | Gating all clocks for ASYNC_ISPMX 0 = Mask 1 = Pass | 0x1 |
| RSVD | [8] | - | Reserved | 0x1 |
| CLK_ASYNC_MAUDIOX | [7] | RW | Gating all clocks for ASYNC_MAUDIOX 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_MFCR | [6] | RW | Gating all clocks for ASYNC_MFCR 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_FSYSD | [5] | RW | Gating all clocks for ASYNC_FSYSD 0 = Mask 1 = Pass | 0x1 |
| RSVD | [4] | - | Reserved | 0x1 |
| CLK_ASYNC_LCD0X | [3] | RW | Gating all clocks for ASYNC_LCD0X 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_CAMX | [2] | RW | Gating all clocks for ASYNC_CAMX 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMURIGHT | [1] | RW | Gating all clocks for PPMURIGHT 0 = Mask 1 = Pass | 0x1 |
| CLK_GPIO_RIGHT | [0] | RW | Gating all clocks for GPIO_RIGHT 0 = Mask 1 = Pass | 0x1 |

5.9.1.14 CLK_GATE_IP_PERIR

- Base Address: 0x1003_0000
- Address = Base Address + 0x8960, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--|-------------|
| RSVD | [31:19] | - | Reserved | 0x1FFF |
| CLK_CMU_ISPPART | [18] | RW | Gating all clocks for CMU_ISPPART 0 = Mask 1 = Pass | 0x1 |
| CLK_TMU_APBIF | [17] | RW | Gating all clocks for TMU_APBIF 0 = Mask 1 = Pass | 0x1 |
| CLK_KEYIF | [16] | RW | Gating all clocks for KEYIF 0 = Mask 1 = Pass | 0x1 |
| CLK_RTC | [15] | RW | Gating all clocks for RTC 0 = Mask 1 = Pass | 0x1 |
| CLK_WDT | [14] | RW | Gating all clocks for WDT 0 = Mask 1 = Pass | 0x1 |
| CLK_MCT | [13] | RW | Gating all clocks for System Timer 0 = Mask 1 = Pass | 0x1 |
| CLK_SECKEY | [12] | RW | Gating all clocks for SECKEY 0 = Mask 1 = Pass | 0x1 |
| CLK_HDMI_CEC | [11] | RW | Gating all clocks for HDMI_CEC 0 = Mask 1 = Pass | 0x1 |
| CLK_TZPC5 | [10] | RW | Gating all clocks for TZPC5 0 = Mask 1 = Pass | 0x1 |
| CLK_TZPC4 | [9] | RW | Gating all clocks for TZPC4 0 = Mask 1 = Pass | 0x1 |
| CLK_TZPC3 | [8] | RW | Gating all clocks for TZPC3 0 = Mask 1 = Pass | 0x1 |
| CLK_TZPC2 | [7] | RW | Gating all clocks for TZPC2 0 = Mask 1 = Pass | 0x1 |
| CLK_TZPC1 | [6] | RW | Gating all clocks for TZPC1 0 = Mask | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------------------|-----|------|--|-------------|
| | | | 1 = Pass 0 = Mask | |
| CLK_TZPC0 | [5] | RW | Gating all clocks for TZPC0 0 = Mask 1 = Pass | 0x1 |
| CLK_CMU_COREPART | [4] | RW | Gating all clocks for CMU_COREPART 0 = Mask 1 = Pass | 0x1 |
| CLK_CMU_TOPPART | [3] | RW | Gating all clocks for CMU_TOPPART 0 = Mask 1 = Pass | 0x1 |
| CLK_PMU_APBIF | [2] | RW | Gating all clocks for PMU_APBIF 0 = Mask 1 = Pass | 0x1 |
| CLK_SYSREG | [1] | RW | Gating all clocks for SYSREG 0 = Mask 1 = Pass | 0x1 |
| CLK_CHIP_ID | [0] | RW | Gating all clocks for CHIP ID 0 = Mask 1 = Pass | 0x1 |

5.9.1.15 CLKOUT_CMU_RIGHTBUS

- Base Address: 0x1003_0000
- Address = Base Address + 0x8A00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX Selection 00000 = SCLK_MPLL/2 00001 = SCLK_APLL/2 00010 = ACLK_GDR 00011 = ACLK_GPR | 0x0 |

5.9.1.16 CLKOUT_CMU_RIGHTBUS_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0x8A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.17 EPLL_LOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC010, Reset Value = 0x0000_0FFF

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| PLL_LOCKTIME | [15:0] | RW | Required period to generate a stable clock output Set (3000 cycles \times PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF = 1/(FIN/PDIV) The maximum PLL lock time is 250 usec where FIN is 24 MHz, PDIV is 2 and PLL_LOCKTIME is 6000. | 0xFFFF |

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking \geq the maximum locktime)

5.9.1.18 VPLL_LOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC020, Reset Value = 0x0000_0FFF

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| PLL_LOCKTIME | [15:0] | RW | Required period to generate a stable clock output Set (3000 cycles × PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF = 1/(FIN/PDIV) The maximum PLL lock time is 250 usec where FIN is 24 MHz, PDIV is 2 and PLL_LOCKTIME is 6000. | 0xFFFF |

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking \geq the maximum locktime)

5.9.1.19 EPLL_CON0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC110, Reset Value = 0x0060_0302

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|--|-------------|
| ENABLE | [31] | RW | PLL Enable Control 0 = Disables 1 = Enables | 0x0 |
| RSVD | [30] | - | Reserved | 0x0 |
| LOCKED | [29] | R | PLL Locking Indication 0 = Unlocks 1 = Locks This field is set after the locking time. EPLL_LOCK SFR register sets the locking time. It is a Read Only register. | 0x0 |
| RSVD | [28:25] | - | Reserved | 0x0 |
| MDIV | [24:16] | RW | PLL M Divide Value | 0x60 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| PDIV | [13:8] | RW | PLL P Divide Value | 0x3 |
| RSVD | [7:3] | - | Reserved | 0x0 |
| SDIV | [2:0] | RW | PLL S Divide Value | 0x2 |

The reset value of EPLL_CON0 generates a 192 MHz output clock for the input clock frequency of 24 MHz.

The equation to calculate the output frequency is: $F_{OUT} = (MDIV + K/65536) \times F_{IN}/(PDIV \times 2SDIV)$

The conditions MDIV, PDIV, SDIV, and K should meet are:

- PDIV: $1 \leq PDIV \leq 63$
- MDIV: $16 \leq MDIV \leq 511$
- SDIV: $0 \leq SDIV \leq 5$
- K: $0 \leq K \leq 65535$
- $F_{ref} = F_{IN}/PDIV$, where $4 \text{ MHz} \leq F_{ref} \leq 30 \text{ MHz}$
- $F_{VCO} = (MDIV + K/65536) \times FIN/PDIV$
- F_{OUT} should fall in the range of: $22 \text{ MHz} \leq F_{OUT} \leq 1400 \text{ MHz}$
Do not set the value PDIV or MDIV to all zeros.

Refer to the section [5.3.2 Recommended PLL PMS Value for EPLL](#) for recommended PMS values.

5.9.1.20 EPLL_CON1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC114, Reset Value = 0x6601_0000

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| SEL_PF | [30:29] | RW | Modulation Method Control 00 = Down spread 01 = Up spread 1x = Center spread | 0x3 |
| MRR | [28:24] | RW | Modulation Rate Control | 0x6 |
| MFR | [23:16] | RW | Modulation Frequency Control | 0x1 |
| K | [15:0] | RW | PLL 16-bit DSM (Delta-Sigma Modulator) | 0x0 |

Refer to the section [5.3.2 Recommended PLL PMS Value for EPLL](#) for the recommended value of K.

5.9.1.21 EPLL_CON2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC118, Reset Value = 0x0000_0080

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:13] | – | Reserved | 0x0 |
| EXTAFC | [12:8] | RW | AFC value | 0x0 |
| DCC_ENB | [7] | RW | Decides whether Duty Cycle Corrector (DCC) is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal. | 0x1 |
| AFC_ENB | [6] | RW | Decides whether Adaptive Frequency Calibrator (AFC) is enabled or not. When AFC is enabled, it calibrates VCO automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal. | 0x0 |
| SSCG_EN | [5] | RW | Specifies if the dithered mode is enabled or not. 0 = Disables dithered mode 1 = Enables dithered mode | 0x0 |
| BYPASS | [4] | RW | If BYPASS = 1, then it enables bypass mode ($F_{OUT} = F_{IN}$) If BYPASS = 0, then PLL3600X operates normally. | 0x0 |
| FVCO_EN | [3] | RW | Enable pin for F_{VCO_OUT} | 0x0 |
| FSEL | [2] | RW | Pin selection for monitoring purposes. $F_{VCO_OUT} = F_{REF}$, if FSEL is set to 0 $F_{VCO_OUT} = F_{EED}$, FSEL is set to 1 | 0x0 |
| ICP_BOOST | [1:0] | RW | ICP_BOOST | 0x0 |

If AFC_ENB is set to logic LOW, then it enables the AFC. If AFC_ENB is set to logic HIGH, then EXT AFC [4:0] controls the VCO frequency tuning range.

EXTAFC specifies the decimal value of EXT AFC[4:0] as:

- EXT AFC = EXT AFC[4:0]

The hexadecimal values specified for EXT AFC[4:0] registers are:

- 5'b0 0000 ≤ EXT AFC[4:0] ≤ 5'b1 1111

NOTE: The other PLL control inputs should be set as:

DCC_ENB = 1 ICP_BOOST = 0

SSCG_EN = 0 (Disable dithered mode)

AFC_ENB = 0 EXT AFC = 0

5.9.1.22 VPLL_CON0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC120, Reset Value = 0x006F_0302

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|--|-------------|
| ENABLE | [31] | RW | PLL Enable Control 0 = Disables 1 = Enables | 0x0 |
| RSVD | [30] | - | Reserved | 0x0 |
| LOCKED | [29] | R | PLL Locking Indication 0 = Unlocks 1 = Locks | 0x0 |
| RSVD | [28:25] | - | Reserved | 0x0 |
| MDIV | [24:16] | RW | PLL M Divide Value | 0x6F |
| RSVD | [15:14] | - | Reserved | 0x0 |
| PDIV | [13:8] | RW | PLL P Divide Value | 0x3 |
| RSVD | [7:3] | - | Reserved | 0x0 |
| SDIV | [2:0] | RW | PLL S Divide Value | 0x2 |

The reset value of VPLL_CON0 generates a 222.75 MHz output clock for an input clock frequency of 24 MHz. Equation to calculate the output frequency is: $F_{OUT} = (MDIV + K/65535) \times F_{IN}/(PDIV \times 2SDIV)$

Where, MDIV, PDIV, SDIV, and K should meet the following conditions:

- PDIV: $1 \leq PDIV \leq 63$
- MDIV: $16 \leq MDIV \leq 511$
- SDIV: $0 \leq SDIV \leq 5$
- K: $0 \leq K \leq 65535$
- $F_{ref} = F_{IN}/PDIV$ Fref should fall in the range of: $4 \text{ MHz} \leq F_{ref} \leq 30 \text{ MHz}$
- $F_{VCO} = (MDIV + K/v) \times F_{IN}/PDIV$
- $F_{OUT}: 22 \text{ MHz} \leq F_{OUT} \leq 1400 \text{ MHz}$

Do not set the value PDIV or MDIV to all zeros.

Refer to the section [5.3.3 Recommended PLL PMS Value for VPLL](#) for the recommended PMS values.

5.9.1.23 VPLL_CON1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC124, Reset Value = 0x6601_6000

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| SEL_PF | [30:29] | RW | Modulation Method Control 00 = Down spread 01 = Up spread 1x = Center spread | 0x3 |
| MRR | [28:24] | RW | Modulation Rate Control | 0x6 |
| MFR | [23:16] | RW | Modulation Frequency Control | 0x1 |
| K | [15:0] | RW | PLL DSM | 0x464 |

The equation to calculate the Modulation Frequency (MF) is: $MF = FFIN/PDIV/MFR/32[\text{Hz}]$

The equation to calculate the Modulation Rate (MR) is:

- $MR = MFR \times MRR/MDIV/64 \times 100[\%]$

The conditions that MFR and MRR should meet are:

- MFR should fall in the range of: $0 \leq MFR \leq 255$
- MRR should fall in the range of: $1 \leq MRR \leq 31$
- $0 \leq MRR \times MFR \leq 512$
- SEL_PF[1:0]: $2'b00 \leq SEL_PF \leq 2'b10$

5.9.1.24 VPLL_CON2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC128, Reset Value = 0x0000_0080

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| RSVD | [31:13] | – | Reserved | 0x0 |
| EXTAFC | [12:8] | RW | AFC value | 0x0 |
| DCC_ENB | [7] | RW | Decides whether DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal. | 0x1 |
| AFC_ENB | [6] | RW | Decides whether AFC is enabled or not. When enabled, VCO is calibrated automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal. | 0x0 |
| SSCG_EN | [5] | RW | Specifies if the dithered mode is enabled or not. 0 = Disables dithered mode 1 = Enables dithered mode | 0x0 |
| BYPASS | [4] | RW | If BYPASS = 1, then it enables bypass mode ($F_{OUT} = F_{IN}$) IF BYPASS = 0, then the PLL3600X operates normally. | 0x0 |
| FVCO_EN | [3] | RW | Enable pin for F_{VCO_OUT} | 0x0 |
| FSEL | [2] | RW | Specifies pin selection for monitoring purposes $F_{VCO_OUT} = F_{REF}$, if F_{SEL} is set to 0 $F_{VCO_OUT} = F_{EED}$, if F_{SEL} is set to 1 | 0x0 |
| ICP_BOOST | [1:0] | RW | ICP_BOOST | 0x0 |

If AFC_ENB is set to logic LOW, then it enables the AFC. If AFC_ENB is set to logic HIGH, then EXTAFC [4:0] controls the VCO frequency tuning range.

EXTAFC specifies the decimal value of EXTAFC[4:0] as:

- EXTAFC = EXTAFC[4:0]

The hexadecimal values specified for EXTAFC[4:0] registers are:

- 5'b0 0000 ≤ EXTAFC[4:0] ≤ 5'b1 1111

NOTE: The other PLL control inputs should be set as:

DCC_ENB = 1
ICP_BOOST = 0
AFC_ENB = 0
EXTAFC = 0

5.9.1.25 CLK_SRC_TOP0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC210, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|---------|------|--|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| MUX_ONENAND_SEL | [28] | RW | Controls MUXONENAND 0 = ACLK_133 1 = ACLK_160 | 0x0 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| MUX_ACLK_133_SEL | [24] | RW | Controls MUXACLK_133 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| MUX_ACLK_160_SEL | [20] | RW | Controls MUXACLK_160 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| MUX_ACLK_100_SEL | [16] | RW | Controls MUXACLK_100 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| MUX_ACLK_200_SEL | [12] | RW | Controls MUXACLK_200 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| MUX_VPLL_SEL | [8] | RW | Controls MUXVPLL 0 = FINPLL 1 = FOUTVPLL | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_EPLL_SEL | [4] | RW | Controls MUXEPLL 0 = FINPLL 1 = FOUTEPLL | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| MUX_ONENAND_1_SEL | [0] | RW | Controls MUXONENAND_1 0 = MOUTONENAND 1 = SCLKVPLL | 0x0 |

5.9.1.26 CLK_SRC_TOP1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC214, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------------------|---------|------|---|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| MUX_ACLK_400_MCUISP_SUB_SEL | [24] | RWX | Controls MUXACLK_400_MCUISP_SUB 0 = FINPLL 1 = DIVOUT_ACLK_400_MCUISP | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| MUX_ACLK_200_SUB_SEL | [20] | RWX | Controls MUXACLK_200_SUB 0 = FINPLL 1 = DIVOUT_ACLK_200 | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| MUX_ACLK_266_GPS_SUB_SEL | [16] | RWX | Controls MUXACLK_266_GPS_SUB 0 = FINPLL 1 = DIVOUT_ACLK_266_GPS | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| MUX_MPLL_USER_SEL_T | [12] | RW | Controls MUXMPLL 0 = FINPLL 1 = SCLKMPLL | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| MUX_ACLK_400_MCUISP_SEL | [8] | RW | Controls MUXACLK_400_MCUISP 0 = SCLKMPLL_USER_T 1 = SCLKAPLL | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_ACLK_266_GPS_SEL | [4] | RW | Controls MUXACLK_266_GPS 0 = SCLKMPLL_USER_T 1 = SCLKAPLL | 0x0 |
| RSVD | [3:0] | - | Reserved | 0x0 |

5.9.1.27 CLK_SRC_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC220, Reset Value = 0x1111_1111

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| CSIS1_SEL | [31:28] | RW | Controls MUXCSIS1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCSIS1 is the source clock of CSIS1. | 0x1 |
| CSIS0_SEL | [27:24] | RW | Controls MUXCSIS0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCSIS0 is the source clock of CSIS0. | 0x1 |
| CAM1_SEL | [23:20] | RW | Controls MUXCAM1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXCAM1 is the source clock of CAM_B_CLKOUT. | 0x1 |
| CAM0_SEL | [19:16] | RW | Controls MUXCAM0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| | | | Others = Reserved MUXCAM0 is the source clock of CAM_A_CLKOUT. | |
| FIMC3_LCLK_SEL | [15:12] | RW | Controls MUXFIMC3_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC3_LCLK is the source clock of FIMC3 local clock. | 0x1 |
| FIMC2_LCLK_SEL | [11:8] | RW | Controls MUXFIMC2_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC2_LCLK is the source clock of FIMC2 local clock. | 0x1 |
| FIMC1_LCLK_SEL | [7:4] | RW | Controls MUXFIMC1_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXFIMC1_LCLK is the source clock of FIMC1 local clock. | 0x1 |
| FIMC0_LCLK_SEL | [3:0] | RW | Controls MUXFIMC0_LCLK 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|---|-------------|
| | | | MUXFIMC0_LCLK is the source clock of FIMC0 local clock. | |

5.9.1.28 CLK_SRC_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC224, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| HDMI_SEL | [0] | RW | Controls MUXHDMI 0 = SCLK_PIXEL 1 = SCLK_HDMIPHY MUXHDMI is the source clock of HDMI link. | 0x0 |

5.9.1.29 CLK_SRC_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC228, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| MFC_SEL | [8] | RW | Controls MUXMFC 0 = MOUTMFC_0 1 = MOUTMFC_1 MUXMFC is the source clock of MFC core. | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MFC_1_SEL | [4] | RW | Controls MUXMFC_1 0 = SCLKEPLL 1 = SCLKVPLL MUXMFC_1 is the source clock of MFC core. | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| MFC_0_SEL | [0] | RW | Controls MUXMFC_0 0 = SCLKMPPLL 1 = SCLKAPPLL MUXMFC_0 is the source clock of MFC core. | 0x0 |

5.9.1.30 CLK_SRC_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC22C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| G3D_SEL | [8] | RW | Controls MUXG3D 0 = MOUTG3D_0 1 = MOUTG3D_1 MUXG3D is the source clock of G3D core. | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| G3D_1_SEL | [4] | RW | Controls MUXG3D_1 0 = SCLKEPLL 1 = SCLKVPLL MUXG3D_1 is the source clock of G3D core. | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| G3D_0_SEL | [0] | RW | Controls MUXG3D_0 0 = SCLKMPLL 1 = SCLKAPLL MUXG3D_0 is the source clock of G3D core. | 0x0 |

5.9.1.31 CLK_SRC_LCD0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC234, Reset Value = 0x0000_1111

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| MIPIO_SEL | [15:12] | RW | Controls MUXMIPIO 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHYS 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMIPIO is the source clock of MIPI_DSIM0. | 0x1 |
| MDNIE_PWM0_SEL | [11:8] | RW | Controls MUXMDNIE_PWM0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHYS 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMDNIE_PWM0 is the source clock of MDNIE_PWM0. | 0x1 |
| MDNIE0_SEL | [7:4] | RW | Controls MUXMDNIE0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHYS 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMDNIE0 is the source clock of MDNIE0. | 0x1 |
| FIMD0_SEL | [3:0] | RW | Controls MUXFIMD0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHYS 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|--|-------------|
| | | | 1000 = SCLKVPLL Others = Reserved MUXFIMD0 is the source clock of FIMD0. | |

5.9.1.32 CLK_SRC_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC238, Reset Value = 0x0000_1111

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| UART_ISP_SEL | [15:12] | RW | Controls MUXUART_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART_ISP is the source clock of MIPI_DSIM1. | 0x1 |
| SPI1_ISP_SEL | [11:8] | RW | Controls MUXSPI1_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI1_ISP is the source clock of SPI1_ISP. | 0x1 |
| SPI0_ISP_SEL | [7:4] | RW | Controls MUXSPI0_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI0_ISP is the source clock of SPI0_ISP. | 0x1 |
| PWM_ISP_SEL | [3:0] | RW | Controls MUXPWM_ISP 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|---|-------------|
| | | | Others = Reserved MUXPWM_ISP is the source clock of PWM_ISP. | |

5.9.1.33 CLK_SRC_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC23C, Reset Value = 0x0000_0005

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:4] | - | Reserved | 0x0 |
| AUDIO0_SEL | [3:0] | RW | <p>Controls MUXAUDIO0 0000 = AUDIOCDCLK0 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO0 is the source clock of AUDIO0.</p> | 0x5 |

5.9.1.34 CLK_SRC_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC240, Reset Value = 0x0001_1111

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| MIPIHSI_SEL | [24] | RW | Control MUXMIPHSI, which is the source clock of MIPIHSI 0 = SCLKMPPLL_USER_T 1 = SCLKAPLL | 0x0 |
| RSVD | [23:20] | - | Reserved | 0x0 |
| MMC4_SEL | [19:16] | RW | Controls MUXMMC4 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC4 is the source clock of MMC4. | 0x1 |
| MMC3_SEL | [15:12] | RW | Controls MUXMMC3 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC3 is the source clock of MMC3. | 0x1 |
| MMC2_SEL | [11:8] | RW | Controls MUXMMC2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC2 is the source clock of MMC2. | 0x1 |
| MMC1_SEL | [7:4] | RW | Controls MUXMMC1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|----------|-------|------|--|-------------|
| | | | 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC1 is the source clock of MMC1. | |
| MMC0_SEL | [3:0] | RW | Controls MUXMMC0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXMMC0 is the source clock of MMC0. | 0x1 |

5.9.1.35 CLK_SRC_PERILO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC250, Reset Value = 0x0001_1111

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| RSVD | [31:20] | - | Reserved It should be 1'b1. | 0x0 |
| UART4_SEL | [19:16] | RW | Controls MUXUART4 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART4 is the source clock of UART4. | 0x1 |
| UART3_SEL | [15:12] | RW | Controls MUXUART3 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART3 is the source clock of UART3. | 0x1 |
| UART2_SEL | [11:8] | RW | Controls MUXUART2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART2 is the source clock of UART2. | 0x1 |
| UART1_SEL | [7:4] | RW | Controls MUXUART1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|---|-------------|
| | | | 1000 = SCLKVPLL Others = Reserved MUXUART1 is the source clock of UART1. | |
| UART0_SEL | [3:0] | RW | Controls MUXUART0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXUART0 is the source clock of UART0. | 0x1 |

5.9.1.36 CLK_SRC_PERI1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC254, Reset Value = 0x0111_0055

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:28] | - | Reserved | 0x0 |
| SPI2_SEL | [27:24] | RW | Controls MUXSPI2 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI2 is the source clock of SPI2. | 0x1 |
| SPI1_SEL | [23:20] | RW | Controls MUXSPI1 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI1 is the source clock of SPI1. | 0x1 |
| SPI0_SEL | [19:16] | RW | Controls MUXSPI0 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0101 = SCLK_HDMIPHY 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXSPI0 is the source clock of SPI0. | 0x1 |
| RSVD | [15:10] | - | Reserved | 0x0 |
| SPDIF_SEL | [9:8] | RW | Controls MUXSPDIF 00 = SCLK_AUDIO0 01 = SCLK_AUDIO1 10 = SCLK_AUDIO2 11 = SPDIF_EXTCLK MUXSPDIF is the source clock of SPDIF. | 0x0 |
| AUDIO2 | [7:4] | RW | Controls MUXAUDIO2 | 0x5 |

| Name | Bit | Type | Description | Reset Value |
|----------------|-------|------|--|-------------|
| _SEL | | | 0000 = AUDIOCDCLK2 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO2 is the source clock of AUDIO2. | |
| AUDIO1 _SEL | [3:0] | RW | Controls MUXAUDIO1 0000 = AUDIOCDCLK1 0001 = Reserved 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = XXTI 0101 = XusbXTI 0110 = SCLKMPPLL_USER_T 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXAUDIO1 is the source clock of AUDIO1. | 0x5 |

5.9.1.37 CLK_SRC_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC258, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| JPEG_SEL | [8] | RW | Controls MUXJPEG 0 = MOUTJPEG_0 1 = MOUTJPEG_1 MUXJPEG is the source clock of JPEG core. | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| JPEG_1_SEL | [4] | RW | Controls MUXJPEG_1 0 = SCLKEPLL 1 = SCLKVPLL MUXJPEG_1 is the source clock of JPEG core. | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| JPEG_0_SEL | [0] | RW | Controls MUXJPEG_0 0 = SCLKMPPLL_USER_T 1 = SCLKAPLL MUXJPEG_0 is the source clock of JPEG core. | 0x0 |

5.9.1.38 CLK_SRC_MASK_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC320, Reset Value = 0x1111_1111

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| CSIS1_MASK | [28] | RW | Mask output clock of MUXCSIS1 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| CSIS0_MASK | [24] | RW | Mask output clock of MUXCSIS0 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| CAM1_MASK | [20] | RW | Mask output clock of MUXCAM1 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| CAM0_MASK | [16] | RW | Mask output clock of MUXCAM0 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| FIMC3_LCLK_MASK | [12] | RW | Mask output clock of MUXFIMC3_LCLK 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| FIMC2_LCLK_MASK | [8] | RW | Mask output clock of MUXFIMC2_LCLK 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| FIMC1_LCLK_MASK | [4] | RW | Mask output clock of MUXFIMC1_LCLK 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| FIMC0_LCLK_MASK | [0] | RW | Mask output clock of MUXFIMC0_LCLK 0 = Mask 1 = Unmask | 0x1 |

5.9.1.39 CLK_SRC_MASK_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC324, Reset Value = 0x0000_0111

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| HDMI_MASK | [0] | RW | Mask output clock of MUXHDMI 0 = Mask 1 = Unmask | 0x1 |

5.9.1.40 CLK_SRC_MASK_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC334, Reset Value = 0x0000_1111

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--|-------------|
| RSVD | [31:13] | - | Reserved | 0x0 |
| MIPIO_MASK | [12] | RW | Mask output clock of MUXMIPIO 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| MDNIE_PWM0_MASK | [8] | RW | Mask output clock of MUXMDNIE_PWM0 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MDNIE0_MASK | [4] | RW | Mask output clock of MUXMDNIE0 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| FIMD0_MASK | [0] | RW | Mask output clock of MUXFIMD0 0 = Mask 1 = Unmask | 0x1 |

5.9.1.41 CLK_SRC_MASK_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC338, Reset Value = 0x0000_1111

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:13] | - | Reserved | 0x0 |
| UART_ISP_MASK | [12] | RW | Mask output clock of MUXUART_ISP 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| SPI1_ISP_MASK | [8] | RW | Mask output clock of MUXSPI1_ISP 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| SPI0_ISP_MASK | [4] | RW | Mask output clock of MUXSPI0_ISP 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| PWM_ISP_MASK | [0] | RW | Mask output clock of MUXPWM_ISP 0 = Mask 1 = Unmask | 0x1 |

5.9.1.42 CLK_SRC_MASK_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC33C, Reset Value = 0x0000_0001

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|--|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| AUDIO0_MASK | [0] | RW | Mask output clock of MUXAUDIO0 0 = Mask 1 = Unmask | 0x1 |

5.9.1.43 CLK_SRC_MASK_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC340, Reset Value = 0x0101_1111

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| MIPIHSI_MASK | [24] | RW | Mask output clock of MUXMIPHSI 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [23:17] | - | Reserved | 0x0 |
| MMC4_MASK | [16] | RW | Mask output clock of MUXMMC4 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| MMC3_MASK | [12] | RW | Mask output clock of MUXMMC3 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| MMC2_MASK | [8] | RW | Mask output clock of MUXMMC2 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MMC1_MASK | [4] | RW | Mask output clock of MUXMMC1 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| MMC0_MASK | [0] | RW | Mask output clock of MUXMMC0 0 = Mask 1 = Unmask | 0x1 |

5.9.1.44 CLK_SRC_MASK_PERI0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC350, Reset Value = 0x0001_1111

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:17] | – | Reserved | 0x0 |
| UART4_MASK | [16] | RW | Mask output clock of MUXUART4 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [15:13] | – | Reserved | 0x0 |
| UART3_MASK | [12] | RW | Mask output clock of MUXUART3 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [11:9] | – | Reserved | 0x0 |
| UART2_MASK | [8] | RW | Mask output clock of MUXUART2 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | – | Reserved | 0x0 |
| UART1_MASK | [4] | RW | Mask output clock of MUXUART1 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | – | Reserved | 0x0 |
| UART0_MASK | [0] | RW | Mask output clock of MUXUART0 0 = Mask 1 = Unmask | 0x1 |

5.9.1.45 CLK_SRC_MASK_PERI1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC354, Reset Value = 0x0111_0111

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| SPI2_MASK | [24] | RW | Mask output clock of MUXSPI2 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| SPI1_MASK | [20] | RW | Mask output clock of MUXSPI1 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| SPI0_MASK | [16] | RW | Mask output clock of MUXSPI0 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [15:9] | - | Reserved | 0x0 |
| SPDIF_MASK | [8] | RW | Mask output clock of MUXSPDIF 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| AUDIO2_MASK | [4] | RW | Mask output clock of MUXAUDIO2 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| AUDIO1_MASK | [0] | RW | Mask output clock of MUXAUDIO1 0 = Mask 1 = Unmask | 0x1 |

5.9.1.46 CLK_MUX_STAT_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC410, Reset Value = 0x1111_1111

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| ONENAND_SEL | [30:28] | R | Selection signal status of MUXONENAND 001 = DOUT133 010 = DOUT166 1xx = Status that the mux is changing | 0x1 |
| RSVD | [27] | - | Reserved | 0x0 |
| ACLK_133_SEL | [26:24] | R | Selection signal status of MUXACLK_133 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [23] | - | Reserved | 0x0 |
| ACLK_160_SEL | [22:20] | R | Selection signal status of MUXACLK_160 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [19] | - | Reserved | 0x0 |
| ACLK_100_SEL | [18:16] | R | Selection signal status of MUXACLK_100 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [15] | - | Reserved | 0x0 |
| ACLK_200_SEL | [14:12] | R | Selection signal status of MUXACLK_200 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [11] | - | Reserved | 0x0 |
| VPLL_SEL | [10:8] | R | Selection signal status of MUXVPLL 001 = FINVPLL 010 = FOUTVPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | - | Reserved | 0x0 |
| EPLL_SEL | [6:4] | R | Selection signal status of MUXEPLL 001 = FINPLL 010 = FOUTEPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | - | Reserved | 0x0 |
| ONENAND_1_SEL | [2:0] | R | Selection signal status of MUXONENAND_1 001 = MOUTONENAND 010 = SCLKVPLL | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|---------------------------------------|-------------|
| | | | 1xx = Status that the mux is changing | |

5.9.1.47 CLK_MUX_STAT_TOP1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC414, Reset Value = 0x0111_1110

| Name | Bit | Type | Description | Reset Value |
|--------------------------|---------|------|--|-------------|
| RSVD | [31:27] | - | Reserved | 0x0 |
| ACLK_400_MCUISP _SUB_SEL | [26:24] | R | Selection signal status of MUXACLK_400_MCUISP 001 = FINPLL 010 = FOUTPOST_ACLK_400_MCUISP 1xx = Status that the mux is changing | 0x1 |
| RSVD | [23] | - | Reserved | 0x0 |
| ACLK_200_SUB _SEL | [22:20] | R | Selection signal status of MUXACLK_200 001 = FINPLL 010 = FOUTPOST_ACLK_200 1xx = Status that the mux is changing | 0x1 |
| RSVD | [19] | - | Reserved | 0x0 |
| ACLK_266_GPS _SUB_SEL | [18:16] | R | Selection signal status of MUXACLK_266_GPS 001 = FINPLL 010 = FOUTPOST_ACLK_266_GPS 1xx = Status that the mux is changing | 0x1 |
| RSVD | [15] | - | Reserved | 0x0 |
| MPLL_USER_SEL_T | [14:12] | R | Selection signal status of MUXMPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [11] | - | Reserved | 0x0 |
| ACLK_400_MCUISP _SEL | [10:8] | R | Selection signal status of MUXACLK_400_MCUISP 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | - | Reserved | 0x0 |
| ACLK_266_GPS _SEL | [6:4] | R | Selection signal status of MUXACLK_266_GPS 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3:0] | - | Reserved | 0x0 |

5.9.1.48 CLK_MUX_STAT_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC428, Reset Value = 0x0000_0111

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:11] | – | Reserved | 0x0 |
| MFC_SEL | [10:8] | R | Selection signal status of MUXMFC 001 = MOUTMFC_0 010 = MOUTMFC_1 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | – | Reserved | 0x0 |
| MFC_1_SEL | [6:4] | R | Selection signal status of MUXMFC_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | – | Reserved | 0x0 |
| MFC_0_SEL | [2:0] | R | Selection signal status of MUXMFC_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.49 CLK_MUX_STAT_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC42C, Reset Value = 0x0000_0111

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:11] | – | Reserved | 0x0 |
| G3D_SEL | [10:8] | R | Selection signal status of MUXG3D 001 = MOUTG3D_0 010 = MOUTG3D_1 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | – | Reserved | 0x0 |
| G3D_1_SEL | [6:4] | R | Selection signal status of MUXG3D_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | – | Reserved | 0x0 |
| G3D_0_SEL | [2:0] | R | Selection signal status of MUXG3D_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.50 CLK_MUX_STAT_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC458, Reset Value = 0x0000_0111

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:11] | - | Reserved | 0x0 |
| JPEG_SEL | [10:8] | R | Selection signal status of MUXJPEG 001 = MOUTJPEG_0 010 = MOUTJPEG_1 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | - | Reserved | 0x0 |
| JPEG_1_SEL | [6:4] | R | Selection signal status of MUXJPEG_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | - | Reserved | 0x0 |
| JPEG_0_SEL | [2:0] | R | Selection signal status of MUXJPEG_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.51 CLK_DIV_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC510, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------------|---------|------|--|-------------|
| RSVD | [31:27] | - | Reserved | 0x0 |
| ACLK_400_MCUISP_RATIO | [26:24] | RW | DIVACLK_266 Clock Divider Ratio ACLK_400_MCUISP = [MOUTACLK_400_MCUISP/(ACLK_400_MCUISP_RATIO + 1)] | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| ACLK_266_GPS_RATIO | [22:20] | RW | DIVACLK_266 Clock Divider Ratio ACLK_266_GPS = [MOUTACLK_266_GPS/(ACLK_266_GPS_RATIO + 1)] | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| ONENAND_RATIO | [18:16] | RW | DIVONENAND Clock Divider Ratio SCLK_ONENAND = [MOUTONENAND_1/(ONENAND_RATIO + 1)] | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| ACLK_133_RATIO | [14:12] | RW | DIVACLK_133 Clock Divider Ratio ACLK_133 = [MOUTACLK_133/(ACLK_133_RATIO + 1)] | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |
| ACLK_160_RATIO | [10:8] | RW | DIVACLK_160 Clock Divider Ratio ACLK_160 = [MOUTACLK_160/(ACLK_160_RATIO + 1)] | 0x0 |
| ACLK_100_RATIO | [7:4] | RW | DIVACLK_100 Clock Divider Ratio ACLK_100 = [MOUTACLK_100/(ACLK_100_RATIO + 1)] | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| ACLK_200_RATIO | [2:0] | RW | DIVACLK_200 Clock Divider Ratio ACLK_200 = [MOUTACLK_200/(ACLK_200_RATIO + 1)] | 0x0 |

5.9.1.52 CLK_DIV_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC520, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| CSIS1_RATIO | [31:28] | RW | DIVCSIS1 Clock Divider Ratio SCLK_CSIS1 = MOUTCSIS1/(CSIS1_RATIO + 1) | 0x0 |
| CSIS0_RATIO | [27:24] | RW | DIVCSIS0 Clock Divider Ratio SCLK_CSIS0 = MOUTCSIS0/(CSIS0_RATIO + 1) | 0x0 |
| CAM1_RATIO | [23:20] | RW | DIVCAM1 Clock Divider Ratio SCLK_CAM1 = MOUTCAM1/(CAM1_RATIO + 1) | 0x0 |
| CAM0_RATIO | [19:16] | RW | DIVCAM0 Clock Divider Ratio SCLK_CAM0 = MOUTCAM0/(CAM0_RATIO + 1) | 0x0 |
| FIMC3_LCLK_RATIO | [15:12] | RW | DIVFIMC3_LCLK Clock Divider Ratio SCLKFIMC3_LCLK = [MOUTFIMC3_LCLK / (FIMC3_LCLK_RATIO + 1)] | 0x0 |
| FIMC2_LCLK_RATIO | [11:8] | RW | DIVFIMC2_LCLK Clock Divider Ratio SCLKFIMC2_LCLK = [MOUTFIMC2_LCLK / (FIMC2_LCLK_RATIO + 1)] | 0x0 |
| FIMC1_LCLK_RATIO | [7:4] | RW | DIVFIMC1_LCLK Clock Divider Ratio SCLKFIMC1_LCLK = [MOUTFIMC1_LCLK / (FIMC1_LCLK_RATIO + 1)] | 0x0 |
| FIMC0_LCLK_RATIO | [3:0] | RW | DIVFIMC0_LCLK Clock Divider Ratio SCLKFIMC0_LCLK = [MOUTFIMC0_LCLK / (FIMC0_LCLK_RATIO + 1)] | 0x0 |

5.9.1.53 CLK_DIV_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC524, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| RSVD | [31:4] | - | Reserved | 0x0 |
| TV_BLK_RATIO | [3:0] | RW | DIVTV_BLK Clock Divider Ratio SCLK_PIXEL = SCLKVPLL/(TV_BLK_RATIO + 1) | 0x0 |

5.9.1.54 CLK_DIV_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC528, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:4] | - | Reserved | 0x0 |
| MFC_RATIO | [3:0] | RW | DIVMFC Clock Divider Ratio SCLK_MFC = MOUTMFC/(MFC_RATIO + 1) | 0x0 |

5.9.1.55 CLK_DIV_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC52C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|---|-------------|
| RSVD | [31:4] | - | Reserved | 0x0 |
| G3D_RATIO | [3:0] | RW | DIVG3D Clock Divider Ratio SCLK_G3D= MOUTG3D/(G3D_RATIO + 1) | 0x0 |

5.9.1.56 CLK_DIV_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC534, Reset Value = 0x0070_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|---------|------|--|-------------|
| RSVD | [31:24] | - | Reserved | 0x0 |
| MIPI0_PRE_RATIO | [23:20] | RW | DIVMIPI0_PRE Clock Divider Ratio SCLK_MIPI0 = DOUTMIPI0/(MIPI0_PRE_RATIO + 1) | 0x7 |
| MIPI0_RATIO | [19:16] | RW | DIVMIPI0 Clock Divider Ratio SCLK_MIPIDPHY4L = MOUTMIPI0 (/MIPI0_RATIO + 1) | 0x0 |
| MDNIE_PWM0_PRE_RATIO | [15:12] | RW | DIVMDNIE_PWM0_PRE Clock Divider Ratio SCLK_MDNIE_PWM0 = DOUTMDNIE_PWM0 (/MDNIE_PWM0_PRE_RATIO + 1) | 0x0 |
| MDNIE_PWM0_RATIO | [11:8] | RW | DIVMDNIE_PWM0 Clock Divider Ratio DOUTMDNIE_PWM0 = MOUTMDNIE_PWM0 (/MDNIE_PWM0_RATIO + 1) | 0x0 |
| MDNIE0_RATIO | [7:4] | RW | DIVMDNIE0 Clock Divider Ratio SCLK_MDNIE0 = MOUTMDNIE0 (/MDNIE0_RATIO + 1) | 0x0 |
| FIMD0_RATIO | [3:0] | RW | DIVFIMD0 Clock Divider Ratio SCLK_FIMD0 = MOUTFIMD0/(FIMD0_RATIO + 1) | 0x0 |

5.9.1.57 CLK_DIV_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC538, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|--|-------------|
| UART_ISP_RATIO | [31:28] | RW | DIVUART_ISP Clock Divider Ratio SCLK_UART_ISP = [DOUTUART_ISP/(UART_ISP_RATIO + 1)] | 0x0 |
| SPI1_ISP_PRE_RATIO | [27:20] | RW | DIVSPI1_ISP_PRE Clock Divider Ratio SCLK_SPI1_ISP = [DOUTSPI1_ISP/(SPI1_ISP_PRE_RATIO + 1)] | 0x0 |
| SPI1_ISP_RATIO | [19:16] | RW | DIVSPI1_ISP Clock Divider Ratio DOUTSPI1_ISP = [MOUTSPI1_ISP/(SPI1_ISP_RATIO + 1)] | 0x0 |
| SPI0_ISP_PRE_RATIO | [15:8] | RW | DIVSPI0_ISP_PRE Clock Divider Ratio SCLK_SPI0_ISP = [DOUTSPI0_ISP/(SPI0_ISP_PRE_RATIO + 1)] | 0x0 |
| SPI0_ISP_RATIO | [7:4] | RW | DIVSPI0_ISP Clock Divider Ratio DOUTSPI0_ISP = [MOUTSPI0_ISP/(SPI0_ISP_RATIO + 1)] | 0x0 |
| PWM_ISP_RATIO | [3:0] | RW | DIVPWM_ISP Clock Divider Ratio SCLK_PWM_ISP = [MOUTPWM_ISP/(PWM_ISP_RATIO + 1)] | 0x0 |

5.9.1.58 CLK_DIV_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC53C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:12] | - | Reserved | 0x0 |
| PCM0_RATIO | [11:4] | RW | DIVPCM0 Clock Divider Ratio SCLK_PCM0 = SCLK_AUDIO0/(PCM0_RATIO + 1) | 0x0 |
| AUDIO0_RATIO | [3:0] | RW | DIVAUDIO0 Clock Divider Ratio SCLK_AUDIO0 = MOUTAUDIO0/(AUDIO0_RATIO + 1) | 0x0 |

5.9.1.59 CLK_DIV_FSYS0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC540, Reset Value = 0x00B0_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [31:24] | - | Reserved | 0x0 |
| MIPIHSI_RATIO | [23:20] | RW | DIVMIPHSI Clock Divider Ratio SCLK_MIPHSI = [MOUTMIPHSI/(MIPIHSI_RATIO + 1)] | 0xB |
| RSVD | [19:0] | - | Reserved | 0x0 |

5.9.1.60 CLK_DIV_FSYS1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC544, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| MMC1_PRE_RATIO | [31:24] | RW | DIVMMC1_PRE Clock Divider Ratio SCLK_MMC1=DOUTMMC1/(MMC1_PRE_RATIO + 1)] | 0x0 |
| RSVD | [23:20] | - | Reserved | 0x0 |
| MMC1_RATIO | [19:16] | RW | DIVMMC1 Clock Divider Ratio DOUTMMC1 = MOUTMMC1/(MMC1_RATIO + 1) | 0x0 |
| MMC0_PRE_RATIO | [15:8] | RW | DIVMMC0_PRE Clock Divider Ratio SCLK_MMC0 =[DOUTMMC0/(MMC0_PRE_RATIO + 1)] | 0x0 |
| RSVD | [7:4] | - | Reserved | 0x0 |
| MMC0_RATIO | [3:0] | RW | DIVMMC0 Clock Divider Ratio DOUTMMC0 = MOUTMMC0/(MMC0_RATIO + 1) | 0x0 |

5.9.1.61 CLK_DIV_FSYS2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC548, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| MMC3_PRE_RATIO | [31:24] | RW | DIVMMC3_PRE Clock Divider Ratio SCLK_MMC3 =[DOUTMMC3/(MMC3_PRE_RATIO + 1)] | 0x0 |
| RSVD | [23:20] | - | Reserved | 0x0 |
| MMC3_RATIO | [19:16] | RW | DIVMMC3 Clock Divider Ratio DOUTMMC3 = MOUTMMC3/(MMC3_RATIO + 1) | 0x0 |
| MMC2_PRE_RATIO | [15:8] | RW | DIVMMC2_PRE Clock Divider Ratio SCLK_MMC2 =[DOUTMMC2/(MMC2_PRE_RATIO + 1)] | 0x0 |
| RSVD | [7:4] | - | Reserved | 0x0 |
| MMC2_RATIO | [3:0] | RW | DIVMMC2 Clock Divider Ratio DOUTMMC2 = MOUTMMC2/(MMC2_RATIO + 1) | 0x0 |

5.9.1.62 CLK_DIV_FSYS3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC54C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| MMC4_PRE_RATIO | [15:8] | RW | DIVMMC4_PRE Clock Divider Ratio SCLK_MMC4 =[DOUTMMC4/(MMC4_PRE_RATIO + 1)] | 0x0 |
| RSVD | [7:4] | - | Reserved | 0x0 |
| MMC4_RATIO | [3:0] | RW | DIVMMC4 Clock Divider Ratio DOUTMMC4 = MOUTMMC4/(MMC4_RATIO + 1) | 0x0 |

5.9.1.63 CLK_DIV_PERILO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC550, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| RSVD | [31:20] | - | Reserved | 0x0 |
| UART4_RATIO | [19:16] | RW | DIVUART4 Clock Divider Ratio SCLK_UART4 = MOUTUART4/(UART4_RATIO + 1) | 0x0 |
| UART3_RATIO | [15:12] | RW | DIVUART3 Clock Divider Ratio SCLK_UART3 = MOUTUART3/(UART3_RATIO + 1) | 0x0 |
| UART2_RATIO | [11:8] | RW | DIVUART2 Clock Divider Ratio SCLK_UART2 = MOUTUART2/(UART2_RATIO + 1) | 0x0 |
| UART1_RATIO | [7:4] | RW | DIVUART1 Clock Divider Ratio SCLK_UART1 = MOUTUART1/(UART1_RATIO + 1) | 0x0 |
| UART0_RATIO | [3:0] | RW | DIVUART0 Clock Divider Ratio SCLK_UART0 = MOUTUART0/(UART0_RATIO + 1) | 0x0 |

5.9.1.64 CLK_DIV_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC554, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| SPI1_PRE_RATIO | [31:24] | RW | DIVSPI1_PRE Clock Divider Ratio SCLK_SPI1 = DOUTSPI1/(SPI1_PRE_RATIO + 1) | 0x0 |
| RSVD | [23:20] | - | Reserved | 0x0 |
| SPI1_RATIO | [19:16] | RW | DIVSPI1 Clock Divider Ratio DOUTSPI1 = MOUTSPI1/(SPI1_RATIO + 1) | 0x0 |
| SPI0_PRE_RATIO | [15:8] | RW | DIVSPI0_PRE Clock Divider Ratio SCLK_SPI0 = DOUTSPI0/(SPI0_PRE_RATIO + 1) | 0x0 |
| RSVD | [7:4] | - | Reserved | 0x0 |
| SPI0_RATIO | [3:0] | RW | DIVSPI0 Clock Divider Ratio DOUTSPI0 = MOUTSPI0/(SPI0_RATIO + 1) | 0x0 |

5.9.1.65 CLK_DIV_PERIL2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC558, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| SPI2_PRE_RATIO | [15:8] | RW | DIVSPI2_PRE Clock Divider Ratio SCLK_SPI2 = DOUTSPI2/(SPI2_PRE_RATIO + 1) | 0x0 |
| RSVD | [7:4] | - | Reserved | 0x0 |
| SPI2_RATIO | [3:0] | RW | DIVSPI2 Clock Divider Ratio DOUTSPI2 = MOUTSPI2/(SPI2_RATIO + 1) | 0x0 |

5.9.1.66 CLK_DIV_PERIL3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC55C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:8] | - | Reserved | 0x0 |
| SLIMBUS_RATIO | [7:4] | RW | DIVSLIMBUS Clock Divider Ratio SCLK_SLIMBUS = SCLKEPLL/(SLIMBUS_RATIO + 1) | 0x0 |
| RSVD | [3:0] | - | Reserved | 0x0 |

5.9.1.67 CLK_DIV_PERIL4

- Base Address: 0x1003_0000
- Address = Base Address + 0xC560, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| PCM2_RATIO | [27:20] | RW | DIVPCM2 Clock Divider Ratio SCLK_PCM2 = SCLK_AUDIO2/(PCM2_RATIO + 1) | 0x0 |
| AUDIO2_RATIO | [19:16] | RW | DIVAUDIO2 Clock Divider Ratio SCLK_AUDIO2 = [MOUTAUDIO2/(AUDIO2_RATIO + 1)] | 0x0 |
| RSVD | [15:12] | – | Reserved | 0x0 |
| PCM1_RATIO | [11:4] | RW | DIVPCM1 Clock Divider Ratio SCLK_PCM1 = SCLK_AUDIO1/(PCM1_RATIO + 1) | 0x0 |
| AUDIO1_RATIO | [3:0] | RW | DIVAUDIO1 Clock Divider Ratio SCLK_AUDIO1 = [MOUTAUDIO1/(AUDIO1_RATIO + 1)] | 0x0 |

5.9.1.68 CLK_DIV_PERIL5

- Base Address: 0x1003_0000
- Address = Base Address + 0xC564, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:14] | – | Reserved | 0x0 |
| I2S2_RATIO | [13:8] | RW | DIVI2S2 Clock Divider Ratio SCLK_I2S2 = SCLK_AUDIO2/(I2S2_RATIO + 1) | 0x0 |
| RSVD | [7:6] | – | Reserved | 0x0 |
| I2S1_RATIO | [5:0] | RW | DIVI2S1 Clock Divider Ratio SCLK_I2S1 = SCLK_AUDIO1/(I2S1_RATIO + 1) | 0x0 |

5.9.1.69 CLK_DIV_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC568, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|--|-------------|
| RSVD | [31:4] | – | Reserved | 0x0 |
| JPEG_RATIO | [3:0] | RW | DIVJPEG Clock Divider Ratio ACLK_JPEG = MOUTJPEG/(JPEG_RATIO + 1) | 0x0 |

5.9.1.70 CLKDIV2_RATIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC580, Reset Value = 0x0110_1011

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:26] | - | Reserved | 0x0 |
| GPS_BLK | [25:24] | RW | PCLK Divider Ratio in GPS_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4 | 0x1 |
| RSVD | [23:22] | - | Reserved | 0x0 |
| TV_BLK | [21:20] | RW | PCLK Divider Ratio in TV_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4 | 0x1 |
| RSVD | [19:14] | - | Reserved | 0x0 |
| LCD_BLK | [13:12] | RW | PCLK Divider Ratio in LCD_BLK for 160 MHz domain 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4 | 0x1 |
| RSVD | [11:6] | - | Reserved | 0x0 |
| CAM_BLK | [5:4] | RW | PCLK Divider Ratio in CAM_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4 | 0x1 |
| RSVD | [3:2] | - | Reserved | 0x0 |
| FSYS_BLK | [1:0] | RW | PCLK Divider Ratio in FSYS_BLK 0 = Reserved 1 = Divides by 2 2 = Divides by 3 3 = Divides by 4 | 0x1 |

5.9.1.71 CLK_DIV_STAT_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC610, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| DIV_ACLK_400_MCUISP | [24] | R | DIVACLK_400_MCUISP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| DIV_ACLK_266_GPS | [20] | R | DIVACLK_266_GPS Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_ONENAND | [16] | R | DIVONENAND Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_ACLK_133 | [12] | R | DIVACLK_133 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_ACLK_160 | [8] | R | DIVACLK_160 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_ACLK_100 | [4] | R | DIVACLK_100 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_ACLK_200 | [0] | R | DIVACLK_200 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.72 CLK_DIV_STAT_CAM0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC620, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| DIV_CSIS1 | [28] | R | DIVCSIS1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| DIV_CSIS0 | [24] | R | DIVCSIS0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| DIV_CAM1 | [20] | R | DIVCAM1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_CAM0 | [16] | R | DIVCAM0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_FIMC3_LCLK | [12] | R | DIVFIMC3_LCLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_FIMC2_LCLK | [8] | R | DIVFIMC2_LCLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_FIMC1_LCLK | [4] | R | DIVFIMC1_LCLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_FIMC0_LCLK | [0] | R | DIVFIMC0_LCLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.73 CLK_DIV_STAT_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC624, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_TV_BLK | [0] | R | DIVTV_BLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.74 CLK_DIV_STAT_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC628, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_MFC | [0] | R | DIVMFC Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.75 CLK_DIV_STAT_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC62C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_G3D | [0] | R | DIVG3D Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.76 CLK_DIV_STAT_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC634, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|---|-------------|
| RSVD | [31:21] | - | Reserved | 0x0 |
| DIV_MIPI0_PRE | [20] | R | DIVMIPI0_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_MIPI0 | [16] | R | DIVMIPI0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_MDNIE_PWM0_PRE | [12] | R | DIVMDNIE_PWM0_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_MDNIE_PWM0 | [8] | R | DIVMDNIE_PWM0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_MDNIE0 | [4] | R | DIVMDNIE0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_FIMD0 | [0] | R | DIVFIMD0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.77 CLK_DIV_STAT_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC638, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| DIV_UART_ISP | [28] | R | DIVUART_ISP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [27:21] | - | Reserved | 0x0 |
| DIV_SPI1_ISP_PRE | [20] | R | DIVSPI1_ISP_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_SPI1_ISP | [16] | R | DIVSPI1_ISP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:9] | - | Reserved | 0x0 |
| DIV_SPI0_ISP_PRE | [8] | R | DIVSPI0_ISP_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_SPI0_ISP | [4] | R | DIVSPI0_ISP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_PWM_ISP | [0] | R | DIVPWM_ISP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.78 CLK_DIV_STAT_MAUDIO

- Base Address: 0x1003_0000
- Address = Base Address + 0xC63C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:5] | - | Reserved | 0x0 |
| DIV_PCM0 | [4] | R | DIVPCM0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_AUDIO0 | [0] | R | DIVAUDIO0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.79 CLK_DIV_STAT_FSYS0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC640, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| RSVD | [31:21] | - | Reserved | 0x0 |
| DIV_MIPIHSI | [20] | R | DIVMIPIHISI Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:0] | - | Reserved | 0x0 |

5.9.1.80 CLK_DIV_STAT_FSYS1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC644, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| DIV_MMC1_PRE | [24] | R | DIVMMC1_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:17] | - | Reserved | 0x0 |
| DIV_MMC1 | [16] | R | DIVMMC1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:9] | - | Reserved | 0x0 |
| DIV_MMC0_PRE | [8] | R | DIVMMC0_PR Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_MMC0 | [0] | R | DIVMMC0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.81 CLK_DIV_STAT_FSYS2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC648, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| DIV_MMC3_PRE | [24] | R | DIVMMC3_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:17] | - | Reserved | 0x0 |
| DIV_MMC3 | [16] | R | DIVMMC3 Stats 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:9] | - | Reserved | 0x0 |
| DIV_MMC2_PRE | [8] | R | DIVMMC2_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_MMC2 | [0] | R | DIVMMC2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.82 CLK_DIV_STAT_FSYS3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC64C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| DIV_MMC4_PRE | [8] | R | DIVMMC4_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_MMC4 | [0] | R | DIVMMC4 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.83 CLK_DIV_STAT_PERI0

- Base Address: 0x1003_0000
- Address = Base Address + 0xC650, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| DIV_UART4 | [16] | R | DIVUART4 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_UART3 | [12] | R | DIVUART3 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_UART2 | [8] | R | DIVUART2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_UART1 | [4] | R | DIVUART1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_UART0 | [0] | R | DIVUART0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.84 CLK_DIV_STAT_PERIL1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC654, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| DIV_SPI1_PRE | [24] | R | DIVSPI1_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:17] | - | Reserved | 0x0 |
| DIV_SPI1 | [16] | R | DIVSPI1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:9] | - | Reserved | 0x0 |
| DIV_SPI0_PRE | [8] | R | DIVSPI0_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_SPI0 | [0] | R | DIVSPI0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.85 CLK_DIV_STAT_PERIL2

- Base Address: 0x1003_0000
- Address = Base Address + 0xC658, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| DIV_SPI2_PRE | [8] | R | DIVSPI2_PRE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_SPI2 | [0] | R | DIVSPI2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.86 CLK_DIV_STAT_PERIL3

- Base Address: 0x1003_0000
- Address = Base Address + 0xC65C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|--|-------------|
| RSVD | [31:5] | - | Reserved | 0x0 |
| DIV_SLIMBUS | [4] | R | DIVSLIMBUS Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:0] | - | Reserved | 0x0 |

5.9.1.87 CLK_DIV_STAT_PERIL4

- Base Address: 0x1003_0000
- Address = Base Address + 0xC660, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:21] | - | Reserved | 0x0 |
| DIV_PCM2 | [20] | R | DIVPCM2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_AUDIO2 | [16] | R | DIVAUDIO2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:5] | - | Reserved | 0x0 |
| DIV_PCM1 | [4] | R | DIVPCM1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_AUDIO1 | [0] | R | DIVAUDIO1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.88 CLK_DIV_STAT_PERIL5

- Base Address: 0x1003_0000
- Address = Base Address + 0xC664, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| DIV_I2S2 | [8] | R | DIVI2S2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:1] | - | Reserved | 0x0 |
| DIV_I2S1 | [0] | R | DIVI2S1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.89 CLK_DIV_STAT_CAM1

- Base Address: 0x1003_0000
- Address = Base Address + 0xC668, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_JPEG | [0] | R | DIVJPEG Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.90 CLKDIV2_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0xC680, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| GPS_BLK | [24] | R | PCLK Divider Status in TV_BLK 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| TV_BLK | [20] | R | PCLK Divider Status in TV_BLK 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:13] | - | Reserved | 0x0 |
| LCD_BLK | [12] | R | PCLK Divider Status in LCD_BLK for 160 MHz domain 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:5] | - | Reserved | 0x0 |
| CAM_BLK | [4] | R | PCLK Divider Status in CAM_BLK 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| FSYS_BLK | [0] | R | PCLK Divider Status in FSYS_BLK 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.91 CLK_GATE_BUS_FSYS1

- Address = 0x1003_C744, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------------------|---------|------|--|-------------|
| RSVD | [31:23] | - | Reserved | 0x1FF |
| PCLK_ASYNCAXIS_GPS_FSYSD | [22] | RW | Gating APB clock for ASYNCAXIS_GPS_FSYSD 0 = Mask 1 = Pass | 0x1 |
| PCLK_AXI_FSYSS | [21] | RW | Gating APB clock for AXI_FSYSS 0 = Mask 1 = Pass | 0x1 |
| PCLK_AXI_FSYSD | [20] | RW | Gating APB clock for AXI_FSYSD 0 = Mask 1 = Pass | 0x1 |
| RSVD | [19:18] | - | Reserved | 0x3 |
| PCLK_PPMUFILE | [17] | RW | Gating APB clock for PPMUFILE 0 = Mask 1 = Pass | 0x1 |
| PCLK_ADC | [16] | RW | Gating APB clock for FSYS ADC 0 = Mask 1 = Pass | 0x1 |
| RSVD | [15:0] | - | Reserved | 0xFFFF |

5.9.1.92 CLK_GATE_IP_CAM

- Base Address: 0x1003_0000
- Address = Base Address + 0xC920, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------------|---------|------|---|-------------|
| RSVD | [31:18] | — | Reserved | 0xFFFF |
| RSVD | [19] | — | Reserved | 0x1 |
| CLK_PIXELASYN_CM1 | [18] | RW | Gating all clocks for PIXELASYN CM1 0 = Mask 1 = Pass | 0x1 |
| CLK_PIXELASYN_CM0 | [17] | RW | Gating all clocks for PIXELASYN CM0 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUCAMIF | [16] | RW | Gating all clocks for PPMUCAMIF 0 = Mask 1 = Pass | 0x1 |
| RSVD | [12:15] | — | Reserved | 0xF |
| CLK_SMMUJPEG | [11] | RW | Gating all clocks for SMMUJPEG 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUFIMC3 | [10] | RW | Gating all clocks for SMMUFIMC3 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUFIMC2 | [9] | RW | Gating all clocks for SMMUFIMC2 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUFIMC1 | [8] | RW | Gating all clocks for SMMUFIMC1 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUFIMC0 | [7] | RW | Gating all clocks for SMMUFIMC0 0 = Mask 1 = Pass | 0x1 |
| CLK_JPEG | [6] | RW | Gating all clocks for JPEG 0 = Mask 1 = Pass | 0x1 |
| CLK_CSIS1 | [5] | RW | Gating all clocks for CSIS1 0 = Mask 1 = Pass | 0x1 |
| CLK_CSIS0 | [4] | RW | Gating all clocks for CSIS0 0 = Mask 1 = Pass | 0x1 |
| CLK_FIMC3 | [3] | RW | Gating all clocks for FIMC3 0 = Mask 1 = Pass | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|-----------|-----|------|---|-------------|
| CLK_FIMC2 | [2] | RW | Gating all clocks for FIMC2 0 = Mask 1 = Pass | 0x1 |
| CLK_FIMC1 | [1] | RW | Gating all clocks for FIMC1 0 = Mask 1 = Pass | 0x1 |
| CLK_FIMC0 | [0] | RW | Gating all clocks for FIMC0 0 = Mask 1 = Pass | 0x1 |

5.9.1.93 CLK_GATE_IP_TV

- Base Address: 0x1003_0000
- Address = Base Address + 0xC924, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:6] | – | Reserved | 0x3FFFFFFF |
| CLK_PPMUTV | [5] | RW | Gating all clocks for PPMUTV 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUTV | [4] | RW | Gating all clocks for SMMUTV 0 = Mask 1 = Pass | 0x1 |
| CLK_HDMI | [3] | RW | Gating all clocks for HDMI link 0 = Mask 1 = Pass | 0x1 |
| RSVD | [2] | – | Reserved | 0x1 |
| CLK_MIXER | [1] | RW | Gating all clocks for MIXER 0 = Mask 1 = Pass | 0x1 |
| CLK_VP | [0] | RW | Gating all clocks for VP 0 = Mask 1 = Pass | 0x1 |

5.9.1.94 CLK_GATE_IP_MFC

- Base Address: 0x1003_0000
- Address = Base Address + 0xC928, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:5] | – | Reserved | 0x7FFFFFFF |
| CLK_PPMUMFC_R | [4] | RW | Gating all clocks for PPMUMFC_R 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUMFC_L | [3] | RW | Gating all clocks for PPMUMFC_L 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUMFC_R | [2] | RW | Gating all clocks for SMMUMFC_R 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUMFC_L | [1] | RW | Gating all clocks for SMMUMFC_L 0 = Mask 1 = Pass | 0x1 |
| CLK_MFC | [0] | RW | Gating all clocks for MFC 0 = Mask 1 = Pass | 0x1 |

5.9.1.95 CLK_GATE_IP_G3D

- Base Address: 0x1003_0000
- Address = Base Address + 0xC92C, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|---|-------------|
| RSVD | [31:3] | — | Reserved | 0x1FFFFFFF |
| RSVD | [2] | — | Reserved | 0x1 |
| CLK_PPMUG3D | [1] | RW | Gating all clocks for PPMUG3D 0 = Mask 1 = Pass | 0x1 |
| CLK_G3D | [0] | RW | Gating all clocks for G3D 0 = Mask 1 = Pass | 0x1 |

5.9.1.96 CLK_GATE_IP_LCD

- Base Address: 0x1003_0000
- Address = Base Address + 0xC934, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:6] | — | Reserved | 0x3FFFFFFF |
| CLK_PPMULCD0 | [5] | RW | Gating all clocks for PPMULCD0 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUFIMD0 | [4] | RW | Gating all clocks for SMMUFIMD0 0 = Mask 1 = Pass | 0x1 |
| CLK_DSIM0 | [3] | RW | Gating all clocks for DSIM0 0 = Mask 1 = Pass | 0x1 |
| CLK_MDNIE0 | [2] | RW | Gating all clocks for MDNIE0 0 = Mask 1 = Pass | 0x1 |
| CLK_MIE0 | [1] | RW | Gating all clocks for MIE0 0 = Mask 1 = Pass | 0x1 |
| CLK_FIMD0 | [0] | RW | Gating all clocks for FIMD0 0 = Mask 1 = Pass | 0x1 |

5.9.1.97 CLK_GATE_IP_ISP

- Base Address: 0x1003_0000
- Address = Base Address + 0xC938, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------------|--------|------|---|-------------|
| RSVD | [31:4] | - | Reserved | 0xFFFFFFFF |
| CLK_UART_ISP_SCLK | [3] | RW | Gating SCLK clocks for UART_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_SPI1_ISP_SCLK | [2] | RW | Gating SCLK clocks for SPI1_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_SPI0_ISP_SCLK | [1] | RW | Gating SCLK clocks for SPI0_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_PWM_ISP_SCLK | [0] | RW | Gating SCLK clocks for PWM_ISP 0 = Mask 1 = Pass | 0x1 |

5.9.1.98 CLK_GATE_IP_FSYS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC940, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:18] | - | Reserved | 0x1FF |
| CLK_PPMUFILE | [17] | RW | Gating all clocks for PPMUFILE 0 = Mask 1 = Pass | 0x1 |
| CLK_NFCON | [16] | RW | Gating all clocks for NFCON 0 = Mask 1 = Pass | 0x1 |
| CLK_ONENAND | [15] | RW | Gating all clocks for ONENAND 0 = Mask 1 = Pass | 0x1 |
| RSVD | [14] | - | Reserved | 0x1 |
| CLK_USBDEVICE | [13] | RW | Gating all clocks for USB Device 0 = Mask 1 = Pass | 0x1 |
| CLK_USBHOST | [12] | RW | Gating all clocks for USB HOST 0 = Mask 1 = Pass | 0x1 |
| CLK_SROMC | [11] | RW | Gating all clocks for SROM 0 = Mask 1 = Pass | 0x1 |
| CLK_MIPIHSI | [10] | RW | Gating all clocks for MIPIHSI 0 = Mask 1 = Pass | 0x1 |
| CLK_SDMMC4 | [9] | RW | Gating all clocks for SDMMC4 0 = Mask 1 = Pass | 0x1 |
| CLK_SDMMC3 | [8] | RW | Gating all clocks for SDMMC3 0 = Mask 1 = Pass | 0x1 |
| CLK_SDMMC2 | [7] | RW | Gating all clocks for SDMMC2 0 = Mask 1 = Pass | 0x1 |
| CLK_SDMMC1 | [6] | RW | Gating all clocks for SDMMC1 0 = Mask 1 = Pass | 0x1 |
| CLK_SDMMC0 | [5] | RW | Gating all clocks for SDMMC0 0 = Mask 1 = Pass | 0x1 |
| CLK_TSI | [4] | RW | Gating all clocks for TSI | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|---|-------------|
| | | | 0 = Mask 1 = Pass | |
| RSVD | [3:2] | - | Reserved | 0x3 |
| CLK_PDMA1 | [1] | RW | Gating all clocks for PDMA1 0 = Mask 1 = Pass | 0x1 |
| CLK_PDMA0 | [0] | RW | Gating all clocks for PDMA0 0 = Mask 1 = Pass | 0x1 |

5.9.1.99 CLK_GATE_IP_GPS

- Base Address: 0x1003_0000
- Address = Base Address + 0xC94C, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|---|-------------|
| RSVD | [31:4] | — | Reserved | 0xFFFFFFFF |
| RSVD | [3] | — | Reserved | 0x1 |
| CLK_PPMUGPS | [2] | RW | Gating all clocks for PPMUGPS 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUGPS | [1] | RW | Gating all clocks for SMMUGPS 0 = Mask 1 = Pass | 0x1 |
| CLK_GPS | [0] | RW | Gating all clocks for GPS 0 = Mask 1 = Pass | 0x1 |

5.9.1.100 CLK_GATE_IP_PERIL

- Base Address: 0x1003_0000
- Address = Base Address + 0xC950, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| RSVD | [31:28] | - | Reserved | 0xF |
| CLK_AC97 | [27] | RW | Gating all clocks for AC97 0 = Mask 1 = Pass | 0x1 |
| CLK_SPDIF | [26] | RW | Gating all clocks for SPDIF 0 = Mask 1 = Pass | 0x1 |
| CLK_SLIMBUS | [25] | RW | Gating all clocks for Slimbus 0 = Mask 1 = Pass | 0x1 |
| CLK_PWM | [24] | RW | Gating all clocks for PWM 0 = Mask 1 = Pass | 0x1 |
| CLK_PCM2 | [23] | RW | Gating all clocks for PCM2 0 = Mask 1 = Pass | 0x1 |
| CLK_PCM1 | [22] | RW | Gating all clocks for PCM1 0 = Mask 1 = Pass | 0x1 |
| CLK_I2S2 | [21] | RW | Gating all clocks for I2S2 0 = Mask 1 = Pass | 0x1 |
| CLK_I2S1 | [20] | RW | Gating all clocks for I2S1 0 = Mask 1 = Pass | 0x1 |
| RSVD | [19] | - | Reserved | 0x1 |
| CLK_SPI2 | [18] | RW | Gating all clocks for SPI2 0 = Mask 1 = Pass | 0x1 |
| CLK_SPI1 | [17] | RW | Gating all clocks for SPI1 0 = Mask 1 = Pass | 0x1 |
| CLK_SPI0 | [16] | RW | Gating all clocks for SPI0 0 = Mask 1 = Pass | 0x1 |
| RSVD | [15] | - | Reserved | 0x1 |
| CLK_I2CHDMI | [14] | RW | Gating all clocks for I2CHDMI 0 = Mask 1 = Pass | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|-----------|------|------|---|-------------|
| CLK_I2C7 | [13] | RW | Gating all clocks for I2C7 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C6 | [12] | RW | Gating all clocks for I2C6 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C5 | [11] | RW | Gating all clocks for I2C5 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C4 | [10] | RW | Gating all clocks for I2C4 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C3 | [9] | RW | Gating all clocks for I2C3 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C2 | [8] | RW | Gating all clocks for I2C2 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C1 | [7] | RW | Gating all clocks for I2C1 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C0 | [6] | RW | Gating all clocks for I2C0 0 = Mask 1 = Pass | 0x1 |
| RSVD | [5] | - | Reserved | 0x1 |
| CLK_UART4 | [4] | RW | Gating all clocks for UART4 0 = Mask 1 = Pass | 0x1 |
| CLK_UART3 | [3] | RW | Gating all clocks for UART3 0 = Mask 1 = Pass | 0x1 |
| CLK_UART2 | [2] | RW | Gating all clocks for UART2 0 = Mask 1 = Pass | 0x1 |
| CLK_UART1 | [1] | RW | Gating all clocks for UART1 0 = Mask 1 = Pass | 0x1 |
| CLK_UART0 | [0] | RW | Gating all clocks for UART0 0 = Mask 1 = Pass | 0x1 |

5.9.1.101 CLK_GATE_BLOCK

- Base Address: 0x1003_0000
- Address = Base Address + 0xC970, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| RSVD | [31:8] | - | Reserved | 0xFFFFFFFF |
| CLK_GPS | [7] | RW | Gating all clocks for GPS_BLK (GPS) 0 = Mask 1 = Pass | 0x1 |
| RSVD | [6:5] | - | Reserved | 0x3 |
| CLK_LCD | [4] | RW | Gating all clocks for LCD_BLK (FIMD0, MIE0, and DSIM0) 0 = Mask 1 = Pass | 0x1 |
| CLK_G3D | [3] | RW | Gating all clocks for G3D_BLK (G3D) 0 = Mask 1 = Pass | 0x1 |
| CLK_MFC | [2] | RW | Gating all clocks for MFC_BLK (MFC) 0 = Mask 1 = Pass | 0x1 |
| CLK_TV | [1] | RW | Gating all clocks for TV_BLK (VP, MIXER, TVENC, and HDMI) 0 = Mask 1 = Pass | 0x1 |
| CLK_CAM | [0] | RW | Gating all clocks for CAM_BLK (FIMC0, FIMC1, FIMC2, and FIMC3) 0 = Mask 1 = Pass | 0x1 |

5.9.1.102 CLKOUT_CMU_TOP

- Base Address: 0x1003_0000
- Address = Base Address + 0xCA00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX Selection 00000 = EPLL_FOUT 00001 = VPLL_FOUT 00010 = SCLK_HDMI24M 00011 = SCLK_USBPHY0 00101 = SCLK_HDMIPHY 00110 = AUDIOCDCLK0 00111 = AUDIOCDCLK1 01000 = AUDIOCDCLK2 01001 = SPDIF_EXTCLK 01010 = ACLK_160 01011 = ACLK_133 01100 = ACLK_200 01101 = ACLK_100 01110 = SCLK_MFC 01111 = SCLK_G3D 10000 = ACLK_400_MCUIISP 10001 = CAM_A_PCLK 10010 = CAM_B_PCLK 10011 = S_RXBYTECLKHS0_2L 10100 = S_RXBYTECLKHS0_4L 10101 = RX_HALF_BYTE_CLK_CSIS0 10110 = RX_HALF_BYTE_CLK_CSIS1 10111 = SCLK_JPEG 11000 = SCLK_PWM_ISP 11001 = SCLK_SPI0_ISP 11010 = SCLK_SPI1_ISP 11011 = SCLK_UART_ISP 11100 = SCLK_MIPIHSI 11101 = SCLK_HDMI 11110 = SCLK_FIMD0 11111 = SCLK_PCM0 | 0x0 |

5.9.1.103 CLKOUT_CMU_TOP_DIV_STAT

- Base Address: 0x1003_0000
- Address = Base Address + 0xCA04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.104 MPLL_LOCK

- Base Address: 0x1004_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0FFF

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| PLL_LOCKTIME | [15:0] | RW | Required period to generate a stable clock output Set (270cycles × PDIV) to PLL_LOCKTIME for the PLL maximum lock time. 1 cycle = 1/FREF=1/(FIN/PDIV) The maximum PLL lock time is 22.5 usec where FIN is 24 MHz, PDIV is 2 and PLL_LOCKTIME is 540. | 0xFFFF |

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking \geq the maximum locktime)

5.9.1.105 MPLL_CON0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0108, Reset Value = 0x0064_0300

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| ENABLE | [31] | RW | PLL Enable Control 0 = Disables 1 = Enables | 0x0 |
| RSVD | [30] | - | Reserved | 0x0 |
| LOCKED | [29] | R | PLL Locking Indication 0 = Unlocks 1 = Locks If ENABLE_LOCK_DET = 0, then this field is set to 1 after the locking time. The lock-time is set using the MPLL_LOCK SFR register. If ENABLE_LOCK_DET = 1, then this field is set when the hardware lock detector meets the PLL locking condition. This bit is Read only. | 0x0 |
| RSVD | [28] | - | Reserved | 0x0 |
| FSEL | [27] | RW | Monitors Frequency Select Pin 0 = $F_{VCO_OUT} = F_{REF}$ 1 = $F_{VCO_OUT} = F_{VCO}$ | 0x0 |
| RSVD | [26] | - | Reserved | 0x0 |
| MDIV | [25:16] | RW | PLL M Divide Value | 0x64 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| PDIV | [13:8] | RW | PLL P Divide Value | 0x3 |
| RSVD | [7:3] | - | Reserved | 0x0 |
| SDIV | [2:0] | RW | PLL S Divide Value | 0x0 |

The reset value of MPLL_CON0 generates a 800 MHz output clock for an input clock frequency of 24 MHz.

The equation to calculate the output frequency is: $F_{OUT} = MDIV \times FIN / (PDIV \times 2SDIV)$: $21.9 \text{ MHz} \leq F_{OUT} \leq 1400 \text{ MHz}$

- The conditions MDIV, PDIV, SDIV for APLL and MPLL should meet are:
MDIV: $1 \leq MDIV \leq 63$
PDIV: $1 \leq PDIV \leq 63$
SDIV: $0 \leq SDIV \leq 5$
- $F_{ref} = FIN / PDIV$ Fref should fall in the range of: $2 \text{ MHz} \leq F_{ref} \leq 12 \text{ MHz}$
 $F_{VCO} = MDIV \times FIN / PDIV$ FVCO should fall in the range of: $700 \text{ MHz} \leq F_{VCO} \leq 1400 \text{ MHz}$

Refer to the section [5.3.1 Recommended PLL PMS Value for APLL and MPLL](#) for recommended PMS values.

5.9.1.106 MPLL_CON1

- Base Address: 0x1004_0000
- Address = Base Address + 0x010C, Reset Value = 0x0080_3800

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| RESV1 | [24] | RW | Specifies status of Linear-Region Detector (LDR) when it detects a low signal | 0x0 |
| RESV0 | [23] | RW | Specifies VCO range boost-up when the signal is high. | 0x1 |
| BYPASS | [22] | RW | If BYPASS = 1, then it enables bypass mode ($F_{OUT} = F_{IN}$) If BYPASS = 0, then the PLL3500X operates normally. | 0x0 |
| DCC_ENB | [21] | RW | Decides whether DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal. | 0x0 |
| AFC_ENB | [20] | RW | Decides whether AFC is enabled or not. 0 = Enables AFC 1 = Disables AFC It is an active low signal. | 0x0 |
| RSVD | [19:18] | - | Reserved | 0x0 |
| RSVD | [17] | - | Reserved | 0x0 |
| FEED_EN | [16] | RW | Enable pin for FEED_OUT | 0x0 |
| LOCK_CON_OUT | [15:14] | RW | Specifies lock detector settings of the output margin. | 0x0 |
| LOCK_CON_IN | [13:12] | RW | Specifies lock detector settings of the input margin. | 0x3 |
| LOCK_CON_DLY | [11:8] | RW | Specifies lock detector settings of the detection resolution. | 0x8 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| AFC | [4:0] | RW | AFC value | 0x0 |

Refer to the section [5.3.1 Recommended PLL PMS Value for APLL and MPLL](#) for recommended AFC_ENB and AFC values.

NOTE: The other PLL control inputs should be set as:

| | |
|------------------|------------------|
| RESV1 = 0 | RESV0 = 0 |
| DCC_ENB = 1 | EXTAFC = 0 |
| LOCK_CON_IN = 3 | LOCK_CON_OUT = 0 |
| LOCK_CON_DLY = 8 | AFC_ENB = 0 |

5.9.1.107 CLK_SRC_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0200, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|---------|------|--|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| MUX_G2D_ACP_SEL | [28] | RW | Control MUXG2D_ACP, which is the source clock of G2D_ACP core 0 = MOUTG2D_ACP_0 1 = MOUTG2D_ACP_1 | 0x0 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| MUX_G2D_ACP_1_SEL | [24] | RW | Control MUXG2D_ACP_1, which is the source clock of G2D_ACP core 0 = SCLKEPLL 1 = SCLKVPLL | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| MUX_G2D_ACP_0_SEL | [20] | RW | Control MUXG2D_ACP_0, which is the source clock of G2D_ACP core 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| MUX_PWI_SEL | [19:16] | RW | Controls MUXPWI 0000 = XXTI 0001 = XusbXTI 0010 = SCLK_HDMI24M 0011 = SCLK_USBPHY0 0100 = SCLK_USBPHY1 0101 = SCLK_HDMIPHY 0110 = SCLKMPLL 0111 = SCLKEPLL 1000 = SCLKVPLL Others = Reserved MUXPWI is the clock source of PWI. | 0x1 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| MUX_MPLL_SEL | [12] | RW | Controls MUXMPLL 0 = FINPLL 1 = MOUTMPLLFOUT | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| MUX_DPHY_SEL | [8] | RW | Controls MUXDPHY 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_DMC_BUS_SEL | [4] | RW | Controls MUXDMC_BUS 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|---|-------------|
| MUX_C2C_SEL | [0] | RW | Controls MUXC2C 0 = SCLKMPLL 1 = SCLKAPLL | 0x0 |

5.9.1.108 CLK_SRC_MASK_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0300, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| PWI_MASK | [16] | RW | Mask output clock of MUXPWI 0 = Mask 1 = Unmask | 0x1 |
| RSVD | [15:0] | - | Reserved | 0x0 |

5.9.1.109 CLK_MUX_STAT_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0400, Reset Value = 0x1110_1111

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| G2D_ACP_SEL | [30:28] | R | Selection signal status of MUXG2D_ACP 001 = MOUTG2D_ACP_0 010 = MOUTG2D_ACP_1 1xx = On changing | 0x1 |
| RSVD | [27] | - | Reserved | 0x0 |
| G2D_ACP_1_SEL | [26:24] | R | Selection signal status of MUXG2D_ACP_1 001 = SCLKEPLL 010 = SCLKVPLL 1xx = On changing | 0x1 |
| RSVD | [23] | - | Reserved | 0x0 |
| G2D_ACP_0_SEL | [22:20] | R | Selection signal status of MUXG2D_ACP_0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = On changing | 0x1 |
| RSVD | [19:15] | - | Reserved | 0x0 |
| MPLL_SEL | [14:12] | R | Selection signal status of MUXMPLL 001 = FINPLL 010 = MOUTMPLLFOUT 1xx = Status that the mux is changing | 0x1 |
| DPHY_SEL | [10:8] | R | Selection signal status of MUXDMC0 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [7] | - | Reserved | 0x0 |
| DMC_BUS_SEL | [6:4] | R | Selection signal status of MUXDMC_BUS 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [3] | - | Reserved | 0x0 |
| C2C_SEL | [2:0] | R | Selection signal status of MUXC2C 001 = SCLKMPLL 010 = SCLKAPLL 1xx = Status that the mux is changing | 0x1 |

5.9.1.110 CLK_DIV_DMC0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0500, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| RSVD | [31:23] | - | Reserved | 0x0 |
| DMCP_RATIO | [22:20] | RW | DIVCK133 Clock Divider Ratio ACLK_DMCP = ACLK_DMCD/(DMCP_RATIO + 1) | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| DMCD_RATIO | [18:16] | RW | DIVDMCD Clock Divider Ratio ACLK_DMCD = DOUTDMC/(DMCD_RATIO + 1) | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| DMC_RATIO | [14:12] | RW | DIVDMC Clock Divider Ratio DOUTDMC = MOUTDMC_BUS/(DMC_RATIO + 1) | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |
| DPHY_RATIO | [10:8] | RW | DIVDPHY Clock Divider Ratio SCLK_DPHY = MOUTDPHY/(DPHY_RATIO + 1) | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| ACP_PCLK_RATIO | [6:4] | RW | DIVACP Clock Divider Ratio PCLK_ACP = ACLK_ACP/(ACP_PCLK_RATIO + 1) | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| ACP_RATIO | [2:0] | RW | DIVACP Clock Divider Ratio ACLK_ACP = MOUTDMC_BUS/(ACP_RATIO + 1) | 0x0 |

5.9.1.111 CLK_DIV_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0504, Reset Value = 0x0000_1000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| DPM_RATIO | [30:24] | RW | DIVDPM Clock Divider Ratio It decides frequency of DPM channel clock. | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| DVSEM_RATIO | [22:16] | RW | DIVDVSEM Clock Divider Ratio It decides frequency for PWM frame time slot in DVS emulation mode. | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| C2C_ACLK_RATIO | [14:12] | RW | C2C_ACLK Clock Divider Ratio ACLK_C2C = [MOUTC2C_ACLK/(C2C_ACLK_RATIO + 1)] | 0x1 |
| PWI_RATIO | [11:8] | RW | DIVPWI Clock Divider Ratio SCLK_PWI = MOUTPWI/(PWI_RATIO + 1) | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| C2C_RATIO | [6:4] | RW | C2C clock divider ratio SCLK_C2C = MOUTC2C/(C2C_RATIO + 1) | 0x0 |
| G2D_ACP_RATIO | [3:0] | RW | DIVG2D_ACP clock divider ratio SCLK_G2D_ACP= MOUTG2D_ACP/(G2D_ACP_RATIO + 1) | 0x0 |

5.9.1.112 CLK_DIV_STAT_DMC0

- Base Address: 0x1004_0000
- Address = Base Address + 0x0600, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:21] | - | Reserved | 0x0 |
| DIV_DMCP | [20] | R | DIVDMCP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_DMCD | [16] | R | DIVDMCD Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_DMC | [12] | R | DIVDMC Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_DPHY | [8] | R | DIVDPHY Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_ACP_PCLK | [4] | R | DIVACP_PCLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_ACP | [0] | R | DIVACP Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.113 CLK_DIV_STAT_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0604, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| DIV_DPM | [24] | R | DIVDPM Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:17] | - | Reserved | 0x0 |
| DIV_DVSEM | [16] | R | DIVDVSEM Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_C2C_ACLK | [12] | R | DIVC2C_ACLK Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_PWI | [8] | R | DIVPWI Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_C2C | [4] | R | DIVC2C status 0 = Stable 1 = Divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_G2D_ACP | [0] | R | DIVG2D_ACP status 0 = Stable 1 = Divider is changing | 0x0 |

5.9.1.114 CLK_GATE_IP_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0900, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|--|-------------|
| CLK_GPIOC2C | [31] | RW | Gating all clocks for GPIOC2C 0 = Mask 1 = Pass | 0x1 |
| RSVD | [30:29] | - | Reserved | 0x3 |
| CLK_ASYNC_CPU_XIUR | [28] | RW | Gating all clocks for ASYNC_CPU_XIUR 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_C2C_XIUL | [27] | RW | Gating all clocks for ASYNC_C2C_XIUL 0 = Mask 1 = Pass | 0x1 |
| CLK_C2C | [26] | RW | Gating all clocks for C2C 0 = Mask 1 = Pass | 0x1 |
| RSVD | [25] | - | Reserved | 0x1 |
| CLK_SMMUG2D_ACP | [24] | RW | Gating all clocks for SMMUG2D_ACP 0 = Mask 1 = Pass | 0x1 |
| CLK_G2D_ACP | [23] | RW | Gating all clocks for G2D_ACP 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_GDR | [22] | RW | Gating all clocks for ASYNC_GDR 0 = Mask 1 = Pass | 0x1 |
| CLK_ASYNC_GDL | [21] | RW | Gating all clocks for ASYNC_GDL 0 = Mask 1 = Pass | 0x1 |
| CLK_GIC | [20] | RW | Gating all clocks for GIC 0 = Mask 1 = Pass | 0x1 |
| RSVD | [19] | - | Reserved | 0x1 |
| CLK_IEM_IEC | [18] | RW | Gating all clocks for IEM IEC 0 = Mask 1 = Pass | 0x1 |
| CLK_IEM_APP | [17] | RW | Gating all clocks for IEM APP 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUACP | [16] | RW | Gating all clocks for PPMUCPU 0 = Mask 1 = Pass | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [15:14] | - | Reserved | 0x3 |
| CLK_ID_REMAPPER | [13] | RW | Gating all clocks for ID_REMAPPER 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMUSSS | [12] | RW | Gating all clocks for SMMUSSS 0 = Mask 1 = Pass | 0x1 |
| RSVD | [11] | - | Reserved | 0x1 |
| CLK_PPMUCPU | [10] | RW | Gating all clocks for PPMUCPU 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUDMC1 | [9] | RW | Gating all clocks for PPMUDMC1 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUDMC0 | [8] | RW | Gating all clocks for PPMUDMC0 0 = Mask 1 = Pass | 0x1 |
| RSVD | [7] | - | Reserved | 0x1 |
| CLK_FBMDMC1 | [6] | RW | Gating all clocks for FBMDMC1 0 = Mask 1 = Pass | 0x1 |
| CLK_FBMDMC0 | [5] | RW | Gating all clocks for FBMDMC0 0 = Mask 1 = Pass | 0x1 |
| CLK_SSS | [4] | RW | Gating all clocks for SSS 0 = Mask 1 = Pass | 0x1 |
| RSVD | [3] | - | Reserved | 0x1 |
| CLK_INT_COMB | [2] | RW | Gating all clocks for INT_COMB 0 = Mask 1 = Pass | 0x1 |
| RSVD | [1] | - | Reserved | 0x1 |
| CLK_DREX2 | [0] | RW | Gating all clocks for DREX2 0 = Mask 1 = Pass | 0x1 |

5.9.1.115 CLK_GATE_IP_DMC1

- Base Address: 0x1004_0000
- Address = Base Address + 0x0904, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|--|-------------|
| RSVD | [31:4] | - | Reserved | 0xFFFFFFFF |
| CLK_TZASC_LR | [3] | RW | Gating all clocks for TZASC_LR 0 = Mask 1 = Pass | 0x1 |
| CLK_TZASC_LW | [2] | RW | Gating all clocks for TZASC_LW 0 = Mask 1 = Pass | 0x1 |
| CLK_TZASC_RR | [1] | RW | Gating all clocks for TZASC_RR 0 = Mask 1 = Pass | 0x1 |
| CLK_TZASC_RW | [0] | RW | Gating all clocks for TZASC_RW 0 = Mask 1 = Pass | 0x1 |

5.9.1.116 CLKOUT_CMU_DMC

- Base Address: 0x1004_0000
- Address = Base Address + 0x0A00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX Selection 00000 = ACLK_DMCD 00001 = ACLK_DMCP 00010 = ACLK_ACP 00011 = PCLK_ACP 00100 = SCLK_DMC 00101 = SCLK_DPHY 00110 = MPLL_FOUT/2 00111 = SCLK_PWI 01000 = Reserved 01001 = SCLK_C2C 01010 = ACLK_C2C | 0x0 |

5.9.1.117 CLKOUT_CMU_DMC_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x0A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.118 DCGIDX_MAP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x1000, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|---|-------------|
| DCGIDX_MAP0 | [31:0] | RW | IEC Configuration for DCG Index Map[31:0] | 0xFFFFFFFF |

5.9.1.119 DCGIDX_MAP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x1004, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|--|-------------|
| DCGIDX_MAP1 | [31:0] | RW | IEC Configuration for DCG Index Map[63:32] | 0xFFFFFFFF |

5.9.1.120 DCGIDX_MAP2

- Base Address: 0x1004_0000
- Address = Base Address + 0x1008, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|--|-------------|
| DCGIDX_MAP2 | [31:0] | RW | IEC Configuration for DCG Index Map[95:64] | 0xFFFFFFFF |

5.9.1.121 DCGPERF_MAP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x1020, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---------------------------|-------------|
| DCGPERF_MAP0 | [31:0] | RW | DCG Performance Map[31:0] | 0xFFFFFFFF |

5.9.1.122 DCGPERF_MAP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x1024, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|----------------------------|-------------|
| DCGPERF_MAP1 | [31:0] | RW | DCG Performance Map[63:32] | 0xFFFFFFFF |

5.9.1.123 DVCIDX_MAP

- Base Address: 0x1004_0000
- Address = Base Address + 0x1040, Reset Value = 0x00FF_FFFF

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x0 |
| DCGPERF_MAP0 | [23:0] | RW | IEC Configuration for DVC Index Map[23:0] | 0xFFFFFFF |

5.9.1.124 FREQ_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x1060, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---------------------------------|-------------|
| RSVD | [31:24] | – | Reserved | 0x0 |
| FREQ_CPU | [23:0] | RW | Maximum Frequency of CPU in KHz | 0x0 |

5.9.1.125 FREQ_DPM

- Base Address: 0x1004_0000
- Address = Base Address + 0x1064, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--------------------------|-------------|
| RSVD | [31:24] | – | Reserved | 0x0 |
| FREQ_DPM | [23:0] | RW | Maximum Frequency of DPM | 0x0 |

5.9.1.126 DVSEMCLK_EN

- Base Address: 0x1004_0000
- Address = Base Address + 0x1080, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|----------------------------|-------------|
| RSVD | [31:1] | – | Reserved | 0x0 |
| DVSEMCLK_EN | [0] | RW | DVS Emulation Clock Enable | 0x0 |

5.9.1.127 MAXPERF

- Base Address: 0x1004_0000
- Address = Base Address + 0x1084, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:1] | – | Reserved | 0x0 |
| MAXPERF_EN | [0] | RW | Maximum Performance Enable 0 = Disables 1 = Enables | 0x0 |

5.9.1.128 DMC_PAUSE_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x1094, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:19] | – | Reserved | 0x0 |
| STATE | [18:16] | R | Specifies current status for debugging | 0x0 |
| RSVD | [15:1] | – | Reserved | 0x0 |
| DMC_PAUSE_ENABLE | [0] | RW | Enable pause function for DREX2 DVFS DREX2 pause function works when DMC_RATIO or DMCD_RATIO in CLK_DIV_DMC0 register is changed. | 0x0 |

5.9.1.129 DDRPHY_LOCK_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x1098, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|---|-------------|
| USE_CTRL_LOCKED | [31] | RW | Use ctrl_locked signal coming from LPDDR_PHY to check DLL lock-time duration 0 = Uses internal counter to measure DLL lock duration 1 = Uses ctrl_locked signal | 0x0 |
| CTRL_START_ENABLE | [30] | RW | Enable Clearing of ctrl_start signal | 0x0 |
| CTRL_RESYNC_ENABLE | [29] | RW | Enable ctrl_resync pulse generation | 0x0 |
| CTRL_RESYNC_MASK | [28] | RW | Mask ctrl_resync pulse form DREX2 during DDRPHY DLL Locking time | 0x0 |
| RSVD | [28:18] | – | Reserved | 0x0 |
| CURR_STATE | [17:16] | R | Specifies current status for debugging | 0x0 |
| DUR_LOCK_WAIT | [15:8] | RW | Sets Duration for DLL Lock Wait of DDR_PHY | 0x0 |
| DUR_CTRL_ST_CLR | [7:0] | RW | Sets Duration for clearing ctrl_start signal of DDR_PHY | 0x0 |

5.9.1.130 C2C_STATE

- Base Address: 0x1004_0000
- Address = Base Address + 0x109C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|------------------------------|-------------|
| RSVD | [31:3] | – | Reserved | 0x0 |
| CURR_STATE | [2:0] | R | Current State of C2C SEC FSM | 0x0 |

5.9.1.131 APLL_LOCK

- Base Address: 0x1004_0000
- Address = Base Address + 0x4000, Reset Value = 0x0000_0FFF

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:16] | - | Reserved | 0x0 |
| PLL_LOCKTIME | [15:0] | RW | <p>Required period to generate a stable clock output Set (270cycles × PDIV) to PLL_LOCKTIME for the PLL maximum lock time.</p> <p>1 cycle = 1/FREF = 1/(FIN/PDIV)</p> <p>The maximum PLL lock time is 22.5 usec where FIN is 24 MHz, PDIV is 2 and PLL_LOCKTIME is 540.</p> | 0xFFFF |

The maximum lock time means the waiting time for locking in the worst case. Therefore, the user of this PLL must wait for more than the maximum lock time unconditionally before the PLL is locked. (Waiting time before locking \geq the maximum locktime)

5.9.1.132 APLL_CON0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4100, Reset Value = 0x0064_0300

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| ENABLE | [31] | RW | PLL Enable Control 0 = Disables 1 = Enables | 0x0 |
| RSVD | [30] | - | Reserved | 0x0 |
| LOCKED | [29] | R | PLL Locking Indication 0 = Unlocks 1 = Locks If ENABLE_LOCK_DET = 0, then this field is set to 1 after the locking time. The lock-time is set using the APLL_LOCK SFR register. If ENABLE_LOCK_DET = 1, then this field is set when the hardware lock detector meets the PLL locking condition. This bit is Read only. | 0x0 |
| RSVD | [28] | - | Reserved | 0x0 |
| FSEL | [27] | RWX | Monitors Frequency Select Pin 0 = $F_{VCO_OUT} = F_{REF}$ 1 = $F_{VCO_OUT} = F_{VCO}$ | 0x0 |
| RSVD | [26] | - | Reserved | 0x0 |
| MDIV | [25:16] | RWX | PLL M Divide Value | 0xC8 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| PDIV | [13:8] | RWX | PLL P Divide Value | 0x6 |
| RSVD | [7:3] | - | Reserved | 0x0 |
| SDIV | [2:0] | RWX | PLL S Divide Value | 0x1 |

The reset value of APLL_CON0 generates an 800 MHz output clock for an input clock frequency of 24 MHz.

The equation to calculate the output frequency is: $F_{OUT} = MDIV \times FIN / (PDIV \times 2SDIV)$

F_{OUT} should fall in the range of: $21.9 \text{ MHz} \leq F_{OUT} \leq 1400 \text{ MHz}$

The conditions MDIV, PDIV, SDIV for APLL and MPLL should meet are:

- PDIV: $1 \leq PDIV \leq 63$
- MDIV: $64 \leq MDIV \leq 1023$
- SDIV: $0 \leq SDIV \leq 5$
- $F_{ref} = FIN / PDIV$ F_{ref} should fall in the range of: $2 \text{ MHz} \leq F_{ref} \leq 12 \text{ MHz}$
- $F_{VCO} = MDIV \times F_{IN} / PDIV$ F_{VCO} should fall in the range of: $700 \text{ MHz} \leq F_{VCO} \leq 1400 \text{ MHz}$

Refer to the section [5.3.1 Recommended PLL PMS Value for APLL and MPLL](#) for recommended PMS values.

5.9.1.133 APLL_CON1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4104, Reset Value = 0x0080_3800

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| RESV1 | [24] | RW | Specifies status of Linear-Region Detector (LDR) when it detects a low signal. | 0x0 |
| RESV0 | [23] | RW | Specifies VCO range boost-up when the signal is high. | 0x1 |
| BYPASS | [22] | RW | If BYPASS = 1, bypass mode is enabled. ($F_{OUT} = F_{IN}$) If BYPASS = 0, PLL3500X operates normally. | 0x0 |
| DCC_ENB | [21] | RW | Decides whether the DCC is enabled or not. 0 = Enables DCC 1 = Disables DCC It is an active low signal. | 0x0 |
| AFC_ENB | [20] | RWX | Decides whether AFC is enabled or not. When AFC is enabled, it calibrates VCO automatically. 0 = Enables AFC 1 = Disables AFC It is an active low signal. | 0x0 |
| RSVD | [19:18] | - | Reserved | 0x0 |
| RSVD | [17] | - | Reserved | 0x0 |
| FEED_EN | [16] | RW | Enable signal for FEED_OUT | 0x0 |
| LOCK_CON_OUT | [15:14] | RW | Specifies Lock detector settings of the output margin. | 0x0 |
| LOCK_CON_IN | [13:12] | RW | Specifies Lock detector settings of the input margin. | 0x3 |
| LOCK_CON_DLY | [11:8] | RW | Specifies Lock detector settings of the detection resolution. | 0x8 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| AFC | [4:0] | RWX | AFC value | 0x0 |

AFC automatically selects adaptive frequency curve of VCO using switched current bank for wide range, high phase noise (or Jitter), and fast lock time.

Refer to the section [5.3.1 Recommended PLL PMS Value for APLL and MPLL](#) for recommended AFC_ENB and AFC values.

