|  |  |  |
| --- | --- | --- |
|  | |  |
| **用户指南** | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| wujian100\_open Userguide | | |
| Number |  | |
| **Revision** | 1.0 | |
| **Security** | Public | |
| **Date** | 2019-10-15 | |



|  |  |
| --- | --- |
| Copyright © 2019 T-HEAD Semiconductor Co.,Ltd. All rights reserved.  This document is the property of T-HEAD Semiconductor Co.,Ltd. This document may only be distributed to: (i) a T-HEAD party having a legitimate business need for the information contained herein, or (ii) a non-T-HEAD party having a legitimate business need for the information contained herein. No license, expressed or implied, under any patent, copyright or trade secret right is granted or implied by the conveyance of this document. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of T-HEAD Semiconductor Co.,Ltd.  Trademarks and Permissions  The T-HEAD Logo and all other trademarks indicated as such herein are trademarks of T-HEAD Semiconductor Co.,Ltd. All other products or service names are the property of their respective owners.  Notice  The purchased products, services and features are stipulated by the contract made between T-HEAD and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.  The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.  Copyright © 2019 平头哥半导体有限公司，保留所有权利.  本文档的所有权及知识产权归属于平头哥半导体有限公司及其关联公司(下称“平头哥”)。本文档仅能分派给：(i)拥有合法雇佣关系，并需要本文档的信息的平头哥员工，或(ii)非平头哥组织但拥有合法合作关系，并且其需要本文档的信息的合作方。对于本文档，未经平头哥半导体有限公司明示同意，则不能使用该文档。在未经平头哥半导体有限公司的书面许可的情形下，不得复制本文档的任何部分，传播、转录、储存在检索系统中或翻译成任何语言或计算机语言。  商标申明  平头哥的LOGO和其它所有商标归平头哥半导体有限公司及其关联公司所有，未经平头哥半导体有限公司的书面同意，任何法律实体不得使用平头哥的商标或者商业标识。  注意  您购买的产品、服务或特性等应受平头哥商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，平头哥对本文档内容不做任何明示或默示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。平头哥半导体有限公司不对任何第三方使用本文档产生的损失承担任何法律责任。 | |
| 平头哥半导体有限公司 T-HEAD Semiconductor Co.,LTD | |
| 地址: | 杭州市西湖区西斗门路3号天堂软件园A座15楼 邮编: 310012 |
| 网址: | www.T-head.cn |

版本历史(Revisions)

| Rev | Description | Author(s) | Date |
| --- | --- | --- | --- |
| 1.0 | Initial verision | T-Head | 2019/10/15 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录(Contents)

[1 System Overview 7](#_Toc22058909)

[1.1 Functional Features 7](#_Toc22058910)

[1.2 Address Map 8](#_Toc22058911)

[1.2.1 Peripheral Address Map 9](#_Toc22058912)

[1.3 PAD I/O 12](#_Toc22058913)

[1.4 Interrupt source 15](#_Toc22058914)

[2 Timer （TIM） 18](#_Toc22058915)

[2.1 Register Description 18](#_Toc22058916)

[2.1.1 Register Memory map 18](#_Toc22058917)

[2.1.2 Register Field Description 19](#_Toc22058918)

[2.2 Work Flow 23](#_Toc22058919)

[3 Direct Memory Access （DMA） 24](#_Toc22058920)

[3.1 Overview 24](#_Toc22058921)

[3.2 Function Desription 24](#_Toc22058922)

[3.3 Register description 24](#_Toc22058923)

[3.3.1 DMA channel base address 24](#_Toc22058924)

[3.3.2 Address map 25](#_Toc22058925)

[3.3.3 Register field description 27](#_Toc22058926)

[3.4 Work Flow 34](#_Toc22058927)

[4 Unified Serial Interface (USI) 35](#_Toc22058928)

[4.1 Overview 35](#_Toc22058929)

[4.2 PAD ports mapping 35](#_Toc22058930)

[4.3 Register 35](#_Toc22058931)

[5 Watchdog （WDT） 51](#_Toc22058932)

[5.1 Register Description 51](#_Toc22058933)

[5.1.1 Register Memory map 51](#_Toc22058934)

[5.1.2 Register Field Description 52](#_Toc22058935)

[5.2 Work Flow 55](#_Toc22058936)

[6 Pulse Width Modulation （PWM） 56](#_Toc22058937)

[6.1 Register Description 56](#_Toc22058938)

[6.1.1 Register Memory Map 56](#_Toc22058939)

[6.1.2 Register Field Description 59](#_Toc22058940)

[6.2 Work Flow 105](#_Toc22058941)

[7 Real-Time Clock （RTC） 106](#_Toc22058942)

[7.1 Register Description 106](#_Toc22058943)

[7.1.1 Register Memory Map 106](#_Toc22058944)

[7.1.2 Registers Field Description 107](#_Toc22058945)

[7.2 Work Flow 110](#_Toc22058946)

[8 General-purpose I/O （GPIO） 112](#_Toc22058947)

[8.1 Register description 112](#_Toc22058948)

[8.1.1 Register Memory Map 112](#_Toc22058949)

[8.1.2 Registers Filed Description 112](#_Toc22058950)

[8.1.3 Work Flow 117](#_Toc22058951)

表格目录(Tables)

[**Table 1-1Memory Address Map** 8](#_Toc22058952)

[**Table 1-2Peripheral Address Map** 8](#_Toc22058953)

[**Table 1-3PIN List** 11](#_Toc22058954)

[**Table 1-4 Interrupt source** 14](#_Toc22058955)

[**Table 2‑2 Timers Memory Map** 17](#_Toc22058956)

[**Table 2‑3 Timer1LoadCount Field Description** 18](#_Toc22058957)

[Table 2‑4 Timer1CurrentValue Register Field Description 18](#_Toc22058958)

[Table 2‑5 Timer1Control Register Field Description 18](#_Toc22058959)

[Table 2‑6 Timer1 int clr Field Description 19](#_Toc22058960)

[Table 2‑7 Timer1Int Status Field Description 19](#_Toc22058961)

[Table 2‑8 Timer2LoadCount Field Description 20](#_Toc22058962)

[Table 2‑9 Timer2CurrentValue Register Field Description 20](#_Toc22058963)

[Table 2‑10 Timer2 Control Register Field Description 20](#_Toc22058964)

[Table 2‑11 Timer2 int clr Field Description 21](#_Toc22058965)

[Table 2‑12 Timer2Int Status Field Description 22](#_Toc22058966)

[Table 3‑1 DMA Memory Map for one channel 25](#_Toc22058967)

[Table 6–1 PWMCFG Field Description 58](#_Toc22058968)

[Table 6–2 PWMINVERTTRIG Field Description 61](#_Toc22058969)

[Table 6–3 PWM01TRIG Field Description 62](#_Toc22058970)

[Table 6–4 PWM23TRIG Field Description 62](#_Toc22058971)

[Table 6–5 PWM45TRIG Field Description 63](#_Toc22058972)

[Table 6–6 PWMNTEN1 Field Description 63](#_Toc22058973)

[Table 6–7 PWMINTEN2 Field Description 66](#_Toc22058974)

[Table 6–8 PWMRIS1 Field Description 69](#_Toc22058975)

[Table 6–9 PWMRIS2 Field Description 71](#_Toc22058976)

[Table 6–10 PWMIC1 Field Description 74](#_Toc22058977)

[Table 6–11 PWMIC2 Field Description 77](#_Toc22058978)

[Table 6–12 PWMIS1 Field Description 79](#_Toc22058979)

[Table 6–13 PWMIS2 Field Description 82](#_Toc22058980)

[Table 6–14 PWMCTL Field Description 84](#_Toc22058981)

[Table 6–15 PWMnmLOAD Field Description 86](#_Toc22058982)

[Table 6–16 PWMnmCOUNT Field Description 87](#_Toc22058983)

[Table 6–17 PWMnCMP Field Description 87](#_Toc22058984)

[Table 6–18 PWMnmDB Field Description 88](#_Toc22058985)

[Table 6–19 CAPCTL Field Description 89](#_Toc22058986)

[Table 6–20 CAPINTEN Field Description 91](#_Toc22058987)

[Table 6–21 CAPRIS Field Description 93](#_Toc22058988)

[Table 6–22 CAPIC Field Description 94](#_Toc22058989)

[Table 6–23 CAPIS Field Description 96](#_Toc22058990)

[Table 6–24 CAPnmT Field Description 98](#_Toc22058991)

[Table 6–25 CAPnmMATCH Field Description 99](#_Toc22058992)

[Table 6–26 TIMINTEN Field Description 99](#_Toc22058993)

[Table 6–27 TIMRIS Field Description 100](#_Toc22058994)

[Table 6–28 TIMIC Field Description 101](#_Toc22058995)

[Table 6–29 TIMIS Field Description 102](#_Toc22058996)

[Table 6–30 TIMnmLOAD Field Description 103](#_Toc22058997)

[Table 6–31 TIMnmCOUNT Field Description 103](#_Toc22058998)

[Table 6–32 CNTnmVAL Field Description 104](#_Toc22058999)

[**Table 8‑1 Address Map of GPIO Registers** 111](#_Toc22059000)

[**Table 8‑2 gpio\_output\_data Field Description** 112](#_Toc22059001)

[**Table 8‑3 gpio\_direction Field Description** 112](#_Toc22059002)

[**Table 8‑4 gpio\_ctl Field Description** 112](#_Toc22059003)

[**Table 8‑5 gpio\_inten Field Description** 113](#_Toc22059004)

[**Table 8‑6 gpio\_intmask Field Description** 113](#_Toc22059005)

[**Table 8‑7 gpio\_inttype\_level Field Description** 114](#_Toc22059006)

[**Table 8‑8 gpio\_int\_polarity Field Description** 114](#_Toc22059007)

[**Table 8‑9 gpio\_intstatus Field Description** 115](#_Toc22059008)

[**Table 8‑10 gpio\_rawintstatus Field Description** 115](#_Toc22059009)

[**Table 8‑11 gpio\_int\_clr Field Description** 116](#_Toc22059010)

[**Table 8‑12 gpio\_input\_data Field Description** 116](#_Toc22059011)

术语与参考资料(Abbreviations and Reference)

**术语与缩略语**

|  |  |  |
| --- | --- | --- |
| Abbreviations 缩略语 | Full spelling  英文全名 | Chinese explanation 中文解释 |
|  |  |  |
|  |  |  |

**参考资料**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 文档编号 | 标题 | 版本 | 日期 |
|  |  |  |  |  |
|  |  |  |  |  |

# System Overview

## Functional Features

* **CPU Part**
  + **32-bit general purpose CPU E902**
  + **CoreTim**
    - The circulating decrement counter is 24-bit count width
  + **VIC**
    - Supports 64 Interrupts nesting
    - Each interrupt has corresponding priority
* **Memory**
  + 64KB ISRAM and 3×64KB DSRAM on-chip SRAM
* **Characteristics of Peripherals**
  + **DMAC(x1)**
    - 16 channels
    - Supports block/group trigger mode transactions
  + **TIM (×8)**
    - connects to APB1 and APB0
    - All of timers are 32-bit count width
    - All of timers support for two operation modes: free-running and user-defined count
  + **GPIO (×1)**
    - Connects to APB
    - GPIO supports 32-bit width;
    - Each bit of GPIO supports interrupt generation
  + **USI (×3)**
    - Universal Asynchronous Receiver/Transmitter(UART)
    - Inter-integrated Circuit(I2C)
    - Serial Peripheral Interface(SPI)
  + **RTC** 
    - RTC connects to APB
    - 32-bit count width
    - Incrementing counter and comparator for interrupt generation
  + **PWM**
    - Connects to APB
    - 12 input/output channels
    - 6 PWM generators, each with 1 32-bit counter, 2 PWM comparator, 1 PWM signal generator, 1 interrupt generator
    - Each PWM signal generator contains 2 channels
    - PWM output enable or disable of each PWM signal
    - Optional output inversion of each PWM signal (polarity control)
    - 6 32-bit counter and each has the following characteristics:
      * Up or Up/Down mode
      * Output frequency controlled by a 16-bit load value
    - 32-bit input capture modes
      * Input edge count mode
      * Input edge time mode
  + **WDT**
    - One WDT in APB
    - Each WDT has 32 count width
    - Counter counts down from a pre-set value to zero to indicate the occurrence of a timeout

## Address Map

For this chip, the memory address is divided into several parts: on-chip memory, off-chip memory, TCIP and peripherals. The detailed address map is shown as below.

|  |  |  |
| --- | --- | --- |
| Address Range | Size | Usage |
| 0x0000\_0000~0x0000\_7FFF | 64KB | Internal SRAM（inst） |
| 0x2000\_0000~0x2002\_FFFF | 192KB | Internal SRAM(data) |
| 0x4000\_0000~0x401F\_FFFF | 2MB | MAIN BUS Peripherals |
| 0x4020\_0000~0x0x7FFF\_FFFF | 1024M-2M | Low speed Peripherals |
| 0x8000\_0000~0x9FFF\_FFFF | 768M | MAIN BUS Peripherals |
| 0xE000\_E000~0xE000\_EFFF | 4KB | TCIP |
| Other | - | Reserved |

**Table 1-1****Memory Address Map**

### Peripheral Address Map

The register blocks for all on-chip peripheral devices are located on 4KB boundaries. Those modules that may require additional address space are assigned additional 4KB blocks. Within a 4KB block, peripheral registers may be incompletely decoded. The description of each peripheral will define the result of accesses to unimplemented registers. For registers that do not implement all 32 bits, the unimplemented bits will return zero when read, and write to the unimplemented bits will have no effect. In general, unimplemented bits should return zero to ensure future compatibility.

**Table 1-2****Peripheral Address Map**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address Range | IP name | Size | Master/Slave | | Description |
|  | E902 |  | M0/M2 | | Core |
|  | DMA |  | M3 | | DMA |
|  | MDummy0 |  | M4 | | main\_mdummy\_top0 |
|  | MDummy1 |  | M5 | | main\_mdummy\_top1 |
|  | MDummy2 |  | M6 | | main\_mdummy\_top2 |
| 0x0000\_0000~0x0000\_27FF | ISRAM | 64K | S0 | | ROM |
| 0x1000\_0000~0x1007\_FFFF | MemDummy | 512K | S1 | | instmem\_dummy top0 |
| 0x2000\_0000~0x2000\_FFFF | SRAM | 64K | S2 | | DATA SRAM |
| 0x2001\_0000~0x2001\_FFFF | SRAM | 64K | S3 | | DATA SRAM |
| 0x2002\_0000~0x2002\_FFFF | SRAM | 64K | S4 | | DATA SRAM |
| 0x3000\_0000~0x3007\_FFFF | MemDummy | 512K | S5 | | datamem\_dummy\_top1 |
| 0x4000\_0000~0x4000\_3FFF | DMA | 16K | S6 | | DMA Controller |
| 0x4001\_0000~0x4001\_FFFF | Dummy | 64K | S7 | | main\_dummy\_top0 |
| 0x4002\_0000~0x4002\_FFFF | Dummy | 64K | S8 | | main\_dummy\_top1 |
| 0x4010\_0000~0x401F\_FFFF | Dummy | 1M | S9 | | main\_dummy\_top2 |
| 0x4020\_0000~0x7FFF\_FFFF | LSBUS | 1024M-2M | S10 | | AHB LS BUS |
| 0x8000\_0000~0x9FFF\_FFFF | Dummy | 512M | S11 | | main\_dummy\_top3 |
| Other | REV | - | REV | | Reserved |
| AHB LS BUS | | | | | |
| 0x4020\_0000~0x4020\_0FFF | Dummy | 4K | S0 | | lsbus\_dummy\_top0 |
| 0x4030\_0000~0x403F\_FFFF | Dummy | 1M | S1 | | lsbus\_dummy\_top1 |
| 0x5000\_0000~0x5004\_FFFF | APB0 | 320K | S2 | | APB0 |
| 0x6000\_0000~0x6004\_FFFF | APB1 | 320K | S3 | | APB1 |
| 0x7000\_0000~0x77FF\_FFFF | Dummy | 128M | S4 | | lsbus\_dummy\_top2 |
| 0x7800\_0000~0x7FFF\_FFFF | Dummy | 128M | S5 | | lsbus\_dummy\_top3 |
| APB0 | | | | | |
| 0x5000\_0000~0x5000\_03FF | TIM0 | 1K | | P0 | Timer 0 |
| 0x5000\_0400~0x5000\_07FF | TIM2 | 1K | | P1 | Timer 2 |
| 0x5000\_0800~0x5000\_0BFF | TIM4 | 1K | | P2 | Timer 4 |
| 0x5000\_0C00~0x5000\_0FFF | TIM6 | 1K | | P3 | Timer 6 |
| 0x5002\_8000~0x5002\_8FFF | USI0 | 16K | | P4 | USI0 |
| 0x5002\_9000~0x5002\_9FFF | USI2 | 16K | | P5 | USI2 |
| 0x5000\_4000~0x5000\_7FFF | Dummy | 16K | | P6 | apb0\_dummy\_top1 |
| 0x5000\_8000~0x5000\_BFFF | WDT | 16K | | P7 | WDT |
| 0x5000\_C000~0x5000\_FFFF | Dummy | 16K | | P8 | apb0\_dummy\_top2 |
| 0x5001\_0000~0x5001\_3FFF | Dummy | 16K | | P9 | apb0\_dummy\_top3 |
| 0x5001\_4000~0x5001\_7FFF | Dummy | 16K | | P10 | apb0\_dummy\_top4 |
| 0x5001\_8000~0x5001\_BFFF | Dummy | 16K | | P11 | apb0\_dummy\_top5 |
| 0x5001\_C000~0x5001\_FFFF | PWM | 16K | | P12 | PWM |
| 0x5002\_0000~0x5002\_3FFF | Dummy | 16K | | P13 | apb0\_dummy\_top7 |
| 0x5002\_4000~0x5002\_7FFF | Dummy | 16K | | P14 | apb0\_dummy\_top8 |
| 0x5003\_0000~0x5003\_3FFF | Dummy | 16K | | P15 | apb0\_dummy\_top9 |
| APB1 | | | | | |
| 0x6000\_0000~0x6000\_03FF | TIM1 | 1K | | P0 | Timer 1 |
| 0x6000\_0400~0x6000\_07FF | TIM3 | 1K | | P1 | Timer 3 |
| 0x6000\_0800~0x6000\_0BFF | TIM5 | 1K | | P2 | Timer 5 |
| 0x6000\_0C00~0x6000\_0FFF | TIM7 | 1K | | P3 | Timer 7 |
| 0x6002\_8000~0x6002\_BFFF | USI1 | 16K | | P4 | USI1 |
| 0x6001\_8000~0x6001\_BFFF | GPIO | 16K | | P5 | GPIO |
| 0x6000\_4000~0x6000\_7FFF | RTC | 16K | | P6 | RTC |
| 0x6000\_8000~0x6000\_BFFF | Dummy | 16K | | P7 | apb1\_dummy\_top1 |
| 0x6000\_C000~0x6000\_FFFF | Dummy | 16K | | P8 | apb1\_dummy\_top2 |
| 0x6001\_0000~0x6001\_3FFF | Dummy | 16K | | P9 | apb1\_dummy\_top3 |
| 0x6001\_4000~0x6001\_7FFF | Dummy | 16K | | P10 | apb1\_dummy\_top4 |
| 0x6001\_C000~0x6001\_FFFF | Dummy | 16K | | P11 | apb1\_dummy\_top5 |
| 0x6002\_0000~0x6002\_3FFF | Dummy | 16K | | P12 | apb1\_dummy\_top6 |
| 0x6002\_4000~0x6002\_7FFF | Dummy | 16K | | P13 | apb1\_dummy\_top7 |
| 0x6002\_C000~0x6002\_FFFF | Dummy | 16K | | P14 | apb1\_dummy\_top8 |
| 0x6003\_0000~0x6003\_3FFF | PMU | 16K | | P15 | Power Management  Unit(dummy) |

## PAD I/O

Each of IO can be shared by less than 4 peripherals at one time. The details of AFIO function as follows.

