

# 系统级模块 用户手册

## 目录

1 Timer.....	4
1.1 特性.....	4
1.2 操作步骤.....	4
1.3 Timer IO Mapping.....	5
1.4 寄存器介绍.....	6
2 EFLASH.....	8
2.1 特性.....	8
2.2 功能描述.....	8
2.2.1 闪存结构.....	8
2.2.2 闪存读保护.....	9
2.2.3 闪存烧写和擦除操作.....	9
2.3 寄存器介绍.....	9
3 Watchdog.....	13
3.1 特性.....	13
3.2 寄存器介绍.....	13
4 GPIO.....	15
4.1 特性.....	15
4.2 寄存器介绍.....	15
5 EVSYS（事件控制单元）.....	17
5.1 简介.....	17
5.2 操作步骤.....	19
5.3 寄存器介绍.....	19
6 DMAC(DMA 控制器).....	21
6.1 特性.....	21
6.2 结构框图.....	22
6.3 DMAC 请求通道.....	22
6.4 DMAC 中断.....	22
6.5 寄存器介绍.....	23
7 系统配置寄存器.....	33
8 历史.....	62

## 关于本手册

本手册提供了系统模块在使用过程中需要了解的内容,包括模块介绍、使用步骤和寄存器等。包括了以下模块: Timer、eflash、Watchdog、GPIO、EVSYS 和系统配置寄存器。

## 关于一些记数约定

本手册使用了以下通用约定,这些约定在本手册内有效:

- 十六进制数表示为 **suffix'h**。例如: **16'ha301**, 表示该数的二进制数位宽为 16,十六进制数形式值为 **a301**。
- 十进制数表示为 **suffix'd**。例如: **2'd3**, 表示该数的二进制数位宽为 2,十进制数形式值为 3。
- 二进制数表示为 **suffix'b**。例如: **3b'001**, 表示该数的二进制数位宽为 3,二进制数形式值为 001。

# 1 Timer

TMF6200 中共有 8 个通用定时器(TMF5 系列处理器的定时器数量请参考 TMF5200 数据表和 TMF5201 数据表文档)，并且每个定时器均可输出 PWM。

## 1.1 特性

- (1) 可以用作计数器和定时器。
- (2) 支持外部 inc 引脚。内部系统时钟、内部 32K RCOSC、外部高速晶振均可用作计数器时钟源。
- (3) 支持 capture 功能，定时抓取外部 capture pin 上的计数值。
- (4) 支持独立 PWM 输出模式。

## 1.2 操作步骤

### A. 定时器计数模式操作流程：

- (1) 设置计数器初始值：设置 TMRx\_CNT 寄存器。
- (2) 设置计数周期，例如，设置 TMRx\_PR = 0xffff 则计数周期为 0xffff。
- (3) 设置 PWM 周期值：设置 TMRx\_PWM 寄存器。
- (4) 设置计数器时钟源：设置 Inc\_src\_sel (TMRx\_CON[2:0])。  
注意：如果定时器 inc 引脚用作计数器时钟源，应配置相应的 IO 映射 (IO\_MAP1 寄存器) 并将 IO 传输方向设置为输入 (DIR 寄存器)。
- (5) 配置计数器预分频：配置 psc 位 (TMRx\_CON[10:8])。
- (6) 配置计数器中断使能：配置 Tmr\_ie 位 (TMRx\_CON[13])。
- (7) 设置计数器工作模式，启动计数器后将开始计数：配置 Mode\_sel (TMRx\_CON[5:4])。

### B. 定时器 PWM 模式操作流程：

- (1) 设置定时器 PWM 输出引脚的 IO 映射并且 IO 传输方向将自动设置：设置 IO\_MAP1 寄存器和 DIR 寄存器。
- (2) 设置计数器初始值：设置 TMRx\_CNT 寄存器。
- (3) 设置 PWM 的高电平计数值：设置 TMRx\_PWM 寄存器。
- (4) 设置 PWM 的周期值：设置 TMRx\_PR 寄存器。
- (5) 设置计数器时钟源：设置 Inc\_src\_sel (TMRx\_CON[2:0])。  
注意：如果定时器 inc 引脚用作计数器时钟源，应配置相应的 IO 映射 (IO\_MAP1 寄存器) 并将 IO 传输方向设置为输入 (DIR 寄存器)。
- (6) 配置计数器预分频：配置 psc 位 (TMRx\_CON[10:8])。
- (7) 配置计数器中断使能：配置 Tmr\_ie 位 (TMRx\_CON[13])。
- (8) 设置 PWM 模式，启动计数器后将开始计数：配置 Mode\_sel (TMRx\_CON[5:4])。

### C. 定时器 capture 模式操作流程：

- (1) 配置定时器 capture 模式的输入引脚的 IO 映射并设置 IO 方向为输入。
- (2) 设置计数器初始值：设置 TMRx\_CNT 寄存器。
- (3) 配置计数器周期：设置 TMRx\_PR 寄存器。
- (4) 配置 PWM 占空比：设置 TMRx\_PWM 寄存器。
- (5) 设置计数器时钟源：设置 Inc\_src\_sel (TMRx\_CON[2:0])。

## 1Timer

注意：如果定时器 inc 引脚用作计数器时钟源，应配置相应的 IO 映射（IO\_MAPx 寄存器）并将 IO 传输方向设置为输入（DIR 寄存器）。

- (6) 设置 capture pin 的边沿触发：配置 Edge\_sel（TMRx\_CON[7:6]）。
- (7) 配置计数器预分频：配置 psc 位（TMRx\_CON[10:8]）。
- (8) 配置计数器 capture 中断使能：配置 Cap\_ie 位（TMRx\_CON[12]）。
- (9) capture 模式选择位，当启动计数器后将启动计数：配置 Mode\_sel 位（TMRx\_CON[5:4]）。
- (10) 当 Capture pending（TMRx\_CON[14]）置 1 时，TMRx\_PWM 的值 = 固定时间内 capture pin 上的事件计数值。

### 1.3 Timer IO Mapping

Module	Pin	io_map1[8]	io_map1[9]	io_map1[0]	io_map1[1]
Timer0	tmr0pwm	PC12	PD9		
	tmr0inc			PA0	PD4
	tmr0cap			PA0	PD4
Module	Pin	io_map1[10]	io_map1[11]	io_map1[2]	io_map1[3]
Timer1	tmr1pwm	PC13	PD10		
	tmr1inc			PA1	PD5
	tmr1cap			PA1	PD5
Module	Pin	io_map1[12]	io_map1[13]	io_map1[4]	io_map1[5]
Timer2	tmr2pwm	PC14	PD11		
	tmr2inc			PA2	PC15
	tmr2cap			PA2	PC15
Module	Pin	io_map1[14]	io_map1[15]	io_map1[6]	io_map1[7]
Timer3	tmr3pwm	PC15	PD12		
	tmr3inc			PA3	PC14
	tmr3cap			PA3	PC14
Module	Pin	io_map1[20]		io_map1[16]	
Timer10	tmr10pwm	PD6			
	tmr10inc			PA4	
	tmr10cap			PA4	
Module	Pin	io_map1[21]		io_map1[17]	
Timer11	tmr11pwm	PD7			
	tmr11inc			PA5	
	tmr11cap			PA5	

**1Timer**

Module	Pin	io_map1[22]	io_map1[18]	
Timer12	tmr12pwm	PD8		
	tmr12inc		PC13	
	tmr12cap		PC13	
Module	Pin	io_map1[23]	io_map1[19]	
Timer13	tmr13pwm	PD9		
	tmr13inc		PC12	
	tmr13cap		PC12	

## 1.4 寄存器介绍

### 1-1 TMRx\_CON

地址偏移: 0x00

Bit	名称	复位	读/写	说明
15	Timer pending	0	RW	定时器 pending: (写 1 清除 pending) 0: 没有 pending 1: pending
14	Capture pending	0	RW	Capture pending: (写 1 清除 pending) 0: 没有 pending 1: pending
13	Tmr_ie	0	RW	定时器模式的中断使能位。 0: 关闭 1: 启动
12	Cap_ie	0	RW	定时器 capture 模式中断使能位。 0: 关闭 1: 启动
11	保留	-	-	保留。
10:8	psc	0	RW	定时器预分频选择位。 000: 0 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频
7:6	Edge_sel	0	RW	Capture 模式的边沿触发选择位。 00: 上边沿触发 01: 下边沿触发 1x: 上下边沿触发

**1Timer**

5:4	Mode_sel	0	RW	定时器模式选择位。 00: 关闭定时器 01: 选择定时器计数器模式 10: 选择定时器 PWM 模式 11: 选择定时器 capture 模式
3	保留	-	-	保留。
2:0	Inc_src_sel	0	RW	定时器计数器源选择位。 000: 定时器 inc 引脚上升沿 001: 定时器 inc 引脚下降沿 010: 26MHz 晶振除以 1 的时钟的上升沿和下降沿 011: 32KHz RCOSC 除以 2 的时钟的上升沿和下降沿 1xx: 系统时钟上升沿

**1-2 TMRx\_PR**

地址偏移: 0x04

Bit	名称	复位	读/写	说明
15:0	PR	16'hffff	RW	该寄存器设置计数周期。

**1-3 TMRx\_CNT**

地址偏移: 0x08

Bit	名称	复位	读/写	说明
15:0	CNT	16'h0	RW	该寄存器设置计数初始值。

**1-4 TMRx\_PWM**

地址偏移: 0x0C

Bit	名称	复位	读/写	说明
15:0	PWM	0	RW	该寄存器 PWM 占空比。

## 2 EFLASH

### 2.1 特性

- (1) 128K 字节的 eflash。
- (2) 存储器结构：
  - 主闪存空间：128K 字节；
  - 副闪存空间（系统内存）：2K 字节；
- (3) 支持带预取缓冲器的接口
- (4) 支持闪存读写操作。
- (5) 支持低功耗工作模式。

### 2.2 功能描述

#### 2.2.1 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 32 页（每页 4K 字节）分块，以页为单位设置写保护

模块	名称	地址	大小（字节）
主闪存空间	页 0	0x0800_0000 – 0x0800_0FFF	4K
	页 1	0x0800_1000 – 0x0800_1FFF	4K
	页 2	0x0800_2000 – 0x0800_2FFF	4K
	页 3	0x0800_3000 – 0x0800_3FFF	4K
	...	...	4K
	页 30	0x0801_E000 – 0x0801_EFFF	4K
	页 31	0x0801_F000 – 0x0801_FFBF	3.5K
	用户配置区	0x0801_FC00 – 0x0801_FFFF	512
副闪存空间	扇区 0	0x0802_0000 – 0x0802_01FF	512
	扇区 1	0x0802_0200 – 0x0802_03FF	512
	扇区 2	0x0802_0400 – 0x0802_05FF	512
	芯片信息区	0x0802_0600 – 0x0802_07FF	512
闪存寄存器接口	CTRLR0	0x4003_1000 – 0x4003_1003	4
	KST	0x4003_1004 – 0x4003_1007	4
	DONE	0x4003_1008 – 0x4003_100B	4
	PROG_ADDR	0x4003_1010 – 0x4003_1013	4
	PROG_DATA	0x4003_1018 – 0x4003_101B	4
	TIME_REG0	0x4003_1030 – 0x4003_1033	4
	NVR_PASSWORD	0x4003_1050 – 0x4003_1053	4
	MAIN_PASSWORD	0x4003_1054 – 0x4003_1057	4
	CRC_DMA	0x4003_1058 – 0x4003_105B	4
	CRC_OUT	0x4003_105C – 0x4003_105F	4

### 2.2.2 闪存读保护

在产品整个工作电压范围内都可以通过读操作存储指令或者数据。Flash运行在26MHz的工作频率上，若工作频率提升到30MHz以上，需要让Flash的读时序执行分频。

芯片带有cache缓冲区和预取缓冲区，提升Flash的访问效率。

若用户配置区经过自定义的配置后，当SWD/UART等Debug接口连接上时，会自动对Flash执行保护机制。

读操作有下列2个寄存器完成：

- 配置寄存器 (CTRLR0)；
- 时序0寄存器(TIME\_REG0)；

### 2.2.3 闪存烧写和擦除操作

烧写和擦除操作在整个产品工作电压范围内都可以完成。

烧写和擦除操作有下列 6 个寄存器完成，先根据烧写的时钟配置好烧写时序 (TIME\_REG1)，再配置烧写密码和地址，最后配置好编程数据，即可开始执行烧写，然后等烧写结束。在上述操作中，烧写和擦除都需要配置下列寄存器：

- (1) 时序 1 寄存器 (TIMW\_REG1)；
- (2) 密码 0 寄存器 (NVR\_PASSWORD)；
- (3) 密码 1 寄存器 (MAIN\_PASSWORD)；
- (4) 烧写地址寄存器 (PROG\_ADDR)；
- (5) 烧写数据寄存器 (PROG\_DATA)；
- (6) 状态寄存器 (DONE)；

当 Flash 正在执行烧写或擦除操作时，任何对 Flash 的访问都会令总线停顿，直到写/擦除操作完成后才会继续执行，这意味着在写/擦除 Flash 的同时不可以对它取指和访问数据。

## 2.3 寄存器介绍

### 2-1 CTRLR0

地址偏移：0x00

Bit	名称	复位值	读/写	说明
31:18	保留	0	RO	保留。
17:16	Program Clock select	0	RW	闪存烧写的时钟源选择位，建议使用晶振。 00: RC 8M 时钟 01: PLL 时钟 10: ADC PLL 时钟 11: 晶振，推荐使用
15:10	保留	0	RO	保留。
9	保留	0	RW	必须软件设置该位为 0。
8	保留	0	RW	必须软件设置该位为 0。
7	保留	0	RW	必须软件设置该位为 0。
6	保留	0	RO	保留。
5	保留	0	RW	必须软件设置该位为 0。
4	保留	1	RW	必须软件设置该位为 1。
3	保留	1	RW	必须软件设置该位为 1。