NOTE: The other PLL control inputs should be set as:

| | |
|------------------|------------------|
| RESV1 = 0 | RESV0 = 0 |
| DCC_ENB = 1 | EXTAFC = 0 |
| LOCK_CON_IN = 3 | LOCK_CON_OUT = 0 |
| LOCK_CON_DLY = 8 | AFC_ENB = 0 |

5.9.1.134 CLK_SRC_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4200, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------|---------|------|--|-------------|
| RSVD | [31:25] | - | Reserved | 0x0 |
| MUX_MPLL_USER_SEL_C | [24] | RW | Controls MUXMPLL 0 = FINPLL 1 = FOUTMPLL | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| MUX_HPM_SEL | [20] | RW | Controls MUXHPM 0 = MOUTAPLL 1 = SCLKMPLL | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| MUX_CORE_SEL | [16] | RW | Controls MUXCORE 0 = MOUTAPLL 1 = SCLKMPLL | 0x0 |
| RSVD | [15:1] | - | Reserved | 0x0 |
| MUX_APPL_SEL | [0] | RW | Controls MUXAPLL 0 = FINPLL 1 = MOUTAPLLFOUT | 0x0 |

5.9.1.135 CLK_MUX_STAT_CPU

- Address = 0x1004_4400, Reset Value = 0x0111_0001

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:27] | - | Reserved | 0x0 |
| MPLL_USER_SEL_C | [26:24] | R | Selection signal status of MUXMPPLL 001 = FINMPLL 010 = FOUTMPLL 1xx = Status that the mux is changing | 0x1 |
| HPM_SEL | [22:20] | R | Selection signal status of MUXHPM 001 = MOUTAPLL 010 = SCLKMPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [19] | - | Reserved | 0x0 |
| CORE_SEL | [18:16] | R | Selection signal status of MUXCORE 001 = MOUTAPLL 010 = SCLKMPLL 1xx = Status that the mux is changing | 0x1 |
| RSVD | [15:3] | - | Reserved | 0x0 |
| RSVD | [7:3] | - | Reserved | 0x0 |
| APLL_SEL | [2:0] | R | Selection signal status of MUXAPLL 001 = FINPLL 010 = MOUTAPLLFOUT 1xx = Status that the mux is changing | 0x1 |

5.9.1.136 CLK_DIV_CPU0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4500, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| CORE2_RATIO | [30:28] | RW | DIVCORE2 Clock Divider Ratio ARMCLK = DOUTCORE/(CORE2_RATIO + 1) | 0x0 |
| RSVD | [27] | - | Reserved | 0x0 |
| APLL_RATIO | [26:24] | RW | DIVAPLL Clock Divider Ratio SCLKAPLL = MOUTAPLL/APLL_RATIO + 1) | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| PCLK_DBG_RATIO | [22:20] | RW | DIVPCLK_DBG Clock Divider Ratio PCLK_DBG = ATCLK/(PCLK_DBG_RATIO + 1) | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| ATB_RATIO | [18:16] | RW | DIVATB Clock Divider Ratio ATCLK = MOUTCORE/(ATB_RATIO + 1) | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| PERIPH_RATIO | [14:12] | RW | DIVPERIPH Clock Divider Ratio PERIPHCLK = DOUTCORE/(PERIPH_RATIO + 1) | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |
| COREM1_RATIO | [10:8] | RW | DIVCOREM1 Clock Divider Ratio ACLK_COREM1 = ARMCLK/(COREM1_RATIO + 1) | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| COREM0_RATIO | [6:4] | RW | DIVCOREM0 Clock Divider Ratio ACLK_COREM0 = ARMCLK/(COREM0_RATIO + 1) | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| CORE_RATIO | [2:0] | RWX | DIVCORE Clock Divider Ratio DIVCORE_OUT = MOUTCORE/(CORE_RATIO + 1) | 0x0 |

5.9.1.137 CLK_DIV_CPU1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4504, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| RSVD | [31:11] | - | Reserved | 0x0 |
| CORES_RATIO | [10:8] | RW | DIVCORES Clock Divider Ratio ACLK_CORES = ARMCLK/(CORES_RATIO + 1) | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| HPM_RATIO | [6:4] | RWX | DIVHPM Clock Divider Ratio SCLK_HPM = DOUTCOPY/(HPM_RATIO + 1) | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| COPY_RATIO | [2:0] | RWX | DIVCOPY Clock Divider Ratio DOUTCOPY = MOUTHPM/(COPY_RATIO + 1) | 0x0 |

5.9.1.138 CLK_DIV_STAT_CPU0

- Base Address: 0x1004_0000
- Address = Base Address + 0x4600, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| DIV_CORE2 | [28] | R | DIVCORE2 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| DIV_APLL | [24] | R | DIVAPLL Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [23:21] | - | Reserved | 0x0 |
| DIV_PCLK_DBG | [20] | R | DIVPCLK_DBG Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [19:17] | - | Reserved | 0x0 |
| DIV_ATB | [16] | R | DIVATB Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [15:13] | - | Reserved | 0x0 |
| DIV_PERIPH | [12] | R | DIVPERIPH Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [11:9] | - | Reserved | 0x0 |
| DIV_COREM1 | [8] | R | DIVCOREM1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_COREM0 | [4] | R | DIVCOREM0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_CORE | [0] | R | DIVCORE Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.139 CLK_DIV_STAT_CPU1

- Base Address: 0x1004_0000
- Address = Base Address + 0x4604, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| DIV_CORES | [8] | R | DIVCORES Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_HPM | [4] | R | DIVHPM Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_COPY | [0] | R | DIVCOPY Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.140 CLK_GATE_IP_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4900, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:2] | - | Reserved | 0xFFFF_FFF3 |
| CLK_CSSYS | [1] | RW | Gating all clocks for CoreSight and SecureJTAG 0 = Mask 1 = Pass | 0x1 |
| CLK_HPM | [0] | RW | Gating all clocks for HPM 0 = Mask 1 = Pass | 0x1 |

5.9.1.141 CLKOUT_CMU_CPU

- Base Address: 0x1004_0000
- Address = Base Address + 0x4A00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [31:17] | - | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | - | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX Selection 00000 = APLL_FOUT/2 00001 = Reserved 00010 = Reserved 00011 = Reserved 00100 = ARMCLK/2 00101 = ACLK_COREM0 00110 = ACLK_COREM1 00111 = ACLK_CORES 01000 = ATCLK 01001 = PERIPHCLK 01010 = PCLK_DBG 01011 = SCLK_HPM ATCLK and PCLK_DBG are the gated clocks. You should not gate ATCLK or PCLK_DBG clocks before changing the DIV_RATIO value on selection of ATCLK or PCLK_DBG. | 0x0 |

5.9.1.142 CLKOUT_CMU_CPU_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x4A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.143 ARMCLK_STOPCTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5000, Reset Value = 0x0404_0404

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| L2_PRE_WAIT_CNT | [31:24] | RW | Specifies clock freeze cycle before the CLAMP_L2_0 and CLAMP_L2_1rising transition | 0x4 |
| L2_POST_WAIT_CNT | [23:16] | RW | Specifies clock freeze cycle after the L2RET1N_0 and L2RET1N_1 rising transition | 0x4 |
| PRE_WAIT_CNT | [15:8] | RW | Specifies clock freeze cycle before the ARM clamp (CLAMPCORE0, CLAMPCORE1, CLAMPCOREOUT, CLAMPL2_0, and CLAMPL2_1) or reset signal (nCPURESET, nDBGRESET, nSCURESET, L2nRESET, nWDRESET, nPERIPHRESET, and nPTMRESET) transition | 0x4 |
| POST_WAIT_CNT | [7:0] | RW | Specifies clock freeze cycle after the ARM clamp (CLAMPCORE0, CLAMPCORE1, CLAMPCOREOUT, CLAMPL2_0, and CLAMPL2_1) or reset signal (nCPURESET, nDBGRESET, nSCURESET, L2nRESET, nWDRESET, nPERIPHRESET, and nPTMRESET) transition | 0x4 |

5.9.1.144 ATCLK_STOPCTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5004, Reset Value = 0x0000_0404

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:16] | – | Reserved | 0x0 |
| PRE_WAIT_CNT | [15:8] | RW | Specifies clock freeze cycle before the ATRESETn, nPRESETDBG, and CSSYS_nRESET signal transition | 0x4 |
| POST_WAIT_CNT | [7:0] | RW | Specifies clock freeze cycle after the ATRESETn, nPRESETDBG, and CSSYS_nRESET signal transition | 0x4 |

5.9.1.145 PWR_CTRL

- Base Address: 0x1004_0000
- Address = Base Address + 0x5020, Reset Value = 0x0000_04FF

| Name | Bit | Type | Description | Reset Value |
|--------------------------|---------|------|---|-------------|
| RSVD | [31] | - | Reserved | 0x0 |
| CORE2_RATIO | [30:28] | RW | DIVCORE2 on WFI/WFE Set DIVCORE2 clock divider ratio when both ARM cores are in Wait For Interrupt/Event state | 0x0 |
| RSVD | [27:21] | - | Reserved | 0x0 |
| CSCLK_AUTO_ENB_IN_DEBUG | [20] | RW | Forces CoreSight clocks to toggle when the debugger is attached 0 = Disables 1 = Enables | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| CORE_RATIO | [18:16] | RW | DIVCORE on WFI/WFE Set DIVCORE clock divider ratio when both ARM cores are in Wait For Interrupt/Event state | 0x0 |
| RSVD | [15:11] | - | Reserved | 0x0 |
| F4D_CORESIGHT_EN | [10] | RW | Gating F4D Coresight clocks both ARM cores in IDLE mode 0 = Mask 1 = Pass | 0x1 |
| DIVCORE2_DOWN_ENB | [9] | RW | Enable ARMCLK Down feature when both ARM cores are in IDLE mode for DIVCORE2 0 = Disables 1 = Enables | 0x0 |
| DIVCORE_DOWN_ENB | [8] | RW | Enable ARMCLK Down feature when both ARM cores are in IDLE mode for DIVCORE 0 = Disables 1 = Enables | 0x0 |
| USE_STANDBYWFE_ARM_CORE3 | [7] | RW | Use ARM CORE3 STANDBYWFE to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFE_ARM_CORE2 | [6] | RW | Use ARM CORE2 STANDBYWFE to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFE_ARM_CORE1 | [5] | RW | Use ARM CORE1 STANDBYWFE to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFE_ARM_CORE0 | [4] | RW | Use ARM CORE0 STANDBYWFE to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFI_ARM_CORE3 | [3] | RW | Use ARM CORE3 STANDBYWFI to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFI_ARM_CORE2 | [2] | RW | Use ARM CORE2 STANDBYWFI to change ARMCLK frequency in ARM IDLE state | 0x1 |
| USE_STANDBYWFI | [1] | RW | Use ARM CORE1 STANDBYWFI to change | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|------------------------------|-----|------|--|-------------|
| _ARM_CORE1 | | | ARMCLK frequency in ARM IDLE state | |
| USE_STANDBYWFI _ARM_CORE0 | [0] | RW | Use ARM CORE0 STANDBYWFI to change ARMCLK frequency in ARM IDLE state | 0x1 |

5.9.1.146 PWR_CTRL2

- Base Address: 0x1004_0000
- Address = Base Address + 0x5024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| RSVD | [31:26] | - | Reserved | 0x0 |
| DIVCORE2_UP_ENB | [25] | RW | Enable ARMCLK Up feature when both ARM cores exit from IDLE mode for DIVCORE2 0 = Disables 1 = Enables | 0x0 |
| DIVCORE_UP_ENB | [24] | RW | Enable ARMCLK Up feature when both ARM cores exit from IDLE mode for DIVCORE 0 = Disables 1 = Enables | 0x0 |
| DUR_STANDBY2 | [23:16] | RW | Sets duration to change to the normal divider value from the middle divider value This bit should be left-shifted by 4-bit before comparing it to the counter value. | 0x0 |
| DUR_STANDBY1 | [15:8] | RW | Sets duration to change to the middle divider value from the divider value in ARM idle state. This bit should be left-shifted by 4-bit before comparing it to counter value. | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| UP_CORE2_RATIO | [6:4] | RW | Specifies DIVCORE2 clock divider ratio when ARM0 or ARM1 cores are not in a wait state for an interrupt or event to occur. | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| UP_CORE_RATIO | [2:0] | RW | Specifies DIVCORE clock divider ratio when ARM0 or ARM1 cores are not in a wait state for an interrupt or event to occur. | 0x0 |

5.9.1.147 L2_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5400, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:29] | - | Reserved | 0x0 |
| L2IDLE | [28] | R | Indicates L2 cache controller is in idle state | 0x0 |
| RSVD | [27:25] | - | Reserved | 0x0 |
| L2_CLKSTOPPED | [24] | R | Indicates L2 cache controller is in standby-mode | 0x0 |
| RSVD | [23] | - | Reserved | 0x0 |
| TAGSETUPLAT | [22:20] | R | Setup Latency for Tag RAM | 0x0 |
| RSVD | [19] | - | Reserved | 0x0 |
| TAGREADLAT | [18:16] | R | Read access Latency for Tag RAM | 0x0 |
| RSVD | [15] | - | Reserved | 0x0 |
| TAGWRITELAT | [14:12] | R | Write access Latency for Tag RAM | 0x0 |
| RSVD | [11] | - | Reserved | 0x0 |
| DATASETUPLAT | [10:8] | R | Setup Latency for Data RAM | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| DATAREADLAT | [6:4] | R | Read access Latency for Data RAM | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| DATAWRITELAT | [2:0] | R | Write access Latency for Data RAM | 0x0 |

5.9.1.148 CPU_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5410, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| RSVD | [31:12] | – | Reserved | 0x0 |
| PMUPRIV | [11:8] | R | Returns status of the Cortex-A9 processor 0 = User mode 1 = Privileged mode | 0x0 |
| PMUSECURE | [7:4] | R | Returns security status of the Cortex-A9 processor 0 = Non-secure state 1 = Secure state | 0x0 |
| SMPNAMP | [3:0] | R | Specifies signals AMP or SMP mode for each Cortex-A9 processor 0 = Asymmetric signal 1 = Symmetric signal | 0x0 |

5.9.1.149 PTM_STATUS

- Base Address: 0x1004_0000
- Address = Base Address + 0x5420, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| RSVD | [31:4] | – | Reserved | 0x0 |
| PTMPWRUP0 | [3] | R | PTM for CPU0 is active | 0x0 |
| PTMPWRUP1 | [2] | R | PTM for CPU1 is active | 0x0 |
| PTMIDLEnACK0 | [1] | R | PTM for CPU0 is an idle state indicator | 0x0 |
| PTMIDLEnACK1 | [0] | R | PTM for CPU1 is an idle state indicator | 0x0 |

5.9.1.150 CLK_DIV_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8300, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:7] | - | Reserved | 0x0 |
| ISPDIV1_RATIO | [6:4] | RW | ISPDIV1 Clock Divider Ratio ISPDIV1_CLK = ACLK_200/(ISPDIV1_RATIO + 1) | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| ISPDIV0_RATIO | [2:0] | RW | ISPDIV0 Clock Divider Ratio ISPDIV0_CLK = ACLK_200/(ISPDIV0_RATIO + 1) | 0x0 |

5.9.1.151 CLK_DIV_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8304, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:11] | - | Reserved | 0x0 |
| MCUISPDIV1_RATIO | [10:8] | RW | MCUISPDIV1 Clock Divider Ratio MCUISPDIV1_CLK = [MOUTMCUISPDIV0_CLK/(MCUISPDIV1_RATIO + 1)] | 0x0 |
| RSVD | [7] | - | Reserved | 0x0 |
| MCUISPDIV0_RATIO | [6:4] | RW | MCUISPDIV0 Clock Divider Ratio MCUISPDIV0_CLK = [ACLK_400_MCUIPS / (MCUISPDIV0_RATIO + 1)] | 0x0 |
| RSVD | [3] | - | Reserved | 0x0 |
| MPWMMDIV_RATIO | [2:0] | RW | MPWM Clock Divider Ratio MPWMMDIV_CLK = [MOUTISPDIV1_CLK / (MPWMMDIV_RATIO + 1)] | 0x0 |

5.9.1.152 CLK_DIV_STAT_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8400, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|---|-------------|
| RSVD | [31:5] | - | Reserved | 0x0 |
| DIV_ISPDIV1 | [4] | R | ISPDIV1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_ISPDIV0 | [0] | R | ISPDIV0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.153 CLK_DIV_STAT_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8404, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|--------|------|--|-------------|
| RSVD | [31:9] | - | Reserved | 0x0 |
| DIV_MCUISPDIV1 | [8] | R | DIVMCUISP1 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [7:5] | - | Reserved | 0x0 |
| DIV_MCUISPDIV0 | [4] | R | DIVMCUISP0 Status 0 = Stable 1 = Status that the divider is changing | 0x0 |
| RSVD | [3:1] | - | Reserved | 0x0 |
| DIV_MPWMDIV | [0] | R | DIVMPWM Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.154 CLK_GATE_IP_ISP0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8800, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| CLK_UART_ISP | [31] | RW | Gating all clocks for UART_ISP except SCLK 0 = Mask 1 = Pass | 0x1 |
| CLK_WDT_ISP | [30] | RW | Gating all clocks for WDT_ISP 0 = Mask 1 = Pass | 0x1 |
| RSVD | [29] | - | Reserved | 0x1 |
| CLK_PWM_ISP | [28] | RW | Gating all clocks for PWM_ISP except SCLK 0 = Mask 1 = Pass | 0x1 |
| CLK_MTCADC_ISP | [27] | RW | Gating all clocks for MTCADC_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C1_ISP | [26] | RW | Gating all clocks for I2C1_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_I2C0_ISP | [25] | RW | Gating all clocks for I2C0_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_MPWM_ISP | [24] | RW | Gating all clocks for MPWM_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_MCUCTL_ISP | [23] | RW | Gating all clocks for MCUCTL_ISP 0 = Mask 1 = Pass | 0x1 |
| RSVD | [22] | - | Reserved | 0x1 |
| CLK_PPMUISPX | [21] | RW | Gating all clocks for PPMUISPX 0 = Mask 1 = Pass | 0x1 |
| CLK_PPMUISPMX | [20] | RW | Gating all clocks for PPMUISPMX 0 = Mask 1 = Pass | 0x1 |
| RSVD | [13:19] | - | Reserved | 0x7F |
| CLK_SMMU_LITE1 | [12] | RW | Gating all clocks for SMMU_LITE1 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMU_LITE0 | [11] | RW | Gating all clocks for SMMU_LITE0 0 = Mask 1 = Pass | 0x1 |

| Name | Bit | Type | Description | Reset Value |
|--------------|------|------|--|-------------|
| CLK_SMMU_FD | [10] | RW | Gating all clocks for SMMU_FD 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMU_DRC | [9] | RW | Gating all clocks for SMMU_DRC 0 = Mask 1 = Pass | 0x1 |
| CLK_SMMU_ISP | [8] | RW | Gating all clocks for SMMU_ISP 0 = Mask 1 = Pass | 0x1 |
| CLK_GICISP | [7] | RW | Gating all clocks for GICISP 0 = Mask 1 = Pass | 0x1 |
| RSVD | [6] | - | Reserved | 0x1 |
| CLK_MCUISP | [5] | RW | Gating all clocks for MCUISP 0 = Mask 1 = Pass | 0x1 |
| CLK_LITE1 | [4] | RW | Gating all clocks for LITE1 0 = Mask 1 = Pass | 0x1 |
| CLK_LITE0 | [3] | RW | Gating all clocks for LITE0 0 = Mask 1 = Pass | 0x1 |
| CLK_FD | [2] | RW | Gating all clocks for FD 0 = Mask 1 = Pass | 0x1 |
| CLK_DRC | [1] | RW | Gating all clocks for DRC 0 = Mask 1 = Pass | 0x1 |
| CLK_ISP | [0] | RW | Gating all clocks for ISP 0 = Mask 1 = Pass | 0x1 |

5.9.1.155 CLK_GATE_IP_ISP1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8804, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| RSVD | [31:14] | – | Reserved | 0x3FFF |
| CLK_SPI1_ISP | [13] | RW | Gating all clocks for SPI1_ISP except SCLK 0 = Mask 1 = Pass | 0x1 |
| CLK_SPI0_ISP | [12] | RW | Gating all clocks for SPI0_ISP except SCLK 0 = Mask 1 = Pass | 0x1 |
| RSVD | [11:5] | – | Reserved | 0x7F |
| CLK_SMMU_ISPCX | [4] | RW | Gating all clocks for CLK_SMMU_ISPCX 0 = Mask 1 = Pass | 0x1 |
| RSVD | [3:1] | – | Reserved | 0x7 |
| CLK_ASYNCAXIM | [0] | RW | Gating all clocks for CLK_ASYNCAXIM 0 = Mask 1 = Pass | 0x1 |

5.9.1.156 CLKOUT_CMU_ISP

- Base Address: 0x1004_0000
- Address = Base Address + 0x8A00, Reset Value = 0x0001_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:17] | – | Reserved | 0x0 |
| ENB_CLKOUT | [16] | RW | Enable CLKOUT 0 = Disables 1 = Enables | 0x1 |
| RSVD | [15:14] | – | Reserved | 0x0 |
| DIV_RATIO | [13:8] | RW | Divide Ratio Divide ratio = DIV_RATIO + 1 | 0x0 |
| RSVD | [7:5] | – | Reserved | 0x0 |
| MUX_SEL | [4:0] | RW | MUX Selection 00000 = ACLK_MCUISP 00001 = PCLKDBG_MCUISP 00010 = ACLK_DIV0 00011 = ACLK_DIV1 00100 = SCLK_MPWM_ISP | 0x0 |

5.9.1.157 CLKOUT_CMU_ISP_DIV_STAT

- Base Address: 0x1004_0000
- Address = Base Address + 0x8A04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:1] | - | Reserved | 0x0 |
| DIV_STAT | [0] | R | DIVCLKOUT Status 0 = Stable 1 = Status that the divider is changing | 0x0 |

5.9.1.158 CMU_ISP_SPARE0

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B00, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|------------------------|-------------|
| SPARE | [31:0] | RW | CMU_ISP Spare Register | 0x0 |

5.9.1.159 CMU_ISP_SPARE1

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B04, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|------------------------|-------------|
| SPARE | [31:0] | RW | CMU_ISP Spare Register | 0x0 |

5.9.1.160 CMU_ISP_SPARE2

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B08, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|------------------------|-------------|
| SPARE | [31:0] | RW | CMU_ISP Spare Register | 0x0 |

5.9.1.161 CMU_ISP_SPARE3

- Base Address: 0x1004_0000
- Address = Base Address + 0x8B0C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|------------------------|-------------|
| SPARE | [31:0] | RW | CMU_ISP Spare Register | 0x0 |

6 Interrupt Controller

6.1 Overview

Generic Interrupt Controller (GIC) is a centralized resource that supports and manages interrupts in a system.

GIC provides:

- Registers for managing interrupt sources, interrupt behavior, and interrupt routing to one or multiple processors
- Support for
 - The ARM architecture Security Extensions
 - Enabling, disabling, and generating processor interrupts from hardware (peripheral) interrupt sources
 - Generating software interrupts
 - Interrupt masking and prioritization

GIC takes the interrupts asserted at the system level and sends appropriate signals to each connected processor. When GIC implements the Security Extensions, it can implement two interrupt requests to a connected processor. The architecture identifies these requests as IRQ and FIQ.

6.2 Features

The features of GIC are:

- Supports three interrupt types:
 - Software Generated Interrupt (SGI)
 - Private Peripheral Interrupt (PPI)
 - Shared Peripheral Interrupt (SPI)
- Programmable interrupts that enable you to set the:
 - Security state for an interrupt.
 - Priority level of an interrupt.
 - Enabling or disabling of an interrupt.
 - Processors that receive an interrupt.

6.2.1 Security Extensions Support

The ARM GIC architecture Security Extensions support:

- Configuring each interrupt as either Secure or Non-secure
- Signaling Secure interrupts to the target processor by using either the IRQ or FIQ exception request
- Handling priority of secure and Non-secure interrupts, which is a unified scheme.
- Optional lockdown of the configuration of some Secure interrupts.

In an implementation that includes the Security Extensions:

- System software individually defines each implemented interrupt as either Secure or Non-secure.
- The behavior of processor accesses to registers in the GIC depends on whether the access is Secure or Non-secure. When accessing GIC registers:
 - A Non-secure read of a register field that holds state information for a Secure interrupt returns zero
 - GIC ignores any Non-secure write to a register field that holds state information for a secure interrupt.

Non-secure accesses can only read or set information corresponding to Non-secure interrupts. Secure accesses can read or set information corresponding to both Non-secure and Secure interrupts.

- A Non-secure interrupt signals an IRQ interrupt request to a target processor.
- A Secure interrupt can signal either an IRQ or FIQ interrupt request to a target processor.

6.2.2 Implementation-Specific Configurable Features

During implementation of GIC, the features that depend on the configuration are:

- Exynos 4412 GIC Configuration
- Total 160 interrupts including Software Generated Interrupts (SGIs), Private Peripheral Interrupts (PPIs) and Shared Peripheral Interrupts (SPIs) are supported.
- For SPI, you can service maximal $32 \times 4 = 128$ interrupt requests.

[Table 6-1](#) describes the GIC configuration values.

Table 6-1 GIC Configuration Values

| Items | Configuration Values |
|-------------------------------------|---|
| AMBA Protocol | AXI |
| Software Generated Interrupts (SGI) | 16 |
| Private Peripheral Interrupts (PPI) | 8 |
| Shared Peripheral Interrupts (SPI) | 128 |
| Priority Level | 256 |
| Legacy interrupt Support | No |
| Number of CPUs | 2 |
| CPU Interface AXI ID Width | 10 |
| Distributer AXI ID Width | 10 |
| Security Domains | 2 (Supports TrustZone technology) |
| Lockable SPIs | 31 |
| Legacy dialog | – (Legacy interrupts are not used) |
| SGI Register Level Selection | 0xF (default value) |
| SPI Register Level Selection | 0x3FF (default value) |
| PPI Register Level Selection | 0xF (default value) |
| PPI sensitivity | ppi_cx[0] – ppi_cx[5]: edge ppi_cx[6] – ppi_cx[10]: level ppi_cx[11]: edge ppi_cx[12]: level ppi_cx[13] – ppi_cx[14]: edge ppi_cx[15]: level |
| PPI Registering | Synchronized (for all PPI) |
| SPI Registering | Synchronized (for all SPI) |

6.3 Interrupt Source

This section includes:

- Interrupt source connection
- GIC interrupt table

6.3.1 Interrupt Sources Connection

[Figure 6-1](#) illustrates the interrupt sources connection.

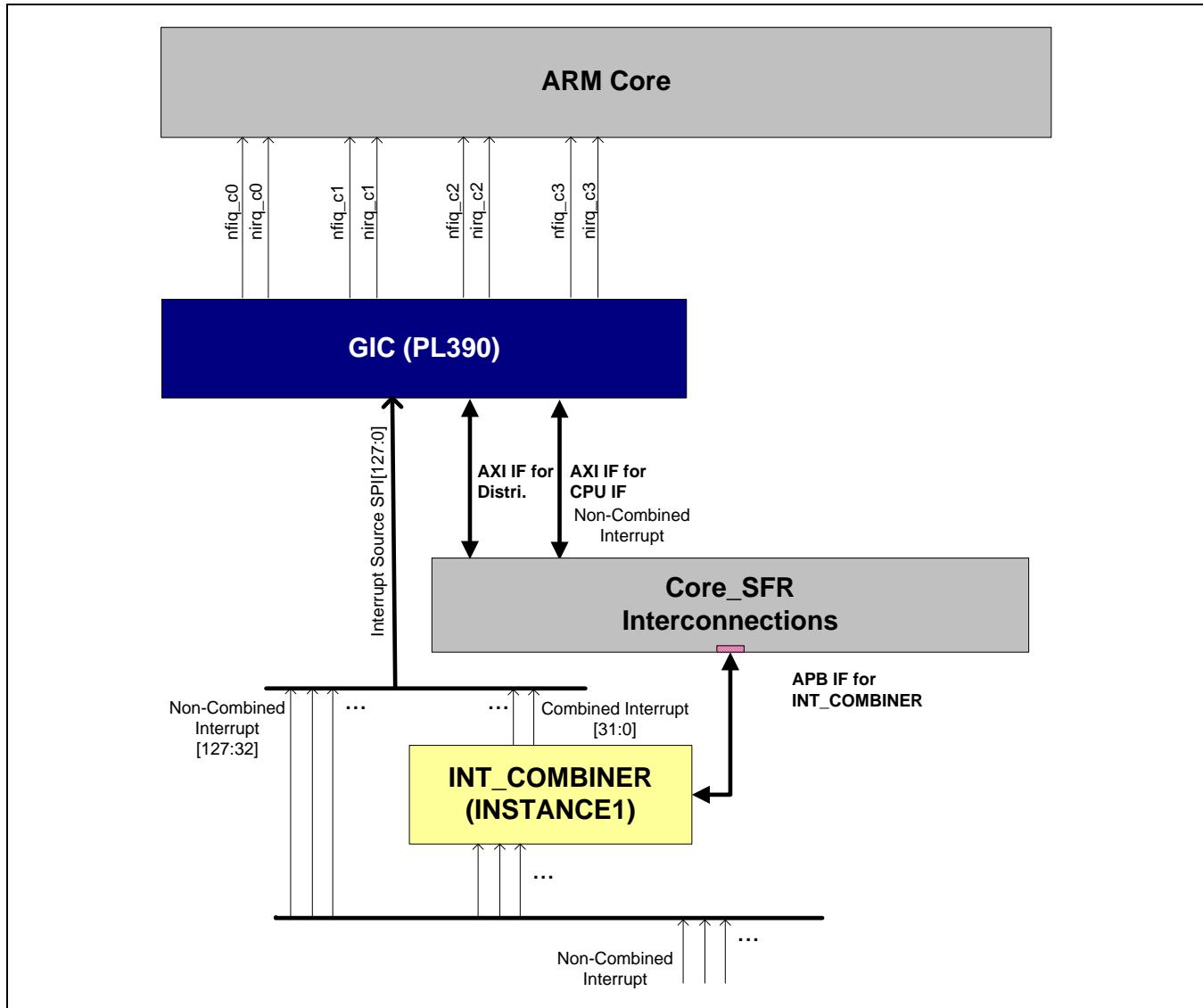


Figure 6-1 Interrupt Sources Connection

GIC interrupt sources are passed INT_COMBINER block that is combined interrupt sources for GIC.

6.3.2 GIC Interrupt Table

Total 160 interrupts including Software Generated Interrupts (SGIs[15:0], ID[15:0]), Private Peripheral Interrupts (PPIs[15:0], ID[31:16]) and Shared Peripheral Interrupts (SPIs[127:0], ID[159:32]) are supported. For SPI, you can service a maximal $32 \times 4 = 128$ interrupt requests.

[Table 6-2](#) describes the GIC interrupt (SPI[127:]).

Table 6-2 GIC Interrupt Table (SPI[127:0])

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|-----|----------------|----------------------|--------------|
| 127 | 159 | – | G3D_IRQGP | – |
| 126 | 158 | – | G3D_IRQPP3 | – |
| 125 | 157 | – | G3D_IRQPP2 | – |
| 124 | 156 | – | G3D_IRQPP1 | – |
| 123 | 155 | – | G3D_IRQPP0 | – |
| 122 | 154 | – | G3D_IRQGPMMU | – |
| 121 | 153 | – | G3D_IRQPPMMU3 | – |
| 120 | 152 | – | G3D_IRQPPMMU2 | – |
| 119 | 151 | – | G3D_IRQPPMMU1 | – |
| 118 | 150 | – | G3D_IRQPPMMU0 | – |
| 117 | 149 | – | G3D_IRQPMU | – |
| 116 | 148 | – | C2C_SSCM[1] | – |
| 115 | 147 | – | TSI | – |
| 114 | 146 | – | CEC | – |
| 113 | 145 | – | SLIMBUS | – |
| 112 | 144 | – | SSS | – |
| 111 | 143 | – | GPS | – |
| 110 | 142 | – | PMU | – |
| 109 | 141 | – | KEYPAD | – |
| 108 | 140 | IntG17_7 | L2_IRQ | MCT |
| | | IntG17_6 | Reserved | – |
| | | IntG17_5 | SYSMMU_ISP_CX[1] | – |
| | | IntG17_4 | SYSMMU_FIMC_FD[1] | – |
| | | IntG17_3 | SYSMMU_FIMC_DRC[1] | – |
| | | IntG17_2 | SYSMMU_FIMC_ISP[1] | – |
| | | IntG17_1 | SYSMMU_FIMC_Lite1[1] | – |
| | | IntG17_0 | SYSMMU_FIMC_Lite0[1] | – |
| 107 | 139 | IntG16_7 | L3_IRQ | MCT |
| | | IntG16_6 | Reserved | – |
| | | IntG16_5 | SYSMMU_ISP_CX[0] | – |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|-----|----------------|----------------------|--------------|
| | | IntG16_4 | SYSMMU_FIMC_FD[0] | – |
| | | IntG16_3 | SYSMMU_FIMC_DRC[0] | – |
| | | IntG16_2 | SYSMMU_FIMC_ISP[0] | – |
| | | IntG16_1 | SYSMMU_FIMC_Lite1[0] | – |
| | | IntG16_0 | SYSMMU_FIMC_Lite0[0] | – |
| 106 | 138 | – | FIMC_lite1 | – |
| 105 | 137 | – | FIMC_lite0 | – |
| 104 | 136 | – | SPDIF | – |
| 103 | 135 | – | PCM2 | – |
| 102 | 134 | – | PCM1 | – |
| 101 | 133 | – | PCM0 | – |
| 100 | 132 | – | AC97 | – |
| 99 | 131 | – | I2S2 | – |
| 98 | 130 | – | I2S1 | – |
| 97 | 129 | – | I2S0 | – |
| 96 | 128 | – | AUDIO_SS | – |
| 95 | 127 | – | ISP[1] | – |
| 94 | 126 | – | MFC | – |
| 93 | 125 | – | HDMI_I2C | – |
| 92 | 124 | – | HDMI | – |
| 91 | 123 | – | MIXER | – |
| 90 | 122 | – | ISP[0] | – |
| 89 | 121 | – | G2D | – |
| 88 | 120 | – | JPEG | – |
| 87 | 119 | – | FIMC3 | – |
| 86 | 118 | – | FIMC2 | – |
| 85 | 117 | – | FIMC1 | – |
| 84 | 116 | – | FIMC0 | – |
| 83 | 115 | – | ROTATOR | – |
| 82 | 114 | – | Reserved | – |
| 81 | 113 | – | Reserved | – |
| 80 | 112 | – | MIPI_CSI_2LANE | – |
| 79 | 111 | – | MIPI_DSI_4LANE | – |
| 78 | 110 | – | MIPI_CSI_4LANE | – |
| 77 | 109 | – | SDMMC | – |
| 76 | 108 | – | HSMMC3 | – |
| 75 | 107 | – | HSMMC2 | – |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|-----|----------------|------------------|--------------------------------|
| 74 | 106 | – | HSMMC1 | – |
| 73 | 105 | – | HSMMC0 | – |
| 72 | 104 | – | GPIO_C2C | – |
| 71 | 103 | – | HSOTG | – |
| 70 | 102 | – | UHOST | USB HOST |
| 69 | 101 | – | G1_IRQ | MCT |
| 68 | 100 | – | SPI2 | – |
| 67 | 99 | – | SPI1 | – |
| 66 | 98 | – | SPI0 | – |
| 65 | 97 | – | I2C7 | – |
| 64 | 96 | – | I2C6 | – |
| 63 | 95 | – | I2C5 | – |
| 62 | 94 | – | I2C4 | – |
| 61 | 93 | – | I2C3 | – |
| 60 | 92 | – | I2C2 | – |
| 59 | 91 | – | I2C1 | – |
| 58 | 90 | – | I2C0 | – |
| 57 | 89 | – | G0_IRQ | – |
| 56 | 88 | – | Reserved | – |
| 55 | 87 | – | UART3 | – |
| 54 | 86 | – | UART2 | – |
| 53 | 85 | – | UART1 | – |
| 52 | 84 | – | UART0 | – |
| 51 | 83 | – | NFC | – |
| 50 | 82 | – | IEM_IEC | – |
| 49 | 81 | – | IEM_APIC | – |
| 48 | 80 | IntG18_7 | Reserved | – |
| | | IntG18_6 | CPU_nIRQOUT[2] | – |
| | | IntG18_5 | PARITYFAILSCU[2] | Parity fails for SCU from CPU2 |
| | | IntG18_4 | PARITYFAIL2 | L1 parity fails for CPU2 |
| | | IntG18_3 | nCTIIRQ[2] | F4Q CTI interrupt for CPU2 |
| | | IntG18_2 | PMUIRQ[2] | F4Q PMU interrupt from CPU2 |
| | | IntG18_1 | Reserved | – |
| | | IntG18_0 | L1_IRQ | MCT |
| 47 | 79 | – | GPIO_LB | – |
| 46 | 78 | – | GPIO_RT | – |
| 45 | 77 | – | RTC_TIC | – |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|----|----------------|------------------|--------------------------------|
| 44 | 76 | – | RTC_ALARM | – |
| 43 | 75 | – | WDT | – |
| 42 | 74 | IntG19_7 | Reserved | – |
| | | IntG19_6 | CPU_nIRQOUT[3] | – |
| | | IntG19_5 | PARITYFAILSCU[3] | Parity fails for SCU from CPU3 |
| | | IntG19_4 | PARITYFAIL3 | L1 parity fails for CPU3 |
| | | IntG19_3 | nCTIIRQ[3] | F4Q CTI interrupt for CPU3 |
| | | IntG19_2 | PMUIRQ[3] | F4Q PMU interrupt from CPU3 |
| | | IntG19_1 | Reserved | – |
| | | IntG19_0 | L0_IRQ | MCT |
| 41 | 73 | – | TIMER4 | – |
| 40 | 72 | – | TIMER3 | – |
| 39 | 71 | – | TIMER2 | – |
| 38 | 70 | – | TIMER1 | – |
| 37 | 69 | – | TIMER0 | – |
| 36 | 68 | – | PDMA1 | – |
| 35 | 67 | – | PDMA0 | – |
| 34 | 66 | – | MDMA | – |
| 33 | 65 | – | C2C_SSCM[0] | – |
| 32 | 64 | – | EINT16_31 | External Interrupt |
| 31 | 63 | – | EINT[15] | External Interrupt |
| 30 | 62 | – | EINT[14] | External Interrupt |
| 29 | 61 | – | EINT[13] | External Interrupt |
| 28 | 60 | – | EINT[12] | External Interrupt |
| 27 | 59 | – | EINT[11] | External Interrupt |
| 26 | 58 | – | EINT[10] | External Interrupt |
| 25 | 57 | – | EINT[9] | External Interrupt |
| 24 | 56 | – | EINT[8] | External Interrupt |
| 23 | 55 | – | EINT[7] | External Interrupt |
| 22 | 54 | – | EINT[6] | External Interrupt |
| 21 | 53 | – | EINT[5] | External Interrupt |
| 20 | 52 | – | EINT[4] | External Interrupt |
| 19 | 51 | – | EINT[3] | External Interrupt |
| 18 | 50 | – | EINT[2] | External Interrupt |
| 17 | 49 | – | EINT[1] | External Interrupt |
| 16 | 48 | – | EINT[0] | External Interrupt |
| 15 | 47 | IntG15_7 | DECERRINTR | F4D |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|----|----------------|------------------|--------------|
| | | IntG15_6 | SLVERRINTR | F4D |
| | | IntG15_5 | ERRRDINTR | F4D |
| | | IntG15_4 | ERRRTINTR | F4D |
| | | IntG15_3 | ERRWDINTR | F4D |
| | | IntG15_2 | ERRWTINTR | F4D |
| | | IntG15_1 | ECNTRINTR | F4D |
| | | IntG15_0 | SCUEVABORT | F4D |
| 14 | 46 | IntG14_6 | CPU_nIRQOUT[1] | F4D |
| | | IntG14_5 | Reserved | — |
| | | IntG14_4 | Reserved | — |
| | | IntG14_3 | Reserved | — |
| | | IntG14_2 | Reserved | — |
| | | IntG14_1 | Reserved | — |
| | | IntG14_0 | Reserved | — |
| 13 | 45 | IntG13_5 | CPU_nIRQOUT[0] | F4D |
| | | IntG13_4 | Reserved | — |
| | | IntG13_3 | Reserved | — |
| | | IntG13_2 | Reserved | — |
| | | IntG13_1 | Reserved | — |
| | | IntG13_0 | Reserved | — |
| 12 | 44 | IntG12_7 | G3 | MCT |
| | | IntG12_6 | G2 | |
| | | IntG12_5 | G1 | |
| | | IntG12_4 | G0 | |
| | | IntG12_3 | Reserved | — |
| | | IntG12_2 | Reserved | — |
| | | IntG12_1 | MIPI_HSI | MIPI |
| | | IntG12_0 | UART4 | UART |
| 11 | 43 | IntG11_3 | LCD0[3] | — |
| | | IntG11_2 | LCD0[2] | — |
| | | IntG11_1 | LCD0[1] | — |
| | | IntG11_0 | LCD0[0] | — |
| 10 | 42 | IntG10_7 | DMC1_PPC_PEREV_M | DMC1 |
| | | IntG10_6 | DMC1_PPC_PEREV_A | DMC1 |
| | | IntG10_5 | DMC0_PPC_PEREV_M | DMC0 |
| | | IntG10_4 | DMC0_PPC_PEREV_A | DMC0 |
| | | IntG10_3 | ADC | General ADC |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|----|----------------|-------------------|------------------------|
| | | IntG10_2 | L2CACHE | F4D |
| | | IntG10_1 | RP_TIMER | – |
| | | IntG10_0 | GPIO_AUDIO | – |
| 9 | 41 | IntG9_7 | PPMU_ISP_X | PPMU for ISP X |
| | | IntG9_6 | PPMU_MFC_M1 | PPMU for MFC_M1 |
| | | IntG9_5 | PPMU_MFC_M0 | PPMU for MFC_M0 |
| | | IntG9_4 | PPMU_3D | PPMU for 3D |
| | | IntG9_3 | PPMU_TV_M0 | PPMU for TV_M0 |
| | | IntG9_2 | PPMU_FILE_D_M0 | PPMU for FILE_D_M0 |
| | | IntG9_1 | PPMU_ISP_MX | PPMU for ISP MX |
| | | IntG9_0 | PPMU_LCD0 | PPMU for LCD0 |
| 8 | 40 | IntG8_7 | PPMU_IMAGE_M0 | PPMU for IMAGE_M0 |
| | | IntG8_6 | PPMU_CAMIF_M0 | PPMU for CAMIF_M0 |
| | | IntG8_5 | PPMU_D_RIGHT_M0 | PPMU for D_right_M0 |
| | | IntG8_4 | PPMU_D_LEFT_M0 | PPMU for D_left_M0 |
| | | IntG8_3 | PPMU_ACP0_M0 | PPMU for ACP0_M0 |
| | | IntG8_2 | PPMU_XIU_R_S1 | PPMU for XIU_R_S1 |
| | | IntG8_1 | PPMU_XIU_R | PPMU for XIU_R |
| | | IntG8_0 | PPMU_XIU_L | PPMU for XIU_L |
| 7 | 39 | IntG7_7 | Reserved | – |
| | | IntG7_6 | SYSMMU_MFC_M1[1] | System MMU for MFC_M1 |
| | | IntG7_5 | SYSMMU_MFC_M0[1] | System MMU for MFC_M0 |
| | | IntG7_4 | SYSMMU_TV_M0[1] | System MMU for TV_M0 |
| | | IntG7_3 | Reserved | – |
| | | IntG7_2 | SYSMMU_LCD0_M0[1] | System MMU for LCD0_M0 |
| | | IntG7_1 | SYSMMU_GPS[1] | System MMU for GPS |
| | | IntG7_0 | SYSMMU_ROTATOR[1] | System MMU for Rotator |
| 6 | 38 | IntG6_7 | SYSMMU_2D[1] | System MMU for 2D |
| | | IntG6_6 | SYSMMU_JPEG[1] | System MMU for JPEG |
| | | IntG6_5 | SYSMMU_FIMC3[1] | System MMU for FIMC3 |
| | | IntG6_4 | SYSMMU_FIMC2[1] | System MMU for FIMC2 |
| | | IntG6_3 | SYSMMU_FIMC1[1] | System MMU for FIMC1 |
| | | IntG6_2 | SYSMMU_FIMC0[1] | System MMU for FIMC0 |
| | | IntG6_1 | SYSMMU_SSS[1] | System MMU for SSS |
| | | IntG6_0 | SYSMMU_MDMA[1] | System MMU for MDMA |
| 5 | 37 | IntG5_7 | Reserved | – |
| | | IntG5_6 | SYSMMU_MFC_M1[0] | System MMU for MFC_M1 |

| SPI Port No | ID | Int_I_Combiner | Interrupt Source | Source Block |
|-------------|----|----------------|-------------------|--------------------------------|
| | | IntG5_5 | SYSMMU_MFC_M0[0] | System MMU for MFC_M0 |
| | | IntG5_4 | SYSMMU_TV_M0[0] | System MMU for TV_M0 |
| | | IntG5_3 | Reserved | — |
| | | IntG5_2 | SYSMMU_LCD0_M0[0] | System MMU for LCD0_M0 |
| | | IntG5_1 | SYSMMU_GPS[0] | System MMU for GPS |
| | | IntG5_0 | SYSMMU_ROTATOR[0] | System MMU for Rotator |
| 4 | 36 | IntG4_7 | SYSMMU_2D[0] | System MMU for 2D |
| | | IntG4_6 | SYSMMU_JPEG[0] | System MMU for JPEG |
| | | IntG4_5 | SYSMMU_FIMC3[0] | System MMU for FIMC3 |
| | | IntG4_4 | SYSMMU_FIMC2[0] | System MMU for FIMC2 |
| | | IntG4_3 | SYSMMU_FIMC1[0] | System MMU for FIMC1 |
| | | IntG4_2 | SYSMMU_FIMC0[0] | System MMU for FIMC0 |
| | | IntG4_1 | SYSMMU_SSS[0] | System MMU for SSS |
| | | IntG4_0 | SYSMMU_MDMA[0] | System MMU for MDMA |
| 3 | 35 | IntG3_6 | nCTIIRQ_ISP | ISP CTI interrupt |
| | | IntG3_5 | PMUIRQ_ISP | ISP PMU interrupt |
| | | IntG3_4 | TMU | — |
| | | IntG3_3 | nCTIIRQ[1] | F4D CTI interrupt for CPU1 |
| | | IntG3_2 | PMUIRQ[1] | F4D PMU interrupt from CPU1 |
| | | IntG3_1 | PARITYFAILSCU[1] | Parity fails for SCU from CPU1 |
| | | IntG3_0 | PARITYFAIL1 | L1 parity fails for CPU1 |
| 2 | 34 | IntG2_6 | PARRINTR | Parity error on L2 tag RAM |
| | | IntG2_5 | PARRDINTR | Parity error on L2 data RAM |
| | | IntG2_4 | TMU | — |
| | | IntG2_3 | nCTIIRQ[0] | F4D CTI interrupt for CPU0 |
| | | IntG2_2 | PMUIRQ[0] | F4D PMU interrupt from CPU0 |
| | | IntG2_1 | PARITYFAILSCU[0] | Parity fails for SCU from CPU0 |
| | | IntG2_0 | PARITYFAIL0 | L1 parity fails for CPU0 |
| 1 | 33 | IntG1_3 | TZASC1[1] | — |
| | | IntG1_2 | TZASC1[0] | — |
| | | IntG1_1 | TZASCO[1] | — |
| | | IntG1_0 | TZASCO[0] | — |
| 0 | 32 | IntG0_3 | MDNIE_LCD0[3] | — |
| | | IntG0_2 | MDNIE_LCD0[2] | — |
| | | IntG0_1 | MDNIE_LCD0[1] | — |
| | | IntG0_0 | MDNIE_LCD0[0] | — |

[Table 6-3](#) describes the GIC interrupt (PPI[15:0]).

Table 6-3 GIC Interrupt Table (PPI[15:0])

| PPI Port No | ID | Interrupt Source | Source Block |
|-------------|----|---|--------------|
| 15 | 31 | Reserved | – |
| 14 | 30 | Reserved | – |
| 13 | 29 | Reserved | – |
| 12 | 28 | L3_IRQ (for CPU3) or L2_IRQ (for CPU2) or L1_IRQ (for CPU1) or L0_IRQ (for CPU0) | MCT |
| 11 | 27 | Reserved | – |
| 10 | 26 | G3_IRQ (for CPU3) or G2_IRQ (for CPU2) or G1_IRQ (for CPU1) or G0_IRQ (for CPU0) | MCT |
| 9 | 25 | Reserved | – |
| 8 | 24 | Reserved | – |
| 7 | 23 | Reserved | – |
| 6 | 22 | Reserved | – |
| 5 | 21 | Reserved | – |
| 4 | 20 | Reserved | – |
| 3 | 19 | Reserved | – |
| 2 | 18 | Reserved | – |
| 1 | 17 | Reserved | – |
| 0 | 16 | Reserved | – |

6.4 Functional Overview

Please refer to the GIC PL390 technical reference manual.

6.5 Register Description

6.5.1 Register Map Summary

- Base Address: 0x1048_0000

| Register | Offset | Description | Reset Value |
|--------------------|--------|---------------------------------------|-------------|
| ICCICR_CPU0 | 0x0000 | CPU interface control register | 0x0000_0000 |
| ICCPMR_CPU0 | 0x0004 | Interrupt priority mask register | 0x0000_0000 |
| ICCBPR_CPU0 | 0x0008 | Binary point register | 0x0000_0000 |
| ICCIAR_CPU0 | 0x000C | Interrupt acknowledge register | 0x0000_03FF |
| ICCEOIR_CPU0 | 0x0010 | End of interrupt register | Undefined |
| ICCRPR_CPU0 | 0x0014 | Running priority register | 0x0000_00FF |
| ICCHPIR_CPU0 | 0x0018 | Highest pending interrupt register | 0x0000_03FF |
| ICCABPR_CPU0 | 0x001C | Aliased binary point register | 0x0000_0000 |
| INTEG_EN_C_CPU0 | 0x0040 | Integration test enable register | 0x0000_0000 |
| INTERRUPT_OUT_CPU0 | 0x0044 | Interrupt output register | 0x0000_0000 |
| ICCIIDR | 0x00FC | CPU interface identification register | 0x3901_043B |
| ICCICR_CPU1 | 0x4000 | CPU interface control register | 0x0000_0000 |
| ICCPMR_CPU1 | 0x4004 | Interrupt priority mask register | 0x0000_0000 |
| ICCBPR_CPU1 | 0x4008 | Binary point register | 0x0000_0000 |
| ICCIAR_CPU1 | 0x400C | Interrupt acknowledge register | 0x0000_03FF |
| ICCEOIR_CPU1 | 0x4010 | End of interrupt register | Undefined |
| ICCRPR_CPU1 | 0x4014 | Running priority register | 0x0000_00FF |
| ICCHPIR_CPU1 | 0x4018 | Highest pending interrupt register | 0x0000_03FF |
| ICCABPR_CPU1 | 0x401C | Aliased binary point register | 0x0000_0000 |
| INTEG_C_EN_CPU1 | 0x4040 | Integration test enable register | 0x0000_0000 |
| INTERRUPT_OUT_CPU1 | 0x4044 | Interrupt output register | 0x0000_0000 |
| ICCICR_CPU2 | 0x8000 | CPU interface control register | 0x0000_0000 |
| ICCPMR_CPU2 | 0x8004 | Interrupt priority mask register | 0x0000_0000 |
| ICCBPR_CPU2 | 0x8008 | Binary point register | 0x0000_0000 |
| ICCIAR_CPU2 | 0x800C | Interrupt acknowledge register | 0x0000_03FF |
| ICCEOIR_CPU2 | 0x8010 | End of interrupt register | Undefined |
| ICCRPR_CPU2 | 0x8014 | Running priority register | 0x0000_00FF |
| ICCHPIR_CPU2 | 0x8018 | Highest pending interrupt register | 0x0000_03FF |
| ICCABPR_CPU2 | 0x801C | Aliased binary point register | 0x0000_0000 |
| INTEG_C_EN_CPU2 | 0x8040 | Integration test enable register | 0x0000_0000 |
| INTERRUPT_OUT_CPU2 | 0x8044 | Interrupt output register | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|--------------------|--------|------------------------------------|-------------|
| ICCICR_CPU3 | 0xC000 | CPU interface control register | 0x0000_0000 |
| ICCPMR_CPU3 | 0xC004 | Interrupt priority mask register | 0x0000_0000 |
| ICCBPR_CPU3 | 0xC008 | Binary point register | 0x0000_0000 |
| ICCIAR_CPU3 | 0xC00C | Interrupt acknowledge register | 0x0000_03FF |
| ICCEOIR_CPU3 | 0xC010 | End of interrupt register | Undefined |
| ICCRPR_CPU3 | 0xC014 | Running priority register | 0x0000_00FF |
| ICCHPIR_CPU3 | 0xC018 | Highest pending interrupt register | 0x0000_03FF |
| ICCABPR_CPU3 | 0xC01C | Aliased binary point register | 0x0000_0000 |
| INTEG_C_EN_CPU3 | 0xC040 | Integration test enable register | 0x0000_0000 |
| INTERRUPT_OUT_CPU3 | 0xC044 | Interrupt output register | 0x0000_0000 |

- Base Address: 0x1049_0000

| Register | Offset | Description | Reset Value |
|---------------|--------|---|-------------|
| ICDDCR | 0x0000 | Distributor control register | 0x0000_0000 |
| ICDICTR | 0x0004 | Interrupt controller type register | 0x0000_FC24 |
| ICDIIDR | 0x0008 | Distributor implementer identification register | 0x0000_043B |
| ICDISR0_CPU0 | 0x0080 | Interrupt security registers (SGI,PPI) | 0x0000_0000 |
| ICDISR1 | 0x0084 | Interrupt security registers (SPI[31:0]) | 0x0000_0000 |
| ICDISR2 | 0x0088 | Interrupt security registers (SPI[63:32]) | 0x0000_0000 |
| ICDISR3 | 0x008C | Interrupt security registers (SPI[95:64]) | 0x0000_0000 |
| ICDISR4 | 0x0090 | Interrupt security registers (SPI[127:96]) | 0x0000_0000 |
| ICDISER0_CPU0 | 0x0100 | Interrupt set-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDISER1 | 0x0104 | Interrupt set-enable register (SPI[31:0]) | 0x0000_0000 |
| ICDISER2 | 0x0108 | Interrupt set-enable register (SPI[63:32]) | 0x0000_0000 |
| ICDISER3 | 0x010C | Interrupt set-enable register (SPI[95:64]) | 0x0000_0000 |
| ICDISER4 | 0x0110 | Interrupt set-enable register (SPI[127:96]) | 0x0000_0000 |
| ICDICER0_CPU0 | 0x0180 | Interrupt clear-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDICER1 | 0x0184 | Interrupt clear-enable register (SPI[31:0]) | 0x0000_0000 |
| ICDICER2 | 0x0188 | Interrupt clear-enable register (SPI[63:32]) | 0x0000_0000 |
| ICDICER3 | 0x018C | Interrupt clear-enable register (SPI[95:64]) | 0x0000_0000 |
| ICDICER4 | 0x0190 | Interrupt clear-enable register (SPI[127:96]) | 0x0000_0000 |
| ICDISPR0_CPU0 | 0x0200 | Interrupt pending-set register (SGI,PPI) | 0x0000_0000 |
| ICDISPR1 | 0x0204 | Interrupt pending-set register (SPI[31:0]) | 0x0000_0000 |
| ICDISPR2 | 0x0208 | Interrupt pending-set register (SPI[63:32]) | 0x0000_0000 |
| ICDISPR3 | 0x020C | Interrupt pending-set register (SPI[95:64]) | 0x0000_0000 |
| ICDISPR4 | 0x0210 | Interrupt pending-set register (SPI[127:96]) | 0x0000_0000 |
| ICDICPR0_CPU0 | 0x0280 | Interrupt pending-clear register (SGI,PPI) | 0x0000_0000 |
| ICDICPR1 | 0x0284 | Interrupt pending-clear register (SPI[31:0]) | 0x0000_0000 |
| ICDICPR2 | 0x0288 | Interrupt pending-clear register (SPI[63:32]) | 0x0000_0000 |
| ICDICPR3 | 0x028C | Interrupt pending-clear register (SPI[95:64]) | 0x0000_0000 |
| ICDICPR4 | 0x0290 | Interrupt pending-clear register(SPI[127:96]) | 0x0000_0000 |
| ICDABR0_CPU0 | 0x0300 | Active bit register (SGI, PPI) | 0x0000_0000 |
| ICDABR1 | 0x0304 | Active bit register (SPI[31:0]) | 0x0000_0000 |
| ICDABR2 | 0x0308 | Active bit register (SPI[63:32]) | 0x0000_0000 |
| ICDABR3 | 0x030C | Active bit register (SPI[95:64]) | 0x0000_0000 |
| ICDABR4 | 0x0310 | Active bit register (SPI[127:96]) | 0x0000_0000 |
| ICDIPR0_CPU0 | 0x0400 | Priority level register (SGI[3:0]) | 0x0000_0000 |
| ICDIPR1_CPU0 | 0x0404 | Priority level register (SGI[7:4]) | 0x0000_0000 |
| ICDIPR2_CPU0 | 0x0408 | Priority level register (SGI[11:8]) | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|--------------|--------|--|-------------|
| ICDIPR3_CPU0 | 0x040C | Priority level register (SGI[15:12]) | 0x0000_0000 |
| ICDIPR4_CPU0 | 0x0410 | Priority level register (PPI[3:0]) | 0x0000_0000 |
| ICDIPR5_CPU0 | 0x0414 | Priority level register (PPI[7:4]) | 0x0000_0000 |
| ICDIPR6_CPU0 | 0x0418 | Priority level register (PPI[11:8]) | 0x0000_0000 |
| ICDIPR7_CPU0 | 0x041C | Priority level register (PPI[15:12]) | 0x0000_0000 |
| ICDIPR8 | 0x0420 | Priority level register (SPI[3:0]) | 0x0000_0000 |
| ICDIPR9 | 0x0424 | Priority level register (SPI[7:4]) | 0x0000_0000 |
| ICDIPR10 | 0x0428 | Priority level register (SPI[11:8]) | 0x0000_0000 |
| ICDIPR11 | 0x042C | Priority level register (SPI[15:12]) | 0x0000_0000 |
| ICDIPR12 | 0x0430 | Priority level register (SPI[19:16]) | 0x0000_0000 |
| ICDIPR13 | 0x0434 | Priority level register (SPI[23:20]) | 0x0000_0000 |
| ICDIPR14 | 0x0438 | Priority level register (SPI[27:24]) | 0x0000_0000 |
| ICDIPR15 | 0x043C | Priority level register (SPI[31:28]) | 0x0000_0000 |
| ICDIPR16 | 0x0440 | Priority level register (SPI[35:32]) | 0x0000_0000 |
| ICDIPR17 | 0x0444 | Priority level register (SPI[39:36]) | 0x0000_0000 |
| ICDIPR18 | 0x0448 | Priority level register (SPI[43:40]) | 0x0000_0000 |
| ICDIPR19 | 0x044C | Priority level register (SPI[47:44]) | 0x0000_0000 |
| ICDIPR20 | 0x0450 | Priority level register (SPI[51:48]) | 0x0000_0000 |
| ICDIPR21 | 0x0454 | Priority level register (SPI[55:52]) | 0x0000_0000 |
| ICDIPR22 | 0x0458 | Priority level register (SPI[59:56]) | 0x0000_0000 |
| ICDIPR23 | 0x045C | Priority level register (SPI[63:60]) | 0x0000_0000 |
| ICDIPR24 | 0x0460 | Priority level register (SPI[67:64]) | 0x0000_0000 |
| ICDIPR25 | 0x0464 | Priority level register (SPI[71:68]) | 0x0000_0000 |
| ICDIPR26 | 0x0468 | Priority level register (SPI[75:72]) | 0x0000_0000 |
| ICDIPR27 | 0x046C | Priority level register (SPI[79:76]) | 0x0000_0000 |
| ICDIPR28 | 0x0470 | Priority level register (SPI[83:80]) | 0x0000_0000 |
| ICDIPR29 | 0x0474 | Priority level register (SPI[87:84]) | 0x0000_0000 |
| ICDIPR30 | 0x0478 | Priority level register (SPI[91:98]) | 0x0000_0000 |
| ICDIPR31 | 0x047C | Priority level register (SPI[95:92]) | 0x0000_0000 |
| ICDIPR32 | 0x0480 | Priority level register (SPI[99:96]) | 0x0000_0000 |
| ICDIPR33 | 0x0484 | Priority level register (SPI[103:100]) | 0x0000_0000 |
| ICDIPR34 | 0x0488 | Priority level register (SPI[107:104]) | 0x0000_0000 |
| ICDIPR35 | 0x048C | Priority level register (SPI[111:108]) | 0x0000_0000 |
| ICDIPR36 | 0x0490 | Priority level register (SPI[115:112]) | 0x0000_0000 |
| ICDIPR37 | 0x0494 | Priority level register (SPI[119:116]) | 0x0000_0000 |
| ICDIPR38 | 0x0498 | Priority level register (SPI[123:120]) | 0x0000_0000 |
| ICDIPR39 | 0x049C | Priority level register (SPI[127:124]) | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-----------------|---------------|---|--------------------|
| ICDIPTR0_CPU0 | 0x0800 | Processor targets register (SGI[3:0]) | 0x0101_0101 |
| ICDIPTR1_CPU0 | 0x0804 | Processor targets register (SGI[7:4]) | 0x0101_0101 |
| ICDIPTR2_CPU0 | 0x0808 | Processor targets register (SGI[11:8]) | 0x0101_0101 |
| ICDIPTR3_CPU0 | 0x080C | Processor targets register (SGI[15:12]) | 0x0101_0101 |
| ICDIPTR4_CPU0 | 0x0810 | Processor targets register (PPI[3:0]) | 0x0101_0101 |
| ICDIPTR5_CPU0 | 0x0814 | Processor targets register (PPI[7:4]) | 0x0101_0101 |
| ICDIPTR6_CPU0 | 0x0818 | Processor targets register (PPI[11:8]) | 0x0101_0101 |
| ICDIPTR7_CPU0 | 0x081C | Processor targets register (PPI[15:12]) | 0x0101_0101 |
| ICDIPTR8 | 0x0820 | Processor targets register (SPI[3:0]) | 0x0000_0000 |
| ICDIPTR9 | 0x0824 | Processor targets register (SPI[7:4]) | 0x0000_0000 |
| ICDIPTR10 | 0x0828 | Processor targets register (SPI[11:8]) | 0x0000_0000 |
| ICDIPTR11 | 0x082C | Processor targets register (SPI[15:12]) | 0x0000_0000 |
| ICDIPTR12 | 0x0830 | Processor targets register (SPI[19:16]) | 0x0000_0000 |
| ICDIPTR13 | 0x0834 | Processor targets register (SPI[23:20]) | 0x0000_0000 |
| ICDIPTR14 | 0x0838 | Processor targets register (SPI[27:24]) | 0x0000_0000 |
| ICDIPTR15 | 0x083C | Processor targets register (SPI[31:28]) | 0x0000_0000 |
| ICDIPTR16 | 0x0840 | Processor targets register (SPI[35:32]) | 0x0000_0000 |
| ICDIPTR17 | 0x0844 | Processor targets register (SPI[39:36]) | 0x0000_0000 |
| ICDIPTR18 | 0x0848 | Processor targets register (SPI[43:40]) | 0x0000_0000 |
| ICDIPTR19 | 0x084C | Processor targets register (SPI[47:44]) | 0x0000_0000 |
| ICDIPTR20 | 0x0850 | Processor targets register (SPI[51:48]) | 0x0000_0000 |
| ICDIPTR21 | 0x0854 | Processor targets register (SPI[55:52]) | 0x0000_0000 |
| ICDIPTR22 | 0x0858 | Processor targets register (SPI[59:56]) | 0x0000_0000 |
| ICDIPTR23 | 0x085C | Processor targets register (SPI[63:60]) | 0x0000_0000 |
| ICDIPTR24 | 0x0860 | Processor targets register (SPI[67:64]) | 0x0000_0000 |
| ICDIPTR25 | 0x0864 | Processor targets register (SPI[71:68]) | 0x0000_0000 |
| ICDIPTR26 | 0x0868 | Processor targets register (SPI[75:72]) | 0x0000_0000 |
| ICDIPTR27 | 0x086C | Processor targets register (SPI[79:76]) | 0x0000_0000 |
| ICDIPTR28 | 0x0870 | Processor targets register (SPI[83:80]) | 0x0000_0000 |
| ICDIPTR29 | 0x0874 | Processor targets register (SPI[87:84]) | 0x0000_0000 |
| ICDIPTR30 | 0x0878 | Processor targets register (SPI[91:98]) | 0x0000_0000 |
| ICDIPTR31 | 0x087C | Processor targets register (SPI[95:92]) | 0x0000_0000 |
| ICDIPTR32 | 0x0880 | Processor targets register (SPI[99:96]) | 0x0000_0000 |
| ICDIPTR33 | 0x0884 | Processor targets register (SPI[103:100]) | 0x0000_0000 |
| ICDIPTR34 | 0x0888 | Processor targets register (SPI[107:104]) | 0x0000_0000 |
| ICDIPTR35 | 0x088C | Processor targets register (SPI[111:108]) | 0x0000_0000 |
| ICDIPTR36 | 0x0890 | Processor targets register (SPI[115:112]) | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-----------------|--------|---|-------------|
| ICDIPTR37 | 0x0894 | Processor targets register (SPI[119:116]) | 0x0000_0000 |
| ICDIPTR38 | 0x0898 | Processor targets register (SPI[123:120]) | 0x0000_0000 |
| ICDIPTR39 | 0x089C | Processor targets register (SPI[127:124]) | 0x0000_0000 |
| ICDICFR0_CPU0 | 0x0C00 | Interrupt configuration register (SGI[15:0]) | 0xAAAA_AAAA |
| ICDICFR1_CPU0 | 0x0C04 | Interrupt configuration register (PPI[15:0]) | 0x7DD5_5FFF |
| ICDICFR2 | 0x0C08 | Interrupt configuration register (SPI[15:0]) | 0x5555_5555 |
| ICDICFR3 | 0x0C0C | Interrupt configuration register (SPI[31:16]) | 0x5555_5555 |
| ICDICFR4 | 0x0C10 | Interrupt configuration register (SPI[47:32]) | 0x5555_5555 |
| ICDICFR5 | 0x0C14 | Interrupt configuration register (SPI[63:48]) | 0x5555_5555 |
| ICDICFR6 | 0x0C18 | Interrupt configuration register (SPI[79:64]) | 0x5555_5555 |
| ICDICFR7 | 0x0C1C | Interrupt configuration register (SPI[95:80]) | 0x5555_5555 |
| ICDICFR8 | 0x0C20 | Interrupt configuration register (SPI[111:95]) | 0x5555_5555 |
| ICDICFR9 | 0x0C24 | Interrupt configuration register (SPI[127:112]) | 0x5555_5555 |
| PPI_STATUS_CPU0 | 0x0D00 | PPI status register | 0x0000_0000 |
| SPI_STATUS0 | 0x0D04 | SPI[31:0] status register | 0x0000_0000 |
| SPI_STATUS1 | 0x0D08 | SPI[63:32] status register | 0x0000_0000 |
| SPI_STATUS2 | 0x0D0C | SPI[95:64] status register | 0x0000_0000 |
| SPI_STATUS3 | 0x0D10 | SPI[127:96] status register | 0x0000_0000 |
| ICDSGIR | 0x0F00 | Software generated interrupt register | Undefined |
| ICDISR0_CPU1 | 0x4080 | Interrupt security registers (SGI,PPI) | 0x0000_0000 |
| ICDISER0_CPU1 | 0x4100 | Interrupt set-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDICER0_CPU1 | 0x4180 | Interrupt clear-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDISPR0_CPU1 | 0x4200 | Interrupt pending-set register (SGI,PPI) | 0x0000_0000 |
| ICDICPR0_CPU1 | 0x4280 | Interrupt pending-clear register (SGI,PPI) | 0x0000_0000 |
| ICDABR0_CPU1 | 0x4300 | Active status register (SGI, PPI) | 0x0000_0000 |
| ICDIPR0_CPU1 | 0x4400 | Priority level register (SGI[3:0]) | 0x0000_0000 |
| ICDIPR1_CPU1 | 0x4404 | Priority level register (SGI[7:4]) | 0x0000_0000 |
| ICDIPR2_CPU1 | 0x4408 | Priority level register (SGI[11:8]) | 0x0000_0000 |
| ICDIPR3_CPU1 | 0x440C | Priority level register (SGI[15:12]) | 0x0000_0000 |
| ICDIPR4_CPU1 | 0x4410 | Priority level register (PPI[3:0]) | 0x0000_0000 |
| ICDIPR5_CPU1 | 0x4414 | Priority level register (PPI[7:4]) | 0x0000_0000 |
| ICDIPR6_CPU1 | 0x4418 | Priority level register (PPI[11:8]) | 0x0000_0000 |
| ICDIPR7_CPU1 | 0x441C | Priority level register (PPI[15:12]) | 0x0000_0000 |
| ICDIPTR0_CPU1 | 0x4800 | Processor targets register (SGI[3:0]) | 0x0202_0202 |
| ICDIPTR1_CPU1 | 0x4804 | Processor targets register (SGI[7:4]) | 0x0202_0202 |
| ICDIPTR2_CPU1 | 0x4808 | Processor targets register (SGI[11:8]) | 0x0202_0202 |
| ICDIPTR3_CPU1 | 0x480C | Processor targets register (SGI[15:12]) | 0x0202_0202 |

| Register | Offset | Description | Reset Value |
|-----------------|--------|--|-------------|
| ICDIPTR4_CPU1 | 0x4810 | Processor targets register (PPI[3:0]) | 0x0202_0202 |
| ICDIPTR5_CPU1 | 0x4814 | Processor targets register (PPI[7:4]) | 0x0202_0202 |
| ICDIPTR6_CPU1 | 0x4818 | Processor targets register (PPI[11:8]) | 0x0202_0202 |
| ICDIPTR7_CPU1 | 0x481C | Processor targets register (PPI[15:12]) | 0x0202_0202 |
| ICDICFR0_CPU1 | 0x4C00 | Interrupt configuration register (SGI[15:0]) | 0xAAAA_AAAA |
| ICDICFR1_CPU1 | 0x4C04 | Interrupt configuration register (PPI[15:0]) | 0x7DD5_5FFF |
| PPI_STATUS_CPU1 | 0x4D00 | PPI status register | 0x0000_0000 |
| ICDISR0_CPU2 | 0x8080 | Interrupt security registers (SGI,PPI) | 0x0000_0000 |
| ICDISER0_CPU2 | 0x8100 | Interrupt set-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDICER0_CPU2 | 0x8180 | Interrupt clear-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDISPR0_CPU2 | 0x8200 | Interrupt pending-set register (SGI,PPI) | 0x0000_0000 |
| ICDICPR0_CPU2 | 0x8280 | Interrupt pending-clear register (SGI,PPI) | 0x0000_0000 |
| ICDABR0_CPU2 | 0x8300 | Active status register (SGI, PPI) | 0x0000_0000 |
| ICDIPR0_CPU2 | 0x8400 | Priority level register (SGI[3:0]) | 0x0000_0000 |
| ICDIPR1_CPU2 | 0x8404 | Priority level register (SGI[7:4]) | 0x0000_0000 |
| ICDIPR2_CPU2 | 0x8408 | Priority level register (SGI[11:8]) | 0x0000_0000 |
| ICDIPR3_CPU2 | 0x840C | Priority level register (SGI[15:12]) | 0x0000_0000 |
| ICDIPR4_CPU2 | 0x8410 | Priority level register (PPI[3:0]) | 0x0000_0000 |
| ICDIPR5_CPU2 | 0x8414 | Priority level register (PPI[7:4]) | 0x0000_0000 |
| ICDIPR6_CPU2 | 0x8418 | Priority level register (PPI[11:8]) | 0x0000_0000 |
| ICDIPR7_CPU2 | 0x841C | Priority level register (PPI[15:12]) | 0x0000_0000 |
| ICDIPTR0_CPU2 | 0x8800 | Processor targets register (SGI[3:0]) | 0x0202_0202 |
| ICDIPTR1_CPU2 | 0x8804 | Processor targets register (SGI[7:4]) | 0x0202_0202 |
| ICDIPTR2_CPU2 | 0x8808 | Processor targets register (SGI[11:8]) | 0x0202_0202 |
| ICDIPTR3_CPU2 | 0x880C | Processor targets register (SGI[15:12]) | 0x0202_0202 |
| ICDIPTR4_CPU2 | 0x8810 | Processor targets register (PPI[3:0]) | 0x0202_0202 |
| ICDIPTR5_CPU2 | 0x8814 | Processor targets register (PPI[7:4]) | 0x0202_0202 |
| ICDIPTR6_CPU2 | 0x8818 | Processor targets register (PPI[11:8]) | 0x0202_0202 |
| ICDIPTR7_CPU2 | 0x881C | Processor targets register (PPI[15:12]) | 0x0202_0202 |
| ICDICFR0_CPU2 | 0x8C00 | Interrupt configuration register (SGI[15:0]) | 0xAAAA_AAAA |
| ICDICFR1_CPU2 | 0x8C04 | Interrupt configuration register (PPI[15:0]) | 0x7DD5_5FFF |
| PPI_STATUS_CPU2 | 0x8D00 | PPI status register | 0x0000_0000 |
| ICDISR0_CPU3 | 0xC080 | Interrupt security registers (SGI,PPI) | 0x0000_0000 |
| ICDISER0_CPU3 | 0xC100 | Interrupt set-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDICER0_CPU3 | 0xC180 | Interrupt clear-enable register (SGI,PPI) | 0x0000_FFFF |
| ICDISPR0_CPU3 | 0xC200 | Interrupt pending-set register (SGI,PPI) | 0x0000_0000 |
| ICDICPR0_CPU3 | 0xC280 | Interrupt pending-clear register (SGI,PPI) | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|-----------------|--------|--|-------------|
| ICDABR0_CPU3 | 0xC300 | Active status register (SGI, PPI) | 0x0000_0000 |
| ICDIPR0_CPU3 | 0xC400 | Priority level register (SGI[3:0]) | 0x0000_0000 |
| ICDIPR1_CPU3 | 0xC404 | Priority level register (SGI[7:4]) | 0x0000_0000 |
| ICDIPR2_CPU3 | 0xC408 | Priority level register (SGI[11:8]) | 0x0000_0000 |
| ICDIPR3_CPU3 | 0xC40C | Priority level register (SGI[15:12]) | 0x0000_0000 |
| ICDIPR4_CPU3 | 0xC410 | Priority level register (PPI[3:0]) | 0x0000_0000 |
| ICDIPR5_CPU3 | 0xC414 | Priority level register (PPI[7:4]) | 0x0000_0000 |
| ICDIPR6_CPU3 | 0xC418 | Priority level register (PPI[11:8]) | 0x0000_0000 |
| ICDIPR7_CPU3 | 0xC41C | Priority level register (PPI[15:12]) | 0x0000_0000 |
| ICDIPTR0_CPU3 | 0xC800 | Processor targets register (SGI[3:0]) | 0x0202_0202 |
| ICDIPTR1_CPU3 | 0xC804 | Processor targets register (SGI[7:4]) | 0x0202_0202 |
| ICDIPTR2_CPU3 | 0xC808 | Processor targets register (SGI[11:8]) | 0x0202_0202 |
| ICDIPTR3_CPU3 | 0xC80C | Processor targets register (SGI[15:12]) | 0x0202_0202 |
| ICDIPTR4_CPU3 | 0xC810 | Processor targets register (PPI[3:0]) | 0x0202_0202 |
| ICDIPTR5_CPU3 | 0xC814 | Processor targets register (PPI[7:4]) | 0x0202_0202 |
| ICDIPTR6_CPU3 | 0xC818 | Processor targets register (PPI[11:8]) | 0x0202_0202 |
| ICDIPTR7_CPU3 | 0xC81C | Processor targets register (PPI[15:12]) | 0x0202_0202 |
| ICDICFR0_CPU3 | 0xCC00 | Interrupt configuration register (SGI[15:0]) | 0xAAAA_AAAA |
| ICDICFR1_CPU3 | 0xCC04 | Interrupt configuration register (PPI[15:0]) | 0x7DD5_5FFF |
| PPI_STATUS_CPU3 | 0xCD00 | PPI status register | 0x0000_0000 |

7 Interrupt Combiner

7.1 Overview

Interrupt controller in Exynos 4412 consists of:

- PrimeCell generic interrupt controller (PL390)
- Interrupt combiner

A few interrupt sources are grouped in Exynos 4412. Interrupt combiner combines several interrupt sources as a group. Several interrupt requests in a group make a group interrupt request and a single request signal. As a result, the interrupt input sources of PrimeCell generic interrupt controller consists of the group interrupt requests from the interrupt combiner and uncombined interrupt sources.

7.2 Features

The features of Interrupt Combiner are:

- 116 interrupt source inputs.
- 18 group interrupt outputs.
- Enables or masks each interrupt source in a group.
- Provides the status of interrupt source in a group before interrupt masking.
- Provides the status of interrupt source in a group after interrupt masking.
- Provides the status of group interrupt output after interrupt masking and combining

7.3 Functional Description

7.3.1 Block Diagram

There is a interrupt combiner in Exynos 4412. The interrupt combiner combines a few interrupts source into 18 group interrupt request outputs. The inputs of the GIC unit outside the ARM Core unit connect to the group interrupt request outputs.

[Figure 7-1](#) illustrates the block diagram of interrupt combiner.

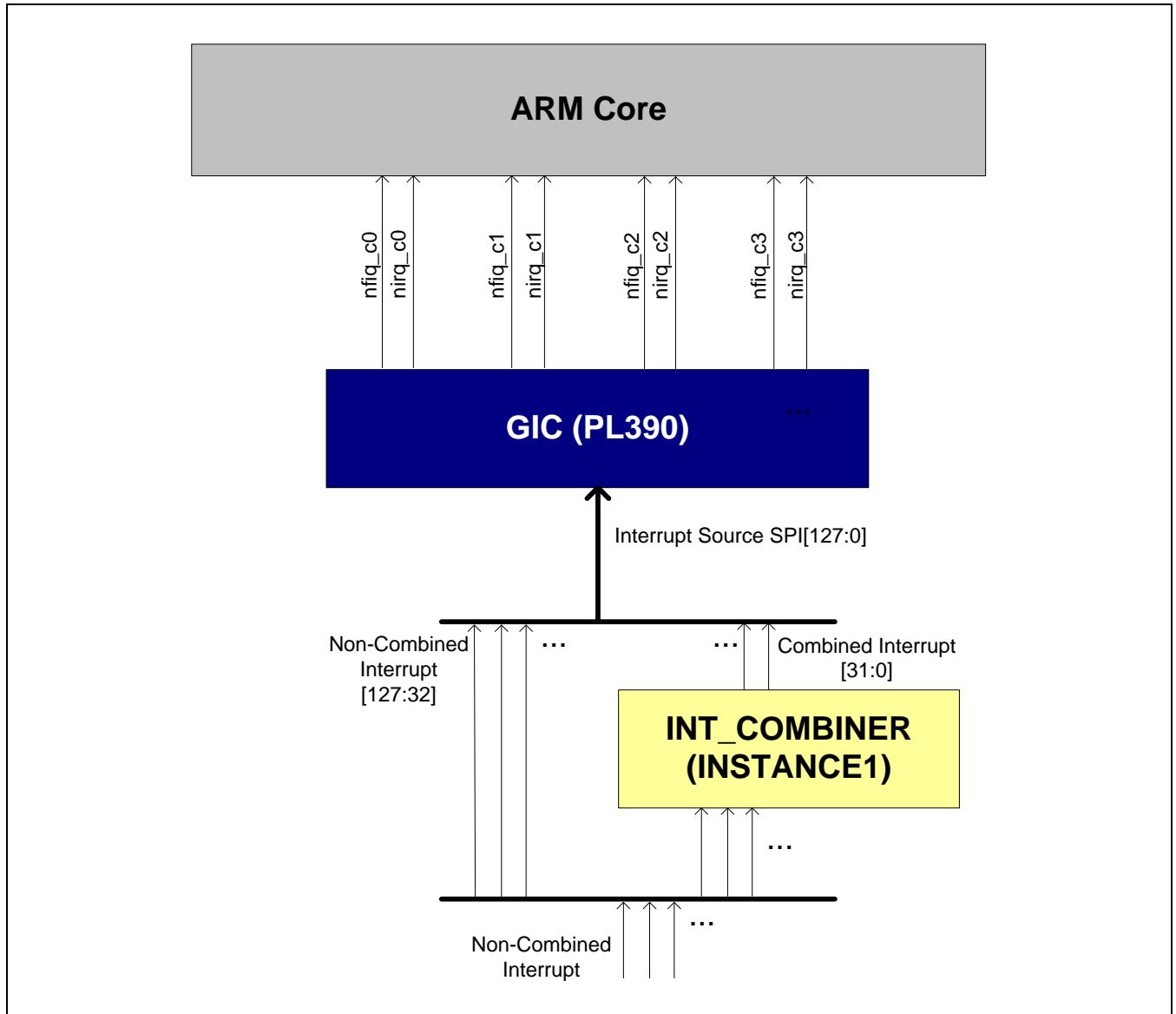


Figure 7-1 Block Diagram of Interrupt Combiner

7.4 Interrupt Sources

This section includes:

- Interrupt Combiner

7.4.1 Interrupt Combiner

[Table 7-1](#) describes the interrupt groups of interrupt combiner.

Table 7-1 Interrupt Groups of Interrupt Combiner

| Combiner Group ID | Combined Interrupt Source Name | Bit | Interrupt Source | Source Block |
|-------------------|--------------------------------|-----|------------------|--------------|
| INTG0 | Reserved | [3] | Reserved | Reserved |
| | | [2] | Reserved | |
| | | [1] | Reserved | |
| | | [0] | Reserved | |
| INTG1 | TZASC | [3] | TZASC1[1] | TZASC |
| | | [2] | TZASC1[0] | |
| | | [1] | TZASC0[1] | |
| | | [0] | TZASC0[0] | |
| INTG2 | PARITYFAIL0/F4DCTI0/PMU0 | [6] | PARRINTR | CPU0 |
| | | [5] | PARRDINTR | |
| | | [4] | TMU | |
| | | [3] | nCTIIRQ[0] | |
| | | [2] | PMUIRQ[0] | |
| | | [1] | PARITYFAILSCU[0] | |
| | | [0] | PARITYFAIL0 | |
| INTG3 | PARITYFAIL1/F4DCTI1/PMU1 | [6] | nCTIIRQ_ISP | CPU1 |
| | | [5] | PMUIRQ_ISP | |
| | | [4] | TMU | |
| | | [3] | nCTIIRQ[1] | |
| | | [2] | PMUIR[1] | |
| | | [1] | PARITYFAILSC[1] | |
| | | [0] | PARITYFAIL1 | |
| INTG4 | SYSMMU[7:0] | [7] | SYSMMU_2D[0] | System MMU |
| | | [6] | SYSMMU_JPEG[0] | |
| | | [5] | SYSMMU_FIMC3[0] | |
| | | [4] | SYSMMU_FIMC2[0] | |
| | | [3] | SYSMMU_FIMC1[0] | |
| | | [2] | SYSMMU_FIMC0[0] | |

| Combiner Group ID | Combined Interrupt Source Name | Bit | Interrupt Source | Source Block |
|-------------------|--------------------------------|-----|-------------------|--------------|
| INTG5 | SYSMMU[15:8] | [1] | SYSMMU_SSS[0] | System MMU |
| | | [0] | SYSMMU_MDMA[0] | |
| | | [7] | Reserved | |
| | | [6] | SYSMMU_MFC_M1[0] | |
| | | [5] | SYSMMU_MFC_M0[0] | |
| | | [4] | SYSMMU_TV_M0[0] | |
| | | [3] | Reserved | |
| | | [2] | SYSMMU_LCD0_M0[0] | |
| | | [1] | SYSMMU_GPS[0] | |
| INTG6 | SYSMMU[23:16] | [0] | SYSMMU_ROTATOR[0] | System MMU |
| | | [7] | SYSMMU_2D[1] | |
| | | [6] | SYSMMU_JPEG[1] | |
| | | [5] | SYSMMU_FIMC3[1] | |
| | | [4] | SYSMMU_FIMC2[1] | |
| | | [3] | SYSMMU_FIMC1[1] | |
| | | [2] | SYSMMU_FIMC0[1] | |
| | | [1] | SYSMMU_SSS[1] | |
| | | [0] | SYSMMU_MDMA[1] | |
| INTG7 | SYSMMU[31:24] | [7] | Reserved | System MMU |
| | | [6] | SYSMMU_MFC_M1[1] | |
| | | [5] | SYSMMU_MFC_M0[1] | |
| | | [4] | SYSMMU_TV_M0[1] | |
| | | [3] | Reserved | |
| | | [2] | SYSMMU_LCD0_M0[1] | |
| | | [1] | SYSMMU_GPS[1] | |
| | | [0] | SYSMMU_ROTATOR[1] | |
| INTG8 | PPMU [7:0] | [7] | PPMU_IMAGE_M0 | PPMU |
| | | [6] | PPMU_CAMIF_M0 | |
| | | [5] | PPMU_D_RIGHT_M0 | |
| | | [4] | PPMU_D_LEFT_M0 | |
| | | [3] | PPMU_ACP0_M0 | |
| | | [2] | PPMU_XIU_R_S1 | |
| | | [1] | PPMU_XIU_R | |
| | | [0] | PPMU_XIU_L | |
| INTG9 | PPMU [14:8] | [6] | PPMU_MFC_M1 | PPMU |
| | | [5] | PPMU_MFC_M0 | |

| Combiner Group ID | Combined Interrupt Source Name | Bit | Interrupt Source | Source Block |
|-------------------|--------------------------------|-----|------------------|--------------|
| INTG10 | DMC1/DMC0/MIU/L2CACHE | [4] | PPMU_3D | |
| | | [3] | PPMU_TV_M0 | |
| | | [2] | PPMU_FILE_D_M0 | |
| | | [1] | PPMU_ISP_MX | |
| | | [0] | PPMU_LCD0 | |
| INTG11 | LCD0 | [7] | DMC1_PPC_PEREV_M | DMC1 |
| | | [6] | DMC1_PPC_PEREV_A | |
| | | [5] | DMC0_PPC_PEREV_M | DMC0 |
| | | [4] | DMC0_PPC_PEREV_A | |
| | | [3] | ADC | General ADC |
| | | [2] | L2CACHE | L2 Cache |
| | | [1] | RP_TIMER | RP |
| | | [0] | GPIO_AUDIO | Audio_SS |
| INTG12 | MCT/MIPI/UART4 | [3] | LCD0[3] | LCD0 |
| | | [2] | LCD0[2] | |
| | | [1] | LCD0[1] | |
| | | [0] | LCD0[0] | |
| | | [7] | G3 | MCT |
| | | [6] | G2 | |
| INTG13 | CPU | [5] | G1 | |
| | | [4] | G0 | |
| | | [1] | MIPI_HSI | MIPI |
| | | [0] | UART4 | UART |
| | | [5] | CPU_nIRQOUT[0] | CPU |
| | | [4] | Reserved | |
| INTG14 | CPU | [3] | Reserved | |
| | | [2] | Reserved | |
| | | [1] | Reserved | |
| | | [0] | Reserved | |
| | | [6] | CPU_nIRQOUT[1] | CPU |
| | | [5] | Reserved | |
| | | [4] | Reserved | |

| Combiner Group ID | Combined Interrupt Source Name | Bit | Interrupt Source | Source Block |
|-------------------|--------------------------------|-----|----------------------|--------------|
| INTG15 | BUS_ERROR/ SCUEVABORT | [7] | DECERRINTR | F4D |
| | | [6] | SLVERRINTR | |
| | | [5] | ERRRDINTR | |
| | | [4] | ERRRTINTR | |
| | | [3] | ERRWDINTR | |
| | | [2] | ERRWTINTR | |
| | | [1] | ECNTRINTR | |
| | | [0] | SCUEVABORT | |
| INTG16 | ISP | [7] | L3_IRQ | ISP |
| | | [6] | Reserved | |
| | | [5] | SYSMMU_ISP_CX[0] | |
| | | [4] | SYSMMU_FIMC_FD[0] | |
| | | [3] | SYSMMU_FIMC_DRC[0] | |
| | | [2] | SYSMMU_FIMC_ISP[0] | |
| | | [1] | SYSMMU_FIMC_LITE0[0] | |
| | | [0] | SYSMMU_FIMC_LITE0[0] | |
| INTG17 | ISP | [7] | L2_IRQ | ISP |
| | | [6] | Reserved | |
| | | [5] | SYSMMU_ISP_CX[1] | |
| | | [4] | SYSMMU_FIMC_FD[1] | |
| | | [3] | SYSMMU_FIMC_DRC[1] | |
| | | [2] | SYSMMU_FIMC_ISP[1] | |
| | | [1] | SYSMMU_FIMC_LITE0[1] | |
| | | [0] | SYSMMU_FIMC_LITE0[1] | |
| INTG18 | PARITYFAIL2/ F4DCTI2/ PMU2 | [6] | CPU_nIRQOUT[2] | CPU |
| | | [5] | PARITYFAILSCU[2] | |
| | | [4] | PARITYFAIL2 | |
| | | [3] | nCTIIRQ[2] | |
| | | [2] | PMUIRQ[2] | |
| | | [1] | Reserved | |
| | | [0] | L1_IRQ | |
| INTG19 | PARITYFAIL3/ F4DCTI3/ PMU3 | [6] | CPU_nIRQOUT[3] | CPU |
| | | [5] | PARITYFAILSCU[3] | |
| | | [4] | PARITYFAIL3 | |
| | | [3] | nCTIIRQ[3] | |
| | | [2] | PMUIRQ[3] | |

| Combiner Group ID | Combined Interrupt Source Name | Bit | Interrupt Source | Source Block |
|-------------------|--------------------------------|-----|------------------|--------------|
| | | [1] | Reserved | |
| | | [0] | L0_IRQ | |

7.5 Functional Description

An interrupt enable bit controls an interrupt source in an interrupt group. IESRn and IECRn registers control the interrupt enable bits. IESRn register can toggle an interrupt bit to "1". If you write "1" to a bit position on IESRn, then it sets the corresponding bit on the interrupt enable bit to "1". However, IECRn register can toggle an interrupt enable bit to "0". If you write "1" to a bit position on IECRn, then it clears the corresponding bit on the interrupt enable bits to "0". This feature will make it easy to address resource sharing issues in a multi-processor system.

There are several interrupt sources in an interrupt group. If an interrupt enable bit is "0", then it masks the corresponding interrupt. All the interrupt sources in an interrupt group, including the masked interrupt sources, are ORed to form a combined interrupt request signal. The interrupt combiner connects the combined group interrupt request output to an input of a GIC.

You can show each interrupt source status before an interrupt-enable bit masks it by reading ISTRn register. You can show the combined group interrupt request output signal by reading CIPSR0 register.

