1 TMF5201 处理器概述

1.1 处理器特性

- **内核**
  - 32 位 Cortex-M3 核，频率高达 82MHz
  - 带有可重构配置处理器 RPU（Reconfigurable Processing Unit），内置 2 个 SINCOS，3 个 RMS，1 个 4 通道 MATRIX 乘法，3 个 DFT，3 个 ACRTAN，3 个 IIR 和 1 个 DATA DMA
- **内存**
  - 32KB BROM Bootloader
  - 128KB eFlash 闪存
  - 112KB 系统 SRAM
- **通信外设**
  - 2 个 UART（支持 RS-232 和 RS-485）
  - 1 个 I²C 接口
  - 1 个 2.0B CAN
- **支持多种 PWM 模式**
  - 内置硬件 12 通道 SPWM
  - 内置硬件 12 通道 SVPWM
  - 内置硬件 14 通道 EPWM
- **系统外设**
  - 通用目的 DMA（DMAC）：4-stream DMA，是一个带 FIFO、可以支持突发情况的控制器
  - EVSYS，事件控制单元
- **正交编码器（QEI）：用于电机控制应用**
  - 获取机械位置数据
  - 有 3 个输入通道，分别支持 2 个相位信号和 1 个索引脉冲输入
  - 16 位的递增/递减位置计数器
  - 支持配置计数方向状态
  - 支持位置测量模式（x2 和 x4）
  - 支持可编程数字噪声滤波器
  - 支持交替的 16 位定时器/计数器模式
  - 支持 QEI 中断
- **通用定时器**
  - 共有 6 个 16 位的通用定时器
  - 6 个定时器均支持产生 PWM 波
- **时钟和系统控制**
  - 1 个 26MHZ 的晶体振荡器
  - 32K 的片上晶振
  - 8MHz 的片上晶振
  - 系统和 ADC 的 2 个 frac-PLL
  - 看门狗定时器模块
- **GPIO**
  - 34 个支持中断的 I/O 口，所有 I/O 口均不支持 5V 容忍
- **模拟前端信号处理**
  - 支持双模式可编程增益放大器
  - 2 个内插式低通滤波器
  - 支持 2 通道的模拟比较器
  - 支持 3 通道的 10 位 DAC
  - 含温度传感器
- **工业级 SARADC**
  - 共 14 个采样通道
  - 12 位的 SARADC
  - 最大采样率达到 156KSPS
  - SNDR>62dB
- **工业级 Fast SARADC**
  - 支持同步采样和连续采样两种模式
  - 14 个多路复用输入
  - 支持 16 个转换通道，支持配置为触发和模拟输入
  - 支持 16 位数据输出的高速 SARADC
  - 每个通道最大采样率达到 3.84MSPS
  - SNDR>62dB
- **高可靠性**
  - ESD HBM 8KV
  - EFT ±4KV

Copyright ©2020, www.tai-action.com. All Rights Reserved.

- **电源管理单元（PMU）**
  - 输入电压范围是：3.0V~3.6V
  - 到内核的LDO-1.5V（包括逻辑电路、片上存储器和eFlash）
  - 内置温度传感器组件
  - 共 64 个引脚（RoHS）
- **支持的工作温度范围**
  - -40℃到 125℃（结温）
- **应用领域**
  - 工业级和消费级电机
  - 数字电源

Copyright ©2020, www.tai-action.com. All Rights Reserved.

目录

Copyright ©2020, www.tai-action.com. All Rights Reserved.

Copyright ©2020, www.tai-action.com. All Rights Reserved.

Copyright ©2020, www.tai-action.com. All Rights Reserved.

珠海泰为电子有限公司
Zhuhai Tai-Action Electronics CO., LTD.

2 系统和存储器架构

TMF5201 是一款集成度高、性能卓越的工业实时控制处理器。TMF5201 内置一个通用的 32 位Cortex-M3 核，实现了出色的计算性能和对异常能力的快速响应。单芯片即可满足电力电子相关领域控制系统的应用需求。内置可重构配置单元（RPU）、多通道的高分辨率ADC、预处理模拟信号前端电路并且支持多种PWM模式。

TMF5201 适用于电力电子领域的所有单芯片解决方案，主要应用于电机系统控制器，数字电源系统控制器，如工业及消费级电机以及其他电力电子领域。

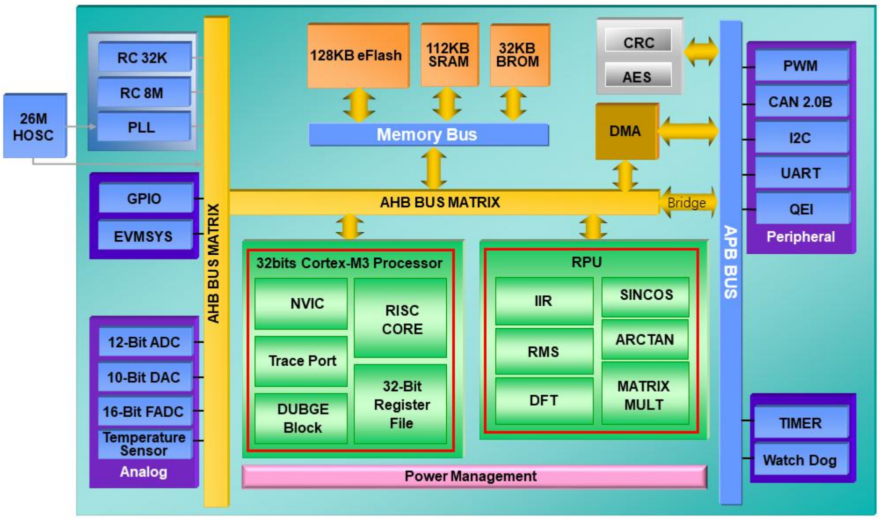


图 2-1 TMF5201 的系统框图

2.1 系统数据通路互联架构

TMF5201 器件采用 32 位多层总线结构，该结构使得系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线和两个 APB 总线。

2.2 内存映射

下方表格 2-1 显示了TMF5201 器件的内存映射，包括代码、SRAM、外设和其他预定义空间。几乎所有的外设都被分配了 4KB的地址空间，这可以简化每个外设的地址译码。

表 2-1 TMF5201 的内存映射表

地址范围	外设或存储器	总线		类型
0x40039000 ---- 0x4003ffff	Reserved	APB2		外设
0x40038000 ---- 0x40038fff	EPWM			
0x40037000 ---- 0x40037fff	FADC			
0x40036000 ---- 0x40036fff	ADC			
0x40035000 ---- 0x40035fff	RPU			
0x40034000 ---- 0x40034fff	Reserved			
0x40033000 ---- 0x40033fff	SVPWM			
0x40032000 ---- 0x40032fff	EVSYS			
0x40031000 ---- 0x40031fff	EFLASH CONTROLLER			
0x40030000 ---- 0x40030fff	Reserved			

Copyright ©2020, www.tai-action.com. All Rights Reserved.

0x40029000 ---- 0x40029fff	CAN_CTRL		AHB	
0x40028000 ---- 0x40028fff	Reserved			
0x40027000 ---- 0x40027fff	GPIO_PD			
0x40026000 ---- 0x40026fff	SYS_CTRL			
0x40025000 ---- 0x40025fff	GPIO_PC			
0x40024000 ---- 0x40024fff	GPIO_PB			
0x40023000 ---- 0x40023fff	Reserved			
0x40022000 ---- 0x40022fff	GPIO_PA			
0x40021000 ---- 0x40021fff	Reserved			
0x40020000 ---- 0x40020fff	DMAC SLAVE			
0x4001e000 ---- 0x4001ffff	Reserved	APB1		
0x4001d000 ---- 0x4001dfff	FRAC PLL1			
0x4001c000 ---- 0x4001cfff	FRAC PLL0			
0x40018000 ---- 0x4001bfff	Reserved			
0x40017000 ---- 0x40017fff	CRC			
0x40016000 ---- 0x40016fff	Reserved			
0x40015030 ---- 0x40015fff	TIMER3			
0x40015020 ---- 0x4001502f	TIMER2			
0x40015010 ---- 0x4001501f	TIMER1			
0x40015000 ---- 0x4001500f	Reserved			
0x40014000 ---- 0x40014fff	WDT1			
0x40013000 ---- 0x40013fff	WDT0			
0x40012000 ---- 0x40012fff	QEI			
0x40010000 ---- 0x40011fff	Reserved			
0x4000d030---- 0x4000ffff	Reserved	APB0		
0x4000d020 ---- 0x4000d02f	TIMER12			
0x4000d010 ---- 0x4000d01f	TIMER11			
0x4000d000 ---- 0x4000d00f	TIMER10			
0x4000b000 ---- 0x4000cfff	Reserved			
0x4000a000 ---- 0x4000afff	ADVTMR0			
0x40009000 ---- 0x40009fff	LVD_CTRL			
0x40006000---- 0x40008fff	Reserved			
0x40005000 ---- 0x40005fff	UART2			
0x40004000 ---- 0x40004fff	Reserved			
0x40003000 ---- 0x40003fff	UART0			

Copyright ©2020, www.tai-action.com. All Rights Reserved.

0x40001000---- 0x40002fff	Reserved			
0x40000000 ---- 0x40000fff	I2C0			
…				
0x2001b800 ---- 0x2001bfff	SRAM16		Memory Bus	SRAM
0x2001b000 ---- 0x2001b7ff	SRAM15			
0x2001a800 ---- 0x2001afff	SRAM14			
0x2001a000 ---- 0x2001a7ff	SRAM13			
0x20019800 ---- 0x20019fff	SRAM12			
0x20019000 ---- 0x200197ff	SRAM11			
0x20018800 ---- 0x20018fff	SRAM9			
0x20018000 ---- 0x200187ff	SRAM8			
0x20017800 ---- 0x20017fff	SRAM7			
0x20017000 ---- 0x200177ff	SRAM6			
0x20016800 ---- 0x20016fff	SRAM5			
0x20016000 ---- 0x200167ff	SRAM4			
0x20014000 ---- 0x20015fff	SRAM3			
0x20012000 ---- 0x20013fff	SRAM2			
0x20010000 ---- 0x20011fff	SRAM1			
0x20008000 ---- 0x2000ffff	SRAM10			
0x20000000 ---- 0x20007fff	SRAM0			
…				
0x10008000 ---- 0x1000ffff	SRAM10			Code
0x10000000 ---- 0x10007fff	SRAM0			
0x08000000 ---- 0x08ffffff	EFLASH			
0x00000000 ---- 0x00007fff	32KB ROM			

2.3 I/O 功能复用配置表

表 2-2 功能复用配置表

模块	Pin	io_map0[4]=1	io_map0[14]=1	
UART0	UART0_rx	PA10	PA10	
	UART0_tx	PA11	PA11	
模块	**Pin**	**io_map0[6]=1**	**io_map0[16]=1**	
UART2	UART2_rx	PD2	PD2	
	UART2_tx	PD3	PD3	
模块	**Pin**	**io_map0[0]=1**	**io_map0[17]=1**	**io_map0[18]=1**
IIC0	IIC0_scl	PA6	PA6	PB8
	IIC0_sda	PA7	PA7	PB9
模块	**Pin**	**io_map0[9]=1**		
CAN	CAN_rxd	PD7		
	CAN_txd	PD6		
	CAN_stdy	PD8		

Copyright ©2020, www.tai-action.com. All Rights Reserved.

模块	Pin	io_map0[11]=1		
JTAG-SWD	SWCLK	PD14		
	SWDIO	PD15		
模块	**Pin**	**io_map0[8]=1**		
QEI	QEI_updn	PB12		
	QEI_a	PB13		
	QEI_b	PB14		
	QEI_idx	PB15		
模块	**Pin**	**io_map1[2]=1**		
Timer1	TMR1_cap	PA1		
模块	**Pin**	**io_map1[4]=1**	**io_map1[13]=1**	
Timer2	TMR2_pwm		PD11	
	TMR2_cap	PA2		
模块	**Pin**	**io_map1[6]=1**	**io_map1[15]=1**	
Timer3	TMR3_pwm		PD12	
	TMR3_cap	PA3		
模块	**Pin**	**io_map1[20]=1**		
Timer10	TMR10_pwm	PD6		
模块	**Pin**	**io_map1[21]=1**		
Timer11	TMR11_pwm	PD7		
模块	**Pin**	**io_map1[22]=1**		
Timer12	TMR12_pwm	PD8		
模块	**Pin**	**io_map0[12]=1**		
SPWM	dcp_up_ovload	PB12		
	dcm_up_ovload	PB13		
	dcp_dn_ovload	PB14		
	dcm_dn_ovload	PB15		
	current_ovload_c	PC2		
	igbt1_fault_	PC4		
	SPWM_a0	PB0		
	SPWM _a1	PB1		
	SPWM _a2	PB2		
	SPWM _a3	PB3		
	SPWM _b0	PB4		
	SPWM _b1	PB5		
	SPWM _b2	PB6		
	SPWM _b3	PB7		
	SPWM _c0	PB8		
	SPWM _c1	PB9		
	SPWM _c2	PB10		
	SPWM _c3	PB11		

Copyright ©2020, www.tai-action.com. All Rights Reserved.

模块	Pin			
EPWM	EPWM0_a	io_map0[25]=1	PB0	
	EPWM0_b	io_map0[26]=1	PB1	
	EPWM1_a	io_map0[27]=1	PB2	
	EPWM1_b	io_map0[28]=1	PB3	
	EPWM2_a	io_map0[29]=1	PB4	
	EPWM2_b	io_map0[30]=1	PB5	
	EPWM3_a	io_map1[24]=1	PB6	
	EPWM3_b	io_map1[25]=1	PB7	
	EPWM4_a	io_map1[26]=1	PB8	
	EPWM4_b	io_map1[27]=1	PB9	
	EPWM5_a	io_map1[28]=1	PB10	
	EPWM5_b	io_map1[29]=1	PB11	
	EPWM6_a	io_map1[30]=1	PB12	
	EPWM6_b	io_map1[31]=1	PB13	
	EPWM0_synco	io_map0[31]=1	PB15	
	EPWM_tz2		PC2	
	EPWM_tz4		PC4	

Copyright ©2020, www.tai-action.com. All Rights Reserved.

2.4 系统主从互联矩阵关系表

NUM		从端口	ROM	eflash	sram0	sram10	sram1	sram2	sram3	sram4	sram5
	模块	主端口									
1	MCU	I-CODE	1	1	1	1					
2		D-CODE	1	1	1	1					
3		SYSBUS	1	1	1	1	1	1	1	1	1
4	DMAC	DMACM1									
5		DMACM2		1	1	1	1	1	1		
6	CRC	CRC_DMA			1	1					
7	ERPU	SINCOS0			1		1	1	1		
8		SINCOS1			1		1	1	1		
9		RMS0			1		1	1	1		
10		RMS1			1		1	1	1		
11		RMS2			1		1	1	1		
12		MATRIX-X					1				
13		MATRIX-Y						1			
14		MATRIX-Z							1		
15		ARC_TRIANGLE-0					1	1	1	1	1
16		ARC_TRIANGLE-1					1	1	1		
17		ARC_TRIANGLE-2					1	1	1		
18		DFTTRANS0_REAL					1				
19		DFTTRANS0_IMAG						1			
20		DFTTRANS1_REAL						1			
21		DFTTRANS1_IMAG							1		
22		DFTTRANS2_REAL							1		
23		DFTTRANS2_IMAG					1				
24		DATADMA_SRC			1	1	1	1	1	1	1
25		DATADMA_DST			1	1	1	1	1	1	1
27		IIR_CONFIG0									
28		IIR_CONFIG1									
29		IIR_CONFIG2									
30		IIR_COEF0									
31		IIR_COEF1									
32		IIR_COEF2									
33		IIR_DATA0					1				
34		IIR_DATA1						1			
35		IIR_DATA2							1		
36	ADC	SARADC_DMA			1	1	1	1	1		

Copyright ©2020, www.tai-action.com. All Rights Reserved.

NUM		从端口	sram6	sram7	sram8	sram9	sram11	sram12	sram13	sram14	sram15	sram16
	模块	主端口										
1		I-CODE										
2	MCU	D-CODE										
3		SYSBUS	1	1	1	1	1	1	1	1	1	1
4	DMAC	DMACM1										
5		DMACM2										
6	CRC	CRC_DMA										
7		SINCOS0										
8		SINCOS1										
9		RMS0										
10		RMS1										
11		RMS2										
12		MATRIX-X										
13		MATRIX-Y										
14		MATRIX-Z										
15		ARC_TRIANGLE-0										
16		ARC_TRIANGLE-1	1	1								
17		ARC_TRIANGLE-2			1	1						
18		DFTTRANS0_REAL										
19		DFTTRANS0_IMAG										
20		DFTTRANS1_REAL										
21	ERPU	DFTTRANS1_IMAG										
22		DFTTRANS2_REAL										
23		DFTTRANS2_IMAG										
24		DATADMA_SRC	1	1	1	1						
25		DATADMA_DST	1	1	1	1						
26		SARADC_DMA										
27		IIR_CONFIG0					1					
28		IIR_CONFIG1						1				
29		IIR_CONFIG2							1			
30		IIR_COEF0								1		
31		IIR_COEF1									1	
32		IIR_COEF2										1
33		IIR_DATA0										
34		IIR_DATA1										
35		IIR_DATA2										
36	ADC	SARADC_DMA										

Copyright ©2020, www.tai-action.com. All Rights Reserved.