**Table 1-3****PIN List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num | Pin name | I/O | Width | Description |
| 1 | PIN\_EHS | I | 1 | External high speed osc clock |
| 2 | POUT\_EHS | O | 1 |
| 3 | PIN\_ELS | I | 1 | External low speed osc clock |
| 4 | POUT\_ELS | O | 1 |
| 5 | PAD\_MCURST | I | 1 | PAD system reset，active low |
| 6 | PAD\_JTAG\_TCLK | I | 1 | CPU JTAG TCLK |
| 7 | PAD\_JTAG\_TMS | I/O | 1 | CPU JTAG TMS |
| 8 | PAD\_GPIO\_0 | I/O | 1 | GPIO0\_0 signal |
| 9 | PAD\_GPIO\_1 | I/O | 1 | GPIO0\_1 signal |
| 10 | PAD\_GPIO\_2 | I/O | 1 | GPIO0\_2 signal |
| 11 | PAD\_GPIO\_3 | I/O | 1 | GPIO0\_3 signal |
| 12 | PAD\_GPIO\_4 | I/O | 1 | GPIO0\_4 signal |
| 13 | PAD\_GPIO\_5 | I/O | 1 | GPIO0\_5 signal |
| 14 | PAD\_GPIO\_6 | I/O | 1 | GPIO0\_6 signal |
| 15 | PAD\_GPIO\_7 | I/O | 1 | GPIO0\_7 signal |
| 16 | PAD\_GPIO\_8 | I/O | 1 | GPIO0\_8 signal |
| 17 | PAD\_GPIO\_9 | I/O | 1 | GPIO0\_9 signal |
| 18 | PAD\_GPIO\_10 | I/O | 1 | GPIO0\_10 signal |
| 19 | PAD\_GPIO\_11 | I/O | 1 | GPIO0\_11 signal |
| 20 | PAD\_GPIO\_12 | I/O | 1 | GPIO0\_12 signal |
| 21 | PAD\_GPIO\_13 | I/O | 1 | GPIO0\_13 signal |
| 22 | PAD\_GPIO\_14 | I/O | 1 | GPIO0\_14 signal |
| 23 | PAD\_GPIO\_15 | I/O | 1 | GPIO0\_15 signal |
| 24 | PAD\_GPIO\_16 | I/O | 1 | GPIO0\_16 signal |
| 25 | PAD\_GPIO\_17 | I/O | 1 | GPIO0\_17 signal |
| 26 | PAD\_GPIO\_18 | I/O | 1 | GPIO0\_18 signal |
| 27 | PAD\_GPIO\_19 | I/O | 1 | GPIO0\_19 signal |
| 28 | PAD\_GPIO\_20 | I/O | 1 | GPIO0\_20 signal |
| 29 | PAD\_GPIO\_21 | I/O | 1 | GPIO0\_21 signal |
| 30 | PAD\_GPIO\_22 | I/O | 1 | GPIO0\_22 signal |
| 31 | PAD\_GPIO\_23 | I/O | 1 | GPIO0\_23 signal |
| 32 | PAD\_GPIO\_24 | I/O | 1 | GPIO0\_24 signal |
| 33 | PAD\_GPIO\_25 | I/O | 1 | GPIO0\_25 signal |
| 34 | PAD\_GPIO\_26 | I/O | 1 | GPIO0\_26 signal |
| 35 | PAD\_GPIO\_27 | I/O | 1 | GPIO0\_27 signal |
| 36 | PAD\_GPIO\_28 | I/O | 1 | GPIO0\_28 signal |
| 37 | PAD\_GPIO\_29 | I/O | 1 | GPIO0\_29 signal |
| 38 | PAD\_GPIO\_30 | I/O | 1 | GPIO0\_30 signal |
| 39 | PAD\_GPIO\_31 | I/O | 1 | GPIO0\_31 signal |
| 40 | PAD\_PWM\_CH0 | I/O | 1 | PWM channel0 signal |
| 41 | PAD\_PWM\_CH1 | I/O | 1 | PWM channel1 signal |
| 42 | PAD\_PWM\_CH2 | I/O | 1 | PWM channel2 signal |
| 43 | PAD\_PWM\_CH3 | I/O | 1 | PWM channel3 signal |
| 44 | PAD\_PWM\_CH4 | I/O | 1 | PWM channel4 signal |
| 45 | PAD\_PWM\_CH5 | I/O | 1 | PWM channel5 signal |
| 46 | PAD\_PWM\_CH6 | I/O | 1 | PWM channel6 signal |
| 47 | PAD\_PWM\_CH7 | I/O | 1 | PWM channel7 signal |
| 48 | PAD\_PWM\_CH8 | I/O | 1 | PWM channel8 signal |
| 49 | PAD\_PWM\_CH9 | I/O | 1 | PWM channel9 signal |
| 50 | PAD\_PWM\_CH10 | I/O | 1 | PWM channel10 signal |
| 51 | PAD\_PWM\_CH11 | I/O | 1 | PWM channel11 signal |
| 52 | PAD\_USI0\_NSS | I/O | 1 | USI0 Slave select |
| 53 | PAD\_USI0\_SCLK | I/O | 1 | USI0 Serial clock |
| 54 | PAD\_USI0\_SD0 | I/O | 1 | USI0 Serial data0 |
| 55 | PAD\_USI0\_SD1 | I/O | 1 | USI0 Serial data1 |
| 56 | PAD\_USI1\_NSS | I/O | 1 | USI1 Slave select |
| 57 | PAD\_USI1\_SCLK | I/O | 1 | USI1 Serial clock |
| 58 | PAD\_USI1\_SD0 | I/O | 1 | USI1 Serial data0 |
| 59 | PAD\_USI1\_SD1 | I/O | 1 | USI1 Serial data1 |
| 60 | PAD\_USI2\_NSS | I/O | 1 | USI2 Slave select |
| 61 | PAD\_USI2\_SCLK | I/O | 1 | USI2 Serial clock |
| 62 | PAD\_USI2\_SD0 | I/O | 1 | US2 Serial data0 |
| 63 | PAD\_USI2\_SD1 | I/O | 1 | US2 Serial data1 |

## Interrupt source

**Table 1-4** **Interrupt source**

|  |  |
| --- | --- |
| Number | Interrupt Source |
| 6:0 | REV |
| 1 | CoreTim |
| 15:8 | REV |
| 16 | GPIO0 |
| 18:17 | TIM0[1:0] |
| 20:19 | TIM1[1:0] |
| 22:21 | TIM2[1:0] |
| 24:23 | TIM3[1:0] |
| 25 | PWM |
| 26 | RTC |
| 27 | WDT |
| 28 | USI0 |
| 29 | USI1 |
| 30 | USI2 |
| 31 | PMU |
| 32 | DMAC0 |
| 34:33 | TIM4[1:0] |
| 36:35 | TIM5[1:0] |
| 38:37 | TIM6[1:0] |
| 40:39 | TIM7[1:0] |
| 41 | IMEMDUMMY0 |  DMEMDUMMY0 |
| 42 | MAIN\_DUMMY0 |
| 43 | MAIN\_DUMMY1 |
| 44 | MAIN\_DUMMY2 |
| 45 | MAIN\_DUMMY3 |
| 46 | LSBUS\_DUMMY0 |
| 47 | LSBUS\_DUMMY1 |
| 48 | LSBUS\_DUMMY2 |
| 49 | LSBUS\_DUMMY3 |
| 50 | APB0\_DUMMY1 |
| 51 | APB0\_DUMMY2 |
| 52 | APB0\_DUMMY3 |
| 53 | APB0\_DUMMY4 |
| 54 | APB0\_DUMMY5 |
| 55 | APB0\_DUMMY7 |
| 56 | APB0\_DUMMY8 |
| 57 | APB0\_DUMMY9 |
| 58 | APB1\_DUMMY1 |
| 59 | APB1\_DUMMY2 |
| 60 | APB1\_DUMMY3 |
| 61 | APB1\_DUMMY4 |
| 62 | APB1\_DUMMY5 |  APB1\_DUMMY6 |
| 63 | APB1\_DUMMY7 |  APB1\_DUMMY8 |

# Timer （TIM）



## Register Description

The following tables describe the memory map of the register in the Timers.

### Register Memory map

In each TIMER IP，two timers are provided:Timer1 and Timer2.Following table describes the registers address map of Timer1 and the Timer2.

**Table 2‑2** **Timers Memory Map**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Address  offset | Width | Access | Reset Value | Description |
| Timer1LoadCount | 0x00 | 32 | R/W | 32’b0 | Value to be loaded into Timer1. |
| Timer1CurrentValue | 0x04 | 32 | R | 32’b0 | Current Value of Timer1. |
| Timer1Control  Reg | 0x08 | 4 | R/W | 4’b0 | Control Register for Timer1. |
| Timer1\_int\_clr | 0x0C | 1 | R | 1’b0 | Clears the interrupt from Timer1. |
| Timer1Int Status | 0x10 | 1 | R | 1’b0 | Contains the interrupt status for Timer1. |
| Timer2LoadCount | 0x14 | 32 | R/W | 32’b0 | Value to be loaded into Timer2. |
| Timer2CurrentValue | 0x18 | 32 | R | 32’b0 | Current Value of Timer2 |
| Timer2Control  Reg | 0x1c | 4 | R/W | 4’b0 | Control Register for Timer2 |
| Timer2\_int\_clr | 0x20 | 1 | R | 1’b0 | Clears the interrupt from Timer2 |
| Timer2Int Status | 0x24 | 1 | R | 1’b0 | Contains the interrupt status for Timer2 |

### Register Field Description

The following section mainly describes the Timer1 register field and the Timers system register field. Timer2 register field description just replicates that of Timer1 using the address offset specified in above table.

**Timer1 Load Count**

Address Offset: 0x00

Read/write access: read/write

**Table 2‑3** **Timer1LoadCount Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Timer1 Load Count Register | R/W | Value to be loaded into Timer1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.  Reset Value: 32’b0 |

**Timer1 Current Value**

Address Offset: 0x04

Read/write access: read

Table 2‑4 Timer1CurrentValue Register Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Timer1 Current Value Register | R | Current Value of Timer1.  Reset Value: 32’b0 |

**Timer1 Control Reg**

Address Offset: 0x08

Read/write access: read/write

Table 2‑5 Timer1Control Register Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:5 | Reserved, read as zero | | |
| 4 | Timer Hardware Trigger enable | R/W | 0 (timer hardware trigger disable)  1 (timer hardware trigger enable) |
| 3 | Reserved, read as zero | | |
| 2 | Timer Interrupt Mask | R/W | 0 (timer interrupt not masked)  1 (timer interrupt masked)  Reset Value: 1’b0 |
| 1 | Timer Mode Select | R/W | 0 (free-running)  1 (user-defined running)  Reset Value: 1’b0 |
| 0 | Timer Enable Select | R/W | 0 (disabled)  1 (enabled)  Reset Value: 1’b0 |

**Timer1\_int\_clr**

Address Offset: 0x0c

Read/write access: read

Table 2‑6 Timer1 int clr Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | Reserved, read as zero | | |
| 0 | Timer1 int clr Register | R | Reading from this register returns all zeros(0) and clears the interrupt from Timer1 |

**Timer1Int Status**

Address Offset: 0x10

Read/write access: read

Table 2‑7 Timer1Int Status Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | Reserved, read as zero | | |
| 0 | Timer1Int Status Register | R | Contains the interrupt status for Timer1 |

**Timer2 Load Count**

Address Offset: 0x14

Read/write access: read/write

Table 2‑8 Timer2LoadCount Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Timer2 Load Count Register | R/W | Value to be loaded into Timer2. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.  Reset Value: 32’b0 |

**Timer2 Current Value**

Address Offset: 0x18

Read/write access: read

Table 2‑9 Timer2CurrentValue Register Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Timer2 Current Value Register | R | Current Value of Timer2  Reset Value: 32’b0 |

**Timer2 Control Reg**

Address Offset: 0x1c

Read/write access: read/write

Table 2‑10 Timer2 Control Register Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:5 | Reserved, read as zero | | |
| 4 | Timer Hardware Trigger enable | R/W | 0 (timer hardware trigger disable)  1 (timer hardware trigger enable) |
| 3 | Reserved, read as zero | | |
| 2 | Timer Interrupt Mask | R/W | 0 (timer interrupt not masked)  1 (timer interrupt masked)  Reset Value: 1’b0 |
| 1 | Timer Mode Select | R/W | 0 (free-running)  1 (user-defined running)  Reset Value: 1’b0 |
| 0 | Timer Enable Select | R/W | 0 (disabled)  1 (enabled)  Reset Value: 1’b0 |

**Timer2 int clr**

Address Offset: 0x20

Read/write access: read

Table 2‑11 Timer2 int clr Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | Reserved, read as zero | | |
| 0 | Timer2 int clr Register | R | Reading from this register returns all zeros(0) and clears the interrupt from Timer2 |

**Timer2Int Status**

Address Offset: 0x24

Read/write access: read

Table 2‑12 Timer2Int Status Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | Reserved, read as zero | | |
| 0 | Timer2 Interrupt Status Register | R | Contains the interrupt status for Timer2.  Reset Value: 32’b0 |

## Work Flow

Normal program of the timer is as follows:

1. Disable the timer and program its operating mode by writing to its control register, TimerN Control Reg.

2. Load the timer’s Timer Load Count register, TimerN Load Count.

3. Enable the timer by writing the TimerN Control Reg register.

# Direct Memory Access （DMA）

## Overview

DMAC is compliant with AMBA2.0 AHB-lite Specification, Support max 16 channels. Every channel only supports block trigger mode.

There are mask registers to mask the interrupts and clean registers to clear the interrupts. Block transfer completion interrupt will generate after all block transfer done.

## Function Desription

* 16 channels, each channel has a source addr and a dest addr,data will be transfered from source addr to dest addr, both the source addr and the dest addr can be programed
* Support 16 priority level ,Channel 0 has highest priority than other channels.
* Support enable protect,DMA config can not be changed once enable is set to 1.
* The source and dest Address have 3 types:increase, decrease , or no change
* Programmable block transaction size for each channel with max 4096bytes
* Support source and destination transaction with different width,8/16/32bits can be selected
* Programmable enable and disable of DMA channel
* The low priority channel can be interrupt by high priority channel
* Data litte-endian or big-endian
* Interrupt，Interrupt can be generated when:

a.Block transfer completion

b.Half Block transfer completion

c.Trigger event finish

d.Error condition

Each interrupt can be masked or cleared by registers.

## Register description

### DMA channel base address

Include 17 base address:（16 Channel registers and 1 global registers ）

Channel 0 registers include registers(internal base offset address is:0x0)： SAR0 ，DAR0 ，CH0\_CTRL\_A ，CH0\_CTRL\_B ，CH0\_INT\_MASK ，CH0\_INT\_STATUS ， CH0\_INT\_CLEAR ，CH0\_SOFT\_REQ ，CH0\_EN

Channel 1 registers include registers(internal base offset address is:0x30)：SAR1 ，DAR1 ，CH1\_CTRL\_A ，CH1\_CTRL\_B ，CH1\_INT\_MASK ，CH1\_INT\_STATUS ， CH1\_INT\_CLEAR ，CH1\_SOFT\_REQ ，CH1\_EN

......

Channel n registers include registers(internal base offset address is:0x30\*n)：SARn ，DARn ，CHn\_CTRL\_A ，CHn\_CTRL\_B ，CHn\_INT\_MASK ，CHn\_INT\_STATUS ， CHn\_INT\_CLEAR ，CHn\_SOFT\_REQ ，CHn\_EN

......

Channel 15 registers include registers(internal base offset address is:0x2D0)：SAR15 ，DAR15 ，CH15\_CTRL\_A ，CH15\_CTRL\_B ，CH15\_INT\_MASK ，CH15\_INT\_STATUS ， CH15\_INT\_CLEAR ，CH15\_SOFT\_REQ ，CH15\_EN

16 channels internal offset address is the same.