7.6 Register Description

7.6.1 Register Map Summary

- Base Address: 0x1044_0000

| Register | Offset | Description | Reset Value |
|---------------------------|--------|---|-------------|
| Interrupt Combiner | | | |
| IESR0 | 0x0000 | Interrupt enable set register for group 0 to 3 | 0x00000000 |
| IECR0 | 0x0004 | Interrupt enable clear register for group 0 to 3 | 0x00000000 |
| ISTR0 | 0x0008 | Interrupt status register for group 0 to 3 | Undefined |
| IMSR0 | 0x000C | Interrupt masked status register for group 0 to 3 | Undefined |
| IESR1 | 0x0010 | Interrupt enable set register for group 4 to 7 | 0x00000000 |
| IECR1 | 0x0014 | Interrupt enable clear register for group 4 to 7 | 0x00000000 |
| ISTR1 | 0x0018 | Interrupt status register for group 4 to 7 | Undefined |
| IMSR1 | 0x001C | Interrupt masked status register for group 4 to 7 | Undefined |
| IESR2 | 0x0020 | Interrupt enable set register for group 8 to 11 | 0x00000000 |
| IECR2 | 0x0024 | Interrupt enable clear register for group 8 to 11 | 0x00000000 |
| ISTR2 | 0x0028 | Interrupt status register for group 8 to 11 | Undefined |
| IMSR2 | 0x002C | Interrupt masked status register for group 8 to 11 | Undefined |
| IESR3 | 0x0030 | Interrupt enable set register for group 12 to 15 | 0x00000000 |
| IECR3 | 0x0034 | Interrupt enable clear register for group 12 to 15 | 0x00000000 |
| ISTR3 | 0x0038 | Interrupt masked status register for group 12 to 15 | Undefined |
| IMSR3 | 0x003C | Interrupt status register for group 16 to 17 | Undefined |
| IESR4 | 0x0040 | Interrupt enable set register for group 16 to 17 | 0x00000000 |
| IECR4 | 0x0044 | Interrupt enable clear register for group 16 to 17 | 0x00000000 |
| ISTR4 | 0x0048 | Interrupt masked status register for group 16 to 17 | Undefined |
| IMSR4 | 0x004C | Interrupt status register for group 16 to 17 | Undefined |
| CIPSR0 | 0x0100 | Combined interrupt pending status0 | Undefined |

7.6.2 Interrupt Combiner

7.6.2.1 IESR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| nCTIIRQ_ISP | [30] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| PMUIRQ_ISP | [29] | RW | | 0 |
| TMU | [28] | RW | | 0 |
| nCTIIRQ[1] | [27] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1" | 0 |
| PMUIR[1] | [26] | RW | Read) The current interrupt enable bit. | 0 |
| PARITYFAILSC[1] | [25] | RW | 0 = Masks. 1 = Enables. | 0 |
| PARITYFAIL1 | [24] | RW | | 0 |
| RSVD | [23] | – | Reserved | 0x0 |
| PARRINTR | [22] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| PARRDINTR | [21] | RW | | 0 |
| TMU | [20] | RW | | 0 |
| nCTIIRQ[0] | [19] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| PMUIRQ[0] | [18] | RW | Read) The current interrupt enable bit. | 0 |
| PARITYFAILSCU[0] | [17] | RW | 0 = Masks. 1 = Enables. | 0 |
| PARITYFAIL0 | [16] | RW | | 0 |
| RSVD | [15:12] | – | Reserved | 0x0 |
| TZASC1[1] | [11] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| TZASC1[0] | [10] | RW | | 0 |
| TZASC0[1] | [9] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| TZASC0[0] | [8] | RW | Read) The current interrupt enable bit 0 = Masks. 1 = Enables. | 0 |
| RSVD | [7:4] | – | Reserved | 0x0 |
| RSVD | [3] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| RSVD | [2] | RW | | 0 |
| RSVD | [1] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [0] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |

7.6.2.2 IECR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|-------------|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| nCTIIRQ_ISP | [30] | RW | | 0 |
| PMUIRQ_ISP | [29] | RW | | 0 |
| TMU | [28] | RW | | 0 |
| nCTIIRQ[1] | [27] | RW | | 0 |
| PMUIRQ[1] | [26] | RW | | 0 |
| PARITYFAILSC[1] | [25] | RW | | 0 |
| PARITYFAIL1 | [24] | RW | | 0 |
| RSVD | [23] | – | Reserved | 0x0 |
| PARRINTR | [22] | RW | | 0 |
| PARRDINTR | [21] | RW | | 0 |
| TMU | [20] | RW | | 0 |
| nCTIIRQ[0] | [19] | RW | | 0 |
| PMUIRQ[0] | [18] | RW | | 0 |
| PARITYFAILSCU[0] | [17] | RW | | 0 |
| PARITYFAIL0 | [16] | RW | | 0 |
| RSVD | [15:12] | – | Reserved | 0x0 |
| TZASC1[1] | [11] | RW | | 0 |
| TZASC1[0] | [10] | RW | | 0 |
| TZASC0[1] | [9] | RW | | 0 |
| TZASC0[0] | [8] | RW | | 0 |
| RSVD | [7:4] | – | Reserved | 0x0 |
| RSVD | [3] | RW | | 0 |
| RSVD | [2] | RW | | 0 |
| RSVD | [1] | RW | | 0 |
| RSVD | [0] | RW | | 0 |

7.6.2.3 ISTR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0008, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| nCTIIRQ_ISP | [30] | R | | – |
| PMUIRQ_ISP | [29] | R | | – |
| TMU | [28] | R | | – |
| nCTIIRQ[1] | [27] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | – |
| PMUIRQ[1] | [26] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PARITYFAILSC[1] | [25] | R | | – |
| PARITYFAIL1 | [24] | R | | – |
| RSVD | [23] | – | Reserved | 0x0 |
| PARRINTR | [22] | R | | – |
| PARRDINTR | [21] | R | | – |
| TMU | [20] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | – |
| nCTIIRQ[0] | [19] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PMUIRQ[0] | [18] | R | | – |
| PARITYFAILSCU[0] | [17] | R | | – |
| PARITYFAIL0 | [16] | R | | – |
| RSVD | [15:12] | – | Reserved | 0x0 |
| TZASC1[1] | [11] | R | Interrupt pending status. | – |
| TZASC1[0] | [10] | R | The corresponding interrupt enable bit does not affect this pending status. | – |
| TZASC0[1] | [9] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| TZASC0[0] | [8] | R | | – |
| RSVD | [7:4] | – | Reserved | 0x0 |
| RSVD | [3] | R | Interrupt pending status. | – |
| RSVD | [2] | R | The corresponding interrupt enable bit does not affect this pending status. | – |
| RSVD | [1] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| RSVD | [0] | R | | – |

7.6.2.4 IMSR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31] | – | Reserved | 0x0 |
| nCTIIRQ_ISP | [30] | R | | – |
| PMUIRQ_ISP | [29] | R | | – |
| TMU | [28] | R | | – |
| nCTIIRQ[1] | [27] | R | | – |
| PMUIRQ[1] | [26] | R | | – |
| PARITYFAILSC[1] | [25] | R | | – |
| PARITYFAIL1 | [24] | R | | – |
| RSVD | [23] | – | Reserved | 0x0 |
| PARRINTR | [22] | R | | – |
| PARRDINTR | [21] | R | | – |
| TMU | [20] | R | | – |
| nCTIIRQ[0] | [19] | R | | – |
| PMUIRQ[0] | [18] | R | | – |
| PARITYFAILSCU[0] | [17] | R | | – |
| PARITYFAIL0 | [16] | R | | – |
| RSVD | [15:12] | – | Reserved | 0x0 |
| TZASC1[1] | [11] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". | – |
| TZASC1[0] | [10] | R | | – |
| TZASC0[1] | [9] | R | | – |
| TZASC0[0] | [8] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| RSVD | [7:4] | – | Reserved | 0x0 |
| RSVD | [3] | R | Masked interrupt pending status. | – |
| RSVD | [2] | R | If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". | – |
| RSVD | [1] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| RSVD | [0] | R | | – |

7.6.2.5 IESR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|------|------|---|-------------|
| RSVD | [31] | — | | 0 |
| SYSMMU_MFC_M1[1] | [30] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_MFC_M0[1] | [29] | RW | | 0 |
| SYSMMU_TV_M0[1] | [28] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [27] | — | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_LCD0_M0[1] | [26] | RW | | 0 |
| SYSMMU_GPS[1] | [25] | RW | | 0 |
| SYSMMU_ROTATOR[1] | [24] | RW | | 0 |
| SYSMMU_2D[1] | [23] | RW | | 0 |
| SYSMMU_JPEG[1] | [22] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_FIMC3[1] | [21] | RW | | 0 |
| SYSMMU_FIMC2[1] | [20] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| SYSMMU_FIMC1[1] | [19] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_FIMC0[1] | [18] | RW | | 0 |
| SYSMMU_SSS[1] | [17] | RW | | 0 |
| SYSMMU_MDMA[1] | [16] | RW | | 0 |
| RSVD | [15] | — | | 0 |
| SYSMMU_MFC_M1[0] | [14] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_MFC_M0[0] | [13] | RW | | 0 |
| SYSMMU_TV_M0[0] | [12] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [11] | — | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_LCD0_M0[0] | [10] | RW | | 0 |
| SYSMMU_GPS[0] | [9] | RW | | 0 |
| SYSMMU_ROTATOR[0] | [8] | RW | | 0 |
| SYSMMU_2D[0] | [7] | RW | | 0 |
| SYSMMU_JPEG[0] | [6] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_FIMC3[0] | [5] | RW | | 0 |
| SYSMMU_FIMC2[0] | [4] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| SYSMMU_FIMC1[0] | [3] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_FIMC0[0] | [2] | RW | | 0 |
| SYSMMU_SSS[0] | [1] | RW | | 0 |
| SYSMMU_MDMA[0] | [0] | RW | | 0 |

7.6.2.6 IECR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------|------|------|-------------|-------------|
| RSVD | [31] | — | | 0 |
| SYSMMU_MFC_M1[1] | [30] | RW | | 0 |
| SYSMMU_MFC_M0[1] | [29] | RW | | 0 |
| SYSMMU_TV_M0[1] | [28] | RW | | 0 |
| RSVD | [27] | — | | 0 |
| SYSMMU_LCD0_M0[1] | [26] | RW | | 0 |
| SYSMMU_GPS[1] | [25] | RW | | 0 |
| SYSMMU_ROTATOR[1] | [24] | RW | | 0 |
| SYSMMU_2D[1] | [23] | RW | | 0 |
| SYSMMU_JPEG[1] | [22] | RW | | 0 |
| SYSMMU_FIMC3[1] | [21] | RW | | 0 |
| SYSMMU_FIMC2[1] | [20] | RW | | 0 |
| SYSMMU_FIMC1[1] | [19] | RW | | 0 |
| SYSMMU_FIMC0[1] | [18] | RW | | 0 |
| SYSMMU_SSS[1] | [17] | RW | | 0 |
| SYSMMU_MDMA[1] | [16] | RW | | 0 |
| RSVD | [15] | — | | 0 |
| SYSMMU_MFC_M1[0] | [14] | RW | | 0 |
| SYSMMU_MFC_M0[0] | [13] | RW | | 0 |
| SYSMMU_TV_M0[0] | [12] | RW | | 0 |
| RSVD | [11] | — | | 0 |
| SYSMMU_LCD0_M0[0] | [10] | RW | | 0 |
| SYSMMU_GPS[0] | [9] | RW | | 0 |
| SYSMMU_ROTATOR[0] | [8] | RW | | 0 |
| SYSMMU_2D[0] | [7] | RW | | 0 |
| SYSMMU_JPEG[0] | [6] | RW | | 0 |
| SYSMMU_FIMC3[0] | [5] | RW | | 0 |
| SYSMMU_FIMC2[0] | [4] | RW | | 0 |
| SYSMMU_FIMC1[0] | [3] | RW | | 0 |
| SYSMMU_FIMC0[0] | [2] | RW | | 0 |
| SYSMMU_SSS[0] | [1] | RW | | 0 |
| SYSMMU_MDMA[0] | [0] | RW | | 0 |

7.6.2.7 ISTR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x0018, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|-------------------|------|------|--|-------------|
| RSVD | [31] | — | | — |
| SYSMMU_MFC_M1[1] | [30] | R | | — |
| SYSMMU_MFC_M0[1] | [29] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | — |
| SYSMMU_TV_M0[1] | [28] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| RSVD | [27] | — | | — |
| SYSMMU_LCD0_M0[1] | [26] | R | | — |
| SYSMMU_GPS[1] | [25] | R | | — |
| SYSMMU_ROTATOR[1] | [24] | R | | — |
| SYSMMU_2D[1] | [23] | R | | — |
| SYSMMU_JPEG[1] | [22] | R | | — |
| SYSMMU_FIMC3[1] | [21] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | — |
| SYSMMU_FIMC2[1] | [20] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| SYSMMU_FIMC1[1] | [19] | R | | — |
| SYSMMU_FIMC0[1] | [18] | R | | — |
| SYSMMU_SSS[1] | [17] | R | | — |
| SYSMMU_MDMA[1] | [16] | R | | — |
| RSVD | [15] | — | | — |
| SYSMMU_MFC_M1[0] | [14] | R | | — |
| SYSMMU_MFC_M0[0] | [13] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | — |
| SYSMMU_TV_M0[0] | [12] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| RSVD | [11] | — | | — |
| SYSMMU_LCD0_M0[0] | [10] | R | | — |
| SYSMMU_GPS[0] | [9] | R | | — |
| SYSMMU_ROTATOR[0] | [8] | R | | — |
| SYSMMU_2D[0] | [7] | R | | — |
| SYSMMU_JPEG[0] | [6] | R | | — |
| SYSMMU_FIMC3[0] | [5] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. | — |
| SYSMMU_FIMC2[0] | [4] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| SYSMMU_FIMC1[0] | [3] | R | | — |
| SYSMMU_FIMC0[0] | [2] | R | | — |
| SYSMMU_SSS[0] | [1] | R | | — |
| SYSMMU_MDMA[0] | [0] | R | | — |

7.6.2.8 IMSR1

- Base Address: 0x1044_0000
- Address = Base Address + 0x001C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|-------------------|------|------|--|-------------|
| RSVD | [31] | — | | — |
| SYSMMU_MFC_M1[1] | [30] | R | | — |
| SYSMMU_MFC_M0[1] | [29] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". | — |
| SYSMMU_TV_M0[1] | [28] | R | | — |
| RSVD | [27] | — | | — |
| SYSMMU_LCD0_M0[1] | [26] | R | | — |
| SYSMMU_GPS[1] | [25] | R | | — |
| SYSMMU_ROTATOR[1] | [24] | R | | — |
| SYSMMU_2D[1] | [23] | R | | — |
| SYSMMU_JPEG[1] | [22] | R | | — |
| SYSMMU_FIMC3[1] | [21] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". | — |
| SYSMMU_FIMC2[1] | [20] | R | | — |
| SYSMMU_FIMC1[1] | [19] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| SYSMMU_FIMC0[1] | [18] | R | | — |
| SYSMMU_SSS[1] | [17] | R | | — |
| SYSMMU_MDMA[1] | [16] | R | | — |
| RSVD | [15] | — | | — |
| SYSMMU_MFC_M1[0] | [14] | R | | — |
| SYSMMU_MFC_M0[0] | [13] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". | — |
| SYSMMU_TV_M0[0] | [12] | R | | — |
| RSVD | [11] | — | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| SYSMMU_LCD0_M0[0] | [10] | R | | — |
| SYSMMU_GPS[0] | [9] | R | | — |
| SYSMMU_ROTATOR[0] | [8] | R | | — |
| SYSMMU_2D[0] | [7] | R | | — |
| SYSMMU_JPEG[0] | [6] | R | | — |
| SYSMMU_FIMC3[0] | [5] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". | — |
| SYSMMU_FIMC2[0] | [4] | R | | — |
| SYSMMU_FIMC1[0] | [3] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | — |
| SYSMMU_FIMC0[0] | [2] | R | | — |
| SYSMMU_SSS[0] | [1] | R | | — |
| SYSMMU_MDMA[0] | [0] | R | | — |

7.6.2.9 IESR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| LCD0[3] | [27] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| LCD0[2] | [26] | RW | | 0 |
| LCD0[1] | [25] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| LCD0[0] | [24] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| DMC1_PPC_PEREV_M | [23] | RW | Set the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| DMC1_PPC_PEREV_A | [22] | RW | | 0 |
| DMC0_PPC_PEREV_M | [21] | RW | | 0 |
| DMC0_PPC_PEREV_A | [20] | RW | | 0 |
| ADC | [19] | RW | | 0 |
| L2CACHE | [18] | RW | | 0 |
| RP_TIMER | [17] | RW | | 0 |
| GPIO_AUDIO | [16] | – | | 0 |
| RSVD | [15] | – | Reserved | 0 |
| PPMU_MFC_M1 | [14] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| PPMU_MFC_M0 | [13] | RW | | 0 |
| PPMU_3D | [12] | RW | | 0 |
| PPMU_TV_M0 | [11] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| PPMU_FILE_D_M0 | [10] | RW | | 0 |
| PPMU_ISP_MX | [9] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| PPMU_LCD0 | [8] | RW | | 0 |
| PPMU_IMAGE_M0 | [7] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| PPMU_CAMIF_M0 | [6] | RW | | 0 |
| PPMU_D_RIGHT_M0 | [5] | RW | | 0 |
| PPMU_D_LEFT_M0 | [4] | RW | | 0 |
| PPMU_ACP0_M0 | [3] | RW | | 0 |
| PPMU_XIU_R_S1 | [2] | RW | | 0 |
| PPMU_XIU_R | [1] | RW | | 0 |
| PPMU_XIU_L | [0] | RW | | 0 |

7.6.2.10 IECR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| LCD0[3] | [27] | RW | Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. | 0 |
| LCD0[2] | [26] | RW | | 0 |
| LCD0[1] | [25] | RW | Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". | 0 |
| LCD0[0] | [24] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| DMC1_PPC_PEREV_M | [23] | RW | | 0 |
| DMC1_PPC_PEREV_A | [22] | RW | Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. | 0 |
| DMC0_PPC_PEREV_M | [21] | RW | | 0 |
| DMC0_PPC_PEREV_A | [20] | RW | Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". | 0 |
| ADC | [19] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| L2CACHE | [18] | RW | | 0 |
| RP_TIMER | [17] | RW | | 0 |
| GPIO_AUDIO | [16] | RW | | 0 |
| RSVD | [15] | – | Reserved | 0 |
| PPMU_MFC_M1 | [14] | RW | Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. | 0 |
| PPMU_MFC_M0 | [13] | RW | | 0 |
| PPMU_3D | [12] | RW | | 0 |
| PPMU_TV_M0 | [11] | RW | Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". | 0 |
| PPMU_FILE_D_M0 | [10] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| PPMU_ISP_MX | [9] | RW | | 0 |
| PPMU_LCD0 | [8] | RW | | 0 |
| PPMU_IMAGE_M0 | [7] | RW | | 0 |
| PPMU_CAMIF_M0 | [6] | RW | Clears the corresponding interrupt enable bit to "0". If you clear the interrupt enable bit, interrupt combiner will mask the interrupt. | 0 |
| PPMU_D_RIGHT_M0 | [5] | RW | | 0 |
| PPMU_D_LEFT_M0 | [4] | RW | Write) 0 = Does not change the current setting. 1 = Clears the interrupt enable bit to "0". | 0 |
| PPMU_ACP0_M0 | [3] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| PPMU_XIU_R_S1 | [2] | RW | | 0 |
| PPMU_XIU_R | [1] | RW | | 0 |
| PPMU_XIU_L | [0] | RW | | 0 |

7.6.2.11 ISTR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x0028, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| LCD0[3] | [27] | R | Interrupt pending status. | – |
| LCD0[2] | [26] | R | The corresponding interrupt enable bit does not affect this pending status. | – |
| LCD0[1] | [25] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| LCD0[0] | [24] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| DMC1_PPC_PEREV_M | [23] | R | Interrupts pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| DMC1_PPC_PEREV_A | [22] | R | | – |
| DMC0_PPC_PEREV_M | [21] | R | | – |
| DMC0_PPC_PEREV_A | [20] | R | | – |
| ADC | [19] | R | | – |
| L2CACHE | [18] | R | | – |
| RP_TIMER | [17] | R | | – |
| GPIO_AUDIO | [16] | R | | 0 |
| RSVD | [15] | – | Reserved | 0 |
| PPMU_MFC_M1 | [14] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PPMU_MFC_M0 | [13] | R | | – |
| PPMU_3D | [12] | R | | – |
| PPMU_TV_M0 | [11] | R | | – |
| PPMU_FILE_D_M0 | [10] | R | | – |
| PPMU_ISP_MX | [9] | R | | – |
| PPMU_LCD0 | [8] | R | | – |
| PPMU_IMAGE_M0 | [7] | R | | – |
| PPMU_CAMIF_M0 | [6] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PPMU_D_RIGHT_M0 | [5] | R | | – |
| PPMU_D_LEFT_M0 | [4] | R | | – |
| PPMU_ACP0_M0 | [3] | R | | – |
| PPMU_XIU_R_S1 | [2] | R | | – |
| PPMU_XIU_R | [1] | R | | – |
| PPMU_XIU_L | [0] | R | | – |

7.6.2.12 IMSR2

- Base Address: 0x1044_0000
- Address = Base Address + 0x002C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|--|-------------|
| RSVD | [31:28] | – | Reserved | 0x0 |
| LCD0[3] | [27] | R | Masked interrupt pending status. | – |
| LCD0[2] | [26] | R | If the corresponding interrupt enable bit is "0", the IMSR bit reads out as "0". | – |
| LCD0[1] | [25] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| LCD0[0] | [24] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| DMC1_PPC_PEREV_M | [23] | R | | – |
| DMC1_PPC_PEREV_A | [22] | R | | – |
| DMC0_PPC_PEREV_M | [21] | R | Masked interrupt pending status. | – |
| DMC0_PPC_PEREV_A | [20] | R | If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". | – |
| ADC | [19] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| L2CACHE | [18] | R | | – |
| RP_TIMER | [17] | R | | – |
| GPIO_AUDIO | [16] | R | | 0 |
| RSVD | [15] | – | Reserved | 0 |
| PPMU_MFC_M1 | [14] | R | | – |
| PPMU_MFC_M0 | [13] | R | | – |
| PPMU_3D | [12] | R | Masked interrupt pending status. | – |
| PPMU_TV_M0 | [11] | R | If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". | – |
| PPMU_FILE_D_M0 | [10] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PPMU_ISP_MX | [9] | R | | – |
| PPMU_LCD0 | [8] | R | | – |
| PPMU_IMAGE_M0 | [7] | R | | – |
| PPMU_CAMIF_M0 | [6] | R | | – |
| PPMU_D_RIGHT_M0 | [5] | R | Masked interrupt pending status. | – |
| PPMU_D_LEFT_M0 | [4] | R | If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". | – |
| PPMU_ACP0_M0 | [3] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| PPMU_XIU_R_S1 | [2] | R | | – |
| PPMU_XIU_R | [1] | R | | – |
| PPMU_XIU_L | [0] | R | | – |

7.6.2.13 IESR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| DECERRINTR | [31] | RW | | 0 |
| SLVERRINTR | [30] | RW | | 0 |
| ERRRDINTR | [29] | RW | Sets the corresponding interrupt enable bit to "1" If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| ERRRTINTR | [28] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| ERRWDINTR | [27] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| ERRWTINTR | [26] | RW | | 0 |
| ECNTRINTR | [25] | RW | | 0 |
| SCUEVABORT | [24] | RW | | 0 |
| RSVD | [23] | - | | 0 |
| CPU_nIRQOUT_1 | [22] | RW | Sets the corresponding interrupt enable bit to "1" If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| RSVD | [21] | - | | 0 |
| RSVD | [20] | - | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [19] | - | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| RSVD | [18] | - | | 0 |
| RSVD | [17] | - | | 0 |
| RSVD | [16] | - | | 0 |
| RSVD | [15:14] | - | | 00 |
| CPU_nIRQOUT_0 | [13] | RW | Sets the corresponding interrupt enable bit to "1" If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| RSVD | [12] | - | | 0 |
| RSVD | [11] | - | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [10] | - | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| RSVD | [9] | - | | 0 |
| RSVD | [8] | - | | 0 |
| MCT_G3 | [7] | RW | Sets the corresponding interrupt enable bit to "1" If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| MCT_G2 | [6] | RW | | 0 |
| MCT_G1 | [5] | RW | | 0 |
| MCT_G0 | [4] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| RSVD | [3:2] | - | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0x0 |
| MIPI_HSI | [1] | RW | | 0 |
| UART4 | [0] | RW | | 0 |

7.6.2.14 IECR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|-------------|-------------|
| DECERRINTR | [31] | RW | | 0 |
| SLVERRINTR | [30] | RW | | 0 |
| ERRRDINTR | [29] | RW | | 0 |
| ERRRTINTR | [28] | RW | | 0 |
| ERRWDINTR | [27] | RW | | 0 |
| ERRWTINTR | [26] | RW | | 0 |
| ECNTRINTR | [25] | RW | | 0 |
| SCUEVABORT | [24] | RW | | 0 |
| RSVD | [23] | — | | 0 |
| CPU_nIRQOUT_1 | [22] | RW | | 0 |
| RSVD | [21] | — | | 0 |
| RSVD | [20] | — | | 0 |
| RSVD | [19] | — | | 0 |
| RSVD | [18] | — | | 0 |
| RSVD | [17] | — | | 0 |
| RSVD | [16] | — | | 0 |
| RSVD | [15:14] | — | | 00 |
| CPU_nIRQOUT_0 | [13] | RW | | 0 |
| RSVD | [12] | — | | 0 |
| RSVD | [11] | — | | 0 |
| RSVD | [10] | — | | 0 |
| RSVD | [9] | — | | 0 |
| RSVD | [8] | — | | 0 |
| MCT_G3 | [7] | RW | | 0 |
| MCT_G2 | [6] | RW | | 0 |
| MCT_G1 | [5] | RW | | 0 |
| MCT_G0 | [4] | RW | | 0 |
| RSVD | [3:2] | RW | | 00 |
| MIPI_HSI] | [1] | RW | | 0 |
| UART4 | [0] | RW | | 0 |
| LCD1[0] | [0] | RW | | 0 |

7.6.2.15 ISTR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x0038, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| DECERRINTR | [31] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| SLVERRINTR | [30] | R | | – |
| ERRRDINTR | [29] | R | | – |
| ERRRTINTR | [28] | R | | – |
| ERRWDINTR | [27] | R | | – |
| ERRWTINTR | [26] | R | | – |
| ECNTRINTR | [25] | R | | – |
| SCUEVABORT | [24] | R | | – |
| RSVD | [23] | – | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| CPU_nIRQOUT_1 | [22] | R | | – |
| RSVD | [21] | – | | – |
| RSVD | [20] | – | | – |
| RSVD | [19] | – | | – |
| RSVD | [18] | – | | – |
| RSVD | [17] | – | | – |
| RSVD | [16] | – | | – |
| RSVD | [15:14] | – | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | 00 |
| CPU_nIRQOUT_0 | [13] | R | | – |
| RSVD | [12] | – | | – |
| RSVD | [11] | – | | – |
| RSVD | [10] | – | | – |
| RSVD | [9] | – | | – |
| RSVD | [8] | – | | – |
| MCT_G3 | [7] | R | | 0x0 |
| MCT_G2 | [6] | R | Interrupt pending status. The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | – |
| MCT_G1 | [5] | R | | – |
| MCT_G0 | [4] | R | | – |
| RSVD | [3:2] | – | | – |
| MIPI_HSI] | [1] | R | | – |
| UART4 | [0] | R | | – |
| LCD1[0] | [0] | R | | – |

7.6.2.16 IMSR3

- Base Address: 0x1044_0000
- Address = Base Address + 0x003C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|-------------|-------------|
| DECERRINTR | [31] | R | | – |
| SLVERRINTR | [30] | R | | – |
| ERRRDINTR | [29] | R | | – |
| ERRRTINTR | [28] | R | | – |
| ERRWDINTR | [27] | R | | – |
| ERRWTINTR | [26] | R | | – |
| ECNTRINTR | [25] | R | | – |
| SCUEVABORT | [24] | R | | – |
| RSVD | [23] | – | | 0 |
| CPU_nIRQOUT_1 | [22] | R | | – |
| RSVD | [21] | – | | – |
| RSVD | [20] | – | | – |
| RSVD | [19] | – | | – |
| RSVD | [18] | – | | – |
| RSVD | [17] | – | | – |
| RSVD | [16] | – | | – |
| RSVD | [15:14] | – | | 00 |
| CPU_nIRQOUT_0 | [13] | R | | – |
| RSVD | [12] | – | | – |
| RSVD | [11] | – | | – |
| RSVD | [10] | – | | – |
| RSVD | [9] | – | | – |
| RSVD | [8] | – | | – |
| MCT_G3 | [7] | R | | - |
| MCT_G2 | [6] | R | | - |
| MCT_G1 | [5] | R | | - |
| MCT_G0 | [4] | R | | - |
| RSVD | [3:2] | - | | 00 |
| MIPI_HSI | [1] | R | | – |
| UART4 | [0] | R | | – |
| LCD1[0] | [0] | R | | – |

7.6.2.17 IESR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|------|------|---|-------------|
| RSVD | [31] | — | | 0 |
| CPU_nIRQOUT_3 | [30] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| PARITYFAILSCU3 | [29] | RW | | 0 |
| PARITYFAIL3 | [28] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| CPU_nCTIIRQ_3 | [27] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| CPU_PMUIRQ_3 | [26] | RW | | 0 |
| RSVD | [25] | RW | | 0 |
| MCT_L0 | [24] | RW | | 0 |
| RSVD | [23] | — | | 0 |
| CPU_nIRQOUT_2 | [22] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| PARITYFAILSCU2 | [21] | RW | | 0 |
| PARITYFAIL2 | [20] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| CPU_nCTIIRQ_2 | [19] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| CPU_PMUIRQ_2 | [18] | RW | | 0 |
| RSVD | [17] | RW | | 0 |
| MCT_L1 | [16] | RW | | 0 |
| MCT_L2 | [15] | RW | | 0 |
| RSVD | [14] | — | | 0 |
| SYSMMU_ISP_CX[1] | [13] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_FIMC_FD[1] | [12] | RW | | 0 |
| SYSMMU_FIMC_DRC[1] | [11] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| SYSMMU_FIMC_ISP[1] | [10] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_FIMC_LITE0[1] | [9] | RW | | 0 |
| SYSMMU_FIMC_LITE0[1] | [8] | RW | | 0 |
| MCT_L3 | [7] | RW | | 0 |
| RSVD | [6] | — | | 0 |
| SYSMMU_ISP_CX[0] | [5] | RW | Sets the corresponding interrupt enable bit to "1". If you set the interrupt enable bit, interrupt combiner serves the interrupt request. | 0 |
| SYSMMU_FIMC_FD[0] | [4] | RW | | 0 |
| SYSMMU_FIMC_DRC[0] | [3] | RW | Write) 0 = Does not change the current setting. 1 = Sets the interrupt enable bit to "1". | 0 |
| SYSMMU_FIMC_ISP[0] | [2] | RW | Read) The current interrupt enable bit. 0 = Masks. 1 = Enables. | 0 |
| SYSMMU_FIMC_LITE0[0] | [1] | RW | | 0 |
| SYSMMU_FIMC_LITE0[0] | [0] | RW | | 0 |

7.6.2.18 IECR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|------|------|-------------|-------------|
| RSVD | [31] | — | | 0 |
| CPU_nIRQOUT_3 | [30] | RW | | 0 |
| PARITYFAILSCU3 | [29] | RW | | 0 |
| PARITYFAIL3 | [28] | RW | | 0 |
| CPU_nCTIIRQ_3 | [27] | RW | | 0 |
| CPU_PMUIRQ_3 | [26] | RW | | 0 |
| RSVD | [25] | RW | | 0 |
| MCT_L0 | [24] | RW | | 0 |
| RSVD | [23] | — | | 0 |
| CPU_nIRQOUT_2 | [22] | RW | | 0 |
| PARITYFAILSCU2 | [21] | RW | | 0 |
| PARITYFAIL2 | [20] | RW | | 0 |
| CPU_nCTIIRQ_2 | [19] | RW | | 0 |
| CPU_PMUIRQ_2 | [18] | RW | | 0 |
| RSVD | [17] | RW | | 0 |
| MCT_L1 | [16] | RW | | 0 |
| MCT_L2 | [15] | RW | | 0 |
| RSVD | [14] | — | | 0 |
| SYSMMU_ISP_CX[1] | [13] | RW | | 0 |
| SYSMMU_FIMC_FD[1] | [12] | RW | | 0 |
| SYSMMU_FIMC_DRC[1] | [11] | RW | | 0 |
| SYSMMU_FIMC_ISP[1] | [10] | RW | | 0 |
| SYSMMU_FIMC_LITE0[1] | [9] | RW | | 0 |
| SYSMMU_FIMC_LITE0[1] | [8] | RW | | 0 |
| MCT_L3 | [7] | RW | | 0 |
| RSVD | [6] | — | | 0 |
| SYSMMU_ISP_CX[0] | [5] | RW | | 0 |
| SYSMMU_FIMC_FD[0] | [4] | RW | | 0 |
| SYSMMU_FIMC_DRC[0] | [3] | RW | | 0 |
| SYSMMU_FIMC_ISP[0] | [2] | RW | | 0 |
| SYSMMU_FIMC_LITE0[0] | [1] | RW | | 0 |
| SYSMMU_FIMC_LITE0[0] | [0] | RW | | 0 |

7.6.2.19 ISTR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x0048, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|----------------------|------|------|---|-------------|
| RSVD | [31] | - | | 0 |
| CPU_nIRQOUT_3 | [30] | R | | 0 |
| PARITYFAILSCU3 | [29] | R | Interrupt pending status. | 0 |
| PARITYFAIL3 | [28] | R | The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| CPU_nCTIIRQ_3 | [27] | R | | 0 |
| CPU_PMUIRQ_3 | [26] | R | | 0 |
| RSVD | [25] | R | | 0 |
| MCT_L0 | [24] | R | | 0 |
| RSVD | [23] | - | | 0 |
| CPU_nIRQOUT_2 | [22] | R | | 0 |
| PARITYFAILSCU2 | [21] | R | Interrupt pending status. | 0 |
| PARITYFAIL2 | [20] | R | The corresponding interrupt enable bit does not affect this pending status. 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| CPU_nCTIIRQ_2 | [19] | R | | 0 |
| CPU_PMUIRQ_2 | [18] | R | | 0 |
| RSVD | [17] | R | | 0 |
| MCT_L1 | [16] | R | | 0 |
| MCT_L2 | [15] | R | | 0 |
| RSVD | [14] | - | | 0 |
| SYSMMU_ISP_CX[1] | [13] | R | Interrupt pending status. | 0 |
| SYSMMU_FIMC_FD[1] | [12] | R | The corresponding interrupt enable bit does not affect this pending status. | 0 |
| SYSMMU_FIMC_DRC[1] | [11] | R | | 0 |
| SYSMMU_FIMC_ISP[1] | [10] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| SYSMMU_FIMC_LITE0[1] | [9] | R | | 0 |
| SYSMMU_FIMC_LITE0[1] | [8] | R | | 0 |
| MCT_L3 | [7] | R | | 0 |
| RSVD | [6] | - | | 0 |
| SYSMMU_ISP_CX[0] | [5] | R | Interrupt pending status. | 0 |
| SYSMMU_FIMC_FD[0] | [4] | R | The corresponding interrupt enable bit does not affect this pending status. | 0 |
| SYSMMU_FIMC_DRC[0] | [3] | R | | 0 |
| SYSMMU_FIMC_ISP[0] | [2] | R | 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| SYSMMU_FIMC_LITE0[0] | [1] | R | | 0 |
| SYSMMU_FIMC_LITE0[0] | [0] | R | | 0 |

7.6.2.20 IMSR4

- Base Address: 0x1044_0000
- Address = Base Address + 0x004C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|----------------------|------|------|---|-------------|
| RSVD | [31] | - | | 0 |
| CPU_nIRQOUT_3 | [30] | R | | 0 |
| PARITYFAILSCU3 | [29] | R | | 0 |
| PARITYFAIL3 | [28] | R | | 0 |
| CPU_nCTIIRQ_3 | [27] | R | | 0 |
| CPU_PMUIRQ_3 | [26] | R | | 0 |
| RSVD | [25] | R | | 0 |
| MCT_L0 | [24] | R | | 0 |
| RSVD | [23] | - | | 0 |
| CPU_nIRQOUT_2 | [22] | R | | 0 |
| PARITYFAILSCU2 | [21] | R | | 0 |
| PARITYFAIL2 | [20] | R | | 0 |
| CPU_nCTIIRQ_2 | [19] | R | | 0 |
| CPU_PMUIRQ_2 | [18] | R | | 0 |
| RSVD | [17] | R | | 0 |
| MCT_L1 | [16] | R | | 0 |
| MCT_L2 | [15] | R | | 0 |
| RSVD | [14] | - | | 0 |
| SYSMMU_ISP_CX[1] | [13] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| SYSMMU_FIMC_FD[1] | [12] | R | | 0 |
| SYSMMU_FIMC_DRC[1] | [11] | R | | 0 |
| SYSMMU_FIMC_ISP[1] | [10] | R | | 0 |
| SYSMMU_FIMC_LITE0[1] | [9] | R | | 0 |
| SYSMMU_FIMC_LITE0[1] | [8] | R | | 0 |
| MCT_L3 | [7] | R | | 0 |
| RSVD | [6] | - | | 0 |
| SYSMMU_ISP_CX[0] | [5] | R | Masked interrupt pending status. If the corresponding interrupt enable bit is "0", the IMSR bit reads as "0". 0 = The interrupt is not pending. 1 = The interrupt is pending. | 0 |
| SYSMMU_FIMC_FD[0] | [4] | R | | 0 |
| SYSMMU_FIMC_DRC[0] | [3] | R | | 0 |
| SYSMMU_FIMC_ISP[0] | [2] | R | | 0 |
| SYSMMU_FIMC_LITE0[0] | [1] | R | | 0 |
| SYSMMU_FIMC_LITE0[0] | [0] | R | | 0 |

7.6.2.21 CIPSR0

- Base Address: 0x1044_0000
- Address = Base Address + 0x0100, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|--------|------|------|---|-------------|
| INTG31 | [31] | — | Reserved | — |
| INTG30 | [30] | — | | — |
| INTG29 | [29] | — | | — |
| INTG28 | [28] | — | | — |
| INTG27 | [27] | — | | — |
| INTG26 | [26] | — | | — |
| INTG25 | [25] | — | | — |
| INTG24 | [24] | — | | — |
| INTG23 | [23] | — | | — |
| INTG22 | [22] | — | | — |
| INTG21 | [21] | — | | — |
| INTG20 | [20] | — | | — |
| INTG19 | [19] | R | Combined interrupt pending status. 0 = The combined interrupt is not pending. 1 = The combined interrupt is pending. This means the corresponding interrupt request to the GIC is asserted. | — |
| INTG18 | [18] | R | | — |
| INTG17 | [17] | R | | — |
| INTG16 | [16] | R | | — |
| INTG15 | [15] | R | | — |
| INTG14 | [14] | R | | — |
| INTG13 | [13] | R | | — |
| INTG12 | [12] | R | | — |
| INTG11 | [11] | R | | — |
| INTG10 | [10] | R | | — |
| INTG9 | [9] | R | | — |
| INTG8 | [8] | R | | — |
| INTG7 | [7] | R | | — |
| INTG6 | [6] | R | | — |
| INTG5 | [5] | R | | — |
| INTG4 | [4] | R | | — |
| INTG3 | [3] | R | | — |
| INTG2 | [2] | R | | — |
| INTG1 | [1] | R | | — |
| INTG0 | [0] | R | | — |

8

Direct Memory Access Controller (DMAC)

This chapter includes:

- Overview of DMA Controller
- Register description
- Instruction

8.1 Overview

The two Direct Memory Access (DMA) tops that Exynos 4412 supports:

- Memory-to-Memory (M2M) transfer (DMA_mem)
- Peripheral-to-memory transfer and vice-versa (DMA_peri)

The DMA_mem consists of one PL330 (DMA) and some logics. DMA_peri consists of two PL330s (DMA0 and DMA1) and dma_map.

[Figure 8-1](#) illustrates the two DMA tops.

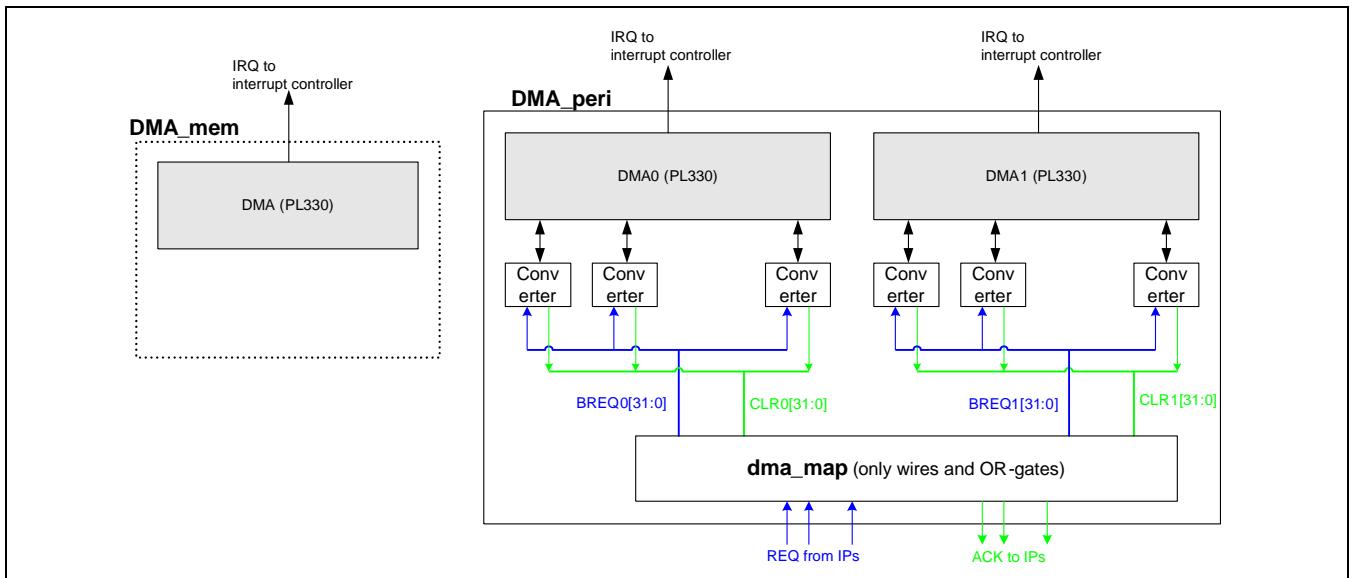


Figure 8-1 Two DMA Tops

The attributes that the DMA_mem DMA Controllers have:

- DMA_mem accesses memory through the AXI_IMGX bus and is located in IMG block.
- DMA_mem supports only the secured AXI transaction.

8.2 Features

[Table 8-1](#) describes the features of DMA Controller. Refer to this table for DMA and for writing DMA assembly code.

Table 8-1 Features of DMA Controller

| Key Features | DMA_mem | DMA_peri |
|---------------------|-----------------------------|------------------------------|
| Supports data size | Up to double word (64-bit) | Up to word (32-bit) |
| Supports burst size | Up to 16 burst | Up to 16 burst |
| Supports channel | 8 channels at the same time | 16 channels at the same time |

Each DMA module has 32 interrupt sources. However, you should send only one interrupt to Interrupt Controller.

[Table 8-2](#) describes the DMA request mapping.

Table 8-2 DMA Request Mapping Table

| Module | No. | |
|-----------|-----|----------------|
| Peri DMA1 | 31 | Reserved |
| | 30 | Reserved |
| | 29 | MIPI_HSI7 |
| | 28 | MIPI_HSI6 |
| | 27 | SPDIF |
| | 26 | Siimbus0AUX_TX |
| | 25 | Siimbus0AUX_RX |
| | 24 | Siimbus5_RX |
| | 23 | Siimbus5_RX |
| | 22 | Siimbus3_TX |
| | 21 | Siimbus3_RX |
| | 20 | Slimbus1_TX |
| | 19 | Slimbus1_RX |
| | 18 | UART3_TX |
| | 17 | UART3_RX |
| | 16 | UART1_TX |
| | 15 | UART1_RX |
| | 14 | UART0_TX |
| | 13 | UART0_RX |
| | 12 | I2S1_TX |
| | 11 | I2S1_RX |
| | 10 | I2S0_TX |
| | 9 | I2S0_RX |

| Module | No. | |
|---------------|------------|-------------|
| Peri DMA0 | 8 | I2S0S_TX |
| | 7 | SPI1_TX |
| | 6 | SPI1_RX |
| | 5 | MIPI_HSI3 |
| | 4 | MIPI_HSI2 |
| | 3 | PCM1_TX |
| | 2 | PCM1_RX |
| | 1 | PCM0_TX |
| | 0 | PCM0_RX |
| | 31 | MIPI_HSI5 |
| | 30 | MIPI_HSI4 |
| | 29 | AC_PCMout |
| | 28 | AC_PCMin |
| | 27 | AC_MICin |
| | 26 | SlimBUS4_TX |
| | 25 | SlimBUS4_RX |
| | 24 | SlimBUS2_TX |
| | 23 | SlimBUS2_RX |
| | 22 | SlimBUS0_TX |
| | 21 | SlimBUS0_RX |
| | 20 | UART4_TX |
| | 19 | UART4_RX |
| | 18 | UART2_TX |
| | 17 | UART2_RX |
| | 16 | UART0_TX |
| | 15 | UART0_RX |
| | 14 | I2S2_TX |
| | 13 | I2S2_RX |
| | 12 | I2S0_TX |
| | 11 | I2S0_RX |
| | 10 | I2S0S_TX |
| | 9 | SPI2_TX |
| | 8 | SPI2_RX |
| | 7 | SPI0_TX |
| | 6 | SPI0_RX |
| | 5 | MIPI_HSI1 |
| | 4 | MIPI_HSI0 |

| Module | No. | |
|---------|-----|---------|
| | 3 | PCM2_TX |
| | 2 | PCM2_RX |
| | 1 | PCM0_TX |
| | 0 | PCM0_RX |
| DMA_mem | - | - |

Caution: When you enable PDMA0 or PDMA1, verify the CLKGATE status.

8.3 Register Description

Most of the Special Function Registers (SFRs) are read-only. The main role of SFR is to verify the PL330 status. There are many SFRs for PL330. This section describes only the Exynos 4412-specific SFRs. Refer to Chapter 3, "PL330 TRM," for more information.

8.3.1 Register Map Summary

- Base Address: 0x1284_0000 (MDMA)

| Register | Offset | Description | Reset Value |
|-----------|------------------|--|-------------|
| DS | 0x0000 | Specifies the DMA status register. Refer to page 3-11 of "PL330 TRM" for more information. | 0x200 |
| DPC | 0x0004 | Specifies the DMA program counter register. Refer to page 3-13 of "PL330 TRM" for more information. | 0x0 |
| RSVD | 0x0008 to 0x001C | Reserved | Undefined |
| INTEN | 0x0020 | Specifies the interrupt enable register. Refer to page 3-13 of "PL330 TRM" for more information. | 0x0 |
| ES | 0x0024 | Specifies the event status register. Refer to page 3-14 of "PL330 TRM" for more information. | 0x0 |
| INTSTATUS | 0x0028 | Specifies the interrupt status register. Refer to page 3-16 of "PL330 TRM" for more information. | 0x0 |
| INTCLR | 0x002C | Specifies the interrupt clear register. Refer to page 3-17 of "PL330 TRM" for more information. | 0x0 |
| FSM | 0x0030 | Specifies the fault status DMA manager register. Refer to page 3-18 of "PL330 TRM" for more information. | 0x0 |
| FSC | 0x0034 | Specifies the fault status DMA channel register. Refer to page 3-19 of "PL330 TRM" for more information. | 0x0 |
| FTM | 0x0038 | Specifies the fault type DMA manager register. Refer to page 3-20 of "PL330 TRM" for more information. | 0x0 |
| RSVD | 0x003C | Reserved | Undefined |
| FTC0 | 0x0040 | Specifies the fault type for DMA channel 0. | 0x0 |
| FTC1 | 0x0044 | Specifies the fault type for DMA channel 1. | 0x0 |
| FTC2 | 0x0048 | Specifies the fault type for DMA channel 2. | 0x0 |
| FTC3 | 0x004C | Specifies the fault type for DMA channel 3. | 0x0 |
| FTC4 | 0x0050 | Specifies the fault type for DMA channel 4. | 0x0 |
| FTC5 | 0x0054 | Specifies the fault type for DMA channel 5. | 0x0 |
| FTC6 | 0x0058 | Specifies the fault type for DMA channel 6. | 0x0 |
| FTC7 | 0x005C | Specifies the fault type for DMA channel 7. | 0x0 |
| RSVD | 0x0060 to 0x00FC | Reserved | Undefined |
| CS0 | 0x0100 | Specifies the channel status for DMA channel 0. | 0x0 |
| CS1 | 0x0108 | Specifies the channel status for DMA channel 1. | 0x0 |

| Register | Offset | Description | Reset Value |
|----------|------------------|--|-------------|
| CS2 | 0x0110 | Specifies the channel status for DMA channel 2. | 0x0 |
| CS3 | 0x0118 | Specifies the channel status for DMA channel 3. | 0x0 |
| CS4 | 0x0120 | Specifies the channel status for DMA channel 4. | 0x0 |
| CS5 | 0x0128 | Specifies the channel status for DMA channel 5. | 0x0 |
| CS6 | 0x0130 | Specifies the channel status for DMA channel 6. | 0x0 |
| CS7 | 0x0138 | Specifies the channel status for DMA channel 7. | 0x0 |
| CPC0 | 0x0104 | Specifies the channel PC for DMA channel 0. | 0x0 |
| CPC1 | 0x010C | Specifies the channel PC for DMA channel 1. | 0x0 |
| CPC2 | 0x0114 | Specifies the channel PC for DMA channel 2. | 0x0 |
| CPC3 | 0x011C | Specifies the channel PC for DMA channel 3. | 0x0 |
| CPC4 | 0x0124 | Specifies the channel PC for DMA channel 4. | 0x0 |
| CPC5 | 0x012C | Specifies the channel PC for DMA channel 5. | 0x0 |
| CPC6 | 0x0134 | Specifies the channel PC for DMA channel 6. | 0x0 |
| CPC7 | 0x013C | Specifies the channel PC for DMA channel 7. | 0x0 |
| RSVD | 0x0140 to 0x03FC | Reserved | Undefined |
| SA_0 | 0x0400 | Specifies the source address for DMA channel 0. | 0x0 |
| SA_1 | 0x0420 | Specifies the source address for DMA channel 1. | 0x0 |
| SA_2 | 0x0440 | Specifies the source address for DMA channel 2. | 0x0 |
| SA_3 | 0x0460 | Specifies the source address for DMA channel 3. | 0x0 |
| SA_4 | 0x0480 | Specifies the source address for DMA channel 4. | 0x0 |
| SA_5 | 0x04A0 | Specifies the source address for DMA channel 5. | 0x0 |
| SA_6 | 0x04C0 | Specifies the source address for DMA channel 6. | 0x0 |
| SA_7 | 0x04E0 | Specifies the source address for DMA channel 7. | 0x0 |
| DA_0 | 0x0404 | Specifies the destination address for DMA channel 0. | 0x0 |
| DA_1 | 0x0424 | Specifies the destination address for DMA channel 1. | 0x0 |
| DA_2 | 0x0444 | Specifies the destination address for DMA channel 2. | 0x0 |
| DA_3 | 0x0464 | Specifies the destination address for DMA channel 3. | 0x0 |
| DA_4 | 0x0484 | Specifies the destination address for DMA channel 4. | 0x0 |
| DA_5 | 0x04A4 | Specifies the destination address for DMA channel 5. | 0x0 |
| DA_6 | 0x04C4 | Specifies the destination address for DMA channel 6. | 0x0 |
| DA_7 | 0x04E4 | Specifies the destination address for DMA channel 7. | 0x0 |
| CC_0 | 0x0408 | Specifies the channel control for DMA channel 0. | 0x00800200 |
| CC_1 | 0x0428 | Specifies the channel control for DMA channel 1. | 0x00800200 |
| CC_2 | 0x0448 | Specifies the channel control for DMA channel 2. | 0x00800200 |
| CC_3 | 0x0468 | Specifies the channel control for DMA channel 3. | 0x00800200 |
| CC_4 | 0x0488 | Specifies the channel control for DMA channel 4. | 0x00800200 |

| Register | Offset | Description | Reset Value |
|-----------|------------------|---|-------------|
| CC_5 | 0x04A8 | Specifies the channel control for DMA channel 5. | 0x00800200 |
| CC_6 | 0x04C8 | Specifies the channel control for DMA channel 6. | 0x00800200 |
| CC_7 | 0x04E8 | Specifies the channel control for DMA channel 7. | 0x00800200 |
| LC0_0 | 0x040C | Specifies the loop counter 0 for DMA channel 0. | 0x0 |
| LC0_1 | 0x042C | Specifies the loop counter 0 for DMA channel 1. | 0x0 |
| LC0_2 | 0x044C | Specifies the loop counter 0 for DMA channel 2. | 0x0 |
| LC0_3 | 0x046C | Specifies the loop counter 0 for DMA channel 3. | 0x0 |
| LC0_4 | 0x048C | Specifies the loop counter 0 for DMA channel 4. | 0x0 |
| LC0_5 | 0x04AC | Specifies the loop counter 0 for DMA channel 5. | 0x0 |
| LC0_6 | 0x04CC | Specifies the loop counter 0 for DMA channel 6. | 0x0 |
| LC0_7 | 0x04EC | Specifies the loop counter 0 for DMA channel 7. | 0x0 |
| LC1_0 | 0x0410 | Specifies the loop counter 1 for DMA channel 0. | 0x0 |
| LC1_1 | 0x0430 | Specifies the loop counter 1 for DMA channel 1. | 0x0 |
| LC1_2 | 0x0450 | Specifies the loop counter 1 for DMA channel 2. | 0x0 |
| LC1_3 | 0x0470 | Specifies the loop counter 1 for DMA channel 3. | 0x0 |
| LC1_4 | 0x0490 | Specifies the loop counter 1 for DMA channel 4. | 0x0 |
| LC1_5 | 0x04B0 | Specifies the loop counter 1 for DMA channel 5. | 0x0 |
| LC1_6 | 0x04D0 | Specifies the loop counter 1 for DMA channel 6. | 0x0 |
| LC1_7 | 0x04F0 | Specifies the loop counter 1 for DMA channel 7. | 0x0 |
| RSVD | 0x0414 to 0x041C | Reserved | Undefined |
| RSVD | 0x0434 to 0x043C | Reserved | Undefined |
| RSVD | 0x0454 to 0x045C | Reserved | Undefined |
| RSVD | 0x0474 to 0x047C | Reserved | Undefined |
| RSVD | 0x0494 to 0x049C | Reserved | Undefined |
| RSVD | 0x04B4 to 0x04BC | Reserved | Undefined |
| RSVD | 0x04D4 to 0x04DC | Reserved | Undefined |
| RSVD | 0x04F4 to 0x0CFC | Reserved | Undefined |
| DBGSTATUS | 0x0D00 | Specifies the debug status register. Refer to page 3-37 of "PL330 TRM" for more information. | 0x0 |
| DBGCMD | 0x0D04 | Specifies the debug command register. Refer to page 3-37 of "PL330 TRM" for more information. | Undefined |

| Register | Offset | Description | Reset Value |
|-------------|------------------|--|-------------------------|
| DBGINST0 | 0x0D08 | Specifies the debug instruction-0 register. Refer to page 3-38 of "PL330 TRM" for more information. | Undefined |
| DBGINST1 | 0x0D0C | Specifies the debug instruction-1 register. Refer to page 3-39 of "PL330 TRM" for more information. | Undefined |
| CR0 | 0x0E00 | Specifies the configuration register 0. Refer to page 3-40 of "PL330 TRM" for more information. | 0x003E_0075 |
| CR1 | 0x0E04 | Specifies the configuration register 1. Refer to page 3-42 of "PL330 TRM" for more information. | 0x0000_0075 |
| CR2 | 0x0E08 | Specifies the configuration register 2. Refer to page 3-43 of "PL330 TRM" for more information. | 0x0 |
| CR3 | 0x0E0C | Specifies the configuration register 3. Refer to page 3-44 of "PL330 TRM" for more information. | 0x0 |
| CR4 | 0x0E10 | Specifies the configuration register 4. Refer to page 3-45 of "PL330 TRM" for more information. | 0x0000_0001 |
| CRDn | 0x0E14 | Specifies the configuration register Dn. Refer to page 3-46 of "PL330 TRM" for more information. | 0x03F7_3733 |
| periph_id_n | 0x0FE0 to 0x0FEC | Specifies the peripheral identification registers 0-3. Refer to page 3-48 of "PL330 TRM" for more information. | Configuration-dependent |
| pcell_id_n | 0x0FF0 to 0x0FFC | Specifies the primecell identification registers 0-3. Refer to page 3-50 of "PL330 TRM" for more information. | Configuration-dependent |

- Base Address: 0x1268_0000, 0x1269_0000 (PDMA0/PDMA1)

| Register | Offset | Description | Reset Value |
|-----------|------------------|--|-------------|
| DS | 0x0000 | Specifies the DMA status register. Refer to page 3-11 of "PL330 TRM" for more information. | 0x00000200 |
| DPC | 0x0004 | Specifies the DMA program counter register. Refer to page 3-13 of "PL330 TRM" for more information. | 0x0 |
| RSVD | 0x0008 to 0x001C | Reserved | Undefined |
| INTEN | 0x0020 | Specifies the interrupt enable register. Refer to page 3-13 of "PL330 TRM" for more information. | 0x0 |
| ES | 0x0024 | Specifies the event status register. Refer to page 3-14 of "PL330 TRM" for more information. | 0x0 |
| INTSTATUS | 0x0028 | Specifies the interrupt status register. Refer to page 3-16 of "PL330 TRM" for more information. | 0x0 |
| INTCLR | 0x002C | Specifies the interrupt clear register. Refer to page 3-17 of "PL330 TRM" for more information. | 0x0 |
| FSM | 0x0030 | Specifies the fault status DMA manager register. Refer to page 3-18 of "PL330 TRM" for more information. | 0x0 |
| FSC | 0x0034 | Specifies the fault status DMA channel register. Refer to page 3-19 of "PL330 TRM" for more information. | 0x0 |
| FTM | 0x0038 | Specifies the fault type DMA manager register. Refer to page 3-20 of "PL330 TRM" for more information. | 0x0 |
| RSVD | 0x003C | Reserved | Undefined |
| FTC0 | 0x0040 | Specifies the fault type for DMA channel 0. | 0x0 |
| FTC1 | 0x0044 | Specifies the fault type for DMA channel 1. | 0x0 |
| FTC2 | 0x0048 | Specifies the fault type for DMA channel 2. | 0x0 |
| FTC3 | 0x004C | Specifies the fault type for DMA channel 3. | 0x0 |
| FTC4 | 0x0050 | Specifies the fault type for DMA channel 4. | 0x0 |
| FTC5 | 0x0054 | Specifies the fault type for DMA channel 5. | 0x0 |
| FTC6 | 0x0058 | Specifies the fault type for DMA channel 6. | 0x0 |
| FTC7 | 0x005C | Specifies the fault type for DMA channel 7. | 0x0 |
| RSVD | 0x0060 to 0x00FC | Reserved | Undefined |
| CS0 | 0x0100 | Specifies the channel status for DMA channel 0. | 0x0 |
| CS1 | 0x0108 | Specifies the channel status for DMA channel 1. | 0x0 |
| CS2 | 0x0110 | Specifies the channel status for DMA channel 2. | 0x0 |
| CS3 | 0x0118 | Specifies the channel status for DMA channel 3. | 0x0 |
| CS4 | 0x0120 | Specifies the channel status for DMA channel 4. | 0x0 |
| CS5 | 0x0128 | Specifies the channel status for DMA channel 5. | 0x0 |
| CS6 | 0x0130 | Specifies the channel status for DMA channel 6. | 0x0 |
| CS7 | 0x0138 | Specifies the channel status for DMA channel 7. | 0x0 |

| Register | Offset | Description | Reset Value |
|----------|------------------|--|-------------|
| CPC0 | 0x0104 | Specifies the channel PC for DMA channel 0. | 0x0 |
| CPC1 | 0x010C | Specifies the channel PC for DMA channel 1. | 0x0 |
| CPC2 | 0x0114 | Specifies the channel PC for DMA channel 2. | 0x0 |
| CPC3 | 0x011C | Specifies the channel PC for DMA channel 3. | 0x0 |
| CPC4 | 0x0124 | Specifies the channel PC for DMA channel 4. | 0x0 |
| CPC5 | 0x012C | Specifies the channel PC for DMA channel 5. | 0x0 |
| CPC6 | 0x0134 | Specifies the channel PC for DMA channel 6. | 0x0 |
| CPC7 | 0x013C | Specifies the channel PC for DMA channel 7. | 0x0 |
| RSVD | 0x0140 to 0x03FC | Reserved | Undefined |
| SA_0 | 0x0400 | Specifies the source address for DMA channel 0. | 0x0 |
| SA_1 | 0x0420 | Specifies the source address for DMA channel 1. | 0x0 |
| SA_2 | 0x0440 | Specifies the source address for DMA channel 2. | 0x0 |
| SA_3 | 0x0460 | Specifies the source address for DMA channel 3. | 0x0 |
| SA_4 | 0x0480 | Specifies the source address for DMA channel 4. | 0x0 |
| SA_5 | 0x04A0 | Specifies the source address for DMA channel 5. | 0x0 |
| SA_6 | 0x04C0 | Specifies the source address for DMA channel 6. | 0x0 |
| SA_7 | 0x04E0 | Specifies the source address for DMA channel 7. | 0x0 |
| DA_0 | 0x0404 | Specifies the destination address for DMA channel 0. | 0x0 |
| DA_1 | 0x0424 | Specifies the destination address for DMA channel 1. | 0x0 |
| DA_2 | 0x0444 | Specifies the destination address for DMA channel 2. | 0x0 |
| DA_3 | 0x0464 | Specifies the destination address for DMA channel 3. | 0x0 |
| DA_4 | 0x0484 | Specifies the destination address for DMA channel 4. | 0x0 |
| DA_5 | 0x04A4 | Specifies the destination address for DMA channel 5. | 0x0 |
| DA_6 | 0x04C4 | Specifies the destination address for DMA channel 6. | 0x0 |
| DA_7 | 0x04E4 | Specifies the destination address for DMA channel 7. | 0x0 |
| CC_0 | 0x0408 | Specifies the channel control for DMA channel 0. | 0x0 |
| CC_1 | 0x0428 | Specifies the channel control for DMA channel 1. | 0x0 |
| CC_2 | 0x0448 | Specifies the channel control for DMA channel 2. | 0x0 |
| CC_3 | 0x0468 | Specifies the channel control for DMA channel 3. | 0x0 |
| CC_4 | 0x0488 | Specifies the channel control for DMA channel 4. | 0x0 |
| CC_5 | 0x04A8 | Specifies the channel control for DMA channel 5. | 0x0 |
| CC_6 | 0x04C8 | Specifies the channel control for DMA channel 6. | 0x0 |
| CC_7 | 0x04E8 | Specifies the channel control for DMA channel 7. | 0x0 |
| LC0_0 | 0x040C | Specifies the loop counter 0 for DMA channel 0. | 0x0 |
| LC0_1 | 0x042C | Specifies the loop counter 0 for DMA channel 1. | 0x0 |
| LC0_2 | 0x044C | Specifies the loop counter 0 for DMA channel 2. | 0x0 |

| Register | Offset | Description | Reset Value |
|-----------|------------------|---|-------------|
| LC0_3 | 0x046C | Specifies the loop counter 0 for DMA channel 3. | 0x0 |
| LC0_4 | 0x048C | Specifies the loop counter 0 for DMA channel 4. | 0x0 |
| LC0_5 | 0x04AC | Specifies the loop counter 0 for DMA channel 5. | 0x0 |
| LC0_6 | 0x04CC | Specifies the loop counter 0 for DMA channel 6. | 0x0 |
| LC0_7 | 0x04EC | Specifies the loop counter 0 for DMA channel 7. | 0x0 |
| LC1_0 | 0x0410 | Specifies the loop counter 1 for DMA channel 0. | 0x0 |
| LC1_1 | 0x0430 | Specifies the loop counter 1 for DMA channel 1. | 0x0 |
| LC1_2 | 0x0450 | Specifies the loop counter 1 for DMA channel 2. | 0x0 |
| LC1_3 | 0x0470 | Specifies the loop counter 1 for DMA channel 3. | 0x0 |
| LC1_4 | 0x0490 | Specifies the loop counter 1 for DMA channel 4. | 0x0 |
| LC1_5 | 0x04B0 | Specifies the loop counter 1 for DMA channel 5. | 0x0 |
| LC1_6 | 0x04D0 | Specifies the loop counter 1 for DMA channel 6. | 0x0 |
| LC1_7 | 0x04F0 | Specifies the loop counter 1 for DMA channel 7. | 0x0 |
| RSVD | 0x0414 to 0x041C | Reserved | Undefined |
| RSVD | 0x0434 to 0x043C | Reserved | Undefined |
| RSVD | 0x0454 to 0x045C | Reserved | Undefined |
| RSVD | 0x0474 to 0x047C | Reserved | Undefined |
| RSVD | 0x0494 to 0x049C | Reserved | Undefined |
| RSVD | 0x04B4 to 0x04BC | Reserved | Undefined |
| RSVD | 0x04D4 to 0x04DC | Reserved | Undefined |
| RSVD | 0x04F4 to 0x0CFC | Reserved | Undefined |
| DBGSTATUS | 0x0D00 | Specifies the debug status register on page 3-37 of "TRM". | 0x0 |
| DBGCMD | 0x0D04 | Specifies the debug command register. Refer to page 3-37 of "PL330 TRM" for more information. | Undefined |
| DBGINST0 | 0x0D08 | Specifies the debug instruction-0 register. Refer to page 3-38 of "PL330 TRM" for more information. | Undefined |
| DBGINST1 | 0x0D0C | Specifies the debug instruction-1 register. Refer to page 3-39 of "pl330 TRM" for more information. | Undefined |
| CR0 | 0x0E00 | Specifies the configuration register 0. Refer to page 3-40 of "PL330 TRM" for more information. | 0x003F_F075 |
| CR1 | 0x0E04 | Specifies the configuration register 1. Refer to page 3-42 of "PL330 TRM" for more information. | 0x0000_0074 |

| Register | Offset | Description | Reset Value |
|-------------|------------------|---|-------------------------|
| CR2 | 0x0E08 | Specifies the configuration register 2. Refer to page 3-43 of "PL330 TRM" for more information. | 0x0000_0000 |
| CR3 | 0x0E0C | Specifies the configuration register 3. Refer to page 3-44 of "PL330 TRM" for more information. | 0x0 |
| CR4 | 0x0E10 | Specifies the configuration register 4. Refer to page 3-45 of "PL330 TRM" for more information. | 0xFFFF_FFFF |
| CRDn | 0x0E14 | Specifies the configuration register Dn. Refer to page 3-46 of "PL330 TRM" for more information. | 0x01F7_3732 |
| periph_id_n | 0x0FE0 to 0x0FEC | Specifies the peripheral identification registers 0-3 Refer to page 3-48 of "PL330 TRM" for more information. | Configuration-dependent |
| pcell_id_n | 0x0FF0 to 0x0FFC | Specifies the primecell identification registers 0-3. Refer to page 3-50 of "PL330 TRM" for more information. | Configuration-dependent |

NOTE: The SFR description shows only the restricted and fixed part of some SFR. PL330 TRM shows detailed information of other parts and other SFRs.

8.3.1.1 CC

CC includes:

- Base Address: 0x1284_0000 (MDMA)
- Base Address: 0x1268_0000, 0x1269_0000 (PDMA0/PDMA1)
- Address = Base Address + 0x0408, Reset Value = 0x0080_0200 (CC_0)
- Address = Base Address + 0x0428, Reset Value = 0x0080_0200 (CC_1)
- Address = Base Address + 0x0448, Reset Value = 0x0080_0200 (CC_2)
- Address = Base Address + 0x0468, Reset Value = 0x0080_0200 (CC_3)
- Address = Base Address + 0x0488, Reset Value = 0x0080_0200 (CC_4)
- Address = Base Address + 0x04A8, Reset Value = 0x0080_0200 (CC_5)
- Address = Base Address + 0x04C8, Reset Value = 0x0080_0200 (CC_6)
- Address = Base Address + 0x04E8, Reset Value = 0x0080_0200 (CC_7)

8.4 Instruction

Please refer to the PL330 TRM, "AMBA DMA Controller DMA-330 technical reference manua revision r1p0" from ARM®.

8.4.1.1 Security Scheme

DMA_mem runs in both secure and non-secure modes, while DMA_peri runs in non-secure mode only.

8.4.1.2 Summary

1. You can configure the DMAC with up to eight DMA channels. Each channel supports single concurrent thread of DMA operation. Additionally, there is a single DMA manager thread to initialize the DMA channel thread.
2. Channel thread
 - Each channel thread can operate the DMA. Accordingly, write an assembly code. If you require a number of independent DMA channels, write a number of assembly codes for each channel.
 - Assemble and link the codes into one file and load this file into the memory.

9 SROM Controller

9.1 Overview

Exynos 4412 SROM Controller (SROMC) supports:

- External 8/16-bit NOR Flash/PROM/SRAM memory.
- 4-bank memory up to maximum 128 Kbyte per bank.

9.2 Features

The features of SROMC are:

- Supports SRAM, various ROMs, and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Up to 128 KB per bank
- Supports 4-bank.
- Fixed memory bank start address
- External wait to extend the bus cycle
- Supports byte and half-word access for external memory

9.3 Block Diagram

[Figure 9-1](#) illustrates the block diagram of SROMC introduction.

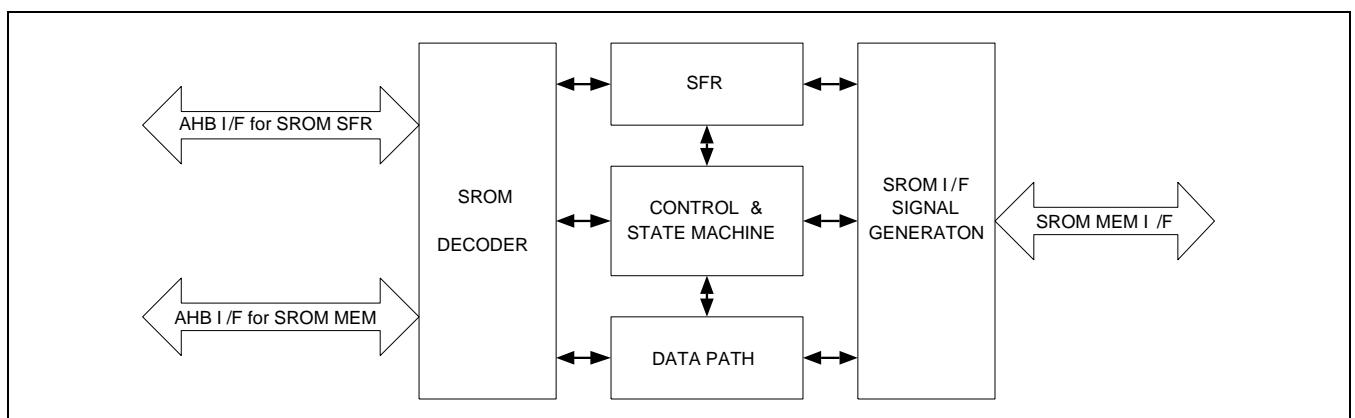


Figure 9-1 Block Diagram of SROMC Introduction

9.4 Functional Description

SROMC supports SROM interface for Bank 0 to Bank 3. This section includes:

- nWAIT Pin Operation
- Programmable Access Cycle

9.4.1 nWAIT Pin Operation

When it enables nWAIT signal corresponding to each memory bank, the external nWAIT pin should prolong the duration of nOE while the memory bank is active. It verifies the nWAIT from tacc-1 and deasserts the nOE at the next clock after sampling nWAIT is high. The nWE signal has the similar relation with nOE signal.

[Figure 9-2](#) illustrates the SROMC nWAIT timing diagram.

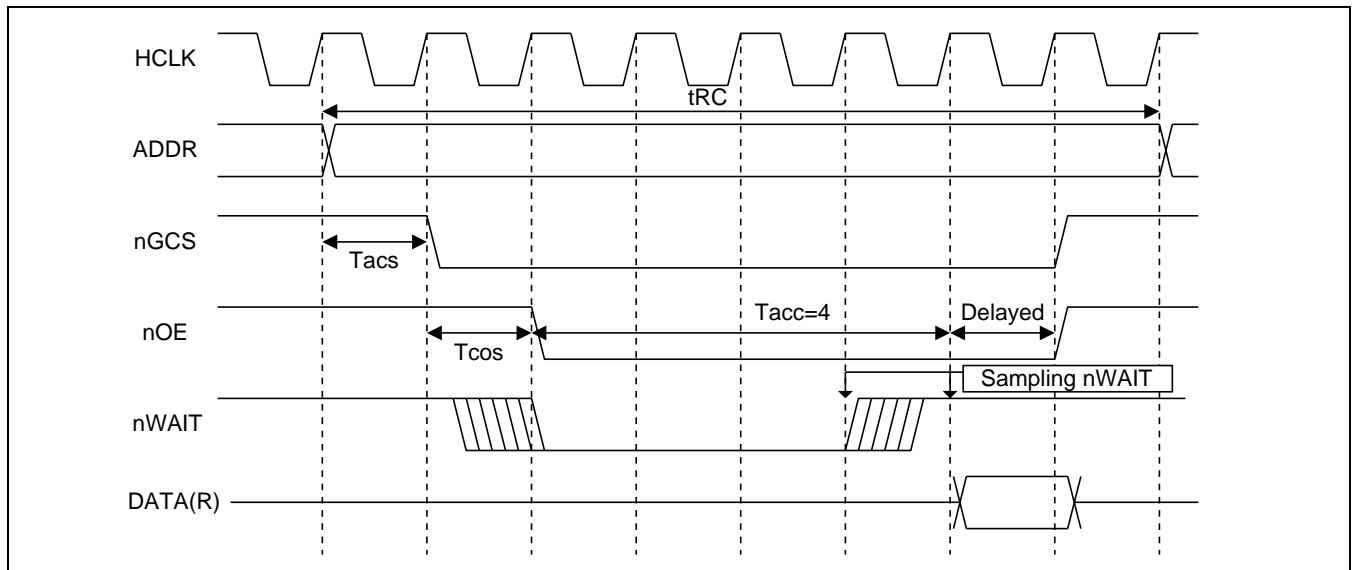


Figure 9-2 SROMC nWAIT Timing Diagram

9.4.2 Programmable Access Cycle

[Figure 9-3](#) illustrates the SROMC read timing diagram.

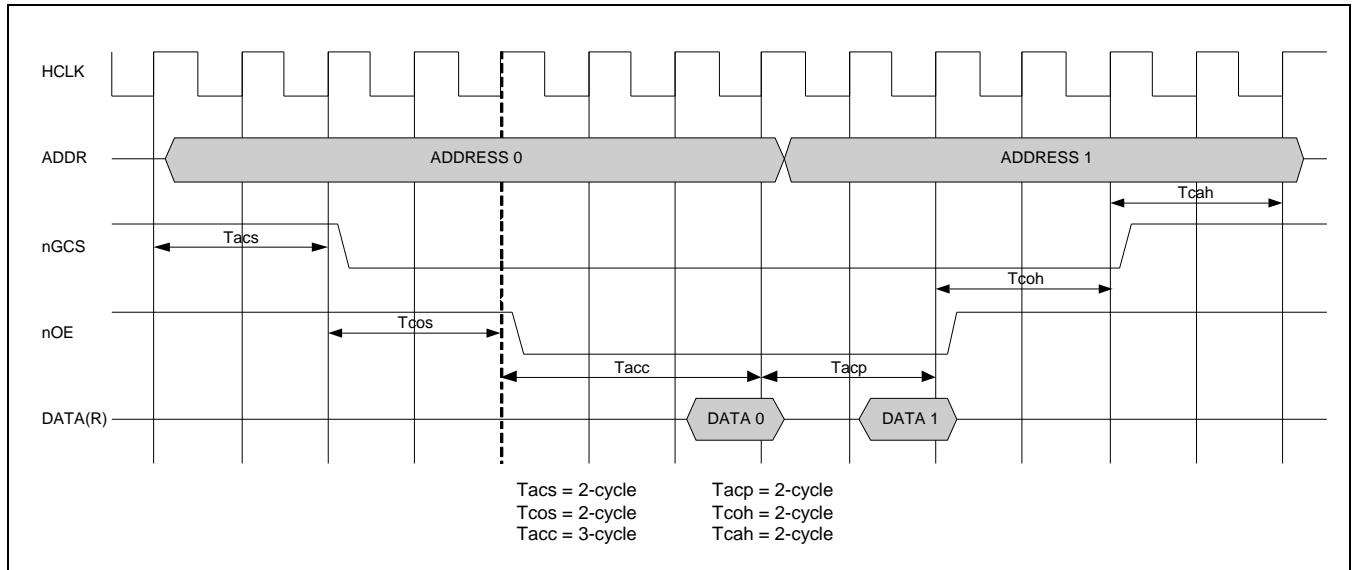


Figure 9-3 SROMC Read Timing Diagram

[Figure 9-4](#) illustrates the SROMC write timing diagram.

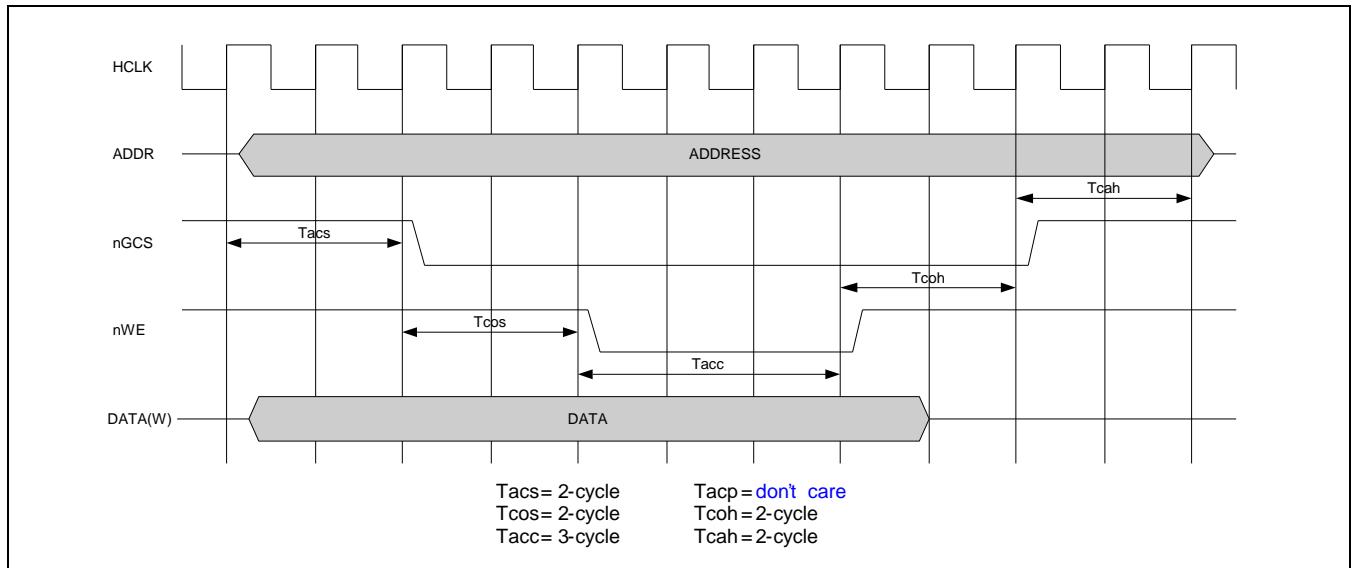


Figure 9-4 SROMC Write Timing Diagram

9.5 I/O Description

This section describes the I/O description of SROMC.

| Signal | I/O | Description | Pad | Type |
|---------------|--------|------------------------------------|-----------|-------|
| nGCS[3:0] | Output | Bank selection signal | Xm0CSn_x | muxed |
| ADDR[15:0] | Output | SROM address bus | Xm0ADDR_x | muxed |
| nOE | Output | SROM output enable | Xm0OEn | muxed |
| nWE | Output | SROM write enable | Xm0WEn | muxed |
| nWBE/nBE[1:0] | Output | SROM byte write enable/byte enable | Xm0BEn_x | muxed |
| DATA[15:0] | In/Out | SROM data bus | Xm0DATA_x | muxed |
| nWAIT | Input | SROM wait input | Xm0WAITn | muxed |

9.6 Register Description

9.6.1 Register Map Summary

- Base Address: 0x1257_0000

| Register | Offset | Description | Reset Value |
|----------|--------|---|-------------|
| SROM_BW | 0x0000 | Specifies the SROM bus width and wait control | 0x0000_0009 |
| SROM_BC0 | 0x0004 | Specifies the SROM bank 0 control register | 0x000F_0000 |
| SROM_BC1 | 0x0008 | Specifies the SROM bank 1 control register | 0x000F_0000 |
| SROM_BC2 | 0x000C | Specifies the SROM bank 2 control register | 0x000F_0000 |
| SROM_BC3 | 0x0010 | Specifies the SROM bank 3 control register | 0x000F_0000 |

9.6.1.1 SROM_BW

- Base Address: 0x1257_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0009

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| RSVD | [31:16] | — | Reserved | 0 |
| ByteEnable3 | [15] | RW | nWBE/nBE (for UB/LB) control for memory bank 3 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]) | 0 |
| WaitEnable3 | [14] | RW | Wait enable control for memory bank 3 0 = Disables WAIT 1 = Enables WAIT | 0 |
| AddrMode3 | [13] | RW | Select SROM ADDR base for memory bank 3 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth3 is "0", SROM_ADDR is byte base address. (It ignores this bit.) | 0 |
| DataWidth3 | [12] | RW | Data bus width control for memory bank 3 0 = 8-bit 1 = 16-bit | 0 |
| ByteEnable2 | [11] | RW | nWBE/nBE (for UB/LB) control for memory bank 2 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]) | 0 |
| WaitEnable2 | [10] | RW | Wait enable control for memory bank 2 0 = Disables WAIT 1 = Enables WAIT | 0 |
| AddrMode2 | [9] | RW | Select SROM ADDR Base for memory bank 2 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth2 is "0", SROM_ADDR is byte base address. (It ignores this bit.) | 0 |
| DataWidth2 | [8] | RW | Data bus width control for memory bank 2 0 = 8-bit 1 = 16-bit | 0 |
| ByteEnable1 | [7] | RW | nWBE/nBE (for UB/LB) control for memory bank 1 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]) | 0 |
| WaitEnable1 | [6] | RW | Wait enable control for memory bank 1 | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|---|-------------|
| | | | 0 = Disables WAIT 1 = Enables WAIT | |
| AddrMode1 | [5] | RW | Select SROM ADDR base for memory bank 1 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth1 is "0", SROM_ADDR is byte base address. (It ignores this bit.) | 0 |
| DataWidth1 | [4] | RW | Data bus width control for memory bank 1 0 = 8-bit 1 = 16-bit | 0 |
| ByteEnable0 | [3] | RW | nWBE/nBE (for UB/LB) control for memory bank 0 0 = Does not use UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Uses UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]) | 1 |
| WaitEnable0 | [2] | RW | Wait enable control for memory bank 0 0 = Disables WAIT 1 = Enables WAIT | 0 |
| AddrMode0 | [1] | RW | Select SROM ADDR base for memory bank 0 0 = SROM_ADDR is half-word base address. (SROM_ADDR[22:0] ← HADDR[23:1]) 1 = SROM_ADDR is byte base address (SROM_ADDR[22:0] ← HADDR[22:0]) NOTE: When DataWidth0 is "0", SROM_ADDR is byte base address. (It ignores this bit.) | 0 |
| DataWidth0 | [0] | RW | Data bus width control for memory bank 0 0 = 8-bit 1 = 16-bit | 1 |

9.6.1.2 SROM_BCn (n = 0 to 3)

- Base Address: 0x1257_0000
- Address = Base Address + 0x0004, Reset Value = 0x000F_0000 (SROM_BC0)
- Address = Base Address + 0x0008, Reset Value = 0x000F_0000 (SROM_BC1)
- Address = Base Address + 0x000C, Reset Value = 0x000F_0000 (SROM_BC2)
- Address = Base Address + 0x0010, Reset Value = 0x000F_0000 (SROM_BC3)

| Name | Bit | Type | Description | Reset Value |
|------|---------|------|---|-------------|
| Tacs | [31:28] | RW | Address set-up before nGCS 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks NOTE: More 1-2 cycles according to bus i/f status | 0000 |
| Tcos | [27:24] | RW | Chip selection set-up before nOE 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks | 0000 |
| RSVD | [23:21] | - | Reserved | 000 |
| Tacc | [20:16] | RW | Access cycle 00000 = 1 Clock 00001 = 2 Clocks 00001 = 3 Clocks 00010 = 4 Clocks 11100 = 29 Clocks 11101 = 30 Clocks 11110 = 31 Clocks 11111 = 32 Clocks | 01111 |
| Tcoh | [15:12] | RW | Chip selection hold on nOE 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks 1100 = 12 Clocks 1101 = 13 Clocks | 0000 |

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|---|-------------|
| | | | 1110 = 14 Clocks 1111 = 15 Clocks | |
| Tcah | [11:8] | RW | Address holding time after nGCSn 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks NOTE: More 1-2 cycles according to bus i/f status | 0000 |
| Tacp | [7:4] | RW | Page mode access cycle at Page mode 0000 = 0 Clock 0001 = 1 Clocks 0010 = 2 Clocks 0011 = 3 Clocks 1100 = 12 Clocks 1101 = 13 Clocks 1110 = 14 Clocks 1111 = 15 Clocks | 0000 |
| RSVD | [3:2] | - | Reserved | - |
| PMC | [1:0] | RW | Page mode configuration 00 = Normal (1 Data) 01 = 4 Data 10 = Reserved 11 = Reserved | 00 |

10 NAND Flash Controller

10.1 Overview

Due to the recent increase in the prices of NOR flash memory and the moderately priced DRAM, and NAND flash, customers prefer to execute boot code on NAND flash and execute the main code on DRAM.

The boot code in Exynos 4412 can be executed on external NAND flash. It copies NAND flash data to DRAM. To validate the NAND flash data, Exynos 4412 includes hardware Error Correction Code (ECC). After the NAND flash content is copied to DRAM, main program will be executed on DRAM.

10.2 Features

The features of NAND flash controller are:

- Auto boot: The boot code is transferred to internal SRAM during reset. After the transfer, the boot code will be executed on the SRAM.
- NAND flash memory interface: Supports 512 Bytes, 2 KB, 4 KB, and 8 KB pages.
- Software mode: You can directly access NAND flash memory, for example, this feature can be used in read/erase/program NAND flash memory.
- Interface: Supports 8-bit NAND flash memory interface bus.
- Generates, detects, and indicates hardware ECC (software correction).
- Supports both Single Level Cell (SLC) and Multi Level Cell (MLC) NAND flash memories.
- ECC: Supports 1-/4-/8-/12-/16-bit ECC.
- SFR interface: Supports byte/half word/word access to Data and ECC data registers, and Word access to other registers.

10.3 Functional Description

10.3.1 Block Diagram

[Figure 10-1](#) illustrates the NAND Flash Controller block diagram.

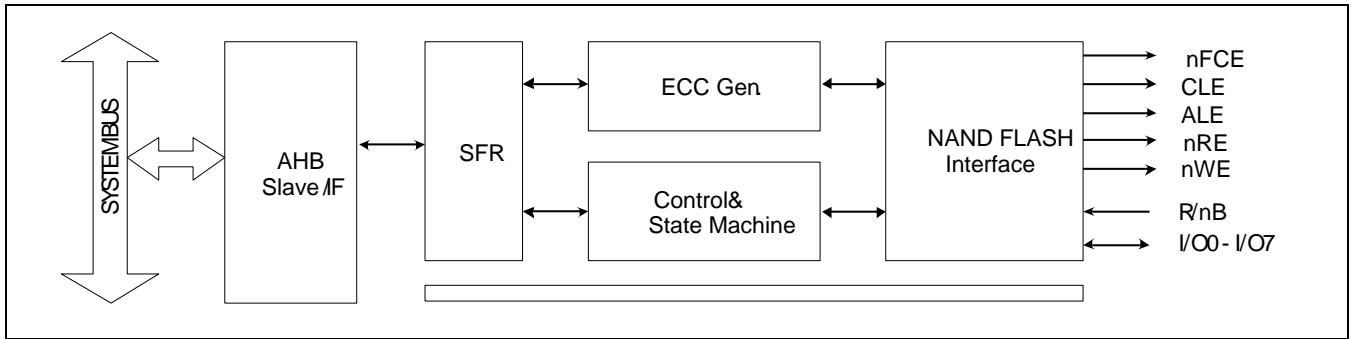


Figure 10-1 NAND Flash Controller Block Diagram

10.3.2 NAND Flash Memory Timing

[Figure 10-2](#) illustrates the CLE and ALE timing ($TACLS = 1$, $TWRPH0 = 0$, $TWRPH1 = 0$).

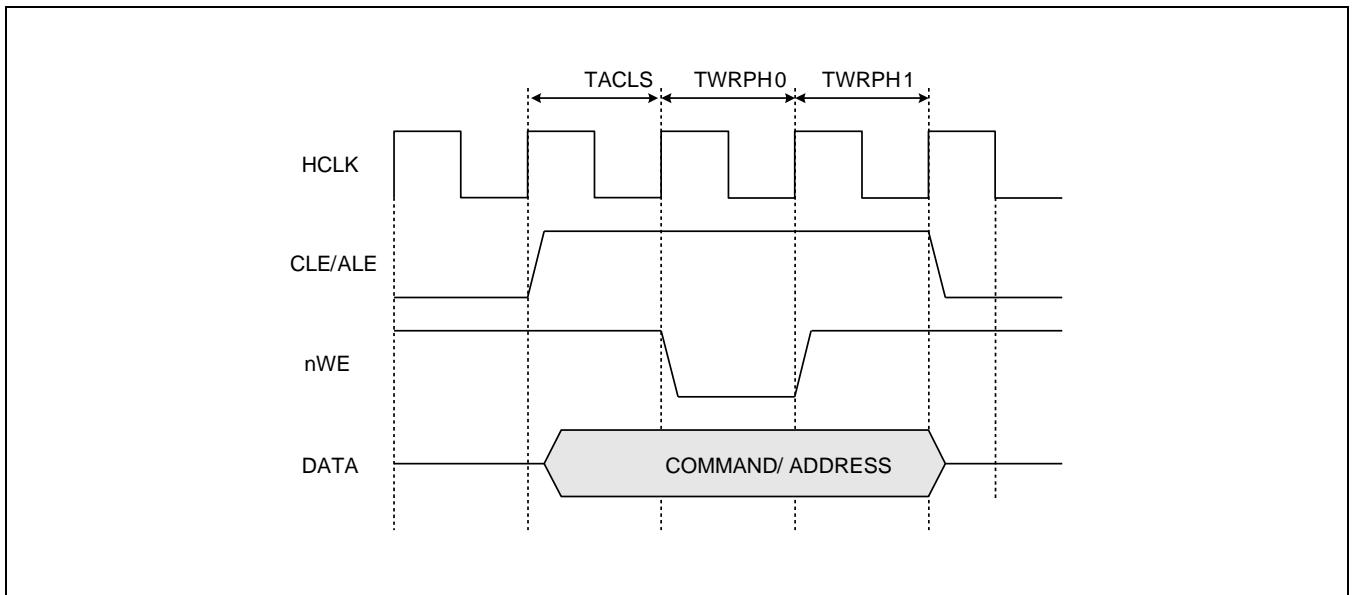


Figure 10-2 CLE and ALE Timing ($TACLS = 1$, $TWRPH0 = 0$, $TWRPH1 = 0$)

[Figure 10-3](#) illustrates the nWE and nRE timing ($TWRPH0 = 0$, $TWRPH1 = 0$).

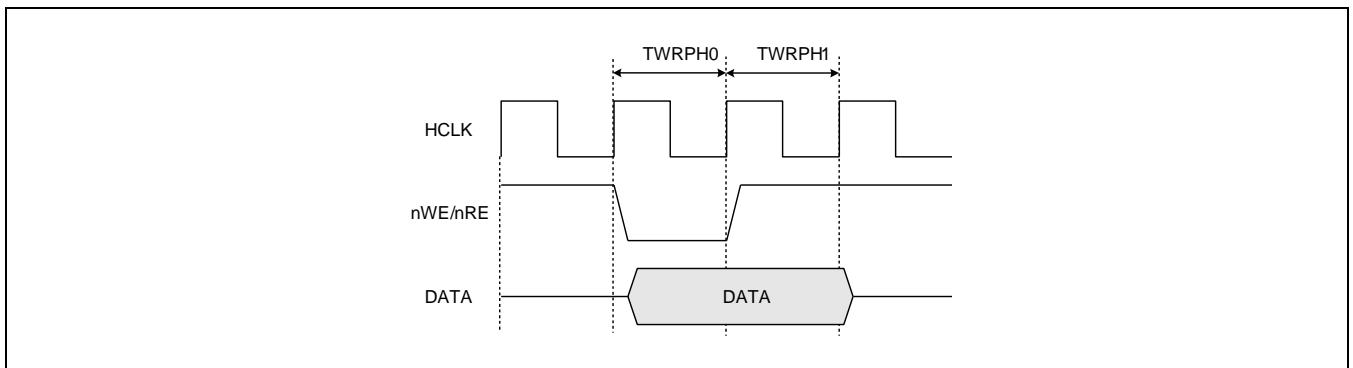


Figure 10-3 nWE and nRE Timing ($TWRPH0 = 0$, $TWRPH1 = 0$)

10.4 Software Mode

Exynos 4412 supports only software mode access. Use this mode to access NAND flash memory. The NAND flash controller supports direct access to interface with the NAND flash memory.

- Writing to the command register (NFCMMD) specifies the NAND flash memory command cycle
- Writing to the address register (NFADDR) specifies the NAND flash memory address cycle
- Writing to the data register (NFDATA) specifies write data to the NAND flash memory (Write cycle)
- Reading from the data register (NFDATA) specifies read data from the NAND flash memory (Read cycle)
- Reading main ECC registers (NFMECCD0/NFMECCD1) and Spare ECC registers (NFSECCD) specifies read data from the NAND flash memory

NOTE: In the software mode, use polling or interrupt to verify the RnB status input pin.

10.4.1 Data Register Configuration

10.4.1.1 8-bit NAND Flash Memory Interface

1. Word Access

| Register | Endian | Bit[31:24] | Bit[23:16] | Bit[15:8] | Bit[7:0] |
|----------|--------|--------------|--------------|--------------|--------------|
| NFDATA | Little | 4th I/O[7:0] | 3rd I/O[7:0] | 2nd I/O[7:0] | 1st I/O[7:0] |

2. Half-word Access

| Register | Endian | Bit[31:24] | Bit[23:16] | Bit[15:8] | Bit[7:0] |
|----------|--------|---------------|---------------|--------------|--------------|
| NFDATA | Little | Invalid value | Invalid value | 2nd I/O[7:0] | 1st I/O[7:0] |

3. Byte Access

| Register | Endian | Bit[31:24] | Bit[23:16] | Bit[15:8] | Bit[7:0] |
|----------|--------|---------------|---------------|---------------|--------------|
| NFDATA | Little | Invalid value | Invalid value | Invalid value | 1st I/O[7:0] |

10.4.2 1/4/8/12/16-bit ECC

NAND flash controller supports 1-/4-/8-/12-/16-bit ECC.

For 1-bit ECC, NAND flash controller includes ECC modules for main and spare (meta) data. Main data ECC module generates ECC parity code for 2048 bytes (maximum) data/message length, whereas spare (meta) data ECC module generates ECC parity code for 32 bytes (maximum).

For 4-bit ECC, NAND flash controller includes an ECC module. It generates 512 or 24 bytes of ECC parity code. Set MsgLength (NFCONF[25]) to select 512 or 24 bytes message length.

For 8-/12-/16-bit ECC, NAND flash controller includes ECC modules for each ECC. You can select data/message length for main and spare (meta) data length. Usually, the length of main data is 512 bytes and the length of spare (meta) data depends on user application.

Since these ECC modules support variable length of main and spare (meta) data, you should set the ECC parity conversion codes to handle free page. Refer to [10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC](#) for more information on ECC parity conversion codes. Free page specifies an erased page. The value of erased page is "0xff". Therefore, set the ECC parity conversion codes to generate "0xff" ECC parity codes for all '0xff' data. This setting allows ECC module to detect errors on a free page.

ECC parity codes are:

- 28-bit ECC Parity Code = 22-bit Line parity + 6-bit Column Parity
- 10-bit ECC Parity Code = 4-bit Line parity + 6-bit Column Parity

Each 1-/4-/8-/12-/16-bit ECC module guarantees up to 1-/4-/8-/12-/16-bit errors, respectively. If the errors cross the number of guaranteed errors, it cannot guarantee the result.

[10.4.3 2048 Byte 1-bit ECC Parity Code Assignment Table](#) and [10.4.4 32 Byte 1-bit ECC Parity Code Assignment Table](#) describes 1-bit ECC parity code assignment.

10.4.3 2048 Byte 1-bit ECC Parity Code Assignment Table

| | DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| MECCn_0 | – P64 | – P64' | – P32 | – P32' | – P16 | – P16' | – P8 | – P8' |
| MECCn_1 | – P1024 | – P1024' | – P512 | – P512' | – P256 | – P256' | – P128 | – P128' |
| MECCn_2 | – P4 | – P4' | – P2 | – P2' | – P1 | – P1' | – P2048 | – P2048' |
| MECCn_3 | 1 | 1 | 1 | 1 | – P8192 | – P8192' | – P4096 | – P4096' |

10.4.4 32 Byte 1-bit ECC Parity Code Assignment Table

| | DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| SECCn_0 | – P2 | – P2' | – P1 | – P1' | – P16 | – P16' | – P8 | – P8' |
| SECCn_1 | – P128 | – P128' | – P64 | – P64' | – P32 | – P32' | – P4 | – P4' |

10.4.5 1-bit ECC Module Features

The ECC Lock (MainECClock and SpareECClock) bit of the control register generates the 1-bit ECC. If ECClock is low, the hardware ECC modules generate the ECC codes.

1-bit ECC Register Configuration

The NAND Flash Memory interface table describes the configuration of 1-bit ECC value read from spare area of external NAND flash memory. The format of ECC read from memory is important to compare the ECC parity code that the hardware modules generate.

NOTE: 4-bit/8-bit/12-bit/16-bit ECC decoding scheme is different compared to 1-bit ECC.

NAND Flash Memory Interface

| Register | Bit[31:24] | Bit[23:16] | Bit[15:8] | Bit[7:0] |
|-----------------|-------------------|-------------------|------------------|-----------------|
| NFMECCD0 | Not used | 2nd ECC | Not used | 1st ECC |
| NFMECCD1 | Not used | 4th ECC | Not used | 3rd ECC |
| NFSECCD | Not used | 2nd ECC | Not used | 1st ECC |

10.4.6 1-bit ECC Programming Guide

1. To use SLC ECC in software mode, reset the ECCType to "0" (enable SLC ECC). ECC module generates ECC parity code for all Read/Write data when MainECClock (NFCON[7]) and SpareECClock (NFCON[6]) are unlocked ("0"). You should reset ECC value. To reset ECC value, write the InitMECC (NFCONT[5]) and InitSECC (NFCON[4]) bit as "1" and clear the MainECClock (NFCONT[7]) bit to "0" (Unlock) before Reading or Writing data. MainECClock (NFCONT[7]) and SpareECClock (NFCONT[6]) bits control whether it generates ECC parity code or not.
2. The ECC module generates ECC parity code on register NFMECC0/1 whenever it Reads or Writes data.
3. After you complete Reading or Writing one page (excluding spare area data), set the MainECClock bit to "1" (Lock). It locks ECC parity code and the value of the ECC status register does not change.
4. To generate spare area ECC parity code, clear SpareECClock (NFCONT[6]) bit as "0" (Unlock).
5. The spare area ECC module generates ECC parity code on register NFSECC whenever it Reads or Writes data.
6. After you complete Reading or Writing spare area, set the SpareECClock bit to "1" (Lock). It locks ECC parity code and it does not change the value of the ECC status register.
7. From now on, you can use these values to record to the spare area or verify the bit error.
8. For example, to verify the bit error of main data area on page Read operation, you should move the ECC parity codes (stored in spare area) to NFMECCD0 and NFMECCD1 after it generates ECC codes for main data area. From this point, the NFECCERR0 and NFECCERR1 have the valid error status values.

NOTE: NFSECCD is for ECC in the spare area. The main data area generates the spare area. (Usually, the user writes the ECC value generated from main data area to spare area. The value is similar to NFMECC0/1).

10.4.7 4-bit ECC Programming Guide (ENCODING)

1. To use 4-bit ECC in software mode, set the MsgLength to 0 (512 byte message length) and the ECCType to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 512 byte read data. Therefore, to reset ECC value write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECClock (NFCONT[7]) bit to "0" (Unlock) before reading data.
MainECClock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever it writes data, the 4-bit ECC module generates ECC parity code internally.
3. After you complete writing 512 byte data (excluding spare area data) it updates the parity codes automatically to NFMECC0 and NFMECC1 registers. If you use 512 byte NAND Flash memory, you can program these values to spare area. However, if you use NAND Flash memory more than 512 byte page, you cannot program immediately. In this case, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.
The parity codes have self-correctable information including parity code itself.
4. To generate spare area ECC parity code, set the MsgLength to "1" (24 byte message length) and the ECC Type to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 24 byte write data. To reset ECC value write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECClock (NFCONT[7]) bit to "0" (unlock) before writing data. MainECClock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
5. Whenever it writes data, the 4-bit ECC module generates ECC parity code internally.
6. When you complete writing 24 byte meta or extra data, it automatically updates the parity codes to NFMECC0 and NFMECC1 registers. You can program these parity codes to spare area.
The parity codes have self-correctable information including parity code itself.

10.4.8 4-bit ECC Programming Guide (DECODING)

1. To use 4-bit ECC in software mode, set the MsgLength to "0" (512 byte message length) and the ECCType to "1" (enable 4-bit ECC). ECC module generates ECC parity code for 512 byte read data. Therefore, to reset ECC value, write the InitMECC (NFCONT[5]) bit as "1" and clear the MainECCLock (NFCONT[7]) bit to "0" (Unlock) before reading data.
MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
3. After you complete reading 512 byte (excluding spare area data), you should read parity codes. MLC ECC module needs parity codes to detect whether error bits have occurred or not. Therefore, you should read ECC parity code immediately after reading 512 byte. After reading ECC parity code, 4-bit ECC engine starts searching for error internally. 4-bit ECC error searching engine requires minimum of 155 cycles to find any error. During this time, you can continue reading main data from external NAND flash memory.
Use ECCDecDone (NFSTAT[6]) to verify whether ECC decoding is completed or not.
4. When ECCDecDone (NFSTAT[6]) is set to "1", NFECCERR0 indicates whether error bit exists or not. If any error exists, refer NFECCERO/1 and NFMLCBITPT registers to fix.
5. If you have more main data to Read, repeat step 1.
6. To verify meta data error, set the MsgLength to 1 (24-byte message length) and the ECCType to "1" (Enable 4-bit ECC). ECC module generates ECC parity code for 24-byte read data. Therefore, you must reset ECC value by writing the InitSECC (NFCONT[4]) bit as "1" and clear the SpareECCLock (NFCONT[6]) bit to "0" (Unlock) before reading data.
SpareECCLock (NFCONT[6]) bit controls whether ECC Parity code is generated or not.
7. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
8. After you complete reading 24 byte (excluding spare area data), you should read parity codes. 4-bit ECC module needs parity codes to detect whether error bits have occurred or not. Therefore, ensure to read ECC parity codes immediately after reading 24 byte. After ECC parity code is read, 4-bit ECC engine starts searching for error internally to verify whether ECC decoding is completed or not.
9. When ECCDecDone (NFSTAT[6]) is set ("1"), NFECCERR0 indicates whether error bit exists or not. If any error exists, you can fix it by referring to NFECCERO/1 and NFMLCBITPT registers.

10.4.9 8/12/16-bit ECC Programming Guide (ENCODING)

1. To use 8/12/16-bit ECC in software mode, set the MsgLength (NFECCCONF[25:16]) to 511 (512 byte message length) and the ECCType to "001/100/101" (enable 8/12/16-bit ECC, respectively). ECC module generates ECC parity code for 512 byte write data. Therefore, reset ECC value by writing the InitMECC (NFECCCONT[2]) bit as "1" before writing data and clear the MainECCLock (NFCONT[7]) bit to "0" (unlock) before writing data.
2. Whenever data is written, the corresponding 8/12/16-bit ECC module generates ECC parity code internally.
3. After you complete writing 512 byte data (excluding spare area data), the parity codes are automatically updated to the NFECCPRG0-NFECCPRGECC6 registers. If you use a NAND flash memory that contains 512 byte page, you can program these values to spare area. However, if you use a NAND flash memory more than 512 byte page, you cannot program immediately. In this case, you should copy these ECC parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.

The parity codes have self-correctable information including parity code itself.

The following table describes the ECC parity size.

| ECC Type | Size of ECC Parity Codes |
|------------|--------------------------|
| 8-bit ECC | 13 byte |
| 12-bit ECC | 20 byte |
| 16-bit ECC | 26 byte |

4. To generate spare area ECC parity code for meta data, the steps are similar (from 1-3), except setting the MsgLength (NFECCCONF[25:16]) to the size that you prefer. When you set InitMECC (NFECCCONT[2]), all ECC parity codes generated for main data are cleared. Therefore, you should copy the ECC parity codes for main data.

NOTE: You should set the ECC parity conversion codes to verify free page error.

Refer to [10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC](#) for more information.

10.4.10 8/12/16-bit ECC Programming Guide (DECODING)

1. To use 8/12/16-bit ECC in software mode, set the MsgLength (NFECCCONF[25:16] to 511 (512 byte message length) and the ECCType to "001/100/101" (enable 8/12/16-bit ECC, respectively). ECC module generates ECC parity code for 512 byte read data. Therefore, you should reset ECC value by writing the InitMECC (NFECCCONT[2]) bit as "1" and clear the MainECClock (NFCONT[7]) bit to "0" (unlock) before read data.
2. Whenever data is read, the 8/12/16-bit ECC module generates ECC parity code internally.
3. After you complete reading 512 byte (excluding spare area data), ensure to read the corresponding parity codes. ECC module requires parity codes to detect whether error bits have occurred or not. Therefore, you should read ECC parity code immediately after reading 512 byte. After reading the ECC parity code, the 8/12/16-bit ECC engine searches for error internally. 8/12/16-bit ECC search engine requires minimum of 155 cycles to find any errors. DecodeDone (NFECCSTAT[24]) can be used to check whether ECC decoding is completed or not.
4. When DecodeDone (NFECCSTAT[24]) is set ("1"), ECCError (NFECCSECSTAT[4:0]) indicates whether error bit exists or not. If any error exists, you can fix it by referencing NFECCERL0 to NFECCERL7 and NFECCERP0 to NFECCERP3 registers.
5. If you have additional main data to Read, repeat the steps 1-4.
6. To verify spare area data (meta data) error, the sequences are similar (steps 1-4), except setting the MsgLength (NFECCCONF[25:16]) to the size that you want.

NOTE: You should set the ECC parity conversion codes to check free page error.

Refer to [10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC](#), for more information.

10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC

The ECC parity conversion codes are there to fix errors, which occur when reading a free page. Free page means the erased page. The 8/12/16-bit ECC modules support variable message size for meta data stored in spare area. Generally, the size of main data (sector) is 512 byte and user should set the corresponding ECC parity conversion codes as the Table describes:

| ECC Type | ECC Parity Conversion Codes |
|------------|---|
| 8-bit ECC | Here, 13 byte ECC parity conversion codes |
| 12-bit ECC | Here, 20 byte ECC parity conversion codes |
| 16-bit ECC | Here, 26 byte ECC parity conversion codes |

Depending on the requirements of users, the message size for meta data stored spare area might differ. Therefore, you can change the size of meta data by changing MsgLength (NFECCCONF[25:16]) and change ECC parity conversion codes.

Steps to determine ECC parity conversion codes according to the size of message length are:

1. Clear all ECC parity conversion registers (NFECCONECC0 to NFECCONECC6) as all zero.
2. Set all registers for page program.
3. Reset InitMECC (NFECCCONT[2] bit as "1".
4. Write "0xff" data as much as the size of meta data.
5. After you write data as MsgLength (NFECCCONF[25:16]), the EncodeDone (NFECCSTAT[25]) is set to "1". It generates the corresponding ECC parity codes.
6. Set ECC parity conversion registers as inverted values of ECC parity codes generated. To ensure ECC parity conversion codes work properly, repeat step 3-5. After you set ECC parity conversion codes, if the generated ECC parity codes are all "0xff", then it is working correctly.

Constraints to support free page function are:

1. Free page check is for only data area (512 byte)
2. If there is an error during reading a page erased (free page), then free page engine indicates that the page is not free page.
3. To detect errors on free page, you should set corresponding conversion codes.

10.4.12 Lock Scheme for Data Protection

NFCON provides a lock scheme to protect data stored in external NAND flash memories from malicious program.

For this scheme, the NFSBLK and NFEBLK registers are used to provide access control methods. Only the memory area between NFSBLK and NFEBLK is erasable and programmable. However, the read access is available to entire memory area.

This lock scheme is only available when you enable LockTight (NFCONT[17]) and LOCK(NFCONT[16]).

1. Unlock Mode

In unlock mode, user can access entire NAND flash memory; there are no constraints to access memory.

2. Soft Lock Mode

In soft lock mode, you can access NAND block area between NFSBLK and NFEBLK.

When you try to program or erase the locked area, an illegal access error occurs
(NFSTAT [5] bit will be set).

3. Lock-Tight Mode

In lock-tight mode, you can access NAND block area between NFSBLK and NFEBLK as soft lock mode. The difference is that you cannot change NFSBLK and NFEBLK registers. You cannot change LockTight (NFCONT[17]) bits also.

When you try to program or erase the locked area, an illegal access error occurs (it sets NFSTAT[5] bit).

The LockTight (NFCONT[17]) bit is only cleared when reset or wake up from sleep mode (It is impossible to clear it by software).

[Figure 10-4](#) illustrates the accessibility of NAND area.

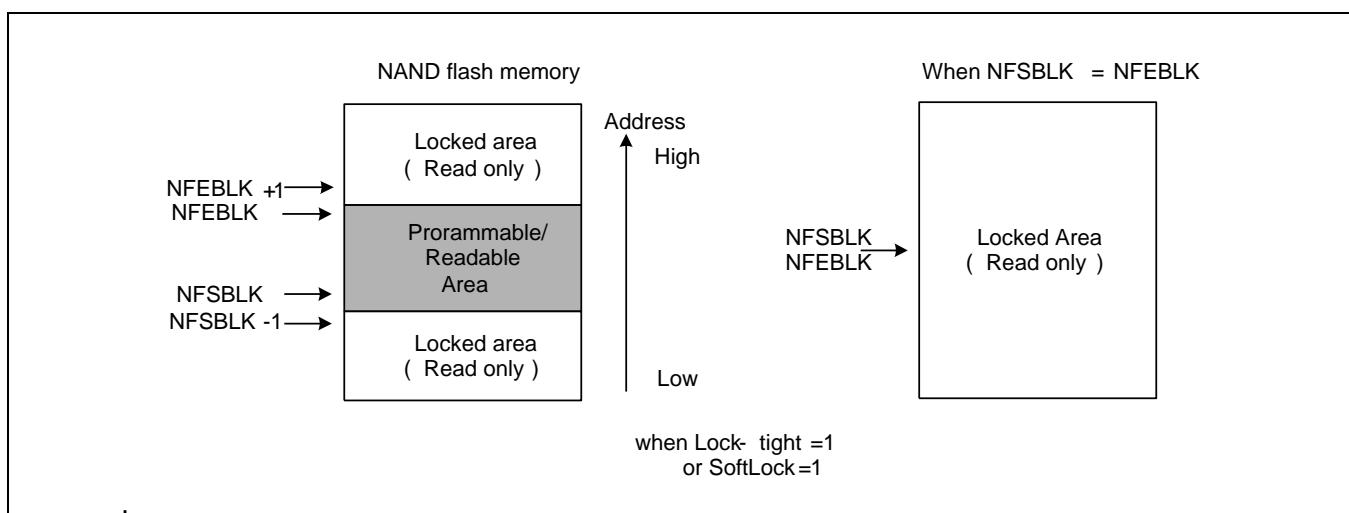


Figure 10-4 Accessibility of NAND Area

NOTE: If the address of NFSBLK and NFEBLK are similar, then it does not allow the erase and program to all NAND memory.

10.5 Programming Constraints

NFCON has a constraint to access an external NAND flash memory. NFCON accesses NAND flash memory through External Bus Interface (EBI) which uses two different clocks source. The constraint occurs because EBI operates using OneNAND external interface clock and EBI interface between NFCON and EBI is handled as asynchronous interface so that a few clock latencies consume for bus handshaking. The clock of NFCON should be set lower than EBI internal operation clock. Refer to the EBI and Clock Management Unit (CMU) manual, for more information.

