



AD3005

32 Bit High Performance Microcontroller

用户手册

版 本 号: V1.0.0.0

版权所有©

西安恩狄集成电路有限公司

本资料内容为西安恩狄集成电路有限公司在现有数据资料基础上编制而成，本资料中所记载的实例以正确的试用方法和标准操作为前提，使用方在应用该等实例时应充分考虑外部诸条件，西安恩狄集成电路有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，西安恩狄集成电路有限公司亦不对使用方使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。文档中所有涉及到第三方软件的，请自行购买正版软件，因第三方软件版权问题涉及到的一切后果，与西安恩狄集成电路有限公司无关。基于使本资料的内容更加完善等原因，西安恩狄集成电路有限公司保留未经预告的修改权。

西安恩狄集成电路有限公司

地 址：陕西省西安市高新区高新一路 19 号思安大厦 501

电 话：+ (86 29) 88322766 网 站：www.admicrochip.com

微信号：恩狄 ADUC



版本修订记录

Bin	Version	Change List	Owner	Data
1	1.0.0.0	初版	YanJia	2021.7.8

目录

版本修订记录.....	1
1 产品特性.....	14
1.1 功能特性.....	14
1.2 系统框图.....	17
1.3 引脚定义.....	18
1.4 引脚对照表.....	19
1.5 引脚功能.....	21
2 CPU 中央处理器.....	25
2.1 存储器.....	25
2.2 系统 SysTick 计数器.....	26
2.2.1. 结构框图.....	26
2.2.2. 使用方法.....	26
2.2.3. SYSTICK 寄存器.....	27
2.3 嵌套向量中断控制器（NVIC）.....	28
2.3.1. 中断和异常向量.....	28
2.3.2. NVIC 寄存器.....	30
2.4 应用中断和复位控制（AIRC）.....	32
2.5 核心寄存器简介.....	33
3 系统控制.....	34
3.1 复位.....	34
3.1.1. POR 复位（上电复位）.....	34
3.1.2. RST 复位（外部复位）.....	35
3.1.3. LVD 复位（低电压复位/掉电复位）.....	35
3.1.4. WDT 复位（看门狗复位）.....	38
3.1.5. SW 复位（软件复位）.....	39
3.1.6. I/O 中断复位（中断唤醒）.....	39
3.2 系统时钟.....	39

3.2.1.	时钟产生框图.....	40
3.2.2.	内部时钟源.....	41
3.2.3.	外部时钟源.....	41
3.2.4.	PLL 时钟	42
3.2.5.	系统时钟（SYSCLK）选择	43
3.2.6.	CLOCK-OUT 功能.....	44
3.3	系统控制.....	44
3.3.1.	系统控制寄存器 0.....	44
3.3.2.	系统控制寄存器 1.....	50
3.4	系统操作模式.....	59
3.4.1.	正常模式（ACTIVE）	60
3.4.2.	低功耗模式（IDLE/POWER SAVE/POWER OFF）	60
3.5	系统唤醒.....	62
3.5.1.	PMU.....	63
4	GPIO 端口	71
4.1	GPIO 概述	71
4.2	GPIO 引脚电路结构图	72
4.3	GPIO 寄存器	72
4.3.1.	PA 寄存器.....	72
4.3.2.	PB 寄存器	77
4.3.3.	PC 寄存器	82
4.3.4.	PD 寄存器	87
5	通用定时器.....	92
5.1	定时器 T0.....	92
5.1.1.	定时器 T0 特性.....	92
5.1.2.	定时器 T0 电路结构.....	92
5.1.3.	定时器 T0 寄存器.....	92
5.2	定时器 T1.....	94
5.2.1.	定时器 T1 特性.....	94