Global register (internal base offset address is :0x330):CHSR,DMACCFG

|  |  |  |  |
| --- | --- | --- | --- |
| Channel 0 base address | 0x0 | Channel 1 base address | 0x30 |
| Channel 2 base address | 0x60 | Channel 3 base address | 0x90 |
| Channel 4 base address | 0xC0 | Channel 5 base address | 0xF0 |
| Channel 6 base address | 0x120 | Channel 7 base address | 0x150 |
| Channel 8 base address | 0x180 | Channel 9 base address | 0x1B0 |
| Channel 10 base address | 0x1E0 | Channel 11 base address | 0x210 |
| Channel 12 base address | 0x240 | Channel 13 base address | 0x270 |
| Channel 14 base address | 0x2A0 | Channel 15 base address | 0x2D0 |
| Global register base address | 0x330 | - | - |

### Address map

The following table describe the register memory map for one channel

Table 3‑1 DMA Memory Map for one channel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Register | Offset | Width | Describe | Reset Value |
| SARn | 0x0 | 32 bits | Channel n Source Address Register | 0x0000\_0000 |
| DARn | 0x4 | 32 bits | Channel n Destination Address Register | 0x0000\_0000 |
| CHn\_CTRL\_A | 0x8 | 32 bits | Channel n Control Register A | 0x0000\_0000 |
| CHn\_CTRL\_B | 0xc | 32 bits | Channel n Control Register B | 0x0000\_0000 |
| CHn\_INT\_MASK | 0x10 | 32 bits | Channel n Mask Interrupt Register | 0x0000\_0000 |
| CHn\_INT\_STATUS | 0x14 | 32 bits | Channel n Status Interrupt Register | 0x0000\_0000 |
| CHn\_INT\_CLEAR | 0x18 | 32 bits | Channel n Clear Interrupt Register | 0x0000\_0000 |
| CHn\_SOFT\_REQ | 0x1c | 32 bits | Channel n Software Handshaking Request Register | 0x0000\_0000 |
| CHn\_EN | 0x20 | 32 bits | Channel n Enable Control Register | 0x0000\_0000 |

Every channel has a base of offset address which is described in 3.3.1.

DMAC global registers base address is 0x330,these registers are the configs or

status of all 16 channels. Such as enable, busy status or pending state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Register | Offset | Width | Describe | Reset Value |
| Reserved | 0x0 | - | - | - |
| Reserved | 0x4 | - | - | - |
| CHSR | 0x8 | 32 bits | Channel busy status register | 0x0000\_0000 |
| DMACCFG | 0xC | 32bits | DMAC configure register | 0x0000\_0000 |

### Register field description

#### Channel n Source Address Register（SARn）

Address Offset:0x0

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 31:0 | R/W | SARN[31:0] | Channel n Source read Address Register  reset value:0x0 |

#### Channel n Destination Address Register（DARn）

Address Offset: 0x4

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 31:0 | R/W | DARN[31:0] | Channel n Destination write Address Register  reset value:0x0 |

#### Channel n Control Register A（CHn\_CTRL\_A)

Address Offset: 0x8

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 31:24 | R/W | - | Reserved |
| 23:12 | R/W | BLOCK\_TL[11:0] | Block Transfer Length  writes this field before the channel is enabled in order to indicate the block Length  (It’s bytes total number :  BLOCK\_TL+1 it’s max length is 4096 bytes)  reset value:0x0 |
| 11：8 | R/W | - | Reserved |
| 7:6 | R/W | SINC[1:0] | Source Address Increment  Indicates whether to increment or decrease the source address on every transfer.  00:increment  01: decrease  1x: no change  reset value: 0x0 |
| 5:4 | R/W | DINC[1:0] | Destination Address Increment  Indicates whether to increment or decrease the destination address on every transfer.  00: increment  01: decrease  1x: no change  reset value: 0x0 |
| 3:2 | R/W | SRC\_TR\_WIDTH[1:0] | Source Transfer Width  mapped to AHB bus “hsize.”  00: 8bit  01: 16bit  10: 32bit  11：reserved  reset value: 0x0 |
| 1:0 | R/W | DST\_TR\_WIDTH[1:0] | Destination Transfer Width  mapped to AHB bus “hsize.”  00: 8bit  01: 16bit  10: 32bit  11：reserved  reset value: 0x0 |

#### Channel n Control Register B（CHn\_CTRL\_B)

Address Offset: 0xc

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 31:19 | Reserved | | |
| 18:15 | R/W | PROTCTL[3:0] | Channel n Protection Control  used PROTCTL[3:0] to drive the AHB HPROT.  PROTCTL[2]:controls the secure access.  1:secure access(It only support secure access)  0:normal access(It support secure/no secure access)  reset value:0x0 |
| 14 | R/W | DSTDTLGC | Destination Write data Little-Big Endian change control  0: little endian  1: big endian  reset value:0x0  Detail infomation can be found Function Description |
| 13 | R/W | SRCDTLGC | Source read data Little-Big Endian change control  0: little endian  1: big endian  reset value:0x0  Detail infomation can be found Function Description |
| 12:3 | Reserved | | |
| 2:1 | R/W | TRGTMDC[1:0] | Trigger transfer mode control register  2’b00:rev  2’b01:rev  2’b1x:Block trigger mode  Reset value:0x0  These bits should be set to 2’b10 or 2’b11 |
| 0 | R/W | INT\_EN | Interrupt Enable  0: all interrupt-generating sources are disabled.  1: all interrupt-generating sources are enabled.  reset value: 0x0 |

#### Channel n Mask Interrupt Register（CHn\_INT\_MASK）

Address Offset:0x10

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 3 | R/W | masktrgetcmpfr | Mask for trigger event complete Interrupt of channel n  0: mask  1: not mask  reset value: 0x0 |
| 2 | R/W | maskhtfr | Mask for transfer half complete Interrupt of channel n  0: mask  1: not mask  reset value: 0x0 |
| 1 | R/W | masktfr | Mask for transfer complete Interrupt of channel n  0: mask  1: not mask  reset value: 0x0 |
| 0 | R/W | maskErr | Mask for Transfer Error Interrupt of Channel n  0: mask  1: not mask  reset value: 0x0 |

#### Channel n Status Interrupt Register（CHn\_INT\_STATUS）

Address Offset:0x14

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 3 | R | statustrgetcmpfr | Status for Transfer Trigger event complete .  **Reset value:0x0** |
| 2 | R | statushtfr | Status for Transfer half of block length Complete Interrupt of Channel n  reset value: 0x0 |
| 1 | R | statustfr | Status for Transfer Block Length Complete Interrupt of Channel n  reset value: 0x0 |
| 0 | R | statusErr | Status for Transfer Error Interrupt of Channel n(It’s only detect AHB bus HRESP error condition)  reset value: 0x0 |

#### Channel n Clear Interrupt Register（CHn\_INT\_CLEAR）

Address Offset:0x18

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 3 | W | cleartrgetcmpfr | Clear for statustrgetcmpif  0: no clear  1: clear  reset value: 0x0 |
| 2 | W | clearhtfr | Clear for statushtfr  0: no clear  1: clear  reset value: 0x0 |
| 1 | W | cleartfr | Clear for Statustfr  0: no clear  1: clear  reset value: 0x0 |
| 0 | W | clearErr | Clear for StatusErr  0: no clear  1: clear  reset value: 0x0 |

#### Channel n Software Handshaking Request Register（CHn\_SOFT\_REQ）

Address Offset:0x1c

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 0 | W | soft\_req | Software Request Register  reset value: 0x0 |

#### Channel n Enable control register （CHn\_EN)

Address Offset:0x20

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 0 | R/W | chn\_en | Channel 0 Enable Register  When this bit set 1,the ADRn，SARn,CHANn\_CTRLA,CHANn\_CTRLB can’t be modified.  reset value: 0x0  After transfer is over,this bit will auto cleared to 0 by hardware |

#### Global registers

Channel busy status register

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 23:16 | - | - | Reserved |
| 15 | R | Ch15bsy | Channel 15 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 14 | R | Ch14bsy | Channel 14 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 13 | R | ch12bsy | Channel 13 busy ,It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 12 | R | Ch12bsy | Channel 12 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 11 | R | Ch11bsy | Channel 11 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 10 | R | ch10bsy | Channel 10 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 9 | R | Ch9bsy | Channel 9 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 8 | R | Ch8bsy | Channel 8 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 7 | R | Ch7bsy | Channel 7 busy ,It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 6 | R | Ch6bsy | Channel 6 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 5 | R | Ch5bsy | Channel 5 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 4 | R | Ch4bsy | Channel 4 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 3 | R | Ch3bsy | Channel 3 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 2 | R | ch2bsy | Channel 2 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 1 | R | ch1bsy | Channel 1 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |
| 0 | R | ch0bsy | Channel 0 busy , It owns the data bus  1 : data transfer busy valid  0 : data transfer busy invalid |

DMACCFG

|  |  |  |  |
| --- | --- | --- | --- |
| bits | R/W | Describe | |
| 31:1 | - | - | Reserved |
| 0 | R/W | DMACEN | Global DMAC enable.  After this bit is set ,all channels can be work. |

## Work Flow

1.Configure register: SARn,DARn,CHn\_CTRL\_A,CHn\_CTRL\_B ,and any other register.And set DMACCFG.DMACEN and CHn\_EN.chnen 1’b1 at last.

2.Configure soft\_req for trigger transfer.

3.Wait int flag,The CHn\_INT\_STATUS.statustfr interrupt flag will clear CHn\_EN.chn\_en to 0.

4.Afer receive CHn\_INT\_CLEAR.cleartrf clear interrupt flag.

Note:SARn,DARn,CHn\_CTRL\_A,CHn\_CTRL\_B is enable protected（use channel internal enable protect ,not DMACEN）.Before set DMACCFG.DMACEN 1’b1,user should configure it’s channel registers.

# Unified Serial Interface (USI)

## Overview

The unified serial interface (USI) module is a serial interface useful for communication with other peripheral or microcontroller devices by the serial interface in USI. There are APB, DMA interface and interrupt interface in USI, it can be integrated in SOC as an APB device easily, transfer data with DMA and interrupt to CPU by the interrupt interface. The USI module can be configured as one function of UART or I2C or SPI. The USI’s function and the value of control registers in USI cannot be changed when the USI is working, Otherwise the data may be lost. The change can take place when the USI module is disable.

## PAD ports mapping

PAD ports mapping with internal serial module’s ports:

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial Ports(PAD)** | **UART** | **I2C** | **SPI** |
| SCLK | RXD | SCL | SCK |
| SD0 | TXD | SDA | MOSI |
| SD1 | CTS |  | MISO |
| NSS | RTS |  | NSS |

## Register

Register and field descriptions

**USI\_CTRL**

* Name: USI module enable control register.
* Address offset: 0x000

All of sub modules (TX FIFO, RX FIFO, UART, I2C and SPI) are controlled by this register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 4 |  | 0x0 | RO | Reserved. |
| 3 | RX\_FIFO\_EN | 0x0 | R/W | RX FIFO enable controller.  0x0: disable RX FIFO.  0x1: enable RX FIFO. |
| 2 | TX\_FIFO\_EN | 0x0 | R/W | TX FIFO enable controller.  0x0: disable TX FIFO.  0x1: enable TX FIFO. |
| 1 | FM\_EN | 0x0 | R/W | Function modules (UART I2C SPI) enable controller.  0x0: disable function modules.  0x1: enable function modules. |
| 0 | USI\_EN | 0x0 | R/W | All modules in USI enable controller.  0x0: disable USI module.  0x1: enable USI module. |

**MODE\_SEL**

* Name: mode select control register.
* Address offset: 0x004

This register is used to select one module of three sub modules (UART I2C and SPI).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 2 |  | 0x0 | RO | Reserved. |
| 1: 0 | MODE\_SEL | 0x0 | R/W | 0x0: UART.  0x1: I2C.  0x2: SPI.  0x3: mode keep the last set value unchanged. |

**TX\_FIFO**

* Name: the transmitter FIFO
* Address offset: 0x008

The TX data is stored in this FIFO, then the USI will read from this FIFO, and send the data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 16 |  | 0x0 | RO | Reserved. |
| 15: 0 | TX\_DATA |  | WO | Transmitting data |

**RX\_FIFO**

* Name: the receiver FIFO
* Address offset: 0x008

If USI received the data, the USI will store the data into RX FIFO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 16 |  | 0x0 | RO | Reserved. |
| 15: 0 | RX\_DATA |  | RO | Received data |

**FIFO\_STA**

* Name: the status of receive FIFO
* Address offset: 0x00C

This register lists the FULL or EMTPY data counter status for TX and RX FIFO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 20 |  | 0x0 | RO | Reserved. |
| 19: 16 | RX\_NUM | 0x0 | RO | The counter indicates how many data word is stored in the RX FIFO. |
| 15: 12 |  | 0x0 | RO | Reserved. |
| 11: 8 | TX\_NUM | 0x0 | RO | The counter indicates how many data words is stored in the TX FIFO. |
| 7: 4 |  | 0x0 | RO | Reserved. |
| 3 | RX\_FULL | 0x0 | RO | 0x0: the TX FIFO is not full.  0x1: the RX FIFO is full. |
| 2 | RX\_EMPTY | 0x1 | RO | 0x0: the RX FIFO is not empty.  0x1: the RX FIFO is empty |
| 1 | TX\_FULL | 0x0 | RO | 0x0: the TX FIFO is not full  0x1: TX FIFO is full |
| 0 | TX\_EMPTY | 0x1 | RO | 0x0: the TX FIFO is not empty.  0x1: the TX FIFO is empty |

**CLK\_DIV0**

* Name: I2C SCL high counter register.
* Address offset: 0x010

**Description:** The clock divider register. In UART mode, the register set the baud rate; in I2C mode, the register sets the SCL clock high-period count; in the SPI mode, the register sets the SCK clock period count.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 24 |  | 0x0 | RO | Reserved. |
| 23: 0 | CLK\_DIV0 | 0x20 | R/W | System clock divided register.  UART: baud rate.  Baud rate = FPCLK/(16 \* (CLK\_DIV0+1))  I2C: the register sets SCL clock high-period count.  FSCL = FPCLK/(CLK\_DIV0+CLK\_DIV1+2)  SPI: divider for SCK clock.  FSCK = FPCLK/CLK\_DIV0, CLK\_DIV0 is even value |

**CLK\_DIV1**

* Name: I2C SCL low counter register.
* Address offset: 0x014

**Description:** This register is only for I2C, the register set the I2C SCL clock low-period count.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 24 |  | 0x0 | RO | Reserved. |
| 23: 0 | CLK\_DIV1 | 0x30 | R/W | I2C: sets SCL clock low-period counter.  UART, SPI: no used. |

**UART\_CTRL**

* Name: UART control register
* Address offset: 0x018

**Description:** All of the features (Data length, STOP bit length and parity) can be configured by this register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 6 |  | 0x0 | RO | Reserved. |
| 5 | EPS | 0x0 | R/W | Even parity select. This bit is used to select between even and odd parity, when parity is enable (PEN set to 1).  0x0: an odd number of logic 1s is transmitted or checked.  0x1: an even number of logic 1s is transmitted or checked. |
| 4 | PEN | 0x0 | R/W | Parity enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.  0x0: parity disable.  0x1: parity enable. |
| 3: 2 | PBIT | 0x0 | R/W | Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits and receives.  0x0: 1 stop bit.  0x1: 1.5 stop pbits.  0x2: 2 stop bits.  0x3: number of stop bits keep the last set value unchanged. |
| 1: 0 | DBIT | 0x3 | R/W | Data bits select. This is used to select the number of data bits per character that the peripheral transmits and receive. The number of bit that may be selected areas follows:  0x0: 5-bit.  0x1: 6-bit.  0x2: 7-bit.  0x3: 8-bit. |

**UART\_STA**

* Name: UART status register.
* Address offset: 0x01C

**Description:** The RXD and TXD working state can be found form this state register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 2 |  | 0x0 | RO | Reserved. |
| 1 | RXD\_WORK | 0x0 | RO | The UART RXD is receiving. |
| 0 | TXD\_WORK | 0x0 | RO | The UART TXD is transmitting. |

**I2C\_MODE**

* Name: I2C mode control register
* Address offset: 0x020

**Description:** The I2C can work in only one mode of two (master or slave) mode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 1 |  | 0x0 | RO | Reserved. |
| 0 | MS\_MODE | 0x1 | R/W | 0x0: I2CS mode.  0x1: I2CM mode. |

**I2C\_ADDR**

* Name: I2C address register.
* Address offset: 0x024

**Description:** Both I2CM and I2CS use this register as I2CS address.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 10 |  | 0x0 | RO | Reserved. |
| 9: 0 | I2C\_ADDR | 0x133 | R/W | The 10-bit I2C slave address. |

**I2CM\_CTRL**

* Name: I2C master mode control register
* Address offset: 0x028

**Description:** The I2CM will implement I2C protocol in I2CM mode by configure this control register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 5 |  | 0 | RO | Reserved. |
| 4 | GCALL | 0x0 | R/W | 0x0: the first address is not general call address  0x1: the first address is general call address  If user want to use the general call address, he has to clear the control bit of ADDR\_MODE. |
| 3 | SBYTE | 0x0 | R/W | 0x0: do not generate START byte.  0x1: generate START byte. |
| 2 | HS\_MODE | 0x0 | R/W | High speed mode. |
| 1 | STOP | 0x0 | R/W | 0: if TX FIFO is empty, the I2CM holds SCL low.  1: if TX FIFO is empty, the I2CM terminates transfer, then generating STOP condition. |
| 0 | ADDR\_MODE | 0x0 | R/W | 0: 7-bit address mode.  1: 10-bit address mode.  If both of GCALL and ADDR\_MODE is set by “1”, I2CM only work in 10-bit I2C address mode. |

**I2CM\_CODE**

* Name: I2C master code register.
* Address offset: 0x02C

**Description:** The I2C master code is used in high-speed mode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 3 |  | 0x0 | RO | Reserved. |
| 2: 0 | MCODE | 0x1 | R/W | I2CM code. |