10.6 I/O Description

| Signal | I/O | Description | Pad | Type |
|--------------|--------------|----------------------|--------------|-------|
| Xm0DATA[7:0] | Input/Output | Address/data bus | Xm0DATA[7:0] | muxed |
| Xm0FRnB[3:0] | Input | Ready and busy | Xm0FRnB[3:0] | muxed |
| Xm0FCLE | Output | Command latch enable | Xm0FCLE | muxed |
| Xm0FALE | Output | Address latch enable | Xm0FALE | muxed |
| Xm0CSn[3:0] | Output | Chip enable | Xm0CSn[3:0] | muxed |
| Xm0REn | Output | Read enable | Xm0OEn | muxed |
| Xm0WEn | Output | Write enable | Xm0WEn | muxed |

10.7 Register Description

10.7.1 Register Map Summary

- Base Address: 0x0CE0_0000

| Register | Offset | Description | Reset Value |
|-----------------------------|--------|---|-------------|
| 1/4-bit ECC Register | | | |
| NFCONF | 0x0000 | Configuration register | 0x0000_1000 |
| NFCONT | 0x0004 | Control register | 0x00C1_00C6 |
| NFCMMD | 0x0008 | Command register | 0x0000_0000 |
| NFADDR | 0x000C | Address register | 0x0000_0000 |
| NFDATA | 0x0010 | Data register | 0x0000_0000 |
| NFMECCD0 | 0x0014 | 1st and 2nd main ECC data register | 0x0000_0000 |
| NFMECCD1 | 0x0018 | 3rd and 4th main ECC data register | 0x0000_0000 |
| NFSECCD | 0x001C | Spare ECC read register | 0xFFFF_FFFF |
| NFSBLK | 0x0020 | Programmable start block address register | 0x0000_0000 |
| NFEBLK | 0x0024 | Programmable end block address register | 0x0000_0000 |
| NFSTAT | 0x0028 | NAND status register | 0xF080_OF0D |
| NFECCERR0 | 0x002C | ECC error status0 register | 0x0003_FFF2 |
| NFECCERR1 | 0x0030 | ECC error status1 register | 0x0000_0000 |
| NFMECC0 | 0x0034 | Generated ECC status0 register | 0xFFFF_FFFF |
| NFMECC1 | 0x0038 | Generated ECC status1 register | 0xFFFF_FFFF |
| NFSECC | 0x003C | Generated spare area ECC status register | 0xFFFF_FFFF |
| NFMLCBITPT | 0x0040 | 4-bit ECC error bit pattern register | 0x0000_0000 |

- Base Address: 0x0CE2_0000

| Register | Offset | Description | Reset Value |
|---------------------------------|--------|--|-------------|
| 8/12/16-bit ECC Register | | | |
| NFECCCONF | 0000 | ECC configuration register | 0x0000_0000 |
| NFECCCONT | 0020 | ECC control register | 0x0000_0000 |
| NFECCSTAT | 0030 | ECC status register | 0x0000_0000 |
| NFECCSECSTAT | 0040 | ECC sector status register | 0x0000_0000 |
| NFECCPRGECC0 | 0090 | ECC parity code0 register for page program | 0x0000_0000 |
| NFECCPRGECC1 | 0094 | ECC parity code1 register for page program | 0x0000_0000 |
| NFECCPRGECC2 | 0098 | ECC parity code2 register for page program | 0x0000_0000 |
| NFECCPRGECC3 | 009C | ECC parity code3 register for page program | 0x0000_0000 |
| NFECCPRGECC4 | 00A0 | ECC parity code4 register for page program | 0x0000_0000 |
| NFECCPRGECC5 | 00A4 | ECC parity code5 register for page program | 0x0000_0000 |
| NFECCPRGECC6 | 00A8 | ECC parity code6 register for page program | 0x0000_0000 |
| NFECCERL0 | 00C0 | ECC error byte location0 register | 0x0000_0000 |
| NFECCERL1 | 00C4 | ECC error byte location1 register | 0x0000_0000 |
| NFECCERL2 | 00C8 | ECC error byte location2 register | 0x0000_0000 |
| NFECCERL3 | 00CC | ECC error byte location3 register | 0x0000_0000 |
| NFECCERL4 | 00D0 | ECC error byte location4 register | 0x0000_0000 |
| NFECCERL5 | 00D4 | ECC error byte location5 register | 0x0000_0000 |
| NFECCERL6 | 00D8 | ECC error byte location6 register | 0x0000_0000 |
| NFECCERL7 | 00DC | ECC error byte location7 register | 0x0000_0000 |
| NFECCERP0 | 00F0 | ECC error bit pattern0 register | 0x0000_0000 |
| NFECCERP1 | 00F4 | ECC error bit pattern1 register | 0x0000_0000 |
| NFECCERP2 | 00F8 | ECC error bit pattern2 register | 0x0000_0000 |
| NFECCERP3 | 00FC | ECC error bit pattern3 register | 0x0000_0000 |
| NFECCONECC0 | 0110 | ECC parity conversion code0 register | 0x0000_0000 |
| NFECCONECC1 | 0114 | ECC parity conversion code1 register | 0x0000_0000 |
| NFECCONECC2 | 0118 | ECC parity conversion code2 register | 0x0000_0000 |
| NFECCONECC3 | 011C | ECC parity conversion code3 register | 0x0000_0000 |
| NFECCONECC4 | 0120 | ECC parity conversion code4 register | 0x0000_0000 |
| NFECCONECC5 | 0124 | ECC parity conversion code5 register | 0x0000_0000 |
| NFECCONECC6 | 0128 | ECC parity conversion code6 register | 0x0000_0000 |

10.7.2 NAND Flash Interface and 1/4-bit ECC Registers

10.7.2.1 NFCONF

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_1000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31:26] | – | Reserved | 0 |
| MsgLength | [25] | RW | 0 = 512 byte message length 1 = 24 byte message length | 0 |
| ECCType0 | [24:23] | RW | This bit indicates the type of ECC to use. 00 = 1-bit ECC 10 = 4-bit ECC 01 = 11 = Disables 1-bit and 4-bit ECC | 0 |
| RSVD | [22:16] | – | Reserved | 0000000 |
| TACLS | [15:12] | RW | CLE and ALE duration setting value (0 – 15) • Duration = HCLK × TACLS | 0x1 |
| TWRPH0 | [11:8] | RW | TWRPH0 duration setting value (0 – 15) • Duration = HCLK × (TWRPH0 + 1) NOTE: You should add additional cycles about 10ns for page read because of additional signal delay on PCB pattern. | 0x0 |
| TWRPH1 | [7:4] | RW | TWRPH1 duration setting value (0 – 15) • Duration = HCLK × (TWRPH1 + 1) | 0x0 |
| PageSize | [3:2] | RW | This bit indicates the page size of NAND flash memory 00 = 2048 byte 01 = 512 byte 10 = 4096 byte 11 = 2048 byte NOTE: Using 1-bit ECC it determines the message length. It does not determine the message length using MsgLength (NFCONF[25] field). NFCON does not consider the actual page size of external NAND. Software handles the page size. | 0 |
| AddrCycle | [1] | RW | This bit indicates the number of address cycle of NAND flash memory. When Page Size is 512 Bytes: 0 = 3 Address cycle 1 = 4 Address cycle When page size is 2 K or 4 K: 0 = 4 Address cycle 1 = 5 Address cycle NOTE: It is only used for Lock scheme. Refer to section 10.4.12 Lock Scheme for Data Protection , for more information. | 0 |
| RSVD | [0] | – | Reserved | 0 |

10.7.2.2 NFCONT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0004, Reset Value = 0x00C1_00C6

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0 |
| Reg_nCE3 | [23] | RW | NAND flash memory nRCS[3] signal control 0 = Force nRCS[3] to low (Enables chip select) 1 = Force nRCS[3] to high (Disables chip select) | 1 |
| Reg_nCE2 | [22] | RW | NAND flash memory nRCS[2] signal control 0 = Force nRCS[2] to low (Enables chip select) 1 = Force nRCS[2] to high (Disables chip select) | 1 |
| RSVD | [21:19] | – | Reserved | 0 |
| MLCEccDirection | [18] | RW | 4-bit, ECC encoding/decoding control 0 = Decoding 4-bit ECC. It is used for page read 1 = Encoding 4-bit ECC. It is used for page program | 0 |
| LockTight | [17] | RW | Lock-tight configuration 0 = Disables lock-tight 1 = Enables lock-tight If this bit is set to 1, you cannot clear this bit. Refer to 10.4.12 Lock Scheme for Data Protection , for more information. | 0 |
| LOCK | [16] | RW | Soft Lock configuration 0 = Disables lock 1 = Enables lock Software can modify soft lock area any time. Refer to 10.4.12 Lock Scheme for Data Protection , for more information. | 1 |
| RSVD | [15:14] | – | Reserved | 00 |
| EnbMLCEncInt | [13] | RW | 4-bit ECC encoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt | 0 |
| EnbMLCDecInt | [12] | RW | 4-bit ECC decoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt | 0 |
| RSVD | [11] | – | Reserved | 0 |
| EnbIllegalAccINT | [10] | RW | Illegal access interrupt control 0 = Disables interrupt 1 = Enables interrupt Illegal access interrupt occurs when CPU tries to program or erase locking area (the area setting in NFSBLK (0xB0E0_0020) to NFEBLK (0xB0E0_0024) – 1. | 0 |
| EnbRnBINT | [9] | RW | RnB status input signal transition interrupt control 0 = Disables RnB interrupt | 0 |

| Name | Bit | Type | Description | Reset Value |
|---------------|-----|------|--|-------------|
| | | | 1 = Enables RnB interrupt | |
| RnB_TransMode | [8] | RW | RnB transition detection configuration 0 = Detects rising edge 1 = Detects falling edge | 0 |
| MECCLock | [7] | RW | Lock Main area ECC generation 0 = Unlocks Main area ECC 1 = Locks Main area ECC Main area ECC status register is NFMECC0/NFMECC1 (0xB0E0_0034/0xB0E0_0038), | 1 |
| SECCLock | [6] | RW | Lock Spare area ECC generation 0 = Unlocks Spare ECC 1 = Locks Spare ECC Spare area ECC status register is NFSECC (0xB0E0_003C) | 1 |
| InitMECC | [5] | RW | 1 = Initializes main area ECC decoder/encoder (Write-only) | 0 |
| InitSECC | [4] | RW | 1 = Initializes spare area ECC decoder/encoder (Write-only) | 0 |
| HW_nCE | [3] | RW | Reserved (HW_nCE) | 0 |
| Reg_nCE1 | [2] | RW | NAND flash memory nRCS[1] signal control | 1 |
| Reg_nCE0 | [1] | RW | NAND flash memory nRCS[0] signal control 0 = Force nRCS[0] to low (enables chip select) 1 = Force nRCS[0] to high (disables chip select) NOTE: The setting all nCE[3:0] zero cannot be allowed. Only one nCE can be asserted to enable external NAND flash memory. The lower bit has more priority when user set all nCE[3:0] zeros. | 1 |
| MODE | [0] | RW | NAND flash controller operating mode 0 = Disables NAND flash controller 1 = Enables NAND flash controller | 0 |

10.7.2.3 NFCMMD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|---------------------------------|-------------|
| RSVD | [31:8] | – | Reserved | 0x000000 |
| REG_CMMMD | [7:0] | RW | NAND flash memory command value | 0x00 |

10.7.2.4 NFADDR

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---------------------------------|-------------|
| RSVD | [31:8] | – | Reserved | 0x000000 |
| REG_ADDR | [7:0] | RW | NAND flash memory address value | 0x00 |

10.7.2.5 NFDATA

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------|--------|------|---|-------------|
| NFDATA | [31:0] | RW | NAND flash Read/program data value for I/O NOTE: Refer to 10.4.1 Data Register Configuration , for more information. | 0x00000000 |

10.7.2.6 NFMECCD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| ECCData1 (ECC1) | [23:16] | RW | 2nd ECC NOTE: In software mode, read this register when you need to read 2nd ECC value from NAND flash memory | 0x00 |
| RSVD | [15:8] | – | Reserved | 0x00 |
| ECCData0 (ECC0) | [7:0] | RW | 1st ECC. NOTE: In software mode, read this register when you need to read 1st ECC value from NAND flash memory. This register has the similar Read function as NFDATA. | 0x00 |

NOTE: It allows only word access.

10.7.2.7 NFMECCD1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| ECCData3 (ECC3) | [23:16] | RW | 4th ECC. NOTE: In software mode, read this register when you need to read 4th ECC value from NAND flash memory | 0x00 |
| RSVD | [15:8] | – | Reserved | 0x00 |
| ECCData2 (ECC2) | [7:0] | RW | 3rd ECC. NOTE: In software mode, read this register when you need to read 3rd ECC value from NAND flash memory. This register has the similar Read function as NFDATA. | 0x00 |

10.7.2.8 NFSECCD

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x001C, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| SECCData1 | [23:16] | RW | 2nd ECC. NOTE: In software mode, read this register when you need to read 2nd ECC value from NAND flash memory | 0xFF |
| RSVD | [15:8] | – | Reserved | 0x00 |
| SECCData0 | [7:0] | RW | 1st ECC. NOTE: In software mode, read this register when you need to read 1st ECC value from NAND flash memory. This register has the similar Read function as NFDATA. | 0xFF |

NOTE: It allows only word access.

10.7.2.9 NFSBLK

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| SBLK_ADDR2 | [23:16] | RW | The 3rd block address of the block erase operation | 0x00 |
| SBLK_ADDR1 | [15:8] | RW | The 2nd block address of the block erase operation | 0x00 |
| SBLK_ADDR0 | [7:0] | RW | The 1st block address of the block erase operation (Only bit[7:5] are valid). | 0x00 |

NOTE: Address of Advance Flash block starts from 3-address cycle. So block address register only requires 3-bytes.

Refer to [10.4.12 Lock Scheme for Data Protection](#), for more information on lock scheme.

10.7.2.10 NFEBLK

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| Eblk_ADDR2 | [23:16] | RW | The 3rd block address of the block erase operation | 0x00 |
| Eblk_ADDR1 | [15:8] | RW | The 2nd block address of the block erase operation | 0x00 |
| Eblk_ADDR0 | [7:0] | RW | The 1st block address of the block erase operation (Only bit[7:5] are valid) | 0x00 |

NOTE: Address of Advance Flash block starts from 3-address cycle. So block address register only requires 3-bytes.

Refer to [10.4.12 Lock Scheme for Data Protection](#) for more information on lock scheme.

10.7.2.11 NFSTAT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0028, Reset Value = 0xF080_0F0D

| Name | Bit | Type | Description | Reset Value |
|-------------------------------|---------|------|--|-------------|
| Flash_RnB_GRP | [31:28] | RW | The status of RnB[3:0] input pin 0 = NAND flash memory busy 1 = NAND flash memory ready to operate | 0xF |
| RnB_TransDetect_GRP | [27:24] | RW | When RnB[3:0] low to high transition occurs, this bit is set and an interrupt is issued if RnB_TransDetect_GRP is enabled. To clear this, write "1". 0 = RnB transition is not detected 1 = RnB transition is detected Transition configuration is set in RnB_TransMode(NFCONT[8]). | - |
| RSVD | [23:12] | - | Reserved | 0x800 |
| Flash_nCE[3:0] (Read-only) | [11:8] | RW | The status of nCE[3:0] output pin | 0xF |
| MLCEncodeDone | [7] | RW | When it completes 4-bit ECC encoding, this bit is set and it issues an interrupt if it enables MLCEncodeDone. The NFMILCECC0 and NFMILCECC1 have valid values. To clear this, write "1". 1 = It completes 4-bit ECC encoding | 0 |
| MLCDecodeDone | [6] | RW | When it completes 4-bit ECC decoding, this bit is set and it issues an interrupt if it enables MLCDecodeDone. The NFMILCBITPT, NFMCLC0, and NFMCLC1 have valid values. To clear this, write "1". 1 = Completes 4-bit ECC decoding | 0 |
| IllegalAccess | [5] | RW | Once Soft Lock or Lock-tight is enabled and any illegal access (program, erase) to the memory takes place, then this bit is set. 0 = It does not detect illegal access 1 = It detects illegal access To clear this value, write 1 to this bit. | 0 |
| RnB_TransDetect | [4] | RW | When RnB[0] low to high transition occurs, this bit is set and an interrupt is issued if RnB_TransDetect is enabled. To clear this, write "1". 0 = It does not detect RnB transition 1 = It detects RnB transition Transition configuration is set in RnB_TransMode(NFCONT[8]). | 0 |
| Flash_nCE[1] (Read-only) | [3] | RW | The status of nCE[1] output pin | 1 |
| Flash_nCE[0] (Read-only) | [2] | RW | The status of nCE[0] output pin | 1 |
| RSVD | [1] | - | Reserved | 0 |

| Name | Bit | Type | Description | Reset Value |
|--------------------------|-----|------|--|-------------|
| Flash_RnB (Read-only) | [0] | RW | The status of RnB[0] input pin 0 = NAND flash memory busy 1 = NAND flash memory ready to operate | 1 |

10.7.2.12 NFECCERR0

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x002C, Reset Value = 0x0003_FFF2

When ECC Type is 1-bit ECC

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:25] | — | Reserved | 0x00 |
| ECCSDataAddr | [24:21] | R | In spare area, indicates which number data is error | 0x0 |
| ECCSBitAddr | [20:18] | R | In spare area, indicates which bit is error | 000 |
| ECCDataAddr | [17:7] | R | In main data area, indicates which number data is error | 0x7FF |
| ECCBitAddr | [6:4] | R | In main data area, indicates which bit is error | 111 |
| ECCSprErrNo | [3:2] | R | Indicates whether spare area bit fail error occurred 00 = No Error 01 = 1-bit error (correctable) 10 = Multiple error 11 = ECC area error | 00 |
| ECCMainErrNo | [1:0] | R | Indicates whether main data area bit fail error occurred 00 = No Error 01 = 1-bit error (Correctable) 10 = Multiple error 11 = ECC area error | 10 |

NOTE: The above values are valid only when both ECC register and ECC status register have valid value.

When ECC Type is 4-bit ECC

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--|-------------|
| MLCECCBusy | [31] | R | Indicates the 4-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy | 0 |
| MLCECCReady | [30] | R | ECC Ready bit | 1 |
| MLCFreePage | [29] | R | Indicates the page data read from NAND flash has all "FF" value. | 0 |
| MLCECCError | [28:26] | R | 4-bit ECC decoding result 000 = No error 001 = 1-bit error 010 = 2-bit error 011 = 3-bit error 100 = 4-bit error 101 = Uncorrectable 11x = Reserved | 000 |
| MLCerrLocation2 | [25:16] | R | Error byte location of 2nd bit error | 0x000 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--------------------------------------|-------------|
| RSVD | [15:10] | – | Reserved | 0x00 |
| MLCErrLocation1 | [9:0] | R | Error byte location of 1st bit error | 0x000 |

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ("1").

10.7.2.13 NFECCERR1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

When ECC Type is 4-bit ECC

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|--------------------------------------|-------------|
| RSVD | [31:26] | — | Reserved | 0x00 |
| MLCErrLocation4 | [25:16] | R | Error byte location of 4th bit error | 0x00 |
| RSVD | [15:10] | — | Reserved | 0x00 |
| MLCErrLocation3 | [9:0] | R | Error byte location of 3rd bit error | 0x000 |

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ("1").

10.7.2.14 NFMECC0

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0034, Reset Value = 0xFFFF_FFFF

When ECC Type is 1-bit ECC

| Name | Bit | Type | Description | Reset Value |
|-------|---------|------|---------------|-------------|
| MECC3 | [31:24] | R | ECC3 for data | 0xFF |
| MECC2 | [23:16] | R | ECC2 for data | 0xFF |
| MECC1 | [15:8] | R | ECC1 for data | 0xFF |
| MECC0 | [7:0] | R | ECC0 for data | 0xFF |

NOTE: The NAND flash controller generate NFMECC0/1 when read or write main area data while the MainECClock (NFCONT[7]) bit is "0" (Unlock).

When ECC Type is 4-bit ECC

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| 4th Parity | [31:24] | R | 4th Check parity generated from main area (512 byte) | 0x00 |
| 3rd Parity | [23:16] | R | 3rd Check parity generated from main area (512 byte) | 0x00 |
| 2nd Parity | [15:8] | R | 2nd Check parity generated from main area (512 byte) | 0x00 |
| 1st Parity | [7:0] | R | 1st Check parity generated from main area (512 byte) | 0x00 |

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECClock (NFCON[7]) bit is "0" (unlock).

10.7.2.15 NFMECC1

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0038, Reset Value = 0xFFFF_FFFF

When ECC Type is 4-bit ECC

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [31:24] | – | Reserved | 0x00 |
| 7th Parity | [23:16] | R | 7th Check parity generated from main area (512 byte) | 0x00 |
| 6th Parity | [15:8] | R | 6th Check parity generated from main area (512 byte) | 0x00 |
| 5th Parity | [7:0] | R | 5th Check parity generated from main area (512 byte) | 0x00 |

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECClock (NFCON[7]) bit is "0" (unlock).

10.7.2.16 NFSECC

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x003C, Reset Value = 0xFFFF_FFFF

| Name | Bit | Type | Description | Reset Value |
|-------|---------|------|------------------------|-------------|
| RSVD | [31:16] | – | Reserved | 0xFFFF |
| SECC1 | [15:8] | R | Spare area ECC1 Status | 0xFF |
| SECC0 | [7:0] | R | Spare area ECC0 Status | 0xFF |

NOTE: The NAND flash controller generates NFSECC when Read or Write spare area data while the SpareECClock (NFCONT[6]) bit is "0" (unlock).

10.7.2.17 NFMLCBITPT

- Base Address: 0x0CE0_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------------|---------|------|-----------------------|-------------|
| 4th Error bit pattern | [31:24] | R | 4th Error bit pattern | 0x00 |
| 3rd Error bit pattern | [23:16] | R | 3rd Error bit pattern | 0x00 |
| 2nd Error bit pattern | [15:8] | R | 2nd Error bit pattern | 0x00 |
| 1st Error bit pattern | [7:0] | R | 1st Error bit pattern | 0x00 |

10.7.3 ECC Registers for 8, 12 and 16-bit ECC

10.7.3.1 NFECCCONF

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31] | — | Reserved | 0 |
| RSVD | [28] | — | Reserved | 0 |
| MsgLength | [25:16] | RW | The ECC message size For 512 byte message, you should set to 511. | — |
| RSVD | [15:4] | — | Reserved | 0 |
| ECCType | [3:0] | RW | These bits indicate what type of ECC is used. 000 = Disables 8/12/16-bit ECC 001 = Reserved 010 = Reserved 011 = 8-bit ECC/512B 100 = 12-bit ECC 101 = 16-bit ECC/512B 110 = Reserved 111 = Reserved | 0x0 |

10.7.3.2 NFECCCONT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:26] | — | Reserved | 0x00 |
| EnbMLCEncInt | [25] | RW | MLC ECC encoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt | 0 |
| EnbMLCDecInt | [24] | RW | MLC ECC decoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt | 0 |
| EccDirection | [16] | RW | MLC ECC encoding/decoding control 0 = Decoding, used for page read 1 = Encoding, used for page program | 0 |
| RSVD | [15:3] | — | Reserved | 0x0 |
| InitMECC | [2] | RW | 1 = Initialize main area ECC decoder/encoder (Write-only) | 0 |
| RSVD | [1] | — | Reserved | 0 |
| ResetECC | [0] | RW | 1 = Reset ECC logic (Write-only) | 0 |

10.7.3.3 NFECCSTAT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| ECCBusy | [31] | R | Indicates the 8-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy | 0 |
| RSVD | [30] | - | Reserved | 1 |
| EncodeDone | [25] | RWX | When MLC ECC encoding is finished, this value set and issue interrupt if EncodeDone is enabled. The NFMLCECC0 and NFMLCECC1 have valid values. To clear this, write "1". 1 = It completes MLC ECC encoding | 0 |
| DecodeDone | [24] | RWX | When MLC ECC decoding is finished, this value set and issue interrupt if DecodeDone is enabled. The NFMLCBITPT, NFMLCL0, and NFMLCEL1 have valid values. To clear this, write "1". 1 = It completes MLC ECC decoding | 0 |
| RSVD | [23:9] | - | Reserved | 0x0000 |
| FreePageStat | [8] | R | It indicates whether the sector is free page or not. | 0 |
| RSVD | [7:0] | - | Reserved | 0x00 |

10.7.3.4 NFECCSECSTAT

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|--------|------|--|-------------|
| ValidErrorStat | [31:8] | R | Each bit indicates which ERL and ERP are valid. | 0x0000_00 |
| ECCErrorNo | [4:0] | R | ECC decoding result when page read 00000 = No error 00001 = 1-bit error 00010 = 2-bit error 00011 = 3-bit error 01110 = 14-bit error 01111 = 15-bit error 10000 = 16-bit error NOTE: If it uses 8-bit ECC, the valid number of error is until 8. If the number exceeds the supported error number, it means that uncorrectable error occurs. | 0x00 |

10.7.3.5 NFECCPRGECCn (n = 0 to 6)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0090, Reset Value = 0x0000_0000 (NFECCPRGECC0)
- Address = Base Address + 0x0094, Reset Value = 0x0000_0000 (NFECCPRGECC1)
- Address = Base Address + 0x0098, Reset Value = 0x0000_0000 (NFECCPRGECC2)
- Address = Base Address + 0x009C, Reset Value = 0x0000_0000 (NFECCPRGECC3)
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000 (NFECCPRGECC4)
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000 (NFECCPRGECC5)
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000 (NFECCPRGECC6)

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| 4th Parity | [31:24] | R | 4th check parity for page program from main area | 0x00 |
| 3rd Parity | [23:16] | R | 3rd check parity for page program from main area | 0x00 |
| 2nd Parity | [15:8] | R | 2nd check parity for page program from main area | 0x00 |
| 1st Parity | [7:0] | R | 1st check parity for page program from main area | 0x00 |
| 8th Parity | [31:24] | R | 8th check parity generated from main area | 0x00 |
| 7th Parity | [23:16] | R | 7th check parity generated from main area | 0x00 |
| 6th Parity | [15:8] | R | 6th check parity generated from main area | 0x00 |
| 5th Parity | [7:0] | R | 5th check parity generated from main area | 0x00 |
| 12th Parity | [31:24] | R | 12th check parity generated from main area | 0x00 |
| 11th Parity | [23:16] | R | 11th check parity generated from main area | 0x00 |
| 10th Parity | [15:8] | R | 10th check parity generated from main area | 0x00 |
| 9th Parity | [7:0] | R | 9th check parity generated from main area | 0x00 |
| 16th Parity | [31:24] | R | 16th check parity generated from main area | 0x00 |
| 15th Parity | [23:16] | R | 15th check parity generated from main area | 0x00 |
| 14th Parity | [15:8] | R | 14th check parity generated from main area | 0x00 |
| 13th Parity | [7:0] | R | 13th check parity generated from main area | 0x00 |
| 20th Parity | [31:24] | R | 20th check parity generated from main area | 0x00 |
| 19th Parity | [23:16] | R | 19th check parity generated from main area | 0x00 |
| 18th Parity | [15:8] | R | 18th check parity generated from main area | 0x00 |
| 17th Parity | [7:0] | R | 17th check parity generated from main area | 0x00 |
| 24th Parity | [31:24] | R | 24th check parity generated from main area | 0x00 |
| 23rd Parity | [23:16] | R | 23rd check parity generated from main area | 0x00 |
| 22th Parity | [15:8] | R | 22th check parity generated from main area | 0x00 |
| 21th Parity | [7:0] | R | 21th check parity generated from main area | 0x00 |
| RSVD | [31:16] | - | Reserved | - |
| 26th Parity | [15:8] | R | 26th check parity generated from main area | 0x00 |
| 25th Parity | [7:0] | R | 25th check parity generated from main area | 0x00 |

NOTE: The NAND flash controller generates these ECC parity codes when write main area data while the MainECCLock (NFCON[7]) bit is "0" (unlock).

10.7.3.6 NFECCERLn (n = 0 to 7)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000 (NFECCERL0)
- Address = Base Address + 0x00C4, Reset Value = 0x0000_0000 (NFECCERL1)
- Address = Base Address + 0x00C8, Reset Value = 0x0000_0000 (NFECCERL2)
- Address = Base Address + 0x00CC, Reset Value = 0x0000_0000 (NFECCERL3)
- Address = Base Address + 0x00D0, Reset Value = 0x0000_0000 (NFECCERL4)
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000 (NFECCERL5)
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000 (NFECCERL6)
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000 (NFECCERL7)

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---------------------------------------|-------------|
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc2 | [25:16] | R | Error byte location of 2nd bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc1 | [9:0] | R | Error byte location of 1st bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc4 | [25:16] | R | Error byte location of 4th bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc3 | [9:0] | R | Error byte location of 3rd bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc6 | [25:16] | R | Error byte location of 6th bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc5 | [9:0] | R | Error byte location of 5th bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc8 | [25:16] | R | Error byte location of 8th bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc7 | [9:0] | R | Error byte location of 7th bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc10 | [25:16] | R | Error byte location of 10th bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc9 | [9:0] | R | Error byte location of 9th bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc12 | [25:16] | R | Error byte location of 12th bit error | 0x000 |
| RSVD | [15:10] | — | Reserved | 0x0 |
| ErrByteLoc11 | [9:0] | R | Error byte location of 11th bit error | 0x000 |
| RSVD | [31:26] | — | Reserved | 0x0 |
| ErrByteLoc14 | [25:16] | R | Error byte location of 14th bit error | 0x000 |

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---------------------------------------|-------------|
| RSVD | [15:10] | - | Reserved | 0x0 |
| ErrByteLoc13 | [9:0] | R | Error byte location of 13th bit error | 0x000 |
| RSVD | [31:26] | - | Reserved | 0x0 |
| ErrByteLoc16 | [25:16] | R | Error byte location of 16th bit error | 0x000 |
| RSVD | [15:10] | - | Reserved | 0x0 |
| ErrByteLoc15 | [9:0] | R | Error byte location of 15th bit error | 0x000 |

NOTE: It updates these values when DecodeDone (NFECCSTAT[24]) is set ("1").

10.7.3.7 NFECCERP_n (n = 0 to 3)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000 (NFECCERP0)
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000 (NFECCERP1)
- Address = Base Address + 0x00F8, Reset Value = 0x0000_0000 (NFECCERP2)
- Address = Base Address + 0x00FC, Reset Value = 0x0000_0000 (NFECCERP3)

| Name | Bit | Type | Description | Reset Value |
|------------------------|---------|------|------------------------|-------------|
| 4th ErrBitPattern | [31:24] | R | 4th Error Bit Pattern | 0x00 |
| 3rd ErrBitPattern | [23:16] | R | 3rd Error bit pattern | 0x00 |
| 2nd ErrBitPattern | [15:8] | R | 2nd Error bit pattern | 0x00 |
| 1st ErrBitPattern | [7:0] | R | 1st Error bit pattern | 0x00 |
| 8th ErrBitPattern | [31:24] | R | 8th Error bit pattern | 0x00 |
| 7th ErrBitPattern | [23:16] | R | 7th Error bit pattern | 0x00 |
| 6th ErrBitPattern | [15:8] | R | 6th Error bit pattern | 0x00 |
| 5th ErrBitPattern | [7:0] | R | 5th Error bit pattern | 0x00 |
| 12th ErrBitPattern | [31:24] | R | 12th Error bit pattern | 0x00 |
| 11th ErrBitPattern | [23:16] | R | 11th Error bit pattern | 0x00 |
| 10th ErrBitPattern | [15:8] | R | 10th Error bit pattern | 0x00 |
| 9th ErrBitPattern | [7:0] | R | 9th Error bit pattern | 0x00 |
| 16th ErrBitPattern | [31:24] | R | 16th Error bit pattern | 0x00 |
| 15th Error bit pattern | [23:16] | R | 15th Error bit pattern | 0x00 |
| 14th ErrBitPattern | [15:8] | R | 14th Error bit pattern | 0x00 |
| 13th ErrBitPattern | [7:0] | R | 13th Error bit pattern | 0x00 |

NOTE: It updates these values when DecodeDone (NFECCSTAT[25]) is set ("1").

10.7.3.8 NFECCCONNECCn (n = 0 to 6)

- Base Address: 0x0CE2_0000
- Address = Base Address + 0x0110, Reset Value = 0x0000_0000 (NFECCCONNECC0)
- Address = Base Address + 0x0114, Reset Value = 0x0000_0000 (NFECCCONNECC1)
- Address = Base Address + 0x0118, Reset Value = 0x0000_0000 (NFECCCONNECC2)
- Address = Base Address + 0x011C, Reset Value = 0x0000_0000 (NFECCCONNECC3)
- Address = Base Address + 0x0120, Reset Value = 0x0000_0000 (NFECCCONNECC4)
- Address = Base Address + 0x0124, Reset Value = 0x0000_0000 (NFECCCONNECC5)
- Address = Base Address + 0x0128, Reset Value = 0x0000_0000 (NFECCCONNECC6)

| Name | Bit | Type | Description | Reset Value |
|----------------------|---------|------|---------------------------------|-------------|
| 4th Conversion Code | [31:24] | RW | 4th ECC Parity Conversion Code | 0x00 |
| 3rd Conversion Code | [23:16] | RW | 3rd ECC Parity conversion code | 0x00 |
| 2nd Conversion Code | [15:8] | RW | 2nd ECC Parity conversion code | 0x00 |
| 1st Conversion Code | [7:0] | RW | 1st ECC Parity conversion code | 0x00 |
| 8th Conversion Code | [31:24] | RW | 8th ECC Parity conversion code | 0x00 |
| 7th Conversion Code | [23:16] | RW | 7th ECC Parity conversion code | 0x00 |
| 6th Conversion Code | [15:8] | RW | 6th ECC Parity conversion code | 0x00 |
| 5th Conversion Code | [7:0] | RW | 5th ECC Parity conversion code | 0x00 |
| 12th Conversion Code | [31:24] | RW | 12th ECC Parity conversion code | 0x00 |
| 11th Conversion Code | [23:16] | RW | 11th ECC Parity conversion code | 0x00 |
| 10th Conversion Code | [15:8] | RW | 10th ECC Parity conversion code | 0x00 |
| 9th Conversion Code | [7:0] | RW | 9th ECC Parity conversion code | 0x00 |
| 16th Conversion Code | [31:24] | RW | 16th ECC Parity conversion code | 0x00 |
| 15th Conversion Code | [23:16] | RW | 15th ECC Parity conversion code | 0x00 |
| 14th Conversion Code | [15:8] | RW | 14th ECC Parity conversion code | 0x00 |
| 13th Conversion Code | [7:0] | RW | 13th ECC Parity conversion code | 0x00 |
| 20th Conversion Code | [31:24] | RW | 20th ECC Parity conversion code | 0x00 |
| 19th Conversion Code | [23:16] | RW | 19th ECC Parity conversion code | 0x00 |
| 18th Conversion Code | [15:8] | RW | 18th ECC Parity conversion code | 0x00 |
| 17th Conversion Code | [7:0] | RW | 17th ECC Parity conversion code | 0x00 |
| 24th Conversion Code | [31:24] | RW | 24th ECC Parity conversion code | 0x00 |
| 23th Conversion Code | [23:16] | RW | 23th ECC Parity conversion code | 0x00 |
| 22th Conversion Code | [15:8] | RW | 22th ECC Parity conversion code | 0x00 |
| 21th Conversion Code | [7:0] | RW | 21th ECC Parity conversion code | 0x00 |
| RSVD | [31:16] | – | Reserved | 0x0000 |
| 26th Conversion Code | [15:8] | RW | 26th ECC Parity conversion code | 0x00 |
| 25th Conversion Code | [7:0] | RW | 25th ECC Parity conversion code | 0x00 |

NOTE: For more information about ECC parity conversion codes, refer to [10.4.11 ECC Parity Conversion Code Guide for 8/12/16-bit ECC.](#)

11 Pulse Width Modulation Timer

11.1 Overview

Exynos 4412 has five 32-bit Pulse Width Modulation (PWM) timers. These timers generate internal interrupts for the ARM subsystem. Additionally, timers 0, 1, 2, and 3 include a PWM function that drives an external I/O signal. The PWM in timer 0 has an optional dead-zone generator capability to support a large current device. Timer 4 is an internal timer without output pins.

The Timers use the APB-PCLK as source clock. Timers 0 and 1 share a programmable 8-bit prescaler that provides the first level of division for the PCLK. Timers 2, 3, and 4 share a different 8-bit prescaler. Each timer has its own private clock-divider that provides a second level of clock division (prescaler divided by 2, 4, 8, or 16).

Each timer has its 32-bit down-counter; the timer clock drives this counter. The Timer Count Buffer registers (TCNTBn) loads initial value of the down-counter. If the down-counter reaches zero, it generates the timer interrupt request to inform the CPU that the timer operation is complete. If the timer down-counter reaches zero, the value of corresponding TCNTBn automatically reloads into the down-counter to start a next cycle. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn does not reload into the counter.

The PWM function uses the value of the TCMPBn register. The timer control logic changes the output level if down-counter value matches the value of the compare register in timer control logic. Therefore, the compare register determines the turn-on time or turn-off time of a PWM output.

Each timer is double-buffer structure with the TCNTBn and TCMPBn registers to allow the timer parameters to update in the middle of a cycle. The new values do not take effect until the current timer cycle completes.

[Figure 11-1](#) illustrates the simple example of a PWM cycle.

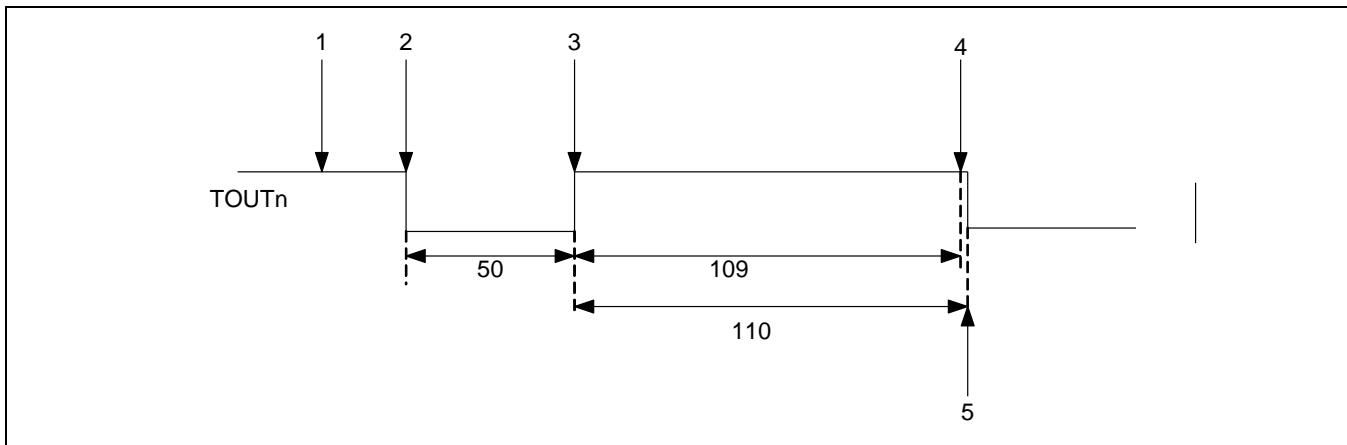


Figure 11-1 Simple Example of a PWM Cycle

Steps to use PWM as a pulse generator are:

5. Initialize the TCNTBn register with 159 ($50 + 109$) and TCMPBn with 109.
6. Start Timer: Set the start bit and manually update this bit to off.
7. The TCNTBn value of 159 is loaded into the down-counter, and then the output $TOUTn$ is set to low.
8. If down-counter counts down the value from TCNTBn to value in the TCMPBn register 109, the output changes from low to high.
9. If the down-counter reaches 0, then it generates an interrupt request.
10. The down-counter automatically reloads TCNTBn. This restarts the cycle.

[Figure 11-2](#) illustrates the clock generation scheme for individual PWM channels.

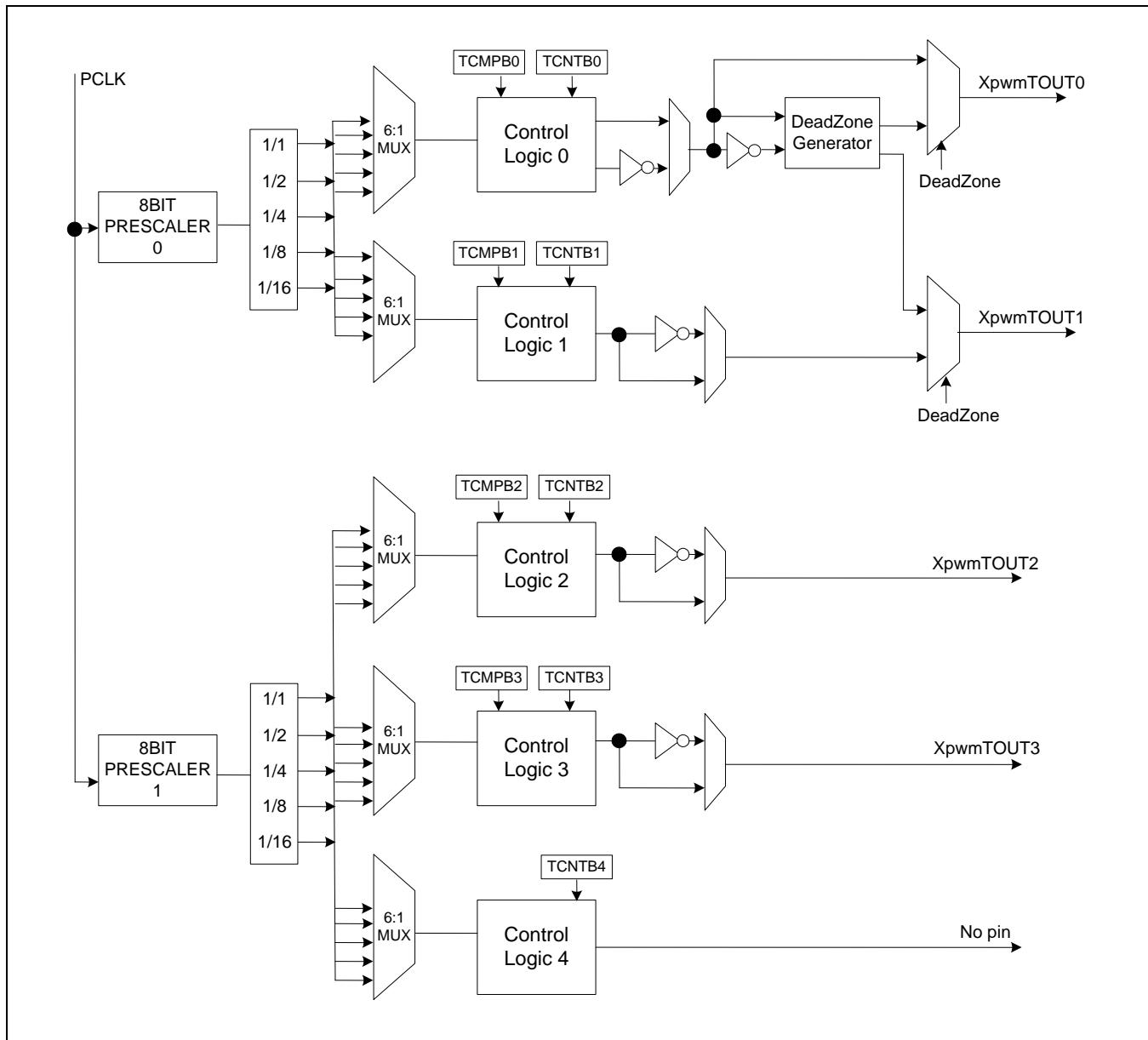


Figure 11-2 PWM TIMER Clock Tree Diagram

Each timer can generate level interrupts.

11.2 Features

The features of PWM are:

- Five 32-bit timers.
- Two 8-bit Clock Prescalers providing first level of division for the PCLK. Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock.
- Programmable Clock Select Logic for individual PWM Channels
- Four Independent PWM Channels with Programmable Duty Control and Polarity
- Static Configuration: It stops PWM.
- Dynamic Configuration: PWM is running
- Auto-Reload and One-Shot Pulse Mode
- Dead Zone Generator on two PWM Outputs
- Level Interrupt Generation

The PWM has two operation modes. They are:

- Auto-Reload Mode:
In this mode, continuous PWM pulses are generated based on programmed duty cycle and polarity.
- One-Shot Pulse Mode:
In this mode, only one PWM pulse is generated based on programmed duty cycle and polarity.

To control the functionality of PWM, 18 special function registers are provided. The PWM is an AMBA slave module which has programmable outputs and a clock input and the PWM connects to the Advanced Peripheral Bus (APB). These 18 special function registers within PWM are accessed via APB transactions.

11.3 PWM Operation

PWM timer of Exynos 4412 can operate as a general timer and a pulse generator with TOUT signal.

11.3.1 Prescaler and Divider

An 8-bit prescaler and 3-bit divider generates these output frequencies:

[Table 11-1](#) describes the minimum and maximum resolution based on prescaler and clock divider values.

Table 11-1 Minimum and Maximum Resolution Based on Prescaler and Clock Divider Values

| 4-bit Divider Settings | Minimum Resolution (Prescaler Value = 1) | Maximum Resolution (Prescaler Value = 255) | Maximum Interval (TCNTBn = 4294967295) |
|------------------------|---|---|---|
| 1/1 (PCLK = 66 MHz) | 0.030 μ s (33.0 MHz) | 3.879 μ s (257.8 kHz) | 16659.27s |
| 1/2 (PCLK = 66 MHz) | 0.061 μ s (16.5 MHz) | 7.758 μ s (128.9 kHz) | 33318.53s |
| 1/4 (PCLK = 66 MHz) | 0.121 μ s (8.25 MHz) | 15.515 μ s (64.5 kHz) | 66637.07s |
| 1/8 (PCLK = 66 MHz) | 0.242 μ s (4.13 MHz) | 31.03 μ s (32.2 kHz) | 133274.14s |
| 1/16 PCLK = 66 MHz) | 0.485 μ s (2.06 MHz) | 62.061 μ s (16.1 kHz) | 266548.27s |

11.3.2 Basic Timer Operation

[Figure 11-3](#) illustrates the timer operations.

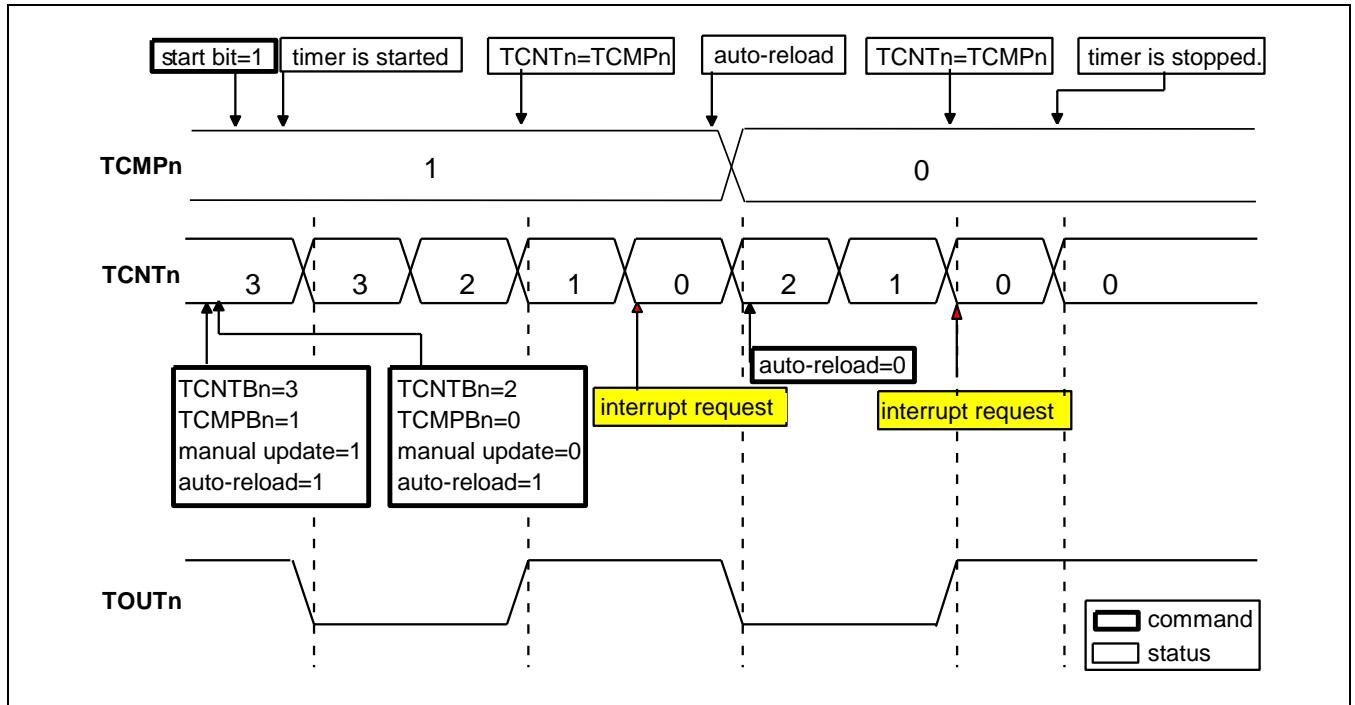


Figure 11-3 Timer Operations

The timer (except the timer channel 4) includes four registers. They are:

- TCNTBn
- TCNTn
- TCMPBn
- TCMPn

If the timer reaches 0, then TCNTBn and TCMPBn registers are loaded into TCNTn and TCMPn. If TCNTn reaches 0, then the interrupt request occurs if it enables the interrupt (TCNTn and TCMPn are the names of the internal registers. It reads the TCNTn register from the TCNTOn register).

To generate interrupt at intervals 3cycle of XpwmTOUTn, set TCNTBn, TCMPBn and TCON register as shown in [Figure 11-3](#).

Steps to generate interrupt:

1. Set TCNTBn = 3 and TCMPBn = 1.
2. Set auto-reload = 1 and manual update = 1.
If manual update bit is 1, then it loads TCNTBn and TCMPBn values to TCNTn and TCMPn.
3. Set TCNTBn = 2 and TCMPBn = 0 for the next operation.
4. Set auto-reload = 1 and manual update = 0.
If you set manual update = 1 at this time, it changes TCNTn to 2 and it changes TCMP to 0.
Therefore, it generates interrupt at interval two-cycle instead of three-cycle.
You should set auto-reload = 1 automatically for the next operation.
5. Set start = 1 for starting the operation. Then TCNTn is down counting.
If TCNTn is 0, it generates interrupt and if auto-reload is enable, it loads TCNTn 2 (TCNTBn value) and it loads TCMPn 0 (TCMPBn value).
6. TCNTn is down counting before it stops.

11.3.3 Auto-Reload and Double Buffering

PWM Timers includes a double buffering feature, which changes the reload value for the next timer operation without stopping the current timer operation.

The timer value is written into TCNTBn (Timer Count Buffer register) and the current counter value of the timer is read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value does not reflect the current state of the counter but the reload value for the next timer duration.

Auto-reload is a copy function that a value of the TCNTBn is copied to the TCNTn when the TCNTn reaches 0. The value written to TCNTBn, is loaded to TCNTn if the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, then TCNTn does not operate further.

[Figure 11-4](#) illustrates the example of double buffering feature.

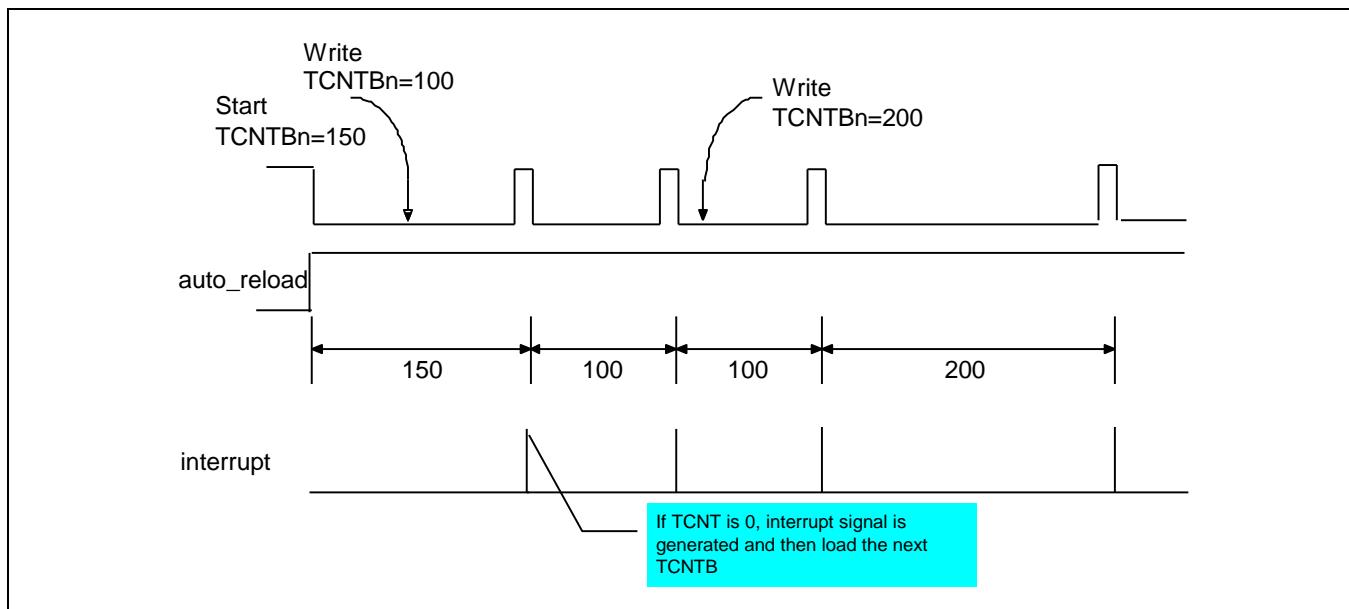


Figure 11-4 Example of Double Buffering Feature

11.3.4 Timer Operation Example

[Figure 11-5](#) illustrates the example of a timer operation.

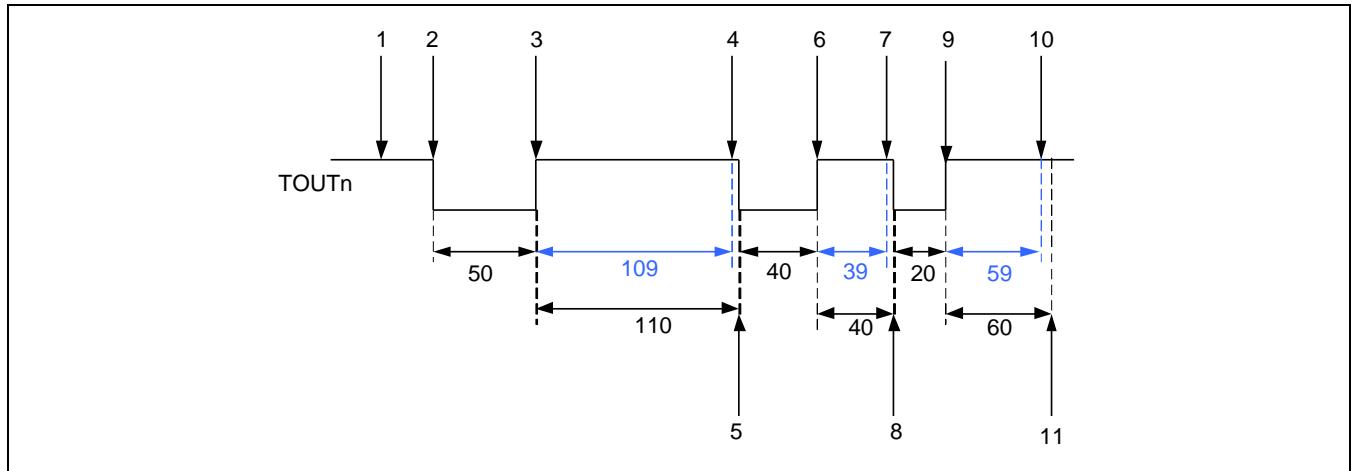


Figure 11-5 Example of a Timer Operation

Steps to use PWM as a timer:

1. Enable the auto-reload feature.
2. Set the TCNTBn as 159 (50 + 109) and TCMPBn as 109.
3. Set the manual update bit On and set the manual update bit Off.
4. Set the inverter On/Off bit. The manual update bit sets TCNTn and TCMPn to the value of TCNTBn and TCMPBn.
5. Set TCNTBn and TCMPBn as 79 (40 + 39) and 39.
6. Start Timer: Set the start bit in TCON.
7. If TCNTn and TCMPn have the same value, then it changes the logic level of TOUTn from low to high
8. When TCNTn reaches 0, it generates interrupt request.
9. It automatically reloads TCNTn and TCMPn with TCNTBn and TCMPBn as (79 (40 + 39)) and 39. In the Interrupt Service Routine (ISR), the TCNTBn and TCMPBn are set as 79 (20 + 59) and 59.
10. If TCNTn and TCMPn have the same value, then it changes the logic level of TOUTn from low to high
11. When TCNTn reaches to 0, it generates interrupt request.
12. It automatically reloads TCNTn and TCMPn with TCNTBn, TCMPBn as (79 (20 + 59)) and 59. It disables the auto-reload and interrupt request to stop the timer in the ISR.
13. If TCNTn and TCMPn have similar value, then it changes the logic level of TOUTn from low to high.
14. Even if TCNTn reaches to 0, it does not generate interrupt request.
15. Because auto-reload is disabled, it does not reload TCNTn and stop the timer.

11.3.5 Initialize Timer (Setting Manual-Up Data and Inverter)

You should define the starting value of the TCNTn, because an auto-reload operation of the timer occurs when the down counter reaches to "0". In this case, the starting value should be loaded by setting "1" to the manual update bit of TCON register.

1. Write the initial value into TCNTBn and TCMPBn.
2. Set the manual update bit and clear only manual update bit of the corresponding timer.

NOTE: We recommend you to set the inverter On/Off bit (whether inverter is used or not).

3. Set the start bit of the corresponding timer to start the timer.

11.3.6 PWM

[Figure 11-6](#) illustrates the example of PWM.

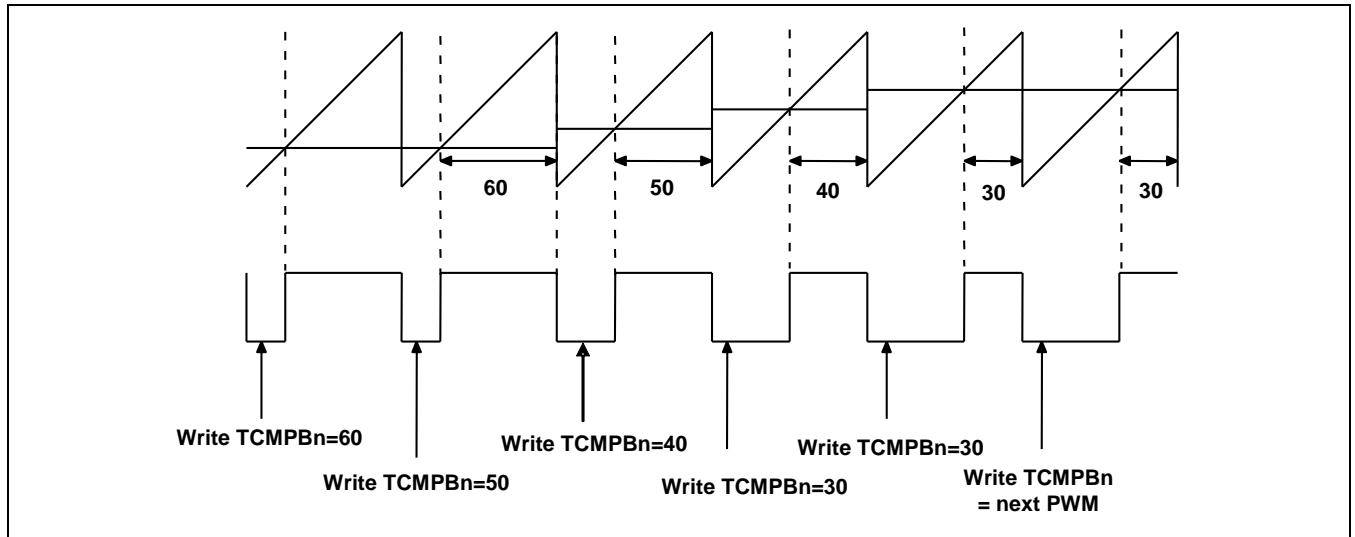


Figure 11-6 Example of PWM

Use TCMPBn to implement the PWM feature. TCNTBn determines PWM frequency. As illustrated in [Figure 11-6](#) TCMPBn determines a PWM value.

For a higher PWM value, decrease the TCMPBn value. For a lower PWM value, increase the TCMPBn value. If you enable the output inverter, the increment/decrement can be disabled.

Due to the double buffering feature, you should write a counter value for next PWM cycle into the TCMPBn register.

11.3.7 During Current ISR. (Interrupt Service Routine) Output Level Control

[Figure 11-7](#) illustrates the inverter On/Off.

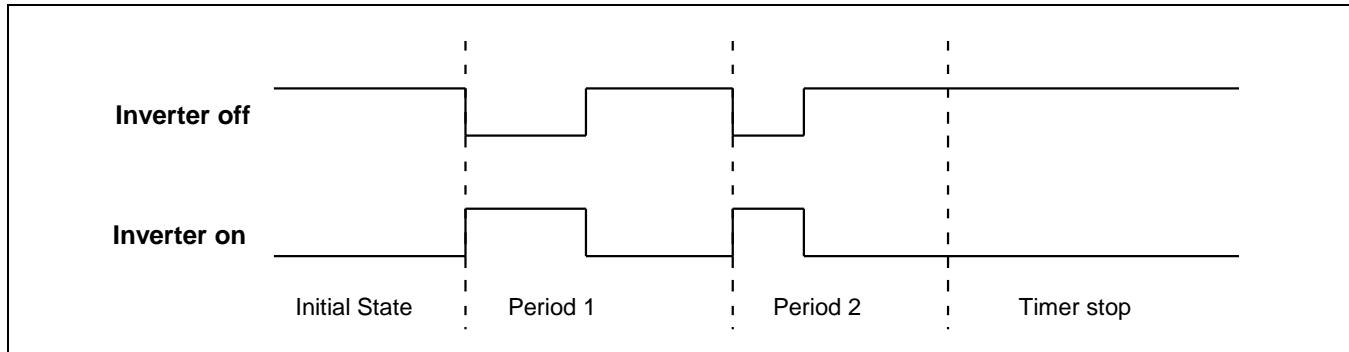


Figure 11-7 Inverter On/Off

Steps to maintain TOUT as high or low when inverter is turned Off:

1. Turn-Off the auto-reload bit. Then, TOUTn goes to high level and it stops the timer after TCNTn reaches to 0. This method is recommended.
2. Stop the timer by clearing the timer start/stop bit to 0. If TCNTn <= TCMPn, the output level is high. If TCNTn > TCMPn, the output level is low.
3. You can invert TOUTn signal by setting "1" to Inverter On/Off bit of TCON register. The inverter removes the additional circuit to adjust the output level.

11.3.8 Dead Zone Generator

Dead Zone Generator feature inserts the time gap between a turn-off and turn-on of two different switching devices. This time gap prohibits the two switching devices turning On simultaneously even for a very short duration.

TOUT_0 specifies the PWM output. nTOUT_0 specifies the inversion of the TOUT_0. If you enable the dead-zone, the output wave-form of TOUT_0 and nTOUT_0 become TOUT_0_DZ and nTOUT_0_DZ. Dead-zone interval cannot turn on TOUT0_DZ and nTOUT_0_DZ simultaneously. For functional accuracy, it should set the dead zone length smaller than compare counter value.

[Figure 11-8](#) illustrates the waveform when it enables Dead Zone feature.

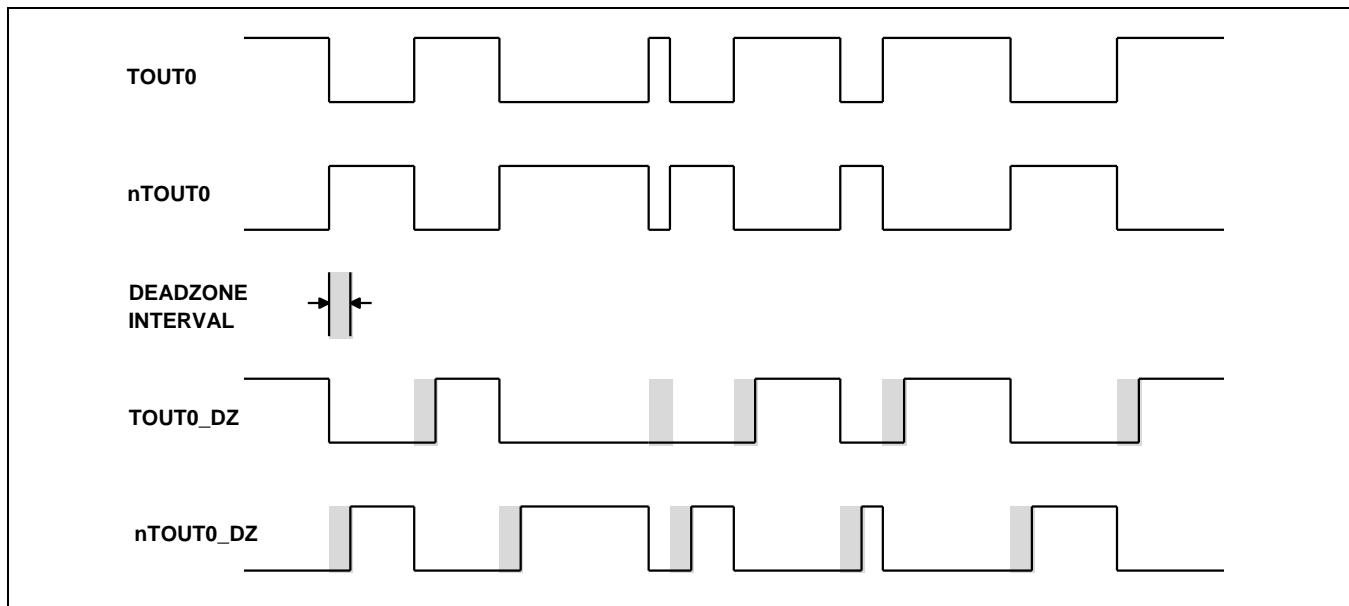


Figure 11-8 Waveform when a Dead Zone Feature is Enabled

11.4 I/O Description

| Signal | I/O | Description | Pad | Type |
|--------|--------|------------------|-------------|-------|
| TOUT_0 | Output | PWMTIMER TOUT[0] | XpwmTOUT[0] | muxed |
| TOUT_1 | Output | PWMTIMER TOUT[1] | XpwmTOUT[1] | muxed |
| TOUT_2 | Output | PWMTIMER TOUT[2] | XpwmTOUT[2] | muxed |
| TOUT_3 | Output | PWMTIMER TOUT[3] | XpwmTOUT[3] | muxed |

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

11.5 Register Description

11.5.1 Register Map Summary

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)

| Register | Offset | Description | Reset Value |
|------------|--------|--|-------------|
| TCFG0 | 0x0000 | Specifies the timer configuration register 0 that configures the two 8-bit prescaler and dead-zone or dead zone length | 0x0000_0101 |
| TCFG1 | 0x0004 | Specifies the timer configuration register 1 that controls five MUX select bit | 0x0000_0000 |
| TCON | 0x0008 | Specifies the timer control register | 0x0000_0000 |
| TCNTB0 | 0x000C | Specifies the timer 0 count buffer register | 0x0000_0000 |
| TCMPB0 | 0x0010 | Specifies the timer 0 compare buffer register | 0x0000_0000 |
| TCNTO0 | 0x0014 | Specifies the timer 0 count observation register | 0x0000_0000 |
| TCNTB1 | 0x0018 | Specifies the timer 1 count buffer register | 0x0000_0000 |
| TCMPB1 | 0x001C | Specifies the timer 1 compare buffer register | 0x0000_0000 |
| TCNTO1 | 0x0020 | Specifies the timer 1 count observation register | 0x0000_0000 |
| TCNTB2 | 0x0024 | Specifies the timer 2 count buffer register | 0x0000_0000 |
| TCMPB2 | 0x0028 | Specifies the timer 2 compare buffer register | 0x0000_0000 |
| TCNTO2 | 0x002C | Specifies the timer 2 count observation register | 0x0000_0000 |
| TCNTB3 | 0x0030 | Specifies the timer 3 count buffer register | 0x0000_0000 |
| TCMPB3 | 0x0034 | Specifies the timer 3 compare buffer register | 0x0000_0000 |
| TCNTO3 | 0x0038 | Specifies the timer 3 count observation register | 0x0000_0000 |
| TCNTB4 | 0x003C | Specifies the timer 4 count buffer register | 0x0000_0000 |
| TCNTO4 | 0x0040 | Specifies the timer 4 count observation register | 0x0000_0000 |
| TINT_CSTAT | 0x0044 | Specifies the timer interrupt control and status register | 0x0000_0000 |

11.5.1.1 TCFG0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0000, Reset Value = 0x0000_0101

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:24] | – | Reserved Bits | 0x00 |
| Dead zone length | [23:16] | RW | Dead zone length | 0x00 |
| Prescaler 1 | [15:8] | RW | Prescaler 1 value for Timer 2, 3, and 4 | 0x01 |
| Prescaler 0 | [7:0] | RW | Prescaler 0 value for timer 0 and 1 | 0x01 |

Timer Input Clock Frequency = PCLK/({prescaler value + 1})/{divider value}

{prescaler value} = 1 to 255

{divider value} = 1, 2, 4, 8, 16

Dead zone length = 0 to 254

NOTE: If deadzone length is set as "n", real Dead Zone length is "n + 1" (n = 0 to 254).

11.5.1.2 TCFG1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31:20] | - | Reserved | 0x000 |
| Divider MUX4 | [19:16] | RW | Selects Mux input for PWM timer 4 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 | 0x0 |
| Divider MUX3 | [15:12] | RW | Selects Mux input for PWM timer 3 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 | 0x0 |
| Divider MUX2 | [11:8] | RW | Selects Mux input for PWM timer 2 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 | 0x0 |
| Divider MUX1 | [7:4] | RW | Selects Mux input for PWM timer 1 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 | 0x0 |
| Divider MUX0 | [3:0] | RW | Selects Mux input for PWM timer 0 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 | 0x0 |

11.5.1.3 TCON

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------------------|---------|------|--|-------------|
| RSVD | [31:23] | – | Reserved Bits | 0x000 |
| Timer 4 auto reload on/off | [22] | RW | 0 = One-shot 1 = Interval mode (auto-reload) | 0x0 |
| Timer 4 manual update | [21] | RW | 0 = No operation 1 = Updates TCNTB4 | 0x0 |
| Timer 4 start/stop | [20] | RW | 0 = Stops Timer 4 1 = Starts Timer 4 | 0x0 |
| Timer 3 auto reload on/off | [19] | RW | 0 = One-shot 1 = Interval mode(auto-reload) | 0x0 |
| Timer 3 output inverter on/off | [18] | RW | 0 = Inverter Off 1 = TOUT_3 inverter-on | 0x0 |
| Timer 3 manual update | [17] | RW | 0 = No operation 1 = Updates TCNTB3 | 0x0 |
| Timer 3 start/stop | [16] | RW | 0 = Stops Timer 3 1 = Starts Timer 3 | 0x0 |
| Timer 2 auto reload on/off | [15] | RW | 0 = One-shot 1 = Interval mode (auto-reload) | 0x0 |
| Timer 2 output inverter on/off | [14] | RW | 0 = Inverter Off 1 = TOUT_2 inverter-on | 0x0 |
| Timer 2 manual update | [13] | RW | 0 = No operation 1 = Updates TCNTB2,TCMPB2 | 0x0 |
| Timer 2 start/stop | [12] | RW | 0 = Stops Timer 2 1 = Starts Timer 2 | 0x0 |
| Timer 1 auto reload on/off | [11] | RW | 0 = One-shot 1 = Interval mode (auto-reload) | 0x0 |
| Timer 1 output inverter on/off | [10] | RW | 0 = Inverter Off 1 = TOUT_1 inverter-on | 0x0 |
| Timer 1 manual update | [9] | RW | 0 = No operation 1 = Updates TCNTB1 andTCMPB1 | 0x0 |
| Timer 1 start/stop | [8] | RW | 0 = Stops Timer 1 1 = Starts Timer 1 | 0x0 |
| Reserved | [7:5] | – | Reserved Bits | 0x0 |
| Dead zone enable/disable | [4] | RW | Enables/Disables Dead zone generator | 0x0 |
| Timer 0 auto reload on/off | [3] | RW | 0 = One-shot 1 = Interval mode (auto-reload) | 0x0 |
| Timer 0 output | [2] | RW | 0 = Inverter Off | 0x0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------------|-----|------|---|-------------|
| inverter on/off | | | 1 = TOUT_0 inverter-on | |
| Timer 0 manual update | [1] | RW | 0 = No operation 1 = Updates TCNTB0 and TCMPB0 | 0x0 |
| Timer 0 start/stop | [0] | RW | 0 = Stops Timer 0 1 = Starts Timer 0 | 0x0 |

11.5.1.4 TCNTB0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|--------|------|-------------------------------|-------------|
| Timer 0 count buffer | [31:0] | RW | Timer 0 Count Buffer register | 0x0000_0000 |

11.5.1.5 TCMPB0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------------|--------|------|---------------------------------|-------------|
| Timer 0 compare buffer | [31:0] | RW | Timer 0 Compare Buffer register | 0x0000_0000 |

11.5.1.6 TCNTO0

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------|--------|------|------------------------------------|-------------|
| Timer 0 count observation | [31:0] | R | Timer 0 Count Observation register | 0x0000_0000 |

11.5.1.7 TCNTB1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|--------|------|-------------------------------|-------------|
| Timer 1 count buffer | [31:0] | RW | Timer 1 Count Buffer register | 0x0000_0000 |

11.5.1.8 TCMPB1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------------|--------|------|---------------------------------|-------------|
| Timer 1 compare buffer | [31:0] | RW | Timer 1 Compare Buffer register | 0x0000_0000 |

11.5.1.9 TCNTO1

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------|--------|------|------------------------------------|-------------|
| Timer 1 count observation | [31:0] | R | Timer 1 Count Observation register | 0x0000_0000 |

11.5.1.10 TCNTB2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|--------|------|-------------------------------|-------------|
| Timer 2 count buffer | [31:0] | RW | Timer 2 Count Buffer register | 0x0000_0000 |

11.5.1.11 TCMPB2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------------|--------|------|---------------------------------|-------------|
| Timer 2 compare buffer | [31:0] | RW | Timer 2 Compare Buffer register | 0x0000_0000 |

11.5.1.12 TCNTO2

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------|--------|------|------------------------------------|-------------|
| Timer 2 count observation | [31:0] | R | Timer 2 Count Observation register | 0x0000_0000 |

11.5.1.13 TCNTB3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|--------|------|-------------------------------|-------------|
| Timer 3 count buffer | [31:0] | RW | Timer 3 Count Buffer register | 0x0000_0000 |

11.5.1.14 TCMPB3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------------|--------|------|---------------------------------|-------------|
| Timer 3 compare buffer | [31:0] | RW | Timer 3 Compare Buffer register | 0x0000_0000 |

11.5.1.15 TCNTO3

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------|--------|------|------------------------------------|-------------|
| Timer 3 count observation | [31:0] | R | Timer 3 Count Observation register | 0x0000_0000 |

11.5.1.16 TCNTB4

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x003C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------------|--------|------|-------------------------------|-------------|
| Timer 4 count buffer | [31:0] | RW | Timer 4 Count Buffer register | 0x0000_0000 |

11.5.1.17 TCNTO4

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------|--------|------|------------------------------------|-------------|
| Timer 4 count observation | [31:0] | R | Timer 4 Count Observation register | 0x0000_0000 |

11.5.1.18 TINT_CSTAT

- Base Address: 0x139D_0000 (PWM)
- Base Address: 0x1216_0000 (PWM_ISP)
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------------|---------|------|---|-------------|
| RSVD | [31:10] | – | Reserved Bits | 0x00000 |
| Timer 4 interrupt status | [9] | RW | Timer 4 interrupt status bit. It clears by writing "1" on this bit. | 0x0 |
| Timer 3 interrupt status | [8] | RW | Timer 3 interrupt status bit. It clears by writing "1" on this bit. | 0x0 |
| Timer 2 interrupt status | [7] | RW | Timer 2 interrupt status bit. It clears by writing "1" on this bit. | 0x0 |
| Timer 1 interrupt status | [6] | RW | Timer 1 interrupt status bit. It clears by writing "1" on this bit. | 0x0 |
| Timer 0 interrupt status | [5] | RW | Timer 0 interrupt status bit. It clears by writing "1" on this bit. | 0x0 |
| Timer 4 interrupt enable | [4] | RW | Enables timer 4 interrupt 0 = Disables Timer 4 interrupt 1 = Enables Timer 4 interrupt | 0x0 |
| Timer 3 interrupt enable | [3] | RW | Enables timer 3 interrupt 0 = Disables Timer 3 interrupt 1 = Enables Timer 3 interrupt | 0x0 |
| Timer 2 interrupt enable | [2] | RW | Enables timer 2 interrupt 0 = Disables Timer 2 interrupt 1 = Enables Timer 2 interrupt | 0x0 |
| Timer 1 interrupt enable | [1] | RW | Enables timer 1 interrupt 0 = Disables Timer 1 interrupt 1 = Enables Timer 1 interrupt | 0x0 |
| Timer 0 interrupt enable | [0] | RW | Enables timer 0 interrupt. 0 = Disables Timer 0 interrupt 1 = Enables Timer 0 interrupt | 0x0 |

12 Watchdog Timer

12.1 Overview

Watchdog Timer (WDT) in Exynos 4412 is a timing device. You can use this device to resume the controller operation after malfunctioning due to noise and system errors. You can use WDT as a normal 16-bit interval timer to request interrupt service. WDT generates the reset signal.

12.2 Features

The features of WDT are:

- Supports normal interval timer mode with interrupt request.
- Activates internal reset signal if the timer count value reaches 0 (time-out).
- Supports level-triggered interrupt mechanism.

12.3 Functional Description

This section includes:

- WDT operation
- WTDAT and WTCNT
- WDT Start
- Consideration of debugging environment

12.3.1 WDT Operation

WDT uses PCLK as its source clock. The 8-bit Prescaler prescales the PCLK frequency to generate the corresponding WDT and it divides the resulting frequency again.

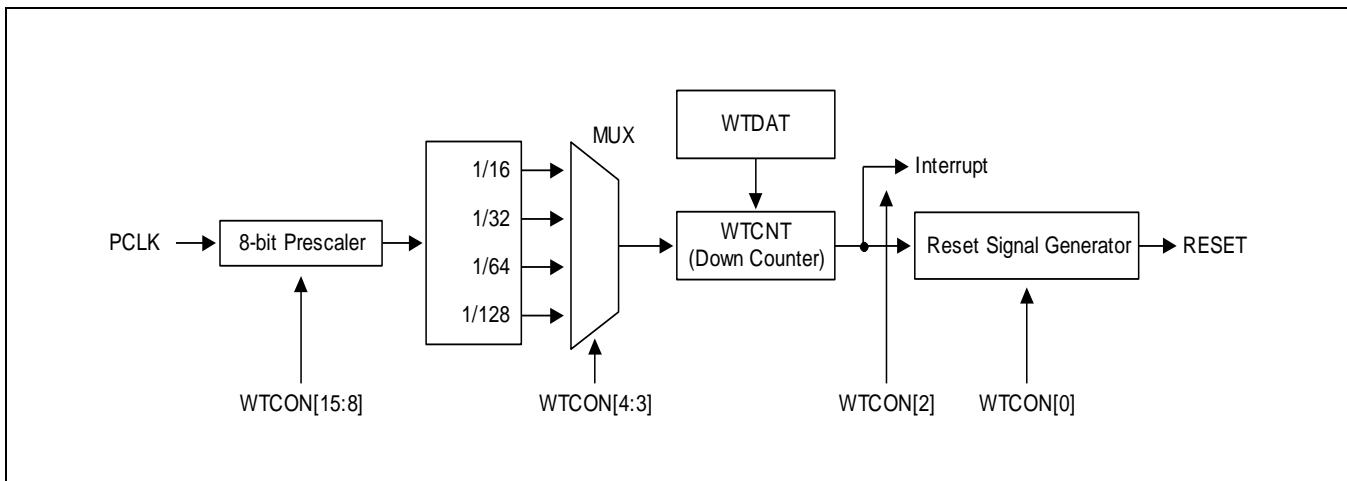


Figure 12-1 Watchdog Timer Block Diagram

[Figure 12-1](#) illustrates the functional block diagram of WDT.

The Watchdog Timer Control (WTCON) specifies the prescaler value and frequency division factor. Valid prescaler values range from 0 to $(2^8 - 1)$. You can select the frequency division factor as: 16, 32, 64, or 128.

Use this equation to calculate the WDT clock frequency and the duration of each timer clock cycle:

$$t_{\text{watchdog}} = 1 / (\text{PCLK} / (\text{Prescaler value} + 1) / \text{Division_factor})$$

12.3.2 WTDAT and WTCNT

After you enable the WDT, you cannot reload the value of the Watchdog Timer Data (WTDAT) register automatically into the Watchdog Timer Counter (WTCNT) register. Therefore, you must write an initial value to the WTCN register before WDT starts.

12.3.3 WDT Start

To start WDT, set WTCON[0] and WTCON[5] as 1.

12.3.4 Consideration of Debugging Environment

WDT should not operate if the Exynos 4412 is in debug mode that uses Embedded In Circuit Debugger (ICE).

WDT determines whether CPU core is currently in the debug mode from the CPU core signal (DBGACK signal). After CPU core asserts the DBGACK signal, it does not activate the reset output of WDT as WDT expires.

12.4 Register Description

12.4.1 Register Map Summary

- Base Address: 0x1006_0000

| Register | Offset | Description | Reset Value |
|----------|--------|---|-------------|
| WTCON | 0x0000 | Watchdog timer control register | 0x0000_8021 |
| WTDAT | 0x0004 | Watchdog timer data register | 0x0000_8000 |
| WTCNT | 0x0008 | Watchdog timer count register | 0x0000_8000 |
| WTCLRINT | 0x000C | Watchdog timer interrupt clear register | Undefined |

12.4.1.1 WTCON

- Base Address: 0x1006_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_8021

| Name | Bit | Type | Description | Reset Value |
|----------------------|---------|------|---|-------------|
| RSVD | [31:16] | – | Reserved | 0 |
| Prescaler value | [15:8] | RW | Prescaler value. The valid range is from 0 to ($2^8 - 1$). | 0x80 |
| RSVD | [7:6] | – | Reserved These two bits should be 00 in normal operation. | 00 |
| WDT timer | [5] | RW | Enables or disables WDT bit. 0 = Disables WDT bit 1 = Enables WDT bit | 1 |
| Clock select | [4:3] | RW | Determines the clock division factor. 00 = 16 01 = 32 10 = 64 11 = 128 | 00 |
| Interrupt generation | [2] | RW | Enables or disables interrupt bit. 0 = Disables interrupt bit 1 = Enables interrupt bit | 0 |
| RSVD | [1] | – | Reserved. This bit should be 0 in normal operation. | 0 |
| Reset enable/disable | [0] | RW | Enables or disables WDT output bit for reset signal. 0 = Disables the reset function of the watchdog timer. 1 = Asserts reset signal of the Exynos 4412 at watchdog time-out. | 1 |

The WTCON register:

- Allows you to enable/disable the watchdog timer
- Selects the clock signal from four different sources
- Enables/disables interrupts
- Enables/disables the watchdog timer output

You can use WDT to restart the Exynos 4412 to recover from malfunction. If you do not want to restart the controller, disable WDT.

If you want to use the normal timer that WDT provides, enable the interrupt and disable the WDT.

12.4.1.2 WTDAT

- Base Address: 0x1006_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_8000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|-----------------------------|-------------|
| RSVD | [31:16] | – | Reserved | 0 |
| Count reload value | [15:0] | RW | WDT count value for reload. | 0x8000 |

The WTDAT register specifies the time-out duration. You cannot load the content of WTDAT into the timer counter at initial WDT operation. However, by using 0x8000 (initial value) drives the WDT counter first time-out. In this case, WDT counter logic reloads the value of WTDAT automatically into WTCNT.

12.4.1.3 WTCNT

- Base Address: 0x1006_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_8000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|------------------------------------|-------------|
| RSVD | [31:16] | – | Reserved | 0 |
| Count value | [15:0] | RW | The current count value of the WDT | 0x8000 |

The WTCNT register contains the current count values for the WDT during normal operation. WDT counter logic cannot automatically load the content of WTDAT register into the timer count register if it enables the WDT initially. Therefore, you should set the WTCNT register to an initial value before enabling it.

12.4.1.4 WTCLRINT

- Base Address: 0x1006_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|--|-------------|
| Interrupt clear | [31:0] | RW | Write any value to clear the interrupt | – |

You can use the WTCLRINT register to clear the interrupt. Interrupt service routine is responsible to clear the relevant interrupt after the interrupt service is complete. Writing any values on this register clears the interrupt. Reading on this register is not allowed.

13 Universal Asynchronous Receiver and Transmitter

13.1 Overview

A Universal Asynchronous Receiver and Transmitter (UART) in Exynos 4412 provide four independent channels with asynchronous and serial input/output (I/O) ports for general purpose (Ch0 to 3). It also provides a dedicated channel for communication with Global Positioning System (GPS) (Ch4). All the ports operate either in an interrupt-based or a DMA-based mode. UART generates either an interrupt or a DMA request to transfer data to and from CPU and UART. UART supports bit rates up to 4 Mbps. Each UART channel contains two First In First Outs (FIFOs) to receive and transmit data as in:

- 256 bytes in Ch0
- 64 bytes in Ch1 and Ch4
- 16 bytes in Ch2 and Ch3

UART includes:

- Programmable Baud rates
- Infrared (IR) transmitter/receiver
- One or two stop bit insertion
- 5-bit, 6-bit, 7-bit, or 8-bit data width and parity checking

As shown in [Figure 13-1](#), each UART contains:

- Baud-rate generator
- Transmitter
- Receiver
- Control unit

The Baud-rate generator uses SCLK_UART. The transmitter and the receiver contain FIFOs and data shifters. The data to be transmitted is written to Tx FIFO, and copied to the transmit shifter. The data is then shifted out by the transmit data pin (TxDn). The received data is shifted from the receive data pin (RxDn), and copied to Rx FIFO from the shifter.

13.2 Features

Features of UART are:

- RxD0, TxD0, RxD1, TxD1, RxD2, TxD2, RxD3, and TxD3 with either DMA-based or interrupt-based operation
- UART Ch 0, 1, 2, and 3 with IrDA 1.0
- UART Ch 0 with 256 byte FIFO, Ch 1 and 4 with 64 byte FIFO, Ch 2 and 3 with 16 byte FIFO
- UART Ch 0, 1, 2 with nRTS0, nCTS0, nRTS1, nCTS1, nCTS2, and nRTS2 for Auto Flow Control (AFC)
- UART Ch 4 communicates with GPS and it supports AFC
- UART supports handshakes transmit/receive.

13.3 UART Description

This section includes UART operations such as:

- Data transmission
- Data reception
- Interrupt generation
- Baud-rate generation
- Loop-back mode
- Infrared modes
- AFC

[Figure 13-1](#) illustrates the block diagram of UART.

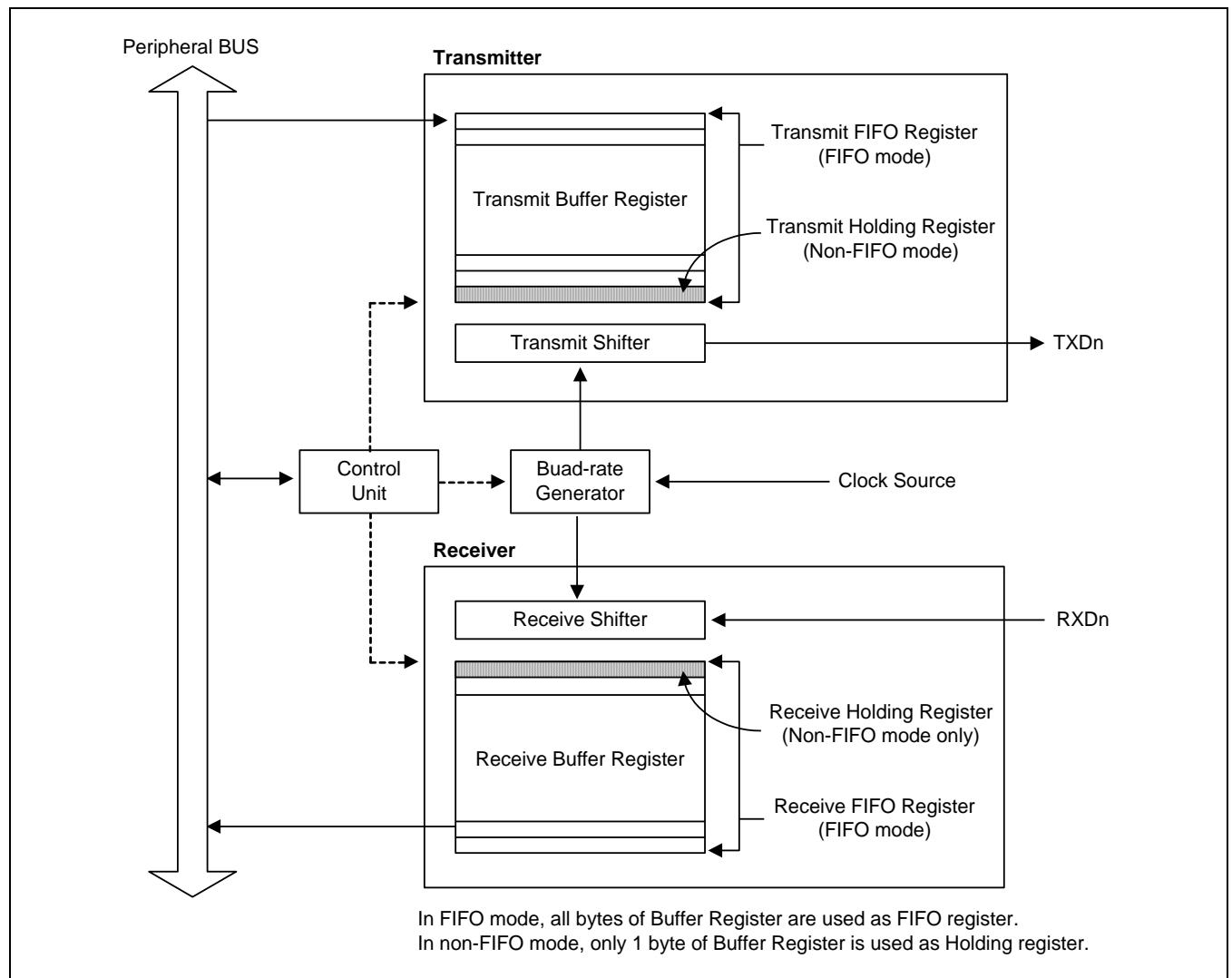


Figure 13-1 Block Diagram of UART

13.3.1 Data Transmission

The data frame for transmission is programmable. It consists of these bits that are specified by the line control register (ULCONn):

- A Start bit
- Five to eight data bits
- An optional parity bit
- One to two stop bits

The transmitter also produces a break condition that forces the serial output to logic 0 state for one-frame transmission time. This block transmits the break signals after it completely transmits the present transmission word. After the break signal transmission, the transmitter continuously transmits data to Tx FIFO (Tx holding register, in case of non-FIFO mode).

13.3.2 Data Reception

The data frame for reception is also programmable. It consists of a start bit, five to eight data bits, an optional parity bit, and one to two stop bits in the line control register (ULCONn). The receiver detects these errors and each of these errors sets an error flag:

- Overrun error: This error indicates that new data has overwritten the old data before the old data was read.
- Parity error: This error indicates that the receiver has detected an unexpected parity condition.
- Frame error: This error indicates that the received data does not have a valid stop bit.
- Break condition: This indicates that the RxDn input is held in the logic 0 state for more than one-frame transmission time.

Receive time-out condition occurs when the Rx FIFO is not empty in the FIFO mode and does not receive any data during the frame time specified in UCON.

13.3.3 AFC

UART0 and UART1 in Exynos 4412 support AFC by using nRTS and nCTS signals.

To connect UART to a Modem, disable the AFC bit in UMCONn register and control the signal of nRTS by using software. The UART4 supports AFC, but it is dedicated for communication with GPS.

In AFC, the nRTS signal depends on the condition of the receiver, whereas the nCTS signals control the operation of transmitter. The transmitter of UART transfers the data to FIFO when nCTS signals are activated. In AFC, nCTS signals means that other FIFO of UART is ready to receive data.

Before the UART receives data, nRTS has to be activated when Rx FIFO has a spare more than 2-byte and has to be inactivated when its receive FIFO has a spare under 1-byte. In AFC, the nRTS signals means that its RX FIFO is ready to receive data).

[Figure 13-2](#) illustrates the UART AFC interface.

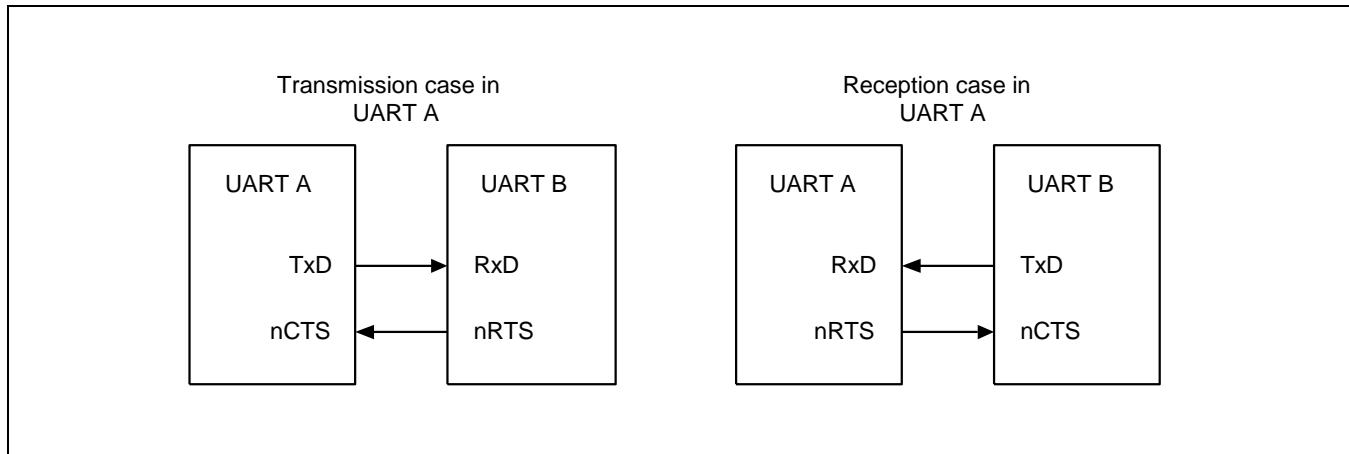


Figure 13-2 UART AFC Interface

13.3.4 Example of Non AFC (Controlling nRTS and nCTS by Software)

This section includes:

- Rx operation with FIFO
- Tx operation with FIFO

13.3.4.1 Rx Operation with FIFO

1. Select the transmit mode (either interrupt or DMA mode).
2. Verify the value of Rx FIFO count in the UFSTATn register. When the value is less than 16, set the value of UMCONn[0] to "1" (activate nRTS). However, when the value equal to or larger than 16, set the value to "0" (inactivate nRTS).
3. Repeat the Step 2 to receive next data.

13.3.4.2 Tx Operation with FIFO

1. Select the transmit mode (either interrupt or DMA mode).
2. Verify the value of UMSTATn[0]. When the value is "1" (activate nCTS), Write data to Tx FIFO register.
3. Repeat the Step 2 to send next data.

13.3.5 Trigger Level of Tx/Rx FIFO and DMA Burst Size in DMA Mode

DMA transaction starts when Tx/Rx data reaches the trigger level of Tx/Rx FIFO of UFCONn register in DMA mode. A single DMA transaction transfers a data whose size is specified as the DMA burst size of UCONn register. The DMA transactions are repeated until Tx/Rx FIFO count is less than the DMA burst size. Thus, DMA burst size should be less than or equal to the trigger level of Tx/Rx FIFO. In general ensure that the trigger level of Tx/Rx FIFO and DMA burst size matches.

13.3.6 RS-232C Interface

To connect UART to the modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD, and nRI signals are required. You can control these signals with general I/O ports by using software as the AFC does not support the RS-232C interface.

13.3.7 Interrupt/DMA Request Generation

Each UART in Exynos 4412 consists of seven status (Tx/Rx/Error) signals, namely:

- Overrun error
- Parity error
- Frame error
- Break condition
- Receive buffer data ready
- Transmit buffer empty
- Transmit shifter empty

The corresponding UART status register (UTRSTATn/UERSTATn) indicates these conditions.

The overrun error, parity error, frame error, and break condition specify the receive error status.

When you set receive-error-status-interrupt-enable bit to 1 in the control register (UCONn), the receive error status generates receive-error-status-interrupt.

When a receive-error-status-interrupt-request is detected, you can identify the source of interrupt by reading the value of UERSTATn.

When you set Receive mode in control register (UCONn) as interrupt request or polling mode, Rx interrupt is generated in this case. When you set Receive mode in control register (UCONn) as interrupt request or polling mode, Rx interrupt is generated in this case. When the receiver transfers data of the receive shifter to the receive FIFO register in FIFO mode, and the number of received data is greater than or equal to the trigger level of Rx FIFO.

In non-FIFO mode, transferring the data of receive shifter to receive holding register causes Rx interrupt in the interrupt request and polling modes.

When the transmitter transfers data from its transmit FIFO register to transmit shifter and the number of data left in transmit FIFO is less than or equal to the trigger level of Tx FIFO, Tx interrupt is generated. This occurs when Transmit mode in control register is selected as Interrupt request or polling mode.

In non-FIFO mode, transferring the data from transmit holding register to transmit shifter causes Tx interrupt in the interrupt request and polling mode.

Remember that the Tx interrupt is always requested when the number of data in the transmit FIFO is smaller than the trigger level. This means that an interrupt is requested as soon as you enable the Tx interrupt, unless you fill the Tx buffer. Fill the Tx buffer first and then enable the Tx interrupt.

The interrupt controllers of Exynos 4412 are of the level-triggered type. Set the interrupt type as "Level" when you program the UART control registers.

When you select Receive and Transmit modes in control register as DMA request mode, DMA request occurs instead of Rx or Tx interrupt in the above situation.

[Table 13-1](#) describes the interrupts in connection with FIFO.

Table 13-1 Interrupts in Connection with FIFO

| Type | FIFO Mode | Non-FIFO Mode |
|-----------------|---|---|
| Rx interrupt | Generated when Rx FIFO count is greater than or equal to the trigger level of received FIFO. Generated when the number of data in FIFO does not reach the trigger level of Rx FIFO and does not receive any data during the specified time (receive time out) in UCON. | Generated by receive holding register whenever receive buffer becomes full. |
| Tx interrupt | Generated when Tx FIFO count is less than or equal to the trigger level of transmit FIFO (trigger level of Tx FIFO). | Generated by transmit holding register whenever transmit buffer becomes empty. |
| Error interrupt | Generated if frame error, parity error, or break signal are detected. Generated if UART receives new data when Rx FIFO is full (overrun error). | Generated by all errors. However when another error occurs at the same time, only one interrupt is generated. |

13.3.8 UART Error Status FIFO

UART contains the error status FIFO besides the Rx FIFO register. The error status FIFO indicates which data among FIFO registers receives an error. An error interrupt is issued only when the data that contains an error is ready to read out. To clear the error status FIFO, you must read out URXHn with an error and UERSTATn.

For example, assume that the UART Rx FIFO receives A, B, C, D, and E characters sequentially and the frame error occurs while receiving "B" and the parity error occurs while receiving "D".

The actual UART receive error does not generate any error interrupt, since it does not read out the character received with an error. The error interrupt occurs if the character is read out.

| Time | Sequence Flow | Error Interrupt | Note |
|------|-------------------------------|---------------------------------------|-------------------------|
| #0 | When no character is read out | – | – |
| #1 | Receives A, B, C, D, and E | – | – |
| #2 | After CPU reads out A | Frame error (in B) interrupt occurs. | "B" has to be read out. |
| #3 | After CPU reads out B | – | – |
| #4 | After CPU reads out C | Parity error (in D) interrupt occurs. | "D" has to be read out. |
| #5 | After CPU reads out D | – | – |
| #6 | After CPU reads out E | – | – |

[Figure 13-3](#) illustrates that UART receives the five characters including two errors.

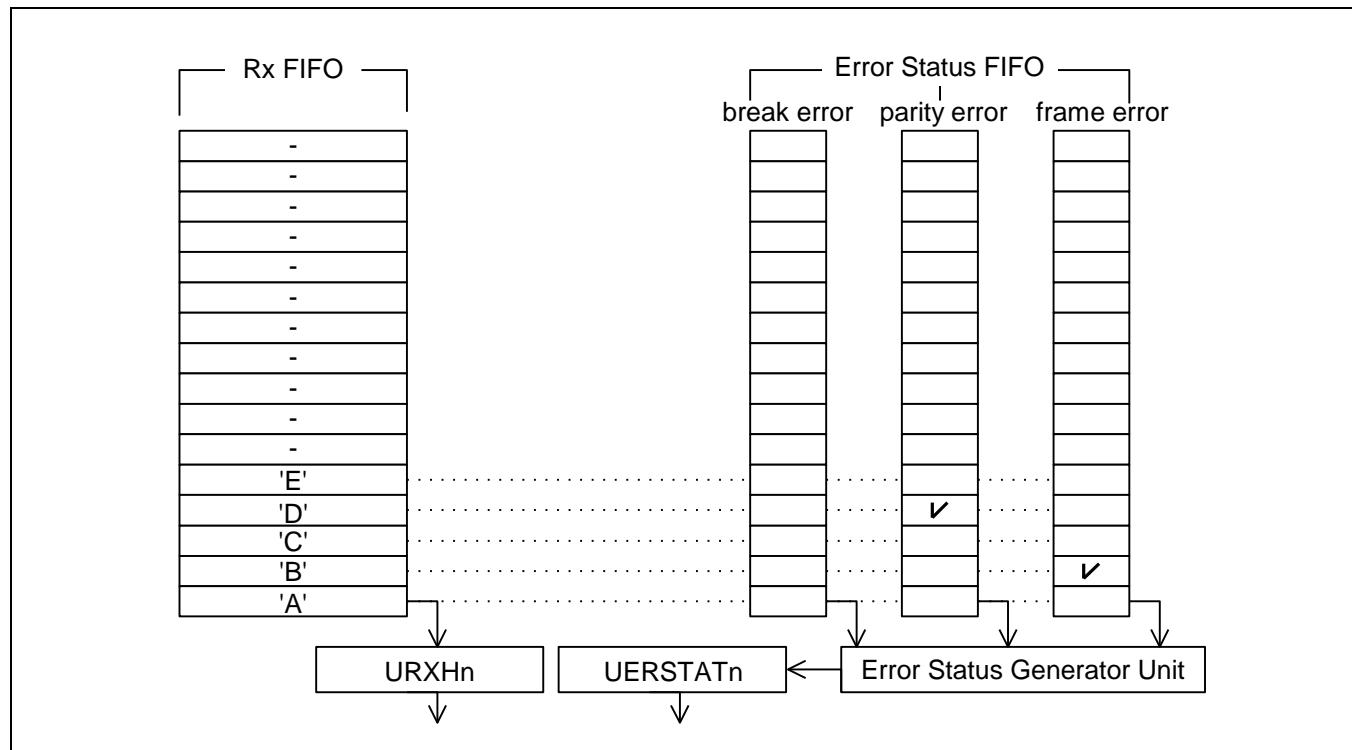


Figure 13-3 UART Receives the Five Characters Including Two Errors

13.3.8.1 Infra-Red Mode

The Exynos 4412 UART block supports both infra-red (IR) transmission and reception. You can select the IR mode by setting the IR-mode bit in the UART line control register (ULCONn). [Figure 13-4](#) illustrates how to implement the IR mode.

In IR transmit mode, the transmit pulse moves at the rate of 3/16, that is, normal serial transmit rate when the transmit data bit is set to 0. However, in IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a 0 value Refer to frame timing diagrams shown in [Figure 13-5](#) and [Figure 13-7](#) for more information.

[Figure 13-4](#) illustrates IrDA function block diagram.