**2 EFLASH**

2	保留	0	RW	必须软件设置该位为 0。
1	保留	0	RO	保留。
0	Cache Enable	0	RW	Cache 使能控制位。 0: 关闭 1: 启动

**2-2 KST**

地址偏移: 0x04

Bit	名称	复位值	读/写	说明
31:27	保留	0	RO	保留。
26	CRC Kick Enable	0	WO	EFLASH CRC Check 使能位。
25:21	保留	0	RO	保留。
20	Cache Clear Kick Enable	0	WO	Cache 初始化触发使能位。
19:11	保留	0	RO	保留。
10	CRC Kick Start	0	WO	EFLASH CRC Check, 当此位和 CRC Kick Enable(KST[26])均置 1 时触发。
9:5	保留	0	RO	保留。
4	Cache Clear Kick Start	0	WO	Cache 初始化触发, 当此位和 Cache Clear Kick Enable (KST[20])均置 1 时触发。
3:0	保留	0	RO	保留。

**2-3 DONE**

地址偏移: 0x08

Bit	名称	复位值	读/写	说明
31:13	保留	0	RO	保留。
12	保留	0	RO	不可更改此值。
11	保留	0	RO	不可更改此值。
10	CRC Done	1	RO	CRC 计算标志位。 0: 进行中 1: 空闲
9:7	保留	0	RO	保留。
6	Program Done	1	RO	闪存烧写标志位。 0: 进行中 1: 空闲
5	保留	0	RO	保留。
4	Cache Clear Done	1	RO	Cache 初始化标志位。 0: 进行中 1: 空闲
3:2	保留	0	RO	保留。
1	Chip Erase Done	1	RO	主闪存空闲完全擦除标志位。 0: 进行中 1: 空闲
0	Sector Erase Done	1	RO	扇区 (512byte) 擦除标志位。

**2 EFLASH**

				0: 进行中 1: 空闲
--	--	--	--	-----------------

**2-4 PROG\_ADDR**

地址偏移: 0x10

Bit	名称	复位值	读/写	说明
31:24	Program Byte	0	RO	保留。
23:2	Program Address	0	RW	闪存烧写地址, 以 word 为单位进行烧写, 范围是 0~130*1024 字节。 0~128*1024 字节为主闪存区域 128*1024~130*1024 为 NVR 区域
1:0	Program address	0	RO	总为 0, 需要字对齐。

**2-5 PROG\_DATA**

地址偏移: 0x18

Bit	名称	复位值	读/写	说明
31:0	Program Data	32'h0	RW	闪存烧写数据, 需要配置好烧写地址后才可进行。 注意: 当 Readm1 (CTRLR0[9]) 有效时, 此寄存器将作为闪存烧写的数据。

**2-6 ERASE\_CTRL**

地址偏移: 0x20

Bit	名称	复位值	读/写	说明
31	Chip Erase Kick Start	0	WO	Chip 完全擦除的触发控制位。 0: 无任何效果 1: 配置好密码后触发
30	Sector Erase Kick Start	0	WO	扇区擦除触发控制位。 0: 无任何效果 1: 配置密码后触发
29:9	保留	0	RO	保留。
8:0	Erase Sector Address	0	RW	擦除的扇区选择位, 范围是 0~259。 0~255: 主闪存空间的扇区 256~259: NVR 空间的扇区

**2-7 TIME\_REG0**

地址偏移: 0x30

Bit	名称	复位值	读/写	说明
31:20	保留	0	RO	保留。
19:16	PGH	1	RW	必须软件设置该位为 1。
15:12	ADS	1	RW	必须软件设置该位为 1。
11:8	ADH	1	RW	必须软件设置该位为 1。
7:4	RW	8	RW	必须软件设置该位为 8。
3:0	RC	0	RW	读周期的最小时间为 25/30ns。

**2-8 TIME\_REG1**

Offset: 0x34

**2 EFLASH**

Bit	名称	复位值	读/写	说明
31:20	保留	0	RO	保留。
18:8	1ms unit	1000	RW	1ms 的时间配置选择位，单位为 1us。
7:0	1us unit	26	RW	1us 的时间配置选择位，默认为 26MHz。

**2-9 NVR\_PASSWORD**

地址偏移: 0x50

Bit	名称	复位值	读/写	说明
31:10	NVR password	0	RW	密码是 0x20150931。仅当成功输入密码后才可对 NVR 进行擦除和烧写。

**2-10 MAIN\_PASSWORD**

地址偏移: 0x54

Bit	名称	复位值	读/写	说明
31:10	Main password	0	RW	密码为 0x20170230，仅当成功输入密码后才能对主闪存区进行擦除和烧写。

**2-11 CRC\_ADDR**

地址偏移: 0x58

Bit	名称	复位值	读/写	说明
31:16	DMA Word Length	0	RW	CRC DMA 长度是 word 的长度。
15:0	DMA Address	0	RW	CRC DMA 的首地址，指向闪存 eflash，是 word 的物理地址。

**2-12 CRC\_OUT**

地址偏移: 0x5C

Bit	名称	复位值	读/写	说明
31:00	CRC Result	0	RO	CRC 的结果。多项式 CRC-32 如下，多项式的结果取反就是官方的结果，写入 eflash 的 CRC 需要官方的值取反表示。 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

## 3 Watchdog

TX5002 内置了一个独立看门狗，用于检测 and 解决由软件错误引起的故障。当看门狗计数器达到 0x0000 时，会触发看门狗中断，产生系统复位。

看门狗时钟由 RC32K 时钟驱动，即使 26M 晶振发生故障，它也仍然有效。可以通过配置计数周期检测应用程序非正常的过迟或过早的操作。

### 3.1 特性

- (1) 支持独立时钟源。
- (2) 支持配置计数时间。
- (3) 支持复位系统。

### 3.2 功能描述

- (1) 往键寄存器(WDG\_KEY)写入 0xCCCC 将开启看门狗，计数器开始计数。
- (2) 往键寄存器(WDG\_KEY)写入 0xDDDD 将关闭看门狗，计数器停止计数。
- (3) 定时向键寄存器(WDG\_KEY)写入 0xAAA 重载计数值，否则计数器为 0 时系统复位。
- (4) 向键寄存器(WDG\_KEY)写入 0x5555 才能配置看门狗的分频比。
- (5) 向键寄存器(WDG\_KEY)写入 0x55AA 将开启看门狗中断。
- (6) 当看门狗中断 Pending = 1 时，向键寄存器(WDG\_KEY)写入 0xAAAA 将清除 Pending。
- (7) 向键寄存器(WDG\_KEY)写入 0xAA55 将关闭看门狗中断。
- (8) 向键寄存器(WDG\_KEY)写入 0x5A5A 将使能唤醒功能。
- (9) 向键寄存器(WDG\_KEY)写入 0xA5A5 将关闭唤醒功能。

### 3.3 寄存器访问保护

看门狗的分频比配置具有写保护功能，只能先往键寄存器(WDG\_KEY)写入 0x5555 才能配置看门狗的分频比。以不同的值写入键寄存器会打乱操作顺序，寄存器就会重新被保护起来，重载操作（写入 0xAAA）也会启动写保护功能。

### 3.4 调试模式

当微控制器进入调试模式时（Corex-M3 内核停止），看门狗会继续运行，需要在调试前先关闭看门狗，否则系统会复位。

### 3.5 寄存器介绍

#### 3-1 WDG\_KEY

地址偏移：0x04

Bit	名称	复位值	读/写	说明
31:16	保留	16'b0	-	保留。
15:0	KEY	16'b0	WO	键值，可写。软件需定时向此位写入 0xAAA 或当计数器等于 0 时看门狗复位。 当 pending = 1 时，如果向此位写入 0xAAAA 将清除 pending；

### 3 Watchdog

				如果向此位写入 0x5555 将对 WDG PSR 寄存器执行读操作； 如果向此位写入 0xCCCC 将启动看门狗； 如果向此位写入 0x55AA 将使能看门狗中断；如果向此位写入 0xAA55 将关闭看门狗中断；如果向此位写入 0x5A5A 将使能唤醒功能；如果向此位写入 0xA5A5 将关闭唤醒功能； 如果向此位写入 0xDDDD 将关闭看门狗功能。
--	--	--	--	---

#### 3-1 WDT\_CON

地址偏移：0x00

Bit	名称	复位值	读/写	说明
31:7	保留	25'b0	-	保留。
6	wdt_pend	0	RO	看门狗计满标志位。
5	int_enable	0	RO	看门狗中断使能控制位。 0: 计满时系统复位 1: 计满时产生中断
4	wdte	0	RW	当 WDG_KEY = 0Xcccc 时, wdte = 1; 当 WDG_KEY = 0XDDDD 时, wdte = 0。
3:0	wdt_psr	0000	RW	分频系数, 在配置此位前, 需要先配置 WDG_KEY = 0x5555。

## 4 GPIO

GPIO 是一个可编程通用目的 I/O 外设 (Programmable General Purpose Programming I/O)。在 TMF6 系列中一共有 64 个 I/O 端口并且都支持中断。

### 4.1 特性

- (1) 支持数字 I/O 输入和输出。
- (2) 支持 GPIO 输出的 bitset 和 bit reset。
- (3) 支持强制数字 GPIO 功能。

GPIOx (x=A,B,C,D) 的物理基地址如下:

名称	物理基地址
GPIOA	0x40022000
GPIOB	0x40024000
GPIOC	0x40025000
GPIOD	0x40027000

### 4.2 寄存器介绍

#### 4-1 DR

地址偏移: 0x0000

Bit	名称	复位	读/写	说明
31:0	DR	R/W	0x0	GPIOx 端口数据寄存器。

#### 4-2 DIR

地址偏移: 0x0004

Bit	名称	复位	读/写	说明
31:0	DIR	R/W	0x0	GPIOx 数据传输方向设置位。 0: 输入 1: 输出

#### 4-3 INTMASK

地址偏移: 0x0008

Bit	名称	复位	读/写	说明
31:0	INTMASK	R/W	0x0	GPIOx 端口中断屏蔽寄存器。 0: 屏蔽 1: 不屏蔽

#### 4-4 PU0EN

地址偏移: 0x000C

Bit	名称	复位	读/写	说明
31:0	PU0EN	R/W	0x0	GPIOx 端口上拉寄存器。 0: 不发生上拉 1: 上拉 10K

#### 4 GPIO

##### 4-5 PD0EN

地址偏移: 0x0018

Bit	名称	复位	读/写	说明
31:0	PD0EN	R/W	0x0	GPIOx 端口下拉寄存器。 1: 下拉 10K 0: 不下拉

##### 4-6 DS

地址偏移: 0x0024

Bit	名称	复位	读/写	说明
31:0	DS	R/W	0x0	GPIO 端口驱动增强寄存器。 1: 24mA 0: 8mA

##### 4-7 HY

地址偏移: 0x0028

Bit	名称	复位	读/写	说明
31:0	HY	R/W	0x0	GPIOx I/O 输入迟滞使能寄存器。 0: 关闭 1: 启动

##### 4-8 OD

地址偏移: 0x002C

Bit	名称	复位	读/写	说明
31:0	OD	R/W	0x0	GPIOx 端口开漏寄存器。 0: 关闭开漏 1: 启动开漏

##### 4-9 SR

地址偏移: 0x0030

Bit	名称	复位	读/写	说明
31:0	SR	R/W	0x0	GPIOx 端口慢速寄存器。 0: 关闭 1: 启动

##### 4-10 DIE

地址偏移: 0x0034

Bit	名称	复位	读/写	说明
31:0	DIE	R/W	0x0	GPIOx 数字 GPIO 强制使能寄存器。 0: 不强制数字 GPIO 功能 1: 强制数字 GPIO 功能

##### 4-11BSRS

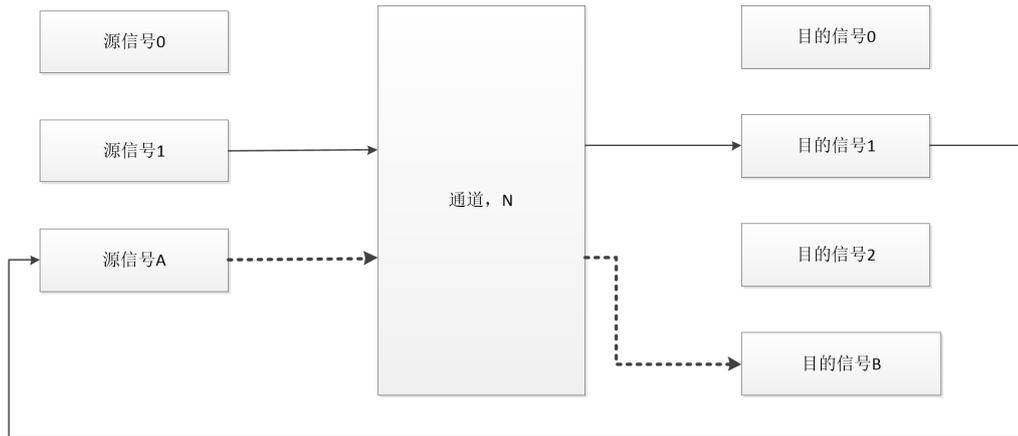
地址偏移: 0x0038

Bit	名称	复位	读/写	说明
31:0	BSRS	R/W	0x0	GPIOx 端口 bit set/reset(H16/L16)设置寄存器。

## 5 EVSYS (事件控制单元)

### 5.1 简介

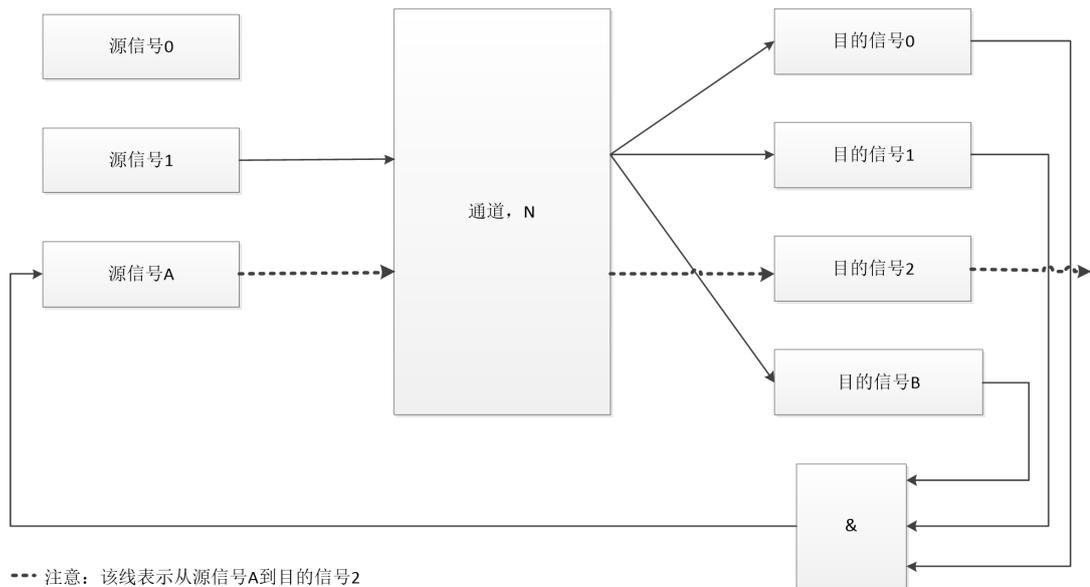
从源触发器和目标触发器列表中选择一组到通道的链接，如下图所示：源信号 1 到目标信号 1，目标信号 1 到源信号 A，源信号 A 到目标信号 B，然后输出最终中断标志。软件可以为每个触发配置目标信号的触发编号 M。例如，目标信号 1 需要运行 M 次，然后在源信号 1 触发目标信号 1 之后触发源信号 A。