2.5 系统配置寄存器

注意：系统配置寄存器基地址为：0x40026000。

名称	偏移量	大小	说明
CLK_CON0	0x00	x25	时钟选择设置。
CLK_CON1	0x04	x24	时钟分频设置。
CLK_CON2	0x08	x16	时钟分频设置。
CLK_CON3	0x0C	x32	GPIO PA-PD 的去抖动时钟分频设置。
CLK_CON4	0x10	x32	模块时钟使能设置。
CLK_CON5	0x14	x32	模块时钟使能设置。
SYS_CON0	0x18	x32	-
SYS_CON1	0x1C	x32	软复位控制。
SYS_CON2	0x20	x32	软复位控制。
SYS_CON3	0x24	x32	GPIO PA/PB 输入去抖使能控制。
SYS_CON4	0x28	x32	GPIO PC/PD 输入去抖使能控制。
SYS_CON5	0x2C	x32	GPIO PA/PB 输入同步使能控制。
SYS_CON6	0x30	x32	GPIO PC/PD 输入的同步使能控制。
SYS_CON7	0x34	x26	模拟功能 GPIO 的数字使能控制。
SYS_CON8	0x38	x32	端口 0-7 的唤醒选择控制。
AIPCON0	0x3C	x32	-
AIPCON1	0x40	x32	-
IO_MAP0	0x44	x32	引脚映射 0/1/2 的使能控制。
IO_MAP1	0x48	x32	引脚映射 0/1 的使能控制。
PMUREG0	0x4C	x32	-
PWMKEY	0x54	x32	PB0-11 PWM IO 键值保护控制。
SYSKEY	0x58	x32	保护系统相关寄存器写操作的密码。
PERIDMAERR0	0x5C	x32	DMA 错误标志位。
PERIDMAERR1	0x60	x32	DMA 错误标志位。
HOSCMNT	0x64	x32	晶振配置控制。
WAKEUP_CON0	0x68	x32	端口唤醒控制。
LP_CON0	0x6C	x32	SRAM 自动关闭控制。
SPWM_SFTCON	0x78	x32	SPWM 软复位控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

3 系统设计

3.1 终端配置与功能

3.1.1 引脚

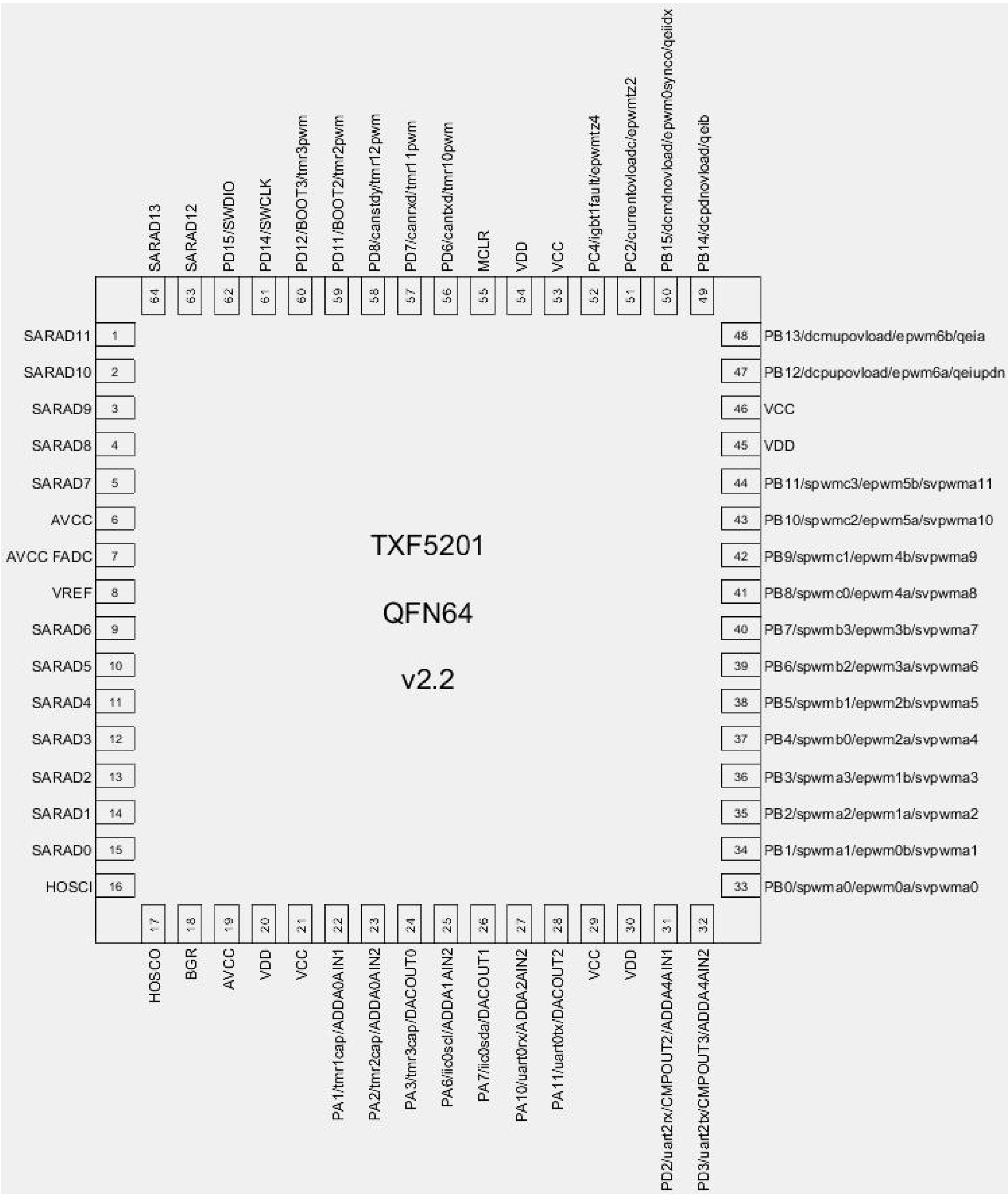


图 **3-1 TMF5201** 引脚示意图

Copyright ©2020, www.tai-action.com. All Rights Reserved.

3.1.2 信号描述

表 3-1 信号描述表

Pin	Func0	Pin 传输方向			说明
		类型	传输方向	电压	
1	SARAD11	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
2	SARAD10	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
3	SARAD9	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
4	SARAD8	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
5	SARAD7	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
6	AVCC	Power	Input	3.3V	模拟电源。
7	AVCC FADC	模拟 IO	Input	3.3V	FARSARADC 的模拟电源，和 AVCC 短接或和 AVCC 一样，使用高性能 LDO 作为输入。
8	VREF	Power	Input	1.5V	参考电压，短接 BGR 或与片外参考源分开使用，注意参考源的选择和输出噪声。
9	SARAD6	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
10	SARAD5	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
11	SARAD4	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
12	SARAD3	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
13	SARAD2	模拟 IO	Output	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
14	SARAD1	模拟 IO	Output	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
15	SARAD0	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
16	HOSCI	模拟 IO	Input	-	26M 晶振输入。
17	HOSCO	模拟 IO	Output	-	26M 晶振输出。
18	BGR	模拟 IO	Output	1.5V	带 2.2u 电容的 ADC 的偏置电压。
19	AVCC	电源	Input	3.3V	模拟电源。
20	VDD	模拟 IO	Output	1.5V	带 2.2u+0.1u 的数字电源。
21	VCC	电源	Input	3.3V	电源输入，使用LDO驱动且容量大于600mA。电容为 2.2u + 0.1u 或选择其他电容作为 LDO

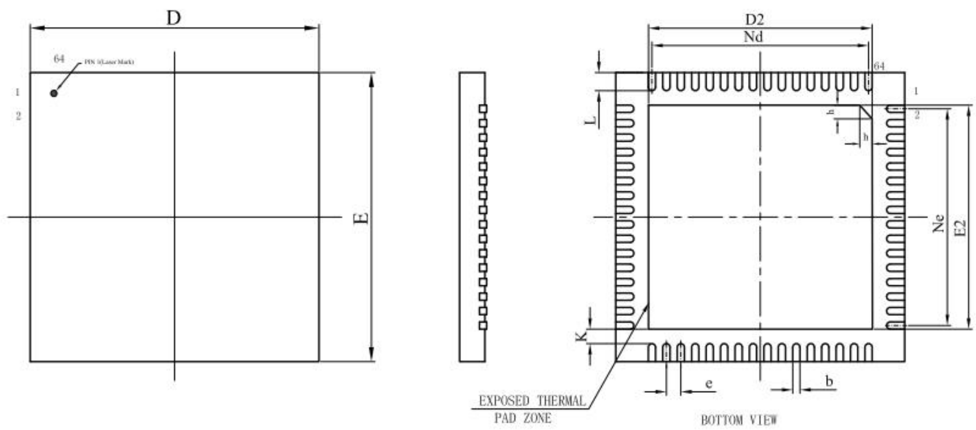
Copyright ©2020, www.tai-action.com. All Rights Reserved.

					功能。
22	PA1	数字 GPIO	Input / Output	3.3V	TMR1_cap; ADDA0_AIN1
23	PA2	数字 GPIO	Input / Output	3.3V	TMR2_cap; ADDA0_AIN2
24	PA3	数字 GPIO	Input / Output	3.3V	TMR3_cap; DACOUT0
25	PA6	数字 GPIO	Input / Output	3.3V	IIC0_scl; ADDA1_AIN2
26	PA7	数字 GPIO	Input / Output	3.3V	IIC0_sda; DACOUT1
27	PA10	数字 GPIO	Input / Output	3.3V	UART0_rx; ADDA2_AIN2
28	PA11	数字 GPIO	Input / Output	3.3V	UART0_tx; DACOUT2
29	VCC	电源	Input	3.3V	电源输入。
30	VDD	模拟 IO	Output	1.5V	数字电源输入。
31	PD2	数字 GPIO	Input / Output	3.3V	UART2_rx; CMPOUT2; ADDA4_AIN1
32	PD3	数字 GPIO	Input / Output	3.3V	UART2_tx; CMPOUT3; ADDA4_AIN2
33	PB0	数字 GPIO	Input / Output	3.3V	SPWM_a0; EPWM0_a; SVPWM_a0
34	PB1	数字 GPIO	Input / Output	3.3V	SPWM_a1; EPWM0_b; SVPWM_a1
35	PB2	数字 GPIO	Input / Output	3.3V	SPWM_a2; EPWM1_a; SVPWM_a2
36	PB3	数字 GPIO	Input / Output	3.3V	SPWM_a3; EPWM1_b; SVPWM_a3
37	PB4	数字 GPIO	Input / Output	3.3V	SPWM_b0; EPWM2_a; SVPWM_a4
38	PB5	数字 GPIO	Input / Output	3.3V	SPWM_b1; EPWM2_b; SVPWM_a5
39	PB6	数字 GPIO	Input / Output	3.3V	SPWM_b2; EPWM3_a; SVPWM_a6
40	PB7	数字 GPIO	Input / Output	3.3V	SPWM_b3; EPWM3_b; SVPWM_a7
41	PB8	数字 GPIO	Input / Output	3.3V	SPWM_c0; EPWM4_a; SVPWM_a8
42	PB9	数字 GPIO	Input / Output	3.3V	SPWM_c1; EPWM4_b; SVPWM_a9
43	PB10	数字 GPIO	Input / Output	3.3V	SPWM_c2; EPWM5_a; SVPWM_a10
44	PB11	数字 GPIO	Input / Output	3.3V	SPWM_c3; EPWM5_b; SVPWM_a11
45	VDD	模拟 IO	Output	1.5V	数字电源输入。
46	VCC	电源	Input	3.3V	电源输入。
47	PB12	数字 GPIO	Input / Output	3.3V	硬件故障信号；EPWM6_a; QEI_updn
48	PB13	数字 GPIO	Input / Output	3.3V	硬件故障信号；EPWM6_b; QEI_a
49	PB14	数字 GPIO	Input / Output	3.3V	硬件故障信号； QEI_b
50	PB15	数字 GPIO	Input / Output	3.3V	硬件故障信号；EPWM0_sync_o; QEI_idx
51	PC2	数字 GPIO	Input / Output	3.3V	硬件故障信号；EPWM_tz2
52	PC4	数字 GPIO	Input / Output	3.3V	硬件故障信号； EPWM_tz4_
53	VCC	电源	Input	3.3V	电源输入。
54	VDD	模拟 IO	Output	1.5V	数字电源输入。
55	MCLR	数字 GPIO	Input / Output	3.3V	MCLR 输入。
56	PD6	数字 GPIO	Input / Output	3.3V	CAN_txd; TMR10_pwm
57	PD7	数字 GPIO	Input / Output	3.3V	CAN_rxd; TMR11_pwm
58	PD8	数字 GPIO	Input / Output	3.3V	CAN_stdy; TMR12_pwm
59	PD11	数字 GPIO	Input / Output	3.3V	Bootloader 启动模式选择 2 ;TMR2_pwm
60	PD12	数字 GPIO	Input / Output	3.3V	Bootloader 启动模式选择 3;TMR3_pwm
61	PD14	数字 GPIO	Input / Output	3.3V	SWCLK

Copyright ©2020, www.tai-action.com. All Rights Reserved.

62	PD15	数字 GPIO	Input / Output	3.3V	SWDIO
63	SARAD12	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。
64	SARAD13	模拟 IO	Input	3.3V	建议使用低噪声、低失真和轨到轨输出 OP 作为缓冲驱动器。

3.1.3 封装信息



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	—	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	7.90	8.00	8.10
D2	6.10	6.20	6.30
e	0.40BSC		
Nd	6.00BSC		
E	7.90	8.00	8.10
E2	6.10	6.20	6.30
Ne	6.00BSC		
L	0.45	0.50	0.55
K	0.20	—	—
h	0.30	0.35	0.40
载体尺寸(mil)	258*258		

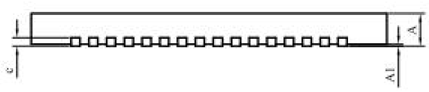


图 3-2 TMF5201 封装图

3.1.4 电学模拟参数

(1) PMU

名称	参数	测试环境	最小值	典型值	最大值	单位
VCCA	模拟电路电源电压	/	/	3.3	/	V
VCC	IO 电源电压		/	3.3	/	V
VDD	VDD LDO 输出电压	负载小于 300m	1.425	1.5	/	V
Ivdd	VDD LDO 最大负	/	/	/	350	mA

Copyright ©2020, www.tai-action.com. All Rights Reserved.

	载					
BGR	参考电压	/	1.493	1.5	1.508	V

(2) 时钟

名称	参数	测试环境	最小值	典型值	最大值	单位
CK32K	内部 32K 时钟	/	30	32	33.92	KHz
RC8M	内部高速时钟	/	/	8	/	MHz
	RC8M 温漂	温度范围：-40~125	-0.5	/	+0.5	%
	RC8M 启动时间	/	/	100	/	us
HXOSC	晶体振荡器振荡频率	/	/	26	/	MHz

(3) PLL1

名称	参数	测试环境	最小值	典型值	最大值	单位
Fpll_in	PLL 输入时钟	/	8	26	32	MHz
Fpll_out	PLL 倍频输出时钟	/	80	/	192	MHz
Tlock	PLL 锁定时间	/	/	100u	600	us

(4) PLL2

名称	参数	测试环境	最小值	典型值	最大值	单位
Fpll_in	PLL 输入时钟	/	8	26	32	MHz
Fpll_out	PLL 倍频输出时钟	/	80	/	200	MHz
Tlock	PLL 锁定时间	/	/	100u	600	us

(5) FSARADC

名称	参数	测试环境	最小值	典型值	最大值	单位
Fs	采样频率	/	/	/	3.69	MHz
Fin	信号频率	SNDR>=62dB	/	/	400	KHz
	满刻度	/	/	/	3	V
	分辨率	/	/	12	/	Bits
INL	积分非线性	/	-2	/	2	LSB
DNL	差分非线性	/	-0.5	/	0.6	LSB
	增益误差	/	-0.27	/	0.27	%FS
	偏移误差	校准后	-6	±1	6	LSB
SNDR	信号-噪声 和失真	Fin<400KHz, Fsmax=3.69MHz	62.2	/	/	dB

(6) SARADC

Copyright ©2020, www.tai-action.com. All Rights Reserved.

名称	参数	测试环境	最小值	典型值	最大值	单位
Fs	采样频率	/	/	/	100	KHz
Fin	信号频率	/	/	/	50	KHz
	满刻度	/	/	/	3	V
	分辨率	/	/	12	/	Bits
INL	积分非线性	/	-1.5	/	1.5	LSB
DNL	差分非线性	/	-1	/	1.5	LSB
SNDR	信号-噪声和失真	Fin=26KHz, Fsmax=100KHz	64.8	/	/	dB
ENOB	/	Fin=26KHz, Fsmax=100KHz	/	/	11	bit

(7) ADDA

a) DA 功能

名称	参数	测试环境	最小值	典型值	最大值	单位
Fclk	时钟频率	/	/	/	2.5	MHz
	满刻度	/	/	/	VCCA	V
	分辨率	/	/	12	/	Bits
INL	积分非线性	/	-4	/	4	LSB
DNL	差分非线性	/	-3.6	/	-3.6	LSB

b) AD 功能

名称	参数	测试环境	最小值	典型值	最大值	单位
Fclk	时钟频率	/	/	/	2.5	MHz
	满刻度	/	/	/	VCCA	V
Fs	采样频率	/	/	/	100	KHz
	分辨率	/	/	12	/	Bits
ENOB	/	Fsmax=1KHz, Fin<100Hz	/	9.5	/	bit

c) CMP 功能

名称	参数	测试环境	最小值	典型值	最大值	单位
Ain	输入幅度	/	0	/	VCCA	V
offset	/	/	-5	/	5	mV

3.2 GPIO

TMF5201 一共有 34 个支持中断的I/O端口，所有I/O口均不支持 5V容忍。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

3.2.1 特性

(1) 支持数字IO输入与输出
(2) 支持GPIO输出的bitset和bit reset
(3) 支持强制数字GPIO功能

3.2.2 GPIO 相关寄存器

注意：GPIO PA 基地址为：0x40022000;
GPIO PB 基地址为：0x40024000;
GPIO PC 基地址为：0x40025000;
GPIO PD 基地址为：0x40027000。

名称	偏移量	大小(x32)	说明
DR	0x0000	1	GPIOx 数据寄存器
DIR	0x0004	1	GPIO 直接寄存器
INTMASK	0x0008	1	GPIO 中断屏蔽寄存器
PU0EN	0x000C	1	GPIO 上拉寄存器，10K
PD0EN	0x0018	1	GPIO 下拉寄存器，10K
DS	0x0024	1	GPIO 驱动增强寄存器
HY	0x0028	1	GPIO 硬件寄存器
OD	0x002C	1	GPIO 端口开漏寄存器
SR	0x0030	1	GPIO 端口慢速寄存器
DIE	0x0034	1	GPIO 数字使能寄存器
BSRS	0x0038	1	GPIO bit set/reset(H16/L16)寄存器

3.3 时钟

TMF5201 时钟源包括 24~26MHz的外部晶振、8M RC时钟和 32K RC时钟。内部晶振受温度影响，综合可以控制在±1%以内。并且，系统时钟和EPWM的模块时钟可以根据不同需求选择相应的时钟源。

3.4 定时器

TMF5201 有 6 个 16 位的通用定时器。所有的定时器均支持 PWM 输出。

3.4.1 特性

(1) 支持计数器和定时器功能
(2) 支持外部 inc 引脚。内部系统时钟、内部 32K RCOSC 和外部高速晶振都可以作用计数器时钟源。
(3) 支持 capture 模式，抓取外部 capture 引脚上的计数值。
(4) 支持独立的 PWM 输出模式。