5.2.2.	定时器 T1 操作模式（通过 T1CON 寄存器选择）	95
5.2.3.	定时器 T1 电路结构.....	96
5.2.4.	定时器 T1 寄存器.....	96
5.3	定时器 T2.....	98
5.3.1.	定时器 T2 特性.....	98
5.3.2.	定时器 T2 电路结构.....	98
5.3.3.	定时器 T2 寄存器.....	98
5.4	定时器 T3.....	100
5.4.1.	定时器 T3 特性.....	100
5.4.2.	定时器 T3 电路结构.....	101
5.4.3.	定时器 T3 寄存器.....	101
6	看门狗定时器.....	103
6.1	看门狗定时器（WDT）	103
6.1.1.	看门狗定时器概述.....	103
6.1.2.	看门狗定时器电路结构.....	104
6.1.3.	看门狗定时器寄存器.....	104
6.2	独立看门狗定时器（IWDT）	106
6.2.1.	独立看门狗定时器概述.....	106
6.2.2.	独立看门狗定时器寄存器.....	106
7	CCP 捕捉/比较/PWM	107
7.1	CCP 概述	107
7.2	捕捉/比较模式.....	107
7.2.1.	捕捉模式电路结构.....	108
7.2.2.	比较模式电路结构.....	109
7.2.3.	CCP 寄存器	109
7.3	PWM 模式.....	120
7.3.1.	PWM 模式概述.....	120
7.3.2.	PWM 寄存器.....	123
8	USART 通用同步/异步收发器	135

8.1	USART 概述.....	135
8.2	USART 模式.....	135
8.2.1.	异步模式.....	135
8.2.2.	同步模式.....	138
8.3	USART 寄存器定义.....	142
8.3.1.	RCSTA 接收控制寄存器.....	142
8.3.2.	TXSTA 发送控制寄存器.....	143
8.3.3.	UARTCON 控制寄存器.....	144
8.3.4.	SPBRG 波特控制寄存器.....	145
8.3.5.	TXREG 发送寄存器.....	146
8.3.6.	RCREG 接收寄存器.....	146
8.3.7.	UARTI 中断标志寄存器.....	146
8.3.8.	BRG 寄存器（波特率发生器）.....	146
9	I2C 接口.....	149
9.1	I2C 概述.....	149
9.1.1.	I2C 简介.....	149
9.1.2.	I2C 特性.....	150
9.2	I2C 从机模式.....	150
9.2.1.	I2C 主机写时的从机模式.....	151
9.2.2.	I2C 主机读时的从机模式.....	151
9.2.3.	支持广播呼叫地址.....	152
9.3	I2C 寄存器.....	153
9.3.1.	I2C 寄存器概述.....	153
9.3.2.	I2C 寄存器定义.....	154
10	SPI 接口.....	160
10.1	SPI 接口概述.....	160
10.2	SPI 接口寄存器定义.....	160
10.2.1.	CTRL（SPI 使能相关寄存器）.....	160
10.2.2.	TX_FIFO（SPI 发送数据寄存器）.....	161

10.2.3.	RX_FIFO (SPI 接收数据寄存器)	161
10.2.4.	FIFO_STA (SPI FIFO 状态寄存器)	161
10.2.5.	CLK_DIV0 (SPI 时钟分频寄存器)	162
10.2.6.	SPI_MODE (SPI 模式选择寄存器)	162
10.2.7.	SPI_CTRL (SPI 控制寄存器)	162
10.2.8.	SPI_STA (SPI 工作状态寄存器)	163
10.2.9.	INTR_EN (SPI FIFO 中断使能寄存器)	163
10.2.10.	INTR_STA (SPI FIFO 中断标志寄存器)	164
10.2.11.	RAW_INTR_STA (SPI FIFO 状态寄存器)	165
10.2.12.	INTR_MASK (SPI 中断使能寄存器)	165
10.2.13.	INTR_CLR (SPI 中断标志清除寄存器)	166
11	VREF 基准电压源	168
11.1	VREF 概述	168
11.1.1.	VREF 简介	168
11.1.2.	VREF 特性	168
11.1.3.	VREF 使用	168
11.2	VREF 寄存器	168
11.2.1.	FVRCON (FVR 控制寄存器)	168
12	RAIL TO RAIL 模拟运放	170
12.1	模拟运放概述	170
12.2	运放 OP0	170
12.2.1.	OP0 运放模式	170
12.2.2.	OP0 运放寄存器	171
12.3	运放 OP1	173
12.3.1.	OP1 运放模式	173
12.3.2.	OP1 运放寄存器	174
13	ADC 模数转换	178
13.1	ADC 模数转换概述	178
13.1.1.	ADC 模数转换结构示意图	179