--- 注意：源信号A到目的信号B

图 5-1 一对一的结构框图

从源触发器和目标触发器列表中选择一组到通道的链接。如下图所示，源信号 1 可以同时触发目标信号 0/1 / B。当所有信号都完成时，触发源信号 A，然后由源信号 A 触发目标信号 2，最后产生中断信号。



--- 注意：该线表示从源信号A到目的信号2

图 5-2 一对一结构框图

表 5-1 源触发器和目的触发器

NO.	源触发器	目的触发器	说明
0	IIR0	IIR0	寄存器控制可以在每个触发器中启动哪个通道滤波器。
1	IIR1	IIR1	寄存器控制可以在每个触发器中启动哪个通道滤波器。
2	IIR2	IIR2	寄存器控制可以在每个触发器中启动哪个通道滤波器。
3	FIR0	FIR0	寄存器控制可以在每个触发器中启动哪个通道滤波器。
4	FIR1	FIR1	寄存器控制可以在每个触发器中启动哪个通道滤波器。
5	FIR2	FIR2	寄存器控制可以在每个触发器中启动哪个通道滤波器。
6	ARCTAN0	ARCTAN0	
7	ARCTAN1	ARCTAN1	
8	ARCTAN2	ARCTAN2	
9	DFTRANS0	DFTRANS0	
10	DFTRANS1	DFTRANS1	
11	DFTRANS2	DFTRANS2	
12	FFT0	FFT0	
13	FFT1	FFT1	
14	FFT2	FFT2	
15	FFT0_dma	FFT0_dma	
16	FFT1_dma	FFT1_dma	
17	FFT2_dma	FFT2_dma	
18	MATRIX0	MATRIX0	
19	MATRIX1	MATRIX1	
20	MATRIX2	MATRIX2	
21	MATRIX3	MATRIX3	
22	RMS0	RMS0	
23	RMS1	RMS1	
24	RMS2	RMS2	
25	SINCOS0	SINCOS0	
26	SINCOS1	SINCOS1	
27	DATA DMA	DATA DMA	
28	HCC	HCC	
29	SARADC0		
30	SARADC1		
31	SARADC2		
32	SARADC3		
33	SARADC4		
34	SARADC5		
35	SARADC6		
36	SARADC7		
37	SARADC8		
38	SARADC9		

**5 EVSYS(事件控制单元)**

39	SARADC10		
40	SARADC11		
41	SARADC12		
42	SARADC13		

注意：源信号为 42，目标信号编号为 28，通道编号 N 为 16。

## 5.2 操作步骤

- (1) 从表中选择源信号 (SRC\_CH\_CON0) 和目标信号 (DST\_CH\_CON0)，选择一个要连接的通道 (CH\_ENA)。需要中断时启用中断 (CH\_INT\_ENA)。
- (2) 开始时，可以选择 CPU 触发 (CPU\_KST) 或硬件自触发 (如 ADC 触发)。

## 5.3 寄存器介绍

### 5-1 CH\_ENA

地址偏移: 0x00

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Channel Enable	0	RW	通道使能位。 0: 关闭 1: 使能

### 5-2 CH\_INT\_ENA

地址偏移: 0x04

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Channel Interrupt Enable	0	RW	通道中断使能位，每一位代表一个通道。 0: 关闭 1: 使能

### 5-3 CPU\_KST

地址偏移: 0x08

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	CPU Kick Start	0	WO	写‘1’启动相应的通道，每一位代表一个通道。

### 5-4 CH\_PND\_CLR

地址偏移: 0x10

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Channel Pending Clear	0	WO	写‘1’清除相应通道的挂起标志，包括 CPU 和硬件挂起标志。

### 5-5 CH\_CPU\_PND

地址偏移: 0x14

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。

**5 EVSYS(事件控制单元)**

N-1:0	Channel CPU Pending	0	RO	相应通道的 CPU 挂起标志。
-------	---------------------	---	----	-----------------

**5-6 CH\_HW\_PND**

地址偏移: 0x18

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Channel Hardware Pending	0	RO	相应通道的硬件挂起标志。

**5-7 CH\_MODE0**

地址偏移: 0x1C

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Channel Mode Select	0	RW	模式选择位。 0: 输入源信号到寄存器, 寄存器直接驱动目标信号。 1: 输入源信号直接驱动目标信号

**5-8 SRC\_CH\_CON0**

地址偏移: 0x20 - 0x5C

Bit	名称	复位值	读/写	功能
31:0	Source channel number	0	RW	源触发信号 0-31 的选择位。

**5-9 SRC\_CH\_CON1**

地址偏移: 0x60 - 0x9C

Bit	名称	复位值	读/写	功能
31:16	Source channel counter	0	RO	通道源触发信号一次完成目标通道的触发次数。
15:11	保留	0	RO	保留。
10:0	Source channel number	0	RO	源触发信号 32-42 的选择位。

**5-10 DST\_CH\_CON0**

地址偏移: 0xA0 - 0xDC

Bit	名称	复位值	读/写	功能
31:N	保留	0	RO	保留。
N-1:0	Destination channel enable	0	RW	目的通道使能位, 每一位代表一个通道。

## 6 DMAC(DMA 控制器)

DMA 控制器提供了一种在外设和存储器之间或存储器和存储器之间传输数据的硬件方式，无需 MCU 的干预，这避免了 MCU 的大规模数据复制的多次进入中断，并最终提高了整体系统性能。每个 DMA 控制器都包含 FIFO 和两个 AHB 总线接口，使 DMA 能够有效地传输数据。DMA 控制器有四个通道，每个通道可以分配给一个或多个特定的外围设备进行数据传输。两个内置总线介体用于处理 DMA 请求的优先级问题。32 位 Cortex-M 内核和 DMA 控制器都通过系统总线处理数据并引入仲裁来处理它们之间的竞争关系。当 MCU 和 DMA 指示相同的外设时，MCU 将在特定的总线周期中暂停。总线 MATRIX 使用固定优先级算法。

### 6.1 特性

- (1) 4 个通道，每个通道连接 16 个特定的外设请求；
- (2) 存储器和外围设备支持单一传输，4 拍、8 拍和 16 拍增量突发传输；
- (3) 支持外设 DMA：2 个 SPI，3 个 UART，2 个 IIC；
- (4) 当外围设备向存储器发送数据时支持改变存储器；
- (5) 支持对所有内部存储的 DMA 访问；
- (6) 支持软件优先级（低，中，高，超高）和硬件优先级（通道数越低，优先级越高）。
- (7) 存储器和外设的数据传输宽度可配置为：字节，半字，字。
- (8) 存储器和外设的数据传输支持固定和增量寻址。
- (9) 支持循环传输模式。
- (10) 支持单数据传输和多种数据传输方式：
  - 多种数据传输方式：当存储器数据和外围数据的宽度不同时，自动打包/解包数据；
  - 单数据传输模式：当且仅当 FIFO 为空时，数据从源地址读取，存储在 FIFO 中，FIFO 的数据写入目标地址；
- (11) 每个通道有 5 种类型的事件标志和独立的中断，支持启用和重置中断。

## 6.2 结构框图

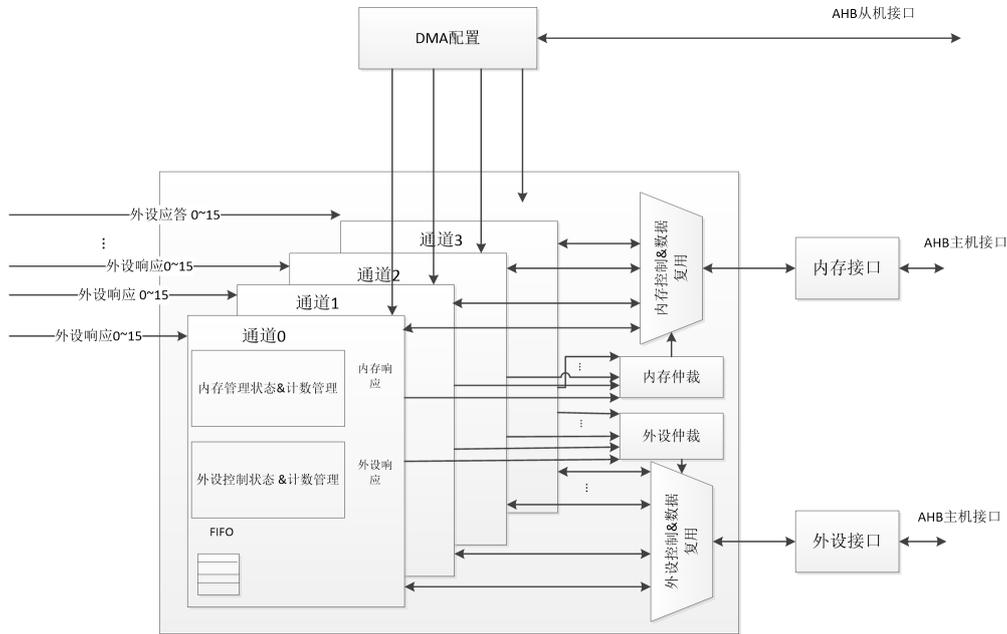


图 6-1 DMAC 结构框图

## 6.3 DMAC 请求通道

表 DMAC 请求通道

DMAC 请求通道	连接的外设	说明
dma req channel 0	I2C_0_dma_tx_req	-
dma req channel 1	I2C_0_dma_rx_req	-
dma req channel 2	I2C_1_dma_tx_req	-
dma req channel 3	I2C_1_dma_rx_req	-
dma req channel 4	SPI0_dma_tx_req	-
dma req channel 5	SPI0_dma_rx_req	-
dma req channel 6	SPI1_dma_tx_req	-
dma req channel 7	SPI1_dma_rx_req	-
dma req channel 8	UART0_dma_tx_req	-
dma req channel 9	UART0_dma_rx_req	-
dma req channel 10	UART1_dma_tx_req	-
dma req channel 11	UART1_dma_rx_req	-
dma req channel 12	UART2_dma_tx_req	-
dma req channel 13	UART2_dma_rx_req	-

## 6.4 DMAC 中断

DMAC 中的每个通道都有 5 种类型的中断源：

## 6 DMAC (DMA 控制器)

- (1) IntBlock: 在 DMA 块传输完成时生成到目的外设。
- (2) IntDstTran: 目的端的握手接口 (硬件或软件握手接口) 完成请求的单/突发事务的最后一次 AHB 传输后产生。
- (3) IntErr: 在 DMA 传输过程中, HRESP 总线上的一个 AHB 从机接收到 ERROR 响应时产生 IntErr 中断。并且取消 DMA 传输、禁用通道。
- (4) IntSrcTran: 源端的握手接口 (硬件或软件握手接口) 完成请求的单/突发事务的最后一次 AHB 传输后产生此中断。
- (5) IntTfr: DMA 传输完成时在目的外设产生此中断。

注意:

- 如果通道的目的地址是 memory, 那么该通道不会产生 IntDstTran 中断。因此, 不用设置相应位。
- 如果源地址或目的地址是 memory, 那么 IntSrcTran/IntDstTran 中断会被忽略, 因为 memory 中不支持“DMA 事务级”。

### 6.5 寄存器介绍

Register6-1 DmaCfgReg0

Offset: 0x398

Bit	名称	复位值	读/写	说明
63:1	保留	63'b0	N/A	-
0	DMA_EN	0	RW	DMAC 使能控制位, 须在所有通道开始活动前完成设置。 0: 禁用 1: 使能 如果任意一个通道是 active 状态时总通道使能位被清零, 那么 DMA_EN (DmaCfgReg[0]) 仍然返回 1 指示有通道为 active 状态, 直到硬件关闭所有通道的活动时 DMA_EN (DmaCfgReg[0]) 才返回 0。

Register6-2 ChEnReg

Offset: 0x3a0

Bit	名称	复位值	读/写	说明
63:12	保留	52'b0	N/A	-
11	CH3_EN_WE	0	W	Channel 3 的写使能控制位。 0: 禁用 1: 使能
10	CH2_EN_WE	0	W	Channel 2 写使能控制位。 0: 禁用 1: 使能
9	CH1_EN_WE	0	W	Channel 1 写使能控制位。 0: 禁用 1: 使能
8	CH0_EN_WE	0	W	Channel 0 写使能控制位。