3.5 EFLASH

3.5.1 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；

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(3) 支持带预取缓冲器的读接口。

(4) E 支持闪存烧写和擦除。

(5) 支持读写保护。

(6) 支持低功耗工作模式。

3.5.2 功能描述

Flash 内存空间是由 32 位宽的存储单元组成，既可以存储数据又可以存储代码。主闪存块分成 32 页，每一页有 4K 字节的大小，并且以页为单位设置写保护。

表 3-2 EFLASH 结构表

模块	名称	地址	大小(byte)
主闪存空间	Page 0	0x0800_0000 – 0x0800_0FFF	4K
	Page 1	0x0800_1000 – 0x0800_1FFF	4K
	Page 2	0x0800_2000 – 0x0800_0BFF	4K
	Page 3	0x0800_3000 – 0x0800_0FFF	4K
	…	…	4K
	Page 30	0x0801_F800 – 0x0801_FBFF	4K
	Page 31	0x0801_FC00 – 0x0801_FFBF	960
	用户配置区	0x0801_FFC0 – 0x0801_FFFF	64
副闪存空间	Sector 0	0x0802_0000 – 0x0802_01FF	512
	Sector 1	0x0802_0200 – 0x0802_03FF	512
	Sector 2	0x0802_0400 – 0x0802_05FF	512
	芯片信息区	0x0802_0600 – 0x0802_06FF	256
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

3.5.2.1 EFLASH 读保护

读操作用于保存指令和数据，可在整个工作电压范围内执行。 一般来说，Flash 运行在 26MHz 的频率下。 如果工作频率增加到 30MHz 以上，Flash 的读取序列需要进行分频。 芯片具有 catch 缓冲区和预取缓冲区以提高 Flash 的访问效率。 如果用户配置完成，则将生成 Flash Protection。

3.5.2.2 EFLASH 的烧写和擦除操作

烧写和擦除操作在整个工作电压范围内都可以执行。 首先，可以根据烧写时钟配置烧写时序；然后配置烧写密码和烧写地址；最后配置烧写数据后开始执行烧写操作并等待完成。

3.5.3 寄存器介绍

注意：eFlash 基地址为：0x08000000。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

名称	偏移量	大小(x32)	说明
CTRLR0	0x00	1	eFlash 配置寄存器。
KST	0x04	1	eFlash 触发控制
DONE	0x08	1	状态寄存器。
PROG_ADDR	0x10	1	烧写地址配置寄存器。
PROG_DATA	0x18	1	烧写数据配置寄存器。
ERASE_CTRL	0x20	1	擦除控制。
TIME_REG0	0x30	1	时序 0 控制寄存器。
TIME_REG1	0x34	1	时序 1 控制寄存器。
NVR_PASSWORD	0x50	1	密码配置寄存器 0.
MAIN_PASSWORD	0x54	1	密码配置寄存器 1.
CRC_ADDR	0x58	1	CRC 地址配置寄存器。
CRC_OUT	0x5C	1	CRC 结果寄存器。

3.6 PMU

PMU，电源管理单元，输入电压范围为 3.0V~3.6V。 该模块的LDO为 1.5V，包括逻辑电路，片上存储和eFlash。 此外，该模块内置BGR和温度传感器。

3.7 CRC

3.7.1 特性

(1) 支持不同长度的多项式，例如 5/6/7/8/16/32 位。
(2) 支持用于自定义多项式。

3.7.2 寄存器

注意：CRC 基地址为 0x40017000。

名称	偏移量	大小(x32)	说明
CRC_CFG	0x00	1	CRC 配置控制。
CRC_INIT	0x04	1	CRC 初始值配置。
CRC_INV	0x08	1	CRC 反转控制。
CRC_POLY	0x0c	1	CRC 多项式配置。
CRC_KST	0x10	1	CRC pending 清除控制。
CRC_STA	0x14	1	CRC pending。
DMA_ADDR	0x1c	1	CRC DMA 首地址配置。
DMA_LEN	0x20	1	CRC DMA 长度配置。
CRC_OUT	0x24	1	CRC 输出结果寄存器。

3.8 Watchdog

此模块是一个可编程看门狗定时器（WDT）外设。

3.8.1 特性

(1) 支持独立时钟源。
(2) 支持配置计数时间。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(3) 支持复位系统。

3.8.2 寄存器介绍

注意：WDT0 基地址为：0x40013000；
WDT1 基地址为：0x40014000。

名称	偏移量	大小(x32)	说明
WDG_KEY	0x04	1	-
WDT_CON	0x00	1	看门狗配置寄存器。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

4 RPU（可重构配置处理器）

4.1 SINCOS

SINCOS 模块用于对输入弧度进行 sine 和 cosine 计算，模块框图如下：

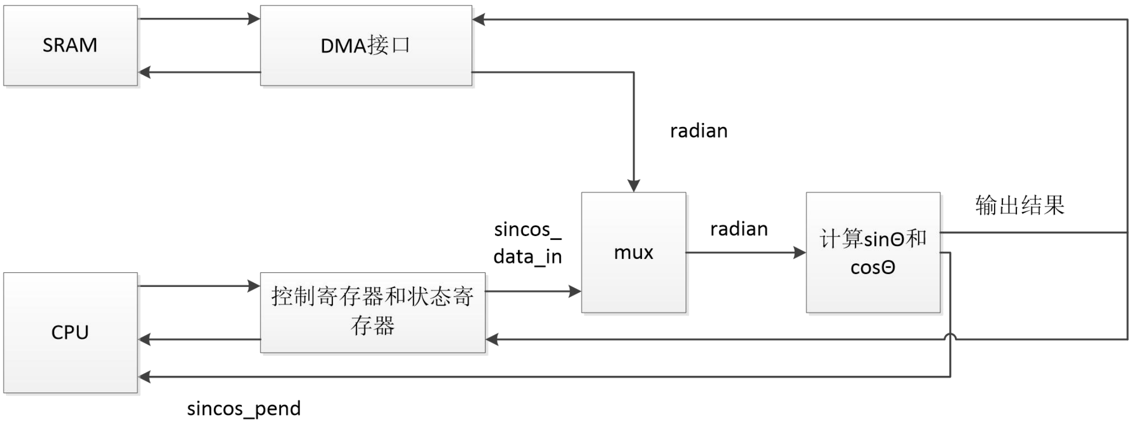


图 4.1 SINCOS 的模块框图

4.1.1 操作步骤

(1) 使能 DMA 模式：配置输出数据存放地址（SINCOS_DATA_OUT_ADR）；如果选择 DMA_MODE，配置弧度的步进寄存器（SINCOS_STEP），如果选择非 DMA_MODE，配置输入数据的存放地址（SINCOS_DATA_IN_ADR）；配置 DMA 的长度（SINCOS_LEN）。

(2) 不使能 DMA 模式，配置寄存器 SINCOS_DATA_IN。

(3) 根据需要配置控制寄存器 SINCOS_CON，使能 SINCOS 模块；

(4) 等待计算结束。可以通过等待 SINCOS_PEND 变为 1 或者 ENABLE 清零的方式实现。

(5) 在非 DMA MODE 即 CPU MODE 时，通过读结果输出寄存器（SINCOS_DATA_OUT）获取结果。当在 DMA MODE 时，结果被存进 SRAM。

(6) 写 1 到 SINCOS_PEND 可清零 SINCOS_PEND。

4.1.2 寄存器

注意：SINCOS0 基地址为：0x40035000；

SINCOS1 基地址为：0x40035040；

名称	偏移量	大小 (x32)	说明
SINCOS_CON	0x00	1	SINCOS 配置寄存器。
SINCOS_LEN	0x04	1	SINCOS 计算完成的次数。
SINCOS_STEP	0x08	1	弧度步进寄存器。
SINCOS_DATA_IN	0x0C	1	CPU 模式下，SINCOS 的弧度配置。
SINCOS_DATA_OUT	0x10	1	CPU 模式下的计算结果。
SINCOS_DATA_IN_ADR	0x14	1	输入弧度的 DMA 首地址。
SINCOS_DATA_OUT_ADRE	0x18	1	输入结果的 DMA 首地址。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

4.2 ARCTAN

该模块输入 cosθ和 sinθ，输出θ，通过坐标旋转算法的方法计算 ARCTAN 的值。一共有两种模式：DMA 模式，通过 DMA 输入 memory 中的数据，再将输出数据存进 memory 中；CPU 模式，输入数据来自 SFR 并将输出数据存进 SFR。不论是输入数据还是输出数据都是 16 位，且一位为符号位，15 位为小数位。ARCTAN 的模块框图如下：

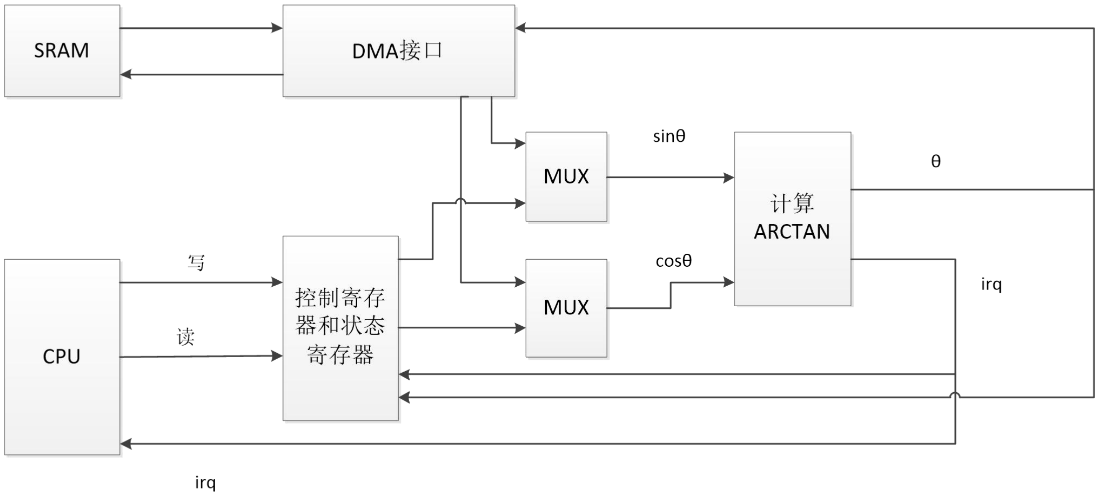


图 4-2 ARCTAN 的模块框图

4.2.1 操作步骤

(1) 在 DMA 模式使能（ARCTAN_CON[2] = 1）下：
- 配置 cosine 输入数据或者 cosine+sine 输入数据的起始地址：ARCTAN_DMA0_STADR；如果 sine 和 cosine 输入数据分开保存，配置 sine 输入数据的起始地址：ARCTAN_DMA1_STADR；
- 配置输出结果的起始地址：ARCTAN_DMA2_STADR；
- 配置 DMA 的长度：ARCTAN_DMA_LEN；

(2) 在非 DMA 模式即 CPU 模式（ARCTAN_CON[2] = 0）下，配置输入数据寄存器：ARCTAN_IN。

(3) 配置控制寄存器 ARCTAN_CON，并使能 ARCTAN 模块（ARCTAN_CON[0] = 1）。

(4) 等待运算结束：等待 PEND（ARCTAN_CON[8]）变为 1 或者 ENABLE（ARCTAN_CON[0]）变为 0。

(5) 如果在 CPU 模式（ARCTAN_CON[2] = 0）下，通过读取输出寄存器获取输出结果：ARCTAN_OUT；在 DMA 模式下，则结果保存在 SRAM。

(6) 清零 PEND：通过写 1 到 PEND（ARCTAN_CON[8]）清零 PEND.

4.2.2 寄存器

注意：ARCTAN0 基地址：0x40035440；
ARCTAN1 基地址：0x40035480；
ARCTAN2 基地址：0x400354C0。

名称	偏移量	大小 (x32)	说明
ARCTAN_CON	0x00	1	ARCTAN 配置寄存器。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

ARCTAN_DMA_LEN	0x04	1	在 DMA 模式下，表示 ARCTAN 的计算次数。
ARCTAN_DMA0_STADR	0x08	1	输入数据 memory buffer 的首地址 0。
ARCTAN_DMA1_STADR	0x0c	1	输入数据 memory buffer 的首地址 1。
ARCTAN_DMA2_STADR	0x10	1	输出数据 memory buffer 首地址。
ARCTAN_IN	0x14	1	在 CPU 模式下，配置输入的 sin θ 和 cos θ 的值。
ARCTAN_OUT	0x18	1	CPU 模式下，存放 ARCTAN θ 的值。

4.3 RMS

RMS 模块框图如下：

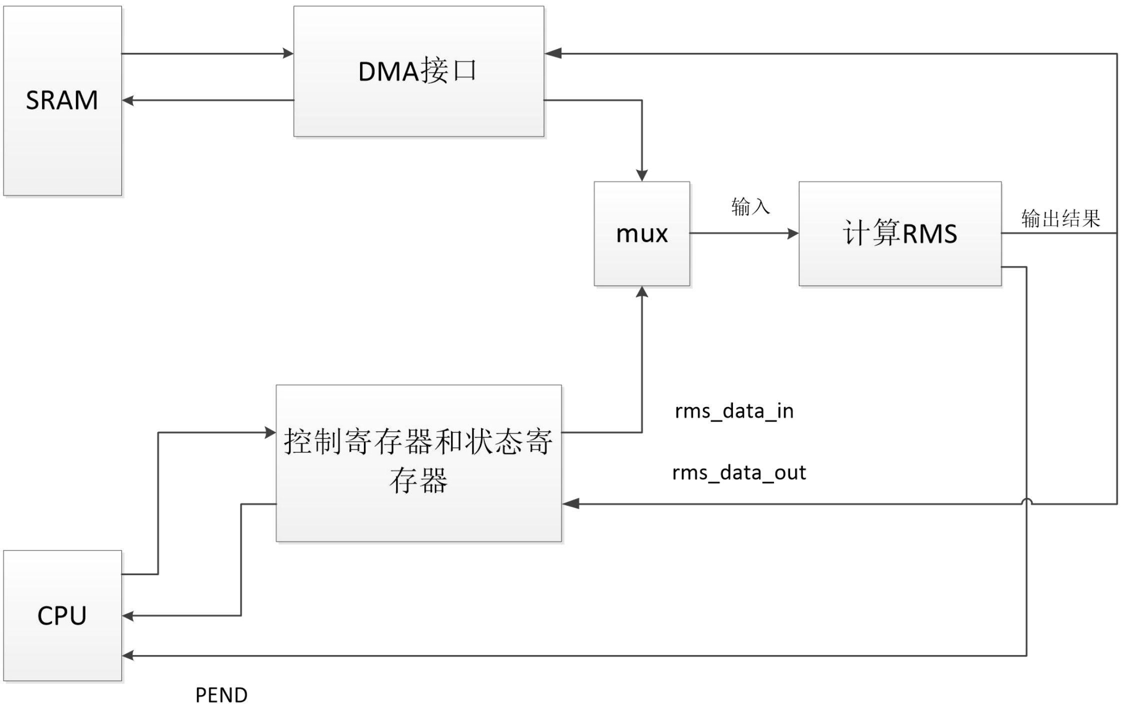


图 4-3 RMS 模块框图

4.3.1 操作步骤

(1) 如果RMS_MODE位(RMS_CON[3:1])等于011,配置输入数据寄存器(RMS_DATA_IN)和输入数据小数位宽寄存器（RMS_IN_FRAC_WIDTH)。

(2) 如果 RMS_MODE（RMS_CON[3:1]）不等于 011，配置输入数据保存地址 RMS_DATA_IN_ADR，输出数据保存地址 RMS_DATA_OUT_ADR，以及 DMA 的长度寄存器（RMS_LEN)。

(3) 配置寄存器 RMS_CON: 根据需要配置是否中断使能和 RMS 的模式，使能 RMS 模块（RMS_CON[0] = 1)。

(4) 等待 RMS 计算完成。可以通过等待 PEND（RMS_CON[8]）置 1 或 ENABLE（RMS_CON[0]）清零实现。

(5) 如果 RMS_MODE（RMS_CON[3:1]）等于 011，通过读取输出寄存器 RMS_DATA_OUTL 和 RMS_DATA_OUTH 获取输出结果。如果 RMS_MODE（RMS_CON[3:1]）不等于 011，所有结果都会被存进 SRAM。

(6) 写 1 到 PEND（RMS_CON[8]）清零 PEND。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

4.3.2 寄存器

注意：RMS0 基地址为：0x40035080；

RMS1 基地址为：0x400350C0；

RMS2 基地址为：0x40035100。

名称	偏移量	大小 (x32)	说明
RMS_CON	0x00	1	RMA 配置寄存器。
RMS_LEN	0x04	1	DMA 长度配置。
RMS_IN_FRAC_WIDTH	0x08	1	RMS0_DATA_IN 数据的小数位宽设置。
RMS_OUT_FRAC_WIDTH	0x0C	1	RMS0_DATA_OUT 数据的小数位宽设置。
RMS_DATA_IN	0x10	1	仅当 RMS_MODE = 011 有效，设置开根号的数据。
RMS_DATA_OUTL	0x14	1	RMS 输出数据的低 32 位。
RMS_DATA_OUTH	0x18	1	RMS 输出数据的高 32 位。
RMS_DATA_IN_ADR	0x1C	1	RMS 输入数据的起始地址，需要 64 位对齐。
RMS_DATA_OUT_ADR	0x20	1	RMS 输出数据的起始地址，需要 64 位对齐。

4.4 MATRIX 乘法

该模块用于计算 3x3 和 3x1 的矩阵乘法，其中，3x3 矩阵来自 9 个 SFR，3x1 矩阵来自 3 个 DMA。该模块一共有 4 个独立的通道。模块会不停地检测 MATRIX_EN 寄存器，当某一位置为 1，相应的通道会立即运行；并且当超过一位被置为 1 时，通道将按以下顺序运行：通道 0->通道 1->通道 2->通道 3。MATRIX Multiply 的模块框图如下：

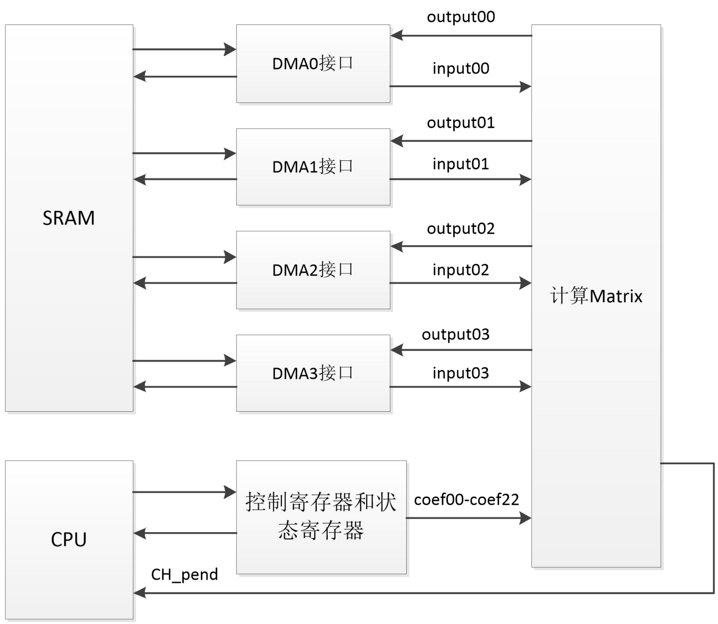


图 4-4 MATRIX Mutiply 的模块框图

Copyright ©2020, www.tai-action.com. All Rights Reserved.