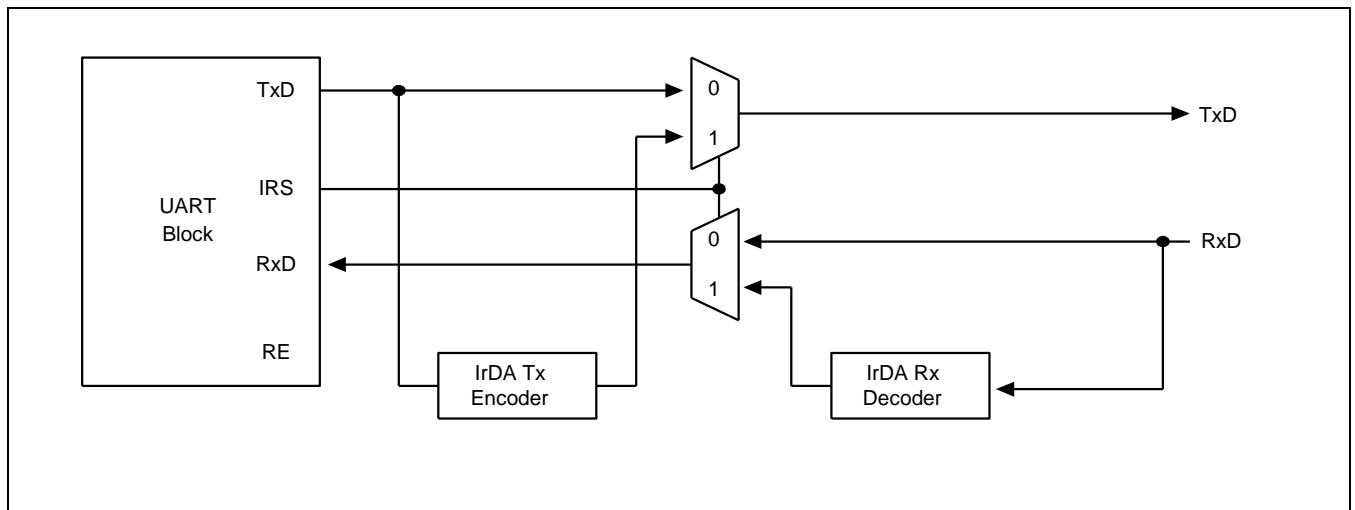


Figure 13-4 IrDA Function Block Diagram

[Figure 13-5](#) illustrates the serial I/O frame timing diagram (Normal UART)

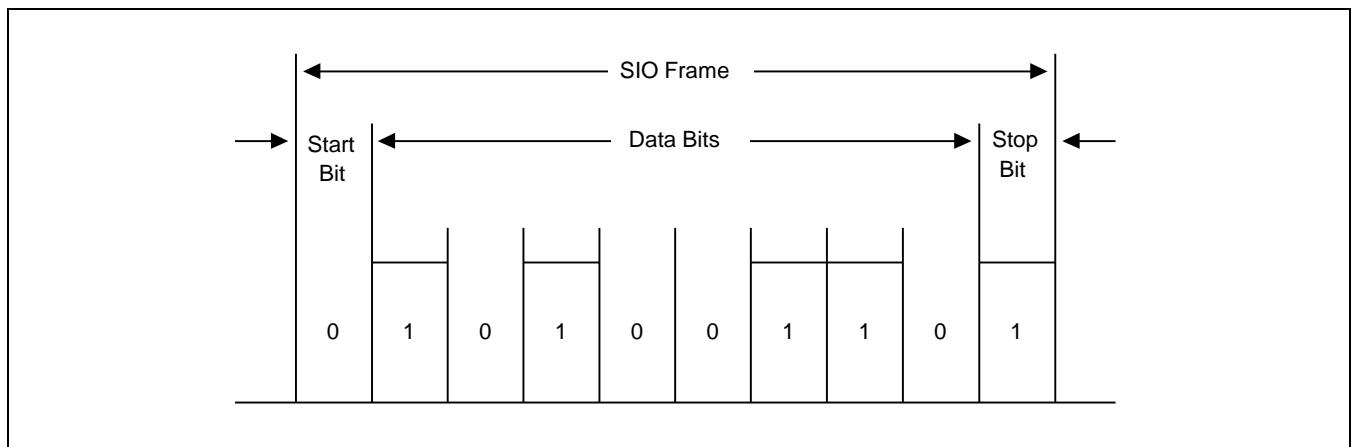


Figure 13-5 Serial I/O Frame Timing Diagram (Normal UART)

[Figure 13-6](#) illustrates the infra-red transmit mode frame timing diagram.

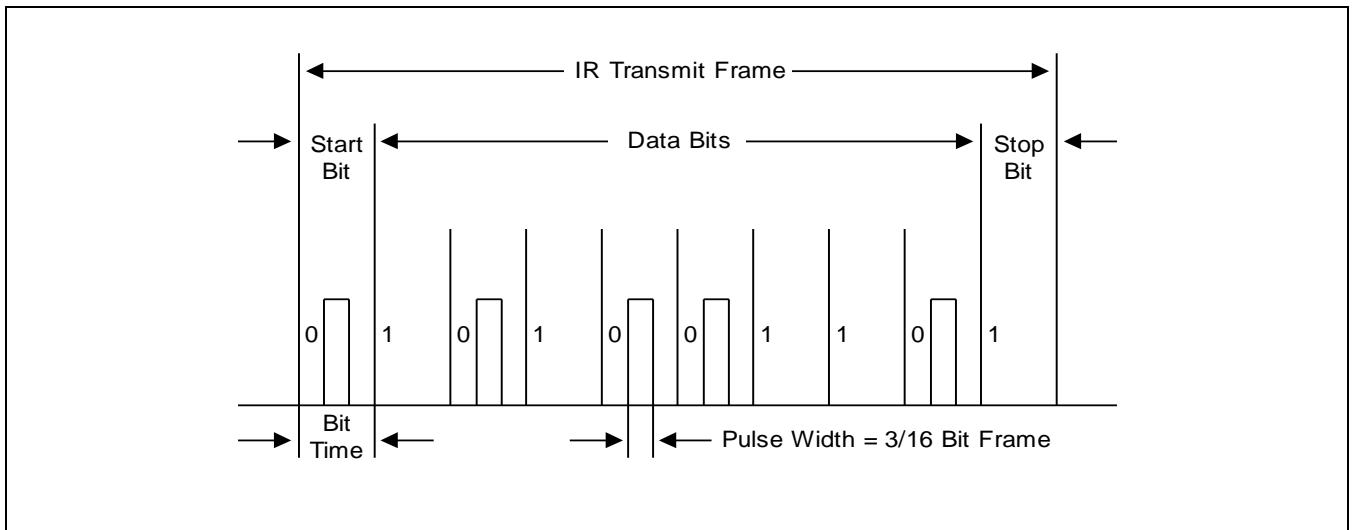


Figure 13-6 Infra-Red Transmit Mode Frame Timing Diagram

[Figure 13-7](#) illustrates the infra-red receive mode frame timing diagram.

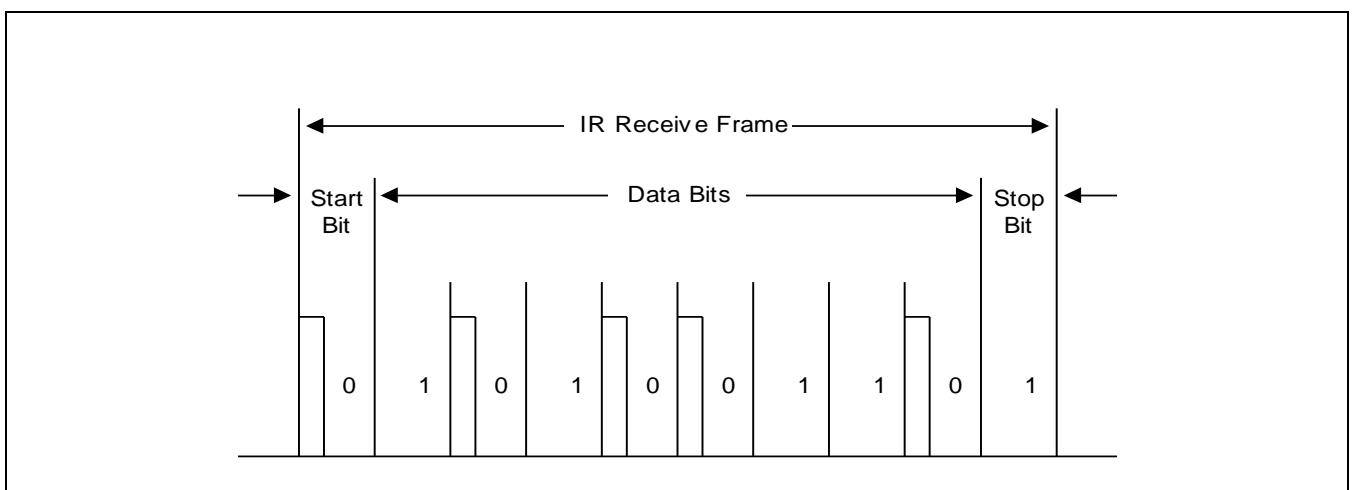


Figure 13-7 Infra-Red Receive Mode Frame Timing Diagram

13.4 UART Input Clock Description

[Figure 13-8](#) illustrates the input clock diagram for UART.

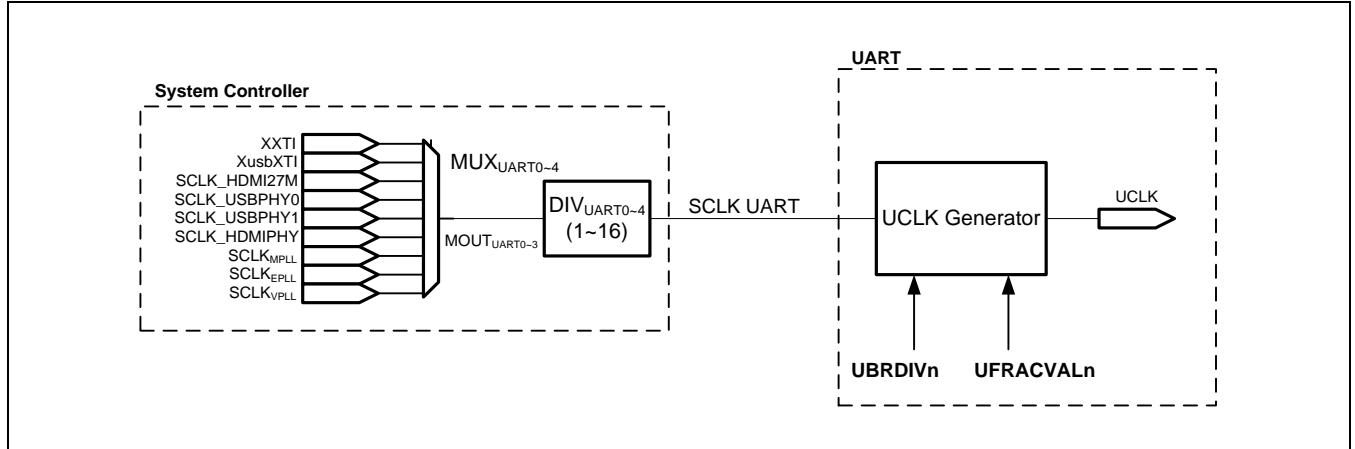


Figure 13-8 Input Clock Diagram for UART

Exynos 4412 provides UART with a variety of clocks. [Figure 13-8](#) illustrates that UART uses SCLK_UART clock, which is from clock controller. You can also select SCLK_UART from various clock sources. Refer to Chapter 7, Clock Controller, for more information.

13.5 I/O Description

| Signal | I/O | Description | Pad | Type |
|-------------|--------|---|----------|-------|
| UART_0_RXD | Input | Receives data for UART0 | XuRXD_0 | muxed |
| UART_0_TXD | Output | Transmits data for UART0 | XuTXD_0 | muxed |
| UART_0_CTSn | Input | Clears to send (active low) for UART0 | XuCTSn_0 | muxed |
| UART_0_RTSp | Output | Requests to send (active low) for UART0 | XuRTSn_0 | muxed |
| UART_1_RXD | Input | Receives data for UART1 | XuRXD_1 | muxed |
| UART_1_TXD | Output | Transmits data for UART1 | XuTXD_1 | muxed |
| UART_1_CTSn | Input | Clears to send (active low) for UART1 | XuCTSn_1 | muxed |
| UART_1_RTSp | Output | Requests to send (active low) for UART1 | XuRTSn_1 | muxed |
| UART_2_RXD | Input | Receives data for UART2 | XuRXD_2 | muxed |
| UART_2_TXD | Output | Transmits data for UART2 | XuTXD_2 | muxed |
| UART_2_CTSn | Input | Clears to send (active low) for UART2 | XuCTSn_2 | muxed |
| UART_2_RTSp | Output | Requests to send (active low) for UART2 | XuRTSn_2 | muxed |
| UART_3_RXD | Input | Receives data for UART3 | XuRXD_3 | muxed |
| UART_3_TXD | Output | Transmits data for UART3 | XuTXD_3 | muxed |

NOTE:

1. Type field indicates whether the signal connects to the dedicated pad or multiplexed signal pads. UART shares external pads with IrDA. To use these pads, set GPIO before the start of UART. Refer to Chapter 6, GPIO, for more information.
2. UART4 has no I/O ports. It communicates with the internal GPS module.

13.6 Register Description

13.6.1 Register Map Summary

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000

| Register | Offset | Description | Reset Value |
|-----------|--------|------------------------------------|-------------|
| ULCONn | 0x0000 | Specifies line control | 0x0000_0000 |
| UCONn | 0x0004 | Specifies control | 0x0000_3000 |
| UFCONn | 0x0008 | Specifies FIFO control | 0x0000_0000 |
| UMCONn | 0x000C | Specifies modem control | 0x0000_0000 |
| UTRSTATn | 0x0010 | Specifies Tx/Rx status | 0x0000_0006 |
| UERSTATn | 0x0014 | Specifies Rx error status | 0x0000_0000 |
| UFSTATn | 0x0018 | Specifies FIFO status | 0x0000_0000 |
| UMSTATn | 0x001C | Specifies modem status | 0x0000_0000 |
| UTXHn | 0x0020 | Specifies transmit buffer | Undefined |
| URXHn | 0x0024 | Specifies receive buffer | 0x0000_0000 |
| UBRDIVn | 0x0028 | Specifies baud rate divisor | 0x0000_0000 |
| UFRACVALn | 0x002C | Specifies divisor fractional value | 0x0000_0000 |
| UINTPn | 0x0030 | Specifies interrupt pending | 0x0000_0000 |
| UINTSPn | 0x0034 | Specifies interrupt source pending | 0x0000_0000 |
| UINTMn | 0x0038 | Specifies interrupt mask | 0x0000_0000 |

13.6.1.1 ULCOnN (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | 0 |
| Infrared Mode | [6] | RW | Determines whether to use the infra-red mode. 0 = Normal mode operation 1 = Infra-red Tx/Rx mode | 0 |
| Parity Mode | [5:3] | RW | Specifies the type of parity that UART generates and checks during UART transmit and receive operation. 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/ checked as 1 111 = Parity forced/ checked as 0 | |
| Number of Stop Bit | [2] | RW | Specifies how many stop bits UART uses to signal end-of-frame signal. 0 = One stop bit per frame 1 = Two stop bit per frame | 0 |
| Word Length | [1:0] | RW | Indicates the number of data bits UART transmits or receives per frame. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits | 0 |

13.6.1.2 UCONn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---|---------|------|--|-------------|
| RSVD | [31:24] | — | Reserved | 0 |
| RSVD | [23] | WO | Reserved | 0 |
| Tx DMA Burst Size | [22:20] | RW | <p>Tx DMA Burst Size It is the data transfer size of one DMA transaction. Tx DMA request triggers the DMA transaction. You must program the DMA program to transfer the same data size as this is the value for a single Tx DMA request.</p> <p>000 = 1 byte (Single) 001 = 4 bytes 010 = 8 bytes 011 = 16 bytes 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved</p> | 0 |
| RSVD | [19] | WO | Reserved | 0 |
| Rx DMA Burst Size | [18:16] | RW | <p>Rx DMA Burst Size It is the data transfer size of one DMA transaction. Rx DMA request triggers the DMA transaction. You must program the DMA program to transfer the same data size as this is the value for a single Rx DMA request.</p> <p>000 = 1 byte (Single) 001 = 4 bytes 010 = 8 bytes 011 = 16 bytes 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved</p> | 0 |
| Rx Timeout Interrupt Interval | [15:12] | RW | <p>Rx Timeout Interrupt Interval Rx interrupt occurs if UART receives no data during $8 \times (N + 1)$ frame time. The default value of this field is 3. It means that the timeout interval is 32 frame time.</p> | 0x3 |
| Rx Time-out with empty Rx FIFO ⁽⁴⁾ | [11] | R/W | <p>Enables Rx time-out feature when Rx FIFO counter is 0. This bit is valid only when UCONn[7] is 1. 0 = Disables Rx time-out feature when Rx FIFO is empty.</p> | 0 |

| Name | Bit | Type | Description | Reset Value |
|----------------------------------|-------|------|---|-------------|
| | | | 1 = Enables Rx time-out feature when Rx FIFO is empty. | |
| Rx Time-out DMA suspend enable | [10] | R/W | Enables the suspension of Rx DMA FSM when Rx Time-out occurs. 0 = Disables suspension of Rx DMA FSM 1 = Enables suspension of Rx DMA FSM | 0 |
| Tx Interrupt Type | [9] | RW | Interrupt request type. (2) 0 = Pulse (UART requests interrupt when the Tx buffer is empty in the non-FIFO mode or when it reaches the trigger level of Tx FIFO in the FIFO mode.) 1 = Level (Interrupt is requested when Tx buffer is empty in the non-FIFO mode or when it reaches the trigger level of Tx FIFO in the FIFO mode.) | 0 |
| Rx Interrupt Type | [8] | RW | Interrupt request type. (2) 0 = Pulse (UART requests interrupt when instant Rx buffer receives data in the non-FIFO mode or when it reaches the trigger level of Rx FIFO in the FIFO mode.) 1 = Level (UART requests interrupt when Rx buffer receives data in the non-FIFO mode or when it reaches the trigger level of Rx FIFO in the FIFO mode.) | 0 |
| Rx Time Out Enable | [7] | RW | Enables/disables Rx time-out interrupts when you enable UART FIFO. The interrupt is a receive interrupt. 0 = Disables 1 = Enables | 0 |
| Rx Error Status Interrupt Enable | [6] | RW | Enables the UART to generate an interrupt upon an exception, such as a break, frame error, parity error, or overrun error during a receive operation. 0 = Does not generate receive error status interrupt. 1 = Generates receive error status interrupt. | 0 |
| Loop-back Mode | [5] | RW | To set this bit to 1 triggers the UART to enter the loop-back mode. This mode is for test purposes only. 0 = Normal operation 1 = Loop-back mode | 0 |
| Send Break Signal | [4] | RWX | To set this bit to 1 triggers UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal. 0 = Normal transmit 1 = Sends the break signal | 0 |
| Transmit Mode | [3:2] | RW | Determines which function is able to Write Tx data to the UART transmit buffer. 00 = Disables 01 = Interrupt request or polling mode 10 = DMA mode 11 = Reserved | 00 |
| Receive Mode | [1:0] | RW | Determines which function is able to Read data from UART receive buffer. 00 = Disables | 00 |

| Name | Bit | Type | Description | Reset Value |
|------|-----|------|--|-------------|
| | | | 01 = Interrupt request or polling mode 10 = DMA mode 11 = Reserved | |

NOTE:

1. DIV_VAL = UBRDIVn + UFRACVAL/16. Refer to [13.6.1.11 UBRDIVn](#) and [13.6.1.12 UFRACVALn](#) for more information.
2. Exynos 4412 uses a level-triggered interrupt controller. Therefore, you must set these bits to 1 for every transfer.
3. If UART does not reach the trigger level of FIFO and does not receive data during the time specified at the "Rx Timeout Interrupt Interval" field in DMA receive mode with FIFO, UART generates the Rx interrupt (receive time out). Ensure to verify the FIFO status and read out the rest.
4. Both UCONn[11] and UCONn[7] should be set to 1 if you want to enable Rx time-out feature when Rx FIFO counter is set to 0.

13.6.1.3 UFCONn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------------|---------|------|--|-------------|
| RSVD | [31:11] | - | Reserved | 0 |
| Tx FIFO Trigger Level | [10:8] | RW | <p>Determines the trigger level of Tx FIFO. When data count of Tx FIFO is less than or equal to the trigger level, Tx interrupt occurs.</p> <p>[Channel 0]</p> <p>000 = 0 byte 001 = 32 bytes 010 = 64 bytes 011 = 96 bytes 100 = 128 bytes 101 = 160 bytes 110 = 192 bytes 111 = 224 bytes</p> <p>[Channel 1, 4]</p> <p>000 = 0 byte 001 = 8 bytes 010 = 16 bytes 011 = 24 bytes 100 = 32 bytes 101 = 40 bytes 110 = 48 bytes 111 = 56 bytes</p> <p>[Channel 2, 3]</p> <p>000 = 0 byte 001 = 2 bytes 010 = 4 bytes 011 = 6 bytes 100 = 8 bytes 101 = 10 bytes 110 = 12 bytes 111 = 14 bytes</p> | 000 |
| RSVD | [7] | - | Reserved | 0 |
| Rx FIFO Trigger Level | [6:4] | RW | <p>Determines the trigger level of Rx FIFO. When data count of Rx FIFO is more than or equal to the trigger level, Rx interrupt occurs.</p> <p>[Channel 0]</p> <p>000 = 32 byte 001 = 64 bytes</p> | 000 |

| Name | Bit | Type | Description | Reset Value |
|---------------|-----|------|--|-------------|
| | | | 010 = 96 bytes 011 = 128 bytes 100 = 160 bytes 101 = 192 bytes 110 = 224 bytes 111 = 256 bytes [Channel 1, 4] 000 = 8 byte 001 = 16 bytes 010 = 24 bytes 011 = 32 bytes 100 = 40 bytes 101 = 48 bytes 110 = 56 bytes 111 = 64 bytes [Channel 2, 3] 000 = 2 byte 001 = 4 bytes 010 = 6 bytes 011 = 8 bytes 100 = 10 bytes 101 = 12 bytes 110 = 14 bytes 111 = 16 bytes | |
| RSVD | [3] | - | Reserved | 0 |
| Tx FIFO Reset | [2] | S | Automatically clears after resetting FIFO 0 = Normal 1 = Tx FIFO reset | 0 |
| Rx FIFO Reset | [1] | S | Automatically clears after resetting FIFO 0 = Normal 1 = Rx FIFO reset | 0 |
| FIFO Enable | [0] | RW | 0 = Disables 1 = Enables | 0 |

NOTE: When UART does not reach the trigger level of FIFO, it does not receive data during the specified timeout interval in DMA receive mode with FIFO. It generates the Rx interrupt (receive time out). Ensure to verify the FIFO status and read out the rest.

13.6.1.4 UMCONn (n = 0, 1, 2, 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------------------|--------|------|--|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| RTS trigger Level | [7:5] | RW | <p>Determines the trigger level of Rx FIFO to control nRTS signal. When it enables AFC bit and Rx FIFO have bytes that are greater than or equal to the trigger level, it deactivates nRTS signal.</p> <p>[Channel 0] 000 = 255 bytes 001 = 224 bytes 010 = 192 bytes 011 = 160 bytes 100 = 128 bytes 101 = 96 bytes 110 = 64 bytes 111 = 32 bytes</p> <p>[Channel 1, 4] 000 = 63 bytes 001 = 56 bytes 010 = 48 bytes 011 = 40 bytes 100 = 32 bytes 101 = 24 bytes 110 = 16 bytes 111 = 8 bytes</p> <p>[Channel 2] 000 = 15 bytes 001 = 14 bytes 010 = 12 bytes 011 = 10 bytes 100 = 8 bytes 101 = 6 bytes 110 = 4 bytes 111 = 2 bytes</p> | 0 |
| Auto Flow Control (AFC) | [4] | RW | 0 = Disables 1 = Enables | 0 |
| Modem Interrupt Enable | [3] | RW | 0 = Disables 1 = Enables | 0 |
| RSVD | [2:1] | — | Reserved (These bits must be 0) | 0 |

| Name | Bit | Type | Description | Reset Value |
|-----------------|-----|------|---|-------------|
| Request to Send | [0] | RW | If AFC bit is enabled, this value will be ignored. In this case, the Exynos 4412 controls nRTS signals automatically. If AFC bit is disabled, the software must control nRTS signal. 0 = "H" level (inactivate nRTS) 1 = "L" level (activate nRTS) | 0 |

NOTE:

1. UART 3 does not support AFC function because the Exynos 4412 has no nRTS3 and nCTS3.
2. In AFC mode, set the trigger level of Rx FIFO lower than the trigger level of RTS because transmitter stops data transfer when it deactivates the nRST signal.

13.6.1.5 UTRSTATn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------------------------------|---------|------|---|-------------|
| RSVD | [31:24] | - | Reserved | 0 |
| RX FIFO count in RX time-out status | [23:16] | R | Capture value of Rx FIFO counter when Rx time-out occurs | 0x00 |
| TX DMA FSM State | [15:12] | R | Current State of Tx DMA FSM 0x0 = IDLE 0x1 = Burst Request 0x2 = Burst Acknowledgement 0x3 = Burst Next (intermediate state for next request) 0x4 = Single Request 0x5 = Single Acknowledgement 0x6 = Single Next (intermediate state for next request) 0x7 = Last Burst Request 0x8 = Last Burst Acknowledgement 0x9 = Last Single Request 0x10 = Last Single Acknowledgement | 0x0 |
| RX DMA FSM State | [11:8] | R | Current State of Rx DMA FSM 0x0 = IDLE 0x1 = Burst Request 0x2 = Burst Acknowledgement 0x3 = Burst Next (intermediate state for next request) 0x4 = Single Request 0x5 = Single Acknowledgement 0x6 = Single Next (intermediate state for next request) 0x7 = Last Burst Request 0x8 = Last Burst Acknowledgement 0x9 = Last Single Request 0x10 = Last Single Acknowledgement | 0x0 |
| RSVD | [7:4] | - | Reserved | 0 |
| RX Time-out status/Clear ¹ | [3] | RWX | RX Time-out status when read. 0 = Rx Time out did not occur 1 = Rx Time out. Clears Rx Time-out status when write. 0 = No operation 1 = Clears Rx Time-out status NOTE: When UCONn[10] is set to 1. writing 1 to this bit resumes Rx DMA FSM that was suspended when Rx time-out | 0 |

| Name | Bit | Type | Description | Reset Value |
|---------------------------|-----|------|---|-------------|
| | | | occurred. | |
| Transmitter empty | [2] | R | This bit is automatically set to 1 when the transmit buffer has no valid data to transmit, and the transmit shift is empty. 0 = Not empty 1 = Transmitter (includes transmit buffer and shifter) empty | 1 |
| Transmit buffer empty | [1] | R | This bit is automatically set to 1 when transmit buffer is empty. 0 = Buffer is not empty 1 = Buffer is empty (in non-FIFO mode, it requests interrupt or DMA). In FIFO mode, it requests interrupt or DMA, when the trigger level of Tx FIFO is set to 00 (Empty). When UART uses FIFO, check for Tx FIFO Count bits and Tx FIFO Full bit in UFSTAT instead of this bit. | 1 |
| Receive buffer data ready | [0] | R | It automatically sets this bit to 1 when receive buffer contains valid data, which is received over the RXDn port. 0 = Buffer is empty 1 = Buffer has a received data (In Non-FIFO mode, it requests interrupt or DMA) When UART uses the FIFO, check for Rx FIFO Count bits and Rx FIFO Full bit in UFSTAT instead of this bit. | 0 |

13.6.1.6 UERSTATn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:4] | – | Reserved | 0 |
| Break Detect | [3] | R | <p>It automatically sets this bit to 1 to indicate that a break signal has been received.</p> <p>0 = No break signal is received 1 = Break signal is received (Interrupt is requested.)</p> | 0 |
| Frame Error | [2] | R | <p>It automatically sets this bit to 1 when a frame error occurs during the receive operation.</p> <p>0 = No frame error occurs during the receive operation 1 = Frame error occurs (Interrupt is requested.) during the receive operation</p> | 0 |
| Parity Error | [1] | R | <p>It automatically sets this bit to 1 when a parity error occurs during the receive operation.</p> <p>0 = No parity error occurs during receive the receive operation 1 = Parity error occurs (Interrupt is requested.) during the receive operation</p> | 0 |
| Overrun Error | [0] | R | <p>It automatically sets this bit to 1 automatically if an overrun error occurs during the receive operation.</p> <p>0 = No overrun error occurs during the receive operation 1 = Overrun error occurs (Interrupt is requested.) during the receive operation</p> | 0 |

NOTE: It clears these bits (UERSATn[3:0]) to 0 when UART error status is Read.

13.6.1.7 UFSTATn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [31:25] | — | Reserved | 0 |
| Tx FIFO Full | [24] | R | It automatically sets this bit to 1 when the transmitted FIFO is full during transmit operation 0 = Not full 1 = Full | 0 |
| Tx FIFO Count | [23:16] | R | Number of data in Tx FIFO NOTE: This field is set to 0 when Tx FIFO is full. | 0 |
| RSVD | [15:10] | — | Reserved | 0 |
| Rx FIFO Error | [9] | R | This bit is set to 1 when Rx FIFO contains invalid data that results from frame error, parity error, or break signal. | 0 |
| Rx FIFO Full | [8] | R | It automatically sets this bit to 1 when the received FIFO is full during receive operation 0 = Not full 1 = Full | 0 |
| Rx FIFO Count | [7:0] | R | Number of data in Rx FIFO NOTE: This field is set to 0 when Rx FIFO is full. | 0 |

13.6.1.8 UMSTAT_n ($n = 0, 1, 2, 4$)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:5] | - | Reserved | 0 |
| Delta CTS | [4] | R | This bit indicates that the nCTS input to the Exynos 4412 has changed its state since the last time CPU read it. Refer to Figure 13-9 for more information. 0 = Has not changed 1 = Has changed NOTE: In UMSTAT4, reset value of this bit is undefined. It depends on the GPIO configuration of GPS. | 0 |
| RSVD | [3:1] | - | Reserved | - |
| Clear to Send | [0] | R | 0 = Does not activates CTS signal (nCTS pin is high) 1 = Activates CTS signal (nCTS pin is low) NOTE: In UMSTAT4, reset value of this bit is undefined. It depends on the GPIO configuration of GPS. | 0 |

[Figure 13-9](#) illustrates the nCTS and delta Clear to Send (CTS) timing diagram.

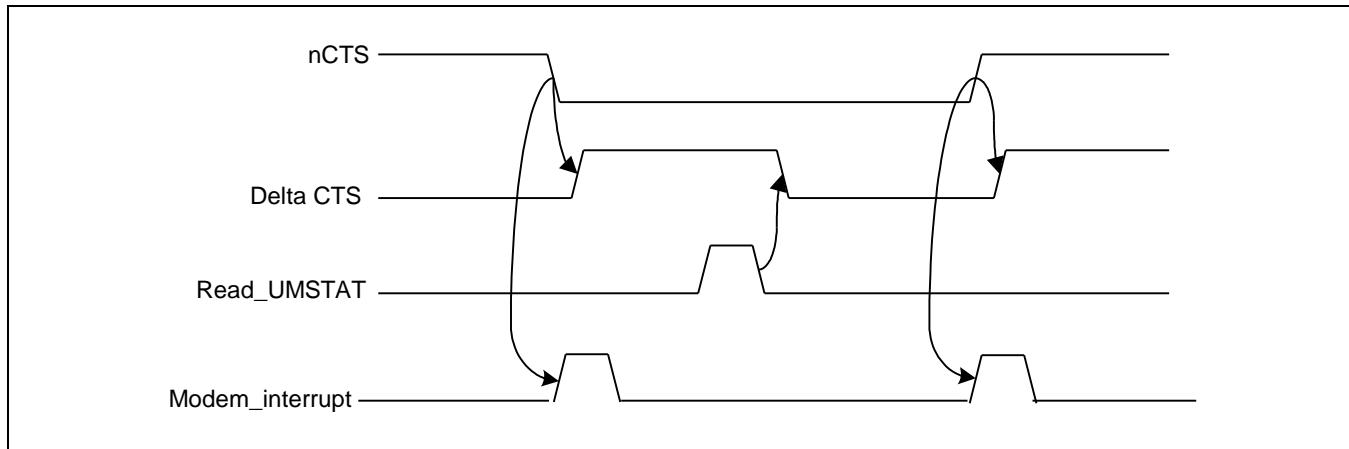


Figure 13-9 nCTS and Delta CTS Timing Diagram

13.6.1.9 UTXHn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|--------------------------|-------------|
| RSVD | [31:8] | — | Reserved | — |
| UTXHn | [7:0] | RWX | Transmits data for UARTn | — |

13.6.1.10 URXHn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|-------------------------|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| URXHn | [7:0] | R | Receives data for UARTn | 0x00 |

NOTE: When an overrun error occurs, CPU must Read URXHn. If not, the next received data makes an overrun error, even though it clears the overrun bit of UERSTATn.

13.6.1.11 UBRDIVn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|--------------------------|-------------|
| RSVD | [31:16] | — | Reserved | 0 |
| UBRDIVn | [15:0] | RW | Baud-rate division value | 0x0000 |

NOTE: When UBRDIV value is set to 0, UFRACVAL value does not affect UART Baud rate.

13.6.1.12 UFRACVALn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:4] | — | Reserved | 0 |
| UFRACVALn | [3:0] | RW | Determines the fractional part of Baud-rate divisor. | 0x0 |

1. UART Baud-Rate Configuration

You can use the value stored in the Baud-rate divisor (UBRDIVn) and divisor fractional value (UFRACVALn) to determine the serial Tx/Rx clock rate (Baud rate) as:

$$\text{DIV_VAL} = \text{UBRDIVn} + \text{UFRACVALn}/16$$

or

$$\text{DIV_VAL} = (\text{SCLK_UART}/(\text{bps} \times 16)) - 1$$

Where, the divisor should be from 1 to (2¹⁶ – 1).

By using UFRACVALn, you can generate the Baud rate more accurately.

For example, if the Baud rate is 115200 bps and SCLK_UART is 40 MHz, UBRDIVn and UFRACVALn are:

$$\begin{aligned}\text{DIV_VAL} &= (40000000/(115200 \times 16)) - 1 \\ &= 21.7 - 1 \\ &= 20.7\end{aligned}$$

UBRDIVn = 20 (integer part of DIV_VAL)

UFRACVALn/16 = 0.7

Therefore, UFRACVALn = 11

2. Baud-Rate Error Tolerance

UART Frame error should be less than 1.87 % (3/160)

$$\text{tUPCLK} = (\text{UBRDIVn} + 1 + \text{UFRACVAL}/16) \times 16 \times 1\text{Frame}/\text{SCLK_UART}$$

tUPCLK = Real UART Clock

tEXTUARTCLK = 1Frame/baud-rate

tEXTUARTCLK = Ideal UART Clock

$$\text{UART error} = (\text{tUPCLK} - \text{tEXTUARTCLK})/\text{tEXTUARTCLK} \times 100 \%$$

* 1Frame = start bit + data bit + parity bit + stop bit.

3. UART Clock and PCLK Relation

There is a constraint on the ratio of clock frequencies for PCLK to UARTCLK.

The frequency of UARTCLK must be no more than 5.5/3 times faster than the frequency of PCLK:

$$\text{FUARTCLK} \leq 5.5/3 \times \text{FPCLK}$$

FUARTCLK = baudrate × 16

This allows sufficient time to Write the received data to the receive FIFO.

13.6.1.13 UINTPn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|-------------------------------|-------------|
| RSVD | [31:4] | — | Reserved | 0 |
| MODEM | [3] | S | Generates modem interrupt. | 0 |
| TXD | [2] | S | Generates transmit interrupt. | 0 |
| ERROR | [1] | S | Generates error interrupt. | 0 |
| RXD | [0] | S | Generates receive interrupt. | 0 |

Interrupt pending contains the information of the generated interrupts.

If one of the 4 bits is logical high ("1"), each UART channel generates interrupt.

NOTE: You must clear this in the interrupt service routine after clearing interrupt pending in Interrupt Controller (INTC). Clear specific bits of UINTP by writing 1's to the bits that you want to clear.

13.6.1.14 UINTSPn (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|-------------------------------|-------------|
| RSVD | [31:4] | — | Reserved | 0 |
| MODEM | [3] | S | Generates modem interrupt. | 0 |
| TXD | [2] | S | Generates transmit interrupt. | 0 |
| ERROR | [1] | S | Generates error interrupt. | 0 |
| RXD | [0] | S | Generates receive interrupt. | 0 |

NOTE: Interrupt Source Pending contains the information of the generated interrupt regardless of the value of Interrupt Mask.

13.6.1.15 UINTM_n (n = 0 to 4)

- Base Address: 0x1380_0000
- Base Address: 0x1381_0000
- Base Address: 0x1382_0000
- Base Address: 0x1383_0000
- Base Address: 0x1384_0000
- Address = Base Address + 0x0038, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|--------|------|---------------------------|-------------|
| RSVD | [31:4] | — | Reserved | 0 |
| MODEM | [3] | RW | Masks modem interrupt. | 0 |
| TXD | [2] | RW | Masks transmit interrupt. | 0 |
| ERROR | [1] | RW | Masks error interrupt. | 0 |
| RXD | [0] | RW | Masks receive interrupt. | 0 |

[Figure 13-10](#) illustrates the block diagram of UINTSP, UINTP, and UINTM.

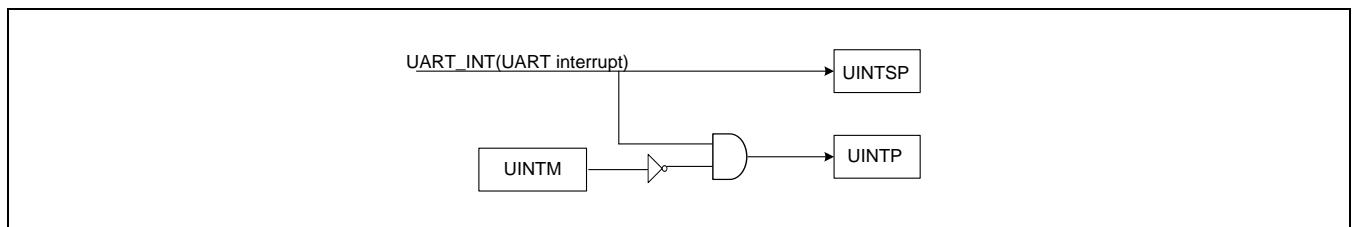


Figure 13-10 Block Diagram of UINTSP, UINTP, and UINTM

Interrupt mask contains the information about masked interrupt sources. When a specific bit is set to 1, UART does not generate interrupt request signal to the Interrupt Controller even though it generates corresponding interrupt.

NOTE: In such cases, the corresponding bit of UINTSP_n is set to 1. When the mask bit is set to 0, CPU services the interrupt requests from the corresponding interrupt source.

14 Inter-Integrated Circuit

14.1 Overview

The Exynos 4412 Reduced Instruction Set Computer (RISC) microprocessor supports four multi-master Inter-Integrated Circuit (I2C) bus serial interfaces. To transmit information between bus masters and peripheral devices, which are connected to the I2C bus, a dedicated Serial Data Line (SDA) and Serial Clock Line (SCL) is used. Both SDA and SCL lines are bi-directional.

In multi-master I2C-bus mode, multiple Exynos 4412 RISC microprocessors either receive or transmit serial data to or from slave devices. The master Exynos 4412 initiates and terminates a data transfer over the I2C bus. The I2C bus in the Exynos4412 uses a standard I2C bus arbitration procedure to realize multi-master and multi-slave transfer.

To control multi-master I2C-bus operations, you must write values to these registers:

- Multi-master I2C-bus control register – I2CCON
- Multi-master I2C-bus control/status register – I2CSTAT
- Multi-master I2C-bus Tx/Rx data shift register – I2CDS
- Multi-master I2C-bus address register – I2CADD

If the I2C-bus is idle, both SDA and SCL lines should be at High level. A High-to-Low transition of SDA initiates a Start condition. A Low-to-High transition of SDA initiates a Stop condition, while SCL remains steady at High level.

The master device always generates Start and Stop conditions. Front 7 bits address value in the data byte is transferred through SDA line after the start condition has been initiated. This address value determines the slave device which the bus master device has selected. The 8th bit determines the direction of the transfer (Read or Write).

Every data byte put on the SDA line should be 8 bits in total. There is no limit either to send or receive bytes during the bus transfer operation. I2C master and slave devices always send the data from the Most Significant Bit (MSB) first, and then acknowledge (ACK) bit immediately follows every byte.

14.2 Features

Features of I2C bus interface are:

- 9 channels multi-master, Slave I2C bus interfaces
(8 channels for general purpose, 1 channel dedicated for High Definition Multimedia Interface (HDMI))
- 7-bit addressing mode
- Serial, 8-bit oriented, and bi-directional data transfer
- Supports up to 100 kbit/s in the Standard mode
- Supports up to 400 kbit/s in the Fast mode.
- Supports master transmit, master receive, slave transmit, and slave receive operation
- Supports interrupt or polling events

14.3 Functional Description

14.3.1 Block Diagram

[Figure 14-1](#) illustrates the block diagram of I2C bus.

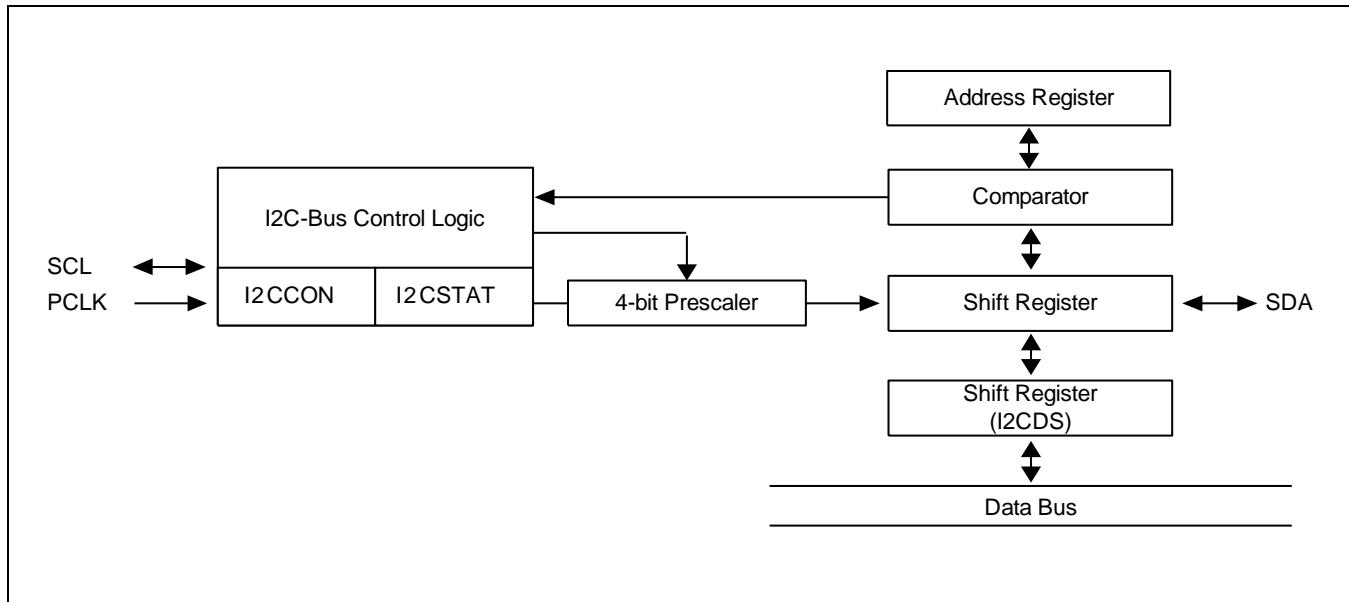


Figure 14-1 I2C-Bus Block Diagram

14.4 I2C-Bus Interface Operation

The four operation modes of the Exynos 4412 I2C-bus interface are:

- Master Transmitter Mode:
- Master Receive Mode
- Slave Transmitter Mode
- Slave Receive Mode

The functional relationships among these operating modes are described in these sections:

- Start and Stop conditions
- Data transfer format
- ACK signal transmission
- Read-Write operation
- Bus arbitration procedures
- Abort conditions
- Configuring IIC-bus

14.4.1 Start and Stop Conditions

When the I2C-bus interface is inactive, it is usually in Slave mode. Alternatively, the interface should be in Slave mode before detecting a Start condition on the SDA line (a Start condition is initiated with a High-to-Low transition of the SDA line, when the clock signal of SCL is High). When controller changes the interface state to master-mode, SDA line initiates data transfer and generates SCL signal.

A Start condition transfers 1-byte serial data through SDA line, and a Stop condition terminates the data transfer. A Stop condition is a Low-to-High transition of the SDA line, while SCL is High. The master generates Start and Stop conditions. I2C bus goes into the busy state when a master or slave device generates a start condition. Alternatively, a Stop condition makes the I2C bus idle state.

When a master initiates a Start condition, it should send a slave address to notify the slave device. 1 byte of address field includes a 7-bit address and 1-bit transfer direction indicator, which shows Write or Read. When bit 8 is 0, it indicates a Write operation (Transmit Operation); when bit 8 is 1, it indicates a request for data Read (Receive Operation).

The master transmits Stop condition to complete the transfer operation. If the master wants to continue the data transmission to the bus, it should generate another Start condition and a slave address. In this manner, there can be various formats of the read-write operation.

[Figure 14-2](#) illustrates the Start and Stop condition.

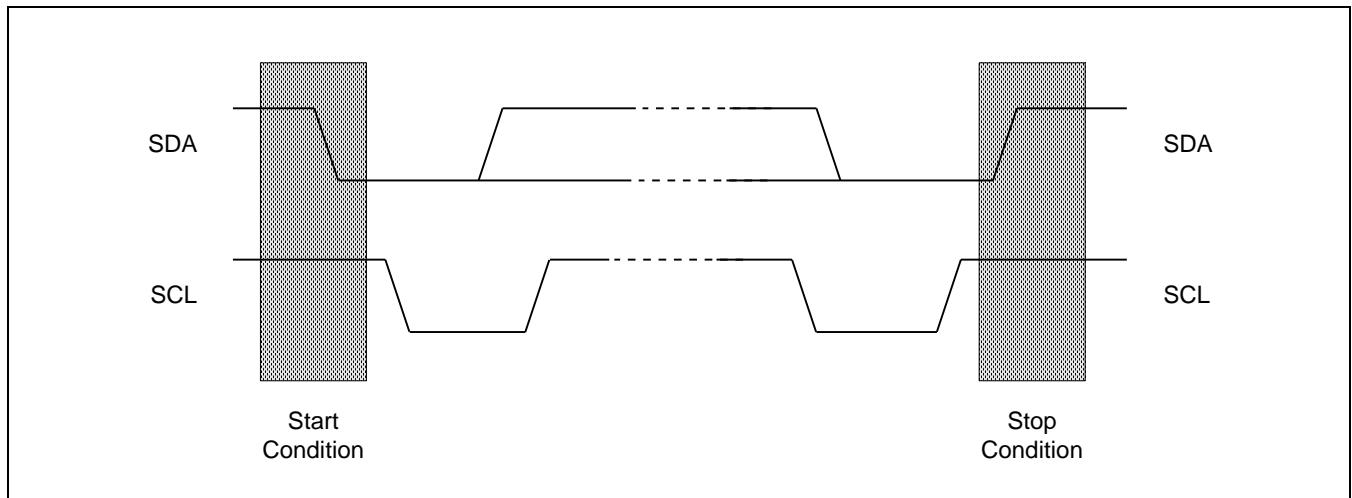


Figure 14-2 Start and Stop Condition

14.4.2 Data Transfer Format

Every byte placed on the SDA line should be 8 bits in length. There is no limit to transmit bytes per transfer. The first byte that follows a Start condition should have the address field. When the I2C-bus is operating in master mode, master transmits the address field. An ACK bit follows each byte. The I2C controller sends first the MSB of the data and address byte to the SDA line.

[Figure 14-3](#) illustrates the I2C-bus interface data format.

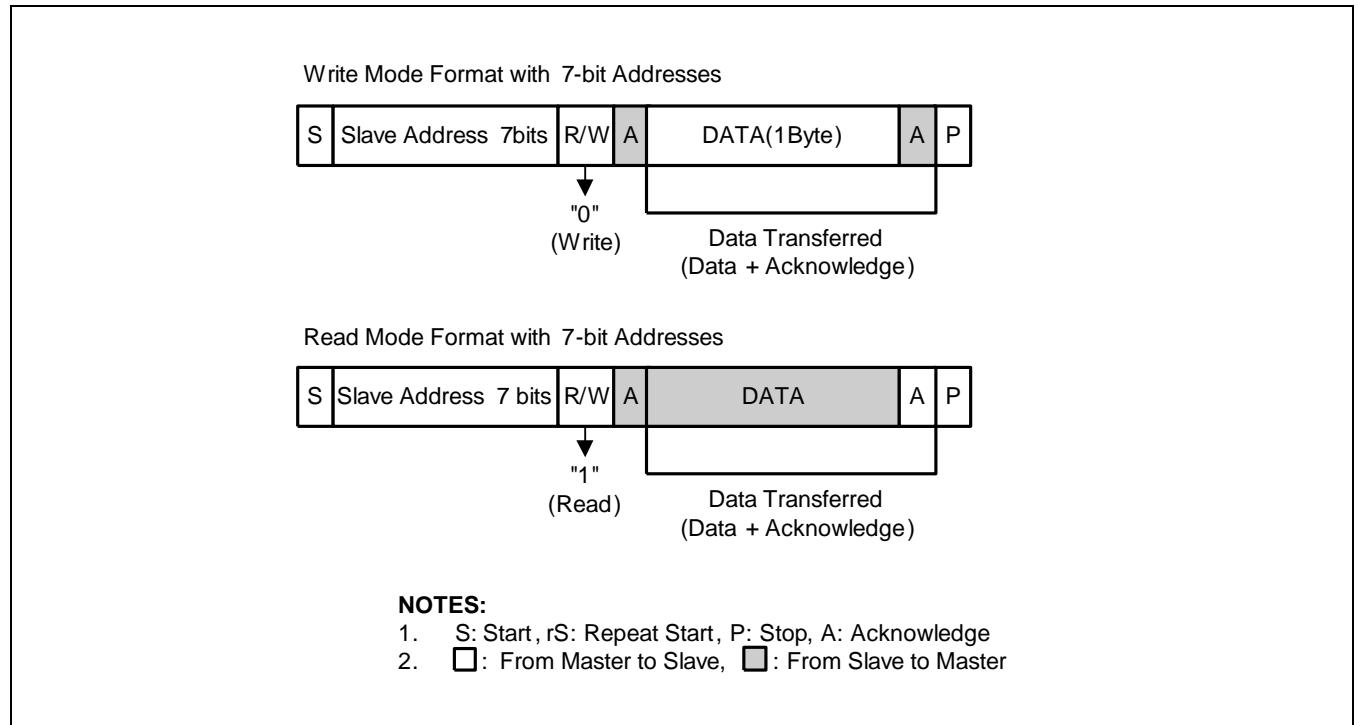


Figure 14-3 I2C-Bus Interface Data Format

[Figure 14-4](#) illustrates the data transfer on the I2C-bus.

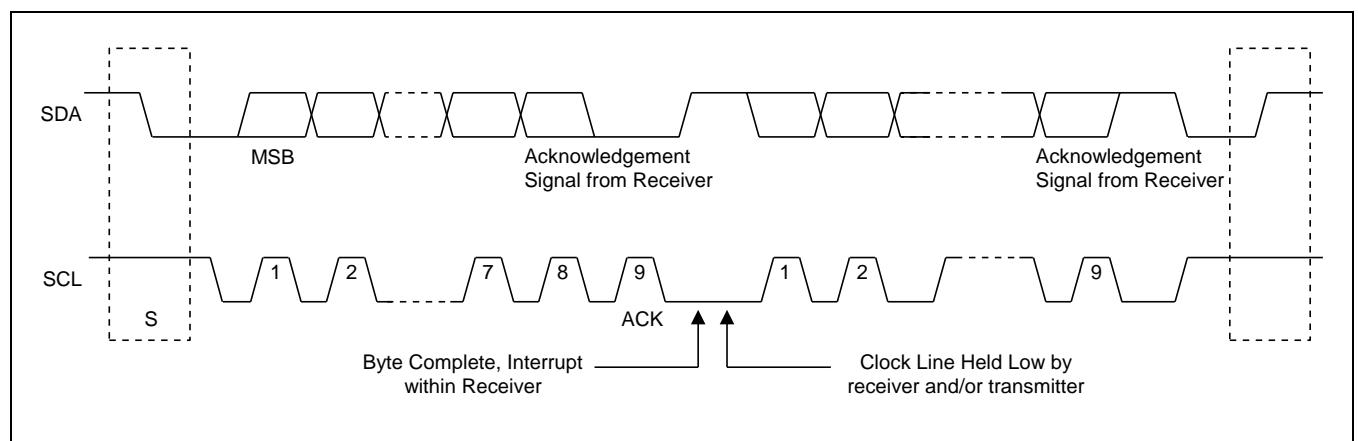


Figure 14-4 Data Transfer on the I2C-Bus

14.4.3 ACK Signal Transmission

To complete a 1-byte transfer operation, the receiver sends an ACK bit to the transmitter. The ACK pulse appears at the ninth clock of the SCL line. The I2C master device generates eight clock cycles to transmit or receive 1 byte data. The master generates clock pulse that is required to transmit the ACK bit.

When the transmitter receives ACK clock pulse, it sets the SDA line to High to release the SDA line. The receiver drives the SDA line Low during the ACK clock pulse to keep the SDA Low. This happens during the High period of the ninth SCL pulse. The software (I2CSTAT) enables or disables ACK bit transmit function. However, the ACK pulse on the ninth clock of SCL should complete the 1-byte data transfer operation.

[Figure 14-5](#) illustrates the acknowledgement on the I2C-bus.

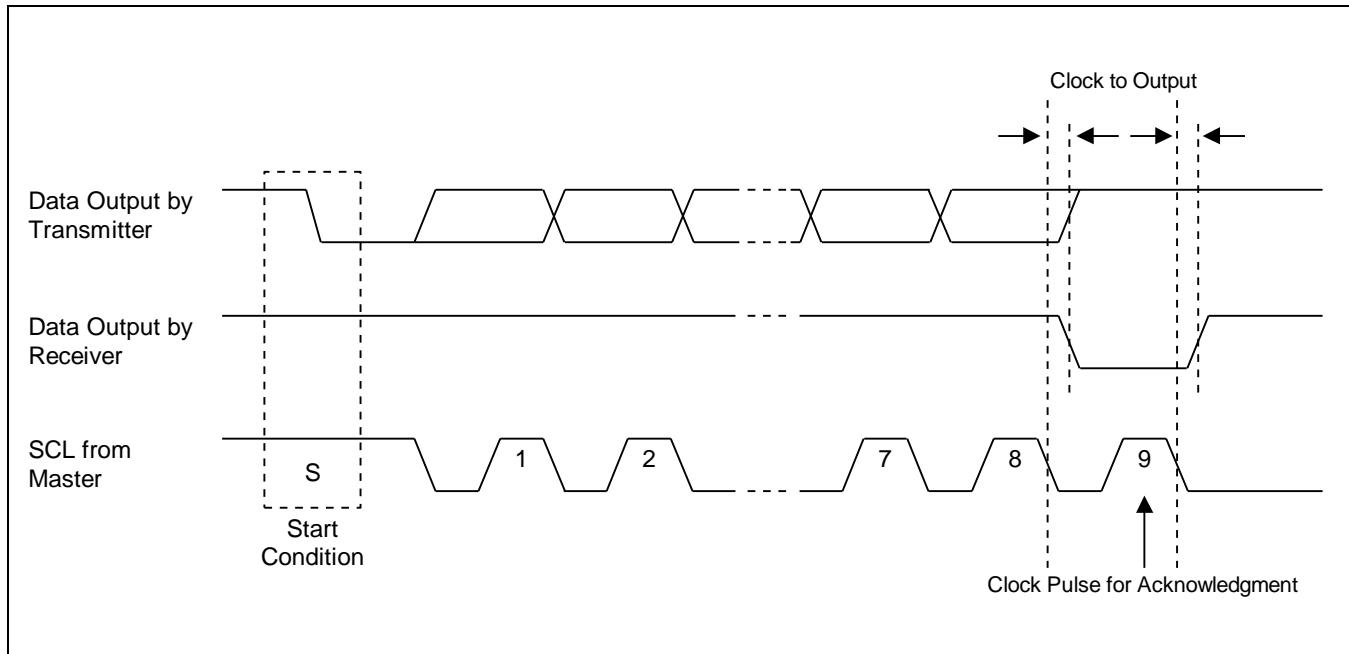


Figure 14-5 Acknowledgement on the I2C-Bus

14.4.4 Read-Write Operation

When the I2C controller transmits data in transmitter mode, the I2C-bus interface waits until I2C-bus Data Shift (I2CDS) register receives the new data. Before you write new data to the register, the SCL line is held Low. The I2C controller releases the SCL line after you write the data. Exynos 4412 holds the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it writes new data to the I2CDS register again.

When the I2C controller receives data in receive mode, the I2C-bus interface waits until I2CDS register is Read. Before you read out the new data, the SCL line is held Low. The I2C controller releases the SCL line after you read the data. Exynos 4412 holds the interrupt to identify the completion of new data reception. After the CPU receives the interrupt request, it reads the data from the I2CDS register.

14.4.5 Bus Arbitration Procedures

Arbitration occurs on the SDA line to prevent the conflict on the bus between two masters. If a master with a SDA High level detects other master with a SDA active Low level, it does not initiate a data transfer. This is because the current level on the bus is not corresponding to initiate a data transfer. The arbitration procedure extends until the SDA line turns High.

When two or more masters assert the SDA line Low simultaneously, each master evaluates whether it has the mastership or not. For the purpose of evaluation, each master detects the address bits. While each master generates the Slave address, it detects the address bit on the SDA line. This is because the SDA line becomes Low instead of High.

Let us assume that one master generates a Low as first address bit, while the other master is maintaining High. In such case, both masters detect Low on the bus. This is because the Low status is superior to the High status in power. When this happens, the Low (as the first bit of address) that generates master, gets the mastership while the High (as the first bit of address) that generates master, withdraws the mastership.

When both masters generate Low as the first bit of address, there is arbitration for the second address bit again. This arbitration continues till the end of last address bit.

14.4.6 Abort Conditions

When a Slave receiver cannot acknowledge the confirmation of the slave address, it holds the level of the SDA line High. In this case, the master generates a Stop condition and cancels the transfer.

When a master receiver is involved in the aborted transfer, it signals the end of Slave transmit operation by canceling the generation of an ACK. This happens after the Master receives the last data byte from the Slave. The Slave transmitter releases the SDA to enable a master to generate a Stop condition.

14.4.7 Configuring I2C-Bus

To control the frequency of SCL, you should write the 4-bit prescaler value in the I2CCON register. The I2C-bus interface address is stored in the I2C-bus address (I2CADD) register. By default, the I2C-bus interface address has an unknown value.

14.4.8 Flowcharts of Operations in Each Mode

Before you execute any I2C Tx/Rx operations:

1. If required, Write own Slave address on I2CADD register.
2. Set I2CCON register:
 - a) Enable interrupt.
 - b) Define SCL period.
3. Set I2CSTAT to enable Serial Output.

[Figure 14-6](#) illustrates the operations for Master/Transmitter mode.

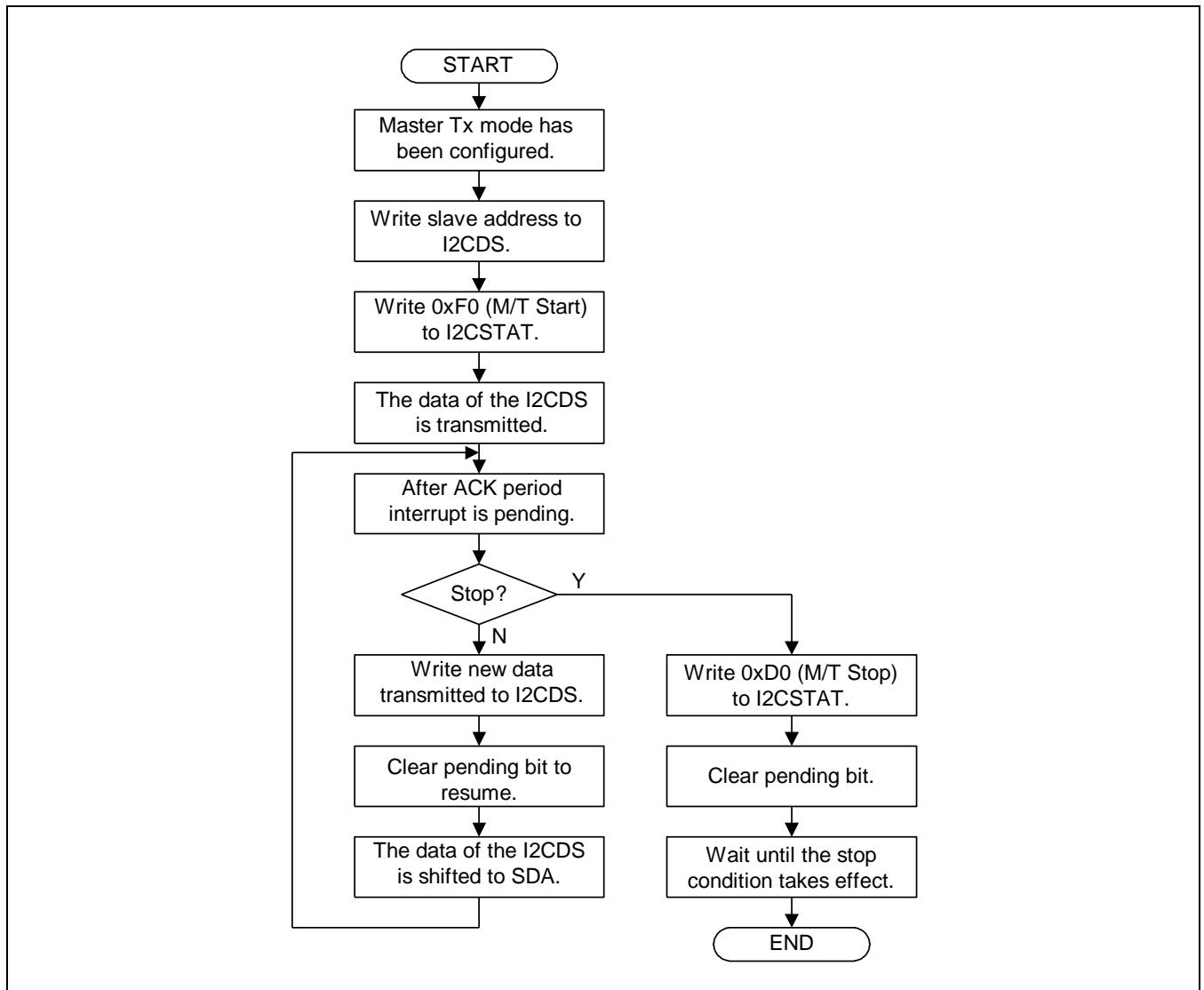


Figure 14-6 Operations for Master/Transmitter Mode

[Figure 14-7](#) illustrates the operations for Master/Receiver Mode.

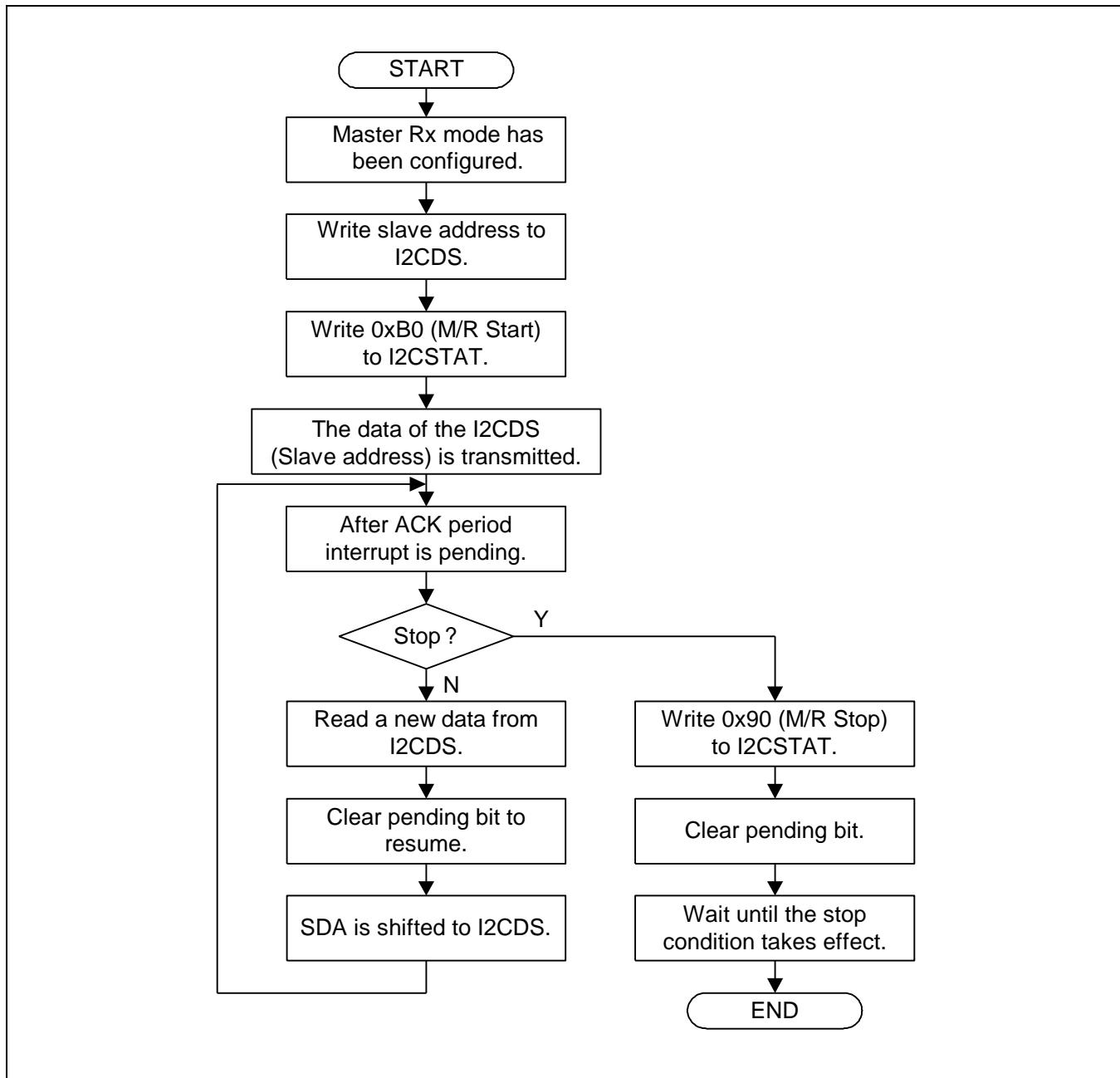


Figure 14-7 Operations for Master/Receiver Mode

[Figure 14-8](#) illustrates the operations for Slave/Transmitter mode.

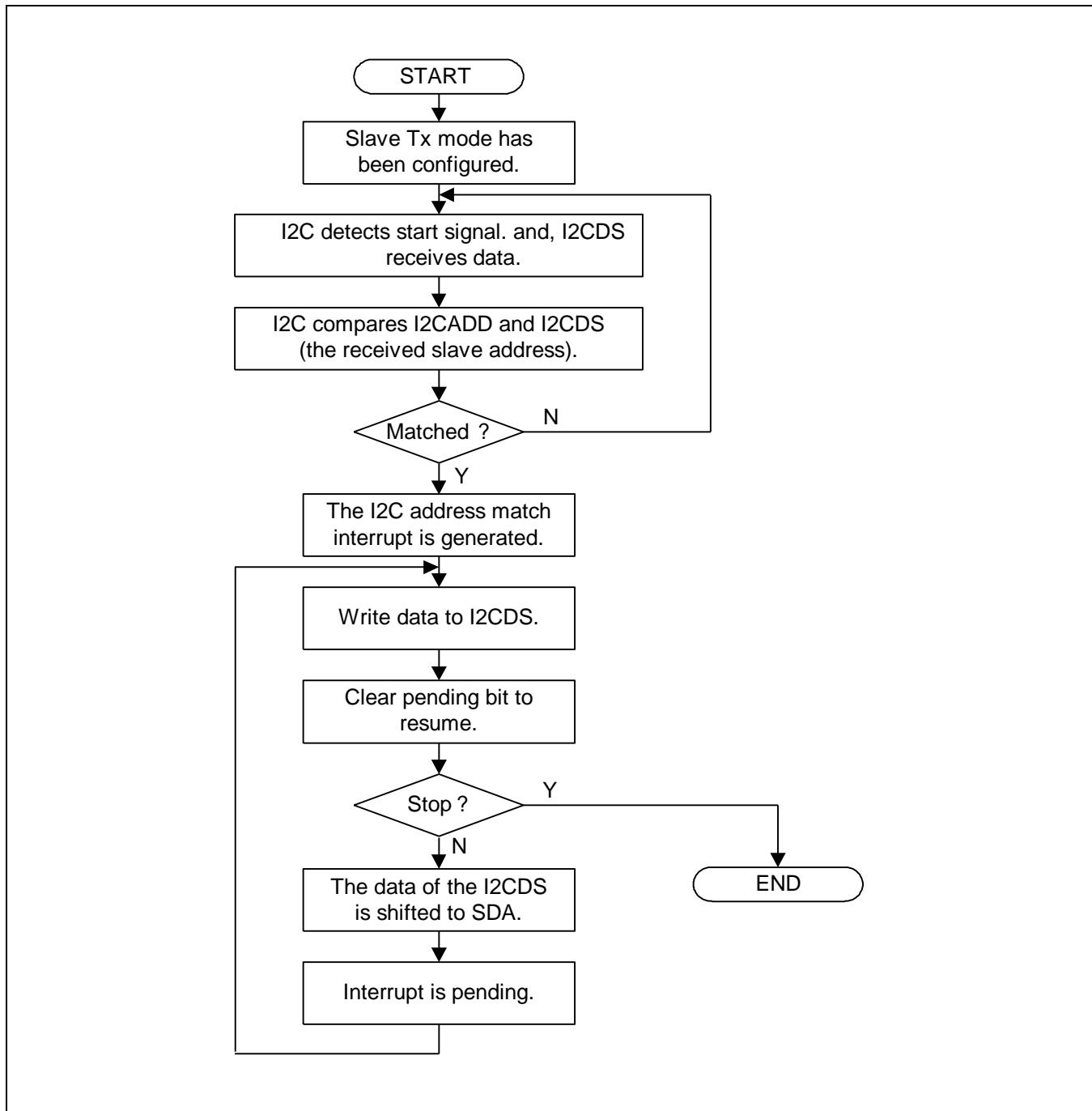


Figure 14-8 Operations for Slave/Transmitter Mode

[Figure 14-9](#) illustrates the operations for Slave/Receiver Mode.

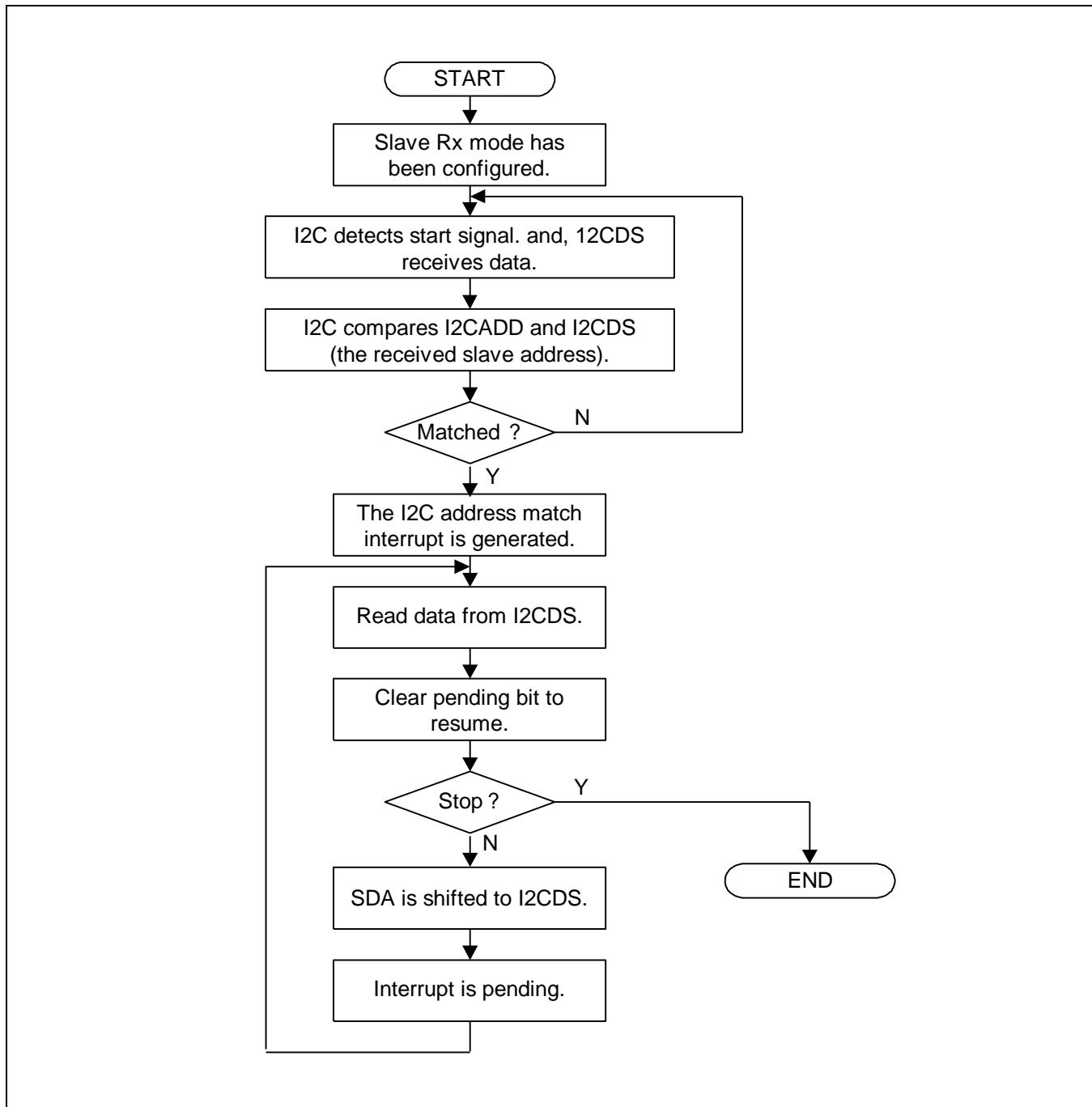


Figure 14-9 Operations for Slave/Receiver Mode

14.5 I/O Description

| Signal | I/O | Description | Pad | Type |
|----------|--------------|--------------------------------------|------------|-------|
| I2C0_SCL | Input/Output | I2C-bus interface0 serial clock line | XI2C0SCL | muxed |
| I2C0_SDA | Input/Output | I2C-bus interface0 serial data line | XI2C0SDA | muxed |
| I2C1_SCL | Input/Output | I2C-bus interface1 serial clock line | XI2C1SCL | muxed |
| I2C1_SDA | Input/Output | I2C-bus interface1 serial data line | XI2C1SDA | muxed |
| I2C2_SCL | Input/Output | I2C-bus interface2 serial clock line | XuRTSn_1 | muxed |
| I2C2_SDA | Input/Output | I2C-bus interface2 serial data line | XuCTSn_1 | muxed |
| I2C3_SCL | Input/Output | I2C-bus interface3 serial clock line | XuRTSn_2 | muxed |
| I2C3_SDA | Input/Output | I2C-bus interface3 serial data line | XuCTSn_2 | muxed |
| I2C4_SCL | Input/Output | I2C-bus interface4 serial clock line | XspiMOSI_0 | muxed |
| I2C4_SDA | Input/Output | I2C-bus interface4 serial data line | XspiMISO_0 | muxed |
| I2C5_SCL | Input/Output | I2C-bus interface5 serial clock line | XspiMOSI_1 | muxed |
| I2C5_SDA | Input/Output | I2C-bus interface5 serial data line | XspiMISO_1 | muxed |
| I2C6_SCL | Input/Output | I2C-bus interface6 serial clock line | Xi2s2SDO | muxed |
| I2C6_SDA | Input/Output | I2C-bus interface6 serial data line | Xi2s2SDI | muxed |
| I2C7_SCL | Input/Output | I2C-bus interface7 serial clock line | XpwmTOUT_3 | muxed |
| I2C7_SDA | Input/Output | I2C-bus interface7 serial data line | XpwmTOUT_2 | muxed |

NOTE: The I2C bus interface for the HDMI has no external I/O.

14.6 Register Description

14.6.1 Register Map Summary

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000

| Register | Offset | Description | Reset Value |
|----------|--------|---|-------------|
| I2CCONn | 0x0000 | Specifies the I2C-bus interface0 control register | 0x0X |
| I2CSTATn | 0x0004 | Specifies the I2C-bus interface0 control/status register | 0x00 |
| I2CADDn | 0x0008 | Specifies the I2C-bus interface0 address register | 0XX |
| I2CDSn | 0x000C | Specifies the I2C-bus interface0 transmit/receive data shift register | 0XX |
| I2CLCN | 0x0010 | Specifies the I2C-bus interface0 multi-master line control register | 0x00 |

14.6.1.1 I2CCONn (n = 0 to 7)

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000
- Address = Base Address + 0x0000, Reset Value = 0x0X

| Name | Bit | Type | Description | Reset Value |
|---|--------|------|---|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| Acknowledge generation ⁽¹⁾ | [7] | RW | I2C-bus acknowledge enable bit 0 = Disables 1 = Enables In Tx mode, the I2CSDA is idle in the ACK time. In Rx mode, the I2CSDA is low in the ACK time. | 0 |
| Tx clock source selection | [6] | RW | Source clock of I2C-bus transmit clock prescaler selection bit 0 = I2CCLK = fPCLK/16 1 = I2CCLK = fPCLK/512 | 0 |
| Tx/Rx Interrupt ⁽⁵⁾ | [5] | RW | I2C-bus Tx/Rx interrupt enable/ disable bit 0 = Disables 1 = Enables | 0 |
| Interrupt pending flag ⁽²⁾⁻⁽³⁾ | [4] | S | I2C-bus Tx/Rx interrupt pending flag You cannot write this bit to 1. If you read this bit as 1, the I2CSCL is tied to Low and the I2C is stopped. To resume the operation, write this bit as 0. 0 = 1) No interrupt is pending (If Read). 2) Clears pending condition and resumes the operation (If Write). 1 = 1) Interrupt is pending (If Read) 2) N/A (If Write) | 0 |
| Transmit clock value ⁽⁴⁾ | [3:0] | RW | I2C-bus transmit clock prescaler 4-bit prescaler value determines the I2C-bus transmit clock frequency according to the formula given here: $\text{Tx clock} = \text{I2CCLK}/(\text{I2CCON}[3:0] + 1).$ | — |

NOTE:

1. While interfacing with EEPROM, the ACK generation is disabled before Reading the last data to generate the STOP condition in Rx mode.

2. An I2C-bus interrupt occurs when:
 - (a) 1-byte Transmit or Receive operation is complete. Alternatively, the ACK period is finished.
 - (b) A general call or a Slave address match occurs.
 - (c) Bus arbitration fails.
3. To adjust the setup time of SDA before SCL rising edge, ensure to Write I2CDS before clearing the I2C interrupt pending bit.
4. I2CCON[6] determines I2CCLK. Tx clock can vary by SCL transition time.
When I2CCON[6] = 0, I2CCON[3:0] = 0x0 or 0x1 is not available.
5. When I2CCON[5] = 0, I2CCON[4] does not operate correctly.
Therefore, set I2CCON[5] = 1 even if you do not use the I2C interrupt.

14.6.1.2 I2CSTATn (n = 0 to 7)

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000
- Address = Base Address + 0x0004, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|---|--------|------|--|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| Mode selection | [7:6] | RWX | I2C-bus Master/Slave Tx/Rx mode select bits 00 = Slave receive mode 01 = Slave transmit mode 10 = Master receive mode 11 = Master transmit mode | 00 |
| Busy signal status/START STOP condition | [5] | S | I2C-bus busy signal status bit 0 = (Read) Not busy (If Read) (write) STOP signal generation 1 = (Read) Busy (If Read) (write) START signal generation. Transfers the data in I2CDS automatically just after the start signal. | 0 |
| Serial output | [4] | S | I2C-bus data output enable/ disable bit 0 = Disables Rx/Tx 1 = Enables Rx/Tx | 0 |
| Arbitration status flag | [3] | R | I2C-bus arbitration procedure status flag bit 0 = Bus arbitration successful 1 = Bus arbitration fails during serial I/O | 0 |
| Address-as-slave status flag | [2] | R | I2C-bus address-as-slave status flag bit 0 = Clears when it detects START/STOP condition 1 = Receives slave address that matches the address value in the I2CADD | 0 |
| Address zero status flag | [1] | R | I2C-bus address zero status flag bit 0 = Clears when it detects START/ STOP condition 1 = Received slave address is 0000000b | 0 |
| Last-received bit status flag | [0] | R | I2C-bus last-received bit status flag bit 0 = Last-received bit is set to 0 (receives ACK). 1 = Last-received bit is set to 1 (does not receive ACK). | 0 |

14.6.1.3 I2CADD_n (n = 0 to 7)

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000
- Address = Base Address + 0x0008, Reset Value = 0xXX

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|--|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| Slave address | [7:0] | RWX | 7-bit slave address, latched from the I2C-bus. When serial output enable = 0 in the I2CSTAT, I2CADD is write-enabled. The I2CADD value is Read any time, regardless of the current serial output enable bit (I2CSTAT) setting. Slave address: [7:1] Not mapped: [0] | — |

14.6.1.4 I2CDS_n (n = 0 to 7)

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000
- Address = Base Address + 0x000C, Reset Value = 0XX

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|--|-------------|
| RSVD | [31:8] | — | Reserved | 0 |
| Data shift | [7:0] | RWX | 8-bit data shift register for I2C-bus Tx/Rx operation. When serial output enable = 1 in the I2CSTAT, I2CDS is write-enabled. The I2CDS value is Read any time, regardless of the current serial output enable bit (I2CSTAT) setting. | — |

14.6.1.5 I2CLCn (n = 0 to 7)

- Base Address: 0x1386_0000
- Base Address: 0x1387_0000
- Base Address: 0x1388_0000
- Base Address: 0x1389_0000
- Base Address: 0x138A_0000
- Base Address: 0x138B_0000
- Base Address: 0x138C_0000
- Base Address: 0x138D_0000
- Base Address: 0x138E_0000
- Address = Base Address + 0x0010, Reset Value = 0x00

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:27] | – | Reserved | 0 |
| Filter enable | [2] | RW | I2C-bus filter enable bit When SDA port is operating as input, set this bit to High. This filter prevents error caused by glitch between two PCLK clocks. 0 = Disables Filter 1 = Enables Filter | 0 |
| SDA output delay | [1:0] | RW | I2C-bus SDA line delay length selection bits The I2C controller delays the SDA line by following clock cycle: 00 = 0 clock 01 = 5 clocks 10 = 10 clocks 11 = 15 clocks | 00 |

15 Serial Peripheral Interface

15.1 Overview

Serial Peripheral Interface (SPI) in Exynos 4412 transfers serial data by using various peripherals. SPI includes two 8, 16, and 32-bit shift registers to transmit and receive data. During an SPI transfer, it simultaneously transmits (shifts out serially) and receives (shifts in serially) data. SPI supports the protocols for National Semiconductor Microwire and Motorola Serial Peripheral Interface.

15.2 Features

The features of SPI include:

- Full duplex
- 8/16/32-bit shift register for Tx/Rx
- Supports 8-bit/16-bit/32-bit bus interface
- Supports the Motorola SPI protocol and National Semiconductor Microwire
- Two independent 32-bits wide transmit and receive FIFOs: depth 64 in port 0 and depth 16 in port 1 and 2
- Master-mode and Slave-mode
- Receive-without-transmit operation
- Tx/Rx maximum frequency at up to 50 MHz

15.2.1 Operation of SPI

The SPI transfers 1-bit serial data between Exynos 4412 and external device. The SPI in Exynos 4412 supports the CPU or DMA to transmit or receive FIFOs separately and to transfer data in both directions simultaneously. SPI has two channels, namely, Tx channel and Rx channel. Tx channel has the path from Tx FIFO to external device. Rx channel has the path from external device to Rx FIFO.

CPU (or DMA) must write data on the register SPI_TX_DATA, to write data in FIFO. Data on the register are automatically moved to Tx FIFOs. To read data from Rx FIFOs, CPU (or DMA) must access the register SPI_RX_DATA and data are automatically sent to the SPI_RX_DATA register.

CMU registers can control SPI operating frequency. Refer to "CMU" chapter for more information.

15.2.1.1 Operation Mode

SPI has two modes, namely, master and slave mode. In master mode, SPICLK is generated and transmitted to external device. XspiCS#, which is the signal to select slave, indicates that the data is valid when XspiCS# is set to low level. XspiCS# must be set low before packets are transmitted or received.

15.2.1.2 FIFO Access

The SPI supports CPU access and DMA access to FIFOs. Data size of CPU access and DMA access to FIFOs are selected either from 8-bit, 16-bit, or 32-bit data. When it selects 8-bit data size, then valid bits are from 0 to 7-bit. User can define the trigger threshold to raise interrupt to CPU. The trigger level of each FIFO in port 0 is set by 4 bytes step from 0 to 252 bytes, and that of each FIFO in port 1 is set by 1 byte step from 0 to 63 bytes. TxDMAOn or RxDMAOn bit of SPI_MODE_CFG register must be set to use DMA access. DMA access supports only single transfer and 4-burst transfer. In Tx FIFO, DMA request signal is high until Tx FIFO is full. In Rx FIFO, DMA request signal is high if FIFO is not empty.

15.2.1.3 Trailing Bytes in the Rx FIFO

When the number of samples in Rx FIFO is less than the threshold value in INT mode or DMA 4-burst mode and it does not receive any additional data, then the remaining bytes are called trailing bytes. To remove these bytes in Rx FIFO, it uses internal timer and interrupt signal. The value of internal timer is set up to 1024 clocks based on APB BUS clock. When timer value is zero, interrupt signal occurs and CPU can remove trailing bytes in FIFO.

15.2.1.4 Packet Number Control

SPI controls the number of packets to be received in master mode. Set SFR (PACKET_CNT_REG) to receive any number of packets. SPI stops generating SPICLK if the number of packets is similar to PACKET_CNT_REG. The size of one packet depends on channel width. (One packet is one byte when you configure channel width as byte, and one packet is four bytes when you configure channel width as word.) It is mandatory to follow software or hardware reset before reloading this function. (Software reset can clear all registers except special function registers, but hardware reset clears all registers.)

15.2.1.5 Chip Select Control

Chip select XspiCS# is active low signal. In other words, a chip is selected when XspiCS# input is 0.

You can control XspiCS# automatically or manually. No need to change. When you use manual control mode, you should clear AUTO_N_MANUAL (default value is 0). NSSOUT bit controls XspiCS# level.

When you use auto control mode, AUTO_N_MANUAL must be set as 1. XspiCS toggled between packet and packet automatically. NCS_TIME_COUNT controls inactive period of XspiCS. NSSOUT is not available at this time.

15.2.1.6 High Speed Operation as Slave

Exynos 4412 SPI supports Tx/Rx operations up to 50 MHz, but there is a limitation. When Exynos 4412 SPI works as a slave, it consumes large delay more than 15 ns in worst operating condition. Such a large delay can cause setup violation at SPI master device. To overcome the problem, Exynos 4412 SPI provides fast slave Tx mode by setting 1 to HIGH_SPEED bit of CH_CFG register. In that mode, it reduces MISO output delay by half cycle, so that the SPI master device has more setup margin.

However, you can use the fast slave Tx mode only when CPHA = 0.

15.2.1.7 Feedback Clock Selection

Under SPI protocol specification, SPI master should capture the input data launched by slave (MISO) with its internal SPICLK. When SPI runs at high operating frequency such as 50 MHz, it is difficult to capture the MISO input because the required arrival time of MISO is half cycle period in Exynos 4412. It is shorter than the arrival time of MISO that consists of SPICLK output delay of SPI master, MISO output delay of SPI slave, and MISO input delay of SPI master. To overcome the problem, Exynos 4412 SPI provides three feedback clocks that are phase-delayed clock of internal SPICLK.

A selection of feedback clock depends on MISO output delay of SPI slave. To capture MISO data correctly, it selects the feedback clock that satisfies the following constraint:

$$t_{SPIMIS}(s) < t_{period}/2 - t_{SPISOD}$$

- * $t_{SPIMIS}(s)$: MISO input setup time of SPI master on a given feedback clock selection "s"
- * t_{SPISOD} : MISO output delay of SPI slave
- * t_{period} : SPICLK cycle period

If multiple feedback clocks meet the constraint, then it should select the feedback clock with smallest phase delay. Because of a feedback clock with large phase delay, it may capture data of next cycle.

For example of Exynos 4412, SPI CH1 with master configuration of 50 MHz operating frequency, 1.8 V external voltage and 15 pF load, if it assumes MISO output delay of SPI slave as 11 ns ($t_{SPIMIS}(s) < 10 \text{ ns} - 11 \text{ ns} = -1 \text{ ns}$), then it should use 270 degree phase-delayed feedback clock.

If the operating clock frequency is 33 MHz and other conditions are similar to the previous example, it is better to use 180 degree phase-delayed feedback clock ($t_{SPIMIS}(s) < 15 \text{ ns} - 11 \text{ ns} = 4 \text{ ns}$).

15.2.1.8 SPI Transfer Format

The Exynos 4412 supports four different formats for data transfer.

[Figure 15-1](#) illustrates four waveforms for SPICLK.

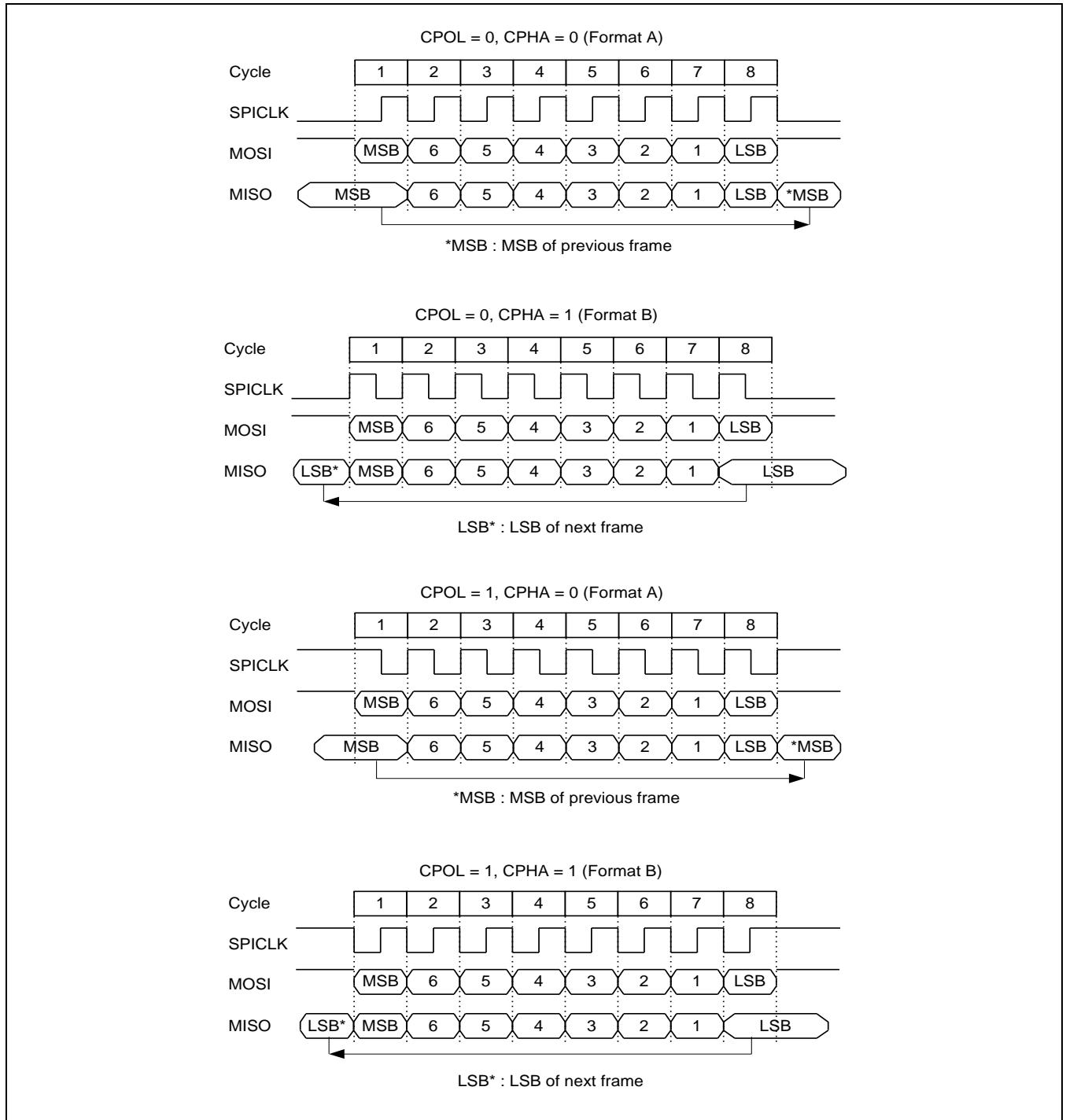


Figure 15-1 SPI Transfer Format

15.3 SPI Input Clock Description

[Figure 15-2](#) illustrates the input clock diagram for SPI.

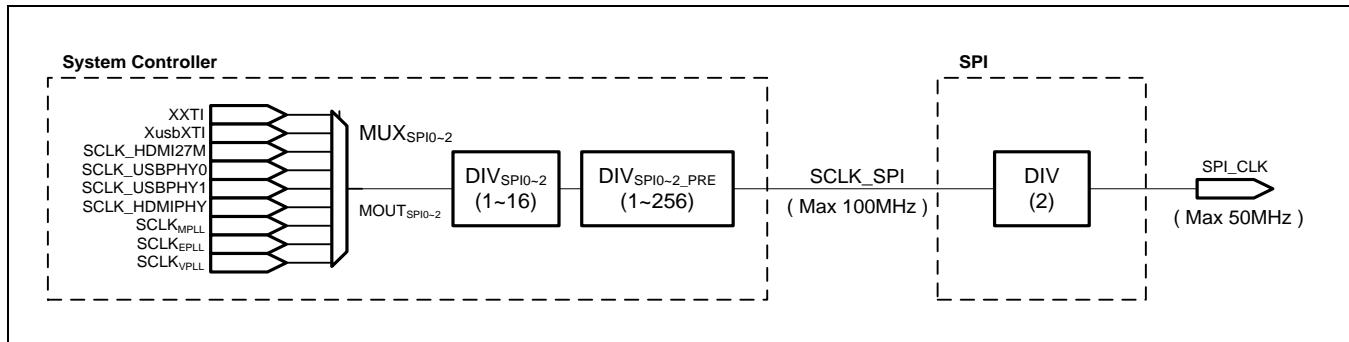


Figure 15-2 Input Clock Diagram for SPI

Exynos 4412 provides SPI with a variety of clocks. As illustrated in the Figure 28-8, the SPI uses SCLK_SPI clock, which is from clock controller. You can also select SCLK_SPI from various clock sources. To select SCLK_SPI, refer to Chapter 7 Clock Controller for more information.

NOTE: SPI has an internal 2x clock divider. You should configure SCLK_SPI to have a double of the SPI operating clock frequency.

15.4 IO Description

The IO description table lists the external signals between the SPI and external device. The unused SPI ports are used as General Purpose I/O ports. Refer to "General Purpose I/O" chapter for more information.

| Signal | I/O | Description | Pad | Type |
|--|--------|--|--------------------------------------|-------|
| SPI_0_CLK SPI_1_CLK SPI_2_CLK | In/Out | XspiCLK is the serial clock used to control time of data transfer. Out: when used as master In: when used as slave | XspiCLK_0 XspiCLK_1 Xi2s2CDCLK | muxed |
| SPI_0_nSS SPI_1_nSS SPI_2_nSS | In/Out | Slave selection signal. All data Tx/Rx sequences are executed if XspiCS is low. Out: when used as master In: when used as slave | XspiCSn_0 XspiCSn_1 Xi2s2LRCK | muxed |
| SPI_0_MISO SPI_1_MISO SPI_2_MISO | In/Out | This port is the input port in Master mode. You can use input mode to get data from slave output port. It transmits data to master through this port in slave mode. Out: when used as slave In: when used as master | XspiMISO_0 XspiMISO_1 Xi2s2SDI | muxed |
| SPI_0_MOSI SPI_1_MOSI SPI_2_MOSI | In/Out | This port is the output port in Master mode. It uses this port to transfer data from master output port. It receives data from master through this port in slave mode. Out: when used as master In: when used as slave | XspiMOSI_0 XspiMOSI_1 Xi2s2SDO | muxed |

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

15.5 Register Description

15.5.1 Register Map Summary

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000

| Register | Offset | Description | Reset Value |
|------------------|--------|------------------------------------|-------------|
| CH_CFGn | 0x0000 | Specifies SPI configuration | 0x0 |
| MODE_CFGn | 0x0008 | Specifies FIFO control | 0x0 |
| CS_REGn | 0x000C | Specifies slave selection control | 0x1 |
| SPI_INT_ENn | 0x0010 | Specifies interrupt enable | 0x0 |
| SPI_STATUSn | 0x0014 | Specifies SPI status | 0x0 |
| SPI_TX_DATAn | 0x0018 | Specifies Tx data | 0x0 |
| SPI_RX_DATAn | 0x001C | Specifies Rx data | 0x0 |
| PACKET_CNT_REGn | 0x0020 | Specifies packet count | 0x0 |
| PENDING_CLR_REGn | 0x0024 | Specifies interrupt pending clear | 0x0 |
| SWAP_CFGn | 0x0028 | Specifies swap configuration | 0x0 |
| FB_CLK_SELn | 0x002C | Specifies feedback clock selection | 0x0 |

Setting Sequence of Special Function Register

Steps to set Special Function Register (nCS manual mode) are:

1. Set Transfer Type. (CPOL and CPHA set)
2. Set Feedback Clock Selection register.
3. Set SPI MODE_CFG register.
4. Set SPI INT_EN register.
5. Set PACKET_CNT_REG register if necessary.
6. Set Tx or Rx Channel on.
7. Set nSSout low to start Tx or Rx operation:
 - a. Set nSSout Bit to low, then start Tx data writing.
 - b. When auto chip selection bit is set, nSSout is controlled automatically.

15.5.1.1 CH_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0000, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | — |
| HIGH_SPEED_EN | [6] | RW | Slave Tx output time control bit If this bit is enabled, slave Tx output time is reduced as much as half period of SPICLKout period. This bit is valid only in CPHA 0. 0 = Disables 1 = Enables | 0 |
| SW_RST | [5] | RW | Software Reset The following registers and bits are cleared by this bit. Rx/Tx FIFO data, SPI_STATUS register will be reset once in the initial time. And after that, if we want to reset the register again, we have to use SW_RST bit manually. 0 = Inactive 1 = Active | 0 |
| SLAVE | [4] | RW | Whether SPI Port is Master or Slave 0 = Master 1 = Slave | 0 |
| CPOL | [3] | RW | Determines whether active high or active low clock 0 = Active high 1 = Active low | 0 |
| CPHA | [2] | RW | Select one of the two fundamentally different transfer format 0 = Format A 1 = Format B | 0 |
| RX_CH_ON | [1] | RW | SPI Rx Channel On 0 = Channel off 1 = Channel on | 0 |
| TX_CH_ON | [0] | RW | SPI Tx Channel On 0 = Channel off 1 = Channel on | 0 |

NOTE: SPI controller should reset when:

1. Reconfiguration of SPI registers is done.
2. Error interrupt has occurred.

15.5.1.2 MODE_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0008, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [31] | — | Reserved | — |
| CH_WIDTH | [30:29] | RW | 00 = Byte 01 = Halfword 10 = Word 11 = Reserved | 0 |
| TRAILING_CNT | [28:19] | RW | Count value from writing the last data in Rx FIFO to flush trailing bytes in FIFO | 0 |
| BUS_WIDTH | [18:17] | RW | 00 = Byte 01 = Halfword 10 = Word 11 = Reserved | 0 |
| RX_RDY_LVL | [16:11] | RW | Rx FIFO trigger level in INT mode. Port 0: Trigger level (bytes) = 4 × N Port 1, 2: Trigger level (bytes) = N (N = value of RX_RDY_LVL field) | 0 |
| TX_RDY_LVL | [10:5] | RW | Tx FIFO trigger level in INT mode. Port 0: Trigger level (bytes) = 4 × N Port 1, 2: Trigger level (bytes) = N (N = value of TX_RDY_LVL field) | 0 |
| RSVD | [4:3] | — | Reserved | — |
| RX_DMA_SW | [2] | RW | Rx DMA mode enable/disable 0 = Disables DMA Mode 1 = Enables DMA Mode | 0 |
| TX_DMA_SW | [1] | RW | Tx DMA mode enable/disable 0 = Disables DMA Mode 1 = Enables DMA Mode | 0 |
| DMA_TYPE | [0] | RW | DMA transfer type, single or 4 bursts. 0 = Single 1 = 4 burst DMA transfer size must be set as the same size in SPI DMA. | 0 |

NOTE:

1. CH_WIDTH is shift-register width.
2. BUS_WIDTH is SPI FIFO width, transfer data size should be aligned with BUS_WIDTH.
For example, Tx/Rx data size must be aligned with 4 bytes if BUS_WIDTH is word.
3. CH_WIDTH must be smaller than BUS_WIDTH or similar to BUS_WIDTH.

15.5.1.3 CS_REGn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x000C, Reset Value = 0x1

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|---|-------------|
| RSVD | [31:10] | – | Reserved | – |
| NCS_TIME_COUNT | [9:4] | RW | NSSOUT inactive time = ((nCS_time_count + 3)/2) × SPICLKout | 0 |
| RSVD | [3:2] | – | Reserved | – |
| AUTO_N_MANUAL | [1] | RW | Chip select toggle manual or auto selection 0 = Manual 1 = Auto | 0 |
| NSSOUT | [0] | RW | Slave selection signal (manual only) 0 = Active 1 = Inactive | 1 |

When AUTO_N_MANUAL is set, then SPI controller controls NSSOUT and does not perform data transfer continuously.

Unit data size depends on CH_WIDTH.

[Figure 15-3](#) illustrates auto chip select mode waveform.