**6 DMAC (DMA 控制器)**

				0: 禁用 1: 使能
7:4	保留	0000	N/A	–
3	CH3_EN	0	R/W	Channel 3 使能控制位。 0: 禁用通道 1: 启动通道
2	CH2_EN	0	R/W	Channel 2 使能控制位。 0: 禁用通道 1: 启动通道
1	CH1_EN	0	R/W	Channel 1 使能控制位。 0: 禁用通道 1: 启动通道
0	CH0_EN	0	R/W	Channel 0 使能控制位。 0: 禁用通道 1: 启动通道

注意：在 DMA 最后一次传输完成后，通过硬件禁用通道可以清除 CH0\_EN、CH1\_EN、CH2\_EN、CH3\_EN 位。软件可以对此位轮询以确定何时可用于新的 DMA 传输。

**Register6-3 SAR0~7**

Offset: 0x000/0x058/0x0b0/0x108/0x160/0x1b8/0x210/0x268

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	–
31:0	SAR	32'b0	R/W	DMA 传输的当前源地址，每次源传输后更新。在块传输中，SINC (CTLx[10:9]) 决定每个源传输是否进行地址递增、递减或不变。

注意：SAR 的烧写地址必须与 SRC\_TR\_WIDTH (CTL[6:4]) 对齐。DMA 通道使能前可由软件烧写起始源地址，或在 DMA 开始传输前可由 LLI 更新起始源地址。在 DMA 传输过程中，此寄存器会被更新以反映当前 AHB 传输的源地址。

**Register6-4 DAR0~7**

Offset: 0x008/0x060/0x0b8/0x110/0x168/0x1c0/0x218/0x270

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	–
31:0	DAR	32'b0	R/W	DMA 传输的当前目的地址，在每次目的传输后更新。在块传输中，DINC (CTL0~7[8:7]) 决定每次目的传输的地址递增、递减或保持不变。

注意：DAR 的烧写地址必须与 DST\_TR\_WIDTH (CTL0~7[3:1]) 对齐。在 DMA 通道使能前可由软件烧写起始目的地址，或在启动 DMA 传输前可由 LLI 更新起始目的地址。DMA 传输时，此寄存器会更新以反映当前 AHB 传输的目的地址。

对于 SARx / DARx 寄存器的重新对齐，在特定情况下，在连续的多块 DMA 传输期间，目标地址可能在一个块的结尾与下一个块的开始之间变得不对齐。发生这种情况时，DMA 会在下一个块开始之前重新对齐目标地址。在以下 DMA 传输设置方案中，DMA 将自动重新对齐目标地址：

目的端连续多块传输，且

**6 DMAC (DMA 控制器)**

$DST\_TR\_WIDTH(CTL0\sim7[3:1]) > SRC\_TR\_WIDTH(CTL0\sim7[6:4])$ , 且  
 $(BLOCK\_TS(CTL0\sim7[43:32]) * SRC\_TR\_WIDTH(CTL0\sim7[6:4]) /$   
 $DST\_TR\_WIDTH(CTL0\sim7[3:1]) \neq \text{整数}$   
 (其中  $SRC\_TR\_WIDTH(CTL0\sim7[6:4])$ ,  $DST\_TR\_WIDTH(CTL0\sim7[3:1])$  是传输的字节宽度)

注意: DAR 的烧写地址必须与  $DST\_TR\_WIDTH(CTL0\sim7[3:1])$  对齐。在 DAM 通道使能前可由软件烧写起始目的地址, 或在 DMA 开始传输之前由 LLI 更新起始目的地址。当正在进行 DMA 传输时, 此寄存器会被更新以反映当前 AHB 传输的目的地址。

对于 SARx/DARx 寄存器的重新对齐, 在特定情况下, 在连续多块 DMA 传输期间, 目标地址在一个块的结尾和后一块的起始之间变得不对齐。发生这种情况时, 在下一块开始前 DMA 会重新对齐目标地址。在下列 DMA 传输设置方案中, DMA 将自动重新对齐目标地址:

- 目的端连续多块传输, 且
- $DST\_TR\_WIDTH(CTL0\sim7[3:1]) > SRC\_TR\_WIDTH(CTL0\sim7[6:4])$ , 且
- $(BLOCK\_TS(CTL0\sim7[43:32]) * SRC\_TR\_WIDTH(CTL0\sim7[6:4]) / DST\_TR\_WIDTH(CTL0\sim7[3:1]) \neq \text{整数}$  (其中  $SRC\_TR\_WIDTH(CTL0\sim7[6:4])$ ,  $DST\_TR\_WIDTH(CTL0\sim7[3:1])$  是传输字节宽度)

Register6-5 CTL0~7

Offset: 0x018/0x070/0x0c8/0x120/0x178/0x1d0/0x228/0x280

Bit	名称	复位值	读/写	说明
63:45	保留	19'b0	N/A	-
44	DONE	0	R/W	Done bit.
43:32	BLOCK_TS	12'h2	R/W	当 DMAC 是流控制器时, 在通道使能之前, 用户可以向此字段写入数据以指示块的大小。写入 BLOCK_TS 的数据指示每个块传输的单个事务的数量。 注意: 当源/目的外设作为流控制器时, 读返回的最大的块大小为 4095, 但实际块大小更大。
31:29	保留	/	N/A	-
28	LLP_SRC_EN	0	R/W	仅当 LLP_SRC_EN 为高电平且 $LOC(LLPx[31:2])$ 为非 0 时使能源端的块链。
27	LLP_DST_EN	0	R/W	仅当 LLP_DST_EN 为高电平且 $LOC(LLPx[31:2])$ 为非 0 时使能目的端的块链。
26:25	SMS	00	R/W	源设备主接口层选择控制位。从访问的源设备 (外设或 memory) 识别主接口层。 00: AHB master 1 01: AHB master 2 1x: reserved
24:23	DMS	00	R/W	目标设备主接口层选择控制位。识别目标设备 (外设或 memory) 所在的主接口层。 00: AHB master 1 01: AHB master 2

**6 DMAC (DMA 控制器)**

				1x: reserved
22:20	TT_FC	11	R/W	传输类型和流控制位。支持下列传输类型： <ul style="list-style-type: none"> <li>➢ Memory 到 Memory</li> <li>➢ Memory 到外设</li> <li>➢ 外设到 Memory</li> <li>➢ 外设到外设</li> </ul> 流控制可以被分配给 DMAC、源端外设或目的外设。
19	保留	/	N/A	-
18	保留	/	R/W	-
17	保留	/	R/W	-
16:14	SRC_MSIZ	10	R/W	源突发事务长度。 源突发事务长度。每次从相应的硬件或软件握手接口发出源突发事务请求时，从源读取每个宽度为 SRC_TR_WIDTH (CTLx [6: 4]) 的数据项数。
13:11	DEST_MSIZ	10	R/W	目的突增事务长度。每次相应的硬件或软件握手接口发出目的突增事务请求时，每个宽度为 DST_TR_WIDTH(CTLx[3:1])的数据项数会被写入目的地址。
10:9	SINC	2'b00	R/W	源地址递增递减控制位。该位指示每次源传输时是否递增/递减源地址。如果器件从固定地址的源外设 FIFO 处取数据，那么设置此位为“不改变地址”。 <ul style="list-style-type: none"> <li>00: 递增</li> <li>01: 递减</li> <li>1x: 地址不变</li> </ul> 注意：对与下一个 SRC_TR_WIDTH (CTLx [6: 4]) 边界的对齐进行递增或递减操作。
8:7	DINC	2'b00	R/W	目标地址递增递减控制位。该位指示每次目的传输时是否递增或递减目的地址。如果设备正在向固定地址的目的外设 FIFO 写数据，此字段设置为“不变”。 <ul style="list-style-type: none"> <li>00: 递增</li> <li>01: 递减</li> <li>1x: 不变</li> </ul> 注意：递增递减操作是为了与下一个 DST_TR_WIDTH (CTLx [3: 1]) 边界对齐。
6:4	SRC_TR_WIDTH	000	R/W	源传输宽度。该值必须 ≤ 32bit。 <ul style="list-style-type: none"> <li>000: 8 bit 的数据宽度</li> <li>001: 16 bit 的数据宽度</li> <li>002: 32 bit 的数据宽度</li> </ul>

**6 DMAC (DMA 控制器)**

3:1	DST_TR_WIDTH	000	R/W	目的传输宽度。该值必须 ≤32bit。 000: 8 bit 的数据宽度 001: 16 bit 的数据宽度 002: 32 bit 的数据宽度
0	INT_EN	1	R/W	中断使能位。此位设置为 1 时，所有中断产生源都会打开，用作通道所有中断的全局屏蔽位。此位为 0 时 raw* interrupt 寄存器仍然有效。

注意：启动通道前用户需要先烧写寄存器。SRC\_MSIZE 和 DEST\_MSIZE (AHB 的突增式长度) 解码如下：

SRC_MSIZE/DEST_MSIZE	要传输的数据项数 (宽度为 SRC_TR_WIDTH 或 DST_TR_WIDTH)
000	1
001	4
010	8
011	16
100	32
101	64
110	128
111	256

TT\_FC 的字段解码

TT_FC	传输类型	流控制器
000	Memory - Memory	DMAC
001	Memory - 外设	DMAC
010	外设 - Memory	DMAC
011	外设 - 外设	DMAC
100	外设 - Memory	外设
101	外设 - 外设	源外设
110	Memory - 外设	外设
111	外设 - 外设	目的外设

Register6-6 SSTAT0~7

Offset: 0x020/0x078/0x0d0/0x128/0x180/0x1d8/0x230/0x288

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	-
31:0	SSTAT	32'b0	R/W	硬件从 SSTATARx 寄存器的内容指向的地址检索的源状态信息。

Register6-7 DSTAT0~7

Offset: 0x028/0x080/0x0d8/0x130/x188/0x1e0/0x238/0x290

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	-
31:0	DSTAT	32'b0	R/W	硬件通过 DSTATARx 寄存器内容指向的地址检索的目的状态信息。

注意：该寄存器是到达 LLI 的 DSTATx 寄存器位置的目的地状态信息的临时占位符。目的状

**6 DMAC (DMA 控制器)**

态信息应该从 LLI 的 DSTATx 寄存器位置开始软件检索目，而不是通过 DMAC 从接口读取此寄存器进行检索。

**Register6-8 SSTATAR0~7**

Offset: 0x030/0x088/0x0e0/0x138/x190/0x1e8/0x240/0x298

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	-
31:0	SSTATAR	32'b0	R/W	硬件可以从中获取源状态信息的指针，该源状态信息在 SSTATx 寄存器中注册，并在下一个块开始之前写入 LLI 的 SSTATx 寄存器位置。

注意：在每个块传输完成之后，硬件可以在用户定义的地址（SSTATARx 寄存器的内容指向的地址）恢复源状态信息。用户可以选择系统 memory 中的任何位置以提供一个 32 位的值指示源传输的状态。

**Register6-9 DSTATAR0~7**

Offset: 0x038/0x090/0x0e8/0x140/x198/0x1f0/0x248/0x2a0

Bit	名称	复位值	读/写	说明
63:32	保留	32'b0	N/A	-
30:0	DSTATAR	32'b0	R/W	硬件可以从中获取目的状态信息的指针，该目的状态信息在 DSTATx 寄存器中注册，并在下一个块开始之前写入 LLI 的 DSTATx 寄存器位置。

注意：在每个块传输完成之后，硬件可以在用户定义的地址（DSTATARx 寄存器的内容指向的地址）恢复目的状态信息。用户可以选择系统 memory 中的任何位置以提供一个 32 位的值指示目的传输的状态。

**Register6-10 CFG0~7**

Offset: 0x040/0x098/0x0f0/0x148/x1a0/0x1f8/0x250/0x2a8

Bit	名称	复位值	读/写	说明
63:47	保留	32'b0	N/A	-
46:43	DEST_PER	0000	R/W	分配一个硬件握手接口给通道 x 的目的地址。
42:39	SRC_PER	0000	R/W	分配一个硬件握手接口给通道 x 的源地址。
38	SS_UPD_EN	0	R/W	源状态更新使能位。源状态信息只能在 SSTATARx 寄存器指向的位置被获取，可以被存入 SSTATx 寄存器并且当此字段为 1 时写入 LLI 的 SSTATx 的位置。
37	DS_UPD_EN	0	R/W	目的状态更新使能位。目的状态信息只能在 DSTATARx 寄存器指向的位置被获取，可以被存入 DSTATx 寄存器并且当此字段为 1 时写入 LLI 的 DSTATx 的位置。
36:34	PROTCTL	100	R/W	用于驱动 AHB HPROT[3:1]总线的保护控制位。推荐使用 HPROT 的默认值指示“无缓存”、“无缓冲”和特权数据访问。复位值用于指示这种访问。HPROT[0]绑定为高电平，因为所有传输均为数据访问，没有操作码获取。此字段（PROTCTL[2:0]）的每一位与 HPROT[3:1]主接口均为一对一的映射。映射如下所示：