4.4.1 操作步骤

(1) 配置系数寄存器：MATRIX_COEF0_XX 和 MATRIX_COEF1_XX，共 18 个寄存器。

(2) 配置系数选择寄存器：MATRIX_COEF_SEL。

(3) 配置使用到的通道的 buffer 长度寄存器：MATRIX_CH0/1/2/3_LEN。

(4) 配置使用到的通道的 buffer 索引寄存器：MATRIX_CH0/1/2/3_INDEX。

(5) 配置使用到的通道的系数的小数位宽寄存器：MATRIX_CH0/1/2/3_COEF_FRAC_WIDTH。

(6) 配置使用到的通道的输出结果的小数位宽寄存器：MATRIX_CH0/1/2/3_OUT_FRAC_WIDTH。

(7) 配置使用到的通道的输入数据 buffer 的起始地址：MATRIX_CH0/1/2/3_DATAIN_STADR0，MATRIX_CH0/1/2/3_DATAIN_STADR1，MATRIX_CH0/1/2/3_DATAIN_STADR2。

(8) 配置使用到的通道的输出数据 buffer 的起始地址：MATRIX_CH0/1/2/3_DATAOUT_STADR0，MATRIX_CH0/1/2/3_DATAOUT_STADR1，MATRIX_CH0/1/2/3_DATAOUT_STADR2。

(9) 如果需要中断，配置中断寄存器：MATRIX_IE。

(10) 使能需要使用的通道，配置 MATRIX_EN 寄存器。

(11) 等待运算结束：通过等待 MATRIX_EN 寄存器相应的位变 0，或者等待 MATRIX_PEND 相应的位变 1 实现。

(12) 清零 PEND 寄存器：写 1 到相应的位清零相应位的 PEND。

4.4.2 寄存器

注意：MATRIX 乘法的基地址为：0x40035140。

名称	偏移量	大小 (x32)	说明
MATRIX_EN	0xEC	1	通道 0~3 的使能控制。
MATRIX_PEND	0XF0	1	通道 0~3 的计算结束标志。
MATRIX_IE	0Xf4	1	通道 0~3 的结束中断使能控制。
MATRIX_COEF_SEL	0XE8	1	通道 0~3 的系数选择。
MATRIX_COEF0_00	0xa0	1	3x3 矩阵成员[0,0]。
MATRIX_COEF0_01	0xa4	1	3x3 矩阵成员[0,1]。
MATRIX_COEF0_02	0xa8	1	3x3 矩阵成员[0,2]。
MATRX_COEF0_10	0xaC	1	3x3 矩阵成员[1,0]。
MATRIX_COEF0_11	0xb0	1	3x3 矩阵成员[1,1]。
MATRIX_COEF0_12	0xb4	1	3x3 矩阵成员[1,2]。
MATRIX_COEF0_20	0xb8	1	3x3 矩阵成员[2,0]。
MATRIX_COEF0_21	0xbC	1	3x3 矩阵成员[2,1]。
MATRIX_COEF0_22	0xC0	1	3x3 矩阵成员[2,2]。
MATRIX_COEF1_00	0xC4	1	3x3 矩阵成员[0,0]。
MATRIX_COEF1_01	0xC8	1	3x3 矩阵成员[0,1]。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

MATRIX_COEF1_02	0xCC	1	3x3 矩阵成员[0,2]。
MATRIX_COEF1_10	0xD0	1	3x3 矩阵成员[1,0]。
MATRIX_COEF1_11	0xD4	1	3x3 矩阵成员[1,1]。
MATRIX_COEF1_12	0xD8	1	3x3 矩阵成员[1,2]。
MATRIX_COEF1_20	0xDC	1	3x3 矩阵成员[2,0]。
MATRIX_COEF1_21	0xE0	1	3x3 矩阵成员[2,1]。
MATRIX_COEF1_22	0xE4	1	3x3 矩阵成员[2,2]。
MATRIX_CH0/1/2/3_DATAIN_STADR0	0x00/0x28/0x50/0x78	1	3x1 输入矩阵的成员[0,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_DATAIN_STADR1	0x40/0x2C/0x54/0x7C	1	3x1 输入矩阵的成员[1,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_DATAIN_STADR2	0x80/0x30/0x58/0x80	1	3x1 输入矩阵的成员[2,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_DATAOUT_STADR0	0xC0/0x34/0x5C/0x84	1	3x1 输出矩阵的成员[0,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_DATAOUT_STADR1	0x10/0x38/0x60/0x88	1	3x1 输出矩阵的成员[1,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_DATAOUT_STADR2	0x14/0x3C/0x64/0x8C	1	3x1 输出矩阵的成员[2,0]的 memory buffer 起始地址。
MATRIX_CH0/1/2/3_LEN	0x18/0x40/0x68/0x90	1	通道 0/1/2/3 输入输出 buffer 的长度。
MATRIX_CH0/1/2/3_INDEX	0x1C/0x44/0x6C/0x94	1	通道 0/1/2/3 输入输出 buffer 的索引。
MATRIX_CH0/1/2/3_COEF_FRAC_WIDTH	0x20/0x48/0x70/0x98	1	通道 0/1/2/3 的系数的小数位宽。
MATRIX_CH0/1/2/3_OUT_FRAC_WIDTH	0x24/0x4C/0x74/0x9C	1	通道 0/1/2/3 的输出矩阵成员的小数位宽。

4.5 DFTRANS

DFT，离散傅里叶变换，用于将时域信号转变为频域信号。该模块的功能是实现 N 个复数乘加的平均值。其中 N 可以是 17 到 1024 的任意值。其中一个复数通过 DMA 获取的，这个复数的实部和虚部是分开存放的；另一个复数是 cos θ ±j*sin θ， 其中 θ =i*DFTRANS_step，i 的取值范围是 0~N。以上复数的实部和虚部都是 16 位数据，其中 1 位符号位，15 位小数位。输出的结果的实部和虚部也都是 16 位数据，其中 1 位符号位，15 位小数位。

DFT 的定义如下：

$$X(e^{jw}) = \frac{1}{N}\sum_{n=0}^{N-1} X[n]e^{-jwn}$$

复数 1：X[n] = a+j*b

a 通过 real_dma 通路到 sram 读取数据；

Copyright ©2020, www.tai-action.com. All Rights Reserved.

b 通过 imag_dma 通路到 sram 读取数据；

复数 2：$e^{-jwn} = \cos(-jwn)+ \sin(-jwn)$

定义 θ[n] = j｜w｜n，公式为：

e-jwn = cos θ [n] – j*sin θ [n] (w>=0)

or

e-jwn = cos θ [n] +j*sin θ [n] (w<0)

θ [0] = 0；

θ [1] = DFTRANS_STEP；

θ [2] = 2 * DFTRANS_STEP；

...

θ [n] = (n) * DFTRANS_STEP；

DFT 的模块框图如下所示：

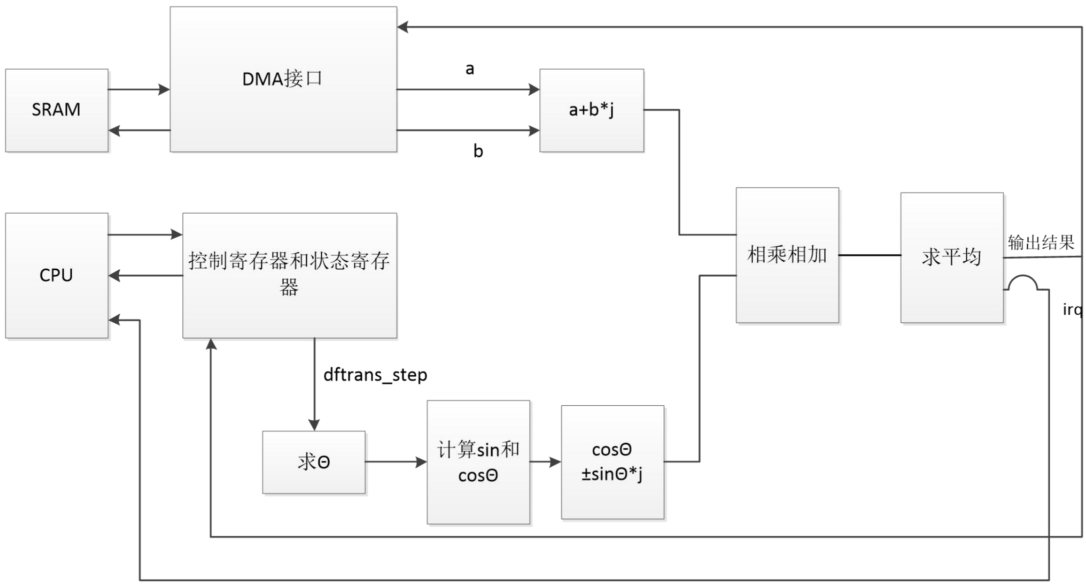


图 4-5 DFT 模块框图

4.5.1 操作步骤

(1) 配置实部输入数据的循环 buffer 的首地址：DFTRANS_REAL_STADR;
(2) 如果需要虚部数据，配置虚部输入数据的循环 buffer 的首地址：DFTRANS_IMAG_STADR；
(3) 配置输入数据的循环 buffer 的长度：DFTRANS_LEN;
(4) 配置输入数据的循环 buffer 的索引：DFTRANS_INDEX;
(5) 配置 COS 和 SIN 的弧度步进：DFTRANS_STEP；
(6) 配置归一化系数：DFTRANS_NORMALIZED_COEF；
(7) 配置结果保存到 sram 中的地址：DFTRANS_OUT_ADDR;
(8) 配置控制寄存器 DFTRANS_CON；使能 DFTRANS 模块：DFTRANS_CON[0]配置为 1；
(9) 等待 PEND（DFTRANS_CON[8]）变 1 或者等待 ENABLE（DFTRANS_CON[0]）变 0，通过读输出结果寄存器（DFTRANS_OUT）读取结果。同时结果也会保存到 SRAM 中的 DFTRANS_OUT_ADDR 地址中。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(10) 通过写 1 到 PEND（DFTRANS_CON[8]）清掉 PEND。

4.5.2 寄存器

注意：DFTTRANS0 基地址：0x40035380；
DFTTRANS1 基地址：0x400353C0；
DFTTRANS2 基地址：0x40035400。

名称	偏移量	大小 (x32)	说明
DFTRANS_CON	0x00	1	DFT 配置寄存器。
DFTRANS_LEN	0x08	1	DFTRANS 虚部、实部数据循环 buffer 的长度。
DFTRANS_INDEX	0x0C	1	DFTRANS 虚部、实部数据循环 buffer 的索引。
DFTRANS_STEP	0xc0	1	SIN、COS 弧度步进，单位为 π 。
DFTRANS_REAL_STADR	0x10	1	复数实部的 DMA 首地址，需要 16 位对齐。
DFTRANS_IMAG_STADR	0x14	1	复数虚部的 DMA 首地址，需要 16 位对齐。
DFTRANS_OUT	0x18	1	输出结果的虚部和实部，1 位符号位。15 位数字位。
DFTRANS_DMA_LEN	0x1C	1	DFTRANS 采样个数，数值为 17~1024。
DFTRANS_NORMALIZED_COEF	0x20	1	归一化系数，计算公式为：(1 /DFTRANS_DMA_LEN) * 2^19
DFTRANS_OUT_ADR	0x24	1	计算结果存储到 SRAM 的 DMA 首地址。

4.6 IIR

IIR 的模块框图如下：

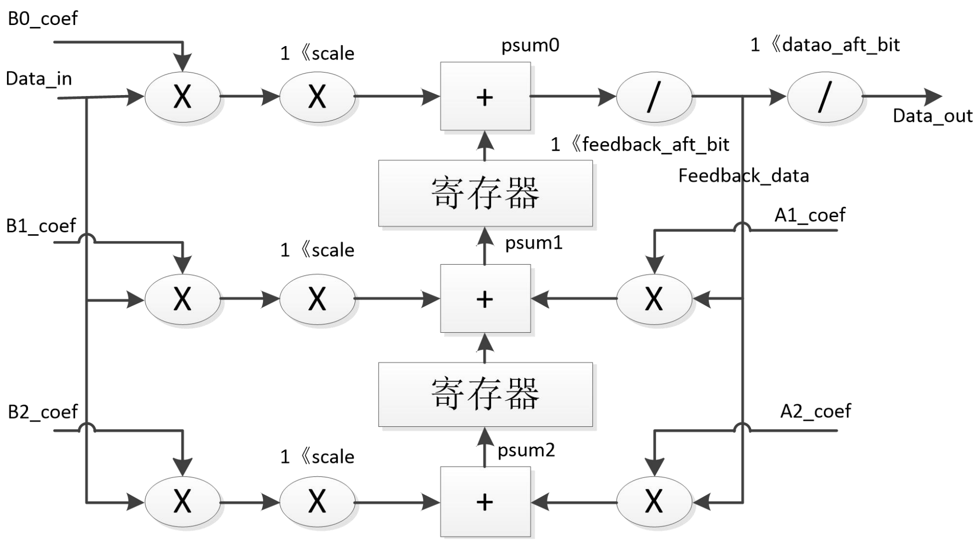


图 4-6 IIR 模块框图

4.6.1 操作步骤

(1) 确定 IIR 的系数信息配置。

(2) 确定 IIR 的配置信息，见 IIR 结构框图所示，注意左边的输入数据、系数和 scale 的总小数位宽，与右边的 feedback、系数的总小数位宽相匹配。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(3) 总共有 3 个 IIR，每个 IIR 都有一组寄存器，通过寄存器 IIR_CPU_KST0/1 来触发相应的所需要的通道；根据 IIR_FILT_PND0/1 来等待滤波结果，或者设置中断（FILT_PND 滤波中断、HALF_PND 半中断、FULL_PND 全中断）。每一个 IIR 都有独立的中断源。

(4) 若不通过软件执行触发，可以通过事件触发器（EVSYS）来实现自动触发。

4.6.2 寄存器

注意：IIR0 基地址：0x40036300；
IIR1 基地址：0x40036360；
IIR2 基地址：0x400363C0。

名称	偏移量	大小(x32)	说明
IIR_CH_ENA0	0x00	1	IIR 通道 0~31 的使能控制。
IIR_CPU_KST0	0x04	1	IIR 通道 0~31 的触发控制。
IIR_CFG_ADDR	0x08	1	存放 IIR 配置信息的首地址。
IIR_INT_ENA0	0x0C	1	IIR 通道 0~31 的中断使能控制。
IIR_HALF_PND0	0x10	1	IIR 通道 0~31 半中断 pending 标志。
IIR_FULL_PND0	0x14	1	IIR 通道 0~31 全中断 pending 标志。
IIR_FILT_PND0	0x18	1	IIR 通道 0~31 的滤波完成标志。
IIR_INT_SRCL0	0x1C	1	IIR 通道 3~31 的中断来源选择位的位 0。
IIR_INT_SRCH0	0x20	1	IIR 通道 3~31 的中断来源选择位的位 1。
IIR_DATA_OUT	0x24	1	IIR 的滤波结果。
EVSYS_CH_ENA0	0x28	1	IIR evsys 通道 0~31 的使能控制。
IIR_CH_ENA1	0x30	1	IIR 通道 32~63 的使能控制。
IIR_CPU_KST1	0x34	1	IIR 通道 32~63 的触发控制。
IIR_INT_ENA1	0x3C	1	IIR 通道 32~63 的中断使能控制。
IIR_HALF_PND1	0x40	1	IIR 通道 32~63 的半中断 pending 标志。
IIR_FULL_PND1	0x44	1	IIR 通道 32~63 的全中断 pending 标志。
IIR_FILT_PND1	0x48	1	IIR 通道 32~63 的滤波完成标志。
IIR_INT_SRCL1	0x4C	1	IIR 通道 32~63 的中断来源选择位位 0。
IIR_INT_SRCH1	0x50	1	IIR 通道 32~63 的中断来源选择位位 1。
EVSYS_CH_ENA1	0x58	1	IIR evsys 通道使能控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

4.7 DATA DMA

DATA DMA 的模块框图如下：

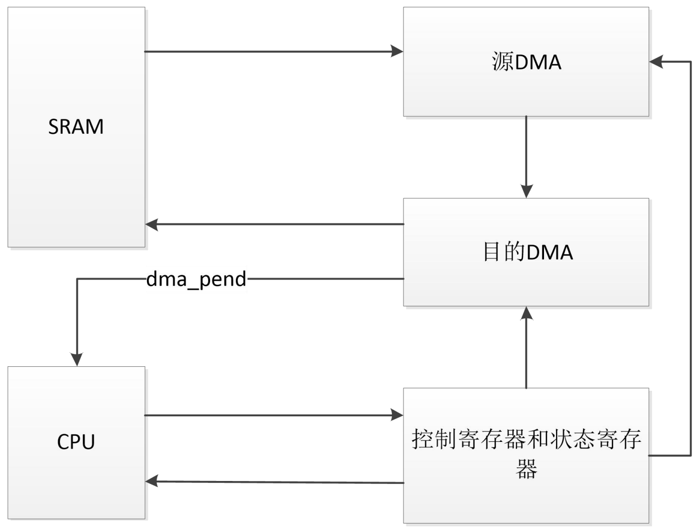


图 4-7 DATA DMA 模块框图

4.7.1 操作步骤

(1) 如果选择DMA_MODE0（DATA_DMA_CON[1] = 0），配置源数据循环buffer的长度和首地址：DATA_DMA_SRC_BUF_LEN，DATA_DMA_SRC_BUF_STADR。
(2) 配置DMA源起始地址：DATA_DMA_SRC_DMA_ADR。
(3) 配置DMA目的起始地址：DATA_DMA_DEST_DMA_ADR。
(4) 配置DMA的长度：DATA_DMA_DMA_LEN。
(5) 配置控制寄存器（DATA_DMA_CON）；使能DMA模块：将DATA_DMA_CON[0]置1。
(6) 等待结束：可以等待PEND（DATA_DMA_CON[8]）变1或者ENABLE（DATA_DMA_CON[0]）变0。
(7) 清零PEND,通过写1到PEND（DATA_DMA_CON[8]）位。