13.1.2.	A/D 转换步骤	180
13.2	ADC 模数转换寄存器	181
13.2.1.	ADCCON0 控制寄存器	181
13.2.2.	ADCCON1 通道选择寄存器	182
13.2.3.	ADCCON2 使能寄存器	183
13.2.4.	ADCCON3 控制寄存器	183
13.2.5.	ADCRES 转换结果寄存器	183
14	DAC 数模转换器	185
14.1	DAC 数模转换器概述	185
14.1.1.	DAC 数模转换器结构示意图	185
14.1.2.	DAC 数模转换器特性	185
14.2	DAC 数模转换器功能介绍	186
14.2.1.	输出电压	186
14.2.2.	触发选择	186
14.2.3.	数据转换	186
14.2.4.	波形生成	187
14.2.5.	数据输出求反	188
14.2.6.	DMA 请求	189
14.3	DAC 数模转换器寄存器	189
14.3.1.	DA0 寄存器	189
14.3.2.	DA1 寄存器	191
15	CLU 逻辑配置单元	195
15.1	CLU 逻辑配置单元概述	195
15.1.1.	CLU 逻辑配置单元结构示意图	195
15.1.2.	CLU 逻辑配置单元特性	196
15.2	CLU 逻辑配置单元功能介绍	196
15.2.1.	配置顺序	196
15.2.2.	输入多路复用器输入选择	197
15.2.3.	输出配置	198

15.2.4.	LUT 配置.....	199
15.3	CLU 逻辑配置单元寄存器	200
15.3.1.	CLEN0 使能寄存器	200
15.3.2.	CLIE0 中断使能寄存器.....	200
15.3.3.	CLIF0 中断标志寄存器	201
15.3.4.	CLUT0 输出寄存器	202
15.3.5.	CLU0MX 多路复用寄存器.....	202
15.3.6.	CLU0FN 功能选择寄存器.....	202
15.3.7.	CLU0CF 配置寄存器	203
15.3.8.	CLU1MX 多路复用寄存器.....	203
15.3.9.	CLU1FN 功能选择寄存器.....	204
15.3.10.	CLU1CF 配置寄存器	204
15.3.11.	CLU2MX 多路复用寄存器.....	205
15.3.12.	CLU2FN 功能选择寄存器.....	205
15.3.13.	CLU2CF 配置寄存器	205
15.3.14.	CLU3MX 多路复用寄存器.....	206
15.3.15.	CLU3FN 功能选择寄存器.....	206
15.3.16.	CLU3CF 配置寄存器	206
15.3.17.	CLUIC0 中断清除寄存器	207
16	DMA 控制器	209
16.1	DMA 概述	209
16.1.1.	DMA 架构示意图	209
16.1.2.	DMA 特性	209
16.2	DMA 功能介绍	210
16.2.1.	通道选择.....	211
16.2.2.	DMA 控制	211
16.2.3.	周期类型.....	213
16.2.4.	错误信息.....	215
16.2.5.	通道控制数据结构配置.....	215

16.3	DMA 寄存器	217
16.3.1.	DMA_STATUS 状态寄存器.....	217
16.3.2.	DMA_CONFIG 配置寄存器.....	218
16.3.3.	DMA_CTRLBASE 通道控制数据库指针寄存器	218
16.3.4.	DMA_ALTCTRLBASE 通道备用控制数据库指针寄存器.....	218
16.3.5.	DMA_CHWAITSTATUS 通道等待请求状态寄存器	218
16.3.6.	DMA_CHSWREQ 通道软件请求寄存器.....	219
16.3.7.	DMA_CHUSEBURSTS 设置通道突发寄存器	219
16.3.8.	DMA_CHUSEBURSTC 清除通道突发寄存器.....	219
16.3.9.	DMA_CHREQMASKS 设置通道掩码请求寄存器.....	220
16.3.10.	DMA_CHREQMASKC 清除通道掩码请求寄存器	220
16.3.11.	DMA_CHENS 设置通道使能寄存器	220
16.3.12.	DMA_CHENC 清除通道使能寄存器	221
16.3.13.	DMA_CHALTS 设置通道备用寄存器	221
16.3.14.	DMA_CHALTC 清除通道备用寄存器	221
16.3.15.	DMA_CHPRIS 设置通道优先级寄存器.....	221
16.3.16.	DMA_CHPRIC 清除通道优先级寄存器	222
16.3.17.	DMA_ERRORC 清除总线错误寄存器	222
16.3.18.	DMA_CHREQSTATUS 通道请求状态寄存器	222
16.3.19.	DMA_CHSREQSTATUS 单通道请求状态寄存器	223
16.3.20.	DMA_IF 中断标志寄存器.....	223
16.3.21.	DMA_IFS 中断标志配置寄存器.....	223
16.3.22.	DMA_IFC 清除中断标志寄存器.....	224
16.3.23.	DMA_IEN 中断使能寄存器.....	224
16.3.24.	DMA_CH0_CTRL 通道控制寄存器.....	224
16.3.25.	DMA_CH1_CTRL 通道控制寄存器.....	224
16.3.26.	DMA_CH2_CTRL 通道控制寄存器.....	225
16.3.27.	DMA_CH3_CTRL 通道控制寄存器.....	225
16.3.28.	SOURCESEL 触发源选择寄存器	225