**6 DMAC (DMA 控制器)**

				PROTCTL[x]	HPROT[x]
				1	HPROT[0]
				PROTCTL[1]	HPROT[1]
				PROTCTL[2]	HPROT[2]
				PROTCTL[3]	HPROT[3]
33	FIFO_MODE	0	R/W	FIFO 模式选择。此字段确定在服务一个突增事务请求之前 FIFO 中有多少空间或数据可用。 0: 空间/数据用于指定的传输宽度的单个 AHB 传输 1: 可用数据 ≥ 目的传输的 FIFO 深度的一半, 且可用空间 > 源传输的 FIFO 的一半。免责条款会在突增事务请求的结尾或在块传输的结尾	
32	FCMODE	0	R/W	流控制模式。当目的外设为流控制器时, 此字段确定什么时候服务源事务请求。 0: 为 0 时服务源事务请求。启动数据预取。 1: 目的事务请求发生后才开始服务源事务请求。在此模式中, 从源传输的数据总数会被限制。所以可以确保在目标块终止之前传输到目标。关闭数据预取。	
31	RELOAD_DST	0	R/W	目标自动重装。在每个块结束时, DARx 寄存器会从它的初始值开始自动重装以进行多块传输。然后才初始化新的块传输。	
30	RELOAD_SRC	0	R/W	源自动重装。在每个块结束时, SARx 寄存器会从它的初始值开始自动重装以进行多块传输。然后才初始化新的块传输。	
29:20	MAX_ABRST	10'h0	R/W	最大的 AMBA 突增长度。最大的 AMBA 突增长度是用于此通道的 DMA 传输。0 值指示软件不限制此通道 DMA 传输的最大 AMBA 突增长度。	
19	SRC_HS_POL	0	R/W	源握手接口极性。 0: 高电平 1: 低电平	
18	DST_HS_POL	0	R/W	目标握手极性。 0: 高电平 1: 低电平	
17	LOCK_B	0	R/W	总线锁定位。当此位为 1 时, AHB 总线主信号 hlock 在 CFGx[LOCK_B_L] 指定的时间内断言。	
16	LOCK_CH	0	R/W	通道锁定位。当通道被授权控制主机总线接口且此字段被断言时, 在 LOCK_CH_L(CTHx[13:12]) 指定的持续时间内没有其他通道被授权控制主机总线接口。向主总线接口仲裁器指示该通道要求在	

**6 DMAC (DMA 控制器)**

				LOCK_CH_L (CTHx [13:12]) 指定的持续时间内独占访问主总线接口。
15:14	LOCK_B_L	00	R/W	总线锁定等级。指示应用于 LOCK_B(CFGx[17])的持续时间。 00: DMA 传输完成的持续时间 01: DMA 块传输完成的持续时间 1x: 完成 DMA 事务的持续时间
13:12	LOCK_CH_L	00	R/W	通道锁定等级。指示应用于 LOCK_CH(CFGx[16])的持续时间。 00: DMA 传输完成的持续时间 01: DMA 块传输完成的持续时间 1x: 完成 DMA 事务的持续时间
11	HS_SEL_SRC	1	R/W	源软件/硬件握手选择位。此字段选择对通道的源请求有效的握手接口。 0: 硬件握手接口。忽略软件初始化的事务请求 1: 软件握手接口。忽略硬件初始化的事务请求 如果源外设为 <b>memory</b> , 那么忽略此位。
10	HS_SEL_DST	1	R/W	目的软件/硬件握手选择位。此字段选择对通道的目的请求有效的握手接口。 0: 硬件握手接口。忽略软件初始化的事务请求 1: 软件握手接口。忽略硬件初始化的事务请求。 如果目的外设为 <b>memory</b> , 那么忽略此字段。
9	FIFO_EMPTY	1	R/W	通道 FIFO 是否为空的指示位。此字段可以和 CH_SUSP(CFGx[8])结合使用, 以彻底禁用通道。 0: 通道 FIFO 为空 1: 通道 FIFO 不为空
8	CH_SUSP	0	R/W	通道暂停控制位。暂停所有来自源的 DMA 数据传输直到此字段清零。无法保证当前事务已经完成。也可以与 FIFO_EMPTY(CFGx[9])结合使用, 以彻底禁用通道而不丢失任何数据。 0: 不暂停 1: 暂停来自源的 DMA 传输
7:5	CH_PRIOR	Channel number	R/W	通道优先级。优先级为 7 是最高优先级, 优先级为 0 是最低优先级。此字段必须在 0~15 的范围内烧写。如果烧写值超出这个范围会出错。注意: 复位值是通道值。例如, 通道 0 的情况下复位值为 0; 通道 1 的情况下复位值为 1。
4:0	保留	5'b0	N/A	-

注意: 用户需要先烧写此寄存器才能启用通道。

Register6-11 RawBlock/RawDstTran/RawErr/RawSrcTran/RawTfr

Offset: 0x2c8/0x2d8/0x2e0/0x2d0/0x2c0

**6 DMAC (DMA 控制器)**

Bit	名称	复位值	读/写	说明
63:16	保留	48'b0	N/A	-
15:0	DSC	16'b0	R/W	原始中断状态。

注意：只有作软件测试目的时这些寄存器才能使用写访问。在正常操作下，不建议向该寄存器进行写操作。在屏蔽前中断事件会被存入下列原始中断状态寄存器中：**RawBlock**, **RawDstTran**, **RawErr**, **RawSrcTran**, and **RawTfr**。每个通道在这些寄存器中都有各个位一一对应。可以在 **ClearTfr**, **ClearBlock**, **ClearSrcTran**, **ClearDstTran**, **ClearErr** 这些寄存器中的相应位写 1 以清除这些原始中断状态寄存器的各个位。

Register6-12 StatusBlock/StatusDstTran/StatusErr/StatusSrcTran/StatusTfr

Offset: 0x2f0/0x300/0x308/0x2f8/0x2e8

Bit	名称	复位值	读/写	说明
63:4	保留	60'b0	N/A	-
3:0	STATUS	0000	R	中断状态

注意：屏蔽后，来自所有通道的所有中断事件都可以存入这些中断状态寄存器中：**StatusBlock**, **StatusDstTran**, **StatusErr**, **StatusSrcTran**, **StatusTfr**。每个中断状态寄存器都会分配一个位给每个通道。这些寄存器的内容用于产生中断信号。

Register6-13 MaskBlock/MaskDstTran/MaskErr/MaskSrcTran/MaskTfr

Offset: 0x318/0x328/0x330/0x320/0x310

Bit	名称	复位值	读/写	说明
63: 12	保留	52'b0	N/A	-
11:8	INT_MASK_WE	0000	W	中断屏蔽写使能。 0: 禁用写操作 1: 启动写操作
7:4	保留	0000	N/A	-
3:0	INT_MASK	0000	R/W	中断屏蔽 0: 已屏蔽 1: 未屏蔽

Register6-14 ClearBlock/ClearDstTran/ClearErr/ClearSrcTran/ClearTfr

Offset: 0x340/0x350/0x358/0x348/0x338

Bit	名称	复位值	读/写	说明
63: 4	保留	60'b0	N/A	-
3:0	CLEAR	0000	W	中断清除。 0: 无影响 1: 清除中断

注意：在同一个周期内，通过向这些清除寄存器（**ClearBlock**/**ClearDstTran**/**ClearErr**/**ClearSrcTran**/**ClearTfr**）的相应位写 1，可以清除原始状态和状态寄存器中的各个位。每个中断清除寄存器都会分配一个位给各个通道。写 0 无效。这些寄存器均不可读。

Register6-15 StatusInt

Offset: 0x360

Bit	名称	复位值	读/写	说明
63:5	保留	59'b0	N/A	-

**6 DMAC (DMA 控制器)**

4	ERR	0	R	与 StatusErr 寄存器的内容进行“或”运算。
3	DSTT	0	R	与 StatusDst 寄存器的内容进行“或”运算。
2	SRCT	0	R	与 StatusSrcTran 寄存器的内容进行“或”运算。
1	BLOCK	0	R	与 StatusBlock 寄存器的内容进行“或”运算。
0	TFR	0	R	与 StatusTfr 寄存器的内容进行“或”运算。

注意：这五个状态寄存器（StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran, StatusErr）的内容都进行“或”运算，以便为组合状态寄存器（StatusInt）的每个中断类型各产生一个位。此寄存器只可读。

## 7 系统配置寄存器

7-1 CLK\_CON0

地址偏移: 0x00

Bit	名称	复位值	读/写	说明
24:23	gpiod_deb_clk_sel	00	RW	GPIO PD 口去抖动时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: CPU 时钟 11: RC 32KHz
22:21	gpioc_deb_clk_sel	00	RW	GPIO PC 口去抖动时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: CPU 时钟 11: RC 32KHz
20:19	gpiob_deb_clk_sel	00	RW	GPIO PB 口去抖动时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: CPU 时钟 11: RC 32KHz
18:17	gpioa_deb_clk_sel	00	RW	GPIO PA 口去抖动时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: CPU 时钟 11: RC 32KHz
16:15	ADC_PLL_refclk_sel	00	RW	ADC PLL 参考时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: 关闭 11: DFT 时钟
14:13	保留	00	RW	保留。
12:11	PLL_refclk_sel	00	RW	PLL 参考时钟选择。 00: XOSC 26MHz 01: RC 8MHz 10: 关闭 11: DFT 时钟
10:9	保留	00	RW	保留。
8:7	epwm_clk_sel	00	RW	EPWM 时钟选择。 00: RC 8MHz 01: XOSC 26MHz 10: PLL 156MHz 11: ADC PLL 156MHz

**7 系统配置寄存器**

6:4	保留	0	RW	不可更改此值。
3:2	保留	00	RW	保留。
1:0	sysclk_sel	00	RW	系统时钟选择。 00: RC 32KHz 01: XOSC 26MHz 10: PLL 的 N 分频 11: RC 8MHz

**7-2 CLK\_CON1**

地址偏移: 0x04

Bit	名称	复位值	读/写	说明
23:16	Apb1_div_val	8'd0	RW	APB1 总线时钟分频值设置。
15:8	Apb0_div_val	8'd0	RW	APB0 总线时钟分频值设置。
7:0	Sysclk_div_val	8'd0	RW	系统时钟分频值设置。

**7-3 CLK\_CON2**

地址偏移: 0x08

Bit	名称	复位值	读/写	说明
15:12	epwm_div_value	0000	RW	EPWM 时钟分频值设置。
11:8	保留	0000	RO	保留。
7:4	adcpll_div_value	0000	RW	ADCPLL 时钟分频值设置。
3:0	pll_div_value	0000	RW	SYSPLL 时钟分频值设置。

**7-4 CLK\_CON3**

地址偏移: 0x0C

Bit	名称	复位值	读/写	说明
31:24	gpiod_deb_clk_div_value	8'd0	RW	GPIO PD 口去抖动时钟分频值设置。
23:16	gpioc_deb_clk_div_value	8'd0	RW	GPIO PC 口去抖动时钟分频值设置。
15:8	gpiob_deb_clk_div_value	8'd0	RW	GPIO PB 口去抖动时钟分频值设置。
7:0	gpioa_deb_clk_div_value	8'd0	RW	GPIO PA 口去抖动时钟分频值设置。

**7-5 CLK\_CON4**

地址偏移: 0x10

Bit	名称	复位值	读/写	说明
31	gpio0_clk_en	1	RW	GPIO PA 模块时钟使能控制。 0: 关闭 1: 启动
30	gpio1_clk_en	1	RW	GPIO PB 模块时钟使能控制。 0: 关闭 1: 启动
29	gpio2_clk_en	1	RW	GPIO PC 模块时钟使能控制。 0: 关闭

**7 系统配置寄存器**

				1: 启动
28	gpio3_clk_en	1	RW	GPIO PD 模块时钟使能控制。 0: 关闭 1: 启动
27	spi0_clk_en	1	RW	SPI0 的模块时钟使能控制。 0: 关闭 1: 启动
26	spi1_clk_en	1	RW	SPI1 的模块时钟使能控制。 0: 关闭 1: 启动
25:24	保留	11	RW	保留。
23	uart0_clk_en	1	RW	Uart0 的模块时钟使能控制。 0: 关闭 1: 启动
22	uart1_clk_en	1	RW	Uart1 的模块时钟使能控制。 0: 关闭 1: 启动
21	uart2_clk_en	1	RW	Uart2 的模块时钟使能控制。 0: 关闭 1: 启动
20	iic0_clk_en	1	RW	IIC0 的模块时钟使能控制。 0: 关闭 1: 启动
19	iic1_clk_en	1	RW	IIC1 的模块时钟使能控制。 0: 关闭 1: 启动
18	保留	1	RW	保留。
17	crc_clk_en	1	RW	CRC32 的模块时钟使能控制。 0: 关闭 1: 启动
16	保留	1	RW	保留。
15	qei_clk_en	1	RW	QEI 的模块时钟使能控制。 0: 关闭 1: 启动
14	evsys_clk_en	1	RW	EVSYS 的模块时钟使能控制。 0: 关闭 1: 启动
13	can_clk_en	1	RW	CAN 的模块时钟使能控制。 0: 关闭 1: 启动
12	timer0_clk_en	1	RW	timer0 的模块时钟使能控制。 0: 关闭

**7 系统配置寄存器**

				1: 启动
11	timer1_clk_en	1	RW	timer1 的模块时钟使能控制。 0: 关闭 1: 启动
10:5	保留	6'h3f	RW	保留。
4	saradc_clk_en	1	RW	SARADC 的模块时钟使能控制。 0: 关闭 1: 启动
3	dac_clk_en	1	RW	DAC 的模块时钟使能控制。 0: 关闭 1: 启动
2	pwrace_clk_en	1	RW	ERPU 的模块时钟使能控制。 0: 关闭 1: 启动
1	eflash_clk_en	1	RW	eflash 的模块时钟使能控制。 0: 关闭 1: 启动
0	eflash_mem_clk_en	1	RW	eflash 的烧写时钟使能控制。 0: 关闭 1: 启动

**7-6 CLK\_CON5**

地址偏移: 0x14

Bit	名称	复位值	读/写	说明
31:30	保留	1	RW	保留。
29	advtmr_clk_en	1	RW	Advance timer 的时钟使能控制。 0: 关闭 1: 启动
28	fsaradc_clk_en	1	RW	FSARADC 时钟使能控制。 0: 关闭 1: 启动
27	sram16_clk_en	1	RW	SRAM16 时钟使能控制。 0: 关闭 1: 启动
26	sram15_clk_en	1	RW	SRAM15 时钟使能控制。 0: 关闭 1: 启动
25	sram14_clk_en	1	RW	SRAM14 时钟使能控制。 0: 关闭 1: 启动
24	sram13_clk_en	1	RW	SRAM13 时钟使能控制。 0: 关闭 1: 启动