4.7.2 寄存器

注意：DATA DMA 基地址：0x40035500。

名称	偏移量	大小(x32)	说明
DATA_DMA_CON	0x00	1	DMA 配置寄存器。
DATA_DMA_SRC_BUF_STADR	0x04	1	DMA 源数据 buffer 首地址。
DATA_DMA_SRC_BUF_LEN	0x08	1	DMA 源数据 buffer 长度。
DATA_DMA_SRC_DMA_ADR	0x0C	1	DMA 源数据 DMA 首地址。
DATA_DMA_DEST_DMA_ADR	0x10	1	DMA 目的数据 DMA 首地址。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

DATA_DMA_DMA_LEN	0x14	1	DMA 长度，单位为 byte。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

5 通信外设

TMF5201 包含如下通信接口模块： IIC、UART、CAN 和 EVSYS。

5.1 IIC

5.1.1 特性

(1) 两线 IIC 串行接口——有一条串行数据线（SDA）和一条串行时钟线（SCL）组成。
(2) 三种传输速度：
  - 标准模式（0 到 100Kb/s）；
  - 快速模式（≤ 400 Kb/s）或快速模式 plus（≤ 1000Kb/s）；
  - 高速模式（≤ 3.4Mb/s）；

(3) 时钟同步。
(4) 支持主从 IIC 操作。
(5) 支持 7 位或 10 位处理。
(6) 支持 7 位或 10 位组合格式传输。
(7) Bulk 传输模式。
(8) 忽略 CBUS 地址（是 IIC 的前身，常常与 IIC 共用 IIC 总线）。
(9) 传输或接收 buffer。
(10) 支持中断或轮询模式
(11) 在所有总线速度下处理位和字节。
(12) 和 APB 外设相同的简单软件接口。
(13) 配置软件驱动支持的元件参数。
(14) DMA 握手接口和 DMAC 握手接口兼容。
(15) 可编程数据线 SDA 保持时间。
(16) 支持总线清除。
(17) 支持器件 ID。
(18) 支持超快速模式。

5.1.2 寄存器

注意：IIC0 基地址为：0x40000000；

IIC1 基地址为：0x40001000。

名称	偏移量	大小(x32)	说明
CON	0x00	1	IIC 配置控制寄存器。
TAR	0x04	1	IIC 主机目的地址设置寄存器。
SAR	0x08	1	IIC 从机地址保存寄存器。
DATA_CMD	0x10	1	IIC 读写控制设置。
SS_SCL_HCNT	0x14	1	标准模式，SCL 高周期计数值配置寄存器。
SS_SCL_LCNT	0x18	1	标准模式，SCL 低周期计数值配置寄存器。
FS_SCL_HCNT	0x1C	1	快速模式/快速模式 plus，SCL 高周期计数值配置寄存器。
FS_SCL_LCNT	0x20	1	快速模式/快速模式 plus，SCL 低周期计数值配

Copyright ©2020, www.tai-action.com. All Rights Reserved.

			置寄存器。
HS_SCL_HCNT	0x24	1	高速模式，SCL 高周期计数值配置寄存器。
HS_SCL_LCNT	0x28	1	高速模式，SCL 低周期计数值配置寄存器。
INTR_STAT	0x2C	1	IIC 中断状态寄存器（受控于 INTR_MASK，未反应真实状态）。
INTR_MASK	0x30	1	IIC 中断屏蔽控制。
RAW_INTR_STAT	0x34	1	IIC 中断状态寄存器（反应真实状态）。
RX_TL	0x38	1	IIC 接收 FIFO 阈值设置。
TX_TL	0x3C	1	IIC 发送 FIFO 阈值设置。
CLR_RX_UNDER	0x44	1	对该寄存器执行读操作将清除 RX_UNDER 中断。
CLR_RX_OVER	0x48	1	对该寄存器执行读操作将清除 RX_OVER 中断。
CLR_TX_ OVER	0x4C	1	对该寄存器执行读操作将清除 TX_OVER 中断。
CLR_RD_REQ	0x50	1	对该寄存器执行读操作将清除 RD_REQ 中断。
CLR_TX_ABRT	0x54	1	对该寄存器执行读操作将清除 TX_ABRT 中断。
CLR_RX_DONE	0x58	1	对该寄存器执行读操作将清除 RX_DONE 中断。
CLR_ACTIVITY	0x5C	1	对该寄存器执行读操作将清除 ACTIVITY 中断。
CLR_STOP_DET	0x60	1	对该寄存器执行读操作将清除 STOP_DET 中断。
CLR_START_DET	0x64	1	对该寄存器执行读操作将清除 START_DET 中断。
CLR_GEN_CALL	0x68	1	对该寄存器执行读操作将清除 GEN_CALL 中断。
ENABLE	0x6C	1	IIC 使能控制及其他控制选择。
STATUS	0x70	1	IIC 状态寄存器。
DMA_CR	0x88	1	DMA 发送/接收使能控制。
DMA_TDLR	0x8C	1	发送逻辑的 DMA 响应的级别控制。
DMA_RDLR	0x90	1	接收逻辑的 DMA 响应的级别控制。
CLR_SCL_STUCK_DET	0xA8	1	对该寄存器执行读操作将清除 SCL_STUCK_DET 中断。
CLR_RESTART_DET	0xB4	1	对该寄存器执行读操作将清除 RESTART_DET 中断。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

5.2 UART

5.2.1 特性

(1) 支持 9 位串行数据传输。
(2) 错误启动位检测。
(3) 支持可编程小数波特率
(4) 支持多点 RS485 接口。
(5) 基于 16550 行业标准的功能。
(6) 支持可编程 FIFO 启动/关闭。
(7) 由下列公式可计算可编程串行数据波特率：波特率 = (串行时钟频率)/(16×分频数)。

5.2.2 寄存器

注意：UART0 基地址为：0x40003000；

UART1 基地址为：0x40004000；

UART2 基地址为：0x40005000。

名称	偏移量	大小(x32)	说明
RBR	0x00	1	UART 接收 buffer 寄存器。
THR	0x00	1	配置 UART 发送将发送的数据。
DLL	0x00	1	UART 计算波特率的分频数低 8 位。
DLH	0x04	1	UART 计算波特率的分频数高 8 位。
IER	0x04	1	UART 中断使能控制。
IIR	0x08	1	UART 最高优先级中断挂起选择和 FIFO 使能控制。
FCR	0x08	1	UART FIFO 控制寄存器。
LCR	0x0C	1	UART 线路控制寄存器。
MCR	0x10	1	UART 调制解调器控制寄存器。
LSR	0x14	1	UART 线路状态寄存器。
MSR	0x18	1	UART 调制解调器状态寄存器。
USR	0x7C	1	UART 接收/发送 FIFO 状态标志。
TFL	0x80	1	UART 发送 FIFO 中的数据数目。
RFL	0x84	1	UART 接收 FIFO 的数据数目。
TCR	0xAC	1	UART 发送控制寄存器。
DE_EN	0xB0	1	UART ‘de’ 信号断言控制。
RE_EN	0xB4	1	UART ‘re’ 信号断言控制。
DET	0xB8	1	UART 驱动输出使能的时序寄存器。
TAT	0xBC	1	UART 回转时间控制。
LCR_EXT	0xCC	1	UART 扩展线路控制寄存器。
CPR	0xF4	1	UART 参数控制寄存器。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

5.3 CAN

5.3.1 特性

(1) 支持的 CAN 总线传输规格：CAN2.0B（高达 8byte 的有效负载，由 Bosch 参考模式认证）

(2) 可编程数据速率：CAN2.0B 定义了数据速率最高为 1Mbit/s

(3) 配置接收 buffer（RB）的大小

- 通用参数选择缓冲槽的数量Ø
- FIFO-like 行为（基于双端口内存）Ø
- “未接受”或“不正确”的已接收消息不会覆盖已存储的消息

(4) 两个发送 buffer

- 1 个主发送 buffer（PTB）
- 可选择可配置的次发送 buffer（STB）

注意：可以选择是否包括 STB。 如果包括，则通用参数选择 16 个 buffer 槽 FIFO-like 的行为

- 一个 PTB 和 STB 双端口存储块

(5) 独立的可编程的内部 29bit 邮箱

- 邮箱的数量由通用参数选择，默认数量为 16

(6) 扩展的特性

- 单发传输模式（PTB 和/或 STB）
- 只接收模式
- 环回模式（内部和外部）
- 收发器待机模式

(7) 扩展的状态和错误报告

- 捕获最后发生的错误
- 捕获仲裁丢失的位置
- 可编程错误警告限制

(8) 可配置的中断源

(9) 完全同步和综合 HDL 设计

5.3.2 操作步骤

(1) 设置 CAN 位时序。硬件复位后，控制器会自动配置系统时钟下为 16MHz 时的 f_{BUS} = 1 M

(2) 如果需要：可以支持/关闭中断请求

- 发送：TPIE 或 TSIE
- 接收：RIE 和 RAFIE, RFIE, ROIE（需要时）

(3) 对于发送：

- 向发送 buffer 写入 1 帧（PTB 或 STB）
- 启动发送：用 TPE、TSONE 或 SALL
- 当需要时等待选择的中断

(4) 对于接收：

- 等待选择的中断
- 从 RB 上读取接收到的帧并通过 RCTRL 寄存器确定帧的类型

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(5) 其他特性：

- 邮箱
- Buffer 控制：TSSTAT 和 RSTAT
- 错误处理（bits EIE, EIF, bits EWL register ERRINT, RECNT, TECNT, EALCA）

5.3.3 寄存器

注意：CAN 基地址为：0x40029000。

名称	偏移	大小(x8)	说明
RBUF	0x00-0x47	1	CAN 接收 buffer 寄存器。
TBUF	0x48 - 0x8f	1	CAN 发送 buffer 寄存器。
CFG_STAT	0x90	1	CAN 配置寄存器。
TCMD	0x91	1	CAN 指令寄存器
TCTRL	0x92	1	CAN 发送控制寄存器
RCTRL	0x93	1	CAN 接收 buffer 的控制寄存器。
RTIE	0x94	1	CAN 发送/接收的中断控制。
RTIF	0x95	1	CAN 发送/接收中断的标志寄存器。
ERRINT	0x96	1	CAN 故障中断使能控制和标志寄存器。
LIMIT	0x97	1	CAN 警告界限设置控制。
BITTIME_0	0x98	1	CAN 同步跳转宽度和位定时配置。
BITTIME_1	0x99	1	CAN 同步跳转宽度和位定时配置。
BITTIME_2	0x9a	1	CAN 同步跳转宽度和位定时配置。
S_PRESC	0x9c	1	预分频设置控制。
F_PRESC	0x9d	1	预分频设置控制。
TDC	0x9e	1	发送延时补偿使能控制。
EALCAP	0xa0	1	错误类型标志和仲裁丢失捕获控制。
RECNT	0xa2	1	接收错误计数值。
TECNT	0xa3	1	发送错误计数值。
ACFCTRL	0xa4	1	邮箱控制。
ACF_EN_0	0xa6	1	邮箱使能控制。
ACF_EN_1	0xa7	1	邮箱使能控制。
ACF_0/ACF_1 /ACF_2/ACF_3	0xa8/0xa9/0xaa /0xab(5:0)	1	CAN 邮箱寄存器
ACF_3	0xab	1	验收掩码 IDE 位的配置控制

Copyright ©2020, www.tai-action.com. All Rights Reserved.

6 中断系统

6.1 简介

32 位 Cortex-M3 核与嵌套向量中断控制器（NVIC）集成，可实现高效异常和中断处理。NVIC 实现了低延迟异常、中断处理和电源管理控制。此外，Cortex-M3 核还与内核紧密耦合。

6.2 特性

(1) 32 位 Cortex-M3 核系统异常；
(2) 91 种可屏蔽外设中断；
(3) 4bit 的中断优先级配置位——共提供 16 个中断优先级；
(4) 高效的中断处理；
(5) 支持异常抢占和咬尾中断；
(6) 三种触发方式：上升沿触发，下降沿触发和任意边沿触发；
(7) 支持软件中断或事件触发；
(8) 支持触发源可配置；

6.3 中断功能描述

32 位 Cortex-M3 核和嵌套向量中断控制器（NVIC）区分处理程序模式下所有异常和进程的优先级。发生异常时，系统会自动将当前处理器的工作状态压入堆栈，并在完成中断服务子程序（ISR）后将其推出堆栈。定向量与当前工作状态堆栈并行执行，这提高了中断进入的效率。 处理器支持位尾中断，可以实现背靠背中断，大大减少工作状态重复切换带来的开销。 表 6-1 显示 32 位 Cortex-M3 核中各种 NVIC 的执行情况。

表 6-1 32 位 Cortex-M3 核中 NVIC 的中断类型

编号	异常类型	优先级	说明
0	-	-	保存
1	Reset	-3	复位
2	NMI	-2	不可屏蔽的中断
3	硬件（hard） fault	-1	各种硬件级别故障
4	MemoryManagement fault	可编程配置	内存管理中断
5	总线（Bus） fault	可编程配置	预取指故障、存储器访问故障
6	用法（Usage）fault	可编程配置	未定义指令或非法状态
7-10	-	-	保留
11	SVcall	可编程配置	通过 SWI 指令实现系统服务调用
12	调试监视器（Debug monitor）	可编程配置	调试监控器
13	-	-	保留
14	PendSV	可编程配置	可挂起的系统服务请求
15	SysTick	可编程配置	系统节拍定时器

SysTick 的校准值配置为 26'h2000000，SysTick 的时钟频率配置为 HCLK。如果 HCLK 的时钟配置为 78MHz，则 SysTick 将每 430ms 中断一次。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

表 6-1 中断向量表

编号	异常类型	优先级	外设中断说明
16	IRQ0	可编程配置	IIC0_IRQn
17	IRQ1	可编程配置	-
18	IRQ2	可编程配置	-
19	IRQ3	可编程配置	USART0_IRQn
20	IRQ4	可编程配置	-
21	IRQ5	可编程配置	USART2_IRQn
22	IRQ6	可编程配置	-
23	IRQ7	可编程配置	-
24	IRQ8	可编程配置	-
25	IRQ9	可编程配置	-
26	IRQ10	可编程配置	QEI_IRQn
27	IRQ11	可编程配置	WDT_IRQn
28	IRQ12	可编程配置	FADC_IRQn
29	IRQ13	可编程配置	DMA0_IRQn
30	IRQ14	可编程配置	CAN_IRQn
31	IRQ15	可编程配置	GPIOA_IRQn
32	IRQ16	可编程配置	GPIOB_IRQn
33	IRQ17	可编程配置	GPIOC_IRQn
34	IRQ18	可编程配置	-
35	IRQ19	可编程配置	-
36	IRQ20	可编程配置	-
37	IRQ21	可编程配置	SVPWM_IRQn
38	IRQ22	可编程配置	-
39	IRQ23	可编程配置	TIM1_IRQn
40	IRQ24	可编程配置	TIM2_IRQn
41	IRQ25	可编程配置	TIM3_IRQn
42	IRQ26	可编程配置	-
43	IRQ27	可编程配置	SARADC_IRQn
44	IRQ28	可编程配置	RMS0_IRQn
45	IRQ29	可编程配置	RMS1_IRQn
46	IRQ30	可编程配置	RMS2_IRQn
47	IRQ31	可编程配置	SINCOS0_IRQn
48	IRQ32	可编程配置	SINCOS1_IRQn
49	IRQ33	可编程配置	MATRIX_MULT_IRQn
50	IRQ34	可编程配置	IIR0_IRQn
51	IRQ35	可编程配置	IIR1_IRQn
52	IRQ36	可编程配置	IIR2_IRQn
53	IRQ37	可编程配置	-
54	IRQ38	可编程配置	-
55	IRQ39	可编程配置	-
56	IRQ40	可编程配置	SPWM_IRQn
57	IRQ41	可编程配置	-
58	IRQ42	可编程配置	-
59	IRQ43	可编程配置	-
60	IRQ44	可编程配置	-
61	IRQ45	可编程配置	-
62	IRQ46	可编程配置	-
63	IRQ47	可编程配置	EPWM_IRQn
64	IRQ48	可编程配置	LVD_IRQn
65	IRQ49	可编程配置	DFTRANS0_IRQn

Copyright ©2020, www.tai-action.com. All Rights Reserved.

66	IRQ50	可编程配置	DFTRANS1_IRQn
67	IRQ51	可编程配置	DFTRANS2_IRQn
68	IRQ52	可编程配置	CRC_DMA_IRQn
69	IRQ53	可编程配置	ARCTAN0_IRQn
70	IRQ54	可编程配置	ARCTAN1_IRQn
71	IRQ55	可编程配置	ARCTAN2_IRQn
72	IRQ56	可编程配置	DATADMA_IRQn
73	IRQ57	可编程配置	WAKEPND_IRQn
74	IRQ58	可编程配置	TIM4_IRQn
75	IRQ59	可编程配置	TIM5_IRQn
76	IRQ60	可编程配置	TIM6_IRQn
77	IRQ61	可编程配置	-
78	IRQ62	可编程配置	EVSYS_IRQn

Copyright ©2020, www.tai-action.com. All Rights Reserved.