**I2CS\_CTRL**

* Name: I2C slave control register
* Address offset: 0x030

**Description:** Many special I2C protocol can be configured by this control register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 1 |  | 0x0 | RO | Reserved. |
| 0 | GCALL\_MODE | 0x0 | R/W | 0x0: receive general call data.  0x1: don’t receive the data of general call. |

**I2C\_FM\_DIV**

* Name: I2C fast-mode clock divider register.
* Address offset: 0x034

**Description:** The I2C fast mode divisor. this divisor is used in I2C master code and START byte transfer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 8 |  | 0x0 | RO | Reserved. |
| 7: 0 | I2C\_FM\_DIV | 0x5 | R/W | The I2C fast mode divisor. |

**I2C\_HOLD**

* Name: the hold time register.
* Address offset: 0x038

**Description:** This register controls the amount of hold time on the SDA signal after a negative of SCL in both master and slave mode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 8 |  | 0x0 | RO | Reserved. |
| 7: 0 | I2C\_HOLD | 0x5 | R/W | The I2C hold time register. The hold time after SCL fall edge, used in both I2CM and I2CS.  TI2C\_HOLD = (I2C\_HOLD+1)\*TPCLK  TPCLK: the PCLK period. |

**I2C\_STA**

* Name: I2C status register
* Address offset: 0x03C

**Description:** I2CM and I2CS state of working can be found in this state register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 9 |  | 0x0 | RO | Reserved. |
| 8 | I2CS\_WORK | 0x0 | RO | 1’b0: I2C slave is idle.  1’b1: I2C slave is active. |
| 7: 2 |  | 0x0 | RO | Reserved. |
| 1 | I2CM\_DATA | 0x0 | RO | If this bit equal 1’b1 that indicates I2C master is transmitting or receiving data. |
| 0 | I2CM\_WORK | 0x0 | RO | 1’b0: I2C master is idle.  1’b1: I2C master is active. |

**SPI\_MODE**

* Name: SPI (master or slave) mode select register.
* Address offset: 0x040

**Description:** The SPI master or slave can be selected by write this register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 1 |  | 0x0 | RO | Reserved. |
| 0 | MS\_MODE | 0x1 | R/W | 0x0: SPI slave mode.  0x1: SPI master mode. |

**SPI\_CTRL**

* Name: SPI control register
* Address offset: 0x044

**Description:** User can control phase, polarity and data length by this control register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 10 |  | 0x0 | RO | Reserved. |
| 9 | NSS\_CTRL | 0x0 | R/W | Infect the NSS output  0X0: the output NSS is controlled by the NSS\_TOGGLE  0x1: the output NSS is controlled by the NSS\_DATA |
| 8 | NSS\_TOGGLE | 0x0 | R/W | 0x0: the NSS toggle at end of each data item.  0x0: the NSS is not change until the SPI master stop. |
| 7 | CPOL | 0x0 | R/W | Clock polarity  0x0: Inactive state of serial clock is low.  0x1: Inactive state of serial clock is high. |
| 6 | CPHA | 0x0 | R/W | Clock phase  0x0: Serial clock toggles in middle of first data bit.  0x1: Serial clock toggles at start of first data bit. |
| 5: 4 | TMOD | 0x0 | R/W | 0x0: Transmit & Receive.  0x1: Transmit Only.  0x2: Receive Only.  0x3: TMOD keep the last set value unchanged. |
| 3: 0 | DATA\_SIZE | 0x7 | R/W | Data frame size. |
|  |  |  |  |  |

* DATA\_SIZE

|  |  |
| --- | --- |
| **DATA\_SIZE value** | **Description** |
| 0x0 | Reserved. If user set this value, it will be replaced by the default value 0x7. |
| 0x1 | Reserved. If user set this value, it will be replaced by the default value 0x7. |
| 0x2 | Reserved. If user set this value, it will be replaced by the default value 0x7. |
| 0x3 | 4-bit data length for each transfer. |
| 0x4 | 5-bit data length for each transfer. |
| 0x5 | 6-bit data length for each transfer. |
| 0x6 | 7-bit data length for each transfer. |
| 0x7 | 8-bit data length for each transfer. |
| 0x8 | 9-bit data length for each transfer. |
| 0x9 | 10-bit data length for each transfer. |
| 0xA | 11-bit data length for each transfer. |
| 0xB | 12-bit data length for each transfer. |
| 0xC | 13-bit data length for each transfer. |
| 0xD | 14-bit data length for each transfer. |
| 0xE | 15-bit data length for each transfer. |
| 0xF | 16-bit data length for each transfer. |

**SPI\_STA**

* Name: SPI status register
* Address offset: 0x048

**Description:** The SPI working status can be found in this register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 1 |  | 0x0 | RO | Reserved. |
| 0 | SPI\_WORKING | 0x0 | RO | SPI master or slave is working. |

**INTR\_CTRL**

* Name: interrupts control register.
* Address offset: 0x04C

**Description:** This register controls the threshold value for the transmit and receive FIFO memory.

| **Field** | **Name** | **Default** | **Access** | **Description** |
| --- | --- | --- | --- | --- |
| 31: 17 |  | 0x0 | RO | Reserved. |
| 16 | TH\_MODE | 0x0 | R/W | 0x1: not less than threshold.  0x0: not larger than threshold. |
| 15: 11 |  | 0x0 | RO | Reserved. |
| 10: 8 | RX\_FIFO\_TH | 0x1 | R/W | RX FIFO threshold level control bits. If the received data in RX FIFO equal the threshold value, an interrupt will be generated.  0x0: threshold value keep the last set value unchanged.  0x1: 1-byte data  0x2: 2-bye data  0x3: 3-byte data  0x4: 4-byte data  0x5: 5-byte data  0x6: 6-byte data  0x7: 7-byte data |
| 7: 3 |  | 0x0 | RO | Reserved. |
| 2: 0 | TX\_FIFO\_TH | 0x1 | R/W | TX FIFO threshold control bits.  0x0: threshold value keep the last set value unchanged.  0x1: 1-byte  0x2: 2-byte  0x3: 3-byte  0x4: 4-byte  0x5: 5-byte  0x6: 6-byte  0x7: 7-byte |

**INTR\_EN**

* Name: interrupts enable control register.
* Address offset: 0x050

**Description:** the interrupt can be enable by set the control bit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 19 |  | 0x0 | RO | Reserved. |
| 18 | SPI\_STOP\_EN | 0x0 | R/W | The interrupt enable control bit.  0x0: disable the interrupt.  0x1: enable the interrupt. |
| 17 | I2C\_AERR\_EN | 0x0 | R/W |
| 16 | I2CS\_GCALL\_EN | 0x0 | R/W |
| 15 | I2CM\_LOSE\_ARBI\_EN | 0x0 | R/W |
| 14 | I2C\_NACK\_EN | 0x0 | R/W |
| 13 | I2C\_STOP\_EN | 0x0 | R/W |
| 12 | UART\_PERR\_EN | 0x0 | R/W |
| 11 | UART\_RX\_STOP\_EN | 0x0 | R/W |
| 10 | UART\_TX\_STOP\_EN | 0x0 | R/W |
| 9 | RX\_WERR\_EN | 0x0 | R/W |
| 8 | RX\_RERR\_EN | 0x0 | R/W |
| 7 | RX\_FULL\_EN | 0x0 | R/W |
| 6 | RX\_EMPTY\_EN | 0x0 | R/W |
| 5 | RX\_THOLD\_EN | 0x0 | R/W |
| 4 | TX\_WERR\_EN | 0x0 | R/W |
| 3 | TX\_RERR\_EN | 0x0 | R/W |
| 2 | TX\_FULL\_EN | 0x0 | R/W |
| 1 | TX\_EMPTY\_EN | 0x0 | R/W |
| 0 | TX\_THOLD\_EN | 0x0 | R/W |

**INTR\_STA**

* Name: interrupts status register.
* Address offset: 0x054

**Description:** which interrupt will be found in this register, all the state of the interrupt can be masked.

| **Field** | **Name** | **Default** | **Access** | **Description** |
| --- | --- | --- | --- | --- |
| 31: 19 |  | 0x0 | RO | Reserved. |
| 18 | SPI\_STOP | 0x0 | RO | See “RAW\_INTR\_STA” on page 35 for a detailed description of these bits. |
| 17 | I2C\_AERR | 0x0 | RO |
| 16 | I2CS\_GCALL | 0x0 | RO |
| 15 | I2CM\_LOSE\_ARBI | 0x0 | RO |
| 14 | I2C\_NACK | 0x0 | RO |
| 13 | I2C\_STOP | 0x0 | RO |
| 12 | UART\_PERR | 0x0 | RO |
| 11 | UART\_RX\_STOP | 0x0 | RO |
| 10 | UART\_TX\_STOP | 0x0 | RO |
| 9 | RX\_WERR | 0x0 | RO |
| 8 | RX\_RERR | 0x0 | RO |
| 7 | RX\_FULL | 0x0 | RO |
| 6 | RX\_EMPTY | 0x0 | RO |
| 5 | RX\_THOLD | 0x0 | RO |
| 4 | TX\_WERR | 0x0 | RO |
| 3 | TX\_RERR | 0x0 | RO |
| 2 | TX\_FULL | 0x0 | RO |
| 1 | TX\_EMPTY | 0x0 | RO |
| 0 | TX\_THOLD | 0x0 | RO |

**RAW\_INTR\_STA**

* Name: interrupts status register.
* Address offset: 0x058

**Description:** The raw of interrupts status register.

| **Field** | **Name** | **Default** | **Access** | **Description** |
| --- | --- | --- | --- | --- |
| 31: 19 |  | 0x0 | RO | Reserved. |
| 18 | RAW\_SPI\_STOP | 0x0 | RO | SPI master: if the TX FIFO is empty, or the RX FIFO is full the interrupt will be generated before the FSM enter the SPIM\_IDLE state.  SPI slave: the interrupt will be generated after one data transfer. |
| 17 | RAW\_I2C\_AERR | 0x0 | RO | If the I2C\_ADDR is loaded a reserved address or a special address in 7-bit address mode, the I2CM will generate a interrupt. |
| 16 | RAW\_I2CS\_GCALL | 0x0 | RO | When the I2CS received a general call, the interrupt will be generated. |
| 15 | RAW\_I2CM\_LOSE\_ARBI | 0x0 | RO | When the I2CM loses arbitration, the interrupt will be generated. |
| 14 | RAW\_I2C\_NACK | 0x0 | RO | The interrupt will generated either the I2CM or I2CS receives NACK. |
| 13 | RAW\_I2C\_STOP | 0x0 | RO | The interrupt will be generated when the I2CM or I2CS finished working before the FSM enter the IDLE state. |
| 12 | RAW\_UART\_PERR | 0x0 | RO | The interrupt will be generated if the parity check fails when the UART RXD received the data. |
| 11 | RAW\_UART\_RX\_STOP | 0x0 | RO | UART RXD: the interrupt will be generated after the data is received. |
| 10 | RAW\_UART\_TX\_STOP | 0x0 | RO | UART TXD: the interrupt will be generated at the STOP bit. |
| 9 | RAW\_RX\_WERR | 0x0 | RO | The interrupt will be generated, when the RX FIFO is written after the RX FIFO is full. |
| 8 | RAW\_RX\_RERR | 0x0 | RO | The interrupt will be generated, when the RX FIFO is read after the RX FIFO is empty. |
| 7 | RAW\_RX\_FULL | 0x0 | RO | The interrupt will be generated when RX FIFO is full. |
| 6 | RAW\_RX\_EMPTY | 0x0 | RO | The interrupt will be generated when RX FIFO is empty. |
| 5 | RAW\_RX\_THOLD | 0x0 | RO | TH\_MODE = 0 (TH\_MODE is field of INTR\_CTRL register)  The interrupt will be generated, when the RX FIFO depth of data not larger than threshold value of RX\_FIFO\_TH (the field in INTR\_CTRL register).  TH\_MODE = 1  The interrupt will be generated, when the RX FIFO depth of data not less than threshold value of RX\_FIFO\_TH. |
| 4 | RAW\_TX\_WERR | 0x0 | RO | The interrupt will be generated, when the TX FIFO is written after the TX FIFO is full. |
| 3 | RAW\_TX\_RERR | 0x0 | RO | The interrupt will be generated, when the TX FIFO is read after the TX FIFO is empty. |
| 2 | RAW\_TX\_FULL | 0x0 | RO | The interrupt will be generated when TX FIFO is full. |
| 1 | RAW\_TX\_EMPTY | 0x0 | RO | The interrupt will be generated when TX FIFO is empty. |
| 0 | RAW\_TX\_THOLD | 0x0 | RO | TH\_MODE = 0  The interrupt will be generated, when the TX FIFO depth of data not larger than threshold value of TX\_FIFO\_TH (the field in INTR\_CTRL register).  TH\_MODE = 1  The interrupt will be generated, when the TX FIFO depth of data not less than threshold value of TX\_FIFO\_TH. |

**INTR\_UNMASK**

* Name: interrupts unmask register.
* Address offset: 0x05C

**Description:** the interrupt can be masked by this control register.

| **Field** | **Name** | **Default** | **Access** | **Description** |
| --- | --- | --- | --- | --- |
| 31: 19 |  | 0x0 | RO | Reserved. |
| 18 | SPI\_STOP\_MASK | 0x0 | R/W | Interrupt unmask register  0x0: masked  0x1: unmasked. |
| 17 | I2C\_AERR\_MASK | 0x0 | R/W |
| 16 | I2CS\_GCALL\_MASK | 0x0 | R/W |
| 15 | I2CM\_LOSE\_ARBI\_MASK | 0x0 | R/W |
| 14 | I2C\_NACK\_MASK | 0x0 | R/W |
| 13 | I2C\_STOP\_MASK | 0x0 | R/W |
| 12 | UART\_PERR\_MASK | 0x0 | R/W |
| 11 | UART\_RX\_STOP\_MASK | 0x0 | R/W |
| 10 | UART\_TX\_STOP\_MASK | 0x0 | R/W |
| 9 | RX\_WERR\_MASK | 0x0 | R/W |
| 8 | RX\_RERR\_MASK | 0x0 | R/W |
| 7 | RX\_FULL\_MASK | 0x0 | R/W |
| 6 | RX\_EMPTY\_MASK | 0x0 | R/W |
| 5 | RX\_THOLD\_MASK | 0x0 | R/W |
| 4 | TX\_WERR\_MASK | 0x0 | R/W |
| 3 | TX\_RERR\_MASK | 0x0 | R/W |
| 2 | TX\_FULL\_MASK | 0x0 | R/W |
| 1 | TX\_EMPTY\_MASK | 0x0 | R/W |
| 0 | TX\_THOLD\_MASK | 0x0 | R/W |

**INTR\_CLR**

* Name: interrupts clear register.
* Address offset: 0x060

**Description:** the control register be use to clear interrupt.

| **Field** | **Name** | **Default** | **Access** | **Description** |
| --- | --- | --- | --- | --- |
| 31: 19 |  | 0x0 | RO | Reserved. |
| 18 | SPI\_STOP\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_SPI\_STOP interrupt of the RAW\_INTR\_STA register. |
| 17 | I2C\_AERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_I2C\_AERR interrupt of the RAW\_INTR\_STA register. |
| 16 | I2CS\_GCALL\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_I2CS\_GCALL interrupt of the RAW\_INTR\_STA register. |
| 15 | I2CM\_LOSE\_ARBI\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_I2CM\_LOSE\_ARBI interrupt of the RAW\_INTR\_STA register. |
| 14 | I2C\_NACK\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_I2C\_NACK interrupt of the RAW\_INTR\_STA register. |
| 13 | I2C\_STOP\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_I2C\_STOP interrupt of the RAW\_INTR\_STA register. |
| 12 | UART\_PERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_UART\_PERR interrupt of the RAW\_INTR\_STA register. |
| 11 | UART\_RX\_STOP\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_UART\_RX\_STOP interrupt of the RAW\_INTR\_STA register. |
| 10 | UART\_TX\_STOP\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_UART\_TX\_STOP interrupt of the RAW\_INTR\_STA register. |
| 9 | RX\_WERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_RX\_WERR interrupt of the RAW\_INTR\_STA register. |
| 8 | RX\_RERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_RX\_RERR interrupt of the RAW\_INTR\_STA register. |
| 7 | RX\_FULL\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_RX\_FULL interrupt of the RAW\_INTR\_STA register. |
| 6 | RX\_EMPTY\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_RX\_EMPTY interrupt of the RAW\_INTR\_STA register. |
| 5 | RX\_THOLD\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_RX\_THOLD interrupt of the RAW\_INTR\_STA register. |
| 4 | TX\_WERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_TX\_WERR interrupt of the RAW\_INTR\_STA register. |
| 3 | TX\_RERR\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_TX\_RERR interrupt of the RAW\_INTR\_STA register. |
| 2 | TX\_FULL\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_TX\_FULL interrupt of the RAW\_INTR\_STA register. |
| 1 | TX\_EMPTY\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_TX\_EMPTY interrupt of the RAW\_INTR\_STA register. |
| 0 | TX\_THOLD\_CLR | 0x0 | WO | Write 1’b1 to clear the RAW\_TX\_THOLD interrupt of the RAW\_INTR\_STA register. |

**DMA\_CTRL**

* Name: DMA interface control register.
* Address offset: 0x064

**Description:** this register is design for controlling the TX FIFO or RX FIFO DMA interface.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 2 |  | 0x0 | RO | Reserved. |
| 1 | RX\_DMA\_EN | 0x0 | R/W | RX FIFO DMA interface enable.  0x0: disable the RX FIFO DMA interface.  0x1: enable the RX FIFO DMA interface. |
| 0 | TX\_DMA\_EN | 0x0 | R/W | TX FIFO DMA interface enable.  0x0: disable the TX FIFO DMA interface.  0x1: enable the TX FIFO DMA interface. |