**7 系统配置寄存器**

23	sram12_clk_en	1	RW	SRAM12 时钟使能控制。 0: 关闭 1: 启动
22	sram11_clk_en	1	RW	SRAM11 时钟使能控制。 0: 关闭 1: 启动
21	sram10_clk_en	1	RW	SRAM10 时钟使能控制。 0: 关闭 1: 启动
20	sram9_clk_en	1	RW	SRAM9 时钟使能控制。 0: 关闭 1: 启动
19	sram8_clk_en	1	RW	SRAM8 时钟使能控制。 0: 关闭 1: 启动
18	sram7_clk_en	1	RW	SRAM7 时钟使能控制。 0: 关闭 1: 启动
17	sram6_clk_en	1	RW	SRAM6 时钟使能控制。 0: 关闭 1: 启动
16	sram5_clk_en	1	RW	SRAM5 时钟使能控制。 0: 关闭 1: 启动
15	sram4_clk_en	1	RW	SRAM4 时钟使能控制。 0: 关闭 1: 启动
14	sram3_clk_en	1	RW	SRAM3 时钟使能控制。 0: 关闭 1: 启动
13	sram2_clk_en	1	RW	SRAM2 时钟使能控制。 0: 关闭 1: 启动
12	sram1_clk_en	1	RW	SRAM1 时钟使能控制。 0: 关闭 1: 启动
11	sram0_clk_en	1	RW	SRAM0 时钟使能控制。 0: 关闭 1: 启动
10	rom_clk_en	1	RW	ROM 时钟使能控制。 0: 关闭 1: 启动

**7 系统配置寄存器**

9	hcc_clk_en	1	RW	HCC 时钟使能控制。 0: 关闭 1: 启动
8	保留	1	RW	不可修改此值。
7:4	保留	1111	RW	保留。
3	ahb0_clk_en	1	RW	AHB0 总线时钟使能控制。 0: 关闭 1: 启动
2	ahb1_clk_en	1	RW	AHB1 总线时钟使能控制。 0: 关闭 1: 启动
1	ahb2_clk_en	1	RW	AHB2 总线时钟使能控制。 0: 关闭 1: 启动
0	dmac_clk_en	1	RW	DMAC 的模块时钟使能控制。 0: 关闭 1: 启动

**7-7 SYS\_CON0**

地址偏移: 0x18

Bit	名称	复位值	读/写	说明
31	保留	0	RW	保留。
30	保留	0	RW	保留。
29	保留	0	RW	不可修改此值。
28	保留	0	RW	不可更改此值。
27	sys_mem_err_resp_en	0	RW	CPU&DMA 总线故障响应使能控制。 0: 关闭 1: 启动
26	sys_mem_access_err_int_en	0	RW	内存访问出现故障时的中断使能控制。 0: 关闭 1: 启动
25	spi0_mcp_map1	0	RW	SPI0 MCP IO map1 使能控制。 0: 关闭 1: 启动
24	spi0_mcp_map0	0	RW	SPI0 MCP IO map1 使能控制。 0: 关闭 1: 启动
23	保留	0	RW	保留。
22	lockup_enable	0	RW	加锁使能控制。 0: 关闭 1: 启动
21	can_fd_enable	0	RW	Can FD 功能使能控制。 0: 关闭

**7 系统配置寄存器**

				1: 启动
20	wdt_reset_en	0	RW	WDT1 的复位使能控制。 0: 关闭 1: 启动
19	wdt_pause	0	RW	WDT1 的暂停使能控制。 0: 关闭 1: 启动
18	wdt_speed_up	0	RW	WDT1 的加速使能控制。 0: 关闭 1: 启动
17:15	保留	0	RW	不可更改此值。
14	pmu_enable	0	RW	PMU 使能控制。 0: 关闭 1: 启动
13	hosc_loss_nmi_en	0	RW	晶振异常时, 产生不可屏蔽中断的使能控制。 0: 关闭 1: 启动
12	nmi_int_enable	0	RW	引脚产生不可屏蔽中断的使能控制。 0: 关闭 1: 启动
11	保留	0	RW	保留。
10:9	保留	0	RW	不可更改此值。
8	cmppin4_test_en	0	RW	SARADDA CMP 信号 4 输出到 IO 的使能控制。 0: 关闭 1: 启动
7	cmppin3_test_en	0	RW	SARADDA CMP 信号 3 输出到 IO 的使能控制。 0: 关闭 1: 启动
6	cmppin2_test_en	0	RW	SARADDA CMP 信号 2 输出到 IO 的使能控制。 0: 关闭 1: 启动
5	cmppin1_test_en	0	RW	SARADDA CMP 信号 1 输出到 IO 的使能控制。 0: 关闭 1: 启动
4	cmppin0_test_en	0	RW	SARADDA CMP 信号 0 输出到 IO 的使能控制。 0: 关闭 1: 启动

**7 系统配置寄存器**

3	保留	0	RW	不可更改此值。
2	uart_mode_protect	0	RW	UART 模式保护使能控制。 0: 关闭 1: 启动
1	保留	0	RW	保留。
0	保留	1	RW	保留。

**7-8 SYS\_CON1**

地址偏移: 0x1C

Bit	名称	复位值	读/写	说明
31:5	保留	27'b0	RW	保留。
4	advtmr_soft_rst_	1	RW	Advtmr0 的模块软复位选择控制位。 0: 软复位 1: 释放软复位
3	gpiod_deb_soft_rst_	1	RW	GPIO D 去抖动的模块软复位选择控制。 0: 软复位 1: 释放软复位
2	gpioc_deb_soft_rst_	1	RW	GPIO C 去抖动的模块软复位选择控制。 0: 软复位 1: 释放软复位
1	gpiob_deb_soft_rst_	1	RW	GPIO B 去抖动的模块软复位选择控制。 0: 软复位 1: 释放软复位
0	gpioa_deb_soft_rst_	1	RW	GPIO A 去抖动的模块软复位选择控制。 0: 软复位 1: 释放软复位

**7-9 SYS\_CON2**

地址偏移: 0x20

Bit	名称	复位值	读/写	说明
31	保留	1	RW	保留。
30	epwm_soft_rst_	1	RW	EPWM 模块软复位控制。 0: 软复位 1: 释放软复位
29	fsaradc_soft_rst_	1	RW	Fsaradc 的模块软复位控制。 0: 软复位 1: 释放软复位
28	保留	1	RW	不可修改此值。
27	qei_soft_rst_	1	RW	QEI 的模块软复位控制。 0: 软复位 1: 释放软复位
26	evsys_soft_rst_	1	RW	EVSYS 的模块软复位控制。 0: 软复位 1: 释放软复位

**7 系统配置寄存器**

25	can_soft_rst_	1	RW	CAN 的模块软复位控制。 0: 软复位 1: 释放软复位
24	gpio_soft_rst_	1	RW	GPIO 的模块软复位控制。 0: 软复位 1: 释放软复位
23	wdt_soft_rst_	1	RW	WATCHDOG 的模块软复位控制。 0: 软复位 1: 释放软复位
22	timer_soft_rst_	1	RW	TIMER 的模块软复位控制。 0: 软复位 1: 释放软复位
21	crc_soft_rst_	1	RW	CRC 的模块软复位控制。 0: 软复位 1: 释放软复位
20	保留	1	RW	保留。
19	hcc_soft_rst_	1	RW	HCC 的模块软复位控制。 0: 软复位 1: 释放软复位
18	saradc_soft_rst_	1	RW	SARADC 的模块软复位控制。 0: 软复位 1: 释放软复位
17	pwrace_soft_rst_	1	RW	ERPU 的模块软复位控制。 0: 软复位 1: 释放软复位
16	adtmr0_soft_rst_	1	RW	ADTIMER0 的模块软复位控制。 0: 软复位 1: 释放软复位
15	uart2_soft_rst_	1	RW	UART2 的模块软复位控制。 0: 软复位 1: 释放软复位
14	uart1_soft_rst_	1	RW	UART1 的模块软复位控制。 0: 软复位 1: 释放软复位
13	uart0_soft_rst_	1	RW	UART0 的模块软复位控制。 0: 软复位 1: 释放软复位
12	spi1_soft_rst_	1	RW	SPI1 的模块软复位控制。 0: 软复位 1: 释放软复位
11	spi0_soft_rst_	1	RW	SPI0 的模块软复位控制。 0: 软复位

**7 系统配置寄存器**

				1: 释放软复位
10	iic1_soft_rst_	1	RW	IIC1 的模块软复位控制。 0: 软复位 1: 释放软复位
9	iic0_soft_rst_	1	RW	IIC0 的模块软复位控制。 0: 软复位 1: 释放软复位
8	dmac_soft_rst_	1	RW	DMAC 的模块软复位控制。 0: 软复位 1: 释放软复位
7	eflash_soft_rst_	1	RW	eFlash 的模块软复位控制。 0: 软复位 1: 释放软复位
6	apb1_soft_rst_	1	RW	APB1 总线的的模块软复位控制。 0: 软复位 1: 释放软复位
5	apb0_soft_rst_	1	RW	APB0 总线的的模块软复位控制。 0: 软复位 1: 释放软复位
4	ahb2_soft_rst_	1	RW	AHB2 总线的模块软复位控制。 0: 软复位 1: 释放软复位
3	ahb1_soft_rst_	1	RW	AHB1 总线的模块软复位控制。 0: 软复位 1: 释放软复位
2	ahb0_soft_rst_	1	RW	AHB0 总线的模块软复位控制。 0: 软复位 1: 释放软复位
1	fpll1_soft_rst_	1	RW	Frac pll1 的模块软复位控制。 0: 软复位 1: 释放软复位
0	fpll0_soft_rst_	1	RW	Frac pll0 的模块软复位控制。 0: 软复位 1: 释放软复位

7-10SYS\_CON3

地址偏移: 0x24

Bit	名称	复位值	读/写	说明
31:16	pb_in_deb_en	16'h0000	RW	GPIO PB 口去抖动使能控制, 1bit 代表 1 位的去抖动使能状态。
15:0	pa_in_deb_en	16'h0000	RW	GPIO PA 口去抖动使能控制, 1bit 代表 1 位的去抖动使能状态。

7-11SYS\_CON4

**7 系统配置寄存器**

地址偏移: 0x28

Bit	名称	复位值	读/写	说明
31:16	pd_in_deb_en	16'h0000	RW	GPIO PD 口去抖动使能控制, 1bit 代表 1 位的去抖动使能状态。
15:0	pc_in_deb_en	16'h0000	RW	GPIO PC 口去抖动使能控制, 1bit 代表 1 位的去抖动使能状态。

7-12SYS\_CON5

地址偏移: 0x2C

Bit	名称	复位值	读/写	说明
31:16	iob_syn_en	16'h0000	RW	GPIO PB 口输入同步使能控制, 1bit 代表 1 位的去抖动使能状态。
15:0	ioa_syn_en	16'h0000	RW	GPIO PA 口输入同步使能控制, 1bit 代表 1 位的去抖动使能状态。

7-13SYS\_CON6

地址偏移: 0x30

Bit	名称	复位值	读/写	说明
31:16	iod_syn_en	16'h0000	RW	GPIO PD 口输入同步使能控制, 1bit 代表 1 位的去抖动使能状态。
15:0	ioc_syn_en	16'h0000	RW	GPIO PC 输入同步使能控制, 1bit 代表 1 位的去抖动使能状态。

7-14SYS\_CON7

地址偏移: 0x34

Bit	名称	复位值	读/写	说明
25	pd12_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD12GPIO 关闭 IO 模拟功能 1: PD12 GPIO 启动 IO 模拟功能
24	pd11_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD11 GPIO 关闭 IO 模拟功能 1: PD11 GPIO 启动 IO 模拟功能
23	pd10_afio_pie	0	RW	模拟 GPIO 的数字使能控制 0: PD10 GPIO 关闭 IO 模拟功能 1: PD10 GPIO 启动 IO 模拟功能
22	pd9_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD9 GPIO 关闭 IO 模拟功能 1: PD9 GPIO 启动 IO 模拟功能
21	pd5_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD5 GPIO 关闭 IO 模拟功能 1: PD5 GPIO 启动 IO 模拟功能
20	pd4_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD4 GPIO 关闭 IO 模拟功能 1: PD4 GPIO 启动 IO 模拟功能
19	pd3_afio_pie	0	RW	模拟 GPIO 的数字使能控制。

**7 系统配置寄存器**

				0: PD3 GPIO 关闭 IO 模拟功能 1: PD3 GPIO 启动 IO 模拟功能
18	pd2_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD2 GPIO 关闭 IO 模拟功能 1: PD2 GPIO 启动 IO 模拟功能
17	pd1_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD1 GPIO 关闭 IO 模拟功能 1: PD1 GPIO 启动 IO 模拟功能
16	pd0_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PD0 GPIO 关闭 IO 模拟功能 1: PD0 GPIO 启动 IO 模拟功能
15	pa15_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA15 GPIO 关闭 IO 模拟功能 1: PA15 GPIO 启动 IO 模拟功能
14	pa14_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA14 GPIO 关闭 IO 模拟功能 1: PA14 GPIO 启动 IO 模拟功能
13	pa13_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA13 GPIO 关闭 IO 模拟功能 1: PA13 GPIO 启动 IO 模拟功能
12	pa12_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA12 GPIO 关闭 IO 模拟功能 1: PA12 GPIO 启动 IO 模拟功能
11	pa11_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA11 GPIO 关闭 IO 模拟功能 1: PA11 GPIO 启动 IO 模拟功能
10	pa10_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA10 GPIO 关闭 IO 模拟功能 1: PA10 GPIO 启动 IO 模拟功能
9	pa9_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA9 GPIO 关闭 IO 模拟功能 1: PA9 GPIO 启动 IO 模拟功能
8	pa8_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA8 GPIO 关闭 IO 模拟功能 1: PA8 GPIO 启动 IO 模拟功能
7	pa7_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA7 GPIO 关闭 IO 模拟功能 1: PA7 GPIO 启动 IO 模拟功能
6	pa6_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA6 GPIO 关闭 IO 模拟功能 1: PA6 GPIO 启动 IO 模拟功能
5	pa5_afio_pie	0	RW	模拟 GPIO 的数字使能控制。