17	HDIV 硬件除法器	227
17.1	HDIV 概述	227
17.1.1.	HDIV 特性	227
17.1.2.	HDIV 功能描述	227
17.1.3.	HDIV 运行流程图	228
17.2	HDIV 寄存器	228
17.2.1.	HDIV_DIVDR 被除数寄存器	228
17.2.2.	HDIV_DIVDSR 除数寄存器	228
17.2.3.	HDIV_DIVQR 商寄存器	228
17.2.4.	HDIV_DIVRR 余数寄存器	229
17.2.5.	HDIV_DIVSTAT 除法器状态寄存器	229
18	FLASH 可编程存储器（FMC）	230
18.1	FLASH 概述	230
18.1.1.	嵌入式 FLASH 存储器	230
18.1.2.	FLASH 特性	230
18.2	FLASH 功能描述	231
18.2.1.	扇区	231
18.2.2.	读操作	231
18.2.3.	编程/擦除	231
18.2.4.	嵌入式引导加载程序	231
18.2.5.	FLASH 存储控制器（FMC）	232
18.2.6.	编程 FLASH 存储器	232
18.2.7.	擦除操作	232
18.2.8.	读保护	233
18.2.9.	硬件 CHECKSUM	233
18.3	FLASH 寄存器	233
18.3.1.	CFG 配置寄存器	233
18.3.2.	STATUS 状态寄存器	234
18.3.3.	CTRL 控制寄存器	234

18.3.4.	DATA 数据寄存器	235
18.3.5.	ADDR 地址寄存器	235
18.3.6.	CHKSUM 校验寄存器	235
18.3.7.	CRC0 寄存器	235
18.3.8.	CRC1 寄存器	236
19	ICACHE 指令高速缓冲存储器 (CMC)	237
19.1	ICACHE 概述	237
19.2	ICACHE 寄存器定义	237
19.2.1.	CFG 配置控制寄存器	237
20	CONFIG FUSE 加密功能	238
20.1	CONFIG FUSE 概述	238
20.2	CONFIG FUSE 寄存器定义	238
20.2.1.	FUSE0 配置寄存器	238
21	SWD 调试	240
21.1	SWD 概述	240
21.1.1.	SWD 特性	240
21.2	SWD 引脚说明	240
21.3	SWD 调试注意事项	240
21.3.1.	局限性	240
21.3.2.	恢复调试功能	240
21.3.3.	SWD 引脚上的内部上拉/下拉电阻	241
22	FSCM 时钟监测	242
22.1	自动防故障装置时钟监控	242
22.2	自动防故障装置检测	242
22.3	故障安全操作	243
22.4	故障安全条件清除	243
22.5	重置或从睡眠中唤醒	243
22.6	FSCM 寄存器	243
23	TSensor 温度传感器	245