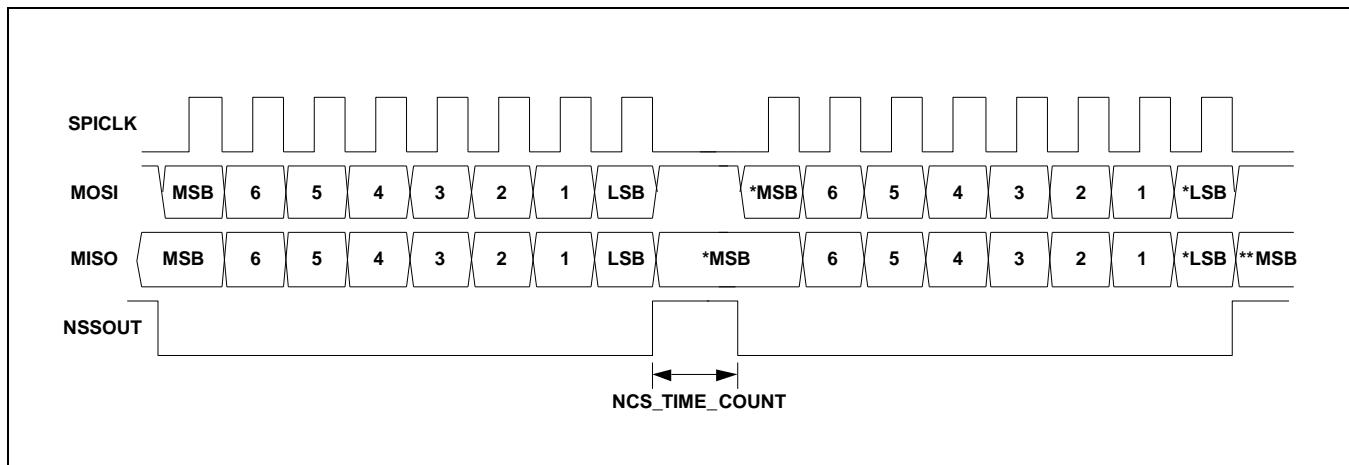


Figure 15-3 Auto Chip Select Mode Waveform (CPOL = 0, CPHA = 0, CH_WIDTH = Byte)

15.5.1.4 SPI_INT_ENn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0010, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|--------------------|--------|------|---|-------------|
| RSVD | [31:7] | — | Reserved | — |
| INT_EN_TRAILING | [6] | RW | Interrupt Enable for trailing count to be 0 0 = Disables 1 = Enables | 0 |
| INT_EN_RX_OVERRUN | [5] | RW | Interrupt Enable for RxOverrun 0 = Disables 1 = Enables | 0 |
| INT_EN_RX_UNDERRUN | [4] | RW | Interrupt Enable for RxUnderrun 0 = Disables 1 = Enables | 0 |
| INT_EN_TX_OVERRUN | [3] | RW | Interrupt Enable for TxOverrun 0 = Disables 1 = Enables | 0 |
| INT_EN_TX_UNDERRUN | [2] | RW | Interrupt Enable for TxUnderrun. In slave mode, this bit must be clear first after turning on slave Tx path. 0 = Disables 1 = Enables | 0 |
| INT_EN_RX_FIFO_RDY | [1] | RW | Interrupt Enable for RxFifoRdy (INT mode) 0 = Disables 1 = Enables | 0 |
| INT_EN_TX_FIFO_RDY | [0] | RW | Interrupt Enable for TxFifoRdy (INT mode) 0 = Disables 1 = Enables | 0 |

15.5.1.5 SPI_STATUSn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0014, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:26] | — | Reserved | — |
| TX_DONE | [25] | R | Indication of transfer done in Shift register (master mode only) 0 = All case except below case 1 = If Tx FIFO and shift register are empty after transmission start | 0 |
| TRAILING_BYTE | [24] | R | Indication that trailing count is 0 | 0 |
| RX_FIFO_LVL | [23:15] | R | Data level in Rx FIFO 0 to 256 bytes in port 0 0 to 64 bytes in port 1, 2 | 0 |
| TX_FIFO_LVL | [14:6] | R | Data level in Tx FIFO 0 to 256 bytes in port 0 0 to 64 bytes in port 1, 2 | 0 |
| RX_OVERRUN | [5] | R | Rx FIFO overrun error 0 = No error 1 = Overrun error | 0 |
| RX_UNDERRUN | [4] | R | Rx FIFO underrun error 0 = No error 1 = Underrun error | 0 |
| TX_OVERRUN | [3] | R | Tx FIFO overrun error 0 = No error 1 = Overrun error | 0 |
| TX_UNDERRUN | [2] | R | Tx FIFO underrun error 0 = No error 1 = Underrun error NOTE: Tx FIFO underrun error will occur if Tx FIFO is empty in slave mode. | 0 |
| RX_FIFO_RDY | [1] | R | 0 = Data in FIFO less than trigger level 1 = Data in FIFO more than trigger level | 0 |
| TX_FIFO_RDY | [0] | R | 0 = Data in FIFO more than trigger level 1 = Data in FIFO less than trigger level | 0 |

15.5.1.6 SPI_TX_DATA_n (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0018, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|--|-------------|
| TX_DATA | [31:0] | W | This field contains the data to be transmitted over the SPI channel. | 0 |

15.5.1.7 SPI_RX_DATA_n (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x001C, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|---|-------------|
| RX_DATA | [31:0] | R | This field contains the data to be received over the SPI channel. | 0 |

15.5.1.8 PACKET_CNT_REG_n (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0020, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|--|-------------|
| RSVD | [31:17] | — | Reserved | — |
| PACKET_CNT_EN | [16] | RW | Enable bit for packet count 0 = Disables 1 = Enables | 0 |
| COUNT_VALUE | [15:0] | RW | Packet count value | 0 |

15.5.1.9 PENDING_CLR_REGn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0024, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|-----------------|--------|------|--|-------------|
| RSVD | [31:5] | — | Reserved | — |
| TX_UNDERRUN_CLR | [4] | RW | Tx underrun pending clear bit 0 = Non-Clear 1 = Clears | 0 |
| TX_OVERRUN_CLR | [3] | RW | Tx overrun pending clear bit 0 = Non-Clear 1 = Clears | 0 |
| RX_UNDERRUN_CLR | [2] | RW | Rx underrun pending clear bit 0 = Non-clear 1 = Clears | 0 |
| RX_OVERRUN_CLR | [1] | RW | Rx overrun pending clear bit 0 = Non-Clear 1 = Clears | 0 |
| TRAILING_CLR | [0] | RW | Trailing pending clear bit 0 = Non-Clear 1 = Clears | 0 |

NOTE: After error interrupt pending clear, SPI controller should be reset.

Error interrupt list: Tx underrun, Tx overrun, Rx underrun, and Rx overrun.

15.5.1.10 SWAP_CFGn (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x0028, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|---------------|--------|------|---------------------------------------|-------------|
| RSVD | [31:8] | — | Reserved | — |
| RX_HWORD_SWAP | [7] | RW | 0 = Off 1 = Swap | 0 |
| RX_BYTE_SWAP | [6] | RW | 0 = Off 1 = Swap | 0 |
| RX_BIT_SWAP | [5] | RW | 0 = Off 1 = Swap | 0 |
| RX_SWAP_EN | [4] | RW | Swap Enable 0 = Normal 1 = Swap | 0 |
| TX_HWORD_SWAP | [3] | RW | 0 = Off 1 = Swap | 0 |
| TX_BYTE_SWAP | [2] | RW | 0 = Off 1 = Swap | 0 |
| TX_BIT_SWAP | [1] | RW | 0 = Off 1 = Swap | 0 |
| TX_SWAP_EN | [0] | RW | Swap Enable 0 = Normal 1 = Swap | 0 |

NOTE: Data size must be larger than swap size.

15.5.1.11 FB_CLK_SEL_n (n = 0 to 2)

- Base Address: 0x1392_0000
- Base Address: 0x1393_0000
- Base Address: 0x1394_0000
- Address = Base Address + 0x002C, Reset Value = 0x0

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| RSVD | [31:2] | – | Reserved | – |
| FB_CLK_SEL | [1:0] | RW | <p>In master mode, SPI uses a clock which is feedback from the SPICLK. The feedback clock is intended to capture safely the slave Tx signal. The slave Tx signal can lag if slave device is very far.</p> <p>There are four types of feedback clocks which experience different path delays. This register selects the feedback clock that you can use.</p> <p>Note that this register value is invalid when SPI operates in slave mode.</p> <p>00 = SPICLK bypass (do not use feedback clock) 01 = A feedback clock with 90 degree phase lagging 10 = A feedback clock with 180 degree phase lagging 11 = A feedback clock with 270 degree phase lagging 90 degree phase lagging means 5 ns delay in 50 MHz operating frequency.</p> | 0x0 |

PAD Driving Strength

PAD driving strength of SPI is controlled by setting drive strength control register in GPIO. SPI related SFR is GPBDRV (for SPI port 0 and 1) and GPCDRV (for SPI port 2).

16 Display Controller

16.1 Overview

Display controller consists of logic for transferring image data from a local bus of the camera interface controller or a video buffer located in system memory to an external LCD driver interface. The LCD driver interface supports three kinds of interfaces. They are RGB-interface, indirect-i80 interface, and YUV interface for write-back. The display controller uses up to five overlay image windows that support various color formats, 256 level alpha blending, color key, x-y position control, soft scrolling, and variable window size, among others.

Display controller supports various color formats such as RGB (1 to 24 BPP) and YCbCr 4:4:4 (only local bus). You can program the display controller to support the different requirements on screen that associates with the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

Display controller transfers the video data and generates the necessary control signals, such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, SYS_CS0, SYS_CS1, and SYS_WE. Additionally generating control signals, display controller contains data ports for video data (RGB_VD[23:0], and SYS_VD)

[Figure 16-1](#) illustrates the block diagram of display controller.

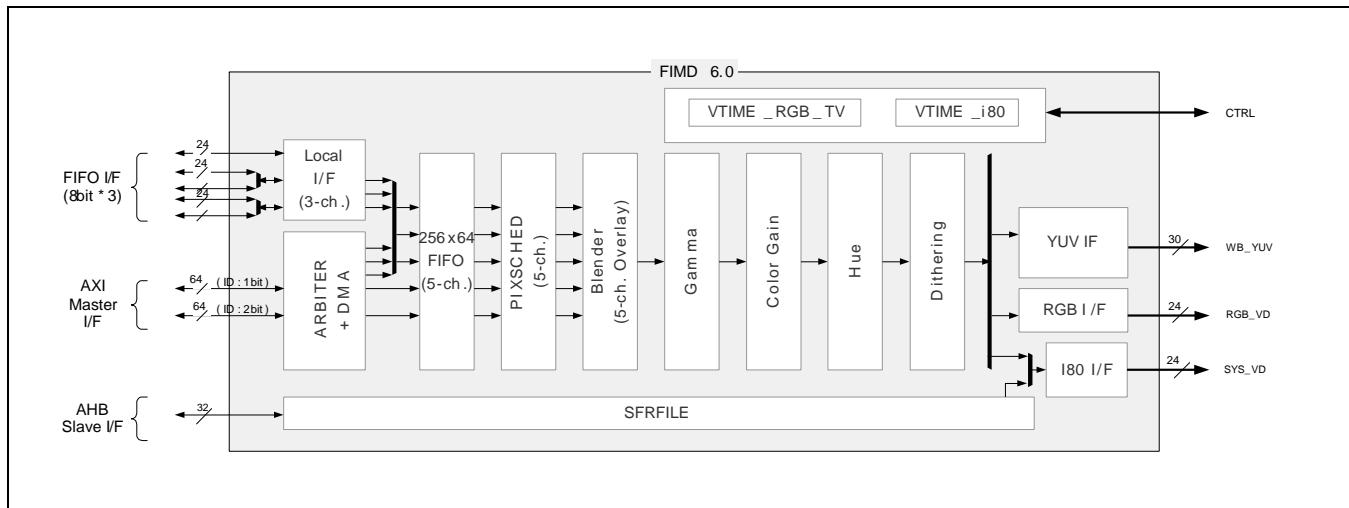


Figure 16-1 Block Diagram of Display Controller

16.2 Features

The features of the display controller include:

| | |
|---------------------------|---|
| Video Output Interface | RGB Interface (24-bit Parallel/8-bit Serial) Indirect i80 interface Write-Back interface |
| Dual Output Mode | Supports i80 and Write-Back Supports RGB and Write-Back |
| PIP (OSD) function | Supports 8-BPP (bit per pixel) palletized color Supports 16-BPP non-palletized color Supports unpacked 18 BPP non-palletized color Supports unpacked 24 BPP non-palletized color |
| | Supports X,Y indexed position |
| | Supports 8-bit Alpha blending (Plane/Pixel) |
| CSC (Internal) | RGB to YCbCr (4:2:2) |
| Source format | Window 0 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC0) Window 1 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC1) Window 2 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color Supports RGB (8:8:8) local input from Local Bus (FIMC2) Window 3/4 Supports 1, 2, 4, or 8 BPP palletized color Supports 16, 18, or 24 BPP non-palletized color |
| Configurable Burst Length | Programmable 4/8/16 Burst DMA |
| Palette | Window 0/1/2/3/4 Supports 256 × 32 bits palette memory (5EA: One palette memory for each window) |
| Soft Scrolling | Horizontal = 1 Byte resolution Vertical = 1 pixel resolution |
| Virtual Screen | Virtual image can have up to 16 MB image size. Each window can have its own virtual area. |
| Transparent Overlay | Supports transparent overlay |
| Color Key (Chroma Key) | Supports color key function Supports simultaneously color key and blending function |
| Partial Display | Supports LCD partial display function through i80 interface |

| | |
|-------------------------------|---|
| Image Enhancement | Supports gamma control |
| | Supports hue control |
| | Supports color gain control |
| Video Clock Source | SCLK_FIMD0 for display controller (from CMU module) |
| Maximum VCLK in RGB Interface | Display Controller = 80 MHz |

16.3 Functional Description

The functional description section describes the functionality of display controller.

16.3.1 Brief Description

The display controller consists of a VSFR, VDMA, VPRCS, VTIME, and video clock generator.

To configure the display controller, the VSFR has

- 121 programmable register sets
- one gamma LUT register set (64 registers)
- one i80 command register set (12 registers)
- five 256×32 palette memories

VDMA is a dedicated display DMA that transfers video data in frame memory to VPRCS. By using this special DMA, you can display video data on screen without CPU intervention.

VPRCS receives video data from VDMA and sends it to display device (LCD) through data ports (RGB_VD, or SYS_VD), after changing the video data into a suitable data format, for example, 8-bit per pixel mode (8 BPP mode) or 16-bit per pixel mode (16 BPP mode).

VTIME consists of programmable logic to support the variable requirement of interface timing and rates commonly found in different LCD drivers. The VTIME block generates RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, VEN_VSYNC, VEN_HSYNC, VEN_FIELD, VEN_HREF, SYS_CS0, SYS_CS1, SYS_WE, and so on.

Using the display controller data, you can select one of the above data paths by setting LCDBLK_CFG Register (0x1001_0210). For more information, refer to the "System Others" manual.

16.3.2 Data Flow

FIFO is in the VDMA. If FIFO is empty or partially empty, the VDMA requests data fetching from frame memory based on burst memory transfer mode. The data transfer rate determines the size of FIFO.

The display controller contains five FIFOs (Three local FIFOs and two DMA FIFOs), since it needs to support the overlay window display mode. Use one FIFO for one screen display mode.

VPRCS fetches data from FIFO. It contains the following functions for final image data: blending, image enhancing, and scheduling. It also supports the overlay function. This can overlay any image up to five window images, whose smaller or same size can be blended with the main window image having programmable alpha blending or color (chroma) key function.

[Figure 16-2](#) shows the data flow from system bus to output buffer.

VDMA has five DMA channels (Ch0-Ch4) and three local input interfaces (CAMIF0, CAMIF1, and (CAMIF2 or CAMIF3)). The Color Space Conversion (CSC) block changes Hue (YCbCr, local input only) data to RGB data for blending operation. Also, the alpha values written in SFR determine the level of blending. Data from output buffer appears in the Video Data Port.

NOTE: The performance of the all these local input interfaces is limited by the scale ratio of the input and output image resolution (TBD).

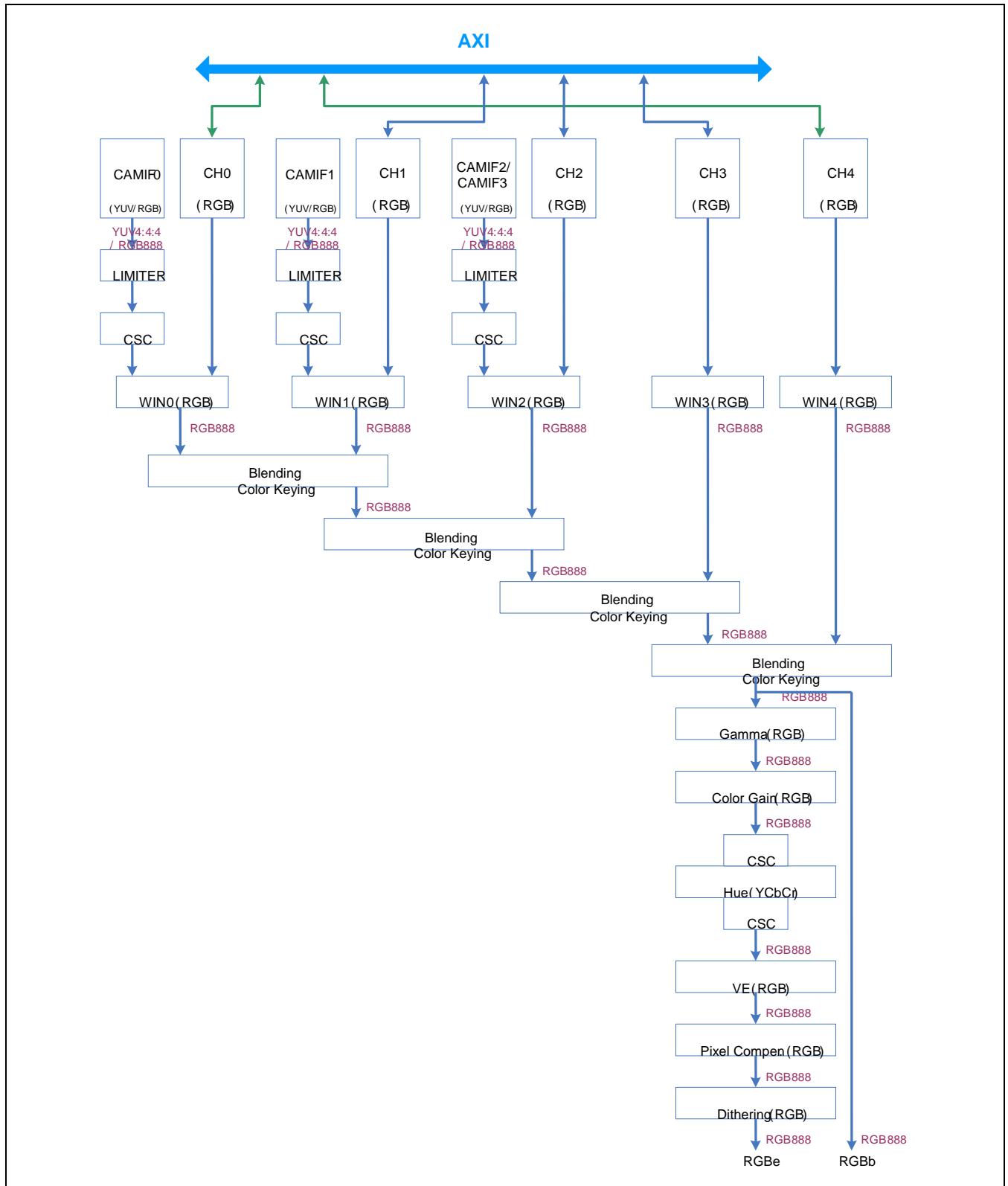


Figure 16-2 Block Diagram of the Data Flow

16.3.2.1 Interface

The display controller supports three types of interfaces:

- The first type is the conventional RGB interface, which uses RGB data, vertical/ horizontal sync, data valid signal, and data sync clock.
- The second type is the indirect i80 Interface, which uses address, data, chip select, read/ write control, and register/ status indicating signal. The LCD driver using i80 Interface contains a frame buffer and can self-refresh, so the display controller updates one still image by writing only one time to the LCD.
- The third type is FIFO interface with CAMIFx selected FIMDxWB_DEST Bit Field on CAMERA_CONTROL Register in System Register for writeback.

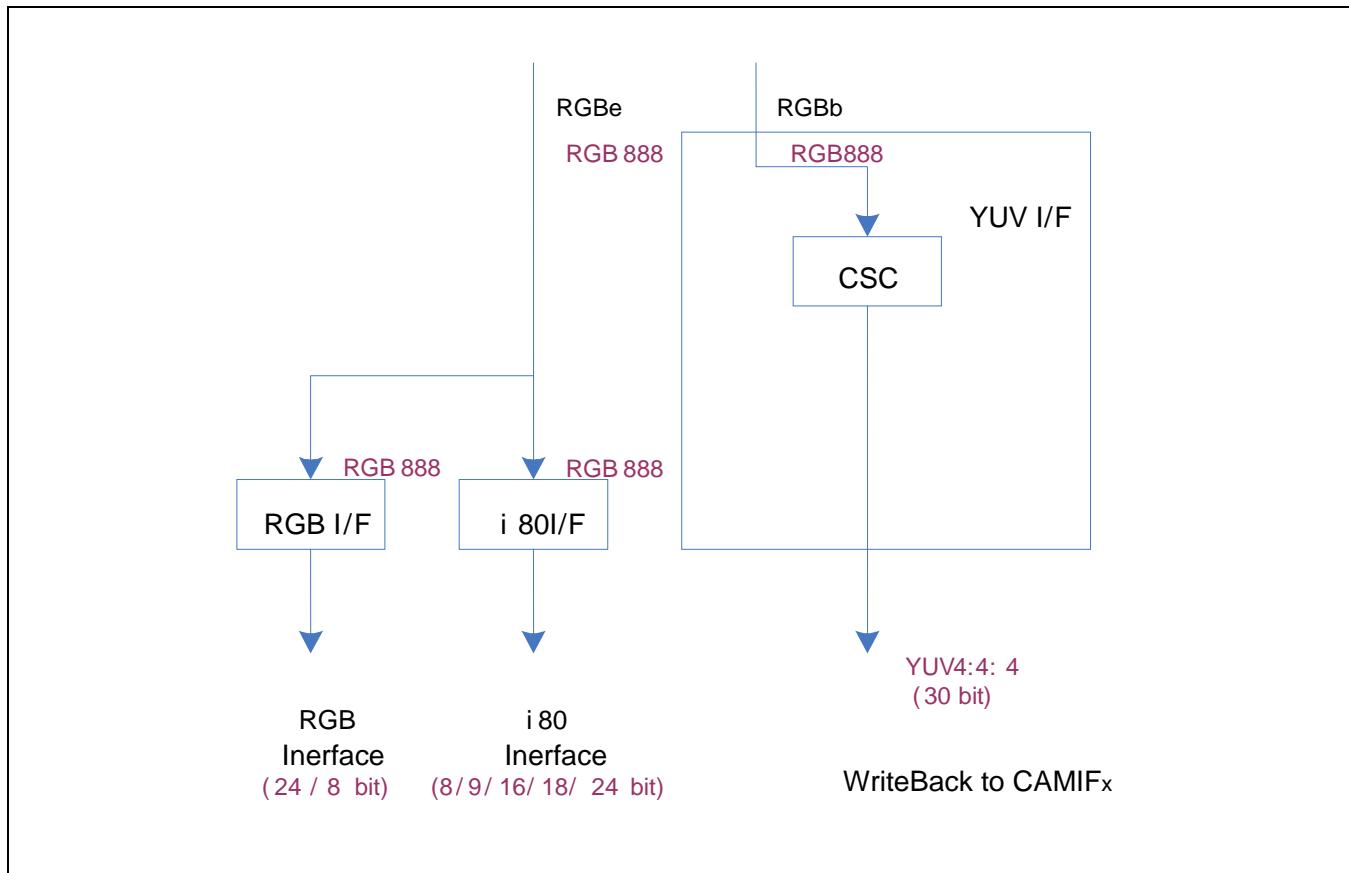


Figure 16-3 Block Diagram of the Interface

16.3.3 Overview of the Color Data

The overview of the color data section describes the RGB data format and 25 BPP display of display controller.

16.3.3.1 RGB Data Format

The display controller requests the specified memory format of frame buffer. [Figure 16-4](#) illustrates some examples of each display mode.

16.3.3.2 25 BPP Display (A888)

[Figure 16-4](#) illustrates the examples of each display mode.

| (BSWP=0, HWSWP =0, WSWP=0) | | | | | | |
|----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:57] | D[56] | D[55:32] | D[31:25] | D[24] | D[23:0] |
| 000H | Dummy Bit | AEN | P1 | Dummy Bit | AEN | P2 |
| 008H | Dummy Bit | AEN | P3 | Dummy Bit | AEN | P4 |
| 010H | Dummy Bit | AEN | P5 | Dummy Bit | AEN | P6 |
| ... | | | | | | |

| (BSWP=0, HWSWP =0, WSWP=0) | | | | | | |
|----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:57] | D[56] | D[55:32] | D[31:25] | D[24] | D[23:0] |
| 000H | Dummy Bit | AEN | P1 | Dummy Bit | AEN | P2 |
| 008H | Dummy Bit | AEN | P3 | Dummy Bit | AEN | P4 |
| 010H | Dummy Bit | AEN | P5 | Dummy Bit | AEN | P6 |
| ... | | | | | | |

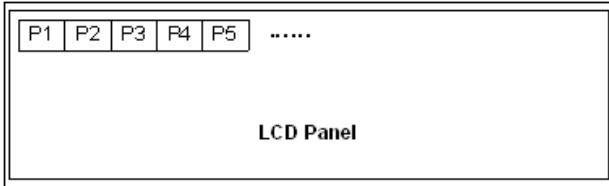


Figure 16-4 Memory Format of 25 BPP(A888) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".
2. D[23:16] = Red data, D[15:8] = Green data, and D[7:0] = Blue data.

16.3.3.2.1 32BPP (8888) Mode

[Figure 16-5](#) illustrates the pixel data that contains alpha value.

| (BYSWP=0 , HWSWP=0 , WSWP=0) | | | | |
|--------------------------------|-------------|----------|-------------|---------|
| | D[63:56] | D[55:32] | D[31:24] | D[23:0] |
| 000H | ALPHA value | P1 | ALPHA value | P2 |
| 008H | ALPHA value | P3 | ALPHA value | P4 |
| 010H | ALPHA value | P5 | ALPHA value | P6 |
| ... | | | | |

| (BYSWP=0 , HWSWP=0 , WSWP=1) | | | | |
|--------------------------------|-------------|----------|-------------|---------|
| | D[63:56] | D[55:32] | D[31:24] | D[23:0] |
| 000H | ALPHA value | P2 | ALPHA value | P1 |
| 008H | ALPHA value | P4 | ALPHA value | P3 |
| 010H | ALPHA value | P6 | ALPHA value | P5 |
| ... | | | | |

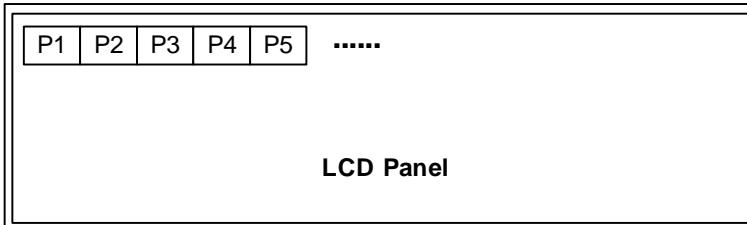
Figure 16-5 Memory Format of 32 BPP (8888) Display

16.3.3.2.2 24 BPP Display (A887)

[Figure 16-6](#) illustrates the 24 BPP display.

| (BSWP=0, HWSWP=0, WSWP=0) | | | | | | |
|-----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:56] | D[55] | D[54:32] | D[31:24] | D[23] | D[22:0] |
| 000H | Dummy Bit | AEN | P1 | Dummy Bit | AEN | P2 |
| 008H | Dummy Bit | AEN | P3 | Dummy Bit | AEN | P4 |
| 010H | Dummy Bit | AEN | P5 | Dummy Bit | AEN | P6 |
| ... | | | | | | |

| (BSWP=0, HWSWP=0, WSWP=1) | | | | | | |
|-----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:56] | D[55] | D[54:32] | D[31:24] | D[23] | D[22:0] |
| 000H | Dummy Bit | AEN | P2 | Dummy Bit | AEN | P1 |
| 008H | Dummy Bit | AEN | P4 | Dummy Bit | AEN | P3 |
| 010H | Dummy Bit | AEN | P6 | Dummy Bit | AEN | P5 |
| ... | | | | | | |



The diagram shows a large rectangular frame representing an LCD panel. Inside the panel, there is a 5x5 grid of smaller rectangles. The first row contains five rectangles labeled P1, P2, P3, P4, and P5 from left to right. To the right of the fifth rectangle is a vertical ellipsis (.....) indicating that the pattern continues. Below the grid, the text "LCD Panel" is centered.

Figure 16-6 Memory Format of 24 BPP (A887) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".
2. D[22:15] = Red data, D[14:7] = Green data, and D[6:0] = Blue data.

16.3.3.2.3 24 BPP Display (888)

[Figure 16-7](#) illustrates the 24 BPP display.

| (BSWP=0, HWSWP=0, WSWP=0) | | | | |
|-----------------------------|-----------|----------|-----------|---------|
| | D[63:56] | D[55:32] | D[31:24] | D[23:0] |
| 000H | Dummy Bit | P1 | Dummy Bit | P2 |
| 008H | Dummy Bit | P3 | Dummy Bit | P4 |
| 010H | Dummy Bit | P5 | Dummy Bit | P6 |
| ... | | | | |

| (BSWP=0, HWSWP=0, WSWP=1) | | | | |
|-----------------------------|-----------|----------|-----------|---------|
| | D[63:56] | D[55:32] | D[31:24] | D[23:0] |
| 000H | Dummy Bit | P2 | Dummy Bit | P1 |
| 008H | Dummy Bit | P4 | Dummy Bit | P3 |
| 010H | Dummy Bit | P6 | Dummy Bit | P5 |
| ... | | | | |

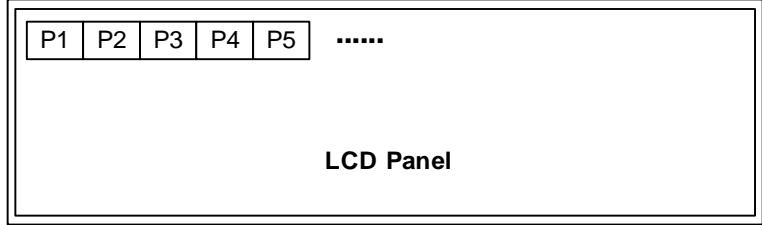


Figure 16-7 Memory Format of 24 BPP (888) Display

NOTE: D[23:16] = Red data, D[15:8] = Green data, and D[7:0] = Blue data.

16.3.3.2.4 19 BPP Display (A666)

[Figure 16-8](#) illustrates the 19 BPP display.

| (BSWP=0, HWSWP=0, WSWP=0) | | | | | | |
|-----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:51] | D[50] | D[49:32] | D[31:19] | D[18] | D[17:0] |
| 000H | Dummy Bit | AEN | P1 | Dummy Bit | AEN | P2 |
| 008H | Dummy Bit | AEN | P3 | Dummy Bit | AEN | P4 |
| 010H | Dummy Bit | AEN | P5 | Dummy Bit | AEN | P6 |
| ... | | | | | | |

| (BSWP=0, HWSWP=0, WSWP=1) | | | | | | |
|-----------------------------|-----------|-------|----------|-----------|-------|---------|
| | D[63:51] | D[50] | D[49:32] | D[31:19] | D[18] | D[17:0] |
| 000H | Dummy Bit | AEN | P2 | Dummy Bit | AEN | P1 |
| 008H | Dummy Bit | AEN | P4 | Dummy Bit | AEN | P3 |
| 010H | Dummy Bit | AEN | P6 | Dummy Bit | AEN | P5 |
| ... | | | | | | |

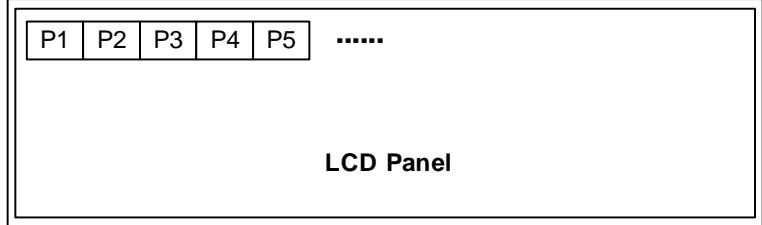


Figure 16-8 Memory Format of 19 BPP (A666) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".
2. D[17:12] = Red data, D[11:6] = Green data, and D[5:0] = Blue data.

16.3.3.2.5 18 BPP Display (666)

[Figure 16-9](#) illustrates the 18 BPP display.

| (BSWP=0, HWSWP=0, WSWP=0) | | | | |
|-----------------------------|-----------|----------|-----------|---------|
| | D[63:50] | D[49:32] | D[31:18] | D[17:0] |
| 000H | Dummy Bit | P1 | Dummy Bit | P2 |
| 008H | Dummy Bit | P3 | Dummy Bit | P4 |
| 010H | Dummy Bit | P5 | Dummy Bit | P6 |
| ... | | | | |

| (BSWP=0, HWSWP=0, WSWP=1) | | | | |
|-----------------------------|-----------|----------|-----------|---------|
| | D[63:50] | D[49:32] | D[31:18] | D[17:0] |
| 000H | Dummy Bit | P2 | Dummy Bit | P1 |
| 008H | Dummy Bit | P4 | Dummy Bit | P3 |
| 010H | Dummy Bit | P6 | Dummy Bit | P5 |
| ... | | | | |

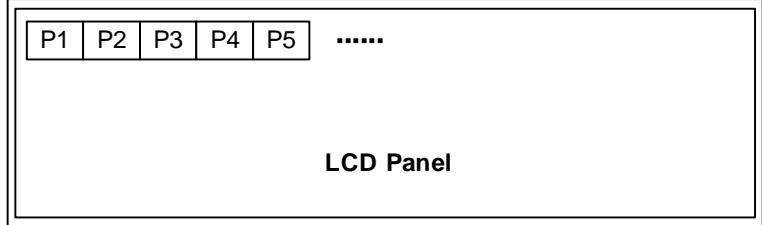


Figure 16-9 Memory Format of 18 BPP (666) Display

NOTE: D[17:12] = Red data, D[11:6] = Green data, and D[5:0] = Blue data.

16.3.3.2.6 16 BPP Display (A555)

[Figure 16-10](#) illustrates the 16 BPP display.

| (BSWP=0, HWSWP=0, WSWP=0) | | | | | | | | |
|-----------------------------|-------|----------|-------|----------|-------|----------|-------|---------|
| | D[63] | D[62:48] | D[47] | D[46:32] | D[31] | D[30:16] | D[15] | D[14:0] |
| 000H | AEN1 | P1 | AEN2 | P2 | AEN3 | P3 | AEN4 | P4 |
| 004H | AEN5 | P5 | AEN6 | P6 | AEN7 | P7 | AEN8 | P8 |
| 008H | AEN9 | P9 | AEN10 | P10 | AEN11 | P11 | AEN12 | P12 |
| ... | | | | | | | | |

| (BSWP=0, HWSWP=0, WSWP=1) | | | | | | | | |
|-----------------------------|-------|----------|-------|----------|-------|----------|-------|---------|
| | D[63] | D[62:48] | D[47] | D[46:32] | D[31] | D[30:16] | D[15] | D[14:0] |
| 000H | AEN3 | P3 | AEN4 | P4 | AEN1 | P1 | AEN2 | P2 |
| 004H | AEN7 | P7 | AEN8 | P8 | AEN5 | P5 | AEN6 | P6 |
| 008H | AEN11 | P11 | AEN12 | P12 | AEN9 | P9 | AEN10 | P10 |
| ... | | | | | | | | |

| (BSWP=0, HWSWP=1, WSWP=0) | | | | | | | | |
|-----------------------------|-------|----------|-------|----------|-------|----------|-------|---------|
| | D[63] | D[62:48] | D[47] | D[46:32] | D[31] | D[30:16] | D[15] | D[14:0] |
| 000H | AEN4 | P4 | AEN3 | P3 | AEN2 | P2 | AEN1 | P1 |
| 004H | AEN8 | P8 | AEN7 | P7 | AEN6 | P6 | AEN5 | P5 |
| 008H | AEN12 | P12 | AEN11 | P11 | AEN10 | P10 | AEN9 | P9 |
| ... | | | | | | | | |

Figure 16-10 Memory Format of 16 BPP (A555) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.

AEN = 0: Selects ALPHA0.

AEN = 1: Selects ALPHA1.

When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.

SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.

For more information, refer to the section on "SFR".

2. D[14:10] = Red data, D[9:5] = Green data, and D[4:0] = Blue data.

16.3.3.2.7 16 BPP Display (1555)

[Figure 16-11](#) illustrates the 16 BPP display.

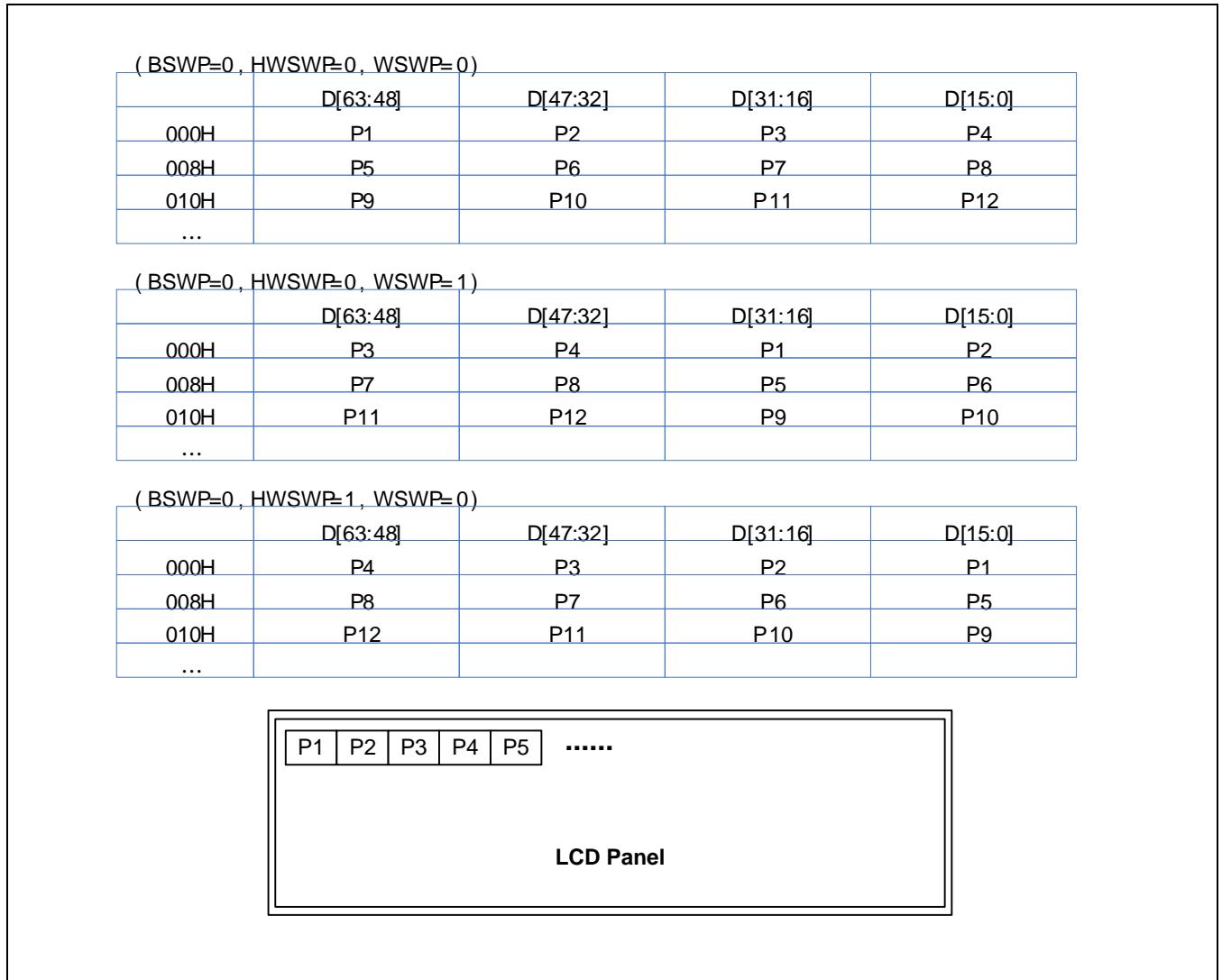


Figure 16-11 Memory Format of 16 BPP (1555) Display

NOTE: {D[14:10], D[15]} = Red data, {D[9:5], D[15]} = Green data, and {D[4:0], D[15]} = Blue data.

16.3.3.2.8 16 BPP Display (565)

[Figure 16-12](#) illustrates the 16 BPP display.

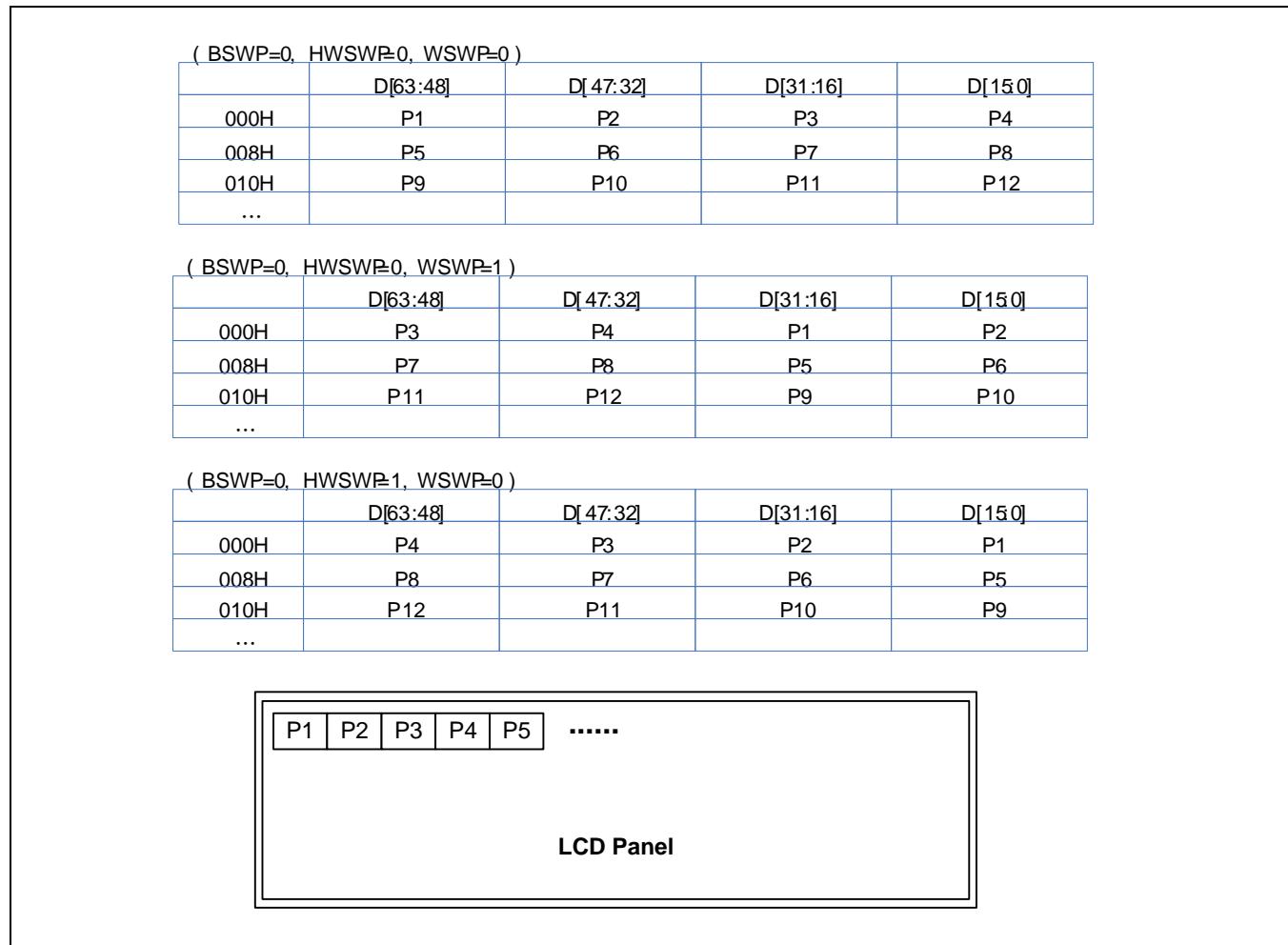


Figure 16-12 Memory Format of 16 BPP (565) Display

NOTE: D[15:10] = Red data, D[10:5] = Green data, and D[4:0] = Blue data.

[Figure 16-13](#) illustrates the 16 BPP (5:6:5) display types.

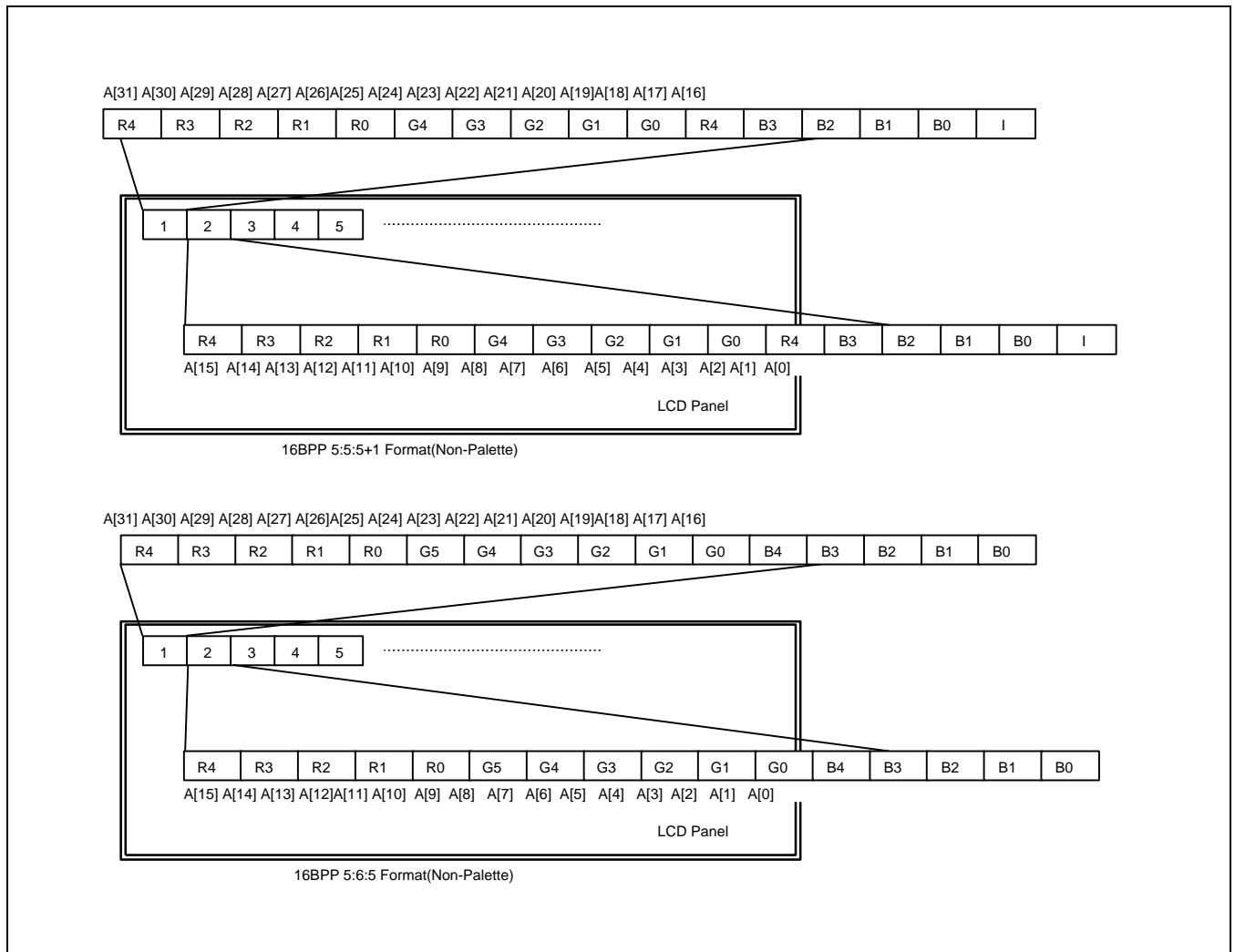


Figure 16-13 16 BPP (5:6:5) Display Types

16.3.3.2.9 13 BPP Display (A444)

[Figure 16-14](#) illustrates the 13 BPP display.

| (BYSWP=0, HWSWP=0, WSWP=0) | | | | | | | | | | | | |
|------------------------------|----------|-------|----------|----------|-------|----------|----------|-------|----------|----------|-------|---------|
| | D[63:61] | D[60] | D[59:48] | D[47:45] | D[44] | D[43:32] | D[31:29] | D[28] | D[27:16] | D[15:13] | D[12] | D[11:0] |
| 000H | Dummy | AEN1 | P1 | Dummy | AEN2 | P2 | Dummy | AEN3 | P3 | Dummy | AEN4 | P4 |
| 004H | Dummy | AEN5 | P5 | Dummy | AEN6 | P6 | Dummy | AEN7 | P7 | Dummy | AEN8 | P8 |
| 008H | Dummy | AEN9 | P9 | Dummy | AEN10 | P10 | Dummy | AEN11 | P11 | Dummy | AEN12 | P12 |
| ... | | | | | | | | | | | | |

| (BYSWP=0, HWSWP=1, WSWP=0) | | | | | | | | | | | | |
|------------------------------|----------|-------|----------|----------|-------|----------|----------|-------|----------|----------|-------|---------|
| | D[63:61] | D[60] | D[59:48] | D[47:45] | D[44] | D[43:32] | D[31:29] | D[28] | D[27:16] | D[15:13] | D[12] | D[11:0] |
| 000H | Dummy | AEN4 | P4 | Dummy | AEN3 | P3 | Dummy | AEN2 | P2 | Dummy | AEN1 | P1 |
| 004H | Dummy | AEN8 | P8 | Dummy | AEN7 | P7 | Dummy | AEN6 | P6 | Dummy | AEN5 | P5 |
| 008H | Dummy | AEN12 | P12 | Dummy | AEN11 | P11 | Dummy | AEN10 | P10 | Dummy | AEN9 | P9 |
| ... | | | | | | | | | | | | |

LCD Panel

Figure 16-14 Memory Format of 13 BPP (A444) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
2. D[11:8] = Red data, D[7:4] = Green data, and D[3:0] = Blue data.
3. 16 BPP (4444) mode. (For more information, refer to the section on "SFR") Data has Alpha value.

| (BYSWP=0, HWSWP=0, WSWP=0) | | | | | | | | |
|------------------------------|----------|----------|----------|----------|----------|----------|----------|---------|
| | D[63:60] | D[59:48] | D[47:44] | D[43:32] | D[31:28] | D[27:16] | D[15:12] | D[11:0] |
| 000H | ALPHA1 | P1 | ALPHA2 | P2 | ALPHA3 | P3 | ALPHA4 | P4 |
| 004H | ALPHA5 | P5 | ALPHA6 | P6 | ALPHA7 | P7 | ALPHA8 | P8 |
| 008H | ALPHA9 | P9 | ALPHA10 | P10 | ALPHA11 | P11 | ALPHA12 | P12 |
| ... | | | | | | | | |

16.3.3.2.10 8 BPP Display (A232)

[Figure 16-15](#) illustrates the 8 BPP display.

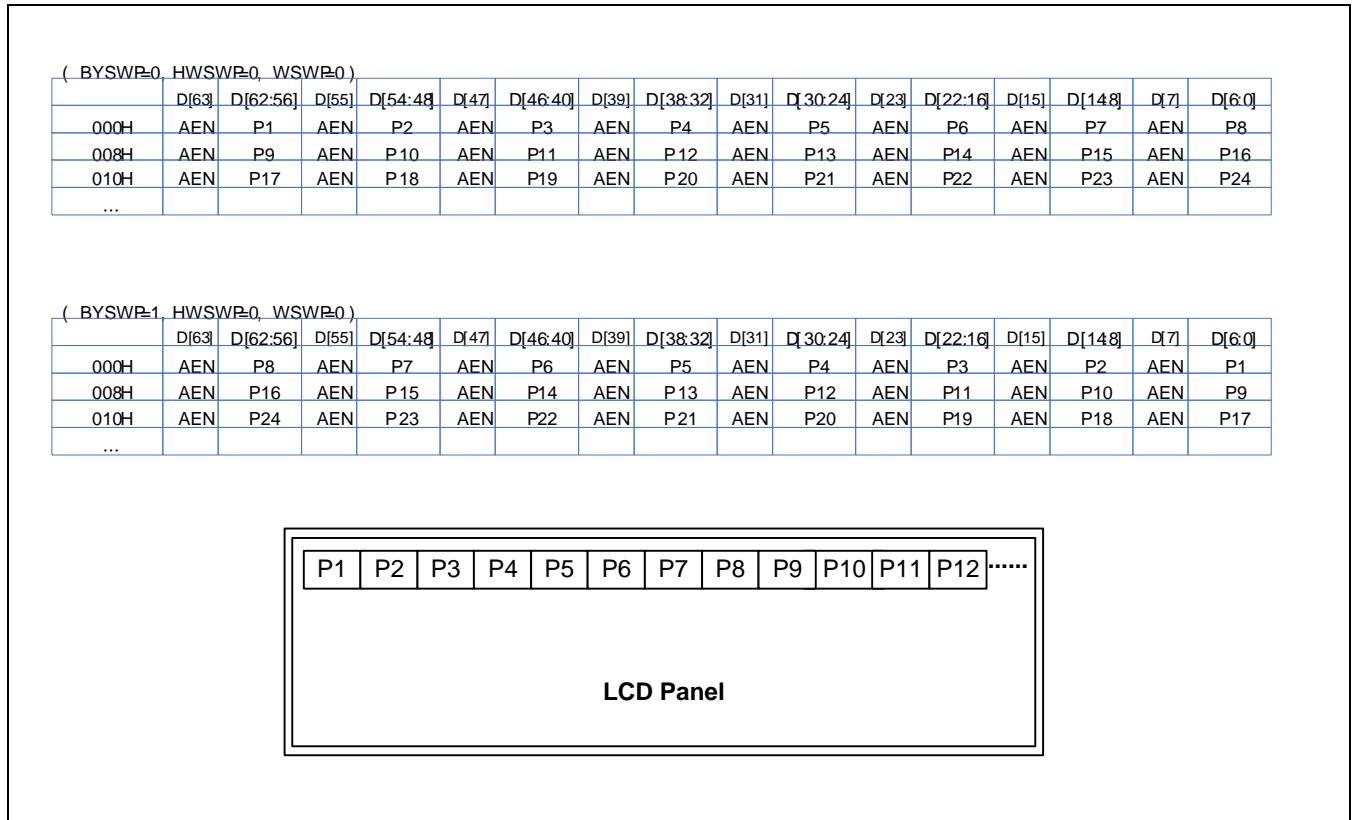


Figure 16-15 Memory Format of 8 BPP (A232) Display

NOTE:

1. AEN: Specifies the transparency value selection bit.
AEN = 0: Selects ALPHA0.
AEN = 1: Selects ALPHA1.
When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.
SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".
2. D[6:5] = Red data, D[4:2] = Green data, and D[1:0] = Blue data.

16.3.3.2.11 8 BPP Display (Palette)

[Figure 16-16](#) illustrates the 8 BPP display.

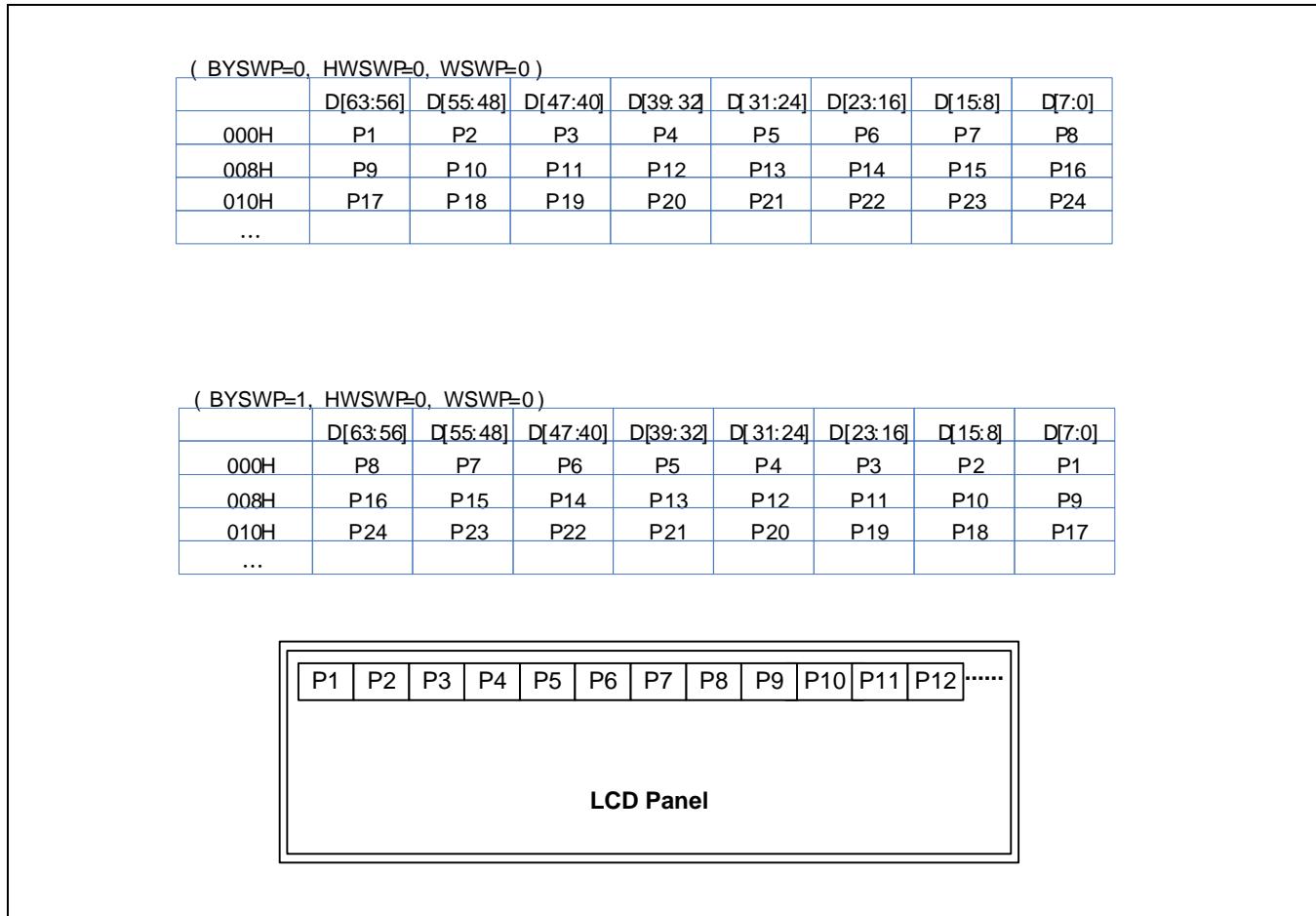


Figure 16-16 Memory Format of 8 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format).

AEN = 0: Selects ALPHA0.

AEN = 1: Selects ALPHA1.

When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.

SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.
For more information, refer to the section on "SFR".

16.3.3.2.12 4 BPP Display (Palette)

[Figure 16-17](#) illustrates the 4 BPP display.

| (BYSWP=0, HWSWP=0, WSWP=0) | | | | | | | | |
|------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | D[63:60] | D[59:56] | D[55:52] | D[51:48] | D[47:44] | D[43:40] | D[39:36] | D[35:32] |
| 000H | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
| 008H | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 |
| ... | | | | | | | | |
| | D[31:28] | D[27:24] | D[23:20] | D[19:16] | D[15:12] | D[11:8] | D[7:4] | D[3:0] |
| 000H | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 |
| 008H | P25 | P26 | P27 | P28 | P29 | P30 | P31 | P32 |
| ... | | | | | | | | |

| (BYSWP=1, HWSWP=0, WSWP=0) | | | | | | | | |
|------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | D[63:60] | D[59:56] | D[55:52] | D[51:48] | D[47:44] | D[43:40] | D[39:36] | D[35:32] |
| 000H | P15 | P16 | P13 | P14 | P11 | P12 | P9 | P10 |
| 008H | P31 | P32 | P29 | P30 | P27 | P28 | P25 | P26 |
| ... | | | | | | | | |
| | D[31:28] | D[27:24] | D[23:20] | D[19:16] | D[15:12] | D[11:8] | D[7:4] | D[3:0] |
| 000H | P7 | P8 | P5 | P6 | P3 | P4 | P1 | P2 |
| 008H | P23 | P24 | P21 | P22 | P19 | P20 | P17 | P18 |
| ... | | | | | | | | |

Figure 16-17 Memory Format of 4 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format)

AEN = 0: Selects ALPHA0.

AEN = 1: Selects ALPHA1.

When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects..

SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.

For more information, refer to the section on "SFR".

16.3.3.2.13 2 BPP Display (Palette)

[Figure 16-18](#) illustrates the 2 BPP display.

| (BYSWP=0 , HWSWP=0 , WSWP=0) | | | | | | | | |
|--------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | D[63:62] | D[61:60] | D[59:58] | D[57:56] | D[55:54] | D[53:52] | D[51:50] | D[49:48] |
| 000H | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
| 008H | P33 | P34 | P35 | P36 | P37 | P38 | P39 | P40 |
| ... | | | | | | | | |
| | D[47:46] | D[45:44] | D[43:42] | D[41:40] | D[39:38] | D[37:36] | D[35:34] | D[33:32] |
| 000H | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 |
| 008H | P41 | P42 | P43 | P44 | P45 | P46 | P47 | P48 |
| ... | | | | | | | | |
| | D[31:30] | D[29:28] | D[27:26] | D[25:24] | D[23:22] | D[21:20] | D[19:18] | D[17:16] |
| 000H | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 |
| 008H | P49 | P50 | P51 | P52 | P53 | P54 | P55 | P56 |
| ... | | | | | | | | |
| | D[15:14] | D[13:12] | D[11:10] | D[9:8] | D[7:6] | D[5:4] | D[3:2] | D[1:0] |
| 000H | P25 | P26 | P27 | P28 | P29 | P30 | P31 | P32 |
| 008H | P57 | P58 | P59 | P60 | P61 | P62 | P63 | P64 |
| ... | | | | | | | | |

Figure 16-18 Memory Format of 2 BPP Display

NOTE: AEN: Specifies the transparency value selection bit (with WPALCON: Palette output format).

AEN = 0: Selects ALPHA0.

AEN = 1: Selects ALPHA1.

When it sets per-pixel blending, then this pixel blends with alpha value that AEN selects.

SFR selects the alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B.

For more information, refer to the section on "SFR".

16.3.4 Palette Usage

The Palette Usage section includes:

- Palette Configuration and Format Control
- Palette Read/Write

16.3.4.1 Palette Configuration and Format Control

The display controller supports 256-color palette to select color mapping. You can select up to 256 colors from 32-bit colors using these formats.

256 color palette consists of 256 (depth) × 32-bit SPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5 (R: G: B), and other formats.

For Example:

See A:5:5:5 format, write palette, as illustrated in [Figure 16-20](#).

1. Connect VD pin to TFT LCD panel (R (5) = VD[23:19], G (5) = VD[15:11], and B (5) = VD[7:3]).
2. AEN bit controls the blending function, enable or disable.
3. Set WPALCON (W1PAL, case window0) register to 0'b101. The 32-bit (8:8:8:8) format has an alpha value directly, without using alpha value register (ALPHA_0/1).

[Figure 16-19](#) illustrates the 32 BPP (8:8:8:8) palette data format.

| INDEX/ Bit Pos. | 3 1 | 3 0 | 2 9 | 2 8 | 2 7 | 2 6 | 2 5 | 2 4 | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 00h | | | | | | | | | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| 01h | | | | | | | | | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | | |
| FFh | | | | | | | | | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| Number of VD | - | - | - | - | - | - | - | - | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Figure 16-19 32 BPP (8:8:8:8) Palette Data Format

[Figure 16-20](#) illustrates the 25 BPP (A: 8:8:8) palette data format.

| INDEX/ Bit Pos. | 3 1 | 3 0 | 2 9 | 2 8 | 2 7 | 2 6 | 2 5 | 2 4 | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------------------|--------|--------|--------|--------|--------|--------|--------|--------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 00 h | - | - | - | - | - | - | - | - | A E N | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| 01 h | - | - | - | - | - | - | - | - | A E N | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FFh | - | - | - | - | - | - | - | - | A E N | R 7 | R 6 | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 7 | G 6 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 7 | B 6 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 |
| Number of VD | - | - | - | - | - | - | - | - | - | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Figure 16-20 25 BPP (A: 8:8:8) Palette Data Format

[Figure 16-21](#) illustrates the 19 BPP (A: 6:6:6) palette data format.

| INDEX/ Bit Pos. | 3 1 | 3 0 | 2 9 | 2 8 | 2 7 | 2 6 | 2 5 | 2 4 | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
|--------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|---|---|
| 00 h | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 | | | |
| 01 h | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FFh | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 5 | R 4 | R 3 | R 2 | R 1 | R 0 | G 5 | G 4 | G 3 | G 2 | G 1 | G 0 | B 5 | B 4 | B 3 | B 2 | B 1 | B 0 | | | |
| Number of VD | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Figure 16-21 19 BPP (A: 6:6:6) Palette Data Format

[Figure 16-22](#) illustrates the 16 BPP (A: 5:5:5) palette data format

| INDEX / Bit Pos . | 3 1 | 3 0 | 2 9 | 2 8 | 2 7 | 2 6 | 2 5 | 2 4 | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 8 | 1 7 | 1 6 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 1 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|
| 00 h | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 4 | R 3 | R 2 | R 1 | R 0 | G 4 | G 3 | G 2 | G 1 | G 0 | B 4 | B 3 | B 2 | B 1 | B 0 | |
| 01 h | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 4 | R 3 | R 2 | R 1 | R 0 | G 4 | G 3 | G 2 | G 1 | G 0 | B 4 | B 3 | B 2 | B 1 | B 0 | |
| | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | | | |
| FFh | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | A E N | R 4 | R 3 | R 2 | R 1 | R 0 | G 4 | G 3 | G 2 | G 1 | G 0 | B 4 | B 3 | B 2 | B 1 | B 0 | |
| Number of VD | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 2 3 | 2 2 | 2 1 | 2 0 | 1 9 | 1 5 | 1 4 | 1 3 | 1 2 | 1 1 | 7 | 6 | 5 | 4 | 3 | | |

Figure 16-22 16 BPP (A: 5:5:5) Palette Data Format

16.3.4.2 Palette Read/Write

You should not access palette memory when the Vertical Status (VSTATUS) register has an ACTIVE status. You should check the VSTATUS to do Read/Write operation on the palette.

16.3.5 Window Blending

This section includes:

- Overview of Window Blending
- Blending Diagram
- Color-Key Function
- Blending and Color-Key Function

16.3.5.1 Overview of Window Blending

Window blending is the main function of VPRCS module. Display controller comprises of five window layers (win 0 to win 4).

Example 16-1 Application

The system uses

win0 as OS window, full TV screen window, and so on.
win1 as small (next channel) TV screen with win2 as menu.
win3 as caption.
win4 as channel information.
win3 and win4 have color limitation while using color index with Color LUT. This feature enhances the system performance by reducing the data rate of total system.

Example 16-2 Total Five Windows

win0 (base): Local/YCbCr, RGB without palette
win1 (Overlay1): RGB with palette
win2 (Overlay2): RGB with palette
win3 (Caption): RGB (1/2/4) with 16-level Color LUT
win4 (Cursor): RGB (1/2) with 4-level Color LUT

Overlay Priority

Win4 > Win3 > Win2 > Win1 > Win0

Color Key

24-bit RGB format should set the register value to color key register.

Example 16-3 Blending Equation

```

<Data blending>
Win01 (R,G,B) = Win0 (R,G,B) × b1 + Win1 (R,G,B) × a1
Win012 (R/G/B) = Win01 (R/G/B) × b2 + Win2 (R/G/B) × a2
Win0123 (R/G/B) = Win012 (R/G/B) × b3 + Win3 (R/G/B) × a3
WinOut (R/G/B) = Win0123 (R/G/B) × b4 + Win4 (R/G/B) × a4

, where,

Win0 (R) = Window 0's Red data,
Win0 (G) = Window 0's Green data,
Win0 (B) = Window 0's Blue data,
Win1 (R) = Window 1's Red data,
...
b1 = Background's Data blending equation1 factor,
a1 = Foreground's Data blending equation1 factor,
b2 = Background's Data blending equation2 factor,
a2 = Foreground's Data blending equation2 factor,

<Alpha value blending>
AR (G,B)01 = AR (G,B)0 × q1 + AR (G,B)1 × p1
AR (G,B)012 = AR (G,B)01 × q2 + AR (G,B)2 × p2
AR (G,B)0123 = AR (G,B)012 × q3 + AR (G,B)3 × p3

, where,

AR0 = Window 0's Red blending factor,
AG0 = Window 0's Green blending factor,
AB0 = Window 0's Blue blending factor,
AR1 = Window 1's Red blending factor, ...
AR01 = Window01's Red blending factor (alpha value blending between AR0 and AR1),
AG01 = Window01's Green blending factor (alpha value blending between AG0 and AG1),
AB01 = Window01's Blue blending factor (alpha value blending between AB0 and AB1),
AR012 = Window012's Red blending factor (alpha value blending between AR01 and AR2),
...
q1 = Background's Alpha value blending equation1 factor,
p1 = Foreground's Alpha value blending equation1 factor,
q2 = Background's Alpha value blending equation2 factor,
p2 = Foreground's Alpha value blending equation2 factor, ...

```

[Figure 16-23](#) illustrates the blending equation.

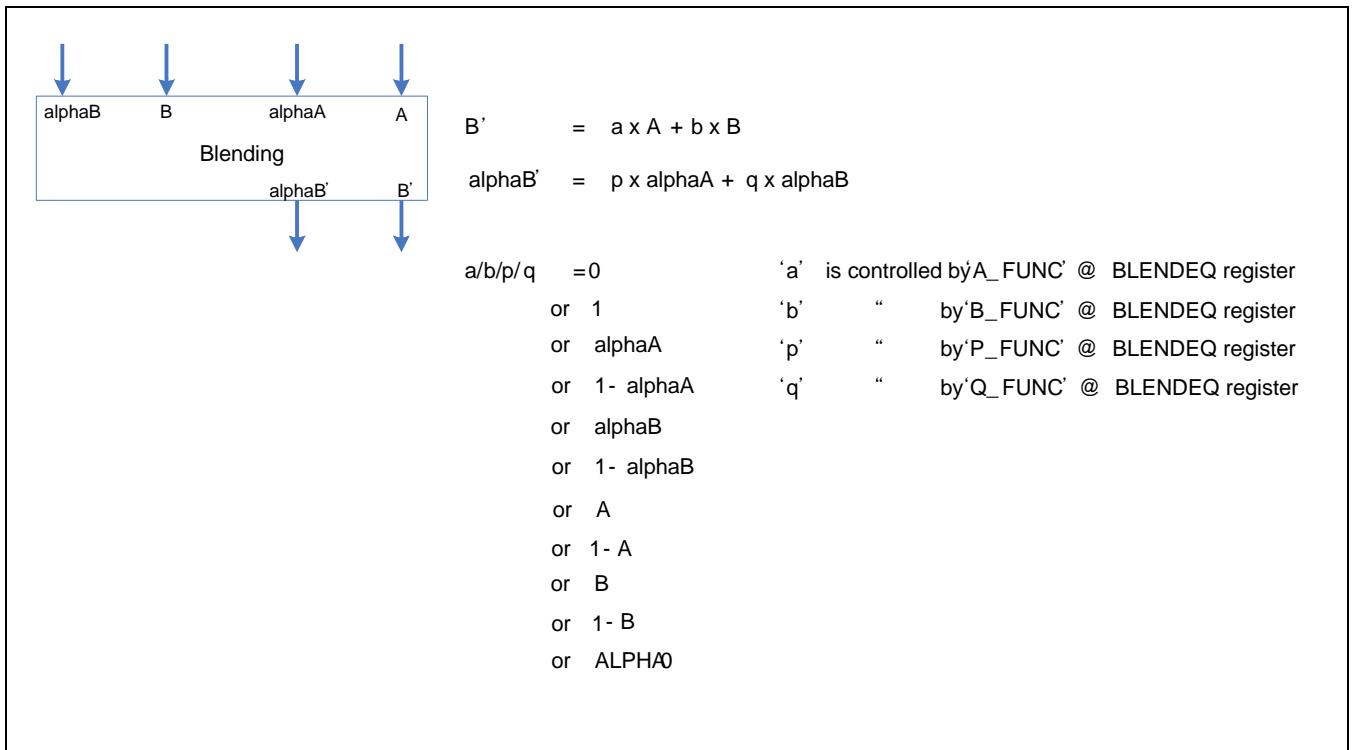


Figure 16-23 Blending Equation

Example 16-4 Default Blending Equation

```
<Data blending>
B' = B × (1 - alphaA) + A × alphaA
Alpha value blending>
alphaB' = 0 (= alphaB × 0 + alphaA × 0)
```

16.3.5.2 Blending Diagram

The display controller can blend five layers for one pixel at the same time. ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, and ALPHA1_B registers control the alpha value (blending factor), which you can implement for each window layer and color (R, G, B).

The example below shows the R (Red) output using ALPHA_R value of each window.

All windows have two kinds of alpha blending value:

Alpha value that enables transparency (AEN value == 1) Alpha value that disables transparency (AEN value == 0)
If you enable WINEN_F and BLD_PIX and disable ALPHA_SEL, then it selects the AR. The equation to select the AR is:

- AR = (Pixel (R)'s AEN value == 1'b1) ? Reg (ALPHA1_R): Reg (ALPHA0_R);
- AG = (Pixel (G)'s AEN value == 1'b1) ? Reg (ALPHA1_G): Reg (ALPHA0_G);
- AB = (Pixel (B)'s AEN value == 1'b1) ? Reg (ALPHA1_B): Reg (ALPHA0_B);
(where, BLD_PIX == 1, ALPHA_SEL == 0)

If you enable WINEN_F and disable BLD_PIX, then the ALPHA_SEL ALPHA0 controls the AR. AEN bit information is not used anymore.

[Figure 16-24](#) illustrates the blending diagram.

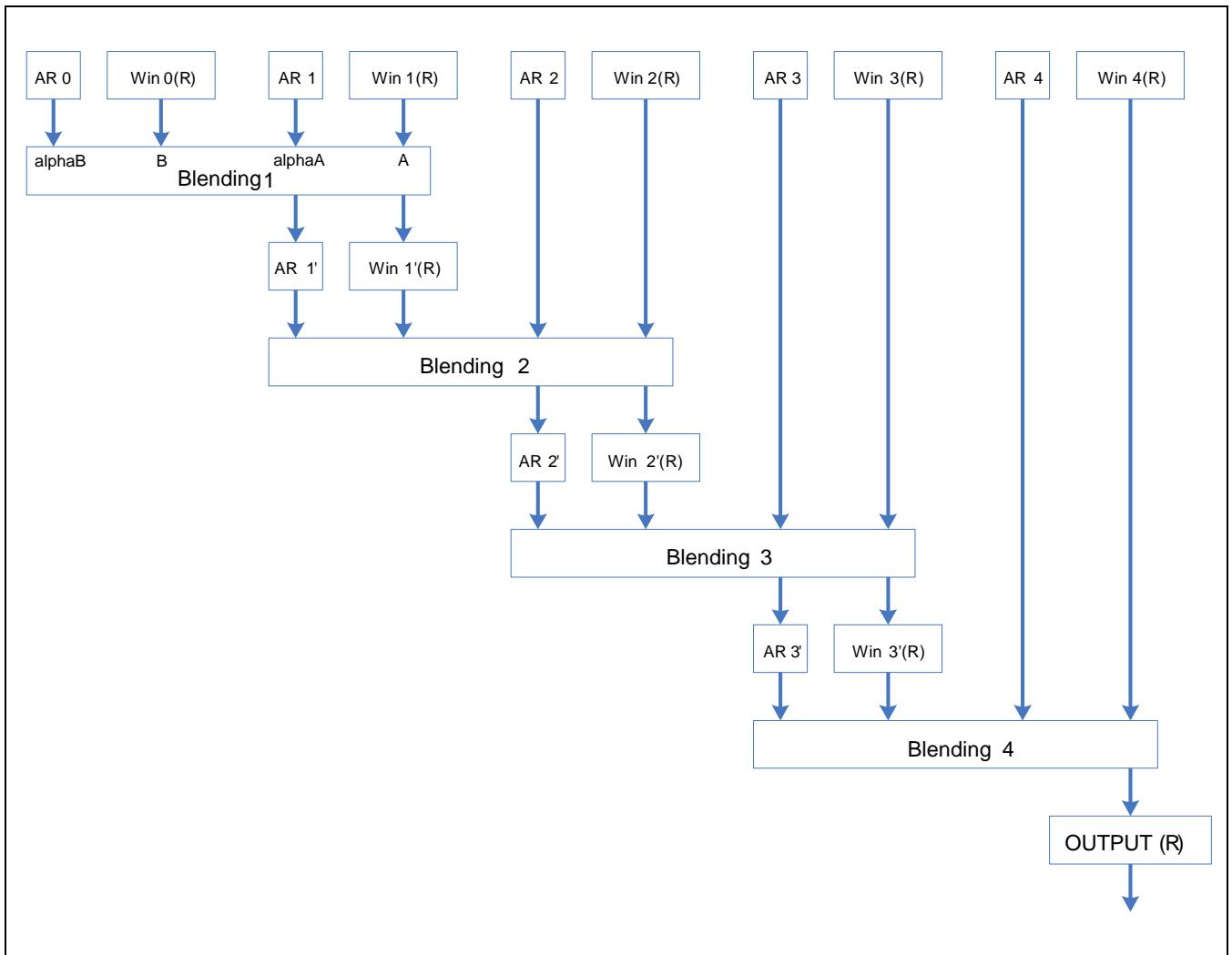


Figure 16-24 Blending Diagram

Example 16-5 Window Blending Factor Decision

Window n's blending factor decision ($n = 0, 1, 2, 3, 4$). For more information, refer to the section on "SFR".

[Figure 16-25](#) illustrates the blending factor decision.

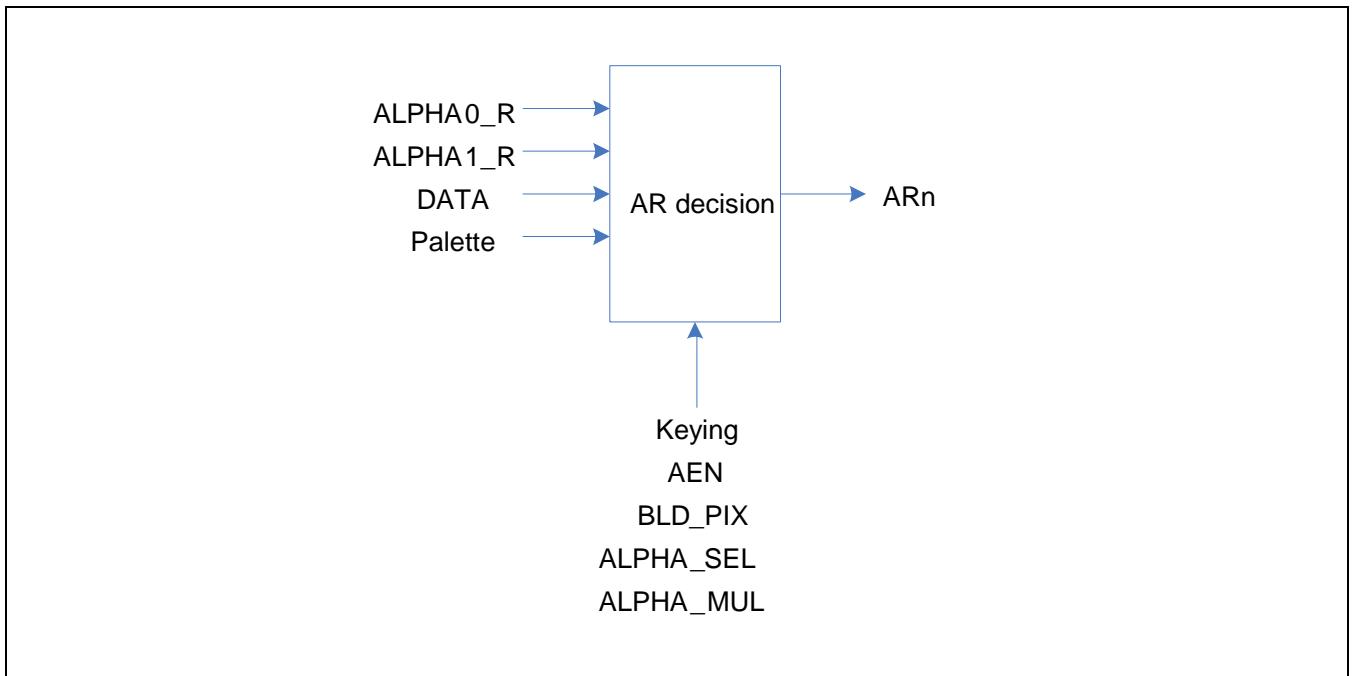


Figure 16-25 Blending Factor Decision

NOTE: If you use DATA[15:12] (BPPMODE_F = b'1110, ARGB4444 format) to blend, then the alpha value is {DATA[15:12], DATA [15:12]} (4-bit → 8-bit expanding).

16.3.5.3 Color-Key Function

The Color-Key function in display controller supports various effects for image mapping. For special functionality, the Color-Key register that specifies the color image of OSD layer is substituted by the background image-either as cursor image or preview image of the camera.

[Figure 16-26](#) illustrates the Color-Key function configurations.

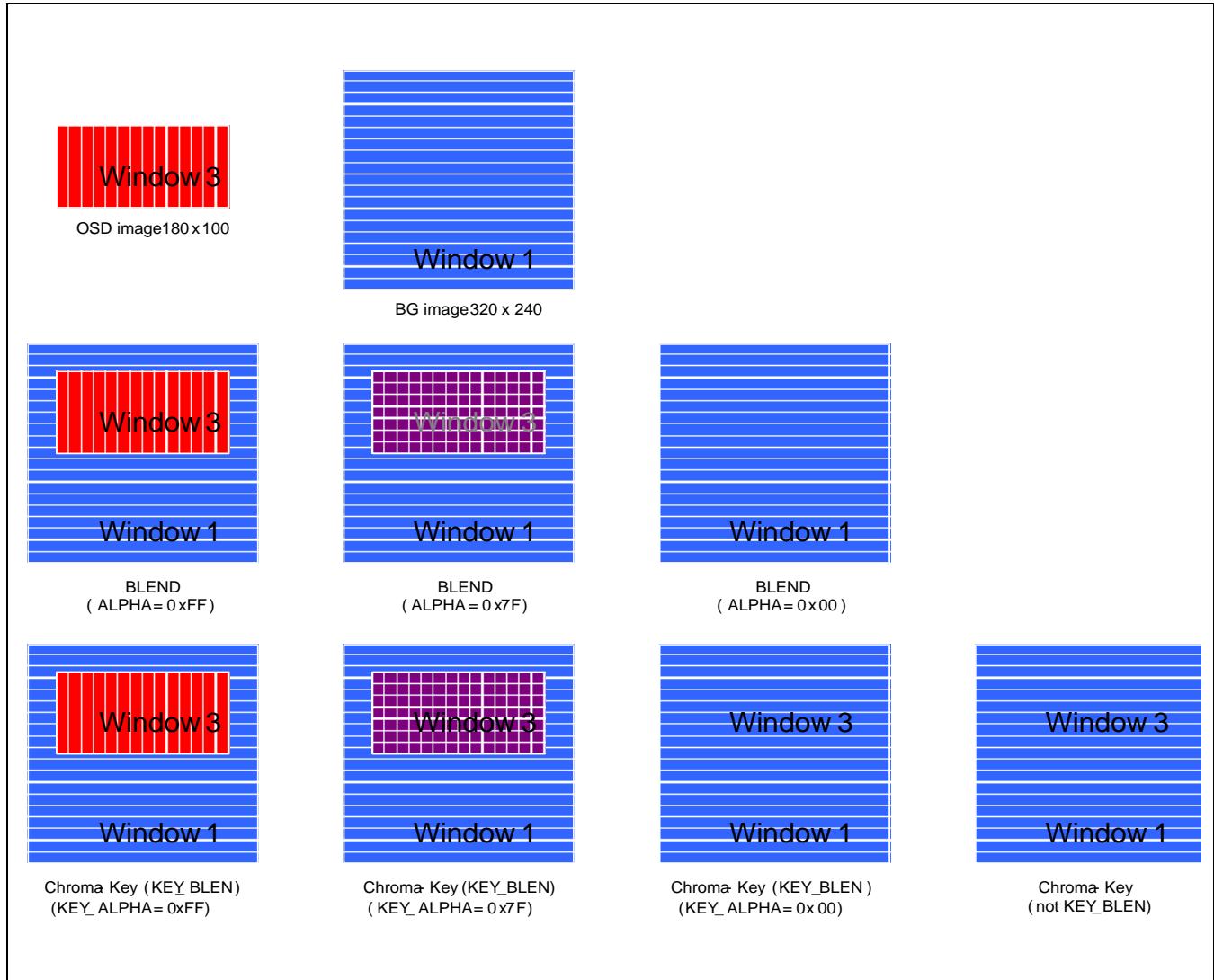


Figure 16-26 Color-Key Function Configurations

16.3.5.4 Blending and Color-Key Function

The display controller supports simultaneous blending function-with two transparency factors and Color-Key function in the same window.

[Figure 16-27](#) illustrates the blending and Color-Key function.

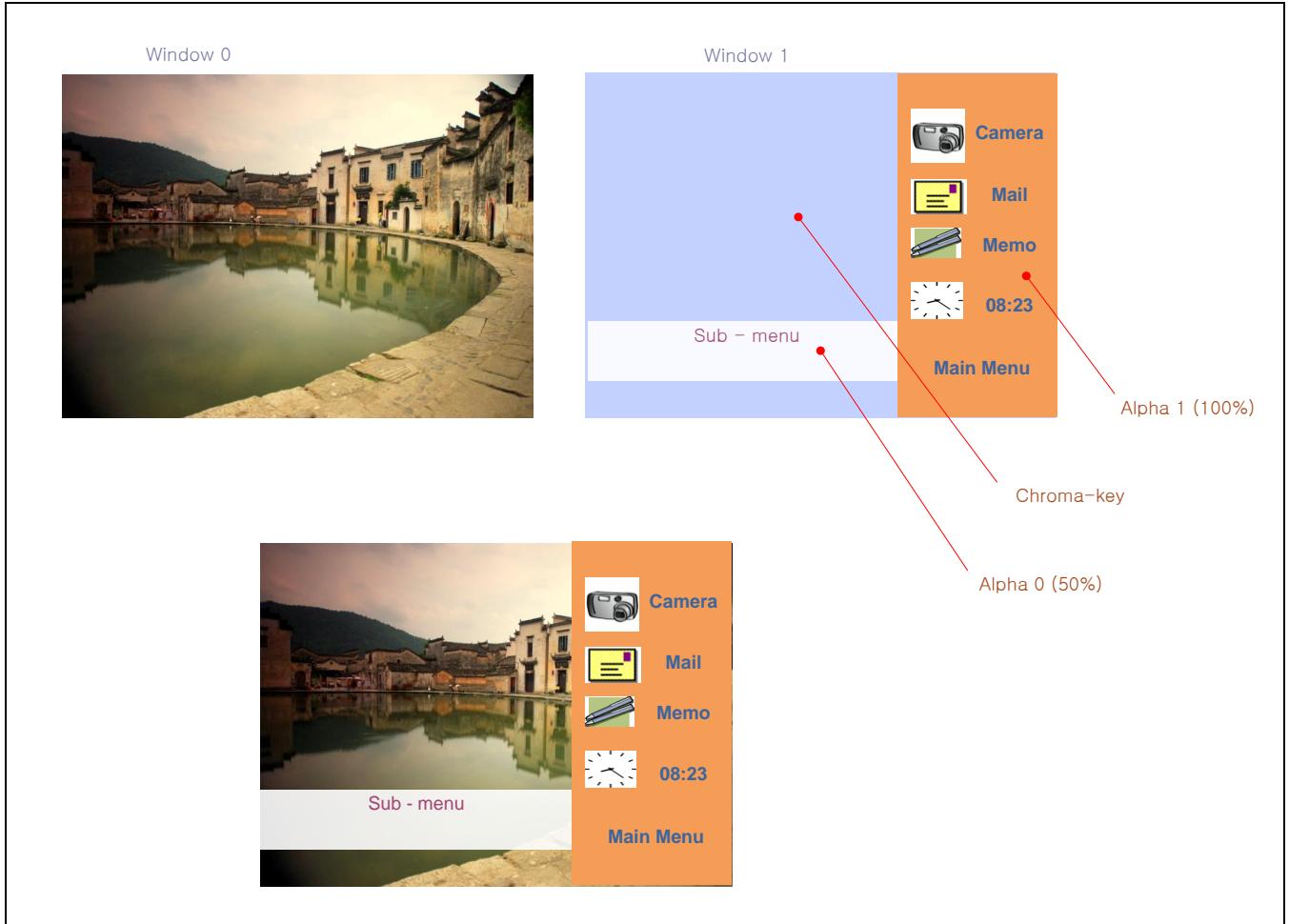


Figure 16-27 Blending and Color-Key Function

[Figure 16-28](#) illustrates the blending decision diagram.

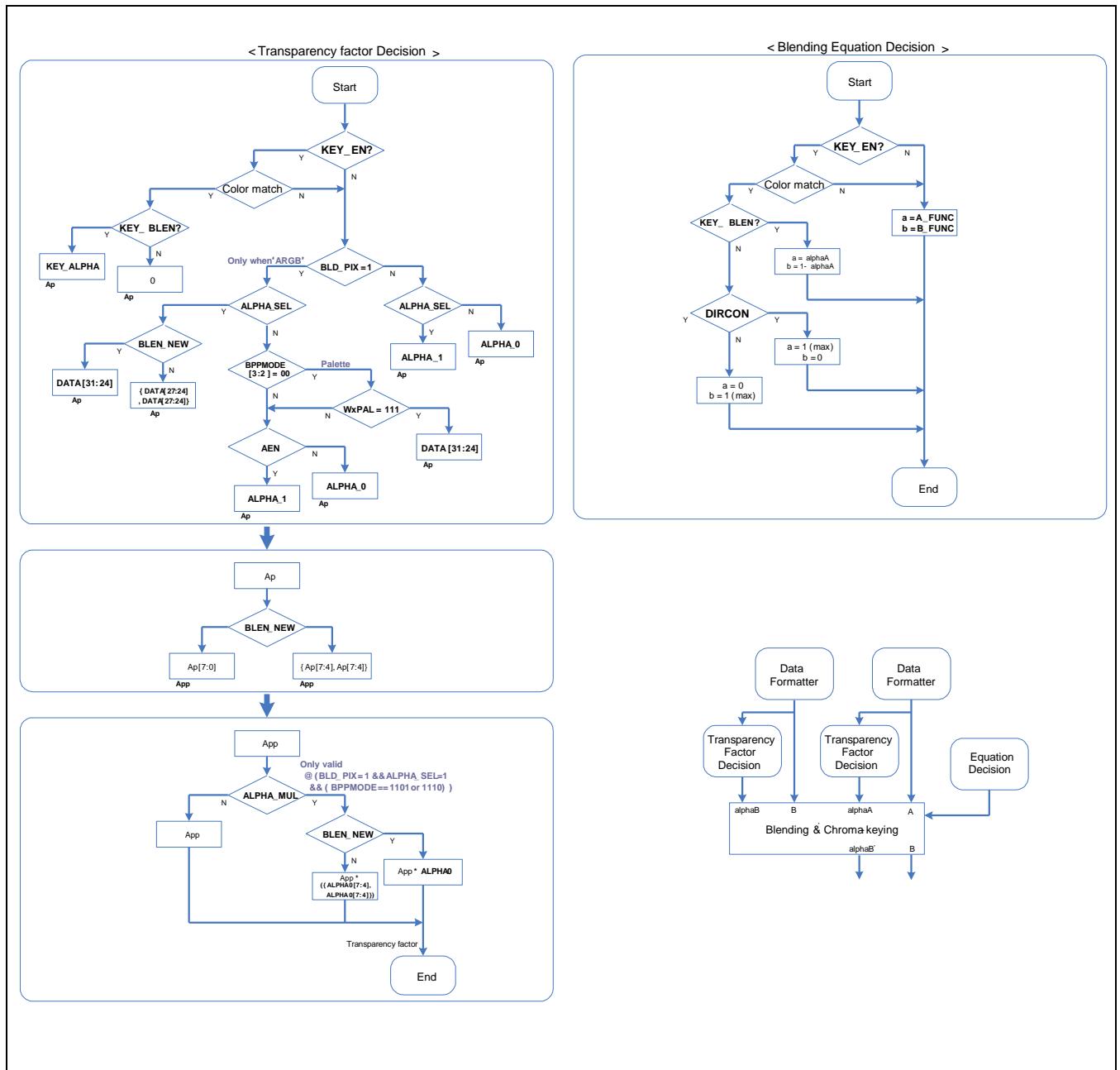


Figure 16-28 Blending Decision Diagram

16.3.6 VTIME Controller Operation

VTIME comprises of two blocks, namely:

- VTIME_RGB_TV for RGB timing control
- VTIME_i80 for indirect i80 interface timing control

16.3.6.1 RGB Interface Controller

VTIME generates control signals such as RGB_VSYNC, RGB_HSYNC, RGB_VDEN, and RGB_VCLK signal for the RGB interface. You can use these control signals while configuring the VIDTC0/1/2 registers in the VSFR register.

You can program configurations of display control registers in the VSFR. Then, the VTIME module generates programmable control signals that support different types of display devices.

The RGB_VSYNC signal causes the LCD line pointer to begin at the top of display. The configuration of both HOZVAL field and LINEVAL registers control pulse generation of RGB_VSYNC and RGB_HSYNC. Based on these equations, the size of the LCD panel determines HOZVAL and LINEVAL:

- HOZVAL = (Horizontal display size) – 1
- LINEVAL = (Vertical display size) – 1

The CLKVAL field in VIDCON0 register controls the rate of RGB_VCLK signal.

$\text{RGB_VCLK (Hz)} = \text{SCLK_FIMDx}/(\text{CLKVAL} + 1)$, where $\text{CLKVAL} \geq 1$ where, $\text{SCLK_FIMDx } (x = 0, 1)$

[Table 16-1](#) describes the relationship of RGB_VCLK and CLKVAL. The minimum value of CLKVAL is 1.

- $\text{RGB_VCLK (Hz)} = \text{SCLK_FIMDx}/(\text{CLKVAL} + 1)$, where $\text{CLKVAL} \geq 1$ where, $\text{SCLK_FIMDx } (x = 0, 1)$

**Table 16-1 Relation 16 BPP between VCLK and CLKVAL
(TFT, Frequency of Video Clock Source = 60 MHz)**

| CLKVAL | 60 MHz/X | VCLK |
|--------|-----------|-----------|
| 2 | 60 MHz/3 | 20.0 MHz |
| 3 | 60 MHz/4 | 15.0 MHz |
| : | : | : |
| 63 | 60 MHz/64 | 937.5 kHz |

VSYNC, VBPD, VFPD, HSYNC, HBPD, HFPD, HOZVAL, and LINEVAL configure RGB_HSYNC and RGB_VSYNC signal. For more information, refer to $\text{RGB_VCLK (Hz)} = \text{SCLK_FIMDx}/(\text{CLKVAL} + 1)$, where $\text{CLKVAL} \geq 1$ where, $\text{SCLK_FIMDx } (x = 0, 1)$

The frame rate is RGB_VSYNC signal frequency. The frame rate associates with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, and CLKVAL registers. Most LCD drivers require their own adequate frame rate.

The equation to calculate frame rate is:

$$\text{Frame Rate} = 1 / [\{ (VSPW + 1) + (VBPD + 1) + (LIINEVAL + 1) + (VFPD + 1) \} \times \{ (HSPW + 1) + (HBPD + 1) + (HFPD + 1) + (HOZVAL + 1) \} \times \{ (CLKVAL + 1) / (\text{Frequency of Clock source}) \}]$$

16.3.6.2 i80 Interface Controller

VTIME_I80 controls display controller for CPU style LCD Driver IC (LDI).

The functions of interface controller are:

- Generates I80 Interface Control Signals
- CPU style LDI Command Control
- Timing Control for VDMA and VDPRCS

16.3.6.3 Output Control Signal Generation

VTIME_I80 generates SYS_CS0, SYS_CS1, SYS_WE, and SYS_RS control signals (for Timing Diagram, refer to RGB_VCLK (Hz) = SCLK_FIMDx/(CLKVAL + 1), where CLKVAL ≥ 1 where, SCLK_FIMDx (x = 0, 1)).

SYS_CS0, SYS_CS1, SYS_WE and SYS_RS timing parameters, LCD_CS_SETUP, LCD_WR_SETUP, LCD_WR_ACT, and LCD_WR_HOLD are set through I80IFCONA0 and I80IFCONA1 SFRs.

16.3.6.4 Partial Display Control

Although partial display is the main feature of CPU style LDI, VTIME_I80 does not support this function in hardware logic.

SFR setting (LINEVAL, HOZVAL, OSD_LeftTopX_F, OSD_LeftTopY_F, OSD_RightBotX_F, OSD_RightBotY_F, PAGEWIDTH, and OFFSIZE) implements partial display function.

16.3.6.5 LDI Command Control

LDI receives both command and data. Command specifies an index for selecting the SFR in LDI. In control signal for command and data, only SYS_RS signal has a special function. Usually, SYS_RS has a polarity of '1' for issuing command and vice versa.

Display controller has two kinds of command control:

- Auto command
- Normal command

Auto command is issued automatically, that is, without software control and at a pre-defined rate (rate = 2, 4, ..., 30). If the rate is equal to 4, then it implies that auto commands are sent to LDI at the end of every four image frames.

Normal command: The software control issues Normal command.

16.3.6.6 Interrupt

Completion of one frame generates Frame Done Interrupt.

16.3.7 Virtual Display

Display controller supports hardware horizontal or vertical scrolling. If the screen scrolls, then it changes the fields of LCDBASEU and LCDBASEL (Refer to [Figure 16-29](#)), but not the values of PAGEWIDTH and OFFSIZE. The size of video buffer in which you store the image should be larger than the LCD panel screen size.

[Figure 16-29](#) illustrates the example of scrolling in virtual display.

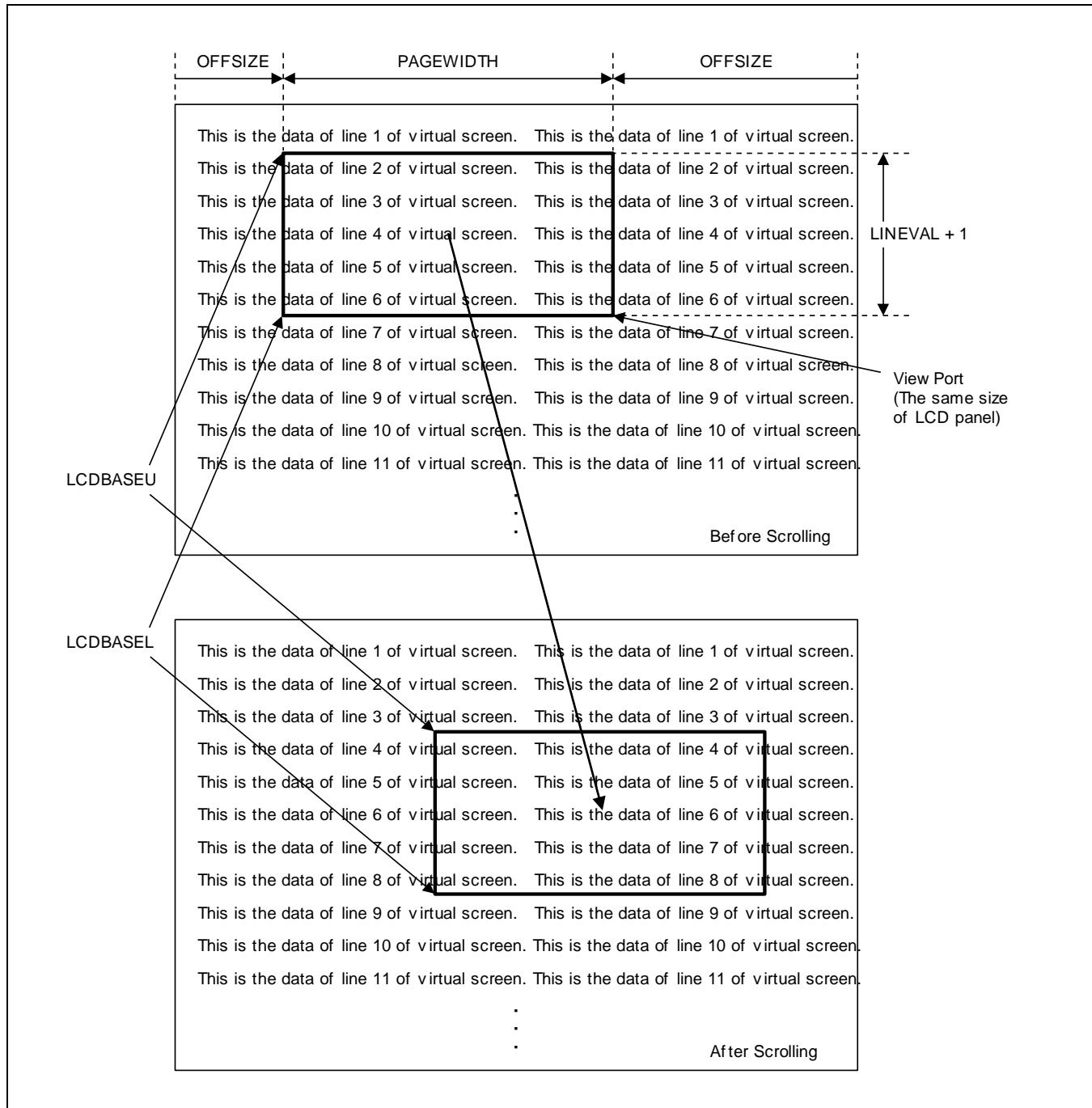


Figure 16-29 Example of Scrolling in Virtual Display

16.3.8 RGB Interface Specification

RGB Interface Spec includes:

- Signals
- LCD RGB Interface Timing
- Parallel Output
- Serial 8-bit Output
- Output Configuration Structure

16.3.8.1 Signals

[Table 16-2](#) describes the signals.