7 DMAC（DMA 控制器）

7.1 简介

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-M3 核内核和 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。

7.2 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次传输，4 次、8 次和 16 次增量突发传输;
(3) 支持外设 DMA：2 个 UART，1 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。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

7.3 结构框图

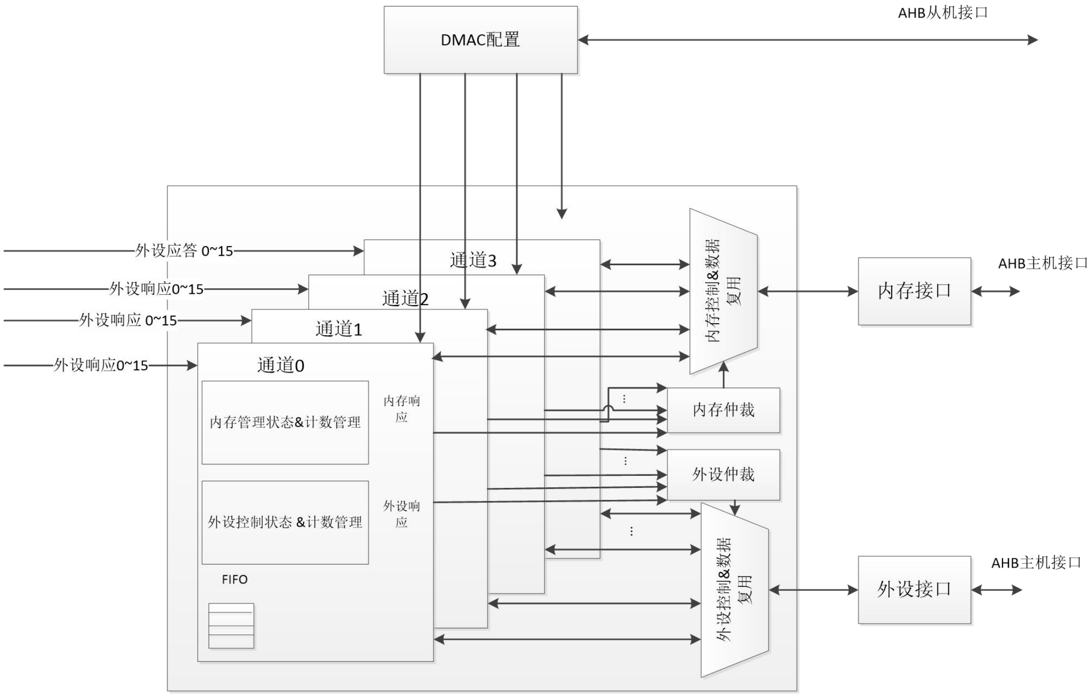


图 7-1 DMAC 结构框图

7.4 DMA 请求通道

表 7-1 DMA 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	-	不可用
dma req channel 3	-	不可用
dma req channel 4	–	不可用
dma req channel 5	–	不可用
dma req channel 6	–	不可用
dma req channel 7	–	不可用
dma req channel 8	UART0_dma_tx_req	-
dma req channel 9	UART0_dma_rx_req	-
dma req channel 10	–	不可用
dma req channel 11	–	不可用
dma req channel 12	UART2_dma_tx_req	-
dma req channel 13	UART2_dma_rx_req	-

Copyright ©2020, www.tai-action.com. All Rights Reserved.

8 ADC

8.1 SARADC & DAC & Comparator

8.1.1 特性

(1) 内置 18 通道 12 位 SARADC，156K SPS /通道，分辨率> 10.5bit（ENOB）

(2) SARADC0~13 仅具有 SARADC 功能，但 SARADC14~16、18 不仅是 SARADC，而且可以作为 DAC 和比较器重复使用

(3) 内置 PGA（可编程增益放大器）x0.5~8

SARADC0~SARADC13 的框图如下：

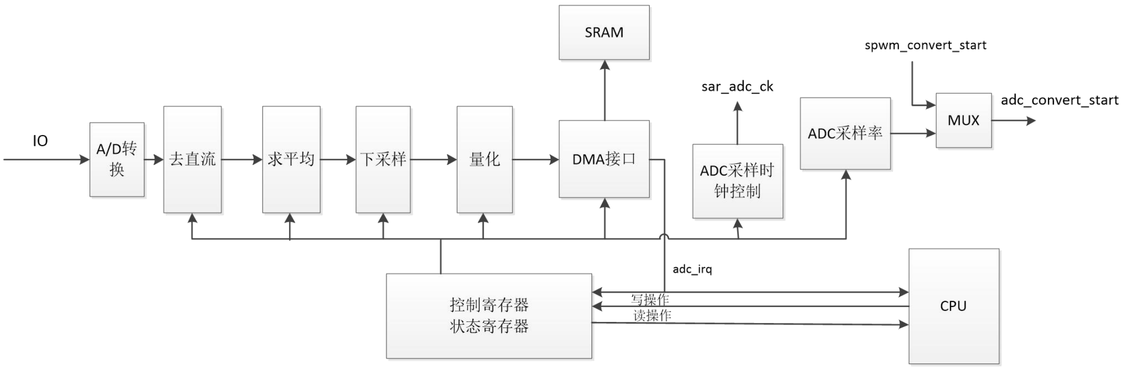


图 8-1 SARADC0~13 框图

SARADC14~16、18 的框图如下：

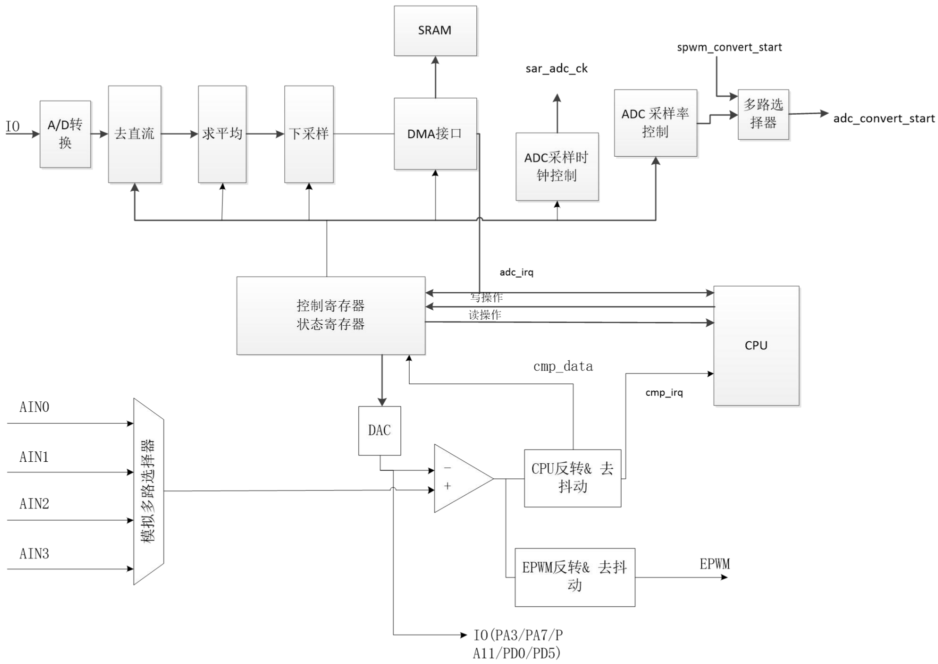


图 8-2 SARADC14~16、18 框图

Copyright ©2020, www.tai-action.com. All Rights Reserved.

8.1.2 操作步骤

(1) 使能 SARADC 时钟：将 saradc_clk_en（CLK_CON4[4]）置 1（该寄存器详细见 Timer&eflash&CRC&Watchdog&GPIO&SYSCTRL User Guide 中）。
(2) 释放 SARADC 软复位：将 saradc_soft_rst_（SYS_CON2[18]）置 1（该寄存器详细见 Timer&eflash&CRC&Watchdog&GPIO&SYSCTRL User Guide 中）。
(3) 根据不同的需求配置不同的寄存器：
   - 需要 DMA 功能时，配置 DMA 长度：配置 DMA_LEN（SARADC_CDIV_DMALENx[12：0]）；
   - 配置 DMA 地址：配置 SARADC_DMASTADDR 寄存器；
   - 启动 DMA：配置 SARADC_CON1 寄存器；

注意：只有 SARADC 有 DMA 功能。

(4) 当 PWM 不启动 ADC 转换时：
   - 设置采样率：配置 CONVERT_DIV （SARADC_CDIV_DMALEN[31:14]）；
   - 启动采样率计数器：设置 CONVERT_CNT_EN（SARADC_CDIV_DMALEN[13:0]）；
   - 启动 ADC 时配置内部 convert_cnt 寄存器：配置 SARADC_START_POINT 寄存器；
(5) 配置控制寄存器： SARADC_COM_ACFG 寄存器。
(6) 配置直流偏量：配置 DC_OFFSET （SARADC_ACFG[27:16]）；配置下采样：DOWNSAMPLE_SEL （SARADC_ACFG[30:28]）；求平均时进行以下配置：配置 AVERAGE_SEL（SARADC_ACFG[15:14]）和 AVERAGE_EN（SARADC_ACFG[13]）；根据模拟模块的其他需求进行配置：在 SARADC_ACFG 寄存器中配置。
(7) 配置量化寄存器：SARADC_QUANTIFY_CONx。
(8) 当需要中断时配置中断相关寄存器：SARADC_INT_CONTROL0、SARADC_INT_CONTROL1 和 SARADC_INT_CONTROL2 寄存器。
(9) 使能模拟 ADC 功能：ADCEN （SARADC_ ACFG[0]）。
(10) 启动 ADC：配置 SARADC_CON 寄存器。
(11) 等待相应的标志位再进行操作。

8.2 寄存器

注意：SARADC 基地址为：0x40036000。

名称	偏移量	大小(x32)	说明
SARADC_CON	0x00	1	ADC0~18 使能控制。
SARADC_CON1	0x18	1	ADC0~13 DMA 使能控制。
SARADC_PENDING0	0x1e0	1	ADC0~18 采样完成的标志。
SARADC_PENDING1	0x1e4	1	ADC0~13 DMA 半满的标志。
SARADC_PENDING2	0x1e8	1	ADC0~13 DMA 全满的标志。
SARADC_PENDING0_CLR	0x1ec	1	清除 ADC0~13 采样完成的标志。
SARADC_PENDING1_CLR	0x1f0	1	清除 ADC0~13 DMA 半满的标志。
SARADC_PENDING2_CLR	0x1f4	1	清除 ADC0~13 DMA 全满的标志。
SARADC_INT_CONTROL0	0x1f8	1	ADC0~18 采样完成的中断使能控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

SARADC_INT_CONTROL1	0x1fc	1	ADC0~13 DMA 半满的中断使能控制。
SARADC_INT_CONTROL2	0x200	1	ADC0~13 DMA 全满的中断使能控制。
SARADC_CDIV_DMALEN0 ~SARADC_CDIV _DMALEN13	0X20/0X24/ 0X28/0X2C/ 0X30/0X34/ 0X38/0X3C/ 0X40/0X44/ 0X48/0X4C/ 0X50/0X54	1	采样率配置和 DMA 长度设置。
SARADC_CDIV _DMALEN14~SARADC _CDIV_DMALEN18	0x58/0x5C /0x60/0x64 /0x68	1	采样率控制。
SARADC_DMASTADDR0~SARADC_DMASTADDR13	0X70/0X74 /0X78/0X7C /0X80/0X84 /0X88/0X8C /0X90/0X94 /0X98/0X9C /0XA0/0XA4	1	DMA 首地址。
SARADC_COM_ACFG	0X14	1	-
SARADC_ACFG0~13	0xB0/0xB4 /0xB8/0xBC /0xC0/0xC4 /0xC8/0xCC /0xD0/0xD4 /0xD8/0xDC /0xE0/0xE4	1	ADC 采样相关控制。
SARADC_QUANTIFY _CON0~13	0x148/0x14C/0x150/0x154/0x158/0x15C/0x160/0x164/0x168/0x16C/0x170/0x174/0x178/0x17C	1	ADC 量化相关控制。
SARADC_START _POINT0~18	0x194~0x1DC	1	将计数值转换为 START POINT 的转换启动位。
SARADC_ ACFG14~18	0xE8~0xF8	1	ADC 配置控制和 CMP、ADC、DAC 的使能控制。
DACCMP_CON14~18	0x130~0x140	1	DAC、 ADC 配置控制。
DACCMP_DATA14~18	0x218~0x22	1	DAC、CMP 输出控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

	8		
DACCMP_DEBOUNCE 14~18	0x204~0x214	1	比较器输出去抖控制。
SARADC_DATA_10	0x100	1	ADC0/1 的最终采样数据。
SARADC_DATA_32	0x104	1	ADC2/3 的最终采样数据。
SARADC_DATA_54	0x108	1	ADC4/5 的最终采样数据。
SARADC_DATA_76	0x10C	1	ADC6/7 的最终采样数据。
SARADC_DATA_98	0x110	1	ADC8/9 的最终采样数据。
SARADC_DATA_1110	0x114	1	ADC10/11 的最终采样数据。
SARADC_DATA_1312	0x118	1	ADC12/13 的最终采样数据。
SARADC_DATA_1514	0x11C	1	ADC14/15 的最终采样数据。
SARADC_DATA_1716	0x120	1	ADC16/17 的最终采样数据。
SARADC_DATA_xx18	0x124	1	ADC18 的最终采样数据。

8.3 FSARADC

这是高速 SARADC（FSARADC）控制器模块，主要包括了 2 个采样保持电路和一个 12bit 的模数转换器。FSARADC 支持 16 个转换通道，每个转换通道称为一个 ADC SOC（Start-Of-Conversions）。采样保持电路可以同时采样也可连续采样。每个转换通道可以配置模拟输入和触发源，并且每个触发脉冲可以启动一个单独的转换。

8.3.1 特性

FSARADC 模块功能包括以下：

(1) 双采样保持电路（A，B）。
(2) 支持 A 和 B 同时采样和连续采样。
(3) 支持最多 14 路多路复用模拟输入。
(4) 支持 16 个 SOC，每个 SOC 都被配置为触发源、模拟输入和采样窗口时间。
(5) 配置每个 SOC 的转换优先级。
(6) 每个 SOC 都有可单独寻址的结果寄存器。
(7) 支持多种 SOC 触发源：
   - CPU 或软件触发源；
   - 高达 15 个 EPWM/SPWM/SVPWM 触发源；
   - 内部定时器溢出触发源；
(8) 每个 SOC 都支持 DMA。
(9) 多个中断源：
   - 当某个 SOC 转换结束会产生一个中断；
   - 当某个 SOC 的 DMA FIFO 达到半满会产生一个中断；
   - 当某个 SOC 的 DMA FIFO 全满会产生一个中断；
(10) 支持数据后处理
   - 直流偏量的补偿；
   - 简单的数据平均；
   - 简单的数据下采样；
   - 可配置系数乘法与转换数据；

Copyright ©2020, www.tai-action.com. All Rights Reserved.

模块框图如下：

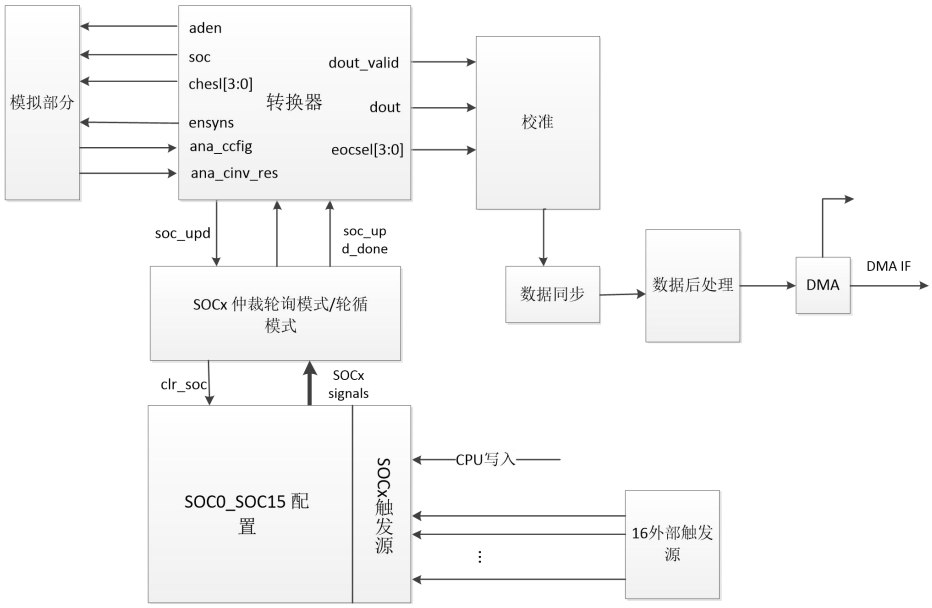


图 8-3 FSARADC 模块框图

8.3.2 操作步骤

(1) 模拟电路初始化：

- 配置 FADCSFRANACON1 寄存器：配置 ana_ens2da, ana_ensha=1, ana_vcmiaen=1；配置 ana_ensadb, ana_enshb=1, ana_vcmiben=1；
- 配置 FADCSFRANACON0 寄存器：配置 ana_ctrl_sel=1, ana_cmpbsen=1, ana_trim=0, ana_cmpen=1, ana_buffen=1, ana_biasen=1, ana_adcen=1；

(2) SOC 控制：

- 配置每路 SOC 的 S/H 窗口：配置 FADCSFRSOCCONx(x=0…7)中的 smp_winx[6:0]（x=0 或 1）；
- 设置寄存器选择每路 SOC 的触发源：FADCSFRSOCCONx(x=0…7)中的 trisel[3:0] 或 fc_tri（FADCSFRSOCFLAG[31:16]）或 FADCSFRSOCTIMER x(n=0…15)寄存器；
- 设置每路 SOC 的模拟输入：FADCSFRSOCCONx(x=0…7)中的 chsel[3:0] ；
- 配置每路 SOC 的 FADCSFRSOCCON8 寄存器的下列位：soc_hpri, simulen；

(3) 配置校准：

  - 配置 FADCSFRWCOEFF0 ~ FADCSFRWCOEFF15 寄存器；
  - 配置 FADCSFRCALIB0 寄存器：CALIB_WMODE=1；
  - 配置 FADCSFRCALIB1 寄存器：CALIB_NORM_OUTSEL=0；

(4) 配置 DMA：

- DMA 首地址：配置 FADCDMAADDR0~FADCDMAADDR15 寄存器；
- DMA FIFO 长度：配置 FADCDMALEN0~FADCDMALEN15 寄存器；

(5) 配置中断：FADCINT0 和 FADCINT1 寄存器

(6) 配置数据后处理：FADCPPROC0CONx(x=0…15)和 FADCPPROC1CONx(x=0…15)

Copyright ©2020, www.tai-action.com. All Rights Reserved.