**7 系统配置寄存器**

				0: PA5 GPIO 关闭 IO 模拟功能 1: PA5 GPIO 启动 IO 模拟功能
4	pa4_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA4 GPIO 关闭 IO 模拟功能 1: PA4 GPIO 启动 IO 模拟功能
3	pa3_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA3 GPIO 关闭 IO 模拟功能 1: PA3 GPIO 启动 IO 模拟功能
2	pa2_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA2 GPIO 关闭 IO 模拟功能 1: PA2 GPIO 启动 IO 模拟功能
1	pa1_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA1 GPIO 关闭 IO 模拟功能 1: PA1 GPIO 启动 IO 模拟功能
0	pa0_afio_pie	0	RW	模拟 GPIO 的数字使能控制。 0: PA0 GPIO 关闭 IO 模拟功能 1: PA0 GPIO 启动 IO 模拟功能

7-15SYS\_CON8

地址偏移: 0x38

Bit	名称	复位值	读/写	说明
31:28	保留	0000	RW	保留。
27:25	Port7_wkup_sel	000	RW	端口 7 唤醒模式选择控制。 000: PD8 001: PD9 010: PD10 011: PD11 100: PD12 101: PD13 110: PD14 111: PD15
24:22	Port6_wkup_sel	000	RW	端口 6 唤醒模式选择控制。 000: PD0 001: PD1 010: PD2 011: PD3 100: PD4 101: PD5 110: PD6 111: PD7
21:19	Port5_wkup_sel	000	RW	端口 5 唤醒模式选择控制。 000: PC8 001: PC9

**7 系统配置寄存器**

				010: PC10 011: PC11 100: PC12 101: PC13 110: PC14 111: PC15
18:16	Port4_wkup_sel	000	RW	端口 4 唤醒模式选择控制。 000: PC0 001: PC1 010: PC2 011: PC3 100: PC4 101: PC5 110: PC6 111: PC7
15:13	Port3_wkup_sel	000	RW	端口 3 唤醒模式选择控制。 000: PB8 001: PB9 010: PB10 011: PB11 100: PB12 101: PB13 110: PB14 111: PB15
12:10	Port2_wkup_sel	000	RW	端口 2 唤醒模式选择控制。 000: PB0 001: PB1 010: PB2 011: PB3 100: PB4 101: PB5 110: PB6 111: PB7
9:7	Port1_wkup_sel	000	RW	端口 1 唤醒模式选择控制。 000: PA8 001: PA9 010: PA10 011: PA11 100: PA12 101: PA13 110: PA14 111: PA15

**7 系统配置寄存器**

6:4	Port0_wkup_sel	000	RW	端口 0 唤醒模式选择控制。 000: PA0 001: PA1 010: PA2 011: PA3 100: PA4 101: PA5 110: PA6 111: PA7
3:0	保留	32'b0	RW	保留。

**7-16AIPCON0**

地址偏移: 0x3C

Bit	名称	复位值	读/写	说明
31:28	保留	0000	RW	保留。
27	pll1_opasel_vdd	1	RW	ADCPLL VI 转换器的输入对选择控制。 0: PMOS 对输入 1: 轨对轨输入
26	pll1_lpfsel_vdd	0	RW	ADCPLL 参考时钟选择控制。 0: 26MHz 1: 8MHz
25	pll0_opasel_vdd	1	RW	SYSPLL VI 转换器输入对选择控制。 0: PMOS 对输入 1: 轨对轨输入
24	pll0_lpfsel_vdd	0	RW	SYSPLL 参考时钟选择控制。 0: 6MHz 1: 8MHz;
23:20	pll1_ten_vdd	0000	RW	ADCPLL 测试信号输出选择控制。 0000: 关闭 0001: ICO 0010: VFBL 0100: CPOUT 1000: 保留
19	pll1_vcodelt_vdd	0	RW	ADCPLL CPOUT_ADKEY 电压使能控制。 0: 关闭 1: 启动
18:17	pll1_gvco_vdd	10	RW	ADCPLL VCO 频率调节增益控制。 00: 最小值 11: 最大值
16:15	pll1_gcp_vdd	11	RW	ADCPLL CP 电流配置。 00: 1uA 01: 2uA 10: 3uA

**7 系统配置寄存器**

				11: 4uA
14:13	pll1_band_vdd	01	RW	ADCPLL 输出的中心频率选择控制。 00: 312MHz 01: 396MHz 10: 466MHz 11: 520MHz
12	pll1_en_vdd	0	RW	ADCPLL 使能控制。 0: 关闭 1: 启动
11:8	pll0_ten_vdd	0000	RW	SYSPLL 测试信号输出选择控制。 0000: 关闭 0001: ICO 0010: VFBL 0100: CPOUT 1000: 保留
7	pll0_vcodet_vdd	0	RW	SYSPLL CPOUT_ADKEY 电压使能控制。 0: 关闭 1: 启动
6:5	pll0_gvco_vdd	10	RW	SYSPLL VCO 频率调节增益控制。 00: 最小值 11: 最大值
4:3	pll0_gcp_vdd	11	RW	SYSPLL CP 电流配置选择。 00: 1uA 01: 2uA 10: 3uA 11: 4uA
2:1	pll0_band_vdd	01	RW	SYSPLL 输出的中心频率配置选择。 00: 312MHz 01: 396MHz 10: 466MHz 11: 520MHz
0	pll0_en_vdd	0	RW	SYSPLL 使能控制。 0: 关闭 1: 启动

**7-17 AIPCON1**

地址偏移: 0x40

Bit	名称	复位值	读/写	说明
31:18	保留	0	RW	保留。
17	rc8m_ten	0	RW	RC8M 测试使能控制。 0: 关闭 1: 启动
16	rc8m_en	1	RW	RC8M 模拟功能使能控制。

**7 系统配置寄存器**

				0: 关闭 1: 启动
15	hxosc_hy_vdd	1	RW	HXOSC 比较器迟滞选择控制。 0: 不发生迟滞 1: 迟滞 $\pm 10\%$
14:12	hxosc_dr_vdd	101	RW	HXOSC 启动电流的配置选择。 000: 0.25mA 001: 0.375mA 010: 0.5mA 011: 0.625mA 100: 0.75mA 101: 0.875mA 110: 1.0mA 111: 1.125mA
11:8	hxosc_cto_vdd	0000	RW	HXOSC HXO 口的谐振电容微调设置。 电容值=CTI*0.2pF; 最小值=0pF; 最大值=3pF;
7:4	hxosc_cti_vdd	0000	RW	HXOSC HXI 口谐振电容微调设置。 电容值=CTI*0.2pF; 最小值=0pF; 最大值=3pF;
3:1	hxosc_cs_vdd	101	RW	HXOSC 谐振电容粗调设置。 000: 0pF 001: 2.5pF 010: 5pF 011: 7.5pF 100: 10pF 101: 12.5pF 110: 15pF; 111: 17.5pF; 每个装置都有一个额外的 3.5pF 寄生电容。
0	hxosc_en_vdd	1	RW	HXOSC 使能控制。 0: 关闭 1: 启动

**7-18IO\_MAP**

地址偏移: 00x44

Bit	名称	复位值	读/写	说明
31	epwmsynco_map	0	RW	EPWM synco 引脚 map0 的使能控制。 0: 关闭 1: 启动
30	epwm2b_map	0	RW	EPWM2b 引脚 map0 的使能控制。

**7 系统配置寄存器**

				0: 关闭 1: 启动
29	epwm2a_map	0	RW	EPWM2a 引脚 map0 的使能控制。 0: 关闭 1: 启动
28	epwm1b_map	0	RW	EPWM1b 引脚 map0 的使能控制。 0: 关闭 1: 启动
27	epwm1a_map	0	RW	EPWM1a 引脚 map0 的使能控制。 0: 关闭 1: 启动
26	epwm0b_map	0	RW	EPWM0b 引脚 map0 的使能控制。 0: 关闭 1: 启动
25	epwm0a_map	0	RW	EPWM0a 引脚 map0 的使能控制。 0: 关闭 1: 启动
24	uart2_map2	0	RW	Uart2 引脚 map2 的使能控制。 0: 关闭 1: 启动
23	adtmr_map1	0	RW	Adtimer 引脚 map1 的使能控制 0: 关闭 1: 启动
22	svpwm_map0	0	RW	SVPWM 引脚 map0 的使能控制。 0: 关闭 1: 启动
21	spi1_map1	0	RW	SPI1 引脚 map1 的使能控制。 0: 关闭 1: 启动
20	spi0_map1	0	RW	SPI0 引脚 map1 的使能控制。 0: 关闭 1: 启动
19	smbus1_map0	0	RW	IIC1 系统管理总线 map0 的使能控制。 0: 关闭 1: 启动
18	smbus0_map1	0	RW	IIC0 系统管理总线 map1 的使能控制。 0: 关闭 1: 启动
17	smbus0_map0	0	RW	IIC0 系统管理总线 map0 的使能控制。 0: 关闭 1: 启动
16	uart2_map1	0	RW	Uart2 引脚 map1 的使能控制。

**7 系统配置寄存器**

				0: 关闭 1: 启动
15	uart1_map1	0	RW	Uart1 引脚 map1 的使能控制。 0: 关闭 1: 启动
14	uart0_map1	0	RW	Uart0 引脚 map1 的使能控制。 0: 关闭 1: 启动
13	trace_map0	0	RW	Trace 引脚 map0 的使能控制。 0: 关闭 1: 启动
12	spwm_map0	0	RW	SPWM 引脚 map0 的使能控制。 0: 关闭 1: 启动
11	jtag_map0	1	RW	Jtag 引脚 map0 的使能控制。 0: 关闭 1: 启动
10	adtmr_map0	0	RW	Adtimer 引脚 map0 的使能控制。 0: 关闭 1: 启动
9	can_map0	0	RW	Can 引脚 map0 的使能控制。 0: 关闭 1: 启动
8	qei_map0	0	RW	QEI 引脚 map0 的使能控制。 0: 关闭 1: 启动
7	mii_map0	0	RW	MII 引脚 map0 的使能控制。 0: 关闭 1: 启动
6	uart2_map0	0	RW	Uart2 引脚 map0 的使能控制。 0: 关闭 1: 启动
5	uart1_map0	0	RW	Uart1 引脚 map0 的使能控制。 0: 关闭 1: 启动
4	uart0_map0	0	RW	Uart0 引脚 map0 的使能控制。 0: 关闭 1: 启动
3	spi1_map0	0	RW	SPI1 引脚 map0 的使能控制。 0: 关闭 1: 启动
2	spi0_map0	0	RW	SPI0 引脚 map0 的使能控制。

**7 系统配置寄存器**

				0: 关闭 1: 启动
1	iic1_map0	0	RW	IIC0 引脚 map0 的使能控制。 0: 关闭 1: 启动
0	iic0_map0	0	RW	IIC0 引脚 map0 的使能控制。 0: 关闭 1: 启动

**7-19IO\_MAP1**

地址偏移: 0x48

Bit	名称	复位值	读/写	说明
31	epwm6b_map	0	RW	EPWM6b 引脚 map0 的使能控制。 0: 关闭 1: 启动
30	epwm6a_map	0	RW	EPWM6a 引脚 map0 的使能控制。 0: 关闭 1: 启动
29	epwm5b_map	0	RW	EPWM5b 引脚 map0 的使能控制。 0: 关闭 1: 启动
28	epwm5a_map	0	RW	EPWM5a 引脚 map0 的使能控制。 0: 关闭 1: 启动
27	epwm4b_map	0	RW	EPWM4b 引脚 map0 的使能控制。 0: 关闭 1: 启动
26	epwm4a_map	0	RW	EPWM4a 引脚 map0 的使能控制。 0: 关闭 1: 启动
25	epwm3b_map	0	RW	EPWM3b 引脚 map0 的使能控制。 0: 关闭 1: 启动
24	epwm3a_map	0	RW	EPWM3a 引脚 map0 的使能控制。 0: 关闭 1: 启动
23	tmr13_pwm_map0	0	RW	Timer13 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
22	tmr12_pwm_map0	0	RW	Timer12 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
21	tmr11_pwm_map0	0	RW	Timer11 pwm 引脚 map0 的使能控制。

**7 系统配置寄存器**

				0: 关闭 1: 启动
20	tmr10_pwm_map0	0	RW	Timer10 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
19	tmr13_inccap_map0	0	RW	Timer13 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
18	tmr12_inccap_map0	0	RW	Timer12 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
17	tmr11_inccap_map0	0	RW	Timer11 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
16	tmr10_inccap_map0	0	RW	Timer10 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
15	tmr3_pwm_map1	0	RW	Timer3 pwm 引脚 map1 的使能控制。 0: 关闭 1: 启动
14	tmr3_pwm_map0	0	RW	Timer3 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
13	tmr2_pwm_map1	0	RW	Timer2 pwm 引脚 map1 的使能控制。 0: 关闭 1: 启动
12	tmr2_pwm_map0	0	RW	Timer2 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
11	tmr1_pwm_map1	0	RW	Timer1 pwm 引脚 map1 的使能控制。 0: 关闭 1: 启动
10	tmr1_pwm_map0	0	RW	Timer1 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
9	tmr0_pwm_map1	0	RW	Timer0 pwm 引脚 map1 的使能控制。 0: 关闭 1: 启动
8	tmr0_pwm_map0	0	RW	Timer0 pwm 引脚 map0 的使能控制。 0: 关闭 1: 启动
7	tmr3_inccap	0	RW	Timer3 inc/cap 引脚 map1 的使能控制。