**DMA\_THRESHOLD**

* Name: DMA interface threshold control register.
* Address offset: 0x068

**Description:** the threshold value for DMA trigger.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 12 |  | 0x0 | RO | Reserved. |
| 11: 8 | RX\_DMA\_TH | 0x4 | R/W | The counter is the threshold level for RX FIFO DMA interface. If the RX FIFO DMA interface is enable, and the number of received data words in RX FIFO not less than the threshold value, then the RX FIFO DMA interface requests that DMA read data from RX FIFO.  0x0: the threshold keep the last set value unchanged.  0x1: 1-data frame  …  0x5: 5-data frame  0x6: 6-data frame  0x7: 7-data frame  0x8: 8-data frame  0x9-0xF: Reserved. |
| 7: 4 |  | 0x0 | RO | Reserved. |
| 3: 0 | TX\_DMA\_TH | 0x4 | R/W | The counter is the threshold level for TX FIFO DMA interface. If the TX FIFO DMA interface is enable, and the number of stored data words in TX FIFO not larger than the threshold value, then the TX FIFO DMA interface requests that DMA write data to TX FIFO.  0x0: the threshold keep the last set value unchanged.  0x1: 1-data frame  …  0x5: 5-data frame  0x6: 6-data frame  0x7: 7-data frame  0x8: 8-data frame  0x9-0xF: Reserved. |

**SPI\_NSS\_DATA**

* Name: SPI NSS data register.
* Address offset: 0x06C

**Description:** User can input NSS data by this nss data register.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Name** | **Default** | **Access** | **Description** |
| 31: 1 |  | 0x0 | RO | Reserved. |
| 0 | NSS\_DATA | 0x0 | R/W | 0x0: if NSS\_CTRL is set 1, the “0” will be propagated to NSS  0x1: if NSS\_CTRL is set 1, the “1” will be propagated to NSS. |

# Watchdog （WDT）

## Register Description

### Register Memory map

**Table 5‑1 Memory Map of WDT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Address Offset** | **Width** | **Access** | **Reset value** | **Description** |
| WDT\_CR | 0x00 | 5 | R/W | 5’h2 | WDT control register |
| WDT\_time\_out | 0x04 | 8 | R/W | 8’h0 | WDT timeout range register |
| WDT\_current\_value | 0x08 | 32 | R | 32’hffff | WDT current counter value register |
| WDT\_restart | 0x0c | 8 | W | 8’h0 | WDT counter restart register |
| WDT\_int\_status | 0x10 | 1 | R | 1’b0 | WDT interrupt status register |
| WDT\_int\_clr | 0x14 | 1 | R | 1’b0 | WDT interrupt clear register |

### Register Field Description

**Control Register (WDT\_CR)**

Address Offset: 0x00

Read/write access: read/write

**Table 5‑2 WDT\_CR Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:5 | N/A | Reserved and read as zero (0). | |
| 4:2 | RPL | R/W | This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to  256 pclk cycles.  000 – 2 pclk cycles  001 – 4 pclk cycles  010 – 8 pclk cycles  011 – 16 pclk cycles  100 – 32 pclk cycles  101 – 64 pclk cycles  110 – 128 pclk cycles  111 – 256 pclk cycles  Reset Value: 3’b0. |
| 1 | RMOD | R/W | 0 = Generate a system reset.  1 = First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.  Reset Value: 1 |
| 0 | WDT\_EN | R/W | Once this bit has been enabled, it can only be cleared by a system reset.  0 = WDT disabled.  1 = WDT enabled.  Reset Value: 0 |

**Timeout Range Register (WDT\_time\_out)**

Address Offset: 0x04

Read/write access: read/write

**Table 5‑3 WDT\_time\_out Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:8 | Reserved and read as zero (0). | | |
| 7:4 | TOP\_INIT | R/W | Used to select the timeout period that the watchdog counter restarts from for the first counter restart(kick). This register should be written after reset and before the WDT is enabled.  A change of the TOP\_INIT is seen only once the WDT has been enabled, and any change after the first kick is not seen as subsequent kicks use the period specified by the TOP bits.  The range of values available for a 32-bit watchdog counter are:  Where i=TOP\_INIT and t=timeout period  for i = 0 to 15, t=32’hffff  Reset Value: 4’h0 |
| 3:0 | TOP | R/W | Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick).  The range of values available are:  where i = TOP and t = timeout period  for i = 0 to 15, t =32’hffff  Reset Value: 4’h0 |

**Current Counter Value Register (WDT\_current\_value)**

Address Offset: 0x08

Read/write access: read

**Table 5‑4 WDT\_current\_value Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:0 | Current Counter Value Register | R | This register, when read, is the current value of the internal counter. This value is read coherently whenever it is read.  Reset Value: 32’hffff |

**Counter Restart Register (WDT\_CRR)**

Address Offset: 0x0c

Read/write access: write

**Table 5‑5 WDT\_CRR Field Descriptio**n

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:8 | Reserved and read as zero. | | |
| 7:0 | Counter Restart Register | W | This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.  Reset Value: 0 |

**Interrupt Status Register (WDT\_int\_status)**

Address Offset: 0x10

Read/write access: read

**Table 5‑6 WDT\_STAT Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:1 | Reserved and read as zero. | | |
| 0 | Interrupt Status Register | R | This register shows the interrupt status of the WDT. 1 = Interrupt is active regardless of polarity. 0 = Interrupt is inactive.  Reset Value: 0 |

**Interrupt Clear Register (WDT\_int\_clr)**

Address Offset: 0x14

Read/write access: read

**Table 5‑7 WDT\_int\_clr Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:1 | Reserved and read as zero. | | |
| 0 | Interrupt Clear Register | R | Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter. Reset Value: 0 |

## Work Flow

This section describes the configuration of the WDR for a simple use. The special configuration should refer to the WDT register description.

Configure the WDT for use:

1. Disable the WDT by writing watchdog control register WDT\_CR.
2. Configure the WDT\_time\_out.
3. Enable the WDT by writing the WDT\_CR.

# Pulse Width Modulation （PWM）

## Register Description

### Register Memory Map

**Table 6‑3 all internal registers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Address Offset** | **Width** | **R/W** | **Description** |
| PWMCFG | 0x00 | 32 bits | R/W | PWM configure  **Reset Value:** 0x0 |
| PWMINVERTTRIG | 0x04 | 32 bits | R/W | PWM signal is inverted  **Reset Value:** 0x0 |
| PWM01TRIG  PWM23TRIG  PWM45TRIG | 0x08  0x0c  0x10 | 32 bits | R/W | Contain the trigger generate compare value  **Reset Value:** 0x0 |
| PWMINTEN1 | 0x14 | 32 bits | R/W | Interrupt enable for group2/1/0  **Reset Value:** 0x0 |
| PWMINTEN2 | 0x18 | 32 bits | R/W | Interrupt enable for group5/4/3  **Reset Value:** 0x0 |
| PWMRIS1 | 0x1c | 32 bits | RO | Raw interrupt status for group2/1/0  **Reset Value:** 0x0 |
| PWMRIS2 | 0x20 | 32 bits | RO | Raw interrupt status for group5/4/3  **Reset Value:** 0x0 |
| PWMIC1 | 0x24 | 32 bits | R/W | Interrupt clear for group2/1/0  **Reset Value:** 0x0 |
| PWMIC2 | 0x28 | 32 bits | R/W | Interrupt clear for group5/4/3  **Reset Value:** 0x0 |
| PWMIS1 | 0x2c | 32 bits | RO | Interrupt status for group2/1/0  **Reset Value:** 0x0 |
| PWMIS2 | 0x30 | 32 bits | RO | Interrupt status for group5/4/3  **Reset Value:** 0x0 |
| PWMCTL | 0x34 | 32 bits | R/W | Configure the PWM generation blocks  **Reset Value:** 0x0 |
| PWM01LOAD  PWM23LOAD  PWM45LOAD | 0x38  0x3c  0x40 | 32 bits | R/W | Contain the load value of the PWM counter  **Reset Value:** 0x0 |
| PWM01COUNT  PWM23COUNT  PWM45COUNT | 0x44  0x48  0x4c | 32 bits | RO | Contain the current value of the PWM counter  **Reset Value:** 0x0 |
| PWM0CMP  PWM1CMP  PWM2CMP  PWM3CMP  PWM4CMP  PWM5CMP | 0x50  0x54  0x58  0x5c  0x60  0x64 | 32 bits | R/W | Contain a value to be compared against the counter  **Reset Value:** 0x0 |
| PWM01DB  PWM23DB  PWM45DB | 0x68  0x6c  0x70 | 32 bits | R/W | Contain the number of clock ticks to delay  **Reset Value:** 0x0 |
| CAPCTL | 0x74 | 32 bits | R/W | Input capture control  **Reset Value:** 0x0 |
| CAPINTEN | 0x78 | 32 bits | R/W | Input capture interrupt enable  **Reset Value:** 0x0 |
| CAPRIS | 0x7c | 32 bits | RO | Input capture raw interrupt status  **Reset Value:** 0x0 |
| CAPIC | 0x80 | 32 bits | R/W | Input capture interrupt clear  **Reset Value:** 0x0 |
| CAPIS | 0x84 | 32 bits | RO | Input capture interrupt status  **Reset Value:** 0x0 |
| CAP01T  CAP23T  CAP45T | 0x88  0x8c  0x90 | 32 bits | RO | Input capture counter value  **Reset Value:** 0x0 |
| Cap01match  Cap23match  Cap45match | 0x94  0x98  0x9c | 32 bits | R/W | **Capture match value low 16bits**  **Reset Value:0x0** |
| Tim\_int\_en | 0xa0 | 32 bits | R/W | **Tim int enable**  **Reset Value:0x0** |
| timris | 0xa4 | 32 bits | RO | **Timer interrupt status**  **Reset Value:0x0** |
| Tim\_int\_clr | 0xa8 | 32 bits | R/W | **Timer interrupt clear**  **Reset Value:0x0** |
| timis | 0xac | 32 bits | RO | **Timer interrupt status**  **Reset Value:0x0** |
| Tim01load  Tim23load  Tim45load | 0xb0  0xb4  0xb8 | 32 bits | R/W | **Timer load value low 16bits**  **Reset Value:0x0** |
| Tim01count  Tim23count  Tim45count | 0xbc  0xc0  0xc4 | 32 bits | RO | **Timer count value low 16bits**  **Reset Value:0x0** |
| Cnt01val  Cnt23val  Cnv45val | 0xc8  0xcc  0xd0 | 32 bits | RO | **Current count value low 16bits**  **Reset Value:0x0** |

### Register Field Description

**PWMCFG**

This register configures the global operation of the PWM module.

* Name: PWM configuration
* Address Offset: 0x00

Table 6–1 PWMCFG Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:28 | Reserved and read as zero. | | |
| 27 | cntdiven | R/W | Counter frequency division configuration  1 : frequency division enable  0 : frequency division disable, so the counter clock source is system clock |
| 26:24 | cntdiv | R/W | Counter frequency division selection  111 : divide by 128  110 : divide by 128  101 : divide by 64  100 : divide by 32  011 : divide by 16  010 : divide by 8  001 : divide by 4  000 : divide by 2 |
| 23 | tim5en | R/W | Group 5 output compare configuration  1 : output compare enable  0 : output compare disable |
| 22 | tim4en | R/W | Group 4 output compare configuration  1 : output compare enable  0 : output compare disable |
| 21 | tim3en | R/W | Group 3 output compare configuration  1 : output compare enable  0 : output compare disable |
| 20 | tim2en | R/W | Group 2 output compare configuration  1 : output compare enable  0 : output compare disable |
| 19 | tim1en | R/W | Group 1 output compare configuration  1 : output compare enable  0 : output compare disable |
| 18 | tim0en | R/W | Group 0 output compare configuration  1 : output compare enable  0 : output compare disable |
| 17 | cap5en | R/W | Group 5-channel 10 input capture configuration  1 : input capture enable  0 : input capture disable |
| 16 | cap4en | R/W | Group 4-channel 8 input capture configuration  1 : input capture enable  0 : input capture disable |
| 15 | cap3en | R/W | Group 3-channel 6 input capture configuration  1 : input capture enable  0 : input capture disable |
| 14 | cap2en | R/W | Group 2-channel 4 input capture configuration  1 : input capture enable  0 : input capture disable |
| 13 | cap1en | R/W | Group 1-channel 2 input capture configuration  1 : input capture enable  0 : input capture disable |
| 12 | cap0en | R/W | Group 0-channel 0 input capture configuration  1 : input capture enable  0 : input capture disable |
| 11 | pwm11en | R/W | Channel 11 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 10 | pwm10en | R/W | Channel 10 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 9 | pwm9en | R/W | Channel 9 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 8 | pwm8en | R/W | Channel 8 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 7 | pwm7en | R/W | Channel 7 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 6 | pwm6en | R/W | Channel 6 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 5 | pwm5en | R/W | Channel 5 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 4 | pwm4en | R/W | Channel 4 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 3 | pwm3en | R/W | Channel 3 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 2 | pwm2en | R/W | Channel 2 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 1 | pwm1en | R/W | Channel 1 PWM configuration  1 : PWM output enable  0 : PWM output disable |
| 0 | pwm0en | R/W | Channel 0 PWM configuration  1 : PWM output enable  0 : PWM output disable |

**PWMINVERTTRIG**

This register controls the ADC trigger generation capabilities of the PWM generator (group) and provides a master control of the polarity of the PWM signals on the device pins.

* Name: PWM Output Inversion and trigger signal
* Address Offset: 0x04

Table 6–2 PWMINVERTTRIG Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:12 | Reserved and read as zero. | | |
| 11 | pwm11inv | R/W | Invert PWM11 signal (channel11)  1 : PWM5 signal is inverted  0 : PWM5 signal is not inverted |
| 10 | pwm10inv | R/W | Invert PWM10 signal (channel10)  1 : PWM4 signal is inverted  0 : PWM4 signal is not inverted |
| 9 | pwm9inv | R/W | Invert PWM9 signal (channel9)  1 : PWM3 signal is inverted  0 : PWM3 signal is not inverted |
| 8 | pwm8inv | R/W | Invert PWM8 signal (channel8)  1 : PWM2 signal is inverted  0 : PWM2 signal is not inverted |
| 7 | pwm7inv | R/W | Invert PWM7 signal (channel7)  1 : PWM1 signal is inverted  0 : PWM1 signal is not inverted |
| 6 | pwm6inv | R/W | Invert PWM6 signal (channel6)  1 : PWM0 signal is inverted  0 : PWM0 signal is not inverted |
| 5 | pwm5inv | R/W | Invert PWM5 signal (channel5)  1 : PWM5 signal is inverted  0 : PWM5 signal is not inverted |
| 4 | pwm4inv | R/W | Invert PWM4 signal (channel4)  1 : PWM4 signal is inverted  0 : PWM4 signal is not inverted |
| 3 | pwm3inv | R/W | Invert PWM3 signal (channel3)  1 : PWM3 signal is inverted  0 : PWM3 signal is not inverted |
| 2 | pwm2inv | R/W | Invert PWM2 signal (channel2)  1 : PWM2 signal is inverted  0 : PWM2 signal is not inverted |
| 1 | pwm1inv | R/W | Invert PWM1 signal (channel1)  1 : PWM1 signal is inverted  0 : PWM1 signal is not inverted |
| 0 | pwm0inv | R/W | Invert PWM0 signal (channel0)  1 : PWM0 signal is inverted  0 : PWM0 signal is not inverted |

**PWM01TRIG**

This register contains the value generating trigger signals.

* Name: PWM group 0 and group 1 trigger compare value
* Address Offset: 0x08

Table 6–3 PWM01TRIG Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | pwm1trig | R/W | The value generating group 1 trigger signal |
| 15:0 | pwm0trig | R/W | The value generating group 0 trigger signal |

**PWM23TRIG**

This register contains the value generating trigger signals.

* Name: PWM group 2 and group 3 trigger compare value
* Address Offset: 0x0c

Table 6–4 PWM23TRIG Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | pwm3trig | R/W | The value generating group 3 trigger signal |
| 15:0 | pwm2trig | R/W | The value generating group 2 trigger signal |

**PWM45TRIG**

This register contains the value generating trigger signals.

* Name: PWM group 4 and group 5 trigger compare value
* Address Offset: 0x10

Table 6–5 PWM45TRIG Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | pwm5trig | R/W | The value generating group 5 trigger signal |
| 15:0 | pwm4trig | R/W | The value generating group 4 trigger signal |

**PWMINTEN1**

This register controls the global interrupt generation capabilities of the PWM module.

* Name: PWM Interrupt enable for group 2/1/0
* Address Offset: 0x14

Table 6–6 PWMNTEN1 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:30 | Reserved and read as zero. | | |
| 29 | Int2encmpbd | R/W | Interrupt for counter=comparator B Down (channel5)  1 : a raw interrupt occurs when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 28 | Int2encmpad | R/W | Interrupt for counter=comparator A Down (channel4)  1 : a raw interrupt occurs when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 27 | Int2encmpbu | R/W | Interrupt for counter=comparator B Up (channel5)  1 : a raw interrupt occurs when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 26 | Int2encmpau | R/W | Interrupt for counter=comparator A Up (channel4)  1 : a raw interrupt occurs when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 25 | Int2encntload | R/W | Interrupt for counter=Load (channel5/4)  1 : a raw interrupt occurs when the counter matches the values in the PWM23LOAD register value  0 : no interrupt |
| 24 | Int2encntzero | R/W | Interrupt for counter=0 (channel5/4)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |
| 23:22 | Reserved and read as zero. | | |
| 21 | Int1encmpbd | R/W | Interrupt for counter=comparator B Down (channel3)  1 : a raw interrupt occurs when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 20 | Int1encmpad | R/W | Interrupt for counter=comparator A Down (channel2)  1 : a raw interrupt occurs when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 19 | Int1encmpbu | R/W | Interrupt for counter=comparator B Up (channel3)  1 : a raw interrupt occurs when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 18 | Int1encmpau | R/W | Interrupt for counter=comparator A Up (channel2)  1 : a raw interrupt occurs when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 17 | Int1encntload | R/W | Interrupt for counter=Load (channel3/2)  1 : a raw interrupt occurs when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 16 | Int1encntzero | R/W | Interrupt for counter=0 (channel3/2)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Int0encmpbd | R/W | Interrupt for counter=comparator B Down (channel1)  1 : a raw interrupt occurs when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 12 | Int0encmpad | R/W | Interrupt for counter=comparator A Down (channel0)  1 : a raw interrupt occurs when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 11 | Int0encmpbu | R/W | Interrupt for counter=comparator B Up (channel1)  1 : a raw interrupt occurs when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 10 | Int0encmpau | R/W | Interrupt for counter=comparator A Up (channel0)  1 : a raw interrupt occurs when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 9 | Int0encntload | R/W | Interrupt for counter=Load (channel1/0)  1 : a raw interrupt occurs when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 8 | Int0encntzero | R/W | Interrupt for counter=0 (channel1/0)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |
| 7:0 | Reserved and read as zero. | | |

**PWMINTEN2**

This register controls the global interrupt generation capabilities of the PWM module.