寄存器。

(7) 使能 FSARADC：m_en（FADCSFRADCCON0[0]）置 1, fadc_en（FADCCON0[0]）。

(8) 配置每路 SOC 的 FADCSFRSOCCON8 寄存器，使能触发：trien（FADCSFRSOCCON8[]）

(9) 等待中断或标志位置 1，从 FADCRES0-FADCRES7 或 DMA FIFO 处获取转换的数据。

(10) 清除相关的标志位，返回（9）并等待其他结果。

8.3.3 寄存器

注意：FSARADC 基地址：0x40037000。

A. 系统时钟域寄存器

(1) 系统时钟域寄存器可以通过寄存器地址直接访问；

(2) FADCCON0 ~ FADCRES7 为系统时钟域寄存器；

名称	偏移量	大小(x32)	说明
FADCCON0	0x0000	1	系统时钟域相关功能使能控制。
FADCACSCON	0x0004	1	高速时钟域读写控制。
FADCACSDAT	0x0008	1	对指定 ads_addr 寄存器进行读写操作。
FADCINT0	0x000c	1	SOC0~15 的中断使能控制。
FADCINT1	0x0010	1	SOC0~15 DMA 半满/全满的中断使能控制。
FADCSMPFLAG	0x0014	1	SOC0~15 转换完成标志。
FADCDMAFLAG0	0x0018	1	SOC0~15 DMA 半满的中断标志及该中断标志清除。
FADCDMAFLAG1	0x001c	1	SOC0~15 DMA 全满的中断标志及该中断标志清除。
FADCDMAADDR0 - FADCDMAADDR15	0x0020-0x005c	1	SOC0~15 DMA 首地址配置。
FADCDMALEN0 - FADCDMALEN15	0x0060-0x009c	1	SOC0~15 DMA 长度与使能控制。
FADCPPROC0CON0 - FADCPPROC0CON15	0x00a0-0x00dc	1	数据后处理相关配置。
FADCPPROC1CON0 - FADCPPROC1CON15	0x00e0- 0x012c	1	SOC0~15 量化相关配置。
FADCRES0-FADCRES7	0x0120-0x013c	1	在非 DMA 模式下保存的 SOC 2N/SOC 2N+1 的转换结果。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

B. FADC 高速时钟域寄存器

FADC 高速时钟域寄存器可以由 FADCACSCON，FADCACSDAT 访问。

名称	偏移量	大小(x32)	说明
FADCSFRANACON0	0x0000	1	初始化 FADC 时 SDK 已配置好。
FADCSFRANACON1	0x0038	1	初始化 FADC 时 SDK 已配置好。
FADCSFRADCCON0	0x0002	1	-
FADCSFRSOCCON (n=0…7)	0x0003 - 0x00b5	1	SOC0~15 硬件触发源和模拟通道的配置选择。
FADCSFRSOCCON8	0x000b	1	SOC 相关配置控制。
FADCSFRSOCFLAG	0x000c	1	SOC 软件触发相关配置控制。
FADCSFRSOCOVFL	0x000d	1	SOC 触发溢出标志及清除控制。
FADCSFRSOCTIMER (n=0…15)	0x0020- 0x005c	1	SOC 计数器 0~15 相关配置控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

9 PWM

9.1 EPWM

9.2 特性

在 TMF5201 中一共有 7 个 EPWM 模块，每个 EPWM 模块都支持以下功能：

(1) 专用的 16 位时间控制器，可用于周期和频率的控制。

(2) 两个 PWM 输出（EPWMxA 和 EPWMxB）可进行以下配置：

- ➢ 两个独立的 PWM 输出进行单边控制；
- ➢ 两个独立的 PWM 输出进行双边对称控制；
- ➢ 一个独立的 PWM 输出进行双边不对称控制；

(3) 通过软件对 PWM 信号进行异步覆盖控制。

(4) 与其他 EPWM 模块有关的可编程超前或滞后相位控制。

(5) 在一个循环基础上的硬件锁定（同步）相位关系。

(6) 具有独立的上升沿和下降沿死区延迟控制。

(7) 可编程故障区（trip zone）用于故障时的周期循环（cycle-by-cycle trip）控制和单次（one-shot trip）控制。

(8) 一个控制条件可使 PWM 输出为强制高、低或高阻抗逻辑电平。

(9) EPWM 模块中的所有事件都可以触发 CPU 中断和启动 ADC 开始转换（ADC SOC）。

(10) 可编程事件有效降低了中断时 CPU 的负担。

如图所示，每个 EPWM 模块都连接着输入输出信号。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

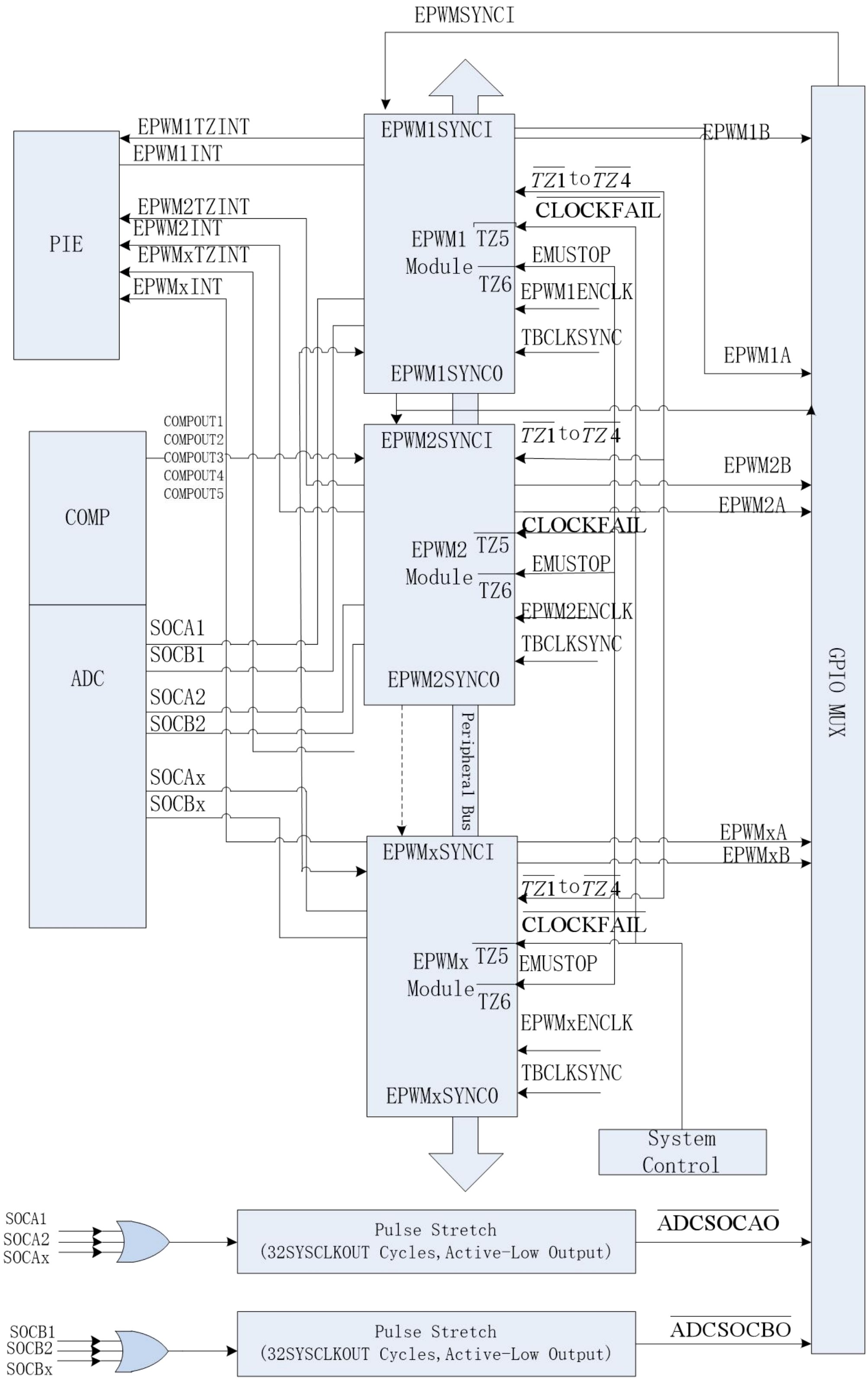


图 9-1 EPWM 模块框图

9.2.1 操作步骤

不同的情况需要进行不同的操作，详情请看详细版 PWM 文件各个子模块介绍。

9.2.2 寄存器

注意：EPWM 基地址：0x40038000。

名称	偏移量	大小(x32)	说明
EPWM_TTCTL	0x00	1	EPWMx（x=0…6）的使能控制。
EPWM_ADCSEL0	0x250	1	启动ADC0~8采样的EPWM SOC 信号选择。
EPWM_ADCSEL1	0x254	1	启动 ADC10~13 采样的 EPWM SOC 信号选择。
EPWMx_TBCTL	0x00/0x54/0xa8/0xfc/0x150 /0x1a4/0x1f8	1	-
EPWMx_TBPRD_SD	0x04/0x58/0xac/0x100/0x154 /0x1a8/0x1fc	1	-
EPWMx_TBPHASE	0x08/0x5c/0xb0/0x104/0x158 /0x1ac/0x200	1	EPWM 时基计数器方向选择/相位选择。
EPWMx_CMPCTL	0x0c/0x60/0xb4 /0x108/0x15c/0x1b0/0x204	1	时基比较器 A/B/C 相关配置选择。
EPWMx_CMPA_SD	0x10/0x64/0xb8/0x10c/0x160 /0x1b4/0x208	1	时基计数比较器 A 的读取值。
EPWMx_CMPB_SD	0x14/0x68/0xbc/0x110/0x164 /0x1b8/0x20c	1	时基计数比较器 B 的读取值。
EPWMx_CMPC_SD	0x18/0x6c/0xC0/0x114/0x168 /0x1bC/0x210	1	时基计数比较器 C 的读取值。
EPWMx_AQCTLAB	0x1C/0x70/0xC4/0x118/0x16C /0x1C0/0x214	1	-
EPWMX_AQSFRC	0x20/0x74/0xC8/0x11C/0x170 /0x1C4/0x218	1	-
EPWMX_AQCSFRC	0x24/0x78/0xCC/0x120 /0x174/0x1C8/0x21C	1	输出 A/B 上的连续软件强制事件选择。
EPWMX_DBCTL	0x28/0x7C/0xD0/0x124/0x178/0x1CC/0x220	1	-
EPWMX_DBDELAY	0x2C/0x80/0xD4/0x128/0x17C /0x1D0/0x224	1	上升沿/下降沿延时计数器。
EPWMx_ETCTL	0x30/0x84/0xD8/0x12C/0x180 /0x1D4/0x228	1	-
EPWMx_ETFLAG	0x34/0x88/0xDC/0x130/0x184 /0x1D8/0x22C	1	EPWM 状态标志位。
EPWMx_DCCTL	0x38/0x8C/0xE0/0x134 /0x188/0x1DC/0x230	1	-

Copyright ©2020, www.tai-action.com. All Rights Reserved.

EPWMx_DCTRIPSEL	0x3C/0x90/0xE4/0x138/0x18C /0x1E0/0x234	1	-
EPWMx_DCCAP	0x40/0x94/0xE8/0x13C/0x190 /0x1E4/0x238	1	-
EPWMx_BLANKOFFSET	0x44/0x98/0xEC/0x140/0x194 /0x1E8/0x23C	1	Blanking window 偏移量设置。
EPWMx_WINWIDTH	0x48/0x9C/0xF0/0x144/0x198/0x 1EC/0x240	1	Blanking Window 宽度设置。
EPWMx_TZCTL	0x4C/0xA0/0xF4/0x148 /0x19C/0x1F0/0x244	1	EPWM 相关使能控制。
EPWMx_TZFLAG	0x50/0xA4/0xF8/0x14C/0x1A0 /0x1F4/0x248	1	EPWM 相关中断使能控制。

9.3 SPWM

SPWM，正弦脉冲宽度调制，可以在 IO 输出指定占空比和周期的 PWM 波。支持三电平模式和两电平模式，支持单规则采样，即一个三角形周期只采样一次占空比。SPWM 的模块框图如下：

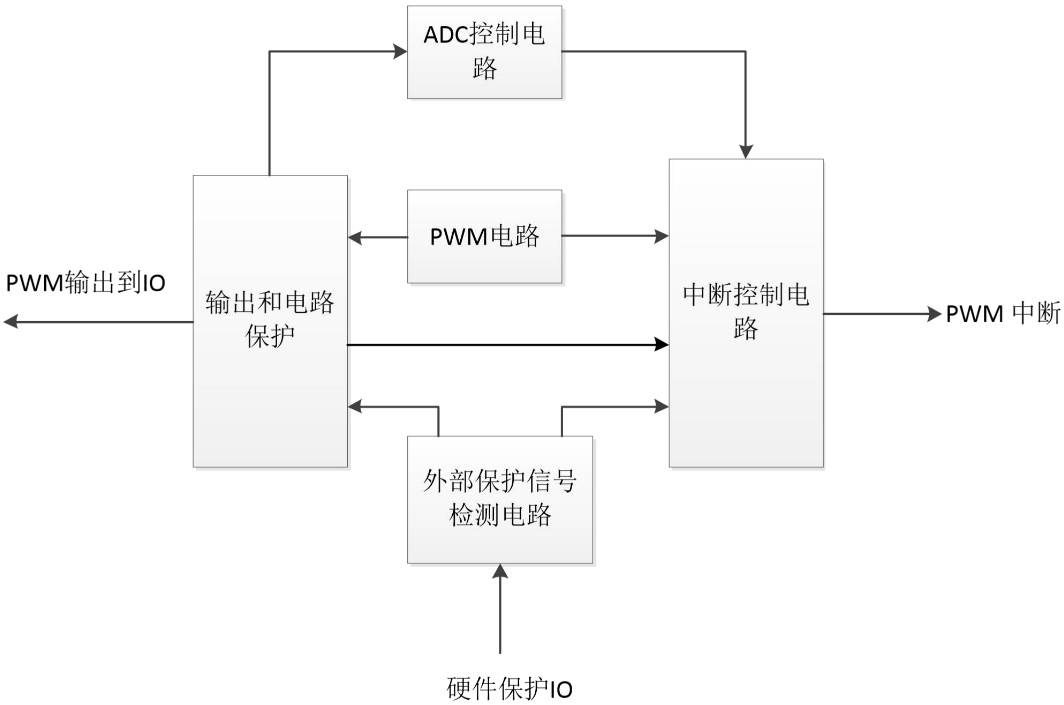


图 9-2 SPWM 模块框图

9.3.1 操作步骤

(1) 配置 SPWM_CON 寄存器，将 SFR_CFG_EN（SPWM_CON[24]）设为 1。

(2) 配置 SPWM_PERIOD0 和 SPWM_PERIOD1 寄存器：设置周期和死区时间，最小脉冲宽度推荐写 0。

(3) 配置 SPWM_CNT0，SPWM_CNT1，SPWM_CNT2 寄存器：设置为三电平或者两电平并选择是否打开 load data 中断。其他全部选择默认值即可。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(4) 配置 FAULT_INFO 和 FAULT_INVERT 寄存器，可关闭指定的 fault 检测（FAULT_INFO[15:0]），并且可以设置 fault valid 是低电平或高电平有效（FAILT_INVERT[15:0]）。
(5) 配置 SPWM_MATCH 和 SPWM_ADCC 寄存器，可以在 SPWM 周期内任意一个时间产生 match pending 并触发中断，并且可以在 SPWM_ADCC 中配置是否触发 ADC 采样。
(6) 配置 SPWM_CON 寄存器：设置 FAULT_DEBOUNCE（SPWM_CON[23:20]）进行 fault 检测防抖时间；设置 SYSERR_PT_EN（SPWM_CON[25]） = 1，打开系统错误检测。设置 SPWM_ADCC 寄存器可以触发某个或某几个 ADC 采样。
(7) 配置 SPWM_CON 寄存器，写 SET_PWM_EN（SPWM_CON[0]）为 1，写 SFR_CFG_EN（SPWM_CON[24]）为 0 关闭 SFR 写使能。
(8) 定时将当前数据写入 PWM_BUFDATA（SPWM_DATABUF0/1/2[15:0]）。也可以通过将寄存器 SPWM_CNT0，1，2 的 LOAD_INT_EN（SPWM_CNT0/1/2[18]）置 1 来触发更新数据到 PWM_BUFDATA。

9.3.2 寄存器

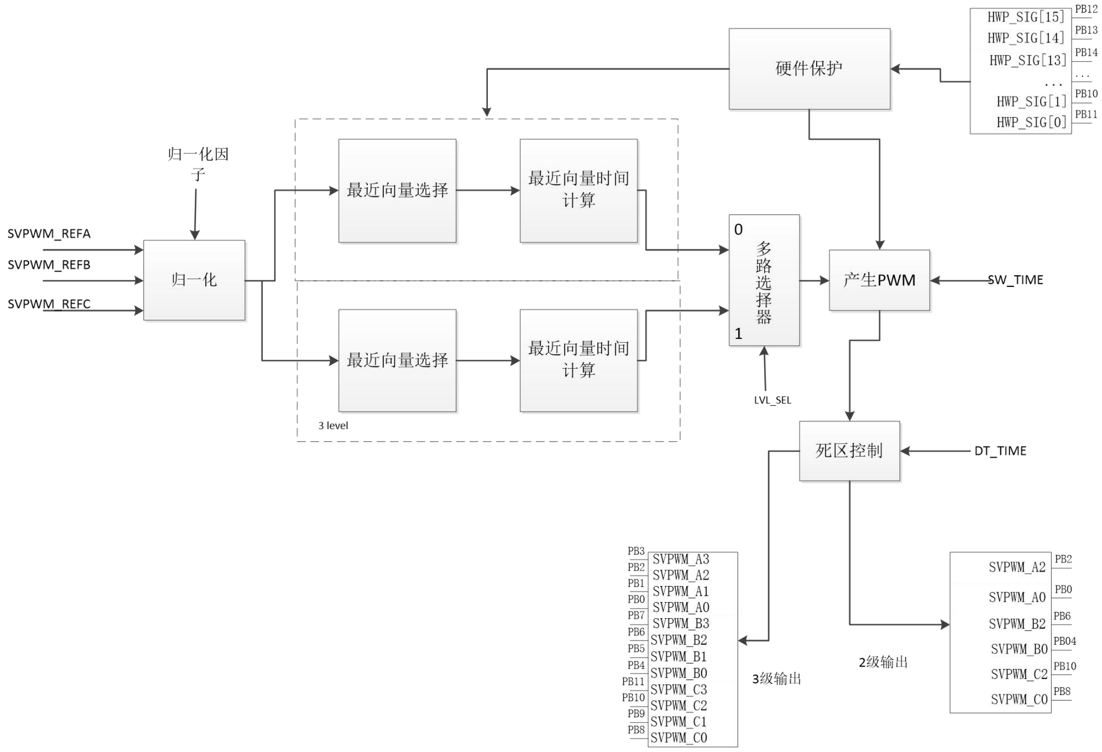
注意：SPWM 基地址：0x40035280。

名称	偏移量	大小(x32)	说明
SPWM_CON	0x00	1	SPWM 配置寄存器。
SPWM_PERIOD0	0x04	1	PWM 周期配置。
SPWM_PERIOD1	0x28	1	最小脉冲、死区时间配置。
SPWM_MATCH	0x2c	1	-
SPWM_ADCC	0x30	1	ADC 控制使能。
SPWM_CNT0/1/2	0x34/0x38/0x3c	1	-
FAULT_INFO	0x20	1	故障 pending 及故障检测使能。
FAULT_INVERT	0x24	1	故障反转控制。
SPWM_DATAUSE0/1/2	0x08/0x10/0x18	1	发送到 PWM 发生器的数据。
SPWM_DATABUF0/1/2	0x0c/0x14/0x1c	1	发送到 SPWM_DATAUSE0/1/2 寄存器的数据。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