Table 16-2 RGB Interface Signals of Display Controller

| Signal | In/Out | Description | Display Controller | |
|---------------|---------------|------------------------------------|---------------------------|---------------------|
| | | | PADs | GPIO Control |
| LCD_HSYNC | O | Horizontal Synchronization. Signal | XvHsync | GPF0CON[0] |
| LCD_VSYNC | O | Vertical Synchronization. Signal | XvVsync | GPF0CON[1] |
| LCD_VCLK | O | LCD Video Clock | XvVclk | GPF0CON[3] |
| LCD_VDEN | O | Data Enable | XvVden | GPF0CON[2] |
| LCD_VD[0] | O | RGB data output | XvVd_0 | GPF0CON[4] |
| LCD_VD[1] | O | RGB data output | XvVd_1 | GPF0CON[5] |
| LCD_VD[2] | O | RGB data output | XvVd_2 | GPF0CON[6] |
| LCD_VD[3] | O | RGB data output | XvVd_3 | GPF0CON[7] |
| LCD_VD[4] | O | RGB data output | XvVd_4 | GPF1CON[0] |
| LCD_VD[5] | O | RGB data output | XvVd_5 | GPF1CON[1] |
| LCD_VD[6] | O | RGB data output | XvVd_6 | GPF1CON[2] |
| LCD_VD[7] | O | RGB data output | XvVd_7 | GPF1CON[3] |
| LCD_VD[8] | O | RGB data output | XvVd_8 | GPF1CON[4] |
| LCD_VD[9] | O | RGB data output | XvVd_9 | GPF1CON[5] |
| LCD_VD[10] | O | RGB data output | XvVd_10 | GPF1CON[6] |
| LCD_VD[11] | O | RGB data output | XvVd_11 | GPF1CON[7] |
| LCD_VD[12] | O | RGB data output | XvVd_12 | GPF2CON[0] |
| LCD_VD[13] | O | RGB data output | XvVd_13 | GPF2CON[1] |
| LCD_VD[14] | O | RGB data output | XvVd_14 | GPF2CON[2] |
| LCD_VD[15] | O | RGB data output | XvVd_15 | GPF2CON[3] |
| LCD_VD[16] | O | RGB data output | XvVd_16 | GPF2CON[4] |
| LCD_VD[17] | O | RGB data output | XvVd_17 | GPF2CON[5] |
| LCD_VD[18] | O | RGB data output | XvVd_18 | GPF2CON[6] |

| Signal | In/Out | Description | Display Controller | |
|---------------|---------------|--------------------|---------------------------|---------------------|
| | | | PADs | GPIO Control |
| LCD_VD[19] | O | RGB data output | XvVD_19 | GPF2CON[7] |
| LCD_VD[20] | O | RGB data output | XvVD_20 | GPF3CON[0] |
| LCD_VD[21] | O | RGB data output | XvVD_21 | GPF3CON[1] |
| LCD_VD[22] | O | RGB data output | XvVD_22 | GPF3CON[2] |
| LCD_VD[23] | O | RGB data output | XvVD_23 | GPF3CON[3] |

While using RGB interface, the VT_LBLKx bit fields in LCDBLKC_CFG (0x1001_0210) register should be set to RGB Interface out (2'b00), even though you use DSI Video Mode.

16.3.8.2 LCD RGB Interface Timing

[Figure 16-30](#) illustrates the LCD RGB interface timing.

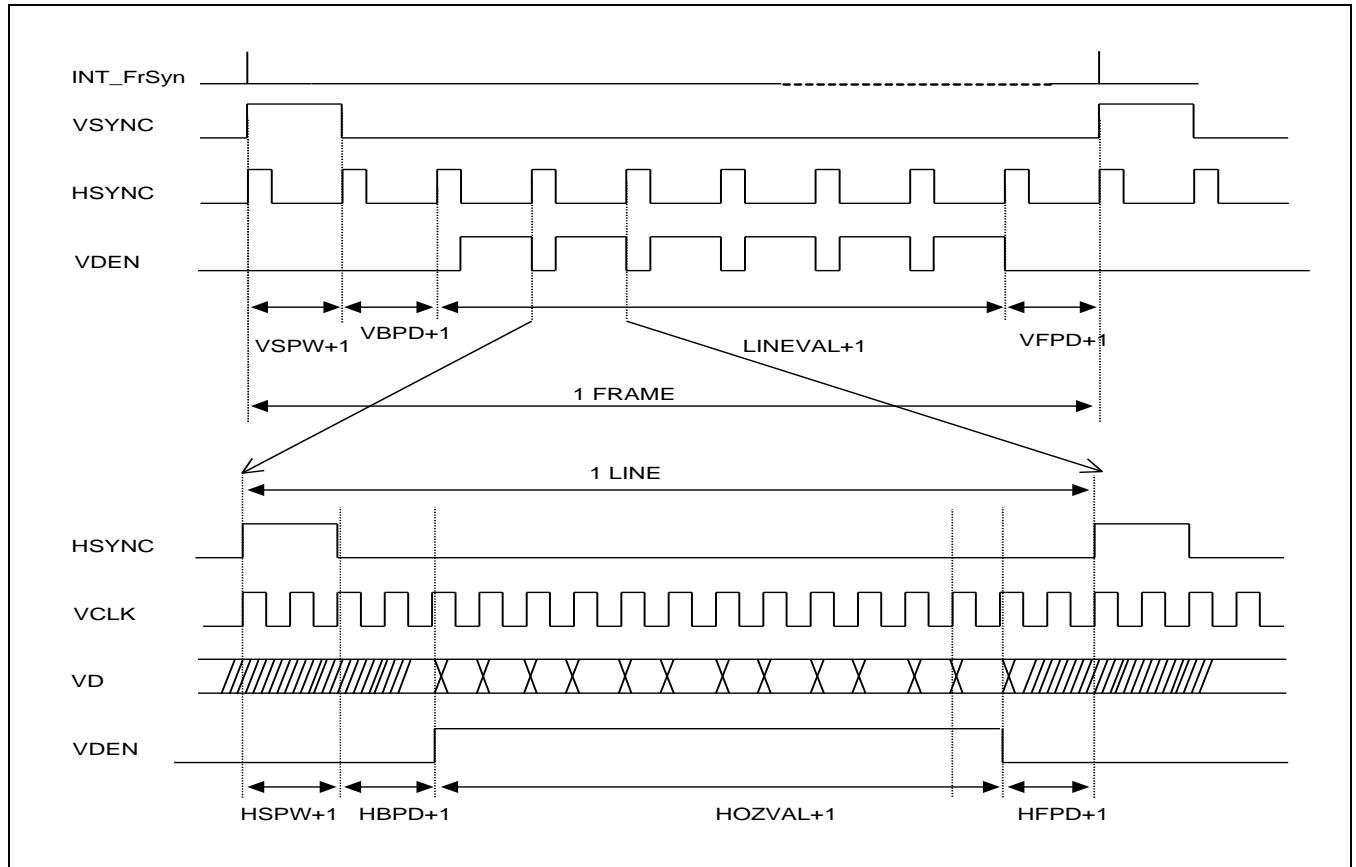


Figure 16-30 LCD RGB Interface Timing

16.3.9 LCD Indirect i80 System Interface

LCD Indirect i80 System Interface includes:

- Signals
- Indirect i80 System Interface Write Cycle Timing

16.3.9.1 Signals

[Table 16-3](#) describes the signals.

Table 16-3 LCD Indirect i80 System Interface Signals of Display Controller

| Signal | In/Out | Description | Display Controller | |
|-------------------|---------------|----------------------|---------------------------|---------------------|
| | | | PAD | GPIO Control |
| SYS_VD[0] | I/O | Data bit[0] | XvVD_0 | GPF0CON[4] |
| SYS_VD[1] | I/O | Data bit[1] | XvVD_1 | GPF0CON[5] |
| SYS_VD[2] | I/O | Data bit[2] | XvVD_2 | GPF0CON[6] |
| SYS_VD[3] | I/O | Data bit[3] | XvVD_3 | GPF1CON[7] |
| SYS_VD[4] | I/O | Data bit[4] | XvVD_4 | GPF1CON[0] |
| SYS_VD[5] | I/O | Data bit[5] | XvVD_5 | GPF1CON[1] |
| SYS_VD[6] | I/O | Data bit[6] | XvVD_6 | GPF1CON[2] |
| SYS_VD[7] | I/O | Data bit[7] | XvVD_7 | GPF1CON[3] |
| SYS_VD[8] | I/O | Data bit[8] | XvVD_8 | GPF1CON[4] |
| SYS_VD[9] | I/O | Data bit[9] | XvVD_9 | GPF1CON[5] |
| SYS_VD[10] | I/O | Data bit[10] | XvVD_10 | GPF1CON[6] |
| SYS_VD[11] | I/O | Data bit[11] | XvVD_11 | GPF1CON[7] |
| SYS_VD[12] | I/O | Data bit[12] | XvVD_12 | GPF2CON[0] |
| SYS_VD[13] | I/O | Data bit[13] | XvVD_13 | GPF2CON[1] |
| SYS_VD[14] | I/O | Data bit[14] | XvVD_14 | GPF2CON[2] |
| SYS_VD[15] | I/O | Data bit[15] | XvVD_15 | GPF2CON[3] |
| SYS_VD[16] | I/O | Data bit[16] | XvVD_16 | GPF2CON[4] |
| SYS_VD[17] | I/O | Data bit[17] | XvVD_17 | GPF2CON[5] |
| SYS_CS0 | O | Chip select for LCD0 | XvHSYNC | GPF0CON[0] |
| SYS_CS1 | O | Chip select for LCD1 | XvVSYNC | GPF0CON[1] |
| SYS_WE | O | Write enable | XvVCLK | GPF0CON[3] |
| SYS_OE | O | Output enable | XvSYS_OE | GPF3CON[5] |
| SYS_RS/SYS_ADD[0] | O | Address Output[0] | XvVDEN | GPF0CON[2] |
| SYS_ST/SYS_ADD[1] | O | Address Output[1] | – | Internal Connection |

NOTE:

1. SYS_ST/SYS_ADD[1] is valid in DSI Mode (VIDCON0 [30] = 1)
SYS_ADD[1] = SYS_ST: 0 when VDOUT is from Frame
SYS_ADD[1] = SYS_ST: 1 when VDOUT is from Command
2. While using RGB interface, set the VT_LBLKx bit fields in LCDBLKC_CFG (0x1001_0210) register to i80 interface out (2'b01), even though you use DSI Command Mode.

16.3.9.2 Indirect i80 System Interface Write Cycle Timing

[Figure 16-31](#) illustrates the indirect i80 system interface write cycle timing.

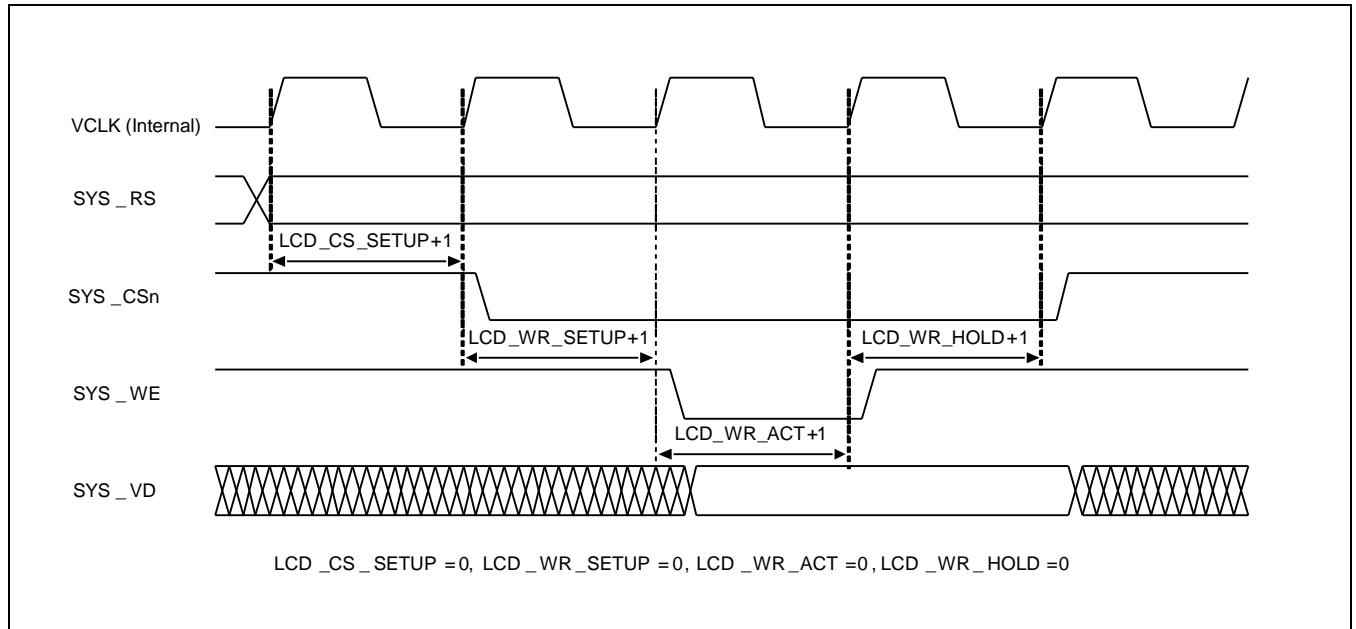


Figure 16-31 Indirect i80 System Interface Write Cycle Timing

Table 16-4 describes the timing reference code (XY Definition).

Table 16-4 Timing Reference Code (XY Definition)

| - | Parallel RGB | | | Serial RGB | |
|--------|--------------|--------------|--------------|--------------|--------------|
| | 24 BPP (888) | 18 BPP (666) | 16 BPP (565) | 24 BPP (888) | 18 BPP (666) |
| VD[23] | R[7] | R[5] | R[4] | D[7] | D[5] |
| VD[22] | R[6] | R[4] | R[3] | D[6] | D[4] |
| VD[21] | R[5] | R[3] | R[2] | D[5] | D[3] |
| VD[20] | R[4] | R[2] | R[1] | D[4] | D[2] |
| VD[19] | R[3] | R[1] | R[0] | D[3] | D[1] |
| VD[18] | R[2] | R[0] | - | D[2] | D[0] |
| VD[17] | R[1] | - | - | D[1] | - |
| VD[16] | R[0] | - | - | D[0] | - |
| VD[15] | G[7] | G[5] | G[5] | - | - |
| VD[14] | G[6] | G[4] | G[4] | - | - |
| VD[13] | G[5] | G[3] | G[3] | - | - |
| VD[12] | G[4] | G[2] | G[2] | - | - |
| VD[11] | G[3] | G[1] | G[1] | - | - |
| VD[10] | G[2] | G[0] | G[0] | - | - |
| VD[9] | G[1] | - | - | - | - |
| VD[8] | G[0] | - | - | - | - |
| VD[7] | B[7] | B[5] | B[4] | - | - |
| VD[6] | B[6] | B[4] | B[3] | - | - |
| VD[5] | B[5] | B[3] | B[2] | - | - |
| VD[4] | B[4] | B[2] | B[1] | - | - |
| VD[3] | B[3] | B[1] | B[0] | - | - |
| VD[2] | B[2] | B[0] | - | - | - |
| VD[1] | B[1] | - | - | - | - |
| VD[0] | B[0] | - | - | - | - |

16.4 I/O Description

[Table 16-5](#) describes the I/O.

Table 16-5 I/O Signals of Display Controller

| Signal | In/Out | Description | PAD for Display Controller | Type ⁽¹⁾ |
|-----------------------|--------|---|----------------------------|---------------------|
| LCD_HSYNC | Out | Horizontal Synchronization. Signal | XvHsync | Muxed |
| LCD_VSYNC | Out | Vertical Synchronization. Signal | XvVsync | Muxed |
| LCD_VCLK | Out | LCD Video Clock | XvVclk | Muxed |
| LCD_VDEN | Out | Data Enable | XvVden | Muxed |
| LCD_VD[23:0] | Out | RGB Data Output | XvVd_23 to XvVd_0 | Muxed |
| SYS_VD[17:0] | In/Out | Data to/from Display Controller from/to Display Module | XvVd_17 to XvVd_0 | Muxed |
| SYS_CS0 | Out | Chip select for LCD0 | XvHsync | Muxed |
| SYS_CS1 | Out | Chip select for LCD1 | XvVsync | Muxed |
| SYS_WE | Out | Write Enable | XvVclk | Muxed |
| SYS_OE | Out | Output Enable | XvSys_Oe | Muxed |
| SYS_RS/ SYS_ADD[0] | Out | Address Output SYS_ADD[0] is Register/State select | XvVden | Muxed |
| VSYNC_LDI | Out | VSYNC signal for Vsync Interface ⁽²⁾ | XvVsync_Ldi | Muxed |
| LCD_FRM | Out | Frame Synchronization Signal for general use ⁽³⁾ | XpwmTout_0 | Muxed |

NOTE:

1. Type field indicates type (or kind) of the pad whether it is dedicated to a signal or connected to multiplexed signals.
2. VSYNCEN register controls VSYNC_LD signal.
3. FRMEN control register controls LCD_FRM signal.

16.5 Register Description

Overview

The registers you can use to configure display controller are:

1. VIDCON0: Configures video output format and displays enable/disable.
2. VIDCON1: Specifies RGB I/F control signal.
3. VIDCON2: Specifies output data format control.
4. VIDCON3: Specifies image enhancement control.
5. I80IFCONx: Specifies CPU interface control signal.
6. VIDTCONx: Configures video output timing and determines the size of display.
7. WINCONx: Specifies each window feature setting.
8. VIDOSDxA, VIDOSDxB: Specifies window position setting.
9. VIDOSDxC, D: Specifies On Screen Display (OSD) size setting.
10. VIDWxALPHA0/1: Specifies alpha value setting.
11. BLENDEQx: Specifies blending equation setting.
12. VIDWxxADDx: Specifies source image address setting.
13. WxKEYCONx: Specifies color key setting register.
14. WxKEYALPHA: Specifies color key alpha value setting.
15. WINxMAP: Specifies window color control.
16. GAMMALUT_xx: Specifies gamma value setting.
17. COLORGAINCON: Specifies color gain value setting.
18. HUExx: Specifies Hue coefficient and offset value setting.
19. WPALCON: Specifies palette control register.
20. WxRTQOSCON: Specifies RTQoS control register.
21. WxPDATAx: Specifies window palette data of each index.
22. SHDOWCON: Specifies shadow control register.
23. WxRTQOSCON: Specifies QoS control register.

16.5.1 Register Map Summary

- Base Address = 0x11C0_0000.

| Register | Offset | Description | Reset Value |
|-------------------------|--------|---|-------------|
| Control Register | | | |
| VIDCON0 | 0x0000 | Specifies video control 0 register. | 0x0000_0000 |
| VIDCON1 | 0x0004 | Specifies video control 1 register. | 0x0000_0000 |
| VIDCON2 | 0x0008 | Specifies video control 2 register. | 0x0000_0000 |
| VIDCON3 | 0x000C | Specifies video control 3 register. | 0x0000_0000 |
| VIDTCCON0 | 0x0010 | Specifies video time control 0 register. | 0x0000_0000 |
| VIDTCCON1 | 0x0014 | Specifies video time control 1 register. | 0x0000_0000 |
| VIDTCCON2 | 0x0018 | Specifies video time control 2 register. | 0x0000_0000 |
| VIDTCCON3 | 0x001C | Specifies video time control 3 register. | 0x0000_0000 |
| WINCON0 | 0x0020 | Specifies window control 0 register. | 0x0000_0000 |
| WINCON1 | 0x0024 | Specifies window control 1 register. | 0x0000_0000 |
| WINCON2 | 0x0028 | Specifies window control 2 register. | 0x0000_0000 |
| WINCON3 | 0x002C | Specifies window control 3 register. | 0x0000_0000 |
| WINCON4 | 0x0030 | Specifies window control 4 register. | 0x0000_0000 |
| SHADOWCON | 0x0034 | Specifies window shadow control register. | 0x0000_0000 |
| WINCHMAP2 | 0x003C | Specifies window and channel mapping control register. | 0x7D51_7D51 |
| VIDOSD0A | 0x0040 | Specifies video window 0's position control register. | 0x0000_0000 |
| VIDOSD0B | 0x0044 | Specifies video window 0's position control register. | 0x0000_0000 |
| VIDOSD0C | 0x0048 | Specifies video window 0's size control register. | 0x0000_0000 |
| VIDOSD1A | 0x0050 | Specifies video window 1's position control register. | 0x0000_0000 |
| VIDOSD1B | 0x0054 | Specifies video window 1's position control register | 0x0000_0000 |
| VIDOSD1C | 0x0058 | Specifies video window 1's alpha control register. | 0x0000_0000 |
| VIDOSD1D | 0x005C | Specifies video window 1's size control register. | 0x0000_0000 |
| VIDOSD2A | 0x0060 | Specifies video window 2's position control register. | 0x0000_0000 |
| VIDOSD2B | 0x0064 | Specifies video window 2's position control register. | 0x0000_0000 |
| VIDOSD2C | 0x0068 | Specifies video window 2's alpha control register. | 0x0000_0000 |
| VIDOSD2D | 0x006C | Specifies video window 2's size control register. | 0x0000_0000 |
| VIDOSD3A | 0x0070 | Specifies video window 3's position control register. | 0x0000_0000 |
| VIDOSD3B | 0x0074 | Specifies video window 3's position control register. | 0x0000_0000 |
| VIDOSD3C | 0x0078 | Specifies video window 3's alpha control register. | 0x0000_0000 |
| VIDOSD4A | 0x0080 | Specifies video window 4's position control register. | 0x0000_0000 |
| VIDOSD4B | 0x0084 | Specifies video window 4's position control register. | 0x0000_0000 |
| VIDOSD4C | 0x0088 | Specifies video window 4's alpha control register. | 0x0000_0000 |
| VIDW00ADD0B0 | 0x00A0 | Specifies window 0's buffer start address register, buffer 0. | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|--------------|--------|---|-------------|
| VIDW00ADD0B1 | 0x00A4 | Specifies window 0's buffer start address register, buffer 1. | 0x0000_0000 |
| VIDW00ADD0B2 | 0x20A0 | Specifies window 0's buffer start address register, buffer 2. | 0x0000_0000 |
| VIDW01ADD0B0 | 0x00A8 | Specifies window 1's buffer start address register, buffer 0. | 0x0000_0000 |
| VIDW01ADD0B1 | 0x00AC | Specifies window 1's buffer start address register, buffer 1. | 0x0000_0000 |
| VIDW01ADD0B2 | 0x20A8 | Specifies window 1's buffer start address register, buffer 2. | 0x0000_0000 |
| VIDW02ADD0B0 | 0x00B0 | Specifies window 2's buffer start address register, buffer 0. | 0x0000_0000 |
| VIDW02ADD0B1 | 0x00B4 | Specifies window 2's buffer start address register, buffer 1. | 0x0000_0000 |
| VIDW02ADD0B2 | 0x20B0 | Specifies window 2's buffer start address register, buffer 2. | 0x0000_0000 |
| VIDW03ADD0B0 | 0x00B8 | Specifies window 3's buffer start address register, buffer 0. | 0x0000_0000 |
| VIDW03ADD0B1 | 0x00BC | Specifies window 3's buffer start address register, buffer 1. | 0x0000_0000 |
| VIDW03ADD0B2 | 0x20B8 | Specifies window 3's buffer start address register, buffer 2. | 0x0000_0000 |
| VIDW04ADD0B0 | 0x00C0 | Specifies window 4's buffer start address register, buffer 0. | 0x0000_0000 |
| VIDW04ADD0B1 | 0x00C4 | Specifies window 4's buffer start address register, buffer 1. | 0x0000_0000 |
| VIDW04ADD0B2 | 0x20C0 | Specifies window 4's buffer start address register, buffer 2. | 0x0000_0000 |
| VIDW00ADD1B0 | 0x00D0 | Specifies window 0's buffer end address register, buffer 0. | 0x0000_0000 |
| VIDW00ADD1B1 | 0x00D4 | Specifies window 0's buffer end address register, buffer 1. | 0x0000_0000 |
| VIDW00ADD1B2 | 0x20D0 | Specifies window 0's buffer end address register, buffer 2. | 0x0000_0000 |
| VIDW01ADD1B0 | 0x00D8 | Specifies window 1's buffer end address register, buffer 0. | 0x0000_0000 |
| VIDW01ADD1B1 | 0x00DC | Specifies window 1's buffer end address register, buffer 1. | 0x0000_0000 |
| VIDW01ADD1B2 | 0x20D8 | Specifies window 1's buffer end address register, buffer 2. | 0x0000_0000 |
| VIDW02ADD1B0 | 0x00E0 | Specifies window 2's buffer end address register, buffer 0. | 0x0000_0000 |
| VIDW02ADD1B1 | 0x00E4 | Specifies window 2's buffer end address register, buffer 1. | 0x0000_0000 |
| VIDW02ADD1B2 | 0x20E0 | Specifies window 2's buffer end address register, buffer 2. | 0x0000_0000 |
| VIDW03ADD1B0 | 0x00E8 | Specifies window 3's buffer end address register, buffer 0. | 0x0000_0000 |
| VIDW03ADD1B1 | 0x00EC | Specifies window 3's buffer end address register, buffer 1. | 0x0000_0000 |
| VIDW03ADD1B2 | 0x20E8 | Specifies window 3's buffer end address register, buffer 2. | 0x0000_0000 |
| VIDW04ADD1B0 | 0x00F0 | Specifies window 4's buffer end address register, buffer 0. | 0x0000_0000 |
| VIDW04ADD1B1 | 0x00F4 | Specifies window 4's buffer end address register, buffer 1. | 0x0000_0000 |
| VIDW04ADD1B2 | 0x20F0 | Specifies window 4's buffer end address register, buffer 2. | 0x0000_0000 |
| VIDW00ADD2 | 0x0100 | Specifies window 0's buffer size register. | 0x0000_0000 |
| VIDW01ADD2 | 0x0104 | Specifies window 1's buffer size register. | 0x0000_0000 |
| VIDW02ADD2 | 0x0108 | Specifies window 2's buffer size register. | 0x0000_0000 |
| VIDW03ADD2 | 0x010C | Specifies window 3's buffer size register. | 0x0000_0000 |
| VIDW04ADD2 | 0x0110 | Specifies window 4's buffer size register. | 0x0000_0000 |
| VIDINTCON0 | 0x0130 | Specifies video interrupt control register. | 0x0000_0000 |
| VIDINTCON1 | 0x0134 | Specifies video interrupt pending register. | 0x0000_0000 |
| W1KEYCON0 | 0x0140 | Specifies color key control register. | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|--------------|--------|---|-------------|
| W1KEYCON1 | 0x0144 | Specifies color key value (transparent value) register. | 0x0000_0000 |
| W2KEYCON0 | 0x0148 | Specifies color key control register. | 0x0000_0000 |
| W2KEYCON1 | 0x014C | Specifies color key value (transparent value) register. | 0x0000_0000 |
| W3KEYCON0 | 0x0150 | Specifies color key control register. | 0x0000_0000 |
| W3KEYCON1 | 0x0154 | Specifies color key value (transparent value) register. | 0x0000_0000 |
| W4KEYCON0 | 0x0158 | Specifies color key control register. | 0x0000_0000 |
| W4KEYCON1 | 0x015C | Specifies color key value (transparent value) register. | 0x0000_0000 |
| W1KEYALPHA | 0x0160 | Specifies color key alpha value register. | 0x0000_0000 |
| W2KEYALPHA | 0x0164 | Specifies color key alpha value register. | 0x0000_0000 |
| W3KEYALPHA | 0x0168 | Specifies color key alpha value register. | 0x0000_0000 |
| W4KEYALPHA | 0x016C | Specifies color key alpha value register. | 0x0000_0000 |
| DITHMODE | 0x0170 | Specifies dithering mode register. | 0x0000_0000 |
| WIN0MAP | 0x0180 | Specifies window 0's color control. | 0x0000_0000 |
| WIN1MAP | 0x0184 | Specifies window 1's color control. | 0x0000_0000 |
| WIN2MAP | 0x0188 | Specifies window 2's color control. | 0x0000_0000 |
| WIN3MAP | 0x018C | Specifies window 3's color control. | 0x0000_0000 |
| WIN4MAP | 0x0190 | Specifies window 4's color control. | 0x0000_0000 |
| WPALCON_H | 0x019C | Specifies window palette control register. | 0x0000_0000 |
| WPALCON_L | 0x01A0 | Specifies window palette control register. | 0x0000_0000 |
| TRIGCON | 0x01A4 | Specifies i80/ RGB trigger control register. | 0x0000_0000 |
| I80IFCONA0 | 0x01B0 | Specifies i80 interface control 0 for main LDI. | 0x0000_0000 |
| I80IFCONA1 | 0x01B4 | Specifies i80 interface control 0 for sub LDI. | 0x0000_0000 |
| I80IFCONB0 | 0x01B8 | Specifies i80 interface control 1 for main LDI. | 0x0000_0000 |
| I80IFCONB1 | 0x01BC | Specifies i80 interface control 1 for sub LDI. | 0x0000_0000 |
| COLORGAINCON | 0x01C0 | Specifies color gain control register. | 0x1004_0100 |
| LDI_CMDCON0 | 0x01D0 | Specifies i80 interface LDI command control 0. | 0x0000_0000 |
| LDI_CMDCON1 | 0x01D4 | Specifies i80 interface LDI command control 1. | 0x0000_0000 |
| SIFCCON0 | 0x01E0 | Specifies LCD i80 system interface command control 0. | 0x0000_0000 |
| SIFCCON1 | 0x01E4 | Specifies LCD i80 system interface command control 1. | 0x0000_0000 |
| SIFCCON2 | 0x01E8 | Specifies LCD i80 system interface command control 2. | 0x????_???? |
| HUECOEF_CR_1 | 0x01EC | Specifies hue coefficient control register. | 0x0100_0100 |
| HUECOEF_CR_2 | 0x01F0 | Specifies hue coefficient control register. | 0x0000_0000 |
| HUECOEF_CR_3 | 0x01F4 | Specifies hue coefficient control register. | 0x0000_0000 |
| HUECOEF_CR_4 | 0x01F8 | Specifies hue coefficient control register. | 0x0100_0100 |
| HUECOEF_CB_1 | 0x01FC | Specifies hue coefficient control register. | 0x0100_0100 |
| HUECOEF_CB_2 | 0x0200 | Specifies hue coefficient control register. | 0x0000_0000 |
| HUECOEF_CB_3 | 0x0204 | Specifies hue coefficient control register. | 0x0000_0000 |

| Register | Offset | Description | Reset Value |
|--|--------|--|-------------|
| HUECOEF_CB_4 | 0x0208 | Specifies hue coefficient control register. | 0x0100_0100 |
| HUEOFFSET | 0x020C | Specifies hue offset control register. | 0x0180_0080 |
| VIDW0ALPHA0 | 0x021C | Specifies window 0's alpha value 0 register. | 0x0000_0000 |
| VIDW0ALPHA1 | 0x0220 | Specifies window 0's alpha value 1 register. | 0x0000_0000 |
| VIDW1ALPHA0 | 0x0224 | Specifies window 1's alpha value 0 register. | 0x0000_0000 |
| VIDW1ALPHA1 | 0x0228 | Specifies window 1's alpha value 1 register. | 0x0000_0000 |
| VIDW2ALPHA0 | 0x022C | Specifies window 2's alpha value 0 register. | 0x0000_0000 |
| VIDW2ALPHA1 | 0x0230 | Specifies window 2's alpha value 1 register. | 0x0000_0000 |
| VIDW3ALPHA0 | 0x0234 | Specifies window 3's alpha value 0 register. | 0x0000_0000 |
| VIDW3ALPHA1 | 0x0238 | Specifies window 3's alpha value 1 register. | 0x0000_0000 |
| VIDW4ALPHA0 | 0x023C | Specifies window 4's alpha value 0 register. | 0x0000_0000 |
| VIDW4ALPHA1 | 0x0240 | Specifies window 4's alpha value 1 register. | 0x0000_0000 |
| BLENDEQ1 | 0x0244 | Specifies window 1's blending equation control register. | 0x0000_00c2 |
| BLENDEQ2 | 0x0248 | Specifies window 2's blending equation control register. | 0x0000_00c2 |
| BLENDEQ3 | 0x024C | Specifies window 3's blending equation control register. | 0x0000_00c2 |
| BLENDEQ4 | 0x0250 | Specifies window 4's blending equation control register. | 0x0000_00c2 |
| BLENDCON | 0x0260 | Specifies blending control register. | 0x0000_0000 |
| W0RTQOSCON | 0x0264 | Specifies window 0's RTQOS control register. | 0x0000_0000 |
| W1RTQOSCON | 0x0268 | Specifies window 1's RTQOS control register. | 0x0000_0000 |
| W2RTQOSCON | 0x026C | Specifies window 2's RTQOS control register. | 0x0000_0000 |
| W3RTQOSCON | 0x0270 | Specifies window 3's RTQOS control register. | 0x0000_0000 |
| W4RTQOSCON | 0x0274 | Specifies window 4's RTQOS control register. | 0x0000_0000 |
| LDI_CMD0 | 0x0280 | Specifies i80 interface LDI command 0. | 0x0000_0000 |
| LDI_CMD1 | 0x0284 | Specifies i80 interface LDI command 1. | 0x0000_0000 |
| LDI_CMD2 | 0x0288 | Specifies i80 interface LDI command 2. | 0x0000_0000 |
| LDI_CMD3 | 0x028C | Specifies i80 interface LDI command 3. | 0x0000_0000 |
| LDI_CMD4 | 0x0290 | Specifies i80 interface LDI command 4. | 0x0000_0000 |
| LDI_CMD5 | 0x0294 | Specifies i80 interface LDI command 5. | 0x0000_0000 |
| LDI_CMD6 | 0x0298 | Specifies i80 interface LDI command 6. | 0x0000_0000 |
| LDI_CMD7 | 0x029C | Specifies i80 interface LDI command 7. | 0x0000_0000 |
| LDI_CMD8 | 0x02A0 | Specifies i80 interface LDI command 8. | 0x0000_0000 |
| LDI_CMD9 | 0x02A4 | Specifies i80 interface LDI command 9. | 0x0000_0000 |
| LDI_CMD10 | 0x02A8 | Specifies i80 interface LDI command 10. | 0x0000_0000 |
| LDI_CMD11 | 0x02AC | Specifies i80 interface LDI command 11. | 0x0000_0000 |
| Gamma LUT Data for 64 Step Mode | | | |
| GAMMALUT_01_00 | 0x037C | Specifies gamma LUT data of the index 0, 1. | 0x0010_0000 |
| GAMMALUT_03_02 | 0x0380 | Specifies gamma LUT data of the index 2, 3. | 0x0030_0020 |

| Register | Offset | Description | Reset Value |
|----------------|--------|---|-------------|
| GAMMALUT_05_04 | 0x0384 | Specifies gamma LUT data of the index 4, 5. | 0x0050_0040 |
| GAMMALUT_07_06 | 0x0388 | Specifies gamma LUT data of the index 6, 7. | 0x0070_0060 |
| GAMMALUT_09_08 | 0x038C | Specifies gamma LUT data of the index 8, 9. | 0x0090_0080 |
| GAMMALUT_11_10 | 0x0390 | Specifies gamma LUT data of the index 10, 11. | 0x00B0_00A0 |
| GAMMALUT_13_12 | 0x0394 | Specifies gamma LUT data of the index 12, 13. | 0x00D0_00C0 |
| GAMMALUT_15_14 | 0x0398 | Specifies gamma LUT data of the index 14, 15. | 0x00F0_00E0 |
| GAMMALUT_17_16 | 0x039C | Specifies gamma LUT data of the index 16, 17. | 0x0110_0100 |
| GAMMALUT_19_18 | 0x03A0 | Specifies gamma LUT data of the index 18, 19. | 0x0130_0120 |
| GAMMALUT_21_20 | 0x03A4 | Specifies gamma LUT data of the index 20, 21. | 0x0150_0140 |
| GAMMALUT_23_22 | 0x03A8 | Specifies gamma LUT data of the index 22, 23. | 0x0170_0160 |
| GAMMALUT_25_24 | 0x03AC | Specifies gamma LUT data of the index 24, 25. | 0x0190_0180 |
| GAMMALUT_27_26 | 0x03B0 | Specifies gamma LUT data of the index 26, 27. | 0x01B0_01A0 |
| GAMMALUT_29_28 | 0x03B4 | Specifies gamma LUT data of the index 28, 29. | 0x01F0_01C0 |
| GAMMALUT_31_30 | 0x03B8 | Specifies gamma LUT data of the index 30, 31. | 0x01F0_01E0 |
| GAMMALUT_33_32 | 0x03BC | Specifies gamma LUT data of the index 32, 33. | 0x0210_0200 |
| GAMMALUT_35_34 | 0x03C0 | Specifies gamma LUT data of the index 34, 35. | 0x0230_0220 |
| GAMMALUT_37_36 | 0x03C4 | Specifies gamma LUT data of the index 36, 37. | 0x0250_0240 |
| GAMMALUT_39_38 | 0x03C8 | Specifies gamma LUT data of the index 38, 39. | 0x0270_0260 |
| GAMMALUT_41_40 | 0x03CC | Specifies gamma LUT data of the index 40, 41. | 0x0290_0280 |
| GAMMALUT_43_42 | 0x03D0 | Specifies gamma LUT data of the index 42, 43. | 0x02B0_02A0 |
| GAMMALUT_45_44 | 0x03D4 | Specifies gamma LUT data of the index 44, 45. | 0x02D0_02C0 |
| GAMMALUT_47_46 | 0x03D8 | Specifies gamma LUT data of the index 46, 47. | 0x02F0_02E0 |
| GAMMALUT_49_48 | 0x03DC | Specifies gamma LUT data of the index 48, 49. | 0x0310_0300 |
| GAMMALUT_51_50 | 0x03E0 | Specifies gamma LUT data of the index 50, 51. | 0x0330_0320 |
| GAMMALUT_53_52 | 0x03E4 | Specifies gamma LUT data of the index 52, 53. | 0x0350_0340 |
| GAMMALUT_55_54 | 0x03E8 | Specifies gamma LUT data of the index 54, 55. | 0x0370_0360 |
| GAMMALUT_57_56 | 0x03EC | Specifies gamma LUT data of the index 56, 57. | 0x0390_0380 |
| GAMMALUT_59_58 | 0x03F0 | Specifies gamma LUT data of the index 58, 59. | 0x03B0_03A0 |
| GAMMALUT_61_60 | 0x03F4 | Specifies gamma LUT data of the index 60, 61. | 0x03D0_03C0 |
| GAMMALUT_63_62 | 0x03F8 | Specifies gamma LUT data of the index 62, 63. | 0x03F0_03E0 |
| GAMMALUT_xx_64 | 0x03FC | Specifies gamma LUT data of the index 64. | 0x0000_0400 |

Gamma LUT Data for 16 Step Mode

| | | | |
|----------------|--------|---|-------------|
| GAMMALUT_R_1_0 | 0X037C | Specifies gamma RED LUT data of the index 0, 1. | 0X0010_0000 |
| GAMMALUT_R_3_2 | 0X0380 | Specifies gamma RED data of the index 2, 3. | 0X0030_0020 |
| GAMMALUT_R_5_4 | 0X0384 | Specifies gamma RED data of the index 4, 5. | 0X0050_0040 |
| GAMMALUT_R_7_6 | 0X0388 | Specifies gamma RED data of the index 6, 7. | 0X0070_0060 |
| GAMMALUT_R_9_8 | 0X038C | Specifies gamma RED data of the index 8, 9. | 0X0090_0080 |

| Register | Offset | Description | Reset Value |
|-------------------------------|--------|---|-------------|
| GAMMALUT_R_11_10 | 0X0390 | Specifies gamma RED data of the index 10, 11. | 0X00B0_00A0 |
| GAMMALUT_R_13_12 | 0X0394 | Specifies gamma RED data of the index 12, 13. | 0X00D0_00C0 |
| GAMMALUT_R_15_14 | 0X0398 | Specifies gamma RED data of the index 14, 15. | 0X00F0_00E0 |
| GAMMALUT_R_16 | 0X039C | Specifies gamma RED data of the index 16. | 0X0110_0100 |
| GAMMALUT_R_1_0 | 0X03A0 | Specifies gamma GREEN LUT data of the index 0, 1. | 0X0130_0120 |
| GAMMALUT_R_3_2 | 0X03A4 | Specifies gamma GREEN data of the index 2, 3. | 0X0150_0140 |
| GAMMALUT_R_5_4 | 0X03A8 | Specifies gamma GREEN data of the index 4, 5. | 0X0170_0160 |
| GAMMALUT_R_7_6 | 0X03AC | Specifies gamma GREEN data of the index 6, 7. | 0X0190_0180 |
| GAMMALUT_R_9_8 | 0X03B0 | Specifies gamma GREEN data of the index 8, 9. | 0X01B0_01A0 |
| GAMMALUT_R_11_10 | 0X03B4 | Specifies gamma GREEN data of the index 10, 11. | 0X01D0_01C0 |
| GAMMALUT_R_13_12 | 0X03B8 | Specifies gamma GREEN data of the index 12, 13. | 0X01F0_01E0 |
| GAMMALUT_R_15_14 | 0X03BC | Specifies gamma GREEN data of the index 14, 15. | 0X0210_0200 |
| GAMMALUT_R_16 | 0X03C0 | Specifies gamma GREEN data of the index 16. | 0X0230_0220 |
| GAMMALUT_R_1_0 | 0X03C4 | Specifies gamma BLUE data of the index 0, 1. | 0X0250_0240 |
| GAMMALUT_R_3_2 | 0X03C8 | Specifies gamma BLUE data of the index 2, 3. | 0X0270_0260 |
| GAMMALUT_R_5_4 | 0X03CC | Specifies gamma BLUE data of the index 4, 5. | 0X0290_0280 |
| GAMMALUT_R_7_6 | 0X03D0 | Specifies gamma BLUE data of the index 6, 7. | 0X02B0_02A0 |
| GAMMALUT_R_9_8 | 0X03D4 | Specifies gamma BLUE data of the index 8, 9. | 0X02D0_02C0 |
| GAMMALUT_R_11_10 | 0X03D8 | Specifies gamma BLUE data of the index 10, 11. | 0X02F0_02E0 |
| GAMMALUT_R_13_12 | 0X03DC | Specifies gamma BLUE data of the index 12, 13. | 0X0310_0300 |
| GAMMALUT_R_15_14 | 0X03E0 | Specifies gamma BLUE data of the index 14, 15. | 0X0330_0320 |
| GAMMALUT_R_16 | 0X03E4 | Specifies gamma BLUE data of the index 16 | 0X0350_0340 |
| RSVD | 0X03E8 | Does not use. | 0X0370_0360 |
| RSVD | 0X03EC | Does not use. | 0X0390_0380 |
| RSVD | 0X03F0 | Does not use. | 0X03B0_03A0 |
| RSVD | 0X03F4 | Does not use. | 0X03D0_03C0 |
| RSVD | 0X03F8 | Does not use. | 0X03F0_03E0 |
| RSVD | 0X03FC | Does not use. | 0X0000_0400 |
| Shadow Windows Control | | | |

| Register | Offset | Description | Reset Value |
|----------------|--------|--|-------------|
| SHD_VIDW00ADD0 | 0x40A0 | Specifies window 0's buffer start address register (shadow). | 0x0000_0000 |
| SHD_VIDW01ADD0 | 0x40A8 | Specifies window 1's buffer start address register (shadow). | 0x0000_0000 |
| SHD_VIDW02ADD0 | 0x40B0 | Specifies window 2's buffer start address register (shadow). | 0x0000_0000 |
| SHD_VIDW03ADD0 | 0x40B8 | Specifies window 3's buffer start address register (shadow). | 0x0000_0000 |
| SHD_VIDW04ADD0 | 0x40C0 | Specifies window 4's buffer start address register (shadow). | 0x0000_0000 |
| SHD_VIDW00ADD1 | 0x40D0 | Specifies window 0's buffer end address register (shadow) | 0x0000_0000 |
| SHD_VIDW01ADD1 | 0x40D8 | Specifies window 1's buffer end address register (shadow) | 0x0000_0000 |
| SHD_VIDW02ADD1 | 0x40E0 | Specifies window 2's buffer end address register (shadow). | 0x0000_0000 |
| SHD_VIDW03ADD1 | 0x40E8 | Specifies window 3's buffer end address register (shadow). | 0x0000_0000 |
| SHD_VIDW04ADD1 | 0x40F0 | Specifies window 4's buffer end address register (shadow). | 0x0000_0000 |
| SHD_VIDW00ADD2 | 0x4100 | Specifies window 0's buffer size register (shadow). | 0x0000_0000 |
| SHD_VIDW01ADD2 | 0x4104 | Specifies window 1's buffer size register (shadow). | 0x0000_0000 |
| SHD_VIDW02ADD2 | 0x4108 | Specifies window 2's buffer size register (shadow). | 0x0000_0000 |
| SHD_VIDW03ADD2 | 0x410C | Specifies window 3's buffer size register (shadow). | 0x0000_0000 |
| SHD_VIDW04ADD2 | 0x4110 | Specifies window 4's buffer size register (shadow). | 0x0000_0000 |

16.5.2 Palette Memory

- Base Address = 0x11C0_0000

| Register | Start Address | End Address | Description | Reset Value |
|-------------|--------------------|--------------------|--|-------------|
| Win0 PalRam | 0x2400 (0x0400) | 0x27FC (0x07FC) | Specifies 0 to 255 entry palette data. | Undefined |
| Win1 PalRam | 0x2800 (0x0800) | 0x2BFC (0x0BFC) | Specifies 0 to 255 entry palette data. | Undefined |
| Win2 PalRam | 0x2C00 | 0x2FFC | Specifies 0 to 255 entry palette data. | Undefined |
| Win3 PalRam | 0x3000 | 0x33FC | Specifies 0 to 255 entry palette data. | Undefined |
| Win4 PalRam | 0x3400 | 0x37FC | Specifies 0 to 255 entry palette data. | Undefined |

16.5.3 Control Register

16.5.3.1 VIDCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|--|-------------|
| RSVD | [31] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| DSI_EN | [30] | RW | Enables MIPI DSI. 0 = Disables 1 = Enables (i80 24-bit data interface, SYS_ADD[1]) | 0 |
| RSVD | [29] | - | Reserve NOTE: This bit should be set to 0. | 0 |
| VIDOUT | [28:26] | RW | Determines output format of Video Controller. 000 = RGB interface 001 = Reserved 010 = Indirect i80 interface for LDI0 011 = Indirect i80 interface for LDI1 100 = Write-Back interface and RGB interface 101 = Reserved 110 = WB Interface and i80 interface for LDI0 111 = WB Interface and i80 interface for LDI1 | 000 |
| L1_DATA16 | [25:23] | RW | Selects output data format mode of indirect i80 interface (LDI1). (VIDOUT[1:0] == 2'b11) 000 = 16-bit mode (16 BPP) 001 = 16 + 2-bit mode (18 BPP) 010 = 9 + 9-bit mode (18 BPP) 011 = 16 + 8-bit mode (24 BPP) 100 = 18-bit mode (18 BPP) 101 = 8 + 8-bit mode (16 BPP) | 000 |
| L0_DATA16 | [22:20] | RW | Selects output data format mode of indirect i80 interface (LDI0). (VIDOUT[1:0] == 2'b10) 000 = 16-bit mode (16 BPP) 001 = 16 + 2-bit mode (18 BPP) 010 = 9 + 9-bit mode (18 BPP) 011 = 16 + 8-bit mode (24 BPP) 100 = 18-bit mode (18 BPP) 101 = 8 + 8-bit mode (16 BPP) | 000 |
| RSVD | [19] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| RGSPSEL | [18] | RW | Selects display mode (VIDOUT[1:0] == 2'b00). 0 = RGB parallel format 1 = RGB serial format Selects the display mode (VIDOUT[1:0] != 2'b00). | 0 |

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| | | | 0 = RGB parallel format | |
| PNRMODE | [17] | RW | Controls inverting RGB_ORDER (atVIDCON3). 0 = Normal: RGBORDER[2] atVIDCON3 1 = Invert: to RGBORDER[2] atVIDCON3 NOTE: You can use this bit for the previous version of FIMD. You do not have to use this bit if you use RGB_ORDER atVIDCON3 register. | 00 |
| CLKVALUP | [16] | RW | Selects CLKVAL_F update timing control. 0 = Always 1 = Start of a frame (only once per frame) | 0 |
| RSVD | [15:14] | - | Reserved | 0 |
| CLKVAL_F | [13:6] | RW | Determines rates of VCLK and CLKVAL[7:0]. $VCLK = FIMD \times SCLK / (CLKVAL + 1)$, where $CLKVAL \geq 1$ NOTE: The maximum frequency of VCLK is 80 MHz. (80 MHz for Display Controller) | 0 |
| VCLKFREE | [5] | RW | Controls VCLK Free Run (only valid at RGB IF mode). 0 = Normal mode (controls using ENVID) 1 = Free-run mode | 0 |
| RSVD | [4:2] | - | Reserved NOTE: This bit should be set to 0. | 0x0 |
| ENVID | [1] | RW | Enables/disables video output and logic immediately. 0 = Disables the video output and display control signal 1 = Enables the video output and display control signal | 0 |
| ENVID_F | [0] | RW | Enables/disables video output and logic at current frame end. 0 = Disables the video output and display control signal 1 = Enables the video output and display control signal If this bit is set to "on" and "off", then "H" is Read and enables the video controller until the end of current frame. (NOTE) | 0 |

NOTE: Display On: ENVID and ENVID_F are set to "1".

Direct Off: ENVID and ENVID_F are set to "0" simultaneously.

Per Frame Off: ENVID_F is set to "0" and ENVID is set to "1".

Caution: 1 = If VIDCON0 is set for Per Frame Off in interlace mode, then the value of INTERLACE_F should be set to "0" in the same time.
2 = If display controller is off using direct off, then it is impossible to turn on the display controller without reset.

16.5.3.2 VIDCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0004, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------------|---------|------|---|-------------|
| LINECNT (read only) | [26:16] | RW | Provides status of the line counter (Read only). Up count from 0 to LINEVAL. | 0 |
| FSTATUS | [15] | RW | Specifies Field Status (Read only). 0 = ODD Field 1 = EVEN Field | 0 |
| VSTATUS | [14:13] | RW | Specifies Vertical Status (Read only). 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch | 0 |
| RSVD | [12:11] | - | Reserved | 0 |
| FIXVCLK | [10:9] | RW | Specifies VCLK hold scheme at data under-flow. 00 = VCLK hold 01 = VCLK running 11 = VCLK running and disables VDEN | 0 |
| RSVD | [8] | - | Reserved | 0 |
| IVCLK | [7] | RW | Controls polarity of the VCLK active edge. 0 = Fetches video data at VCLK falling edge 1 = Fetches video data at VCLK rising edge | 0 |
| IHSYNC | [6] | RW | Specifies HSYNC pulse polarity. 0 = Normal 1 = Inverted | 0 |
| IVSYNC | [5] | RW | Specifies VSYNC pulse polarity. 0 = Normal 1 = Inverted | 0 |
| IVDEN | [4] | RW | Specifies VDEN signal polarity. 0 = Normal 1 = Inverted | 0 |
| RSVD | [3:0] | RW | Reserved | 0x0 |

16.5.3.3 VIDCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [31:28] | – | Reserved | 0 |
| RGB_SKIP_EN | [27] | RW | Enables RGB skip mode. (Only where RGBSEL == 1'b0). 0 = Disables 1 = Enables | 0 |
| RSVD | [26] | – | Reserved | 0 |
| RGB_DUMMY_LOC | [25] | RW | Controls RGB dummy insertion location. (Only where RGBSEL == 1'b1 and RGB_DUMMY_EN == 1'b1) 0 = Last (fourth) position 1 = First position | 0 |
| RGB_DUMMY_EN | [24] | RW | Enables RGB dummy insertion mode. (Only where RGBSEL == 1'b1) 0 = Disables 1 = Enables | 0 |
| RSVD | [23:22] | – | Reserved NOTE: This bit should be set to 0. | 0 |
| RGB_ORDER_E | [21:19] | RW | Controls RGB interface output order. (Even line, line number 2, 4, 6, 8.), where, RGBSEL== 1'b0 000 = RGB 001 = GBR 010 = BRG 100 = BGR 101 = RBG 110 = GRB where, (RGBSEL == 1'b1) or (RGBSEL == 1'b0 and RGB_SKIP_EN = 1'b1) 000 = R → G → B 001 = G → B → R 010 = B → R → G 100 = B → G → R 101 = R → B → G 110 = G → R → B NOTE: PNR0[0] at VIDCON0 should be set to 0, when you use RGB_ORDER_O[2:0] at VIDCON3 register. | 0 |
| RGB_ORDER_O | [18:16] | RW | Controls RGB interface output order (Odd Line, line number 1, 3, 5, 7.), where, RGBSEL == 1'b0 000 = RGB 001 = GBR 010 = BRG 100 = BGR | 0 |

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| | | | <p>101 = RBG 110 = GRB where, (RGBSPSEL == 1'b1) or (RGBSPSEL == 1'b0 and RGB_SKIP_EN = 1'b1) 000 = R → G → B 001 = G → B → R 010 = B → R → G 100 = B → G → R 101 = R → B → G 110 = G → R → B NOTE: PNR0[0] at VIDCON0 should be set to 0, when you use RGB_ORDER_E[2:0] at VIDCON3 register.</p> | |
| RSVD | [15:14] | — | <p>Reserved NOTE: This bit should be set to 1.</p> | 0 |
| TVFORMATSEL | [13:12] | RW | <p>Specifies output format of YUV data. 00 = Reserved 01 = YUV422 1x = YUV444</p> | 0 |
| RSVD | [11:9] | — | Reserved | 0 |
| OrgYCbCr | [8] | RW | <p>Specifies order of YUV data. 0 = Y – CbCr 1 = CbCr – Y</p> | 0 |
| YUVOrd | [7] | RW | <p>Specifies order of Chroma data. 0 = Cb – Cr 1 = Cr – Cb</p> | 0 |
| RSVD | [6:5] | — | Reserved | 0 |
| WB_FRAME_SKIP | [4:0] | RW | <p>Controls WB frame skip rate. The maximum rate is up to 1:30 [only where (VIDOUT[2:0] == 3'b001 or 3'b100 TV encoder interface), (INTERLACE_F == 1'b0) and (TV422 or TVRGB output)]. 00000 = No skip = 1:1 00001 = Skip rate = 1:2 00010 = Skip rate = 1:3 ... 11101 = Skip rate = 1: 0 1111x = Reserved</p> | 0 |

16.5.3.4 VIDCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x000C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------------|---------|------|---|-------------|
| RSVD | [31:21] | — | Reserved NOTE: This bit should be set to 0. | 0 |
| RSVD | [20:19] | — | Reserved | 0 |
| CG_ON | [18] | RW | Enables Control Color Gain. 0 = Disables (bypass) 1 = Enables | 0 |
| RSVD | [17] | — | Reserved | 0 |
| GM_ON | [16] | RW | Enables Control Gamma. 0 = Disables (bypass) 1 = Enables | 0 |
| GM_MODE | [15] | RW | Gamma mode selection 0 = Applies 64 step identical value to all R, G, B data 1 = Applies 16 step independent value to each R, G, B data | 0 |
| HUE_CSC_F_Narrow | [14] | RW | Controls HUE CSC_F Narrow/ Wide. 0 = Wide 1 = Narrow | 0 |
| HUE_CSC_F_EQ709 | [13] | RW | Controls HUE_CSC_F parameter. 0 = Equation. 601 1 = Equation. 709 | 0 |
| HUE_CSC_F_ON | [12] | RW | Enables HUE_CSC_F. 0 = Disables 1 = Enables (when HUE_ON == 1'b1) | 0 |
| RSVD | [11] | — | Reserved. | 0 |
| HUE_CSC_B_Narrow | [10] | RW | Controls HUE CSC_B Narrow/ Wide. 0 = Wide 1 = Narrow | 0 |
| HUE_CSC_B_EQ709 | [9] | RW | Controls HUE_CSC_B parameter. 0 = Equation 601 1 = Equation 709 | 0 |
| HUE_CSC_B_ON | [8] | RW | Enables HUE_CSC_B. 0 = Disables 1 = Enables (when HUE_ON == 1'b1) | 0 |
| HUE_ON | [7] | RW | Enables Control Hue. 0 = Disables (bypass) 1 = Enables | 0 |
| RSVD | [6:2] | — | Reserved NOTE: This bit should be set to 0. | 0 |
| PC_DIR | [1] | RW | Controls Pixel Compensation direction. | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------|-------|------|---|-------------|
| | | | 0 = + 0.5 (positive) 1 = - 0.5 (negative) | |
| PC_ON | [3:0] | RW | Enables Pixel Compensation. 0 = Disables 1 = Enables NOTE: PC_ON == 1'b1 compensates the TV output data. | 0x0 |

16.5.3.5 VIDTCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|---------|------|--|-------------|
| VBPDE | [31:24] | RW | Vertical back porch specifies the number of inactive lines at the start of a frame after vertical synchronization period. (only for even field of YVU interface) | 0x00 |
| VBPD | [23:16] | RW | Vertical back porch specifies the number of inactive lines at the start of a frame after vertical synchronization period. | 0x00 |
| VFPD | [15:8] | RW | Vertical front porch specifies the number of inactive lines at the end of a frame before vertical synchronization period. | 0x00 |
| VSPW | [7:0] | RW | Vertical synchronization pulse width determines the high-level width of VSYNC pulse by counting the number of inactive lines. | 0x00 |

16.5.3.6 VIDTCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|---------|------|---|-------------|
| VFPDE | [31:24] | RW | Vertical front porch specifies the number of inactive lines at the end of a frame before vertical synchronization period. (only for the even field of YVU interface). | 0 |
| HBDP | [23:16] | RW | Horizontal back porch specifies the number of VCLK periods between the falling edge of HSYNC and start of active data. | 0x00 |
| HFPD | [15:8] | RW | Horizontal front porch specifies the number of VCLK periods between the end of active data and rising edge of HSYNC. | 0x00 |
| HSPW | [7:0] | RW | Horizontal synchronization pulse width determines the high-level width of HSYNC pulse by counting the number of VCLK. | 0x00 |

16.5.3.7 VIDTCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0018, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|---|-------------|
| LINEVAL | [21:11] | RW | Determines vertical size of display. In the Interlace mode, (LINEVAL + 1) should be even. | 0 |
| HOZVAL | [10:0] | RW | Determines horizontal size of display. | 0 |

NOTE: HOZVAL = (Horizontal display size) – 1 and LINEVAL = (Vertical display size) – 1.

16.5.3.8 VIDTCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| VSYNCEN | [31] | RW | Enables VSYNC Signal Output. 0 = Disables 1 = Enables VBPD (VFPD, VSPW) + 1 < LINEVAL (when VSYNCEN = 1) | 0 |
| RSVD | [30] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| FRMEN | [29] | RW | Enables FRM signal output. 0 = Disables 1 = Enables | 0 |
| INVFRM | [28] | RW | Controls polarity of FRM pulse. 0 = Active HIGH 1 = Active LOW | 0 |
| FRMV RATE | [27:24] | RW | Controls FRM issue rate (maximum rate up to 1:16). | 0x00 |
| RSVD | [23:16] | RW | Reserved | 0x00 |
| FRMV FPD | [15:8] | RW | Specifies number of line between data active and FRM signal. | 0x00 |
| FRMV SPW | [7:0] | RW | Specifies number of line of FRM signal width. (FRMV FPD + 1) + (FRMV SPW + 1) < LINEVAL + 1 (in RGB) | 0x00 |

16.5.3.9 WINCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0020, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| BUFSTATUS_H | [31] | RW | Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L} 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 | 0 |
| BUFSEL_H | [30] | RW | Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L} 00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available where BUF_MODE == 1'b1) | 0 |
| LIMIT_ON | [29] | RW | Enables CSC source limiter (for clamping xvYCC source). 0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1) | 0 |
| EQ709 | [28] | RW | Controls CSC parameter. 0 = Equation. 601 1 = Equation. 709 (when local SRC data has HD (709) color gamut) | 0 |
| nWide/Narrow | [27:26] | RW | Chooses color space conversion equation from YCbCr to RGB according to input value range (2'00 for YCbCr w wide range and 2'11 for YCbCr narrow range) Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y:235-16, Cb/Cr:240-16 | 00 |
| TRGSTATUS | [25] | RW | Specifies Trigger Status (read only). 0 = Does not issue trigger 1 = Issues trigger | 0 |
| RSVD | [24:23] | - | Reserved. | 00 |
| ENLOCAL_F | [22] | RW | Selects Data access method. 0 = Dedicated DMA 1 = Local Path | 0 |
| BUFSTATUS_L | [21] | RW | Specifies Buffer Status (read only). NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L} | 0 |
| BUFSEL_L | [20] | RW | Selects Buffer set. NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L} | 0 |
| BUFAUTOEN | [19] | RW | Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto Changed by Trigger Input | 0 |
| BITSWP_F | [18] | RW | Specifies Bit swap control bit. 0 = Disables swap | 0 |

| Name | Bit | Type | Description | Reset Value |
|-----------|---------|------|---|-------------|
| | | | 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1. | |
| BYTSPW_F | [17] | RW | Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1. | 0 |
| HAWSPW_F | [16] | RW | Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1. | 0 |
| WSWP_F | [15] | RW | Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: It should be set to 0 when ENLOCAL is 1. | 0 |
| BUF_MODE | [14] | RW | Selects auto-buffering mode. 0 = Double 1 = Triple | 0 |
| InRGB | [13] | RW | Specifies input color space of source image. (Only for "ENLOCAL" enable). 0 = RGB 1 = YCbCr | 0 |
| RSVD | [12:11] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| BURSTLEN | [10:9] | RW | Selects DMA Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst | 0 |
| RSVD | [8:7] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| BLD_PIX_F | [6] | RW | Selects blending category (In case of window0, this is required only for deciding window 0's blending factor.) 0 = Per plane blending 1 = Per pixel blending | 0 |
| BPPMODE_F | [5:2] | RW | Selects Bits Per Pixel (BPP) mode for Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|--|-------------|
| | | | <p>1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6)</p> <p>1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8)</p> <p>1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7)</p> <p>1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8)</p> <p>1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4)</p> <p>1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5)</p> <p>NOTE:</p> <p>1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending.</p> <p>2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)</p> | |
| ALPHA_SEL_F | [1] | RW | <p>Selects Alpha value.</p> <p>When per plane blending case (BLD_PIX == 0):</p> <p>0 = Uses ALPHA0_R/G/B values</p> <p>1 = Uses ALPHA1_R/G/B values</p> <p>When per pixel blending (BLD_PIX == 1):</p> <p>0 = Selected by AEN (A value)</p> <p>1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101)</p> <p>DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)</p> | 0 |
| ENWIN_F | [0] | RW | <p>Enables/disables video output and logic immediately.</p> <p>0 = Disables the video output and video control signal</p> <p>1 = Enables the video output and video control signal</p> | 0 |

16.5.3.10 WINCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0024, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| BUFSIZE_H | [31] | RW | <p>Specifies Buffer Status (read only).</p> <p>00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2</p> <p>NOTE: BUFSIZE = {BUFSIZE_H, BUFSIZE_L}</p> | 0 |
| BUFSEL_H | [30] | RW | <p>Select Buffer set.</p> <p>00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available when BUF_MODE == 1'b1)</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | 0 |
| LIMIT_ON | [29] | RW | <p>Enables Control CSC source limiter (for clamping xvYCC source).</p> <p>0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1)</p> | 0 |
| EQ709 | [28] | RW | <p>Controls CSC parameter.</p> <p>0 = Equation. 601 1 = Equation. 709 (when local SRC data has HD (709) color gamut)</p> | 0 |
| nWide/Narrow | [27:26] | RW | <p>Chooses color space conversion equation from YCbCr to RGB based on input value range (2'00 for YCbCr wide range and 2'11 for YCbCr narrow range).</p> <p>Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y:235-16, Cb/Cr: 240-16</p> | 00 |
| TRGSTATUS | [25] | RW | <p>Specifies Window 0 Software Trigger Update Status (read only).</p> <p>0 = Updates 1 = Does not update</p> <p>If the Software Trigger in window 1 occurs, then this bit is automatically set to "1". It clears this value only after updating the shadow register sets.</p> | 0 |
| RSVD | [24:23] | - | <p>Reserved</p> <p>NOTE: This bit should be set to 0.</p> | 0 |
| ENLOCAL_F | [22] | RW | <p>Selects Data access method.</p> <p>0 = Dedicated DMA 1 = Local Path</p> | 0 |
| BUFSIZE_L | [21] | RW | <p>Specifies Buffer Status (Read only).</p> <p>NOTE: BUFSIZE = {BUFSIZE_H, BUFSIZE_L}</p> | 0 |
| BUFSEL_L | [20] | RW | Selects Buffer set. | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| | | | NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L} | |
| BUFAUTOEN | [19] | RW | Specifies Double Buffer Auto control bit. 0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input | 0 |
| BITSWP_F | [18] | RW | Specifies Bit swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| BYTSPW_F | [17] | RW | Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| HAWSPW_F | [16] | RW | Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| WSWP_F | [15] | RW | Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| BUF_MODE | [14] | RW | Selects auto-buffering mode. 0 = Double 1 = Triple | 0 |
| InRGB | [13] | RW | Indicates input color space of source image. (Only for "EnLcal" enable). 0 = RGB 1 = YCbCr | 0 |
| RSVD | [12:11] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| BURSTLEN | [10:9] | RW | Specifies DMA's Burst Maximum Length selection. 00 = 16word-burst 01 = 8word-burst 10 = 4word-burst | 0 |
| RSVD | [8] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| ALPHA_MUL_F | [7] | RW | Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-------|------|---|-------------|
| BLD_PIX_F | [6] | RW | Selects blending category. 0 = Per plane blending 1 = Per pixel blending | 0 |
| BPPMODE_F | [5:2] | RW | Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending) | 0 |
| ALPHA_SEL_F | [1] | RW | Selects Alpha value. When per plane blending case (BLD_PIX == 0) 0 = Using ALPHA0_R/G/B values 1 = Using ALPHA1_R/G/B values When per pixel blending (BLD_PIX == 1) 0 = Selected by AEN (A value) 1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110) | 0 |
| ENWIN_F | [0] | RW | Enables/disables video output and logic immediately. 0 = Disables the video output and video control signal 1 = Enables the video output and video control signal | 0 |

16.5.3.11 WINCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0028, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| BUFSTATUS_H | [31] | RW | <p>Specifies Buffer Status (Read only).</p> <p>00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | 0 |
| BUFSEL_H | [30] | RW | <p>Selects Buffer set.</p> <p>00 = Buffer set to 0 01 = Buffer set to 1 10 = Buffer set to 2 (only available when BUF_MODE == 1'b1)</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | 0 |
| LIMIT_ON | [29] | RW | <p>Enables CSC source limiter (for clamping xvYCC source).</p> <p>0 = Disables 1 = Enables (when local SRC data has xvYCC color space, InRGB = 1)</p> | 0 |
| EQ709 | [28] | RW | <p>Controls CSC parameter.</p> <p>0 = Equation.601 1 = Equation 709 (when local SRC data has HD (709) color gamut)</p> | 0 |
| nWide/Narrow | [27:26] | RW | <p>Chooses color space conversion equation from YCbCr to RGB based on the input value range (2'00 for YCbCr wide range and 2'11 for YCbCr narrow range).</p> <p>Wide Range: Y/Cb/Cr: 255-0 Narrow Range: Y: 235-16, Cb/Cr: 240-16</p> | 00 |
| RSVD | [25:24] | - | Reserved | 00 |
| LOCALSEL_F | [23] | RW | <p>Selects local path source.</p> <p>0 = CAMIF2 1 = CAMIF3</p> | 0 |
| ENLOCAL_F | [22] | RW | <p>Selects Data access method.</p> <p>0 = Dedicated DMA 1 = Local Path</p> | 0 |
| BUFSTATUS_L | [21] | RW | <p>Specifies Buffer Status (read only).</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | |
| BUFSEL_L | [20] | RW | <p>Selects Buffer set.</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | 0 |
| BUFAUTOEN | [19] | RW | <p>Specifies Double Buffer Auto control bit.</p> <p>0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input</p> | 0 |
| BITSWP_F | [18] | RW | <p>Specifies the Bit swap control bit.</p> <p>0 = Disables swap</p> | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| | | | 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | |
| BYTSPW_F | [17] | RW | Specifies Byte swaps control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| HAWSPW_F | [16] | RW | Specifies Half-Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 0 when ENLOCAL is 1. | 0 |
| WSWP_F | [15] | RW | Specifies Word swap control bit. 0 = Disables swap 1 = Enables swap NOTE: Set it to 0 when ENLOCAL is 1. | 0 |
| BUF_MODE | [14] | RW | Selects auto-buffering mode. 0 = Double 1 = Triple | 0 |
| InRGB | [13] | RW | Specifies input color space of source image (only for "EnLcal" enable). 0 = RGB 1 = YCbCr | 0 |
| RSVD | [12:11] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| BURSTLEN | [10:9] | RW | Selects the DMA's Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst | 0 |
| RSVD | [8] | - | Reserved (should be 0). | 0 |
| ALPHA_MUL_F | [7] | RW | Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B | 0 |
| BLD_PIX_F | [6] | RW | Selects blending category. 0 = Per plane blending 1 = Per pixel blending | 0 |
| BPPMODE_F | [5:2] | RW | Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|--|-------------|
| | | | <p>0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5)</p> <p>NOTE:</p> <p>1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)</p> | |
| ALPHA_SEL_F | [1] | RW | <p>Selects Alpha value.</p> <p>When Per plane blending case BLD_PIX == 0: 0 = Using ALPHA0_R/G/B values 1 = Using ALPHA1_R/G/B values</p> <p>When Per pixel blending BLD_PIX == 1: 0 = Selected by AEN (A value) 1 = Using DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)</p> | 0 |
| ENWIN_F | [0] | RW | <p>Enables/disables the video output and logic immediately.</p> <p>0 = Disables the video output and video control signal 1 = Enables the video output and video control signal</p> | 0 |

16.5.3.12 WINCON3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x002C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| BUFSTATUS_H | [31] | RW | <p>Specifies Buffer Status (read only).</p> <p>00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | — |
| BUFSEL_H | [30] | RW | <p>Selects Buffer set</p> <p>00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 (only available where BUF_MODE == 1'b1)</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | — |
| RSVD | [29:26] | — | <p>Reserved</p> <p>NOTE: This bit should be set to 0.</p> | — |
| TRIGSTATUS | [25] | RW | <p>Specifies Trigger Status (read only)</p> <p>0 = No trigger is issued 1 = Trigger is issued</p> | — |
| RSVD | [24:22] | — | <p>Reserved</p> <p>NOTE: This bit should be set to 0.</p> | — |
| BUFSTATUS_L | [21] | RW | <p>Specifies Buffer Status (read only).</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | — |
| BUFSEL_L | [20] | RW | <p>Selects Buffer set.</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | — |
| BUFAUTOEN | [19] | RW | <p>Specifies Double Buffer Auto control bit.</p> <p>0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input</p> | — |
| BITSWP_F | [18] | RW | <p>Specifies Bit swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| BYTSPW_F | [17] | RW | <p>Specifies Byte swaps control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| HAWSPW_F | [16] | RW | <p>Specifies Half-Word swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| WSWP_F | [15] | RW | <p>Specifies Word swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | |
| BUF_MODE | [14] | RW | <p>Selects auto-buffering mode.</p> <p>0 = Double</p> | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| | | | 1 = Triple | |
| RSVD | [13:11] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| BURSTLEN | [10:9] | RW | Selects DMA Burst Maximum Length. 00 = 16 word- burst 01 = 8 word- burst 10 = 4 word- burst | 0 |
| RSVD | [8] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| ALPHA_MUL_F | [7] | RW | Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE. Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B | 0 |
| BLD_PIX_F | [6] | RW | Selects blending category. 0 = Per plane blending 1 = Per pixel blending | |
| BPPMODE_F | [5:2] | RW | Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized R:5-G:5-B:5) NOTE: 1. 1101 = Supports unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Supports 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending) | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|---|-------------|
| ALPHA_SEL_F | [1] | RW | <p>Selects Alpha value.</p> <p>When Per plane blending case BLD_PIX == 0: 0 = Uses ALPHA0_R/G/B values 1 = Uses ALPHA1_R/G/B values</p> <p>When Per pixel blending BLD_PIX == 1: 0 = Selected by AEN (A value) 1 = Uses DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)</p> | 0 |
| ENWIN_F | [0] | RW | <p>Enables/disables video output and logic immediately.</p> <p>0 = Disables the video output and video control signal 1 = Enables the video output and video control signal</p> | 0 |

16.5.3.13 WINCON4

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0030, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| BUFSTATUS_H | [31] | RW | <p>Specifies Buffer Status (read only).</p> <p>00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | 0 |
| BUFSEL_H | [30] | RW | <p>Selects Buffer set.</p> <p>00 = Buffer is set to 0 01 = Buffer is set to 1 10 = Buffer is set to 2 (only available where BUF_MODE == 1'b1)</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | 0 |
| RSVD | [29:26] | - | <p>Reserved</p> <p>NOTE: This bit should be set to 0.</p> | 0 |
| TRIGSTATUS | [25] | RW | <p>Specifies Trigger Status (read only).</p> <p>0 = Does not issue trigger 1 = Issues trigger</p> | 0 |
| RSVD | [24:22] | - | <p>Reserved</p> <p>NOTE: This bit should be set to 0.</p> | 0 |
| BUFSTATUS_L | [21] | RW | <p>Specifies Buffer Status (read only).</p> <p>NOTE: BUFSTATUS = {BUFSTATUS_H, BUFSTATUS_L}</p> | 0 |
| BUFSEL_L | [20] | RW | <p>Selects Buffer set.</p> <p>NOTE: BUFSEL = {BUFSEL_H, BUFSEL_L}</p> | 0 |
| BUFAUTOEN | [19] | RW | <p>Specifies Double Buffer Auto control bit.</p> <p>0 = Fixed by BUFSEL 1 = Auto changed by Trigger Input</p> | 0 |
| BITSWP_F | [18] | RW | <p>Specifies Bit swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| BYTSPW_F | [17] | RW | <p>Specifies Byte swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| HAWSPW_F | [16] | RW | <p>Specifies Half-Word swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| WSWP_F | [15] | RW | <p>Specifies Word swap control bit.</p> <p>0 = Disables swap 1 = Enables swap</p> | 0 |
| BUF_MODE | [14] | RW | Selects auto-buffering mode. 0 = Double | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| | | | 1 = Triple | |
| RSVD | [13:11] | - | Reserved NOTE: This bit should be set to 0 | 0 |
| BURSTLEN | [10:9] | RW | Selects DMA Burst Maximum Length. 00 = 16 word-burst 01 = 8 word-burst 10 = 4 word-burst | 0 |
| RSVD | [8] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| ALPHA_MUL_F | [7] | RW | Specifies Multiplied Alpha value mode. 0 = Disables multiplied mode 1 = Enables multiplied mode When ALPHA_MUL is 1, set BLD_PIX = 1, ALPHA_SEL = 1, and BPPMODE_F[5:2] = 4'b1101 or 4'b1110. NOTE: Alpha value = alpha_pixel (from data) × ALPHA0_R/G/B | 0 |
| BLD_PIX_F | [6] | RW | Selects blending category. 0 = Per plane blending 1 = Per pixel blending | 0 |
| BPPMODE_F | [5:2] | RW | Selects Bits Per Pixel (BPP) mode in Window image. 0000 = 1 BPP 0001 = 2 BPP 0010 = 4 BPP 0011 = 8 BPP (palletized) 0100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 BPP (non-palletized, R:5-G:6-B:5) 0110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18 BPP (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18 BPP (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19 BPP (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24 BPP (non-palletized, R:8-G:8-B:8) 1100 = Unpacked 24 BPP (non-palletized, A:1-R:8-G:8-B:7) 1101 = Unpacked 25 BPP (non-palletized, A:1-R:8-G:8-B:8) 1110 = Unpacked 13 BPP (non-palletized, A:1-R:4-G:4-B:4) 1111 = Unpacked 15 BPP (non-palletized, R:5-G:5-B:5) NOTE: 1. 1101 = Support unpacked 32 BPP (non-palletized, A:8-R:8-G:8-B:8) for per pixel blending. 2. 1110 = Support 16 BPP (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending) | 0 |

| Name | Bit | Type | Description | Reset Value |
|-------------|-----|------|--|-------------|
| ALPHA_SEL_F | [1] | RW | <p>Selects Alpha value.</p> <p>When Per plane blending case BLD_PIX == 0 :</p> <p>0 = Uses ALPHA0_R/G/B values 1 = Uses ALPHA1_R/G/B values</p> <p>When Per pixel blending BLD_PIX == 1 :</p> <p>0 = Selected by AEN (A value) 1 = Uses DATA[31:24] data in word boundary (only when BPPMODE_F = 4'b1101) DATA[31:28], [15:12] data in word boundary (only when BPPMODE_F = 4'b1110)</p> | 0 |
| ENWIN_F | [0] | RW | <p>Enables/disables video output and logic immediately.</p> <p>0 = Disables the video output and video control signal 1 = Enables the video output and video control signal</p> | 0 |

16.5.3.14 SHADOWCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0034, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------------|---------|------|---|-------------|
| RSVD | [31:15] | – | Reserved NOTE: This bit should be set to 0 | 0 |
| W4_SHADOW _PROTECT | [14] | RW | Protects to update window 4's shadow register (xxx_F). 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0) | 0 |
| W3_SHADOW _PROTECT | [13] | RW | Protects to update window 3's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0) | 0 |
| W2_SHADOW _PROTECT | [12] | RW | Protects to update window 2's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (updates shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0) | 0 |
| W1_SHADOW _PROTECT | [11] | RW | Protects to update window 1's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (update shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0) | 0 |
| W0_SHADOW _PROTECT | [10] | RW | Protects to update window 0's shadow register (xxx_F) 0 = Updates shadow register per frame 1 = Protects to update (update shadow register at next frame after "SHADOW_PROTECT" turns to be 1'b0) | 0 |
| RSVD | [9:8] | – | Reserved | 0 |
| C2_ENLOCAL_F | [7] | RW | Enables Channel 2 Local Path. 0 = Disables 1 = Enables | 0 |
| C1_ENLOCAL_F | [6] | RW | Enables Channel 1 Local Path. 0 = Disables 1 = Enables | 0 |
| C0_ENLOCAL_F | [5] | RW | Enables Channel 0 Local Path. 0 = Disables 1 = Enables | 0 |
| C4_EN_F | [4] | RW | Enables Channel 4. 0 = Disables 1 = Enables | 0 |
| C3_EN_F | [3] | RW | Enables Channel 3. 0 = Disables 1 = Enables | 0 |
| C2_EN_F | [2] | RW | Enables Channel 2. 0 = Disables | 0 |

| Name | Bit | Type | Description | Reset Value |
|---------|-----|------|---|-------------|
| | | | 1 = Enables 0 = Disables | |
| C1_EN_F | [1] | RW | Enables Channel 1. 0 = Disables 1 = Enables | 0 |
| C0_EN_F | [0] | RW | Enables Channel 0. 0 = Disables 1 = Enables | 0 |

16.5.3.15 WINCHMAP2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x003C, Reset Value = 0x7D51_7D51

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| CH4FISEL | [30:28] | RW | Selects Channel 4's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4 | 111 |
| CH3FISEL | [27:25] | RW | Selects Channel 3's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4 | 110 |
| CH2FISEL | [24:22] | RW | Selects Channel 2's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4 | 101 |
| CH1FISEL | [21:19] | RW | Selects Channel 1's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4 | 010 |
| CH0FISEL | [18:16] | RW | Selects Channel 0's channel. 001 = Window 0 010 = Window 1 101 = Window 2 110 = Window 3 111 = Window 4 | 001 |
| W4FISEL | [14:12] | RW | Selects Window 4's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4 | 111 |
| W3FISEL | [11:9] | RW | Selects Window 3's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4 | 110 |

| Name | Bit | Type | Description | Reset Value |
|---------|-------|------|--|-------------|
| W2FISEL | [8:6] | RW | Selects Window 2's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4 | 101 |
| W1FISEL | [5:3] | RW | Selects Window 1's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4 | 010 |
| W0FISEL | [2:0] | RW | Selects Window 0's channel. 001 = Channel 0 010 = Channel 1 101 = Channel 2 110 = Channel 3 111 = Channel 4 | 001 |

16.5.3.16 VIDOSD0A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0040, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| OSD_LeftTopX_F | [21:11] | RW | Specifies the horizontal screen coordinate for left top pixel of OSD image. | 0 |
| OSD_LeftTopY_F | [10:0] | RW | Specifies the vertical screen coordinate for left top pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen y coordinate. The original screen y coordinate should be even.) | 0 |

16.5.3.17 VIDOSD0B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0044, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| OSD_RightBotX_F | [21:11] | RW | Specifies horizontal screen coordinate for right bottom pixel of OSD image. | 0 |
| OSD_RightBotY_F | [10:0] | RW | Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen y coordinate. The original screen y coordinate should be odd value.) | 0 |

NOTE: Registers should have word boundary X position.

Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3....)

16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6....)

8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12....)

16.5.3.18 VIDOSD0C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0048, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|---|-------------|
| RSVD | [25:24] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| OSDSIZE | [23:0] | RW | Specifies the Window Size For example, Height × Width (number of word) | 0 |

16.5.3.19 VIDOSD0C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0050, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| OSD_LeftTopX_F | [21:11] | RW | Specifies Horizontal screen coordinate for left top pixel of OSD image. | 0 |
| OSD_LeftTopY_F | [10:0] | RW | Specifies Vertical screen coordinate for left top pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even.) | 0 |

16.5.3.20 VIDOSD1B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0054, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| OSD_RightBotX_F | [21:11] | RW | Specifies horizontal screen coordinate for right bottom pixel of OSD image. | 0 |
| OSD_RightBotY_F | [10:0] | RW | Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.) | 0 |

NOTE: Registers should have word boundary X position.

Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3....)

16 BPP mode should have X position by 2pixel. (For example, X = 0, 2, 4, 6....)

8 BPP mode should have X position by 4pixel. (For example, X = 0, 4, 8, 12....)

16.5.3.21 VIDOSD1C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0058, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [24] | – | Reserved | 0 |
| ALPHA0_R_H_F | [23:20] | RW | Specifies Red Alpha upper value (case AEN == 0) | 0 |
| ALPHA0_G_H_F | [19:16] | RW | Specifies Green Alpha upper value (case AEN == 0) | 0 |
| ALPHA0_B_H_F | [15:12] | RW | Specifies Blue Alpha upper value (case AEN == 0) | 0 |
| ALPHA1_R_H_F | [11:8] | RW | Specifies Red Alpha upper value (case AEN == 1) | 0 |
| ALPHA1_G_H_F | [7:4] | RW | Specifies Green Alpha upper value (case AEN == 1) | 0 |
| ALPHA1_B_H_F | [3:0] | RW | Specifies Blue Alpha upper value (case AEN == 1) | 0 |

NOTE: For more information, refer to VIDW1ALPHA0, 1 register.

16.5.3.22 VIDOSD1D

- Base Address = 0x11C0_0000
- Address = Base Address + 0x005C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|--|-------------|
| RSVD | [25:24] | – | Reserved NOTE: This bit should be set to 0. | 0 |
| OSDSIZE | [23:0] | RW | Specifies Window Size. For example, Height × Width (number of word) | 0 |

16.5.3.23 VIDOSD2A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0060, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| OSD_LeftTopX_F | [21:11] | RW | Specifies horizontal screen coordinate for left top pixel of OSD image. | 0 |
| OSD_LeftTopY_F | [10:0] | RW | Specifies vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value. | 0 |

16.5.3.24 VIDOSD2B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0064, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| OSD_RightBotX_F | [21:11] | RW | Specifies horizontal screen coordinate for right bottom pixel of OSD image. | 0 |
| OSD_RightBotY_F | [10:0] | RW | Specifies vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.) | 0 |

NOTE: Registers should have word boundary X position.

Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3....)

16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6....)

8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12....)

16.5.3.25 VIDOSD2C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0068, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | - | Reserved | 0 |
| ALPHA0_R_H_F | [23:20] | RW | Specifies the Red Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_G_H_F | [19:16] | RW | Specifies the Green Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_B_H_F | [15:12] | RW | Specifies the Blue Alpha upper value (case AEN == 0). | 0 |
| ALPHA1_R_H_F | [11:8] | RW | Specifies the Red Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_G_H_F | [7:4] | RW | Specifies the Green Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_B_H_F | [3:0] | RW | Specifies the Blue Alpha upper value (case AEN == 1). | 0 |

NOTE: For more information, refer to VIDW2ALPHA0, 1 register.

16.5.3.26 VIDOSD2D

- Base Address = 0x11C0_0000
- Address = Base Address + 0x006C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|--|-------------|
| RSVD | [25:24] | — | Reserved NOTE: This bit should be set to 0. | 0 |
| OSDSIZE | [23:0] | RW | Specifies Window Size For example, Height × Width(Number of Word) | 0 |

16.5.3.27 VIDOSD3A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0070, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| OSD_LeftTopX_F | [21:11] | RW | Specifies Horizontal screen coordinate for left top pixel of OSD image. | 0 |
| OSD_LeftTopY_F | [10:0] | RW | Specifies Vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value. | 0 |

16.5.3.28 VIDOSD3B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0074, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| OSD_RightBotX_F | [21:11] | RW | Specifies Horizontal screen coordinate for right bottom pixel of OSD image. | 0 |
| OSD_RightBotY_F | [10:0] | RW | Specifies Vertical screen coordinate for right bottom pixel of OSD image. (For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value.) | 0 |

NOTE: Registers should have word boundary X position.

Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3....)

16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6....)

8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12....)

16.5.3.29 VIDOSD3C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0078, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| ALPHA0_R_H_F | [23:20] | RW | Specifies the Red Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_G_H_F | [19:16] | RW | Specifies the Green Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_B_H_F | [15:12] | RW | Specifies the Blue Alpha upper value (case AEN == 0). | 0 |
| ALPHA1_R_H_F | [11:8] | RW | Specifies the Red Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_G_H_F | [7:4] | RW | Specifies the Green Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_B_H_F | [3:0] | RW | Specifies the Blue Alpha upper value (case AEN == 1). | 0 |

NOTE: For more information, Refer to [16.5.3.76 VIDW3ALPHA0](#) [16.5.3.77 VIDW3ALPHA1](#) register.

16.5.3.30 VIDOSD4A

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0080, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------------|---------|------|--|-------------|
| OSD_LeftTopX_F | [21:11] | RW | Specifies the Horizontal screen coordinate for left top pixel of OSD image. | 0 |
| OSD_LeftTopY_F | [10:0] | RW | Specifies the Vertical screen coordinate for left top pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be even value. | 0 |

16.5.3.31 VIDOSD4B

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0084, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------|---------|------|---|-------------|
| OSD_RightBotX_F | [21:11] | RW | Specifies Horizontal screen coordinate for right bottom pixel of OSD image. | 0 |
| OSD_RightBotY_F | [10:0] | RW | Specifies Vertical screen coordinate for right bottom pixel of OSD image. For interlace TV output, this value should be set to half of the original screen "y" coordinate. The original screen "y" coordinate should be odd value. | 0 |

NOTE: Registers should have word boundary X position.

Therefore, 24 BPP mode should have X position by 1 pixel. (For example, X = 0, 1, 2, 3....)

16 BPP mode should have X position by 2 pixel. (For example, X = 0, 2, 4, 6....)

8 BPP mode should have X position by 4 pixel. (For example, X = 0, 4, 8, 12....)

16.5.3.32 VIDOSD4C

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0088, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| ALPHA0_R_H_F | [23:20] | RW | Specifies the Red Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_G_H_F | [19:16] | RW | Specifies the Green Alpha upper value (case AEN == 0). | 0 |
| ALPHA0_B_H_F | [15:12] | RW | Specifies the Blue Alpha upper value (case AEN == 0). | 0 |
| ALPHA1_R_H_F | [11:8] | RW | Specifies the Red Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_G_H_F | [7:4] | RW | Specifies the Green Alpha upper value (case AEN == 1). | 0 |
| ALPHA1_B_H_F | [3:0] | RW | Specifies the Blue Alpha upper value (case AEN == 1). | 0 |

NOTE: For more information, Refer to [16.5.3.78 VIDW4ALPHA0](#) [16.5.3.79 VIDW4ALPHA1](#) register

16.5.3.33 VIDW0n (n = 00 to 04) ADD0Bn (n = 0 to 2)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x00A0, Reset Value = 0x0000_0000 (VIDW00ADD0B0)
- Address = Base Address + 0x00A4, Reset Value = 0x0000_0000 (VIDW00ADD0B1)
- Address = Base Address + 0x20A0, Reset Value = 0x0000_0000 (VIDW00ADD0B2)
- Address = Base Address + 0x00A8, Reset Value = 0x0000_0000 (VIDW01ADD0B0)
- Address = Base Address + 0x00AC, Reset Value = 0x0000_0000 (VIDW01ADD0B1)
- Address = Base Address + 0x20A8, Reset Value = 0x0000_0000 (VIDW01ADD0B2)
- Address = Base Address + 0x00B0, Reset Value = 0x0000_0000 (VIDW02ADD0B0)
- Address = Base Address + 0x00B4, Reset Value = 0x0000_0000 (VIDW02ADD0B1)
- Address = Base Address + 0x20B0, Reset Value = 0x0000_0000 (VIDW02ADD0B2)
- Address = Base Address + 0x00B8, Reset Value = 0x0000_0000 (VIDW03ADD0B0)
- Address = Base Address + 0x00BC, Reset Value = 0x0000_0000 (VIDW03ADD0B1)
- Address = Base Address + 0x20B8, Reset Value = 0x0000_0000 (VIDW03ADD0B2)
- Address = Base Address + 0x00C0, Reset Value = 0x0000_0000 (VIDW04ADD0B0)
- Address = Base Address + 0x00C4, Reset Value = 0x0000_0000 (VIDW04ADD0B1)
- Address = Base Address + 0x20C0, Reset Value = 0x0000_0000 (VIDW04ADD0B2)

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|--|-------------|
| VBASEU_F | [31:0] | RW | Specifies A[31:0] of the start address for video frame buffer. | 0 |

16.5.3.34 VIDW0n (n = 00 to 04) ADD1Bn (n = 0 to 2)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x00D0, Reset Value = 0x0000_0000 (VIDW00ADD1B0)
- Address = Base Address + 0x00D4, Reset Value = 0x0000_0000 (VIDW00ADD1B1)
- Address = Base Address + 0x20D0, Reset Value = 0x0000_0000 (VIDW00ADD1B2)
- Address = Base Address + 0x00D8, Reset Value = 0x0000_0000 (VIDW01ADD1B0)
- Address = Base Address + 0x00DC, Reset Value = 0x0000_0000 (VIDW01ADD1B1)
- Address = Base Address + 0x20D8, Reset Value = 0x0000_0000 (VIDW01ADD1B2)
- Address = Base Address + 0x00E0, Reset Value = 0x0000_0000 (VIDW02ADD1B0)
- Address = Base Address + 0x00E4, Reset Value = 0x0000_0000 (VIDW02ADD1B1)
- Address = Base Address + 0x20E0, Reset Value = 0x0000_0000 (VIDW02ADD1B2)
- Address = Base Address + 0x00E8, Reset Value = 0x0000_0000 (VIDW03ADD1B0)
- Address = Base Address + 0x00EC, Reset Value = 0x0000_0000 (VIDW03ADD1B1)
- Address = Base Address + 0x20E8, Reset Value = 0x0000_0000 (VIDW03ADD1B2)
- Address = Base Address + 0x00F0, Reset Value = 0x0000_0000 (VIDW04ADD1B0)
- Address = Base Address + 0x00F4, Reset Value = 0x0000_0000 (VIDW04ADD1B1)
- Address = Base Address + 0x20F0, Reset Value = 0x0000_0000 (VIDW04ADD1B2)

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| VBASEL_F | [31:0] | RW | Specifies A[31:0] of the end address for video frame buffer. VBASEL = VBASEU + (PAGEWIDTH + OFFSIZE) × (LINEVAL + 1) | 0x0 |

16.5.3.35 VIDW0nADD2 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0100, Reset Value = 0x0000_0000 (VIDW00ADD2)
- Address = Base Address + 0x0104, Reset Value = 0x0000_0000 (VIDW01ADD2)
- Address = Base Address + 0x0108, Reset Value = 0x0000_0000 (VIDW02ADD2)
- Address = Base Address + 0x010C, Reset Value = 0x0000_0000 (VIDW03ADD2)
- Address = Base Address + 0x0110, Reset Value = 0x0000_0000 (VIDW04ADD2)

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| OFFSIZE_F | [25:13] | RW | Specifies virtual screen offset size (number of byte). This value defines the difference between address of last byte which displays on the previous video line and address of first byte which will display in the new video line. OFFSIZE_F should have value that is multiple of 4byte size or 0. | 0 |
| PAGEWIDTH_F | [12:0] | RW | Specifies virtual screen page width (number of byte). This value defines the width of view port in the frame. PAGEWIDTH should have bigger value than the burst size and you should align the size word boundary. | 0 |

NOTE: You should align the sum of PAGEWIDTH_F and OFFSIZE_F double-word (8 byte) boundary.

16.5.3.36 VIDINTCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0130, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:26] | — | Reserved | 0 |
| FIFOINTERVAL | [25:20] | RW | Controls interval of the FIFO interrupt. | 0 |
| SYSMAINCON | [19] | RW | Sends complete interrupt enable bit to Main LCD 0 = Disables Interrupt 1 = Enables Interrupt NOTE: This bit is valid if both INTEN and I80IFDONE are high. | 0 |
| SYSSUBCON | [18] | RW | Sends complete interrupt enable bit to Sub LCD 0 = Disables Interrupt 1 = Enables Interrupt NOTE: This bit is valid if both INTEN and I80IFDONE are high. | 0 |
| I80IFDONE | [17] | RW | Enables i80 Interface Interrupt (only for I80 Interface mode). 0 = Disables Interrupt1 = Enables Interrupt NOTE: This bit is valid if INTEN is high. | 0 |
| FRAMESEL0 | [16:15] | RW | Specifies Video Frame Interrupt 0 at start of: 00 = BACK Porch 01 = VSYNC 10 = ACTIVE 11 = FRONT Porch | 0 |
| FRAMESEL1 | [14:13] | RW | Specifies Video Frame Interrupt 1 at start of: 00 = None 01 = BACK Porch 10 = VSYNC 11 = FRONT Porch | 0 |
| INTFRMEN | [12] | RW | Specifies Video Frame Interrupt Enable Control Bit. 0 = Disables Video Frame Interrupt 1 = Enables Video Frame Interrupt NOTE: This bit is valid I when INTEN is high. | 0 |
| FIFOSEL | [11:5] | RW | Specifies FIFO Interrupt control bit. Each bit has a special significance: [11]Window 4 control 0 = Disables 1 = Enables [10]Window 3 control 0 = Disables 1 = Enables [9]Window 2 control 0 = Disables | 0 |

| Name | Bit | Type | Description | Reset Value |
|-----------|-------|------|---|-------------|
| | | | <p>1 = Enables [8]Reserved [7]Reserved [6]Window 1 control 0 = Disables 1 = Enables [5]Window 0 control 0 = Disables 1 = Enables NOTE: This bit is valid I if both INTEN and INTFIFOEN are high.</p> | |
| FIFOLEVEL | [4:2] | RW | <p>Selects Video FIFO Interrupt Level. 000 = 0 – 25 % 001 = 0 – 50 % 010 = 0 – 75 % 011 = 0 % (empty) 100 = 100 % (full)</p> | 0 |
| INTFIFOEN | [1] | RW | <p>Specifies Video FIFO Interrupt Enable Control Bit. 0 = Disables video FIFO level interrupt 1 = Enables video FIFO level interrupt NOTE: This bit is valid if INTEN is high.</p> | 0 |
| INTEN | [0] | RW | <p>Specifies Video Interrupt Enable Control Bit. 0 = Disables video interrupt 1 = Enables video interrupt</p> | 0 |

NOTE:

1. If video frame interrupt occurs, then you can select maximum two points by setting FRAMESEL0 and FRAMESEL1. For example, in case of FRAMESEL0 = 00 and FRAMESEL1 = 11, it triggers video frame interrupt both at the start of back porch and front porch.
2. Interrupt controller has three interrupt sources related to display controller, namely, LCD[0], LCD[1], and LCD[2]. (For more information, refer to Chapter 9 interrupt controller"). LCD[0] specifies FIFO Level interrupt, LCD[1] specifies video frame synchronization interrupt and LCD[2] specifies i80 done interface interrupt.

16.5.3.37 VIDINTCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0134, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|--------|------|--|-------------|
| RSVD | [31:5] | — | Reserved | 0 |
| RSVD | [4:3] | — | Reserved NOTE: This bit should be set to 0. | 0 |
| INTI80PEND | [2] | RW | Specifies i80 done interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = i80 done status asserts the interrupt request | 0 |
| INTFRMPEND | [1] | RW | Specifies frame synchronization interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = Frame synchronization status asserts the interrupt request | 0 |
| INTFIFOPEND | [0] | RW | Specifies FIFO Level interrupt. Writes "1" to clear this bit. 0 = Does not request interrupt 1 = FIFO empty status asserts the interrupt request. | 0 |

16.5.3.38 W1KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0140, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| KEYBLEN_F | [26] | RW | Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area | 0 |
| KEYEN_F | [25] | RW | Enables/Disables Color Key (Chroma key). 0 = Disables color key 1 = Enables color key | 0 |
| DIRCON_F | [24] | RW | Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area) | 0 |
| COMPKEY_F | [23:0] | RW | Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the position bit of COLVAL. | 0 |

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

16.5.3.39 W1KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0144, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| COLVAL_F | [23:0] | RW | Specifies color key value for transparent pixel effect. | 0 |

16.5.3.40 W2KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0148, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| KEYBLEN_F | [26] | RW | Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area | 0 |
| KEYEN_F | [25] | RW | Enables color key (Chroma key). 0 = Disables color key 1 = Enables color key | 0 |
| DIRCON_F | [24] | RW | Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area) | 0 |
| COMPKEY_F | [23:0] | RW | Each bit corresponds to COLVAL [23:0]. If some position bit is set, then it disables the position bit of COLVAL. | 0 |

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

16.5.3.41 W2KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x014C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| COLVAL_F | [23:0] | RW | Specifies color key value for transparent pixel effect. | 0 |

16.5.3.42 W3KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0150, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| KEYBLEN_F | [26] | RW | Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area | 0 |
| KEYEN_F | [25] | RW | Enables Color Key (Chroma key). 0 = Disables color key 1 = Enables color key | 0 |
| DIRCON_F | [24] | RW | Controls Color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area) | 0 |
| COMPKEY_F | [23:0] | RW | Each bit corresponds to COLVAL[23:0]. If some position bit is set, then it disables the position bit of COLVAL. | 0 |

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

16.5.3.43 W3KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0154, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| COLVAL_F | [23:0] | RW | Specifies color key value for transparent pixel effect. | 0 |

16.5.3.44 W4KEYCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0158, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| KEYBLEN_F | [26] | RW | Enables blending. 0 = Disables blending 1 = Enables blending using original alpha for non-key area and KEY_ALPHA for key area | 0 |
| KEYEN_F | [25] | RW | Enables color Key (Chroma key). 0 = Disables color key 1 = Enables color key | 0 |
| DIRCON_F | [24] | RW | Controls color key (Chroma key) direction. 0 = If the pixel value matches foreground image with COLVAL, then it displays the pixel from background image (only in OSD area) 1 = If the pixel value matches background image with COLVAL, then it displays the pixel from foreground image (only in OSD area) | 0 |
| COMPKEY_F | [23:0] | RW | Each bit corresponds to COLVAL[23:0]. If some position bit is set, then it disables the COLVAL position bit. | 0 |

NOTE: Set BLD_PIX = 1, ALPHA_SEL = 0, A_FUNC = 0x2, and B_FUNC = 0x3 to enable alpha blending using color key.

16.5.3.45 W4KEYCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x015C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| COLVAL_F | [23:0] | RW | Specifies color key value for transparent pixel effect. | 0 |

NOTE: Both COLVAL and COMPKEY use 24-bit color data in all BPP modes.

At 24 BPP Mode: 24-bit color value is valid.

- A. COLVAL
 - Red: COLVAL[23:17]
 - Green: COLVAL[15: 8]
 - Blue: COLVAL[7:0]
- B. COMPKEY
 - Red: COMPKEY[23:17]
 - Green: COMPKEY[15: 8]
 - Blue: COMPKEY[7:0]

At 16 BPP (5:6:5) mode: 16-bit color value is valid.

- A. COLVAL
 - Red: COLVAL[23:19]
 - Green: COLVAL[15: 10]
 - Blue: COLVAL[7:3]
- B. COMPKEY
 - Red: COMPKEY[23:19]
 - Green: COMPKEY[15: 10]
 - Blue: COMPKEY[7:3]
 - COMPKEY[18:16] should be 0x7.
 - COMPKEY[9: 8] should be 0x3.
 - COMPKEY[2:0] should be 0x7.

NOTE: COMPKEY register should be set properly for each BPP mode.