* Name: PWM Interrupt enable for group 5/4/3
* Address Offset: 0x18

Table 6–7 PWMINTEN2 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:22 | Reserved and read as zero. | | |
| 21 | Int5encmpbd | R/W | Interrupt for counter=comparator B Down (group5-channel11)  1 : a raw interrupt occurs when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 20 | Int5encmpad | R/W | Interrupt for counter=comparator A Down (group5-channel10)  1 : a raw interrupt occurs when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 19 | Int5encmpbu | R/W | Interrupt for counter=comparator B Up (group5-channel11)  1 : a raw interrupt occurs when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 18 | Int5encmpau | R/W | Interrupt for counter=comparator A Up (group5-channel10)  1 : a raw interrupt occurs when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 17 | Int5encntload | R/W | Interrupt for counter=Load (group5-channel11/10)  1 : a raw interrupt occurs when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 16 | Int5encntzero | R/W | Interrupt for counter=0 (group5-channel11/10)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Int4encmpbd | R/W | Interrupt for counter=comparator B Down (group4-channel9)  1 : a raw interrupt occurs when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 12 | Int4encmpad | R/W | Interrupt for counter=comparator A Down (group4-channel8)  1 : a raw interrupt occurs when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 11 | Int4encmpbu | R/W | Interrupt for counter=comparator B Up (group4-channel9)  1 : a raw interrupt occurs when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 10 | Int4encmpau | R/W | Interrupt for counter=comparator A Up (group4-channel8)  1 : a raw interrupt occurs when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 9 | Int4encntload | R/W | Interrupt for counter=Load (group4-channel8/9)  1 : a raw interrupt occurs when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 8 | Int4encntzero | R/W | Interrupt for counter=0 (group4-channel8/9)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |
| 7:6 | Reserved and read as zero. | | |
| 5 | Int3encmpbd | R/W | Interrupt for counter=comparator B Down (group3-channel7)  1 : a raw interrupt occurs when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 4 | Int3encmpad | R/W | Interrupt for counter=comparator A Down (group3-channel6)  1 : a raw interrupt occurs when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 3 | Int3encmpbu | R/W | Interrupt for counter=comparator B Up (group3-channel7)  1 : a raw interrupt occurs when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 2 | Int3encmpau | R/W | Interrupt for counter=comparator A Up (group3-channel6)  1 : a raw interrupt occurs when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 1 | Int3encntload | R/W | Interrupt for counter=Load (group3-channel7/6)  1 : a raw interrupt occurs when the counter matches the values in the PWM23LOAD register value  0 : no interrupt |
| 0 | Int3encntzero | R/W | Interrupt for counter=0 (group3-channel7/6)  1 : a raw interrupt occurs when the counter is zero  0 : no interrupt |

**PWMRIS1**

This register provides the current set of interrupt sources that asserted, regardless of whether they cause an interrupt to be asserted to the controller.

* Name: PWM raw interrupt status for gourp 2/1/0
* Address Offset: 0x1c

Table 6–8 PWMRIS1 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:30 | Reserved and read as zero. | | |
| 29 | Intris2cmpbd | RO | Raw Interrupt for counter=comparator B Down (group2-channel5)  1 : a raw interrupt asserted when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 28 | Intris2cmpad | RO | Raw Interrupt for counter=comparator A Down (group2-channel4)  1 : a raw interrupt asserted when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 27 | Intris2cmpbu | RO | Raw Interrupt for counter=comparator B Up (group2-channel5)  1 : a raw interrupt asserted when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 26 | Intris2cmpau | RO | Raw Interrupt for counter=comparator A Up (group2-channel4)  1 : a raw interrupt asserted when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 25 | Intris2cntload | RO | Raw Interrupt for counter=Load (group2-channel5/4)  1 : a raw interrupt asserted when the counter matches the values in the PWM23LOAD register value  0 : n5o interrupt |
| 24 | Intris2cntzero | RO | Raw Interrupt for counter=0 (group2-channel5/4)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |
| 23:22 | Reserved and read as zero. | | |
| 21 | Intris1cmpbd | RO | Raw Interrupt for counter=comparator B Down (group1-channel3)  1 : a raw interrupt asserted when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 20 | Intris1cmpad | RO | Raw Interrupt for counter=comparator A Down (group1-channel2)  1 : a raw interrupt asserted when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 19 | Intris1cmpbu | RO | Raw Interrupt for counter=comparator B Up (group1-channel3)  1 : a raw interrupt asserted when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 18 | Intris1cmpau | RO | Raw Interrupt for counter=comparator A Up (group1-channel2)  1 : a raw interrupt asserted when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 17 | Intris1cntload | RO | Raw Interrupt for counter=Load (group1-channel3/2)  1 : a raw interrupt asserted when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 16 | Intris1cntzero | RO | Raw Interrupt for counter=0 (group1-channel3/2)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intris0cmpbd | RO | Raw Interrupt for counter=comparator B Down (group0-channel1)  1 : a raw interrupt asserted when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 12 | Intris0cmpad | RO | Raw Interrupt for counter=comparator A Down (group0-channel0)  1 : a raw interrupt asserted when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 11 | Intris0cmpbu | RO | Raw Interrupt for counter=comparator B Up (group0-channel1)  1 : a raw interrupt asserted when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 10 | Intris0cmpau | RO | Raw Interrupt for counter=comparator A Up (group0-channel0)  1 : a raw interrupt asserted when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 9 | Intris0cntload | RO | Raw Interrupt for counter=Load (group0-channel1/0)  1 : a raw interrupt asserted when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 8 | Intris0cntzero | RO | Raw Interrupt for counter=0 (group0-channel1/0)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |
| 7:0 | Reserved and read as zero. | | |

**PWMRIS2**

This register controls the global interrupt generation capabilities of the PWM module.

* Name: PWM raw interrupt status for group 5/4/3
* Address Offset: 0x20

Table 6–9 PWMRIS2 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:22 | Reserved and read as zero. | | |
| 21 | Intris5cmpbd | RO | Raw Interrupt for counter=comparator B Down (group5-channel11)  1 : a raw interrupt asserted when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 20 | Intris5cmpad | RO | Raw Interrupt for counter=comparator A Down (group5-channel10)  1 : a raw interrupt asserted when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 19 | Intris5cmpbu | RO | Raw Interrupt for counter=comparator B Up (group5-channel11)  1 : a raw interrupt asserted when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 18 | Intris5cmpau | RO | Raw Interrupt for counter=comparator A Up (group5-channel10)  1 : a raw interrupt asserted when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 17 | Intris5cntload | RO | Raw Interrupt for counter=Load (group5-channel11/10)  1 : a raw interrupt asserted when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 16 | Intris5cntzero | RO | Raw Interrupt for counter=0 (group5-channel11/10)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intris4cmpbd | RO | Raw Interrupt for counter=comparator B Down (group4-channel9)  1 : a raw interrupt asserted when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 12 | Intris4cmpad | RO | Raw Interrupt for counter=comparator A Down (group4-channel8)  1 : a raw interrupt asserted when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 11 | Intris4cmpbu | RO | Raw Interrupt for counter=comparator B Up (group4-channel9)  1 : a raw interrupt asserted when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 10 | Intris4cmpau | RO | Raw Interrupt for counter=comparator A Up (group4-channel8)  1 : a raw interrupt asserted when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 9 | Intris4cntload | RO | Raw Interrupt for counter=Load (group4-channel9/8)  1 : a raw interrupt asserted when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 8 | Intris4cntzero | RO | Raw Interrupt for counter=0 (group4-channel9/8)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |
| 7:6 | Reserved and read as zero. | | |
| 5 | Intris3cmpbd | RO | Raw Interrupt for counter=comparator B Down (group3-channel7)  1 : a raw interrupt asserted when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 4 | Intris3cmpad | RO | Raw Interrupt for counter=comparator A Down (group3-channel6)  1 : a raw interrupt asserted when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 3 | Intris3cmpbu | RO | Raw Interrupt for counter=comparator B Up (group3-channel7)  1 : a raw interrupt asserted when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 2 | Intris3cmpau | RO | Raw Interrupt for counter=comparator A Up (group3-channel6)  1 : a raw interrupt asserted when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 1 | Intris3cntload | RO | Raw Interrupt for counter=Load (group3-channel7/6)  1 : a raw interrupt asserted when the counter matches the values in the PWM23LOAD register value  0 : no interrupt |
| 0 | Intris3cntzero | RO | Raw Interrupt for counter=0 (group3-channel7/6)  1 : a raw interrupt asserted when the counter is zero  0 : no interrupt |

**PWMIC1**

This register clears the interrupt.

* Name: PWM interrupt clear for gourp 2/1/0
* Address Offset: 0x24

Table 6–10 PWMIC1 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:30 | Reserved and read as zero. | | |
| 29 | Intic2cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group2-channel5)  1 : clear  0 : not clear |
| 28 | Intic2cmpad | R/W | Clear Interrupt for counter=comparator A Down (group2-channel4)  1 : clear  0 : not clear |
| 27 | Intic2cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group2-channel5)  1 : clear  0 : not clear |
| 26 | Intic2cmpau | R/W | Clear Interrupt for counter=comparator A Up (group2-channel4)  1 : clear  0 : not clear |
| 25 | Intic2cntload | R/W | Clear Interrupt for counter=Load (group2-channel5/4)  1 : clear  0 : not clear |
| 24 | Intic2cntzero | R/W | Clear Interrupt for counter=0 (group2-channel5/4)  1 : clear  0 : not clear |
| 23:22 | Reserved and read as zero. | | |
| 21 | Intic1cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group1-channel3)  1 : clear  0 : not clear |
| 20 | Intic1cmpad | R/W | Clear Interrupt for counter=comparator A Down (group1-channel2)  1 : clear  0 : not clear |
| 19 | Intic1cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group1-channel3)  1 : clear  0 : not clear |
| 18 | Intic1cmpau | R/W | Clear Interrupt for counter=comparator A Up (group1-channel2)  1 : clear  0 : not clear |
| 17 | Intic1cntload | R/W | Clear Interrupt for counter=Load (group1-channel3/2)  1 : clear  0 : not clear |
| 16 | Intic1cntzero | R/W | Clear Interrupt for counter=0 (group1-channel3/2)  1 : clear  0 : not clear |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intic0cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group0-channel1/0)  1 : clear  0 : not clear |
| 12 | Intic0cmpad | R/W | Clear Interrupt for counter=comparator A Down (group0-channel0)  1 : clear  0 : not clear |
| 11 | Intic0cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group0-channel1)  1 : clear  0 : not clear |
| 10 | Intic0cmpau | R/W | Clear Interrupt for counter=comparator A Up (group0-channel0)  1 : clear  0 : not clear |
| 9 | Intic0cntload | R/W | Clear Interrupt for counter=Load (group0-channel1/0)  1 : clear  0 : not clear |
| 8 | Intic0cntzero | R/W | Clear Interrupt for counter=0 (group0-channel1/0)  1 : clear  0 : not clear |
| 7:0 | Reserved and read as zero. | | |

**PWMIC2**

This register clears the interrupt.

* Name: PWM interrupt clear for group 5/4/3
* Address Offset: 0x28

Table 6–11 PWMIC2 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:22 | Reserved and read as zero. | | |
| 21 | Intic5cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group5-channel11)  1 : clear  0 : not clear |
| 20 | Intic5cmpad | R/W | Clear Interrupt for counter=comparator A Down (group5-channel10)  1 : clear  0 : not clear |
| 19 | Intic5cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group5-channel11)  1 : clear  0 : not clear |
| 18 | Intic5cmpau | R/W | Clear Interrupt for counter=comparator A Up (group5-channel10)  1 : clear  0 : not clear |
| 17 | Intic5cntload | R/W | Clear Interrupt for counter=Load (group5-channel11/10)  1 : clear  0 : not clear |
| 16 | Intic5cntzero | R/W | Clear Interrupt for counter=0 (group5-channel11/10)  1 : clear  0 : not clear |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intic4cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group4-channel9)  1 : clear  0 : not clear |
| 12 | Intic4cmpad | R/W | Clear Interrupt for counter=comparator A Down (group4-channel8)  1 : clear  0 : not clear |
| 11 | Intic4cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group4-channel9)  1 : clear  0 : not clear |
| 10 | Intic4cmpau | R/W | Clear Interrupt for counter=comparator A Up (group4-channel8)  1 : clear  0 : not clear |
| 9 | Intic4cntload | R/W | Clear Interrupt for counter=Load (group4-channel9/8)  1 : clear  0 : not clear |
| 8 | Intic4cntzero | R/W | Clear Interrupt for counter=0 (group4-channel9/8)  1 : clear  0 : not clear |
| 7:6 | Reserved and read as zero. | | |
| 5 | Intic3cmpbd | R/W | Clear Interrupt for counter=comparator B Down (group3-channel7)  1 : clear  0 : not clear |
| 4 | Intic3cmpad | R/W | Clear Interrupt for counter=comparator A Down (group3-channel6)  1 : clear  0 : not clear |
| 3 | Intic3cmpbu | R/W | Clear Interrupt for counter=comparator B Up (group3-channel7)  1 : clear  0 : not clear |
| 2 | Intic3cmpau | R/W | Clear Interrupt for counter=comparator A Up (group3-channel6)  1 : clear  0 : not clear |
| 1 | Intic3cntload | R/W | Clear Interrupt for counter=Load (group3-channel7/6)  1 : clear  0 : not clear |
| 0 | Intic3cntzero | R/W | Clear Interrupt for counter=0 (group3-channel7/6)  1 : clear  0 : not clear |

**PWMIS1**

This register provides the current set of interrupt sources that asserted, regardless of whether they cause an interrupt to be asserted to the controller.

* Name: PWM interrupt status for gourp 2/1/0
* Address Offset: 0x2c

Table 6–12 PWMIS1 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:30 | Reserved and read as zero. | | |
| 29 | Intis2cmpbd | RO | Interrupt for counter=comparator B Down (group2-channel5)  1 : a interrupt asserted when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 28 | Intis2cmpad | RO | Interrupt for counter=comparator A Down (group2-channel4)  1 : a interrupt asserted when the counter matches the value in the PWM2CMP register value when counting down  0 : no interrupt |
| 27 | Intis2cmpbu | RO | Interrupt for counter=comparator B Up (group2-channel5)  1 : a interrupt asserted when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 26 | Intis2cmpau | RO | Interrupt for counter=comparator A Up (group2-channel4)  1 : a interrupt asserted when the counter matches the value in the PWM2CMP register value when counting up  0 : no interrupt |
| 25 | Intis2cntload | RO | Interrupt for counter=Load (group2-channel5/4)  1 : a interrupt asserted when the counter matches the values in the PWM23LOAD register value  0 : no interrupt |
| 24 | Intis2cntzero | RO | Interrupt for counter=0 (group2-channel5/4)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |
| 23:22 | Reserved and read as zero. | | |
| 21 | Intis1cmpbd | RO | Interrupt for counter=comparator B Down (group1-channel3)  1 : a interrupt asserted when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 20 | Intis1cmpad | RO | Interrupt for counter=comparator A Down (group1-channel2)  1 : a interrupt asserted when the counter matches the value in the PWM1CMP register value when counting down  0 : no interrupt |
| 19 | Intis1cmpbu | RO | Interrupt for counter=comparator B Up (group1-channel3)  1 : a interrupt asserted when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 18 | Intis1cmpau | RO | Interrupt for counter=comparator A Up (group1-channel2)  1 : a interrupt asserted when the counter matches the value in the PWM1CMP register value when counting up  0 : no interrupt |
| 17 | Intis1cntload | RO | Interrupt for counter=Load (group1-channel3/2)  1 : a interrupt asserted when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 16 | Intis1cntzero | RO | Interrupt for counter=0 (group1-channel3/2)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intis0cmpbd | RO | Interrupt for counter=comparator B Down (group0-channel1)  1 : a interrupt asserted when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 12 | Intis0cmpad | RO | Interrupt for counter=comparator A Down (group0-channel0)  1 : a interrupt asserted when the counter matches the value in the PWM0CMP register value when counting down  0 : no interrupt |
| 11 | Intis0cmpbu | RO | Interrupt for counter=comparator B Up (group0-channel1)  1 : a interrupt asserted when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 10 | Intis0cmpau | RO | Interrupt for counter=comparator A Up (group0-channel0)  1 : a interrupt asserted when the counter matches the value in the PWM0CMP register value when counting up  0 : no interrupt |
| 9 | Intis0cntload | RO | Interrupt for counter=Load (group0-channel1/0)  1 : a interrupt asserted when the counter matches the values in the PWM01LOAD register value  0 : no interrupt |
| 8 | Intis0cntzero | RO | Interrupt for counter=0 (group0-channel1/0)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |
| 7:0 | Reserved and read as zero. | | |

**PWMIS2**

This register controls the global interrupt generation capabilities of the PWM module.