9.4 SVPWM

TMF5201 集成了一个基于空间矢量脉冲宽度调制模块，本模块的主要功能是根据提供的 abc 坐标的输入数据，自动转换为三相 2 电平或者三相 3 电平的互补对称 PWM 输出。模块提供了可配置的死区控制和必要的硬件保护模式，当遇到紧急情况，可及时停止 PWM 输出，以保护外围硬件。框图如下：



9-3 图 SVPWM 模块框图

9.4.1 特性

(1) 支持三相 abc 信号：Ua、Ub、Uc，位宽为 16bit。
(2) 支持三相 2 电平或三相 3 电平的 SVPWM。
(3) 支持可配置的死区控制。
(4) 支持硬件保护。

9.4.2 操作步骤

(1) 将寄存器 SVPWM_CON0 的 SVPWM_CFG_EN 设为 1 去访问另一个控制位。
(2) 将寄存器 SVPWM_CON0 的 LVL_SEL 设为三电平或两电平输出。
(3) 在寄存器 SVPWM_CON1 的 SW_TIME 处设置 switch period。
(4) 配置外部硬件保护：HWPRT_EN（SVPWM_CON0[4]）、FAULT_DEAT_DIS（SVPWN_CON4[15:0]）和 FAULT_INVERT（SVPWM_CON5[15:0]）；你可以关闭指定的故障检测和设置当高电平或低电平时故障信号有效。根据需要，可配置硬件故障信号的防抖动时间：FAULT_DEBC_TIME（SVPWM_CON0）。
(5) 根据需要配置输入信号的归一化因子：NORM_FACTOR_SEL（SVPWM_CON0[3]）、NORM_FACTOR（SVPWM_CON3[15:0]）。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(6) 配置 SVPWM_MATCH_CON 和 SVPWM_ADC_CON 寄存器生成 match pending 并且可在 SVPWM 周期内任一时间触发中断；配置 SVPWM_ADC_CON 寄存器控制 ADC 采样。
(7) 配置 LOADDATA_INT_EN（SVPWM_CON0[5]）、HWP_INT_EN（SVPWM_CON0[6]）、MATCH_INT_EN（SVPWM_CON0[26]）和 DER_INT_EN（SVPWM_CON0[27]）进行中断设置。
(8) 在 SVPWM_REFA, SVPWM_REFB, SVPWM_REFC 中配置 abc 坐标数据。
(9) 启动 SVPWM：配置 M_EN（SVPWM_CON0[0]）为 1。
(10) 等待数据被写入，LOADDATA_PND 变为 1 或 load data 中断发生；然后将数据更新到寄存器 SVPWM_REFA, SVPWM_REFB, SVPWM_REFC 中。
(11) 如果想要关闭 SVPWM，配置 CLR_SVPWMEN（SVPWM_CON0[14]）为 1。

9.5 寄存器

注意：SVPWM 的基地址：0x40033000。

名称	偏移量	大小(x32)	说明
SVPWM_CON0	0x0000	1	SVPWM 配置寄存器。
SVPWM_CON1	0x0004	1	SVPWM 开关周期时间设置。
SVPWM_CON2	0x0008	1	SVPWM 死区时间控制。
SVPWM_CON3	0x000C	1	abc 坐标输入数据归一化因子。
SVPWM_CON4	0x0010	1	硬件故障检测使能控制。
SVPWM_CON5	0x0014	1	SVPWM 硬件故障判断设置。
SVPWM_REFA	0x0018	1	A 相输入参考信号。
SVPWM_REFB	0x001C	1	B 相输入参考信号。
SVPWM_REFC	0x0020	1	C 相输入参考信号。
SVPWM_STC0	0x0024	1	向量 A1/B 的有效时间。
SVPWM_STC1	0x0028	1	向量 A2/C 的有效时间。
SVPWM_MATCH_CON	0x002c	1	当 SVPWM 内部周期计数器的值与 MATCH_CON 的值相等时，将会产生一个脉冲信号标志位和一个中断，此位用于配置对应的匹配值。
SVPWM_ADC_CON	0x0030	1	SPWM ADC 转换使能控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

10 QEI

正交编码器接口（QEI）模块提供增量编码器接口，用于获取机械位置数据，可用于检测旋转运动系统的位置和速度，也可用于实现各种电机控制应用的闭环控制，例如开关磁阻电动机（SR）和交流感应电动机（ACIM）。

下图是 QEI 模块的简单图。QEI 模块包括：

(1) 三个输入通道：两相信号和索引脉冲

(2) 可编程数字噪声滤波器

(3) 获取机械位置数据

(4) 16 位增/减位置计数器

(5) 计算方向状态

(6) 位置测量（x2 和 x4）模式

(7) 备用 16 位定时器/计数器模式

(8) QEI 或计数器事件产生的中断

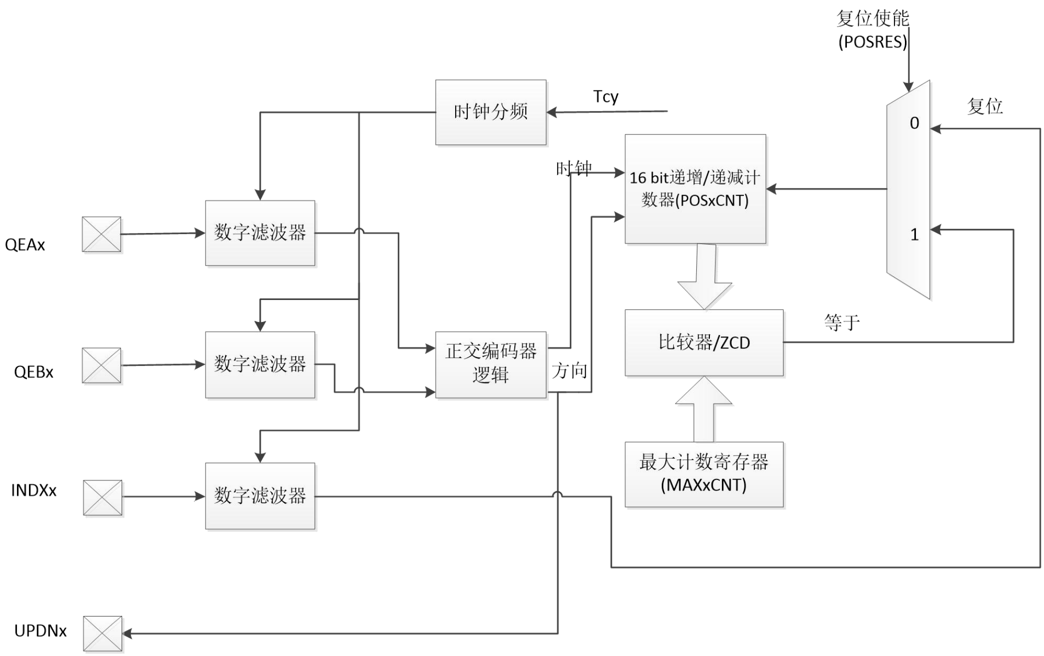


图 10-1：正交编码器模块框图

注意：仅当 QEIM <2：0> = 100 或 110 时，索引脉冲才可能复位。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

10.1 操作步骤

10.1.1 定时器/计数器功能

(1) 如果需要使用 QEA 和 QEB 信号，请配置 QEIO：

SWPAB：根据需要进行配置

QEBP：根据需要配置

QEAP：根据需要配置

(2) 如果需要使用 QEA 和 QEB 信号，请配置 DFLTCON：

QEOUT：根据需要配置

QECK：根据需要配置

(3) 配置 POSCNT，MAXCNT

(4) 配置 QEIETMTH_IE：根据需要配置

(5) 配置 QEI_CON：

UD：根据需要配置

QEIM <2：0>：配置为 001

TG：根据需要配置

TCKPS：根据需要配置

TQCS：根据需要配置

UDSRC：根据需要配置

(6) 等待完成：您可以查询标志 TMTH_F[4]以了解定时是否完成。

(7) 通过将 1 写入 QEICLR 寄存器的相应位来清除标志 TMTH_F [4]。

10.1.2 QEI 功能

(1) 配置 QEI 引脚：需要配置 IO 滤波器寄存器（DFLTCON）和 IO 控制寄存器（QEIO）。

(2) 配置计数器的初始值：配置 POSCNT 寄存器。

(3) 配置计数器的最大值：配置 MAXCNT 寄存器。

(4) 如果需要计算速度，则在 T_MODE（QEI_CON [17] = 1）时配置 QEI_TIMER_PERIOD; 当 M_MODE（QEI_CON [17] = 0）时配置 QEI_ROTATE_PERIOD。

(5) 配置控制寄存器：QEI_CON，并通过将 QEIM[2:0]设置为 010~111 中的一个值来启动 QEI 功能。

(6) 等待完成：通过查看标志位（QEIFLAG 寄存器）可以知道定时器是否完成。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

(7) 如果需要计算速度，则在 M_MODE 中：等待 SPEED_CNT_OV 置 1，然后从 QEI_TIMER_CNT_LATCH 和 QEI_ROTATE_PERIOD 寄存器读取值以计算转速。在 T_MODE 中：等待 SPEED_TIMER_OV 置 1，然后从 QEI_ROTATE_CNT_LATCH 和 QEI_TIMER_PERIOD 寄存器读取值以计算转速。

(8) 通过向 QEICLR 的相应位写入 1 来清除标志。

10.2 寄存器

注意：QEI 基地址：0x40012000。

名称	偏移量	大小(x16)	说明
QEICON	0x00	2	QEI 配置寄存器。
DFLTCON	0x04	2	数字滤波器分频选择和输出使能控制。
POSCNT	0x08	1	16 位位置/时序计数器。
MAXCNT	0x0c	1	计数器最大值设置。
QEIE	0x10	2	QEI 中断使能控制。
QEIFLAG	0x14	2	-
QEICLR	0x18	2	QEI 标志位清除控制。
QEIO	0x1c	2	-
QEI_TIMER_PERIOD	0x20	2	QEI 时间计数器周期设置。
QEI_TIMER_CNT	0x24	2	用于计算转速的时间计数器。
QEI_TIMER_CNT_LATCH	0x28	2	用于计算转速的时间计数锁存器。
QEI_ROTATE_PERIOD	0x2c	2	QEA,QEB 边沿计数器周期设置。
QEI_ROTATE _CNT	0x30	2	QEA,QEB 边沿计数器。
QEI_ROTATE_CNT _LATCH	0x34	2	QEA,QEB 边沿计数器锁存器。
POSCNT _LATCH	0x38	2	当复位 POSCNT 时，将 POSCNT 的值锁存到此寄存器。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

11 事件控制单元（EVSYS）

11.1 功能

从源触发信号和目的触发信号列表中中选择一组连接到某一通道中，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。软件可以为每次触发的目的信号配置触发数 M。例如，在源信号 1 触发目标信号 1 之后，目标信号 1 需要运行 M 次，然后触发源信号 A。

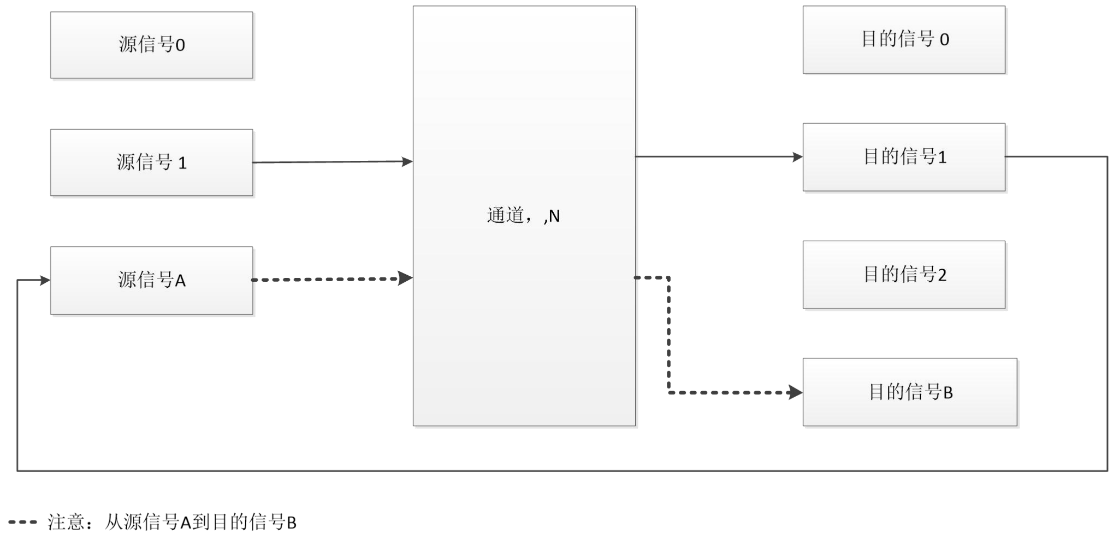


图 11-1 一对一结构图

从源触发器和目标触发器列表中选择一组连接到某一通道中。如下图所示，源信号 1 可以同时触发目标信号 0/1/B。当信号全部都完成时，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。

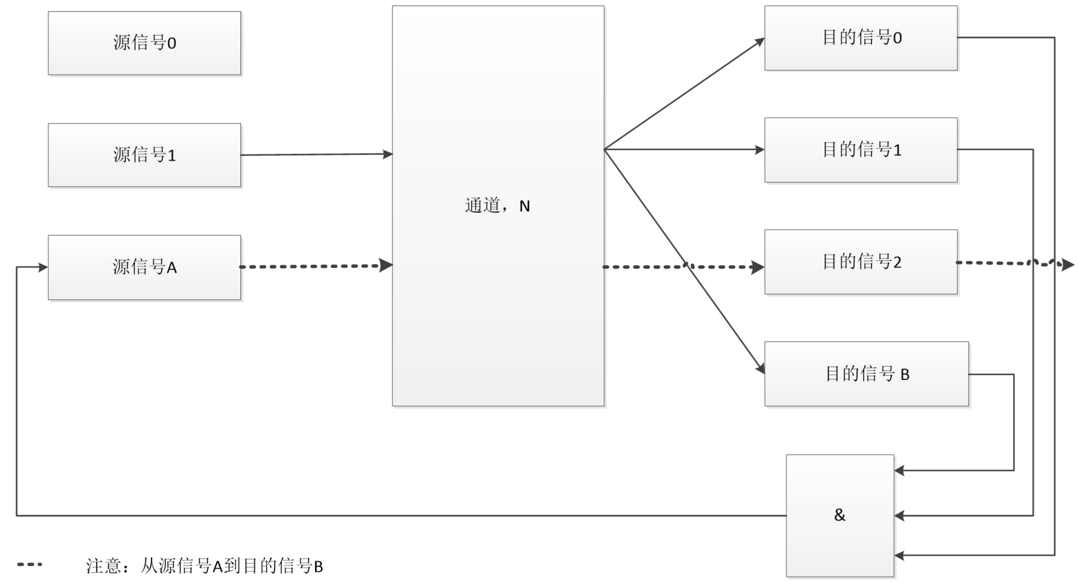


图 11-2 一对多结构图

Copyright ©2020, www.tai-action.com. All Rights Reserved.

表 11-1 源和目标触发器

NO.	源信号	目的信号	说明
0	IIR0	IIR0	寄存器可以在每个触发器中控制启动哪个通道滤波器。
1	IIR1	IIR1	寄存器可以在每个触发器中控制启动哪个通道滤波器。
2	IIR2	IIR2	寄存器可以在每个触发器中控制启动哪个通道滤波器。
3	ARCTAN0	ARCTAN0	
4	ARCTAN1	ARCTAN1	
5	ARCTAN2	ARCTAN2	
6	DFTRANS0	DFTRANS0	
7	DFTRANS1	DFTRANS1	
8	DFTRANS2	DFTRANS2	
9	MATRIX0	MATRIX0	
10	MATRIX1	MATRIX1	
11	MATRIX2	MATRIX2	
12	MATRIX3	MATRIX3	
13	RMS0	RMS0	
14	RMS1	RMS1	
15	RMS2	RMS2	
16	SINCOS0	SINCOS0	
17	SINCOS1	SINCOS1	
18	DATA DMA	DATA DMA	
19	SARADC0		
20	SARADC1		
21	SARADC2		
22	SARADC3		
23	SARADC4		
24	SARADC5		
25	SARADC6		
26	SARADC7		
27	SARADC8		
28	SARADC9		
29	SARADC10		
30	SARADC11		
31	SARADC12		
32	SARADC13		

Copyright ©2020, www.tai-action.com. All Rights Reserved.

11.2 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）。

11.3 寄存器

注意：EVSYS 基地址为 0x40032000。

名称	偏移量	大小(x32)	说明
CH_ENA	0x00	1	通道使能控制。
CH_INT_ENA	0x04	1	通道中断使能控制。
CPU_KST	0x08	1	CPU 的通道启动控制。
CH_PND_CLR	0x10	1	通道 Pending 的清除控制。
CH_CPU_PND	0x14	1	通道的 CPU Pending 标志
CH_HW_PND	0x18	1	通道的硬件 pending 标志。
CH_MODE0	0x1C	1	通道模式选择控制。
SRC_CH_CON0	0x20-0x5C	1	源通道触发信号的选择控制。
SRC_CH_CON1	0x60-0x9C	1	源通道触发信号的选择控制。
DST_CH_CON0	0xA0-0xDC	1	目的通道使能控制。

Copyright ©2020, www.tai-action.com. All Rights Reserved.

12 历史

版本	日期	描述
V1.0	2019/05/09	Official version
V2.0	2019/06/25	增加系统主从互联矩阵关系表
V2.1	2019/07/5	修改时钟、CAN 总线部分描述
V3.0	2019/07/17	增加电学模拟参数表
V3.1	2019/09/19	将 GPDMA 部分改为 DAMC
V3.2	2019/11/26	更正中断系统中断向量号，删除中断向量地址
V3.3	2020/01/02	RISC 处理器改为 Cortex-M3 核

Copyright ©2020, www.tai-action.com. All Rights Reserved.