16.5.3.46 W1KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0160, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|------------------------------|-------------|
| RSVD | [31:14] | – | Reserved | 0 |
| KEYALPHA_R_F | [23:0] | RW | Specifies Key alpha R value. | 0 |
| KEYALPHA_G_F | [15:8] | RW | Specifies Key alpha G value. | 0 |
| KEYALPHA_B_F | [7:0] | RW | Specifies Key alpha B value. | 0 |

16.5.3.47 W2KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0164, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|------------------------------|-------------|
| RSVD | [31:14] | – | Reserved | 0 |
| KEYALPHA_R_F | [23:0] | RW | Specifies Key alpha R value. | 0 |
| KEYALPHA_G_F | [15:8] | RW | Specifies Key alpha G value. | 0 |
| KEYALPHA_B_F | [7:0] | RW | Specifies Key alpha B value. | 0 |

16.5.3.48 W3KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0168, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|------------------------------|-------------|
| RSVD | [31:14] | – | Reserved | 0 |
| KEYALPHA_R_F | [23:0] | RW | Specifies Key alpha R value. | 0 |
| KEYALPHA_G_F | [15:8] | RW | Specifies Key alpha G value. | 0 |
| KEYALPHA_B_F | [7:0] | RW | Specifies Key alpha B value. | 0 |

16.5.3.49 W4KEYALPHA

- Base Address = 0x11C0_0000
- Address = Base Address + 0x016C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|------------------------------|-------------|
| RSVD | [31:14] | – | Reserved. | 0 |
| KEYALPHA_R_F | [23:0] | RW | Specifies Key alpha R value. | 0 |
| KEYALPHA_G_F | [15:8] | RW | Specifies Key alpha G value. | 0 |
| KEYALPHA_B_F | [7:0] | RW | Specifies Key alpha B value. | 0 |

16.5.3.50 DITHMODE

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0170, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|-------|------|---|-------------|
| RSVD | [7] | RW | Does not use for normal access (writing not-zero values to these registers results in abnormal behavior.) | 0 |
| RDithPos | [6:5] | RW | Controls Red Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit | 0 |
| GDithPos | [4:3] | RW | Controls Green Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit | 0 |
| BDithPos | [2:1] | RW | Controls Blue Dither bit. 00 = 8-bit 01 = 6-bit 10 = 5-bit | 0 |
| DITHEN_F | [0] | RW | Enables Dithering bit. 0 = Disables dithering 1 = Enables dithering | 0 |

16.5.3.51 WIN0MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0180, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| MAPCOLEN_F | [24] | RW | Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables | 0 |
| MAPCOLOR | [23:0] | RW | Specifies color value. | 0 |

16.5.3.52 WIN1MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0184, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|--|-------------|
| MAPCOLEN_F | [24] | RW | Specifies the color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables | 0 |
| MAPCOLOR | [23:0] | RW | Specifies the color value. | 0 |

16.5.3.53 WIN2MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0188, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| MAPCOLEN_F | [24] | RW | Specifies the color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables | 0 |
| MAPCOLOR | [23:0] | RW | Specifies color value. | 0 |

16.5.3.54 WIN3MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0_018C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| MAPCOLEN_F | [24] | RW | Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables | 0 |
| MAPCOLOR | [23:0] | RW | Specifies color value. | 0 |

16.5.3.55 WIN4MAP

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0190, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|--------|------|---|-------------|
| MAPCOLEN_F | [24] | RW | Specifies color mapping of window control bit. If it enables this bit, then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables | 0 |
| MAPCOLOR | [23:0] | RW | Specifies color value. | 0 |

16.5.3.56 WPALCON_H

- Base Address = 0x11C0_0000
- Address = Base Address + 0x019C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|-------------|-------------|
| RSVD | [31:19] | – | Reserved | 0 |
| W4PAL_H | [18:17] | RW | W4PAL[2:1] | 0 |
| RSVD | [16:15] | RW | Reserved | 0 |
| W3PAL_H | [14:13] | RW | W3PAL[2:1] | 0 |
| RSVD | [12:11] | RW | Reserved | 0 |
| W2PAL_H | [10: 9] | RW | W2PAL[2:1] | 0 |
| RSVD | [8: 0] | RW | Reserved | 0 |

16.5.3.57 WPALCON_L

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01A0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|---|-------------|
| RSVD | [31:23] | – | Reserved | 0 |
| PALUPDATEEN | [9] | RW | 0 = Normal Mode 1 = Enables (Palette Update) | 0 |
| W4PAL_L | [8] | RW | W4PAL[0] | 0 |
| W3PAL_L | [7] | RW | W3PAL[0] | 0 |
| W2PAL_L | [6] | RW | W2PAL[0] | 0 |
| W1PAL_L | [5: 3] | RW | W1PAL[2:0] | 0 |
| W0PAL_L | [2: 0] | RW | W0PAL[2:0] | 0 |

NOTE:

1. WPALCON = {WPALCON_H,WPALCON_L}

| Name | Description | Reset Value |
|-------------|---|-------------|
| PALUPDATEEN | 0 = Normal Mode 1 = Enables (Palette Update) | 0 |
| W4PAL[3:0] | Specifies size of palette data format of Window 4. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit) | 0 |
| W3PAL[2:0] | Specifies size of palette data format of Window 3. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit) | 0 |
| W2PAL[2:0] | Specifies size of palette data format of Window 2. 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A: 8-bit) | 0 |

| Name | Description | Reset Value |
|------------|---|-------------|
| W1PAL[2:0] | Specifies size of palette data format of Window 1. 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A: 8-bit) | 0 |
| W0PAL[2:0] | Specifies size of palette data format of Window 0. 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A: 8-bit) | 0 |

2. The bit map for W0/ W1 is different from W2/W3/W4.

16.5.3.58 TRIGCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01A4, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------------------------|---------|------|---|-------------|
| RSVD | [31:27] | - | Reserved | 0 |
| SWTRGCM _D _W4BUF | [26] | RW | Specifies Window 4 double buffer trigger. 1 = Enables Software Trigger Command (write only) NOTE: Only when TRGMODE_W4BUF is set to "1" | 0 |
| TRGMODE_W4BUF | [25] | RW | Specifies Window 4 double buffer trigger. 0 = Disables trigger 1 = Enables trigger | 0 |
| RSVD | [24:22] | - | Reserved | 0 |
| SWTRGCM _D _W3BUF | [21] | RW | Specifies Window 3 double buffer trigger. 1 = Enables Software Trigger Command (write only) NOTE: Only when TRGMODE_W3BUF is set to "1" | 0 |
| TRGMODE_W3BUF | [20] | RW | Specifies Window 3 double buffer trigger. 0 = Disables trigger 1 = Enables trigger | 0 |
| RSVD | [19:17] | - | Reserved | 0 |
| SWTRGCM _D _W2BUF | [16] | RW | Specifies Window 2 double buffer trigger. 1 = Enables Software Trigger Command (write only) NOTE: Only when TRGMODE_W2BUF is set to "1" | 0 |
| TRGMODE_W2BUF | [15] | RW | Specifies Window 2 double buffer trigger. 0 = Disables trigger 1 = Enables trigger | 0 |
| RSVD | [14:12] | - | Reserved | 0 |
| SWTRGCM _D _W1BUF | [11] | RW | Specifies Window 1 double buffer trigger. 1 = Enables Software Trigger Command (write only) NOTE: Only when TRGMODE_W1BUF is set to "1" | 0 |
| TRGMODE_W1BUF | [10] | RW | Specifies Window 1 double buffer trigger. 0 = Disables trigger 1 = Enables trigger | 0 |
| RSVD | [9:7] | - | Reserved | 0 |
| SWTRGCM _D _W0BUF | [6] | RW | Specifies Window 0 double buffer trigger. 1 = Enables Software Trigger Command (write only) NOTE: Only when TRGMODE_W0BUF is set to "1" | 0 |
| TRGMODE_W0BUF | [5] | RW | Specifies Window 0 double buffer trigger. 0 = Disables trigger 1 = Enables trigger | 0 |
| RSVD | [4:3] | - | Reserved | 0 |
| SWFRSTATUS | [2] | RW | Specifies Frame Done Status (read only; i80 start) | 0 |

| Name | Bit | Type | Description | Reset Value |
|---------------|-----|------|--|-------------|
| _I80 | | | trigger) 0 = Does not request 1 = Requests NOTE: 1. Clear Condition: Read or New Frame Start 2. Only when TRGMODE is set to "1" | |
| SWTRGCMDS_I80 | [1] | RW | Enables i80 start trigger. 1 = Software Triggering Command (write only) NOTE: Only when TRGMODE is set to "1" | 0 |
| TRGMODE_I80 | [0] | RW | Enables i80 start trigger. 0 = Disables i80 Software Trigger 1 = Enables i80 Software Trigger | 0 |

NOTE: Generates two continuous software trigger inputs in some video clocks (VCLK) recognizes as one.

16.5.3.59 I80IFCONAn (n = 0 to 1)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01B0, Reset Value = 0x0000_0000 (I80IFCONA0)
- Address = Base Address + 0x01B4, Reset Value = 0x0000_0000 (I80IFCONA1)

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [22:20] | – | Reserved | 0 |
| LCD_CS_SETUP | [19:16] | RW | Specifies number of clock cycles for the active period of address signal enable to chip select enable. | 0 |
| LCD_WR_SETUP | [15:12] | RW | Specifies number of clock cycles for the active period of CS signal enable to write signal enable. | 0 |
| LCD_WR_ACT | [11:8] | RW | Specifies number of clock cycles for the active period of chip select enable. | 0 |
| LCD_WR_HOLD | [7:4] | RW | Specifies number of clock cycles for the active period of chip select disable to write signal disable. | 0 |
| RSVD | [3] | – | Reserved | – |
| RSPOL | [2] | RW | Specifies polarity of RS Signal 0 = Low 1 = High | 0 |
| RSVD | [1] | – | Reserved | 0 |
| I80IFEN | [0] | RW | Controls the LCD i80 interface. 0 = Disables 1 = Enables | 0 |

16.5.3.60 I80IFCONBn (n = 0 to 1)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01B8, Reset Value = 0x0000_0000 (I80IFCONB0)
- Address = Base Address + 0x01BC, Reset Value = 0x0000_0000 (I80IFCONB1)

| Name | Bit | Type | Description | Reset Value |
|---------------|---------|------|---|-------------|
| RSVD | [11:10] | - | Reserved | 0 |
| NORMAL_CMD_ST | [9] | RW | 1 = Normal Command Start NOTE: Auto clears after sending out one set of commands | 0 |
| RSVD | [8:7] | - | Reserved | - |
| FRAME_SKIP | [6:5] | RW | Specifies i80 Interface Output Frame Decimation Factor. 00 = 1 (Does not Skip) 01 = 2 10 = 3 | 00 |
| RSVD | [4] | - | Reserved | 0 |
| AUTO_CMD_RATE | [3:0] | RW | 0000 = Disables auto command (if you do not use any auto-command, then you should set AUTO_CMD_RATE as "0000"). 0001 = per 2 Frames 0010 = per 4 Frames 0011 = per 6 Frames ... 1111 = per 30 Frames | 0000 |

16.5.3.61 COLORGAINCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01C0, Reset Value = 0x1004_0100

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---|-------------|
| RSVD | [31:30] | – | Reserved | 0 |
| CG_RGAIN | [29:20] | RW | <p>Specifies color gain value of R data (maximum 4, 8-bit resolution).</p> <p>0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) ... 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 ... 0x3FF = 3.99609375 (maximum)</p> | 0x100 |
| CG_GGAIN | [19:10] | RW | <p>Specifies color gain value of G data (maximum 4, 8-bit resolution).</p> <p>0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) ... 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 ... 0x3FF = 3.99609375 (maximum)</p> | 0x100 |
| CG_BGAIN | [9:0] | RW | <p>Specifies color gain value of B data (maximum 4, 8-bit resolution).</p> <p>0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) ... 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 ... 0x3FF = 3.99609375 (maximum)</p> | 0x100 |

16.5.3.62 LDI_CMDCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01D0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:24] | - | Reserved | - |
| CMD11_EN | [23:22] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD10_EN | [21:20] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD9_EN | [19:18] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD8_EN | [17:16] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD7_EN | [15:14] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD6_EN | [13:12] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD5_EN | [11:10] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD4_EN | [9:8] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |

| Name | Bit | Type | Description | Reset Value |
|---------|-------|------|--|-------------|
| CMD3_EN | [7:6] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD2_EN | [5:4] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Normal and Auto Command Enable | 00 |
| CMD1_EN | [3:2] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |
| CMD0_EN | [1:0] | RW | Controls command 11 00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command | 00 |

16.5.3.63 LDI_CMDCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01D4, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|------------------------|-------------|
| RSVD | [31:10] | – | Reserved. | 0 |
| CMD11_RS | [11] | RW | Controls Command 11 RS | 0 |
| CMD10_RS | [10] | RW | Controls Command 10 RS | 0 |
| CMD9_RS | [9] | RW | Controls Command 9 RS | 0 |
| CMD8_RS | [8] | RW | Controls Command 8 RS | 0 |
| CMD7_RS | [7] | RW | Controls Command 7 RS | 0 |
| CMD6_RS | [6] | RW | Controls Command 6 RS | 0 |
| CMD5_RS | [5] | RW | Controls Command 5 RS | 0 |
| CMD4_RS | [4] | RW | Controls Command 4 RS | 0 |
| CMD3_RS | [3] | RW | Controls Command 3 RS | 0 |
| CMD2_RS | [2] | RW | Controls Command 2 RS | 0 |
| CMD1_RS | [1] | RW | Controls Command 1 RS | 0 |
| CMD0_RS | [0] | RW | Controls Command 0 RS | 0 |

16.5.3.64 SIFCCON0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E0, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|-----|------|---|-------------|
| RSVD | [7] | - | Reserved NOTE: This bit should be set to 0. | 0 |
| SYS_ST_CON | [6] | RW | Controls LCD i80 System Interface ST Signal. 0 = Low 1 = High | 0 |
| SYS_RS_CON | [5] | RW | Controls LCD i80 System Interface RS Signal. 0 = Low 1 = High | 0 |
| SYS_nCS0_CON | [4] | RW | Controls LCD i80 System Interface nCS0 (main) Signal. 0 = Disables (High) 1 = Enables (Low) | 0 |
| SYS_nCS1_CON | [3] | RW | Controls LCD i80 System Interface nCS1 (sub) Signal. 0 = Disables (High) 1 = Enables (Low) | 0 |
| SYS_nOE_CON | [2] | RW | Controls LCD i80 System Interface nOE Signal. 0 = Disables (High) 1 = Enables (Low) | 0 |
| SYS_nWE_CON | [1] | RW | Controls LCD i80 System Interface nWE Signal. 0 = Disables (High) 1 = Enables (Low) | 0 |
| SCOMEN | [0] | RW | Enables LCD i80 System Interface Command Mode. 0 = Disables (Normal Mode) 1 = Enables (Manual Command Mode) | |

16.5.3.65 SIFCCON1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E4, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|---|-------------|
| SYS_WDATA | [23:0] | RW | Controls LCD i80 System Interface Write Data. | 0 |

16.5.3.66 SIFCCON2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01E8, Reset Value = 0x????_????

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| SYS_RDATA | [23:0] | R | Controls LCD i80 System Interface Read Data. | 0 |

16.5.3.67 HUECOEF_CR_n (n = 1 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01EC, Reset Value = 0x0100_0100 (HUECOEF_CR_1)
- Address = Base Address + 0x01F0, Reset Value = 0x0000_0000 (HUECOEF_CR_2)
- Address = Base Address + 0x01F4, Reset Value = 0x0000_0000 (HUECOEF_CR_3)
- Address = Base Address + 0x01F8, Reset Value = 0x0100_0100 (HUECOEF_CR_4)

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| RSVD | [31:26] | - | Reserved | 0 |
| CRG0_x | [25:16] | RW | <p>Specifies Hue matrix coefficient 00 (when "cb + ln_offset" is positive). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) ... 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = - 1.0 (- 256/256) 0h301 = - 0.99609375 (- 255/256) ... 0h3FF = - 0.00390625 (- 1/256) 0h101 to 2FF = Reserved (do not use)</p> | 0x100 |
| RSVD | [15:10] | - | Reserved | 0 |
| CRG1_x | [9:0] | RW | <p>Specifies Hue matrix coefficient 00 (when "cb + ln_offset" is negative). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256) ... 0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = - 1.0 (- 256/256) 0h301 = - 0.99609375 (- 255/256) ... 0h3FF = - 0.00390625 (- 1/256) 0h101 to 2FF = Reserved (do not use)</p> | 0x100 |

16.5.3.68 HUECOEF_CB_n (n = 1 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x01FC, Reset Value = 0x0100_0100 (HUECOEF_CB_1)
- Address = Base Address + 0x0200, Reset Value = 0x0000_0000 (HUECOEF_CB_2)
- Address = Base Address + 0x0204, Reset Value = 0x0000_0000 (HUECOEF_CB_3)
- Address = Base Address + 0x0208, Reset Value = 0x0100_0100 (HUECOEF_CB_4)

| Name | Bit | Type | Description | Reset Value |
|--------|---------|------|---|-------------|
| RSVD | [31:26] | - | Reserved | 0 |
| CBG0_x | [25:16] | RW | <p>Specifies Hue matrix coefficient 00 (when "cb + In_offset" is positive). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256)</p> <p>...</p> <p>0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = - 1.0 (- 256/256) 0h301 = - 0.99609375 (- 255/256)</p> <p>...</p> <p>0h3FF = - 0.00390625 (- 1/256) 0h101 to 2FF = Reserved (do not use)</p> | 0x100 |
| RSVD | [15:10] | - | Reserved | 0 |
| CBG1_x | [9:0] | RW | <p>Specifies Hue matrix coefficient 00 (when "cb + In_offset" is negative). (Signed) 0h000 = 0 0h001 = 0.00390625 (1/256) 0h002 = 0.0078125 (2/256)</p> <p>...</p> <p>0h0FF = 0.99609375 (255/256) 0h100 = 1.0 (256/256) 0h300 = - 1.0 (- 256/256) 0h301 = - 0.99609375 (- 255/256)</p> <p>...</p> <p>0h3FF = - 0.00390625 (- 1/256) 0h101 to 2FF = Reserved (do not use)</p> | 0x100 |

16.5.3.69 HUEOFFSET

- Base Address = 0x11C0_0000
- Address = Base Address + 0x020C, Reset Value = 0x0108_0080

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|------------------|
| RSVD | [31:25] | - | Reserved | 0 |
| OFFSET_IN | [24:16] | RW | Specifies Hue matrix input offset (signed). 0h000 = + 0 0h001 = + 1 0h002 = + 2 ... 0h0FF = + 255 0h100 = - 256 ... 0x1FF = -1 | 0x180 (- 128) |
| RSVD | [15:9] | - | Reserved | 0 |
| OFFSET_OUT | [8:0] | RW | Specifies Hue matrix output offset (signed). 0h000 = + 0 0h001 = + 1 0h002 = + 2 ... 0h0FF = + 255 0h100 = - 256 ... 0x1FF = -1 | 0x080 (+ 128) |

NOTE: Generally, HUE_OFFSET_IN = - 128 and HUE_OFFSET_OUT = + 128

Example 16-6 Hue Equation

```
Cb<hue> = CBG0 • (Cb + OFFSET_IN) + CBG1 • (Cr + OFFSET_IN) + OFFSET_OUT
Cr<hue> = CRG0 • (Cb + OFFSET_IN) + CRG1 • (Cr + OFFSET_IN) + OFFSET_OUT
```

Example 16-7 Coefficient Decision

```
CBG0 = (Cb - 128) ≥ 0 ? CBG0_P : CBG0_N
CBG1 = (Cr - 128) ≥ 0 ? CBG1_P : CBG1_N
CRG0 = (Cb - 128) ≥ 0 ? CRG0_P : CRG0_N
CRG1 = (Cr - 128) ≥ 0 ? CRG1_P : CRG1_N
```

16.5.3.70 VIDW0ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x021C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| ALPHA0_R_F | [23:16] | RW | Specifies Red Alpha value (case AEN == 0). | 0 |
| ALPHA0_G_F | [15:8] | RW | Specifies Green Alpha value (case AEN == 0). | 0 |
| ALPHA0_B_F | [7:0] | RW | Specifies Blue Alpha value (case AEN == 0). | 0 |

16.5.3.71 VIDW0ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0220, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| ALPHA1_R_F | [23:16] | RW | Specifies Red Alpha value (case AEN == 1). | 0 |
| ALPHA1_G_F | [15:8] | RW | Specifies Green Alpha value (case AEN == 1). | 0 |
| ALPHA1_B_F | [7:0] | RW | Specifies Blue Alpha value (case AEN == 1). | 0 |

16.5.3.72 VIDW1ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0224, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA0_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 0). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA0_G_L_F | [11:8] | RW | Specifies Green Alpha lower value (case AEN == 0). | 0 |
| RSVD | [7:4] | – | Reserved | 0 |
| ALPHA0_B_L_F | [3:0] | RW | Specifies Blue Alpha lower value (case AEN == 0). | 0 |

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD1C

ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW1ALPHA0

16.5.3.73 VIDW1ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0228, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA1_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 1). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA1_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 1). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA1_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 1). | 0 |

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD1C
 ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW1ALPHA1

16.5.3.74 VIDW2ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x022C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA0_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 0). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA0_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 0). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA0_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 0). | 0 |

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD2C
 ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW2ALPHA0

16.5.3.75 VIDW2ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0230, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA1_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 1). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA1_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 1). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA1_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 1). | 0 |

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD2C
 ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW2ALPHA1

16.5.3.76 VIDW3ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0234, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA0_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 0). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA0_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 0). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA0_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 0). | 0 |

NOTE: ALPHA0_R (G, B)[7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD3C
 ALPHA0_R (G, B)[3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW3ALPHA0

16.5.3.77 VIDW3ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0238, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:16] | – | Reserved | 0 |
| ALPHA1_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 1). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA1_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 1). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA1_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 1). | 0 |

NOTE: ALPHA1_R (G, B)[7:4] = ALPHA1_R (G, B)_H[3:0]@VIDOSD3C
 ALPHA1_R (G, B)[3:0] = ALPHA1_R (G, B)_L[3:0]@VIDW3ALPHA1

16.5.3.78 VIDW4ALPHA0

- Base Address = 0x11C0_0000
- Address = Base Address + 0x023C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA0_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 0). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA0_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 0). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA0_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 0). | 0 |

NOTE: ALPHA0_R (G, B)[7:4] = ALPHA0_R (G, B)_H[3:0] at VIDOSD4C
 ALPHA0_R (G, B)[3:0] = ALPHA0_R (G, B)_L[3:0] at VIDW4ALPHA0

16.5.3.79 VIDW4ALPHA1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0240, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|--|-------------|
| RSVD | [24] | – | Reserved | 0 |
| RSVD | [23:20] | – | Reserved | 0 |
| ALPHA1_R_L_F | [19:16] | RW | Specifies Red Alpha lower value (case AEN == 1). | 0 |
| RSVD | [15:12] | – | Reserved | 0 |
| ALPHA1_G_L_F | [11: 8] | RW | Specifies Green Alpha lower value (case AEN == 1). | 0 |
| RSVD | [7: 4] | – | Reserved | 0 |
| ALPHA1_B_L_F | [3: 0] | RW | Specifies Blue Alpha lower value (case AEN == 1). | 0 |

NOTE: ALPHA1_R (G, B)[7:4] = ALPHA1_R (G, B)_H[3:0] at VIDOSD4C
 ALPHA1_R (G, B)[3:0] = ALPHA1_R (G, B)_L[3:0] at VIDW4ALPHA1

16.5.3.80 BLENDEQ1

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0244, Reset Value = 0x0000_00C2

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:22] | – | Reserved | 0x000 |
| Q_FUNC_F | [21:18] | RW | Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved | 0x0 |
| RSVD | [17:16] | – | Reserved | 00 |
| P_FUNC_F | [15:12] | RW | Specifies the constant that it uses in alpha. Same as above (see COEF_Q). | 0x0 |
| RSVD | [11:10] | – | Reserved | 00 |
| B_FUNC_F | [9:6] | RW | Specifies the constant that it uses in B. Same as above (see COEF_Q). | 0x3 |
| RSVD | [5:4] | – | Reserved | 00 |
| A_FUNC_F | [3:0] | RW | Specifies the constant that it uses in A. Same as above (see COEF_Q). | 0x2 |

NOTE: For more information, refer to [Figure 16-23](#), "Blending equation".

1. Background = Window 0, foreground = Window 1 (in Blend Equation 1)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL at WINCONx, and WxPAL at WPALCON decides the alphaA and alphaB.

16.5.3.81 BLENDEQ2

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0248, Reset Value = 0x0000_00C2

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---|-------------|
| RSVD | [31:22] | – | Reserved | 0x000 |
| Q_FUNC_F | [21:18] | RW | Specifies constant that it uses in alphaB (alpha value of background (1)). 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved | 0x0 |
| RSVD | [17:16] | – | Reserved | 00 |
| P_FUNC_F | [15:12] | RW | Specifies constant that it uses in alpha. Same as above (see COEF_Q) | 0x0 |
| RSVD | [11:10] | – | Reserved | 00 |
| B_FUNC_F | [9:6] | RW | Specifies constant that it uses in B. Same as above (see COEF_Q) | 0x3 |
| RSVD | [5:4] | – | Reserved | 00 |
| A_FUNC_F | [3:0] | RW | Specifies constant that it uses in A. Same as above (see COEF_Q) | 0x2 |

NOTE: For more information, Refer to [Figure 16-23](#), "Blending equation".

1. Background = Window 01, foreground = Window 2 (in Blend Equation 2)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL at WINCONx, and WxPAL at WPALCON decides the alphaA and alphaB.

16.5.3.82 BLENDEQ3

- Base Address = 0x11C0_0000
- Address = Base Address + 0x024C, Reset Value = 0x0000_00C2

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:22] | – | Reserved | 0x000 |
| Q_FUNC_F | [21:18] | RW | Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved | 0x0 |
| RSVD | [17:16] | – | Reserved | 00 |
| P_FUNC_F | [15:12] | RW | Specifies constant that it uses in alpha. Same as above (see COEF_Q). | 0x0 |
| RSVD | [11:10] | – | Reserved | 00 |
| B_FUNC_F | [9:6] | RW | Specifies constant that it uses in B. Same as above (see COEF_Q). | 0x3 |
| RSVD | [5:4] | – | Reserved | 00 |
| A_FUNC_F | [3:0] | RW | Specifies constant that it uses in A. Same as above (see COEF_Q). | 0x2 |

NOTE: For more information, Refer to [Figure 16-23](#), "Blending equation".

1. Background = Window 012, foreground = Window 3 (in Blend Equation 3)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL @ WINCONx, and WxPAL @ WPALCON decides the alphaA and alphaB.

16.5.3.83 BLENDEQ4

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0250, Reset Value = 0x0000_00C2

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| RSVD | [31:22] | - | Reserved | 0x000 |
| Q_FUNC_F | [21:18] | RW | Specifies constant that it uses in alphaB (alpha value of background (1)) 0000 = 0 (zero) 0001 = 1 (maximum) 0010 = alphaA (2) (alpha value of foreground (1)) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 0110 = ALPHA0 0111 = Reserved 100x = Reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = Reserved | 0x0 |
| RSVD | [17:16] | - | Reserved | 00 |
| P_FUNC_F | [15:12] | RW | Specifies constant that it uses in alpha. Same as above (see COEF_Q). | 0x0 |
| RSVD | [11:10] | - | Reserved | 00 |
| B_FUNC_F | [9:6] | RW | Specifies constant that it uses in B. Same as above (see COEF_Q). | 0x3 |
| RSVD | [5:4] | - | Reserved | 00 |
| A_FUNC_F | [3:0] | RW | Specifies constant that it uses in A. Same as above (see COEF_Q). | 0x2 |

NOTE: For more information, Refer to [Figure 16-23](#), "Blending equation".

1. Background = Window 0123, foreground = Window 4 (in Blend Equation 4)
2. BPPMODE_F, BLD_PIX, ALPHA_SEL @ WINCONx, and WxPAL @ WPALCON decides the alphaA and alphaB. .

16.5.3.84 BLENDCON

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0260, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|--|-------------|
| RSVD | [31:1] | – | Reserved | 0x000 |
| BLEND_NEW | [0] | RW | Specifies alpha value width. 0 = 4-bit alpha value 1 = 8-bit alpha value | 0x0 |

16.5.3.85 WnRTQOSCON (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0264, Reset Value = 0x0000_0000 (W0RTQOSCON)
- Address = Base Address + 0x0268, Reset Value = 0x0000_0000 (W1RTQOSCON)
- Address = Base Address + 0x026C, Reset Value = 0x0000_0000 (W2RTQOSCON)
- Address = Base Address + 0x0270, Reset Value = 0x0000_0000 (W3RTQOSCON)
- Address = Base Address + 0x0274, Reset Value = 0x0000_0000 (W4RTQOSCON)

| Name | Bit | Type | Description | Reset Value |
|--------------|---------|------|---|-------------|
| RSVD | [31:12] | – | Reserved NOTE: This bit should be set to 0. | 0 |
| FIFOLEVEL | [11:4] | RW | Specifies real-time QoS FIFO level. If FIFO depth is less than FIFOLEVEL[7:0], then RTQoS output is 1. | 0 |
| RSVD | [3:2] | – | Reserved NOTE: This bit should be set to 0. | 0 |
| QOS_GATE_DIS | [1] | RW | Disables RTQoS output signal gate. 0 = Gates 1 = Does not gate | 0 |
| RSVD | [0] | – | Reserved NOTE: This bit should be set to 0. | 0 |

16.5.3.86 LDI_CMDn (n = 0 to 11)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x0280, Reset Value = 0x0000_0000 (LDI_CMD0)
- Address = Base Address + 0x0284, Reset Value = 0x0000_0000 (LDI_CMD1)
- Address = Base Address + 0x0288, Reset Value = 0x0000_0000 (LDI_CMD2)
- Address = Base Address + 0x028C, Reset Value = 0x0000_0000 (LDI_CMD3)
- Address = Base Address + 0x0290, Reset Value = 0x0000_0000 (LDI_CMD4)
- Address = Base Address + 0x0294, Reset Value = 0x0000_0000 (LDI_CMD5)
- Address = Base Address + 0x0298, Reset Value = 0x0000_0000 (LDI_CMD6)
- Address = Base Address + 0x029C, Reset Value = 0x0000_0000 (LDI_CMD7)
- Address = Base Address + 0x02A0, Reset Value = 0x0000_0000 (LDI_CMD8)
- Address = Base Address + 0x02A4, Reset Value = 0x0000_0000 (LDI_CMD9)
- Address = Base Address + 0x02A8, Reset Value = 0x0000_0000 (LDI_CMD10)
- Address = Base Address + 0x02AC, Reset Value = 0x0000_0000 (LDI_CMD11)

| Name | Bit | Type | Description | Reset Value |
|---------|--------|------|------------------------|-------------|
| LDI_CMD | [23:0] | RW | Specifies LDI command. | 0 |

16.5.4 Gamma Lookup Table

16.5.4.1 Gamma LUT Data for 64 Step Mode

- Base Address = 0x11C0_0000
- Address = Base Address + 0x037C, Reset Value = 0x0010_0000 (GAMMALUT_01_00)
- Address = Base Address + 0x0380, Reset Value = 0x0030_0020 (GAMMALUT_03_02)
- Address = Base Address + 0x0384, Reset Value = 0x0050_0040 (GAMMALUT_05_04)
- Address = Base Address + 0x0388, Reset Value = 0x0070_0060 (GAMMALUT_07_06)
- Address = Base Address + 0x038C, Reset Value = 0x0090_0080 (GAMMALUT_09_08)
- Address = Base Address + 0x0390, Reset Value = 0x00B0_00A0 (GAMMALUT_11_10)
- Address = Base Address + 0x0394, Reset Value = 0x00D0_00C0 (GAMMALUT_13_12)
- Address = Base Address + 0x0398, Reset Value = 0x00F0_00E0 (GAMMALUT_15_14)
- Address = Base Address + 0x039C, Reset Value = 0x0110_0100 (GAMMALUT_17_16)
- Address = Base Address + 0x03A0, Reset Value = 0x0130_0120 (GAMMALUT_19_18)
- Address = Base Address + 0x03A4, Reset Value = 0x0150_0140 (GAMMALUT_21_20)
- Address = Base Address + 0x03A8, Reset Value = 0x0170_0160 (GAMMALUT_23_22)
- Address = Base Address + 0x03AC, Reset Value = 0x0190_0180 (GAMMALUT_25_24)
- Address = Base Address + 0x03B0, Reset Value = 0x01B0_01A0 (GAMMALUT_27_26)
- Address = Base Address + 0x03B4, Reset Value = 0x01F0_01C0 (GAMMALUT_29_28)
- Address = Base Address + 0x03B8, Reset Value = 0x01F0_01E0 (GAMMALUT_31_30)
- Address = Base Address + 0x03BC, Reset Value = 0x0210_0200 (GAMMALUT_33_32)
- Address = Base Address + 0x03C0, Reset Value = 0x0230_0220 (GAMMALUT_35_34)
- Address = Base Address + 0x03C4, Reset Value = 0x0250_0240 (GAMMALUT_37_36)
- Address = Base Address + 0x03C8, Reset Value = 0x0270_0260 (GAMMALUT_39_38)
- Address = Base Address + 0x03CC, Reset Value = 0x0290_0280 (GAMMALUT_41_40)
- Address = Base Address + 0x03D0, Reset Value = 0x02B0_02A0 (GAMMALUT_43_42)
- Address = Base Address + 0x03D4, Reset Value = 0x02D0_02C0 (GAMMALUT_45_44)
- Address = Base Address + 0x03D8, Reset Value = 0x02F0_02E0 (GAMMALUT_47_46)
- Address = Base Address + 0x03DC, Reset Value = 0x0310_0300 (GAMMALUT_49_48)
- Address = Base Address + 0x03E0, Reset Value = 0x0330_0320 (GAMMALUT_51_50)
- Address = Base Address + 0x03E4, Reset Value = 0x0350_0340 (GAMMALUT_53_52)
- Address = Base Address + 0x03E8, Reset Value = 0x0370_0360 (GAMMALUT_55_54)
- Address = Base Address + 0x03EC, Reset Value = 0x0390_0380 (GAMMALUT_57_56)
- Address = Base Address + 0x03F0, Reset Value = 0x03B0_03A0 (GAMMALUT_59_58)
- Address = Base Address + 0x03F4, Reset Value = 0x03D0_03C0 (GAMMALUT_61_60)
- Address = Base Address + 0x03F8, Reset Value = 0x03F0_03E0 (GAMMALUT_63_62)
- Address = Base Address + 0x03FC, Reset Value = 0x0000_0400 (GAMMALUT_xx_64)

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| GM_LUT_x | [26:18] | RW | Specifies Gamma LUT value register of index x. | Undefined |
| GM_LUT_y | [10: 2] | RW | Specifies Gamma LUT value register of index y. | Undefined |

16.5.4.2 Gamma LUT Data for 16 Step Mode

- Base Address = 0x11C0_0000
- Address = Base Address + 0X037C, Reset Value = 0X0010_0000 (GAMMALUT_R_1_0)
- Address = Base Address + 0X0380, Reset Value = 0X0030_0020 (GAMMALUT_R_3_2)
- Address = Base Address + 0X0384, Reset Value = 0X0050_0040 (GAMMALUT_R_5_4)
- Address = Base Address + 0X0388, Reset Value = 0X0070_0060 (GAMMALUT_R_7_6)
- Address = Base Address + 0X038C, Reset Value = 0X0090_0080 (GAMMALUT_R_9_8)
- Address = Base Address + 0X0390, Reset Value = 0X00B0_00A0 (GAMMALUT_R_11_10)
- Address = Base Address + 0X0394, Reset Value = 0X00D0_00C0 (GAMMALUT_R_13_12)
- Address = Base Address + 0X0398, Reset Value = 0X00F0_00E0 (GAMMALUT_R_15_14)
- Address = Base Address + 0X039C, Reset Value = 0X0110_0100 (GAMMALUT_R_16)
- Address = Base Address + 0X03A0, Reset Value = 0X0130_0120 (GAMMALUT_R_1_0)
- Address = Base Address + 0X03A4, Reset Value = 0X0150_0140 (GAMMALUT_R_3_2)
- Address = Base Address + 0X03A8, Reset Value = 0X0170_0160 (GAMMALUT_R_5_4)
- Address = Base Address + 0X03AC, Reset Value = 0X0190_0180 (GAMMALUT_R_7_6)
- Address = Base Address + 0X03B0, Reset Value = 0X01B0_01A0 (GAMMALUT_R_9_8)
- Address = Base Address + 0X03B4, Reset Value = 0X01D0_01C0 (GAMMALUT_R_11_10)
- Address = Base Address + 0X03B8, Reset Value = 0X01F0_01E0 (GAMMALUT_R_13_12)
- Address = Base Address + 0X03BC, Reset Value = 0X0210_0200 (GAMMALUT_R_15_14)
- Address = Base Address + 0X03C0, Reset Value = 0X0230_0220 (GAMMALUT_R_16)
- Address = Base Address + 0X03C4, Reset Value = 0X0250_0240 (GAMMALUT_R_1_0)
- Address = Base Address + 0X03C8, Reset Value = 0X0270_0260 (GAMMALUT_R_3_2)
- Address = Base Address + 0X03CC, Reset Value = 0X0290_0280 (GAMMALUT_R_5_4)
- Address = Base Address + 0X03D0, Reset Value = 0X02B0_02A0 (GAMMALUT_R_7_6)
- Address = Base Address + 0X03D4, Reset Value = 0X02D0_02C0 (GAMMALUT_R_9_8)
- Address = Base Address + 0X03D8, Reset Value = 0X02F0_02E0 (GAMMALUT_R_11_10)
- Address = Base Address + 0X03DC, Reset Value = 0X0310_0300 (GAMMALUT_R_13_12)
- Address = Base Address + 0X03E0, Reset Value = 0X0330_0320 (GAMMALUT_R_15_14)
- Address = Base Address + 0X03E4, Reset Value = 0X0350_0340 (GAMMALUT_R_16)

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|-------------|
| GM_LUT_x | [26:18] | RW | Specifies Gamma LUT value register of index x. | Undefined |
| GM_LUT_y | [10: 2] | RW | Specifies Gamma LUT value register of index y. | Undefined |

16.5.5 Shadow Windows Control

16.5.5.1 SHD_VIDW0nADD0 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x40A0, Reset Value = 0x0000_0000 (SHD_VIDW00ADD0)
- Address = Base Address + 0x40A8, Reset Value = 0x0000_0000 (SHD_VIDW01ADD0)
- Address = Base Address + 0x40B0, Reset Value = 0x0000_0000 (SHD_VIDW02ADD0)
- Address = Base Address + 0x40B8, Reset Value = 0x0000_0000 (SHD_VIDW03ADD0)
- Address = Base Address + 0x40C0, Reset Value = 0x0000_0000 (SHD_VIDW04ADD0)

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| VBASEU_F | [31:0] | R | Specifies A[31:0] of the start address for video frame buffer (shadow). | 0 |

16.5.5.2 SHD_VIDW0nADD1 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x40D0, Reset Value = 0x0000_0000 (SHD_VIDW00ADD1)
- Address = Base Address + 0x40D8, Reset Value = 0x0000_0000 (SHD_VIDW01ADD1)
- Address = Base Address + 0x40E0, Reset Value = 0x0000_0000 (SHD_VIDW02ADD1)
- Address = Base Address + 0x40E8, Reset Value = 0x0000_0000 (SHD_VIDW03ADD1)
- Address = Base Address + 0x40F0, Reset Value = 0x0000_0000 (SHD_VIDW04ADD1)

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| VBASEL_F | [31:0] | R | Specifies A[31:0] of the end address for video buffer (shadow). | 0x0 |

16.5.5.3 SHD_VIDW0nADD2 (n = 0 to 4)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x4100, Reset Value = 0x0000_0000 (SHD_VIDW00ADD2)
- Address = Base Address + 0x4104, Reset Value = 0x0000_0000 (SHD_VIDW01ADD2)
- Address = Base Address + 0x4108, Reset Value = 0x0000_0000 (SHD_VIDW02ADD2)
- Address = Base Address + 0x410C, Reset Value = 0x0000_0000 (SHD_VIDW03ADD2)
- Address = Base Address + 0x4110, Reset Value = 0x0000_0000 (SHD_VIDW04ADD2)

| Name | Bit | Type | Description | Reset Value |
|-------------|---------|------|--|-------------|
| OFFSIZE_F | [25:13] | R | Specifies virtual screen offset size that is the number of byte (shadow). | 0 |
| PAGEWIDTH_F | [12:0] | R | Specifies virtual screen page width (number of byte). This value defines the width of view port in the frame (shadow). | 0 |

16.5.6 Palette Ram

16.5.6.1 Win0 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2400, 0x0400, Reset Value = 0x0000_0000
- Address = Base Address + 0x2404, 0x0404, Reset Value = 0x0000_0000
- Address = Base Address + 0x27FC, 0x07FC, Reset Value = 0x0000_0000

| Register | Address | Type | Description | Reset Value |
|----------|------------------------|------|---|-------------|
| 00 | 0x0_2400 (0x0_0400) | RW | Specifies Window 0 Palette entry 0 address. | Undefined |
| 01 | 0x0_2404 (0x0_0404) | RW | Specifies Window 0 Palette entry 1 address. | Undefined |
| – | – | – | – | – |
| FF | 0x0_27FC (0x0_07FC) | RW | Specifies Window 0 Palette entry 255 address. | Undefined |

16.5.6.2 Win1 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2800, 0x0800, Reset Value = 0x0000_0000
- Address = Base Address + 0x2804, 0x0804, Reset Value = 0x0000_0000
- Address = Base Address + 0x2BFC, 0x0BFC, Reset Value = 0x0000_0000

| Register | Address | Type | Description | Reset Value |
|----------|------------------------|------|---|-------------|
| 00 | 0x0_2800 (0x0_0800) | RW | Specifies Window 1 Palette entry 0 address. | Undefined |
| 01 | 0x0_2804 (0x0_0804) | RW | Specifies Window 1 Palette entry 1 address. | Undefined |
| – | – | – | – | – |
| FF | 0x0_2BFC (0x0_0BFC) | RW | Specifies Window 1 Palette entry 255 address. | Undefined |

16.5.6.3 Win2 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x2C00,0x0C00, Reset Value = 0x0000_0000
- Address = Base Address + 0x2C04,0x0C04, Reset Value = 0x0000_0000
- Address = Base Address + 0x2FFC,0x0FFC, Reset Value = 0x0000_0000

| Register | Address | Type | Description | Reset Value |
|----------|----------|------|---|-------------|
| 00 | 0x0_2C00 | RW | Specifies Window 2 Palette entry 0 address. | Undefined |
| 01 | 0x0_2C04 | RW | Specifies Window 2 Palette entry 1 address. | Undefined |
| – | – | – | – | – |
| FF | 0x0_2FFC | RW | Specifies Window 2 Palette entry 255 address. | Undefined |

16.5.6.4 Win3 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x3000, Reset Value = 0x0000_0000
- Address = Base Address + 0x3004, Reset Value = 0x0000_0000
- Address = Base Address + 0x33FC, Reset Value = 0x0000_0000

| Register | Address | Type | Description | Reset Value |
|----------|----------|------|---|-------------|
| 00 | 0x0_3000 | RW | Specifies Window 3 Palette entry 0 address. | Undefined |
| 01 | 0x0_3004 | RW | Specifies Window 3 Palette entry 1 address. | Undefined |
| – | – | – | – | – |
| FF | 0x0_33FC | RW | Specifies the Window 3 Palette entry 255 address. | Undefined |

16.5.6.5 Win4 Palette Ram Access Address (not SFR)

- Base Address = 0x11C0_0000
- Address = Base Address + 0x3400, Reset Value = 0x0000_0000
- Address = Base Address + 0x3404, Reset Value = 0x0000_0000
- Address = Base Address + 0x37FC, Reset Value = 0x0000_0000

| Register | Address | Type | Description | Reset Value |
|----------|----------|------|---|-------------|
| 00 | 0x0_3400 | RW | Specifies Window 4 Palette entry 0 address. | Undefined |
| 01 | 0x0_3404 | RW | Specifies Window 4 Palette entry 1 address. | Undefined |
| – | – | – | – | – |
| FF | 0x0_37FC | R/W | Specifies Window 4 Palette entry 255 address. | Undefined |

17 Keypad Interface

17.1 Overview

The Keypad Interface block in Exynos 4412 facilitates communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 14 rows and eight columns. You can use keypad interface on port 0 or port 1. Port 0 and port 1 has the same function. You can use any port for the GPIO connection. Port 0 column is using alive power, therefore, it can use wakeup source without any setting. But port 1 column is using normal power, therefore, it can use wakeup source with GPIO setting for retention. Interrupt delivers the events of key press or key release to the CPU.

There are two types of scans in Keypad Interface. They are, Software Scan and Hardware Scan.

In software scan mode, if one of the interrupt occurs from row lines, then the software should scan the column lines using the proper procedure to detect one or multiple key press or release.

In hardware scan mode, if you press any one of the keys, then the hardware reports the row and column number of the pressed key after it scans the column line automatically. Multiple key press support in hardware scan mode is limited to dual key with other row.

It provides interrupt status register bits at the time of key pressed or key released or both cases (when it enables two interrupt conditions). To prevent the switching noises, keypad interface comprise of internal debouncing filter.

[Figure 17-1](#) illustrates the key matrix interface external connection guide.

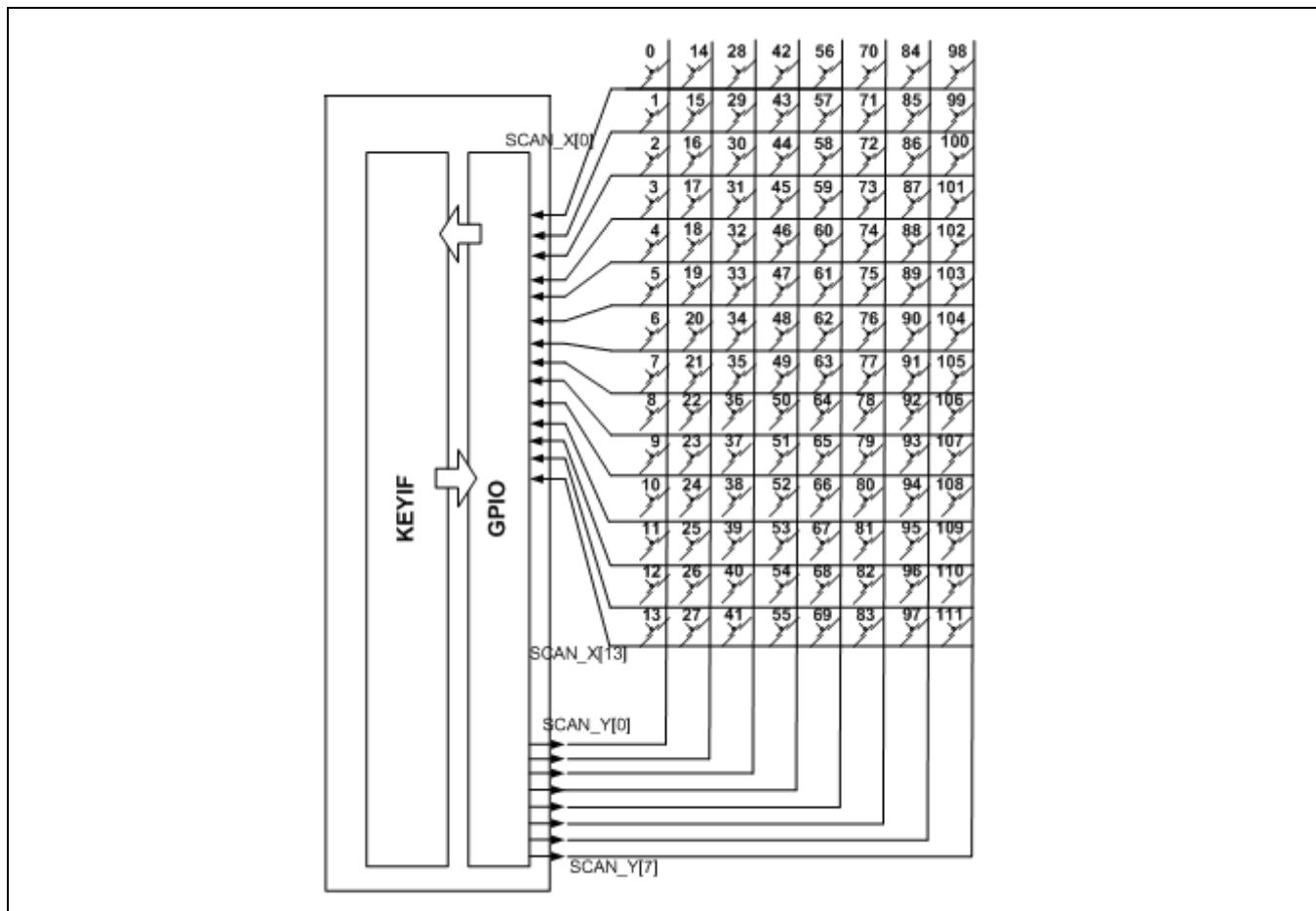


Figure 17-1 Key Matrix Interface External Connection Guide

17.2 Debouncing Filter

Supports debouncing filter for keypad interrupt of any key input. The filtering width is approximately 62.5 usec ("FCLK" two-clock, when the FCLK is 32 kHz). The keypad interrupt (key pressed or key released) to the CPU in software scan mode is an ANDed signal of the all row input lines after filtering.

[Figure 17-2](#) illustrates the internal debouncing filter operation.

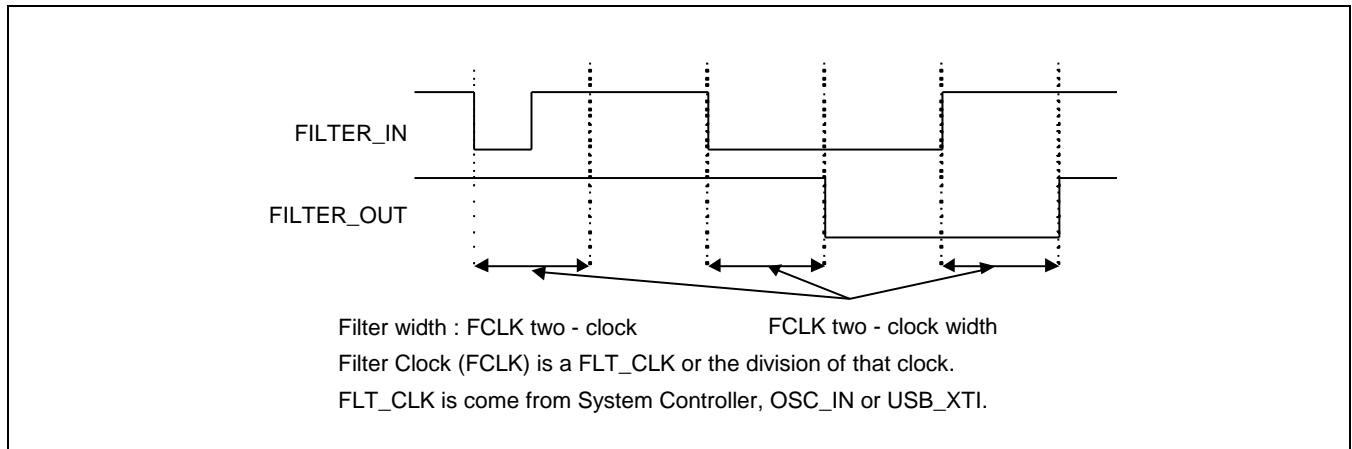


Figure 17-2 Internal Debouncing Filter Operation

17.3 Filter Clock

It divides the KEYPAD interface debouncing filter clock (FCLK) from FLT_CLK that is OSC_IN. You can set compare value for 10-bit up-counter (KEYIFFC). When filter enable bit (FC_EN) is HIGH, filter clock divider is ON. The frequency of FCLK is frequency of $\text{FLT_CLK}/((\text{KEYIFFC} + 1) \times 2)$. On the contrary, if FC_EN is LOW, then the filter clock divider does not divide FLT_CLK.

17.4 Wakeup Source

It uses KEYPAD inputs as a wakeup source. When it uses Key input for wakeup source from Audio playback, STOP, DSTOP, or SLEEP mode, it does not require KEYPAD interface register setting. However, GPIO register (GPX1CON, GPX2CON, GPX3CON, or GPL2CON) should be set for KEYPAD interface and SYSCON register should be set for masking.

17.5 Keypad Scanning Procedure for Software Scan

At initial state, all column lines (outputs) are low level. But column data output tri-state enable bits are all high. Therefore, when it does not use the tri-state enable mode, these bits should be written to zeros. When the status of the key is not pressed, then all row lines (inputs) are high (used pull-up pads). When you press any key, then the corresponding row and column lines are shortened together and a low level is driven on the corresponding row line. This generates a keypad interrupt.

The CPU (software) outputs a LOW on one column line and Hi-Z on the others by setting KEYIFCOLEN and KEYIFCOL fields in KEYIFCOL register. Each time when it writes, the CPU reads the value of the KEYIFROW register and detects if one key of the corresponding column line is pressed. If KEYIF has pull-up PAD, then it reads each KEYIFROW bits as HIGH, except pressed ROW bit. When the scanning procedure ends, it detects the pressed key (one or more).

[Figure 17-3](#) illustrates the keypad scanning procedure.

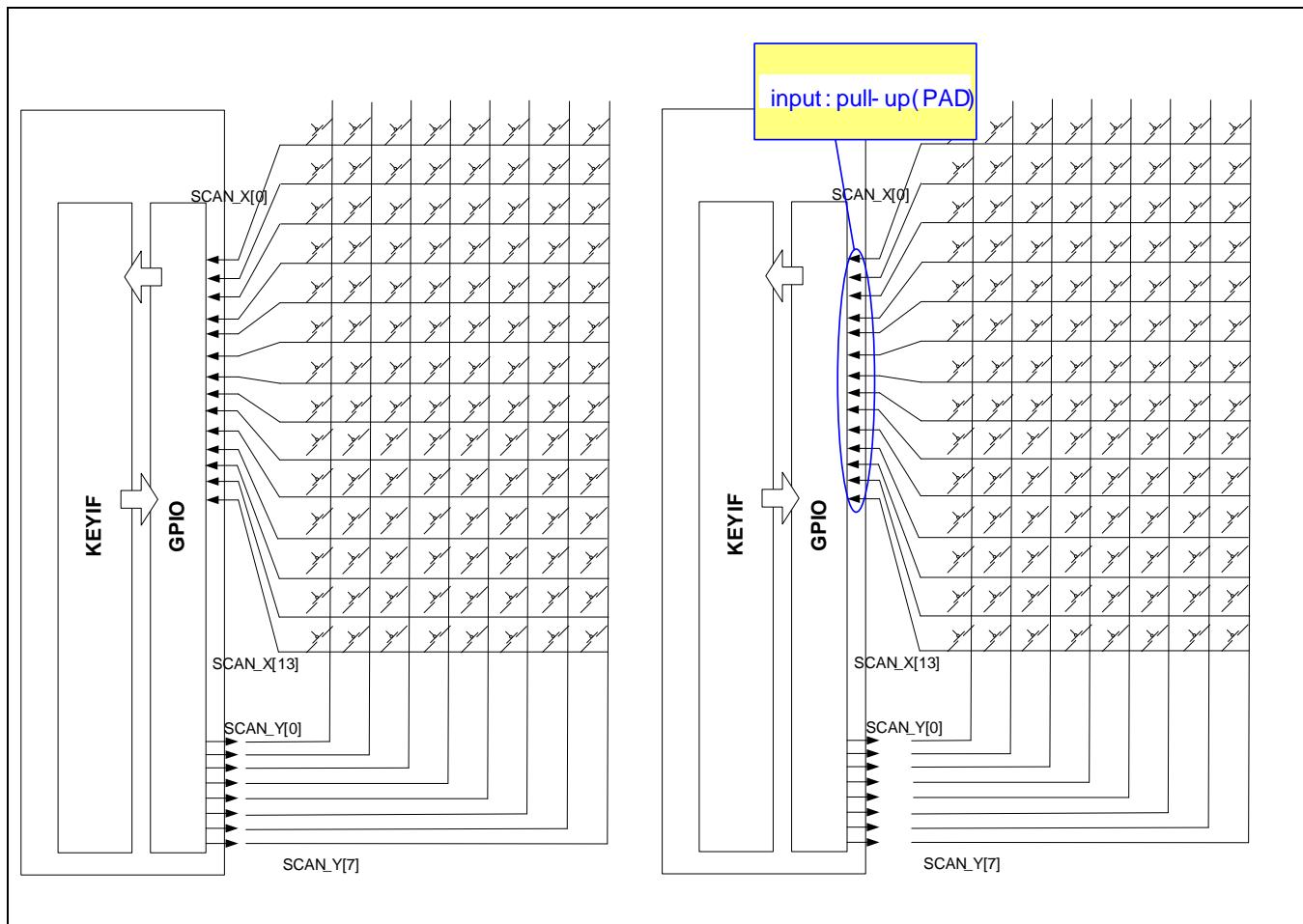


Figure 17-3 Keypad Scanning Procedure

[Figure 17-4](#) illustrates the keypad scanning procedure II.

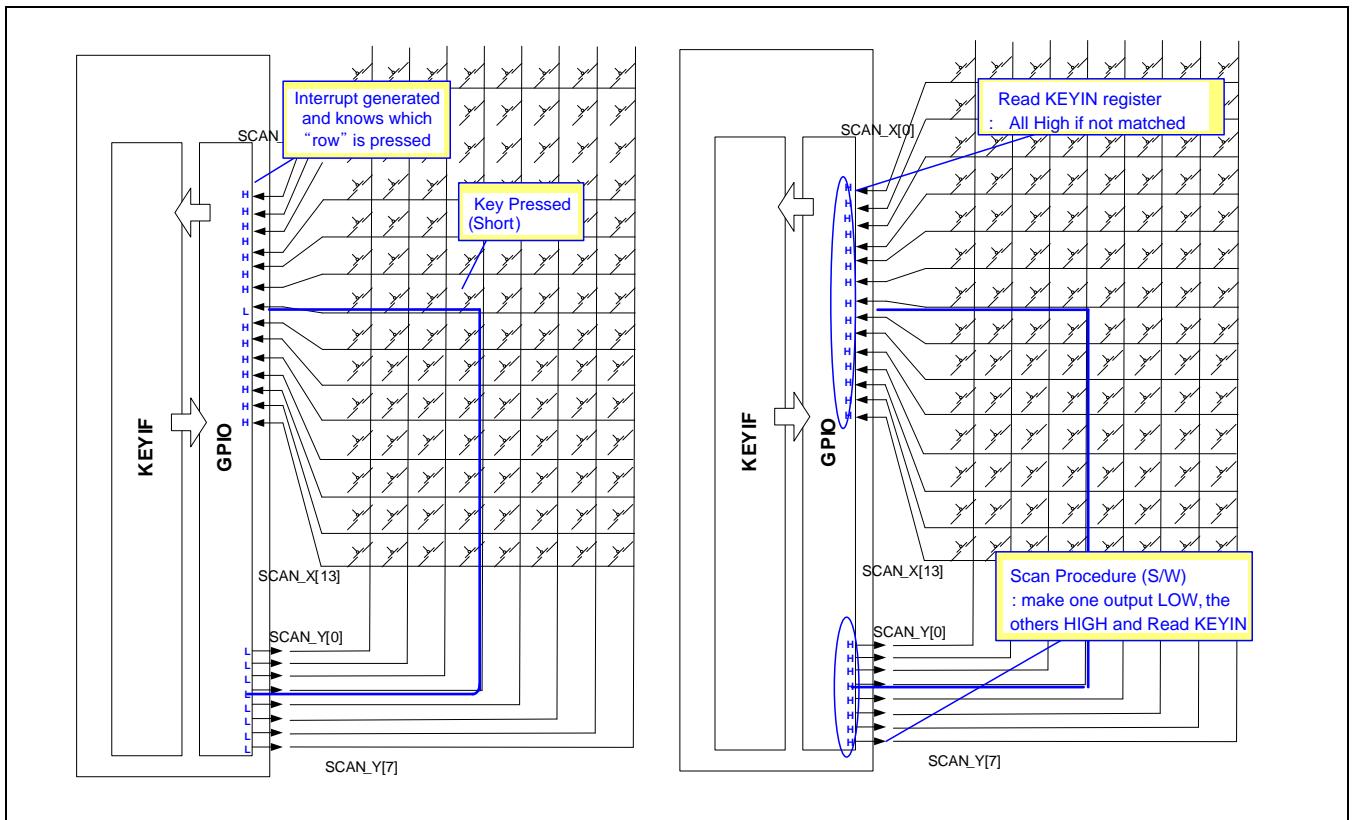


Figure 17-4 Keypad Scanning Procedure II

[Figure 17-5](#) illustrates the keypad scanning procedure III.

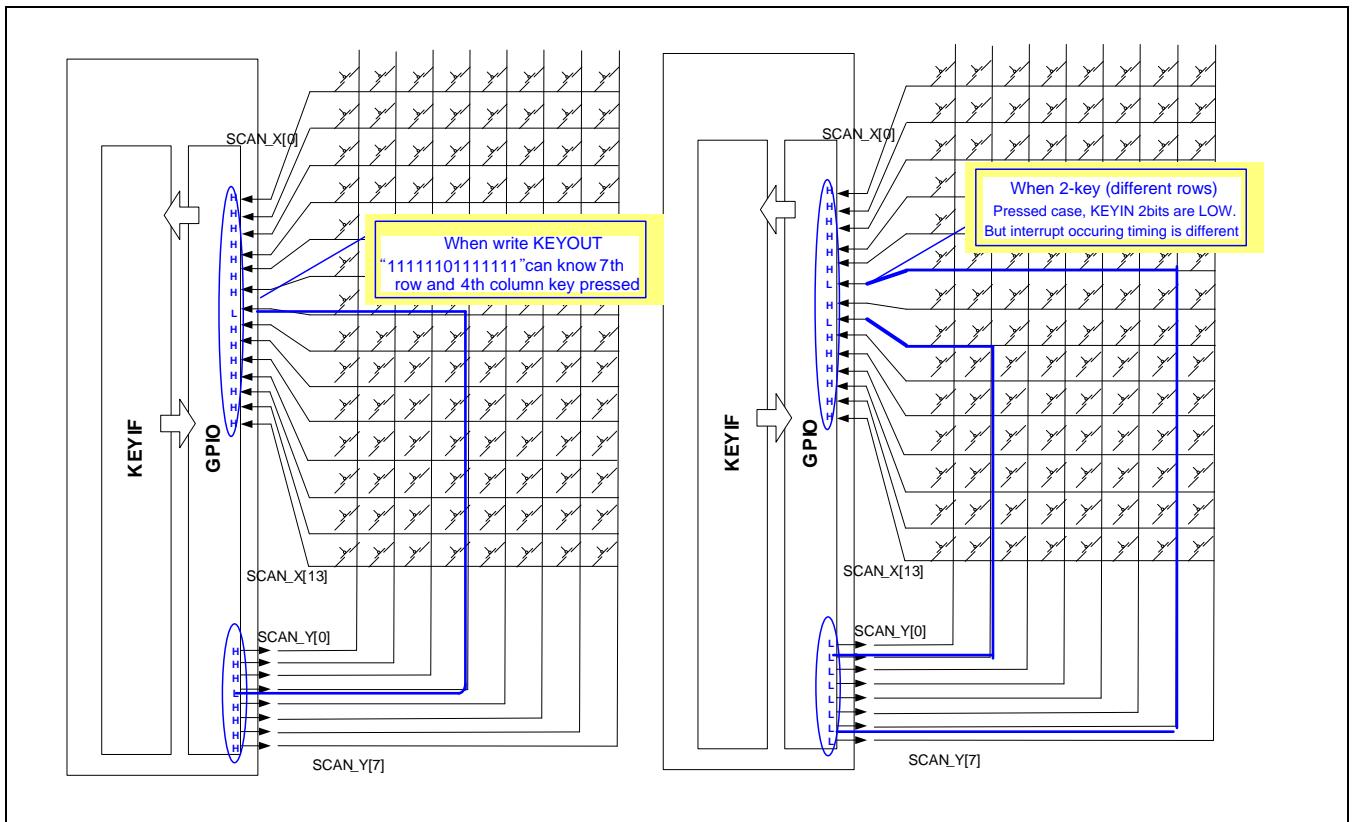


Figure 17-5 Keypad Scanning Procedure III

[Figure 17-6](#) illustrates the keypad scanning procedure when the two-key pressed with different row.

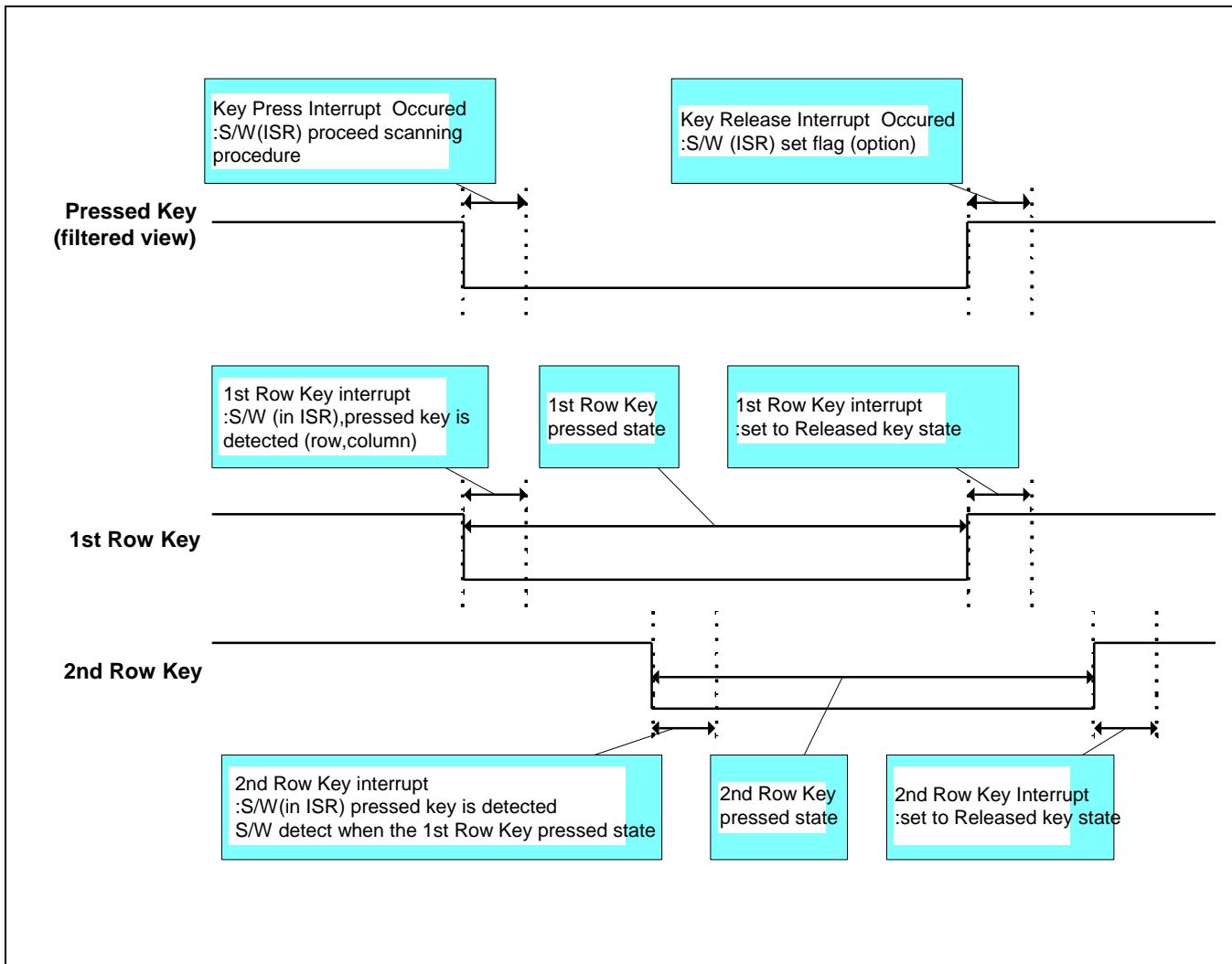


Figure 17-6 Keypad Scanning Procedure when the Two-key Pressed with Different Row

[Figure 17-7](#) illustrates the keypad I/F block diagram.

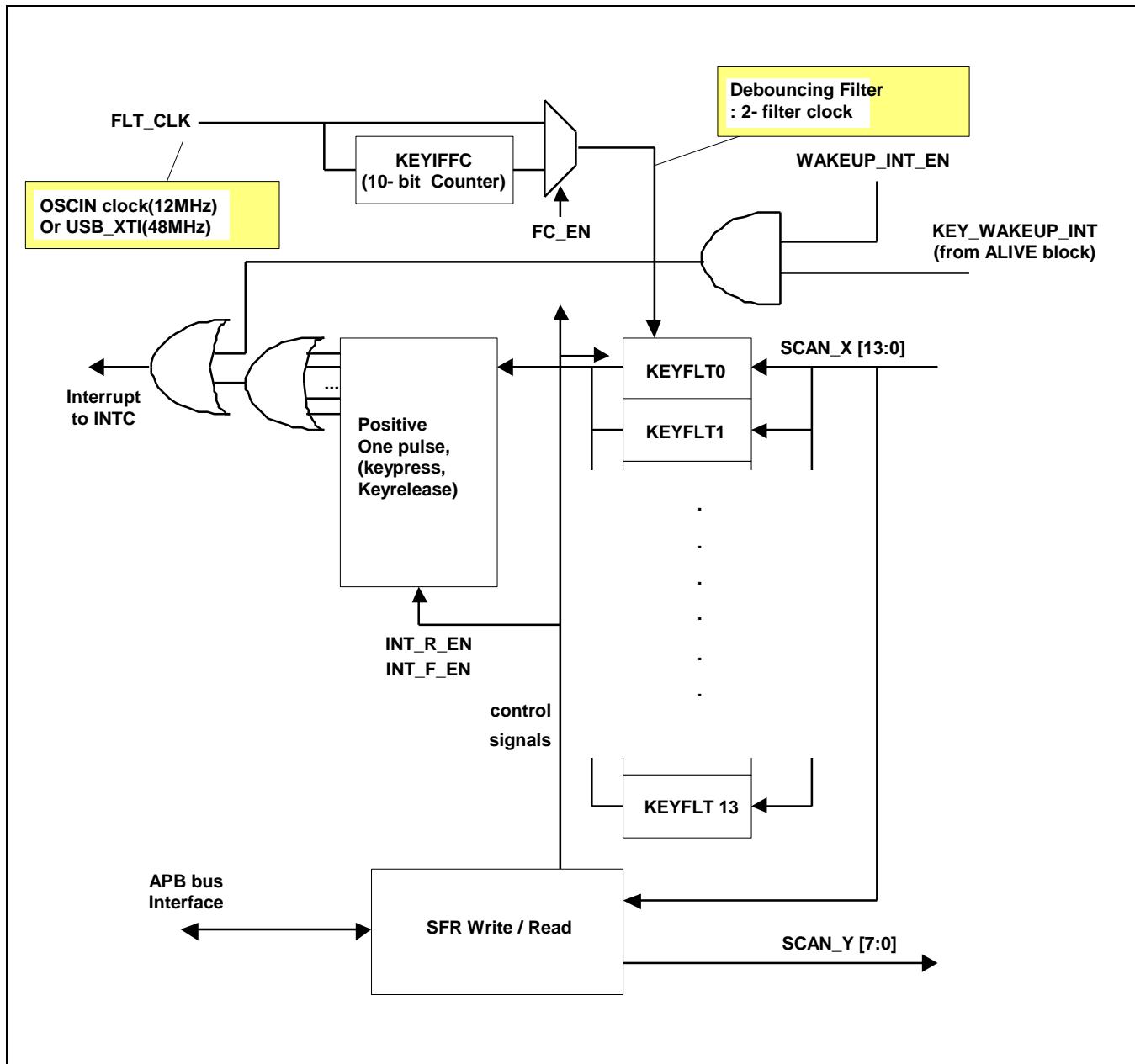


Figure 17-7 Keypad I/F Block Diagram

17.6 Keypad Scanning Procedure for Hardware Scan

At initial stage, the keypad scanning procedures are same as software scan. If any key is pressed, the hardware automatically scans the corresponding row and column lines and writes the information into the register. After scan and write to the register, it generates keypad interrupt.

The CPU (software) can get the row and column number by accessing the KEYIFSCAN1 (first key) or KEYIFSCAN2 (second key) register. The value of KEYIFSCAN1 and KEYIFSCAN2 is valid when the key is pressed. At hardware scan mode, you should set the H_CNT value in KEYIFCON register. The initial value is 0xF. In each scanning step, after driving the column, scanning hardware waits for H_CNT cycle. When the row input signal is stable (after H_CNT cycle), scanning hardware verifies the row input signal.

It limits the multiple key press support in hardware scan mode to dual key with other row.

17.7 I/O Description

[Table 17-1](#) describes the keypad interface I/O.

Table 17-1 Keypad Inaterface I/O Description

| Signal | I/O | Description | Pad | | Type |
|-------------|-----|---------------------------------|-----------------------|---------------------------|-------|
| | | | Port0 | Port1 | |
| KP_ROW[13] | I | KEYPAD interface row[13] data | XEINT_29 (GPX3[5]) | XEINT_29 (GPX3[5]) | muxed |
| KP_ROW [12] | I | KEYPAD interface row[12] data | XEINT_28 (GPX3[4]) | XEINT_28 (GPX3[4]) | muxed |
| KP_ROW [11] | I | KEYPAD interface row[11] data | XEINT_27 (GPX3[3]) | XEINT_27 (GPX3[3]) | muxed |
| KP_ROW [10] | I | KEYPAD interface row[10] data | XEINT_26 (GPX3[2]) | XEINT_26 (GPX3[2]) | muxed |
| KP_ROW [9] | I | KEYPAD interface row[9] data | XEINT_25 (GPX3[1]) | XEINT_25 (GPX3[1]) | muxed |
| KP_ROW [8] | I | KEYPAD interface row[8] data | XEINT_24 (GPX3[0]) | XEINT_24 (GPX3[0]) | muxed |
| KP_ROW [7] | I | KEYPAD interface row[7] data | XEINT_23 (GPX2[7]) | XEINT_23 (GPX2[7]) | muxed |
| KP_ROW [6] | I | KEYPAD interface row[6] data | XEINT_22 (GPX2[6]) | XEINT_22 (GPX2[6]) | muxed |
| KP_ROW [5] | I | KEYPAD interface row[5] data | XEINT_21 (GPX2[5]) | XEINT_21 (GPX2[5]) | muxed |
| KP_ROW [4] | I | KEYPAD interface row[4] data | XEINT_20 (GPX2[4]) | XEINT_20 (GPX2[4]) | muxed |
| KP_ROW [3] | I | KEYPAD interface row[3] data | XEINT_19 (GPX2[3]) | XEINT_19 (GPX2[3]) | muxed |
| KP_ROW [2] | I | KEYPAD interface row[2] data | XEINT_18 (GPX2[2]) | XEINT_18 (GPX2[2]) | muxed |
| KP_ROW [1] | I | KEYPAD interface row[1] data | XEINT_17 (GPX2[1]) | XEINT_17 (GPX2[1]) | muxed |
| KP_ROW [0] | I | KEYPAD interface row[0] data | XEINT_16 (GPX2[0]) | XEINT_16 (GPX2[0]) | muxed |
| KP_COL [7] | O | KEYPAD interface column[7] data | XEINT_15 (GPX1[7]) | XGNSS_GPIO_7 (GPL2[7]) | muxed |
| KP_COL [6] | O | KEYPAD interface column[6] data | XEINT_14 (GPX1[6]) | XGNSS_GPIO_6 (GPL2[6]) | muxed |
| KP_COL [5] | O | KEYPAD interface column[5] data | XEINT_13 (GPX1[5]) | XGNSS_GPIO_5 (GPL2[5]) | muxed |
| KP_COL [4] | O | KEYPAD interface column[4] data | XEINT_12 (GPX1[4]) | XGNSS_GPIO_4 (GPL2[4]) | muxed |
| KP_COL [3] | O | KEYPAD interface column[3] data | XEINT_11 | XGNSS_GPIO_3 | muxed |

| Signal | I/O | Description | Pad | | Type |
|---------------|------------|------------------------------------|-----------------------|---------------------------|-------------|
| | | | Port0 | Port1 | |
| | | | (GPX1[3]) | (GPL2[3]) | |
| KP_COL [2] | O | KEYPAD interface column[2] data | XEINT_10 (GPX1[2]) | XGNSS_GPIO_2 (GPL2[2]) | muxed |
| KP_COL [1] | O | KEYPAD interface column[1] data | XEINT_9 (GPX1[1]) | XGNSS_GPIO_1 (GPL2[1]) | muxed |
| KP_COL [0] | O | KEYPAD INTERFACE COLUMN[0] data | XEINT_8 (GPX1[0]) | XGNSS_GPIO_0 (GPL2[0]) | muxed |

17.8 Register Description

17.8.1 Register Map Summary

- Base Address: 0x100A_0000

| Register | Offset | Description | Reset Value |
|-------------|--------|---|----------------------|
| KEYIFCON | 0x0000 | Specifies KEYPAD interface control register | 0x000F_0000 |
| KEYIFSTSCLR | 0x0004 | Specifies KEYPAD interrupt for software scan status and clear register | 0x0000_0000 |
| KEYIFCOL | 0x0008 | Specifies KEYPAD interface column data output register | 0x0000_FF00 |
| KEYIFROW | 0x000C | Specifies KEYPAD interface row data input register | Reflects input ports |
| KEYIFFC | 0x0010 | Specifies KEYPAD interface debouncing filter clock division register | 0x0000_0000 |
| KEYIFSCAN1 | 0x0014 | Specifies KEYPAD interface output result of hardware scan for first key register | 0x0000_0000 |
| KEYIFSCAN2 | 0x0018 | Specifies KEYPAD interface output result of hardware scan for second key register | 0x0000_0000 |
| KEYIFHSC | 0x001C | Specifies KEYPAD interrupt for hardware scan status and clear register | 0x0000_0000 |

17.8.1.1 KEYIFCON

- Base Address : 0x100A_0000
- Address = Base Address + 0x0000, Reset Value = 0x000F_0000

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|--|-------------|
| H_CNT | [31:16] | RW | Counter value for hardware scan column to row interval | 16'hF |
| RSVD | [15:10] | - | Reserved | - |
| HIZSCAN_EN | [9] | RW | Hi-Z mode scan enable for hardware scan 0 = Normal scan (driving "low and high") 1 = Hi-Z mode scan (driving "low and Hi-Z") In Hi-Z mode, it should disable GPIO internal pull-down. | 1'b0 |
| SEL_HSCAN | [8] | RW | Select hardware scan/software scan 0 = Software scan 1 = Hardware scan | 1'b0 |
| RSVD | [7:4] | - | Reserved | - |
| FC_EN | [3] | RW | 10-bit counter (for debouncing digital filter clock) enable 0 = Disables. Does not use division counter 1 = Enables. uses division counter | 1'b0 |
| DF_EN | [2] | RW | KEYPAD input port debouncing filter enable 0 = Disables 1 = Enables | 1'b0 |
| INT_R_EN | [1] | RW | KEYPAD input port rising edge (key-released) interrupt 0 = Disables 1 = Enables | 1'b0 |
| INT_F_EN | [0] | RW | KEYPAD input port falling edge (key-pressed) interrupt 0 = Disables 1 = Enables | 1'b0 |

NOTE: Selects both edge interrupt when both INT_F_EN and INT_R_EN are set.

17.8.1.2 KEYIFSTSCLR

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0004, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|-------|---------|------|---|-------------|
| R_INT | [29:16] | RW | <p>KEYPAD input "release" interrupt (rising edge) status (read) and clear (write).</p> <p>Read: 0 = Does not occur 1 = Released interrupt occurs</p> <p>Write: Clears released interrupt when write "1"</p> <p>The R_INT[13:0] indicates that each key pressed from 0 to 13 has a dedicated interrupt from R_INT[16] to R_INT[29]</p> | 14'b0 |
| P_INT | [13:0] | RW | <p>KEYPAD input "press" interrupt (falling edge) status(read) and clear(write)</p> <p>Read: 0 = Does not occur 1 = Pressed interrupt occurs</p> <p>Write: Clears pressed interrupt when write "1"</p> <p>The P_INT[13:0] indicate that each key released from 0 to 13 has a dedicated interrupt from P_INT[0] to P_INT[13]</p> | 14'b0 |

NOTE: Clears keypad wakeup interrupt when the write access to the KEYIFSTSCLR.

17.8.1.3 KEYIFCOL

- Base Address: 0x100A_0000
- Address = Base Address +0x0008, Reset Value = 0x0000_FF00

| Name | Bit | Type | Description | Reset Value |
|------------|---------|------|---|--------------|
| RSVD | [31:16] | — | Reserved | — |
| KEYIFCOLEN | [15:8] | RW | <p>KEYPAD interface column data output tri-state enable register</p> <p>Each bit is for each KEYIFCOL bit.</p> <p>0 = Enables output pad tri-state buffer (Normal output, KEY enable) 1 = Disables output pad Tri-state buffer (High-Z output, KEY disable)</p> | 8'b1111_1111 |
| KEYIFCOL | [7:0] | RW | KEYPAD interface column data output register | 8'b0 |

17.8.1.4 KEYIFROW

- Base Address: 0x100A_0000
- Address : Base Address +0x000C, Reset Value = Reflects input ports

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|--|----------------------|
| RSVD | [31:14] | — | Reserved | — |
| KEYIFROW | [13:0] | R | KEYPAD interface row data input register (read only) This register values from input ports are not filtered data. | Reflects input ports |

17.8.1.5 KEYIFFC

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0010, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|---------|------|---|-------------|
| RSVD | [31:10] | — | Reserved | — |
| KEYIFFC | [9:0] | RW | KEYPAD interface debouncing filter clock division register. You can set compare value for 10-bit up-counter. This register value means when FC_EN bit is HIGH. FCLK = FLT_CLK/(KEYIFFC[9:0] + 1) (FLT_CLK is from OSC_IN) | 10'b0 |

17.8.1.6 KEYIFSCAN1

- Base Address : 0x100A_0000
- Address = Base Address+ 0x0014, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---|-------------|
| RSVD | [31:22] | — | Reserved | — |
| ROWSCAN1 | [21:8] | R | KEYPAD interface scan result of row (only pressed row has "1") Clears value when first key is released | 14'b0 |
| COLSCAN1 | [7:0] | R | KEYPAD interface scan result of column (only pressed column has "1") Clears value when first key is released | 8'b0 |

17.8.1.7 KEYIFSCAN2

- Base Address: 0x100A_0000
- Address = Base Address+ 0x0018, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|---------|------|---|-------------|
| RSVD | [31:22] | — | Reserved | — |
| ROWSCAN2 | [21:8] | R | KEYPAD interface scan result of row (only pressed row has "1") Clears value when first key is released | 14'b0 |
| COLSCAN2 | [7:0] | R | KEYPAD interface scan result of column (only pressed column has "1") Clears value when first key is released | 8'b0 |

17.8.1.8 KEYIFHSC

- Base Address: 0x100A_0000
- Address = Base Address+ 0x001C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|----------|--------|------|---|-------------|
| RSVD | [31:4] | — | Reserved | — |
| HSCAN_R2 | [3] | R | KEYPAD input "release" interrupt (rising edge) status(read) and clear(write) for hardware scan of second Key Read: 0 = Does not occur 1 = Released interrupt occurs Write: Clears released interrupt when write "1" | 1'b0 |
| HSCAN_P2 | [2] | R | KEYPAD input "press" interrupt (falling edge) status(read) and clear(write) for HW scan of second Key Read: 0 = Does not occur 1 = Pressed interrupt occurs Write: Clear pressed interrupt when write "1" | 1'b0 |
| HSCAN_R1 | [1] | R | KEYPAD input "release" interrupt (rising edge) status(read) and clear(write) for HW scan of first Key Read: 0 = Does not occur 1 = Released interrupt occurs Write: Clears released interrupt when write "1" | 1'b0 |
| HSCAN_P1 | [0] | R | KEYPAD input "press" interrupt (falling edge) status(read) and clear(write) for hardware scan of first Key Read: 0 = Does not occur 1 = Pressed interrupt occurs Write: Clears pressed interrupt when write "1" | 1'b0 |

18 ADC

This chapter describes the functions and usage of general ADC.

18.1 Overview

The 10-bit or 12-bit CMOS Analog to Digital Converter (ADC) comprises of 4-channel analog inputs. It converts the analog input signal into 10-bit or 12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. ADC supports low power mode.

18.2 Features

The ADC includes the following features:

- Resolution: 10-bit/12-bit (optional)
- Differential Nonlinearity Error: ± 2.0 LSB (Max.)
- Integral Nonlinearity Error: ± 4.0 LSB (Max.)
- Top Offset Error : 0 to + 55 LSB
- Bottom Offset Error : 0 to – 55 LSB
- Maximum Conversion Rate: 1 MSPS
- Low Power Consumption
- Power Supply Voltage: 1.8 V (Typ.), 1.0 V (Typ., Digital I/O Interface)
- Analog Input Range: 0 to 1.8 V

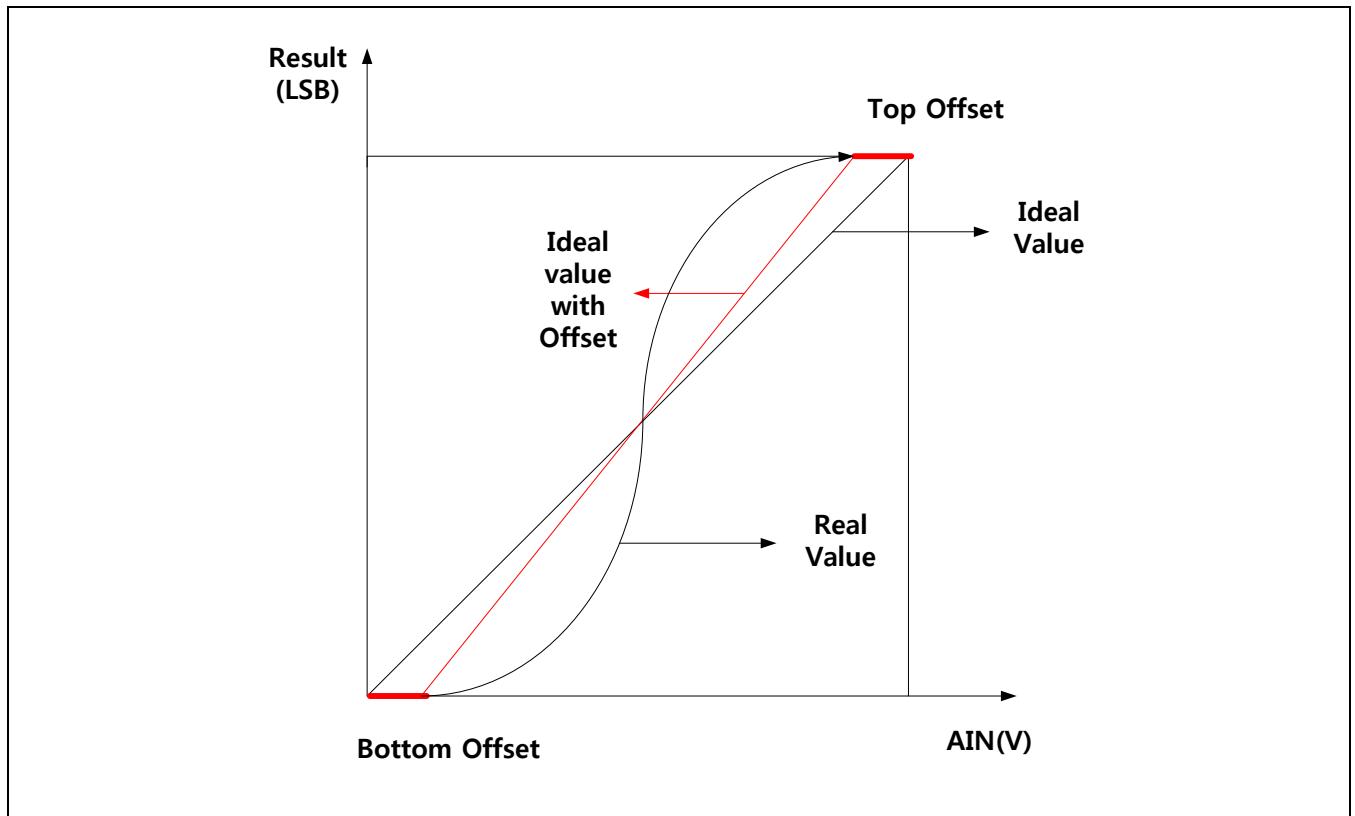


Figure 18-1 ADC Top/Bottom Offset Error Diagram

18.3 Functional Description

18.3.1 Block Diagram

[Figure 18-2](#) is the functional block diagram of general A/D converter.

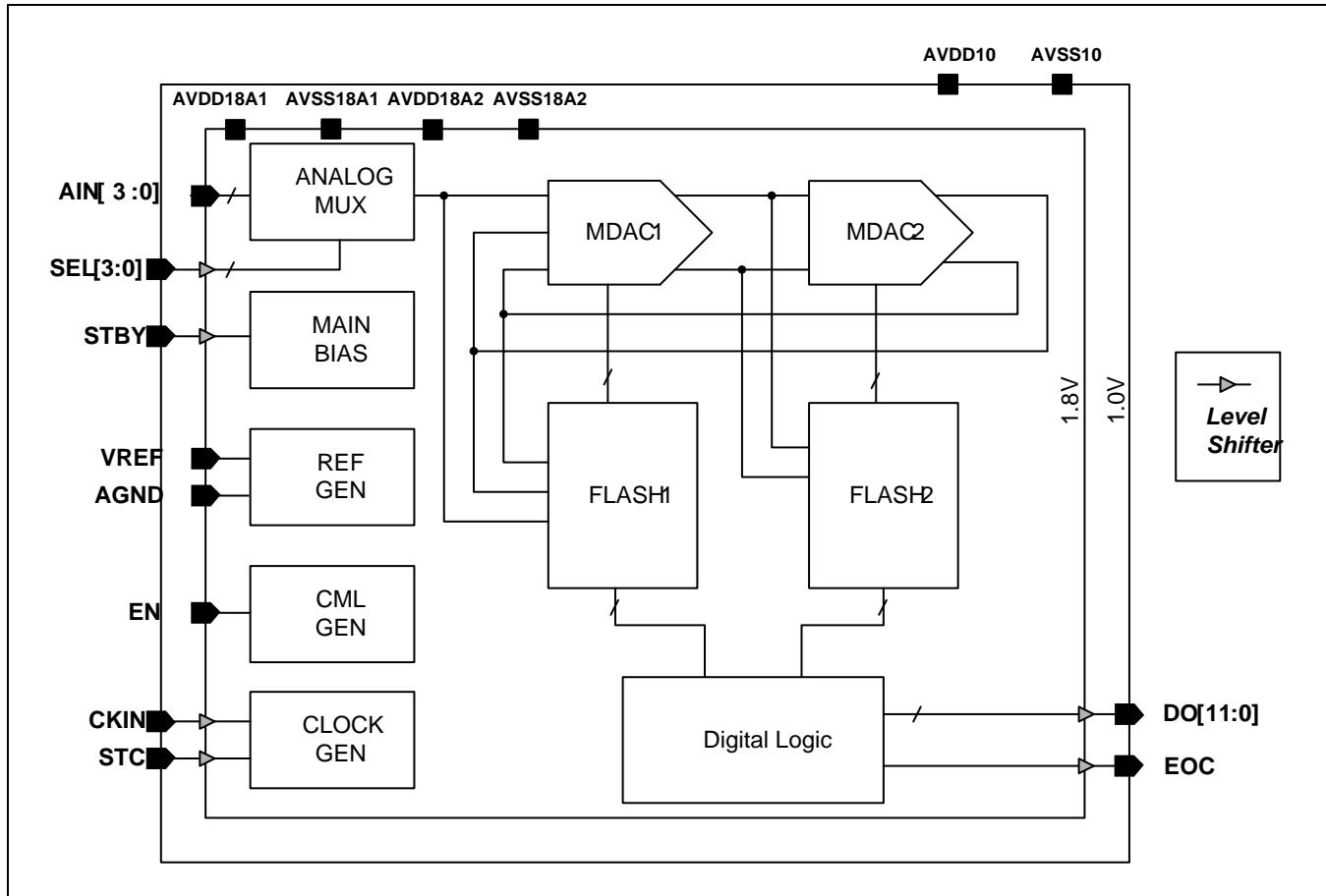


Figure 18-2 ADC Functional Block Diagram

18.3.2 ADC Selection

Exynos 4412 has two ADC blocks, General ADC and MTCADC_ISP. User can select one of ADC blocks by setting ADC_CFG[16] bit in System Register SFR.

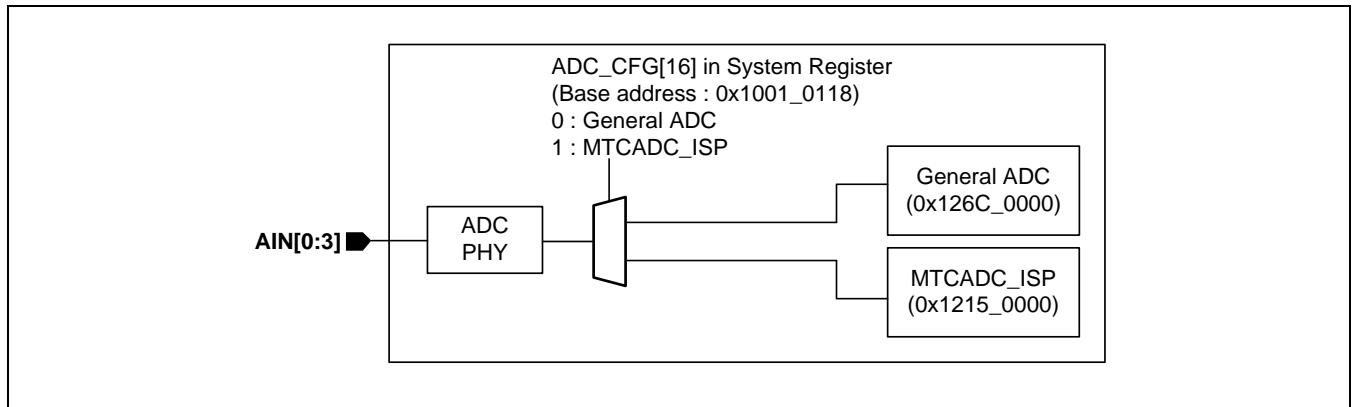


Figure 18-3 ADC Selection

18.3.3 A/D Conversion Time

When the APB bus clock (PCLK) frequency is 66MHz and the prescaler value is 65, total 12-bit conversion time is as follows.

- A/D converter freq. = $66 \text{ MHz}/(65 + 1) = 1 \text{ MHz}$
- Conversion time = $1/(1 \text{ MHz}/5 \text{ cycles}) = 1/200 \text{ kHz} = 5 \mu\text{s}$

NOTE: This A/D converter was designed to operate at maximum 5MHz clock, so the conversion rate can go up to 1MSPS.

18.3.4 ADC Conversion Mode

The operation of this mode is same as AIN0 to AIN3's. To initialize this mode, set the ADCCON (ADC control register). The converted data can be read out from ADCDAT (ADC conversion data register).

18.3.5 Standby Mode

Standby mode is activated when TSSEL bit is "0" and STANDBY bit is "1" in TSADCCON0 register. In this mode, A/D conversion operation is halted and TSDATXn registers hold their values.

18.3.5.1 Programming Notes

1. The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method, the overall conversion time - from A/D converter start to converted data read-may be delayed because of the return time of interrupt service routine and data access time. With polling method, to determine the read time for ADCDATXn register, check the ADCCONn[15]-end of conversion flag-bit.
2. A/D conversion can be activated in different way. After ADCCONn[1]-A/D conversion start-by-read mode-is set to 1. A/D conversion starts simultaneously when converted data is read.

18.4 ADC Input Clock Diagram

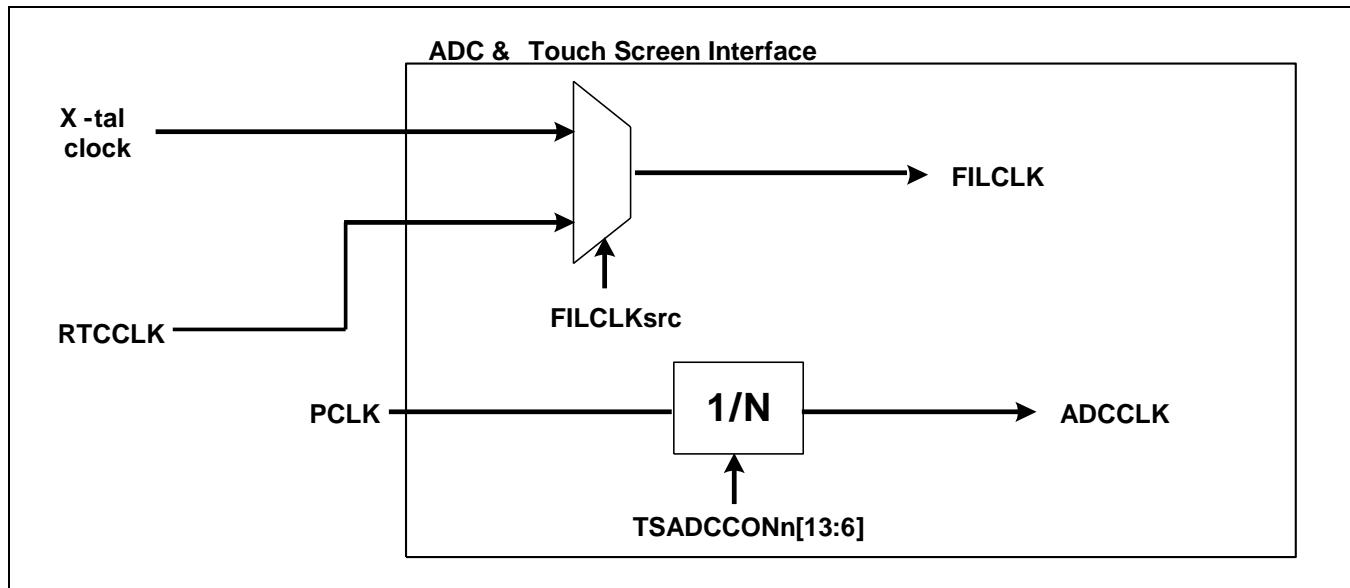


Figure 18-4 Input Clock Diagram for ADC & Touch Screen Interface

18.5 I/O Descriptions

| Signal | I/O | Description | Pad | Type |
|--------|-------|-----------------------------|------------|--------|
| AIN[3] | Input | ADC Channel[3] Analog input | Xadc1AIN_3 | Analog |
| AIN[2] | Input | ADC Channel[2] Analog input | Xadc1AIN_2 | Analog |
| AIN[1] | Input | ADC Channel[1] Analog input | Xadc1AIN_1 | Analog |
| AIN[0] | Input | ADC Channel[0] Analog input | Xadc1AIN_0 | Analog |

18.6 Register Description

18.6.1 Register Map Summary

- Base Address: 0x126C_0000

| Register | Offset | Description | Reset Value |
|-----------|--------|--|-------------|
| ADCCON | 0x0000 | ADC Control Register | 0x0000_3FC4 |
| ADCDLY | 0x0008 | ADC Start or Interval Delay Register | 0x0000_00FF |
| ADCDAT | 0x000C | ADC Conversion Data Register | Undefined |
| CLRINTADC | 0x0018 | Clear ADC Interrupt | Undefined |
| ADCMUX | 0x001C | Specifies the Analog input channel selection | 0x0000_0000 |

18.6.1.1 ADCCON

- Base Address: 0x126C_0000
- Address = Base Address + 0x0000, Reset Value = 0x0000_3FC4

| Name | Bit | Type | Description | Reset Value |
|--------------|--------|------|---|-------------|
| RES | [16] | RW | ADC output resolution selection 0 = 10-bit A/D conversion 1 = 12-bit A/D conversion | 0 |
| ECFLG | [15] | RW | End of conversion flag (Read only) 0 = A/D conversion in process 1 = End of A/D conversion | 0 |
| PRSCEN | [14] | RW | A/D converter prescaler enable 0 = Disable 1 = Enable | 0 |
| PRSCVL | [13:6] | RW | A/D converter prescaler value Data value: 19 to 255 The division factor is $(N + 1)$ when the prescaler value is N. For example, ADC frequency is 5 MHz if APB bus clock is 100 MHz and the prescaler value is 19. NOTE: This A/D converter is designed to operate at maximum 5 MHz clock, so the prescaler value should be set such that the resulting clock does not exceed 5 MHz. | 0xFF |
| RSVD | [5:3] | — | Reserved | 0 |
| STANDBY | [2] | RW | Standby mode select 0 = Normal operation mode 1 = Standby mode NOTE: In standby mode, prescaler should be disabled to reduce more leakage power consumption. | 1 |
| READ_START | [1] | RW | A/D conversion start by read 0 = Disables start by read operation 1 = Enables start by read operation | 0 |
| ENABLE_START | [0] | RW | A/D conversion starts by enable. If READ_START is enabled, this value is not valid. 0 = No operation 1 = A/D conversion starts and this bit is automatically cleared after the start-up. | 0 |

18.6.1.2 ADCDLY

- Base Address: 0x126C_0000
- Address = Base Address + 0x0008, Reset Value = 0x0000_00FF

| Name | Bit | Type | Description | Reset Value |
|-----------|--------|------|---|-------------|
| FILCLKsrc | [16] | RW | Reference clock source for delay. 0 = X-tal clock. 1 = RTC clock. | 0 |
| DELAY | [15:0] | RW | In case of ADC conversion mode (Normal, Separate, Auto conversion); ADC conversion is delayed by counting this value. Counting clock is PCLK. → ADC conversion delay value. In case of waiting for Interrupt mode: When stylus down occurs in waiting for interrupt mode, it generates interrupt signal (INT_PENn) at interval of several ms for Auto X/Y position conversion. If this interrupt occurs in STOP mode, it generates Wake-Up signal, having interval (several ms), for Exiting STOP MODE. NOTE: Do not use zero value (0x0000) | 00ff |

Before ADC conversion, Touch screen uses X-tal clock.

During ADC conversion PCLK (Max. 66 MHz) is used.

18.6.1.3 ADCDAT

- Base Address: 0x126C_0000
- Address = Base Address + 0x000C, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|------|--------|------|--|-------------|
| DATA | [11:0] | R | ADC conversion data value Data value: 0x0 to 0xFFFF | - |

18.6.1.4 CLRINTADC

- Base Address: 0x126C_0000
- Address = Base Address + 0x0018, Reset Value = Undefined

| Name | Bit | Type | Description | Reset Value |
|-----------|-----|------|--|-------------|
| INTADCCLR | [0] | W | INT_ADCn interrupt clear. Cleared if any value is written. | - |

These registers are used to clear the interrupts. Interrupt service routine is responsible to clear interrupts after the interrupt service is completed. Writing any values on this register will clear up the relevant interrupts asserted. When it is read, undefined value will be returned

18.6.1.5 ADCMUX

- Base Address: 0x126C_0000
- Address = Base Address + 0x001C, Reset Value = 0x0000_0000

| Name | Bit | Type | Description | Reset Value |
|---------|-------|------|---|-------------|
| SEL_MUX | [3:0] | RW | Analog input channel select 0000 = AIN 0 0001 = AIN 1 0010 = AIN 2 0011 = AIN 3 | 0 |