* Name: PWM raw interrupt status for group 5/4/3
* Address Offset: 0x30

Table 6–13 PWMIS2 Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:22 | Reserved and read as zero. | | |
| 21 | Intis5cmpbd | RO | Interrupt for counter=comparator B Down (group5-channel11)  1 : a interrupt asserted when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 20 | Intis5cmpad | RO | Interrupt for counter=comparator A Down (group5-channel10)  1 : a interrupt asserted when the counter matches the value in the PWM5CMP register value when counting down  0 : no interrupt |
| 19 | Intis5cmpbu | RO | Interrupt for counter=comparator B Up (group5-channel11)  1 : a interrupt asserted when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 18 | Intis5cmpau | RO | Interrupt for counter=comparator A Up (group5-channel10)  1 : a interrupt asserted when the counter matches the value in the PWM5CMP register value when counting up  0 : no interrupt |
| 17 | Intis5cntload | RO | Interrupt for counter=Load (group5-channel11/10)  1 : a interrupt asserted when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 16 | Intis5cntzero | RO | Interrupt for counter=0 (group5-channel11/10)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |
| 15:14 | Reserved and read as zero. | | |
| 13 | Intis4cmpbd | RO | Interrupt for counter=comparator B Down (group4-channel9)  1 : a interrupt asserted when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 12 | Intis4cmpad | RO | Interrupt for counter=comparator A Down (group4-channel8)  1 : a interrupt asserted when the counter matches the value in the PWM4CMP register value when counting down  0 : no interrupt |
| 11 | Intis4cmpbu | RO | Interrupt for counter=comparator B Up (group4-channel9)  1 : a interrupt asserted when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 10 | Intis4cmpau | RO | Interrupt for counter=comparator A Up (group4-channel8)  1 : a interrupt asserted when the counter matches the value in the PWM4CMP register value when counting up  0 : no interrupt |
| 9 | Intis4cntload | RO | Interrupt for counter=Load (group4-channel9/8)  1 : a interrupt asserted when the counter matches the values in the PWM45LOAD register value  0 : no interrupt |
| 8 | Intis4cntzero | RO | Interrupt for counter=0 (group4-channel9/8)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |
| 7:6 | Reserved and read as zero. | | |
| 5 | Intis3cmpbd | RO | Interrupt for counter=comparator B Down (group3-channel7)  1 : a interrupt asserted when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 4 | Intis3cmpad | RO | Interrupt for counter=comparator A Down (group3-channel6)  1 : a interrupt asserted when the counter matches the value in the PWM3CMP register value when counting down  0 : no interrupt |
| 3 | Intis3cmpbu | RO | Interrupt for counter=comparator B Up (group3-channel7)  1 : a interrupt asserted when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 2 | Intis3cmpau | RO | Interrupt for counter=comparator A Up (group3-channel6)  1 : a interrupt asserted when the counter matches the value in the PWM3CMP register value when counting up  0 : no interrupt |
| 1 | Intis3cntload | RO | Interrupt for counter=Load (group3-channel7/6)  1 : a interrupt asserted when the counter matches the values in the PWM23LOAD register value  0 : no interrupt |
| 0 | Intis3cntzero | RO | Interrupt for counter=0 (group3-channel7/6)  1 : a interrupt asserted when the counter is zero  0 : no interrupt |

**PWMCTL**

This register configures the PWM counter mode.

* Name: PWM control
* Address Offset: 0x34

Table 6–14 PWMCTL Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:18 | Reserved and read as zero. | | |
| 17:16 | Sync5mode | R/W | Synchronize mode (group5-channel11/10)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 15:14 | Sync4mode | R/W | Synchronize mode (group4-channel9/8)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 13:12 | Sync3mode | R/W | Synchronize mode (group3-channel7/6)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 11:10 | sync2mode | R/W | Synchronize mode (group2-channel5/4)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 9:8 | sync1mode | R/W | Synchronize mode (group1-channel3/2)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 7:6 | sync0mode | R/W | Synchronize mode (group0-channel1/0)  00 : update register value when counter reach to zero  01 : update register value when counter reach to load value  10 : update register value when counter reach to zero and load value  11 : not update |
| 5 | pwm5mode | R/W | Counter mode (group5-channel11/10)  1 : count-up/down mode  0 : count-up mode |
| 4 | pwm4mode | R/W | Counter mode (group4-channel9/8)  1 : count-up/down mode  0 : count-up mode |
| 3 | pwm3mode | R/W | Counter mode (group3-channel7/6)  1 : count-up/down mode  0 : count-up mode |
| 2 | pwm2mode | R/W | Counter mode (group2-channel5/4)  1 : count-up/down mode  0 : count-up mode |
| 1 | pwm1mode | R/W | Counter mode (group1-channel3/2)  1 : count-up/down mode  0 : count-up mode |
| 0 | pwm0mode | R/W | Counter mode (group0-channel1/0)  1 : count-up/down mode  0 : count-up mode |

**PWM01LOAD**

**PWM23LOAD**

**PWM45LOAD**

This register contains the load value for the PWM counter.

* Name: PWM load value
* Address Offset: 0x38, 0x3c, 0x40

Table 6–15 PWMnmLOAD Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | loadm | R/W | Counter load m value  (m=1 : group1-channel3/2 ;  m=3 : group3-channel7/6 ;  m=5 : group5-channel11/10)  The counter load value |
| 15:0 | loadn | R/W | Counter load n value  (n=0 : group0-channel1/0 ;  n=2 : group2-channel5/4 ;  n=4 : group4-channel9/8)  The counter load value |

**PWM01COUNT**

**PWM23COUNT**

**PWM45COUNT**

This register contains the current value of the PWM counter.

* Name: PWM current value
* Address Offset: 0x44, 0x48, 0x4c

Table 6–16 PWMnmCOUNT Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | countm | RO | Counter m value  (m=1 : group1-channel3/2 ;  m=3 : group3-channel7/6 ;  m=5 : group5-channel11/10)  The current value of the counter |
| 15:0 | countn | RO | Counter n value  (n=0 : group0-channel1/0 ;  n=2 : group2-channel5/4 ;  n=4 : group4-channel9/8)  The current value of the counter |

**PWM0CMP**

**PWM1CMP**

**PWM2CMP**

**PWM3CMP**

**PWM4CMP**

**PWM5CMP**

This register contain a value to be compared against the counter.

* Name: PWM compare value
* Address Offset: 0x50, 0x54, 0x58, 0x5c, 0x60, 0x64

Table 6–17 PWMnCMP Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | compnb | R/W | Comparator B value  (n=0 : group0-channel1;  n=1 : group1-channel3 ;  n=2 : group2-channel5 ;  n=3 : group2-channel7 ;  n=4 : group4-channel9 ;  n=5 : group5-channel11)  The value to be compared against the counter |
| 15:0 | compna | R/W | Comparator A value  (n=0 : group0-channel0 ;  n=1 : group1-channel2 ;  n=2 : group2-channel4 ;  n=3 : group2-channel6 ;  n=4 : group4-channel8 ;  n=5 : group5-channel10)  The value to be compared against the counter |

**PWM01DB**

**PWM23DB**

**PWM45DB**

These registers contains the number of clock ticks to delay.

* Name: PWM deadband value
* Address Offset: 0x68, 0x6c, 0x70

Table 6–18 PWMnmDB Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:26 | Reserved and read as zero | | |
| 25 | dbmen | R/W | Dead-band m generator (group) enable  (m=1 : group1-channel3/2 ;  m=3 : group3-channel7/6 ;  m=5 : group5-channel11/10)  1 : insert dead-band into the output signals  0 : pass the PWM signals through simply |
| 24 | dbnen | R/W | Dead-band n generator (group) enable  (n=0 : group0-channel1/0 ;  n=2 : group2-channel5/4 ;  n=4 : group4-channel9/8)  1 : insert dead-band into the output signals  0 : pass the PWM signals through simply |
| 23:12 | delaym | R/W | Dead-band delay  (m=1 : group1-channel3/2 ;  m=3 : group3-channel7/6 ;  m=5 : group5-channel11/10)  The number of clock tick to delay |
| 11:0 | delayn | R/W | Dead-band delay  (n=0 : group0-channel1/0 ;  n=2 : group2-channel5/4 ;  n=4 : group4-channel9/8)  The number of clock tick to delay |

**CAPCTL**

This register configures input capture mode.

* Name: Input capture mode control
* Address Offset: 0x74

Table 6–19 CAPCTL Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:18 | Reserved and read as zero | | |
| 17:16 | cap5event | R/W | Capture5 edge event mode (group5-channel10)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 15:14 | cap4event | R/W | Capture4 edge event mode (group4-channel8)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 13:12 | cap3event | R/W | Capture3 edge event mode (group3-channel6)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 11:10 | cap2event | R/W | Capture2 edge event mode (group2-channel4)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 9:8 | cap1event | R/W | Capture1 edge event mode (group1-channel2)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 7:6 | cap0event | R/W | Capture0 edge event mode (group0-channel0)  00: posedge edge  01 : negedge edge  10 : Reserved and read as zero  11 : both edge |
| 5 | cap5mode | R/W | Capture5 mode (group5-channel10)  1 : edge count mode  0 : edge time mode |
| 4 | cap4mode | R/W | Capture4 mode (group4-channel8)  1 : edge count mode  0 : edge time mode |
| 3 | cap3mode | R/W | Capture3 mode (group3-channel6)  1 : edge count mode  0 : edge time mode |
| 2 | cap2mode | R/W | Capture2 mode (group2-channel4)  1 : edge count mode  0 : edge time mode |
| 1 | cap1mode | R/W | Capture1 mode (group1-channel2)  1 : edge count mode  0 : edge time mode |
| 0 | cap0mode | R/W | Capture0 mode (group0-channel0)  1 : edge count mode  0 : edge time mode |

**CAPINTEN**

This register enables the interrupt for input capture function mode.

* Name: interrupt enable for input capture mode
* Address Offset: 0x78

Table 6–20 CAPINTEN Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:12 | Reserved and read as zero. | | |
| 11 | cap5timie | R/W | Capture5 edge time interrupt enable  (group5-channel10)  1 : interrupt enable  0 : interrupt disable |
| 10 | cap4timie | R/W | Capture4 edge time interrupt enable  (group4-channel8)  1 : interrupt enable  0 : interrupt disable |
| 9 | cap3timie | R/W | Capture3 edge time interrupt enable  (group3-channel6)  1 : interrupt enable  0 : interrupt disable |
| 8 | cap2timie | R/W | Capture2 edge time interrupt enable  (group2-channel4)  1 : interrupt enable  0 : interrupt disable |
| 7 | cap1timie | R/W | Capture1 edge time interrupt enable  (group1-channel2)  1 : interrupt enable  0 : interrupt disable |
| 6 | cap0timie | R/W | Capture0 edge time interrupt enable  (group0-channel0)  1 : interrupt enable  0 : interrupt disable |
| 5 | cap5cntie | R/W | Capture5 edge count interrupt enable  (group5-channel10)  1 : interrupt enable  0 : interrupt disable |
| 4 | cap4cntie | R/W | Capture4 edge count interrupt enable  (group4-channel8)  1 : interrupt enable  0 : interrupt disable |
| 3 | cap3cntie | R/W | Capture3 edge count interrupt enable  (group3-channel6)  1 : interrupt enable  0 : interrupt disable |
| 2 | cap2cntie | R/W | Capture2 edge count interrupt enable  (group2-channel4)  1 : interrupt enable  0 : interrupt disable |
| 1 | cap1cntie | R/W | Capture1 edge count interrupt enable  (group1-channel2)  1 : interrupt enable  0 : interrupt disable |
| 0 | cap0cntie | R/W | Capture0 edge count interrupt enable  (group0-channel0)  1 : interrupt enable  0 : interrupt disable |

**CAPRIS**

This register shows the state of raw interrupt for input capture function mode.

* Name: raw interrupt status for input capture mode
* Address Offset: 0x7c

Table 6–21 CAPRIS Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:12 | Reserved and read as zero. | | |
| 11 | cap5timris | RO | Capture5 edge time raw interrupt occurs  (group5-channel10)  1 : interrupt  0 : no interrupt |
| 10 | cap4timris | RO | Capture4 edge time raw interrupt occurs  (group4-channel8)  1 : interrupt  0 : no interrupt |
| 9 | cap3timris | RO | Capture3 edge time raw interrupt occurs  (group3-channel6)  1 : interrupt  0 : no interrupt |
| 8 | cap2timris | RO | Capture2 edge time raw interrupt occurs  (group2-channel4)  1 : interrupt  0 : no interrupt |
| 7 | cap1timris | RO | Capture1 edge time raw interrupt occurs  (group1-channel2)  1 : interrupt  0 : no interrupt |
| 6 | cap0timris | RO | Capture0 edge time raw interrupt occurs  (group0-channel0)  1 : interrupt  0 : no interrupt |
| 5 | cap5cntris | RO | Capture5 edge count raw interrupt occurs  (group5-channel10)  1 : interrupt  0 : no interrupt |
| 4 | cap4cntris | RO | Capture4 edge count raw interrupt occurs  (group4-channel8)  1 : interrupt  0 : no interrupt |
| 3 | cap3cntris | RO | Capture3 edge count raw interrupt occurs  (group3-channel6)  1 : interrupt  0 : no interrupt |
| 2 | cap2cntris | RO | Capture2 edge count raw interrupt occurs  (group2-channel4)  1 : interrupt  0 : no interrupt |
| 1 | cap1cntris | RO | Capture1 edge count raw interrupt occurs  (group1-channel2)  1 : interrupt  0 : no interrupt |
| 0 | cap0cntris | RO | Capture0 edge count raw interrupt occurs  (group0-channel0)  1 : interrupt  0 : no interrupt |

**CAPIC**

This register clears internal interrupt.

* Name: interrupt clear for input capture mode
* Address Offset: 0x80

Table 6–22 CAPIC Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:12 | Reserved and read as zero. | | |
| 11 | cap5timic | R/W | Clear Capture5 edge time interrupt  (group5-channel10)  1 : clear interrupt  0 : not clear interrupt |
| 10 | cap4timic | R/W | Clear Capture4 edge time interrupt  (group4-channel8)  1 : clear interrupt  0 : not clear interrupt |
| 9 | cap3timic | R/W | Clear Capture3 edge time interrupt  (group3-channel6)  1 : clear interrupt  0 : not clear interrupt |
| 8 | cap2timic | R/W | Clear Capture2 edge time interrupt  (group2-channel4)  1 : clear interrupt  0 : not clear interrupt |
| 7 | cap1timic | R/W | Clear Capture1 edge time interrupt  (group1-channel2)  1 : clear interrupt  0 : not clear interrupt |
| 6 | cap0timic | R/W | Clear Capture0 edge time interrupt  (group0-channel0)  1 : clear interrupt  0 : not clear interrupt |
| 5 | cap5cntic | R/W | Clear Capture5 edge count interrupt  (group5-channel10)  1 : clear interrupt  0 : not clear interrupt |
| 4 | cap4cntic | R/W | Clear Capture4 edge count interrupt  (group4-channel8)  1 : clear interrupt  0 : not clear interrupt |
| 3 | cap3cntic | R/W | Clear Capture3 edge count interrupt  (group3-channel6)  1 : clear interrupt  0 : not clear interrupt |
| 2 | cap2cntic | R/W | Clear Capture2 edge count interrupt  (group2-channel4)  1 : clear interrupt  0 : not clear interrupt |
| 1 | cap1cntic | R/W | Clear Capture1 edge count interrupt  (group1-channel2)  1 : clear interrupt  0 : not clear interrupt |
| 0 | cap0cntic | R/W | Clear Capture0 edge count interrupt  (group0-channel0)  1 : clear interrupt  0 : not clear interrupt |

**CAPIS**

This register shows the state of interrupt for input capture function mode.

* Name: interrupt status for input capture mode
* Address Offset: 0x84

Table 6–23 CAPIS Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:12 | Reserved and read as zero. | | |
| 11 | cap5timis | RO | Capture5 edge time interrupt occurs  (group5-channel10)  1 : interrupt  0 : no interrupt |
| 10 | cap4timis | RO | Capture4 edge time interrupt occurs  (group4-channel8)  1 : interrupt  0 : no interrupt |
| 9 | cap3timis | RO | Capture3 edge time interrupt occurs  (group3-channel6)  1 : interrupt  0 : no interrupt |
| 8 | cap2timis | RO | Capture2 edge time interrupt occurs  (group2-channel4)  1 : interrupt  0 : no interrupt |
| 7 | cap1timis | RO | Capture1 edge time interrupt occurs  (group1-channel2)  1 : interrupt  0 : no interrupt |
| 6 | cap0timis | RO | Capture0 edge time interrupt occurs  (group0-channel0)  1 : interrupt  0 : no interrupt |
| 5 | cap5cntis | RO | Capture5 edge count interrupt occurs  (group5-channel10)  1 : interrupt  0 : no interrupt |
| 4 | cap4cntis | RO | Capture4 edge count interrupt occurs  (group4-channel8)  1 : interrupt  0 : no interrupt |
| 3 | cap3cntis | RO | Capture3 edge count interrupt occurs  (group3-channel6)  1 : interrupt  0 : no interrupt |
| 2 | cap2cntis | RO | Capture2 edge count interrupt occurs  (group2-channel4)  1 : interrupt  0 : no interrupt |
| 1 | cap1cntis | RO | Capture1 edge count interrupt occurs  (group1-channel2)  1 : interrupt  0 : no interrupt |
| 0 | cap0cntis | RO | Capture0 edge count interrupt occurs  (group0-channel0)  1 : interrupt  0 : no interrupt |

**CAP01T**

**CAP23T**

**CAP45T**

This register contains the counter value when edge match for input capture function mode.

* Name: captured current value for input capture function mode
* Address Offset: 0x88, 0x8c, 0x90

Table 6–24 CAPnmT Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | CAPnmTm | RO | Input capture m counter value  (m=1 : group1-channel2 ;  m=3 : group3-channel6 ;  m=5 : group5-channel10) |
| 15:0 | CAPnmTn | RO | Input capture n counter value  (n=0 : group1-channel0 ;  n=2 : group3-channel4 ;  n=4 : group5-channel8) |

**CAP01MATCH**

**CAP23MATCH**

**CAP45MATCH**

This register contains the match value for input capture function mode.