**7 系统配置寄存器**

	_map1			0: 关闭 1: 启动
6	tmr3_inccap_map0	0	RW	Timer3 inc/cap pin map0 的使能控制。 0: 关闭 1: 启动
5	tmr2_inccap_map1	0	RW	Timer2 inc/cap 引脚 map1 的使能控制。 0: 关闭 1: 启动
4	tmr2_inccap_map0	0	RW	Timer2 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
3	tmr1_inccap_map1	0	RW	Timer1 inc/cap 引脚 map1 的使能控制。 0: 关闭 1: 启动
2	tmr1_inccap_map0	0	RW	Timer1 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动
1	tmr0_inccap_map1	0	RW	Timer0 inc/cap 引脚 map1 的使能控制。 0: 关闭 1: 启动
0	tmr0_inccap_map0	0	RW	Timer0 inc/cap 引脚 map0 的使能控制。 0: 关闭 1: 启动

**7-20 PWMKEY**

地址偏移: 0x54

Bit	名称	复位值	读/写	说明
31:24	保留	8'h00	RW	保留。
23:16	spwm_c_en	8'h00	RW	PB8,PB9,PB10,PB11 PWM IO 的键值保护控制。 spwm_c_en = pwm_key[23:16] = 8'h55; adttim_c_en = pwm_key[23:16] = 8'h56; gpio_c_en = pwm_key[23:16] = 8'h57; svpwm_c_en = pwm_key[23:16] = 8'h58;
15:8	spwm_b_en	8'h00	RW	PB4,PB5,PB6,PB7 PWM IO 键值保护控制。 spwm_b_en = pwm_key[15:8] = 8'ha1; adttim_b_en = pwm_key[15:8] = 8'ha2; gpio_b_en = pwm_key[15:8] = 8'ha3; svpwm_b_en = pwm_key[15:8] = 8'ha4;
7:0	spwm_a_en	8'h00	RW	PB0,PB1,PB2,PB3 PWM IO 键值保护控制。 spwm_a_en = pwm_key[7:0] = 8'h51; adttim_a_en = pwm_key[7:0] = 8'h52; gpio_a_en = pwm_key[7:0] = 8'h53;

**7 系统配置寄存器**

				svpwm_a_en = pwm_key[7:0] = 8'h54;
--	--	--	--	------------------------------------

**7-21 SYSKEY**

地址偏移: 0x58

Bit	名称	复位值	读/写	说明
31:0	SYS_KEY	32'h0000	RW	系统相关寄存器写操作的密码保护配置。 sys_wr_en = (sys_key = 32'h3fac87e4); spwm_sft_wr_en = (sys_key = 32'h4c5de9b3);

**7-22 PERIDMAERR0**

地址偏移: 0x5C

Bit	名称	复位值	读/写	说明
31	iir_data_dma_err_ack0_ris	0	RW	IIR DATA DMA0 地址故障状态位。 0: 无故障发生 1: 发生故障
30	fir_config_dma_err_ack_ris	0	RW	FIR config DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
29	iir_config_dma_err_ack_ris	0	RW	IIR config DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
28	dftrans2_imag_dma_err_ack_ris	0	RW	DFTTRANS2 虚部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
27	dftrans2_real_dma_err_ack_ris	0	RW	DFTTRANS2 实部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
26	dftrans1_imag_dma_err_ack_ris	0	RW	DFTTRANS1 虚部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
25	dftrans1_real_dma_err_ack_ris	0	RW	DFTTRANS1 实部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
24	dftrans0_imag_dma_err_ack_ris	0	RW	DFTTRANS0 虚部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
23	dftrans0_real_dma_err_ack_ris	0	RW	DFTTRANS0 实部 DMA 的地址故障状态位。 0: 无故障发生 1: 发生故障
22	arc_triangle_dma2_err_ack_ris	0	RW	ARC TRIANGLE DMA2 的故障状态位。 0: 无故障发生 1: 发生故障
21	arc_triangle_dma1	0	RW	ARC TRIANGLE DMA1 的故障状态位。

**7 系统配置寄存器**

	_err_ack_ris			0: 无故障发生 1: 发生故障
20	arc_triangle_dma0_err_ack_ris	0	RW	ARC TRIANGLE DMA0 的故障状态位。 0: 无故障发生 1: 发生故障
19	fft2_realimag_out_dma_err_ack_ris	0	RW	FFT2 实部虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
18	fft2_window_in_dma_err_ack_ris	0	RW	FFT2 加窗 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
17	fft2_imag_in_dma_err_ack_ris	0	RW	FFT2 虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
16	fft2_real_in_dma_err_ack_ris	0	RW	FFT2 实部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
15	fft1_realimag_out_dma_err_ack_ris	0	RW	FFT1 实部虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
14	fft1_window_in_dma_err_ack_ris	0	RW	FFT1 加窗 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
13	fft1_imag_in_dma_err_ack_ris	0	RW	FFT1 虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
12	fft1_real_in_dma_err_ack_ris	0	RW	FFT1 实部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
11	fft0_realimag_out_dma_err_ack_ris	0	RW	FFT0 实部虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
10	fft0_window_in_dma_err_ack_ris	0	RW	FFT0 加窗 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
9	fft0_imag_in_dma_err_ack_ris	0	RW	FFT0 虚部 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
8	fft0_real_in_dma_err_ack_ris	0	RW	FFT0 实部 DMA 故障状态位。 0: 无故障发生 1: 发生故障
7	matrix_dma2_err	0	RW	MATRIX 乘法 DMA2 的故障状态位。

**7 系统配置寄存器**

	_ack_ris			0: 无故障发生 1: 发生故障
6	matrix_dma1_err_ack_ris	0	RW	MATRIX 乘法 DMA1 的故障状态位。 0: 无故障发生 1: 发生故障
5	matrix_dma0_err_ack_ris	0	RW	MATRIX 乘法 DMA0 的故障状态位。 0: 无故障发生 1: 发生故障
4	crcdma_err_ris	0	RW	CRC DMA 的故障状态位。 0: 无故障发生 1: 发生故障
3	rms1dma_err_ris	0	RW	RMS1 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
2	rms0dma_err_ris	0	RW	RMS0 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
1	sincosdma_err_ris	0	RW	SINCOS DMA 的故障状态位。 0: 无故障发生 1: 发生故障
0	saradcdma_err_ris	0	RW	SARADC DMA 的故障状态位。 0: 无故障发生 1: 发生故障

**7-23 PERIDMAERR1**

地址偏移: 0x60

Bit	名称	复位值	读/写	说明
31:22	保留	0	RW	保留。
21	fadc_dma_req_ris	0	RW	FADC DMA 的故障状态位。 0: 无故障发生 1: 发生故障
20	sys_dma_req_ris	0	RW	System DMA 的故障状态位。 0: 无故障发生 1: 发生故障
19	dcode_dma_req_ris	0	RW	Dcode DMA 的故障状态位。 0: 无故障发生 1: 发生故障
18	icode_dma_req_ris	0	RW	Icode DMA 的故障状态位。 0: 无故障发生 1: 发生故障
17	保留	0	RW	不可修改此值。
16	hcc_ragdma_req_ris	0	RW	HCC RAG DMA 的故障状态位。 0: 无故障发生

**7 系统配置寄存器**

				1: 发生故障
15	hcc_phsdma_req_ris	0	RW	HCC PHS DMA 的故障状态位。 0: 无故障发生 1: 发生故障
14	hcc_fftdma_req_ris	0	RW	HCC FFT DMA 的故障状态位。 0: 无故障发生 1: 发生故障
13	iir_coef2_dma_req_ris	0	RW	IIR 的系数 DMA2 的故障状态位。 0: 无故障发生 1: 发生故障
12	iir_coef1_dma_req_ris	0	RW	IIR 的系数 DMA1 的故障状态位。 0: 无故障发生 1: 发生故障
11	iir_coef0_dma_req_ris	0	RW	IIR 的系数 DMA0 的故障状态位。 0: 无故障发生 1: 发生故障
10	rms2_dma_err_req_ris	0	RW	RMS2 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
9	sincos1dma_err_req_ris	0	RW	SINCOS1 DMA 的故障状态位。 0: 无故障发生 1: 发生故障
8	iir_config2_dma_err_req_ris	0	RW	IIR 的配置 DMA2 的故障状态位。 0: 无故障发生 1: 发生故障
7	iir_config1_dma_err_req_ris	0	RW	IIR 的配置 DMA1 的故障状态位。 0: 无故障发生 1: 发生故障
6	data_dma_dest_err_req_ris	0	RW	DATA DMA DEST 的故障状态位。 0: 无故障发生 1: 发生故障
5	data_dma_src_err_req_ris	0	RW	DATA DMA SRC 的故障状态位。 0: 无故障发生 1: 发生故障
4	fir_data_dma_err_ack2_ris	0	RW	FIR DATA DMA2 的故障状态位。 0: 无故障发生 1: 发生故障
3	fir_data_dma_err_ack1_ris	0	RW	FIR DATA DMA1 的故障状态位。 0: 无故障发生 1: 发生故障
2	fir_data_dma_err_ack0_ris	0	RW	FIR DATA DMA0 的故障状态位。 0: 无故障发生

**7 系统配置寄存器**

				1: 发生故障
1	iir_data_dma_err_ack2_ris	0	RW	IIR DATA DMA2 的故障状态位。 0: 无故障发生 1: 发生故障
0	iir_data_dma_err_ack1_ris	0	RW	IIR DATA DMA1 的故障状态位。 0: 无故障发生 1: 发生故障

**7-24 HOSCMNT**

地址偏移: 0x64

Bit	名称	复位值	读/写	说明
31:16	high_limit	16'h1450	RW	晶振加速范围配置。
15	hosc_mnt_en	0	RW	晶振监控使能控制。 0: 关闭 1: 启动
14	保留	0	RW	保留。
13	hosc_loss_sw_en	1	RW	晶振异常时钟的硬切换使能控制。
12:0	low_limit	13'h00a7	RW	晶振减速的范围配置控制。

**7-25 WAKEUP\_CON0**

地址偏移: 0x68

Bit	名称	复位值	读/写	说明
31:24	wkup_pnd	8'b0	RW	端口唤醒 pending。
23:16	保留	8'b0	RW	保留。
15:8	wkup_edge	8'b0	RW	端口唤醒的边沿选择。
7:0	wkup_en	8'b0	RW	端口唤醒使能控制。

**7-26 LP\_CON0**

地址偏移: 0x6C

Bit	名称	复位值	读/写	说明
31:23	保留	9'b0	RW	保留。
22	sram16_auto_dis	0	RW	SRAM16 自动关闭的控制位。 0: 不关闭 1: 关闭
21	sram15_auto_dis	0	RW	SRAM15 自动关闭的控制位。 0: 不关闭 1: 关闭
20	sram14_auto_dis	0	RW	SRAM14 自动关闭的控制位。 0: 不关闭 1: 关闭
19	sram13_auto_dis	0	RW	SRAM13 自动关闭的控制位。 0: 不关闭 1: 关闭
18	sram12_auto_dis	0	RW	SRAM12 自动关闭的控制位。 0: 不关闭

**7 系统配置寄存器**

				1: 关闭
17	sram11_auto_dis	0	RW	SRAM11 自动关闭的控制位。 0: 不关闭 1: 关闭
16	sram9_auto_dis	0	RW	SRAM9 自动关闭的控制位。 0: 不关闭 1: 关闭
15	sram8_auto_dis	0	RW	SRAM8 自动关闭的控制位。 0: 不关闭 1: 关闭
14	sram7_auto_dis	0	RW	SRAM7 自动关闭的控制位。 0: 不关闭 1: 关闭
13	sram6_auto_dis	0	RW	SRAM6 自动关闭的控制位。 0: 不关闭 1: 关闭
12	sram5_auto_dis	0	RW	SRAM5 自动关闭的控制位。 0: 不关闭 1: 关闭
11	sram4_auto_dis	0	RW	SRAM4 自动关闭的控制位。 0: 不关闭 1: 关闭
10	sram3_auto_dis	0	RW	SRAM3 自动关闭的控制位。 0: 不关闭 1: 关闭
9	sram2_auto_dis	0	RW	SRAM2 自动关闭的控制位。 0: 不关闭 1: 关闭
8	sram1_auto_dis	0	RW	SRAM1 自动关闭的控制位。 0: 不关闭 1: 关闭
7	sram10_auto_dis	0	RW	SRAM10 自动关闭的控制位。 0: 不关闭 1: 关闭
6	sram0_auto_dis	0	RW	SRAM0 自动关闭的控制位。 0: 不关闭 1: 关闭
5	rc32k_auto_dis	0	RW	RC32K 自动关闭的控制位。 0: 不关闭 1: 关闭
4	flash_auto_dis	0	RW	Flash 自动关闭的控制位。 0: 不关闭

**7 系统配置寄存器**

				1: 关闭
3	rom_auto_dis	0	RW	ROM 自动关闭的控制位。 0: 不关闭 1: 关闭
2	保留	0	RW	保留。
1	stopclk	0	RW	时钟停止模式使能控制。 0: 关闭 1: 启动
0	sleep	0	RW	睡眠模式使能控制。 0: 关闭 1: 启动

**7-27 SPWM\_SFTCON**

地址偏移: 0x78

Bit	名称	复位值	读/写	说明
31:1	保留	31'b0	R	保留。
0	spwm_sft_con	1	RW	SPWM 的模块软复位控制。 0: 软复位 1: 释放软复位

## 8 历史

版本号	日期	说明
V1.0	2019/04/18	Official version
V2.0	2020/07/23	<ol style="list-style-type: none"><li>1. 删改模拟、测试部分的寄存器位</li><li>2. 修改 AIPCON0 寄存器部分描述及删除 GMAC 相关内容</li><li>3. 增加 DMAC 模块且修改原 GPDMA 为 DMAC</li><li>4. 增加 Watchdog 简介，新增 3.2 功能描述、3.3 寄存器访问保护、3.4 调试模式</li></ol>