* Name: capture match value for input capture function mode
* Address Offset: 0x94, 0x98, 0x9c

Table 6–25 CAPnmMATCH Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | CAPnmMATCHm | R/W | Input capture m match value  (m=1 : group1-channel2 ;  m=3 : group3-channel6 ;  m=5 : group5-channel10) |
| 15:0 | CAPnmMATCHn | R/W | Input capture n match value  (n=0 : group1-channel0 ;  n=2 : group3-channel4 ;  n=4 : group5-channel8) |

**TIMINTEN**

This register configures the interrupt enable for timer function mode.

* Name: interrupt enable for timer mode
* Address Offset: 0xa0

Table 6–26 TIMINTEN Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:6 | Reserved and read as zero. | | |
| 5 | tim5ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group5)  1 : interrupt enable  0 : interrupt disable |
| 4 | tim4ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group4)  1 : interrupt enable  0 : interrupt disable |
| 3 | tim3ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group3)  1 : interrupt enable  0 : interrupt disable |
| 2 | tim2ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group2)  1 : interrupt enable  0 : interrupt disable |
| 1 | tim1ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group1)  1 : interrupt enable  0 : interrupt disable |
| 0 | tim0ie | R/W | A interrupt enable when counter mathc the CMPnmMATCH value. (group0)  1 : interrupt enable  0 : interrupt disable |

**TIMRIS**

This register shows the state of raw internal interrupt for timer function mode.

* Name: raw interrupt status for timer mode
* Address Offset: 0xa4

Table 6–27 TIMRIS Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:6 | Reserved and read as zero. | | |
| 5 | tim5ris | RO | Timer5 raw interrupt occurs. (group5)  1 : interrupt  0 : no interrupt |
| 4 | tim4ris | RO | Timer4 raw interrupt occurs. (group4)  1 : interrupt  0 : no interrupt |
| 3 | tim3ris | RO | Timer3 raw interrupt occurs. (group3)  1 : interrupt  0 : no interrupt |
| 2 | tim2ris | RO | Timer2 raw interrupt occurs. (group2)  1 : interrupt  0 : no interrupt |
| 1 | tim1ris | RO | Timer1 raw interrupt occurs. (group1)  1 : interrupt  0 : no interrupt |
| 0 | tim0ris | RO | Timer0 raw interrupt occurs. (group0)  1 : interrupt  0 : no interrupt |

**TIMIC**

This register shows the interrupt clear for timer function mode.

* Name: interrupt clear for timer mode
* Address Offset: 0xa8

Table 6–28 TIMIC Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:6 | Reserved and read as zero. | | |
| 5 | tim5ic | R/W | Timer5 interrupt. (group5)  1 : clear interrupt  0 : not clear interrupt |
| 4 | tim4ic | R/W | Timer4 interrupt. (group4)  1 : clear interrupt  0 : not clear interrupt |
| 3 | tim3ic | R/W | Timer3 interrupt. (group3)  1 : clear interrupt  0 : not clear interrupt |
| 2 | tim2ic | R/W | Timer2 interrupt. (group2)  1 : clear interrupt  0 : not clear interrupt |
| 1 | tim1ic | R/W | Timer1 interrupt. (group1)  1 : clear interrupt  0 : not clear interrupt |
| 0 | tim0ic | R/W | Timer0 interrupt. (group0)  1 : clear interrupt  0 : not clear interrupt |

**TIMIS**

This register shows the status of interrupt for timer function mode.

* Name: interrupt status for timer mode
* Address Offset: 0xac

Table 6–29 TIMIS Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:6 | Reserved and read as zero. | | |
| 5 | tim5is | RO | Timer5 interrupt occurs. (group5)  1 : interrupt  0 : no interrupt |
| 4 | tim4is | RO | Timer4 interrupt occurs. (group4)  1 : interrupt  0 : no interrupt |
| 3 | tim3is | RO | Timer3 interrupt occurs. (group3)  1 : interrupt  0 : no interrupt |
| 2 | tim2is | RO | Timer2 interrupt occurs. (group2)  1 : interrupt  0 : no interrupt |
| 1 | tim1is | RO | Timer1 interrupt occurs. (group1)  1 : interrupt  0 : no interrupt |
| 0 | tim0is | RO | Timer0 interrupt occurs. (group0)  1 : interrupt  0 : no interrupt |

**TIM01LOAD**

**TIM23LOAD**

**TIM45LOAD**

This register contains the load value for timer function mode.

* Name: load value for timer function mode
* Address Offset: 0xb0, 0xb4, 0xb8

Table 6–30 TIMnmLOAD Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | timloadm | R/W | Timer m load value  (m=1 : group1 ;  m=3 : group3;  m=5 : group5) |
| 15:0 | timloadn | R/W | Timer n load value  (n=0 : group1 ;  n=2 : group3 ;  n=4 : group5) |

**TIM01COUNT**

**TIM23COUNT**

**TIM45COUNT**

This register contains the current value for timer function mode.

* Name: current value for timer function mode
* Address Offset: 0xbc, 0xc0, 0xc4

Table 6–31 TIMnmCOUNT Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | timcntm | RO | Timer m current count value  (m=1 : group1 ;  m=3 : group3;  m=5 : group5) |
| 15:0 | timcntn | RO | Timer n current count value  (n=0 : group1 ;  n=2 : group3 ;  n=4 : group5) |

**CNT01VAL**

**CNT23VAL**

**CNT45VAL**

This register contains the captured input pulse number value for input capture function mode.

* Name: captured value for input capture function mode
* Address Offset: 0xc8, 0xcc, 0xd0

Table 6–32 CNTnmVAL Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Name** | **R/W** | **Description** |
| 31:16 | Cntmval | RO | Counter m captured input pulse number value  (m=1 : group1 ;  m=3 : group3;  m=5 : group5) |
| 15:0 | Cntnval | RO | Counter n captured input pulse number value  (n=0 : group1 ;  n=2 : group3 ;  n=4 : group5) |

## Work Flow

The following example shows how to initialize the PWM generator 0 with a 25-KHz frequency, and with a 75% duty cycles on the PWM0 pin and a 25% duty on the PWM1 pin. This example assumes the bus clock is 40-MHz.

1. Select the PWM count mode, for example, write 0x0 to PWMCTL (count up mode)
2. Set the PWM period. For a 25-KHz frequency, the period = 40us. The APB clock is 40-MHz and divided by 2, so that the PWM clock source is 20-MHz. This translates to 800 clock ticks per period. Use this value to set PWM01LOAD register. Write 0x320 to PWM01LOAD[31:16] and PWM01LOAD[15:0].
3. Set the pulse width of the PWM0 pin for a 75% duty cycle. Write 0xc8 to PWM0CMP[15:0].
4. Set the pulse width of the PWM1 pin for a 25% duty cycle. Write 0x258 to PWM0CMP[31:16].
5. Bypass Dead-Band insert, write 0x0 to PWM01DB.
6. Set the PWM clock frequency division and start the counter and enable output PWM signal, write 0x8000003 to PWMCFG.

# Real-Time Clock （RTC）



## Register Description

### Register Memory Map

The following table descripts the register memory map.

**Table 7‑1 RTC Memory Map**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Address  offset | Width | Access | Reset Value | Description |
| RTC\_current\_value | 0x00 | 32 | R | 0x0 | Current Counter Value Register |
| RTC\_match\_value | 0x04 | 32 | RW | 0x0 | Counter Match Register |
| RTC\_load\_value | 0x08 | 32 | RW | 0x0 | Counter Load Register |
| RTC\_CCR | 0x0C | 4~2bits | RW | 0x0 | Counter Control Register |
| RTC\_int\_status | 0x10 | 32 | R | 0x0 | Interrupt Status Register |
| RTC\_raw\_int\_status | 0x14 | 32 | R | 0x0 | Interrupt Raw Status Register |
| RTC\_int\_clr | 0x18 | 32 | R | 0x0 | End of Interrupt Register |
| RTC\_COMP\_VERSION | 0x1C | 32 | R | 0x0 | Component Version Register |
| RTC\_DIV | 0x20 | 20 | RW | 0x4000 | Rtc clock divider value |

### Registers Field Description

**Current Counter Value Register (RTC\_current\_value)**

Address Offset: 0x00

Read/write access: read-only

Table 7‑2 RTC\_current\_value Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Current Counter Value | R | When read, this register is the current value of the internal counter. This value always is read coherently.  Reset Value: 0x0 |

**Counter Match Register (RTC\_match\_value)**

Address Offset: 0x04

Read/write access: read/write

Table 7‑3 RTC\_match\_value Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Counter Match | R/W | Interrupt Match Register. When the internal counter matches this register, an interrupt is generated, provided interrupt generation is enabled. When appropriate, this value is written coherently. Only when all the bytes are written is the register used by the interrupt detection logic.  **Reset Value:** 0x0 |

**Counter Load Register (RTC\_load\_value)**

Address Offset: 0x08

Read/write access: read/write

Table 7‑4 RTC\_CLR Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:0 | Counter Load | R/W | Loaded into the counter as the loaded value, which is written coherently.  **Reset Value:** 0x0 |

**Counter Control Register (RTC\_CCR)**

Address Offset: 0x0C

Read/write access: read/write

Table 7‑5 RTC\_CCR Fiesld Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:4 | N/A | N/A | Reserved and read as 0 |
| 3 | rtc\_wen | R/W | Allows the user to force the counter to wrap when a match occurs instead of waiting until the maximum count is reached.  0 = Wrap disabled  1 = Wrap enabled  **Reset Value:** 0x0 |
| 2 | Rtc\_en | R/W | Allows the user to control counting in the counter.  0 = Counter disabled  1 = Counter enabled  **Reset Value:** 0x0 |
| 1 | rtc\_mask | R/W | Allows the user to mask interrupt generation.  0 = Interrupt unmasked  1 = Interrupt masked  **Reset Value:** 0x0 |
| 0 | rtc\_ien | R/W | Allows the user to disable interrupt generation.  0 = Interrupt disabled  1 = Interrupt enabled  **Reset Value:** 0x0 |

**Interrupt Status Register (RTC\_int\_status)**

Address Offset: 0x10

Read/write access: read-only

Table 7‑6 RTC\_int\_status Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | N/A | N/A | Reserved and read as 0 |
| 0 | rtc\_stat | R | This register is the masked raw status  0 = Interrupt is inactive  1 = Interrupt is active (regardless of polarity)  **Reset Value:** 0x0 |

**Interrupt Raw Status Register (RTC\_Rawint\_status)**

Address Offset: 0x14

Read/write access: read-only

Table 7‑7 RTC\_Rawint\_status Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | N/A | N/A | Reserved and read as 0 |
| 0 | rtc\_rstat | R | 0 = Interrupt is inactive  1 = Interrupt is active (regardless of polarity)  **Reset Value:** 0x0 |

**End of Interrupt Register (RTC\_int\_clr)**

Address Offset: 0x18

Read/write access: read-only

Table 7‑8 RTC\_int\_clr Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | N/A | N/A | Reserved and read as 0 |
| 0 | rtc\_int\_clr | R | By reading this location, the match interrupt is cleared. Performing read-to-clear on interrupts, the interrupt is cleared at the end of the read.  **Reset Value:** 0x0 |

**CLK DIV Register (RTC\_DIV)**

Address Offset: 0x20

Read/write access: read/write

Table 7‑9 RTC\_DIV Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Name | R/W | Description |
| 31:1 | N/A | N/A | Reserved and read as 0 |
| 19:0 | rtc\_div | R/W | Rtc clcok divider base on this register value  **Reset Value:** 0x4000 |

## Work Flow

This section describe the configuration of the RTC for a simple use. The special configuration should refer to the RTC register description.

Configuration the RTC for use:

1. Set the control register (RTC\_CCR) to reset value to disable the RTC first.
2. Set the RTC counter match register (RTC\_match\_value).
3. Set the RTC counter load register (RTC\_load\_value).
4. Set the RTC counter control register to the corresponding value (RTC\_CCR).

# General-purpose I/O （GPIO）



## Register description

### Register Memory Map

**Table 8‑1** **Address Map of GPIO Registers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Description | Offset | Initial Value | R/W | Access Size/bit |
| gpio\_output\_data | GPIO output data register | 0x00 | 0x0 | R/W | 32 |
| gpio\_direction | GPIO direction register | 0x04 | 0x0 | R/W | 32 |
| gpio\_ctl | GPIO control register | 0x08 | 0x0 | R/W | 32 |
| gpio\_inten | Interrupt enable register | 0x30 | 0x0 | R/W | 32 |
| gpio\_intmask | Interrupt mask register | 0x34 | 0x0 | R/W | 32 |
| gpio\_inttype\_level | Interrupt level register | 0x38 | 0x0 | R/W | 32 |
| gpio\_int\_polarity | Interrupt polarity register | 0x3c | 0x0 | R/W | 32 |
| gpio\_intstatus | Interrupt status of GPIO | 0x40 | 0x0 | R | 32 |
| gpio\_rawintstatus | Raw interrupt status of GPIO (premasking) | 0x44 | 0x0 | R | 32 |
| gpio\_porta\_int\_clr | GPIO clear interrupt register | 0x4c | 0x0 | W | 32 |
| gpio\_input\_data | GPIO input data register | 0x50 | 0x0 | R | 32 |

### Registers Filed Description

* **Gpio\_output\_data:**

Name: GPIO Data Register

Size: 32 bits

Address Offset: 0x00

Read/write access: read/write

**Table 8‑2** **gpio\_output\_data Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | GPIO Data Register | Values written to this register are output on the I/O signals.  For GPIO if the corresponding data direction bits for GPIO are set to Output mode and the corresponding control bit for GPIO is set to Software mode. The value read back is equal to the last value written to this register. | 32’b0 |

* **gpio\_direction:**

Name: GPIO Data Direction Register

Size: 32 bits

Address Offset: 0x04

Read/write access: read/write

**Table 8‑3** **gpio\_direction Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | GPIO Data Direction Register | Values written to this register independently control the direction of the corresponding data bit in GPIO.  0 – Input 1 – Output for each bit | 32’b0 |

* **gpio\_ctl:**

Name: GPIO data source register

Size: 32 bits

Address Offset: 0x08

Read/write access: read/write

**Table 8‑4** **gpio\_ctl Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | GPIO Data Source Register | The data and control source for a signal can come from either software or hardware; this bit selects between them.  0-Software mode for each bit  1-Hardware mode for each bit  Reset Value: 28’h0 | 32’b0 |

* **gpio\_inten:**

Name: Interrupt enable

Size: 32 bits

Address Offset: 0x30

Read/write access: read/write

**Table 8‑5** **gpio\_inten Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Interrupt enable | Allows each bit of GPIO to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on GPIO to become an interrupt; otherwise, GPIO operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of GPIO if the corresponding data direction register is set to Output or if GPIO mode is set to Hardware.  0 – Configure GPIO bit as normal GPIO signal  1 – Configure GPIO bit as interrupt | 32’b0 |

* **gpio\_intmask:**

Name: Interrupt mask

Size: 32 bits

Address Offset: 0x34

Read/write access: read/write

**Table 8‑6** **gpio\_intmask Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Interrupt mask | Controls whether an interrupt on GPIO can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking.  0 – Interrupt bits are unmasked (default) 1 – Mask interrupt | 32’b0 |

* **gpio\_inttype\_level**

Name: Interrupt level

Size: 32 bits

Address Offset: 0x38

Read/write access: read/write

**Table 8‑7** **gpio\_inttype\_level Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Interrupt level | Controls the type of interrupt that can occur on GPIO.  Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive.  0 – Level-sensitive (default) 1 – Edge-sensitive  Reset Value:0x0 | 32’b0 |

* **gpio\_int\_polarity:**

Name: Interrupt polarity

Size: 32 bits

Address Offset: 0x3c

Read/write access: read/write

**Table 8‑8** **gpio\_int\_polarity Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Interrupt polarity | Controls the polarity of edge or level sensitivity that can occur on input of GPIO. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive.  0 – Active-low (default) 1 – Active-high | 32’b0 |

* **gpio\_intstatus:**

Name: Interrupt status

Size: 32 bits

Address Offset: 0x40

Read/write access: read only

**Table 8‑9** **gpio\_intstatus Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Interrupt status | Interrupt status of GPIO | 32’b0 |

* **gpio\_rawintstatus:**

Name: Raw interrupt status

Size: 32 bits

Address Offset: 0x44

Read/write access: read only

**Table 8‑10** **gpio\_rawintstatus Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Raw interrupt status | Raw interrupt of status of GPIO (pre-masking bits) | 32’b0 |

* **gpio\_int\_clr:**

Name: Clear interrupt

Size: 32 bits

Address Offset: 0x4c

Read/write access: write only

**Table 8‑11** **gpio\_int\_clr Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | Clear interrupt | Controls the clearing of edge type interrupts from GPIO. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when GPIO is not configured for interrupts.  0 – No interrupt clear (default) 1 – Clear interrupt | 32’b0 |

* **gpio\_input\_data:**

Name: External GPIO

Size: 32 bits

Address Offset: 0x50

Read/write access: read only

**Table 8‑12** **gpio\_input\_data Field Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Bits | Field Name | Description | Initial value |
| 31:0 | External GPIO | When GPIO is configured as Input, then reading this location reads the values on the signal. When the data direction of GPIO is set as Output, reading this location reads the data register for GPIO. | 32’b0 |

### Work Flow

Configure GPIO output：

1. Set GPIO. gpio\_swporta\_ctl. gpio\_swporta\_ctl[0] = 1’b1, configure GPIO[0] use software mode
2. Set GPIO. gpio\_swporta\_ddr. gpio\_swporta\_ddr[0] = 1’b1, configure GPIO[0] direct is input
3. Set GPIO. gpio\_swporta\_dr. gpio\_swporta\_dr[0] = 1’b1, write data 1’b1 to gpio
4. Monitor PAD GPIO[0] == 1

Configure GPIO input:

1. Set GPIO. gpio\_swporta\_ctl. gpio\_swporta\_ctl[0] = 1’b1, configure GPIO[0] use software mode
2. Set GPIO. gpio\_swporta\_ddr. gpio\_swporta\_ddr[1] = 1’b0, configure GPIO[0] direct is output
3. Connect PAD GPIO[0] to high level
4. Read GPIO. GPIO external port register. GPIO external port register[0] == 1. Check GPIO[0] input data = 1