23.1	TSENSOR 寄存器定义.....	245
23.1.1.	TSCFG1 配置寄存器	245
23.1.2.	TSCFG2 配置寄存器	245
23.1.3.	TSCFG3 配置寄存器	245
24	电气参数.....	247
24.1	直流电气特性.....	247
24.2	交流电气特性.....	248
24.3	LVR 电气特性.....	248
24.4	LDO 电气特性（内核电压）	248
24.5	BGR 电气特性.....	249
24.6	Recommended Operating Conditions.....	249
24.7	Power Consumption.....	249
24.8	Reset and Supply Monitor.....	251
24.9	Flash Memory	252
24.10	Power Management Timing.....	252
24.11	Internal oscillators.....	252
24.12	External Clock Input	253
24.13	ADC.....	253
24.14	Voltage Reference	254
24.15	Temperature Sensor.....	255
24.16	DACS.....	255
24.17	Configurable Logic.....	256
24.18	Port I/O.....	256
24.19	Thermal Conditions.....	257
24.20	Absolute Maximum Ratings	258
25	封装信息.....	259
25.1	QFN32 封装图	259
25.2	QFN24 封装图	261
25.3	QFN20 封装图	263

26	订购信息.....	264
----	-----------	-----

1 产品特性

1.1 功能特性

✧ Cortex-M0 内核 MCU，高达 64MHz 稳定工作频率

✧ F_{CPU}（指令周期）用于外设时钟驱动前，支持最高 128 倍的预分频

✧ 64KB 的 FLASH 可编程存储器和 8KB 的 SRAM 存储器

✧ 35 个中断源

内建嵌套向量中断控制器（NVIC）

支持 I/O/TIMER/CCP/UART/I2C/SPI/ADC/DAC 等多种中断源和各类触发方式

✧ 30 个多功能 I/O 接口

I/O 可配置为上拉/下拉/开漏/模拟输入输出/数字输入输出等功能

I/O 可响应边沿或电平中断源触发

✧ 2 路可编程/不可编程看门狗定时器

WDT：可编程看门狗定时器

IWDT：独立不可编程看门狗定时器

✧ 24 位的 SysTick 系统滴答定时器

✧ 支持不同阈值电压的低电压复位（LVD）方式

✧ 3 路通用定时器

1 个 24 位定时器 0，支持向上计数/自动重载模式

1 个 32 位定时器 1，支持向上计数/单脉冲模式/周期模式/反转输出模式/连续计数模式

2 个 16 位定时器 2，支持向上计数/自动重载模式

✧ CCP 模块支持 Capture/Compare/PWM 工作模式

支持 30 路输入捕捉，同时支持 6 路同时捕捉

支持 6 路同时输出比较

支持 12 路互补的 16 位精度带死区控制的桥接 PWM 输出，每路 PWM 可以设定延时启动的 ADC 功能，ADC 采样后可保存各自通道数据

✧ 支持 UART 通讯功能

✧ 1 路从机模式的硬件 I2C 接口

7/10 位地址长度

可响应 4 个从地址，屏蔽 1 个从地址

支持高速 400KHz 和低速 100KHz 的通讯速度

✧ 1 路主从模式的硬件 SPI 接口

支持 4~16bit 数据宽度，4 级缓存深度

✧ 4 个可配置的 CLU 逻辑单元

每个单元支持 256 种不同的逻辑组合功能

支持同步和异步操作

✧ 1 个硬件除法器模块

支持 32 位有符号/无符号硬件除法

✧ DMA 控制器

支持 RAM/FLASH 到外设，RAM 到 FLASH，外设到 RAM，RAM/FLASH 到 RAM 传输

支持 1 路 DMA 通道，支持多路启动 DMA 通道选择

✧ 2 路高性能轨到轨运放器

支持 1/8/16/24/32/40/48/56 倍信号放大

支持比较器模式

✧ 1 路 12 位的 ADC 转换接口

ADC 转换接口最高支持 30 个不同通道的输入，高达 900ksps 的转换频率

✧ 2 路 12 位的 DAC 转换接口

支持生成递增/递减/三角波形

支持多路输入触发选择

✧ 4 种 MCU 工作模式

ACTIVE/IDLE/POWER SAVE/POWER OFF

✧ 1 个高精度 ($\pm 1^{\circ}\text{C}$) 温度传感器

支持高温报警

✧ 3 种系统时钟源选择

外部高速时钟：晶体模式，4MHz~16MHz

内部高速时钟：RC 模式，32MHz

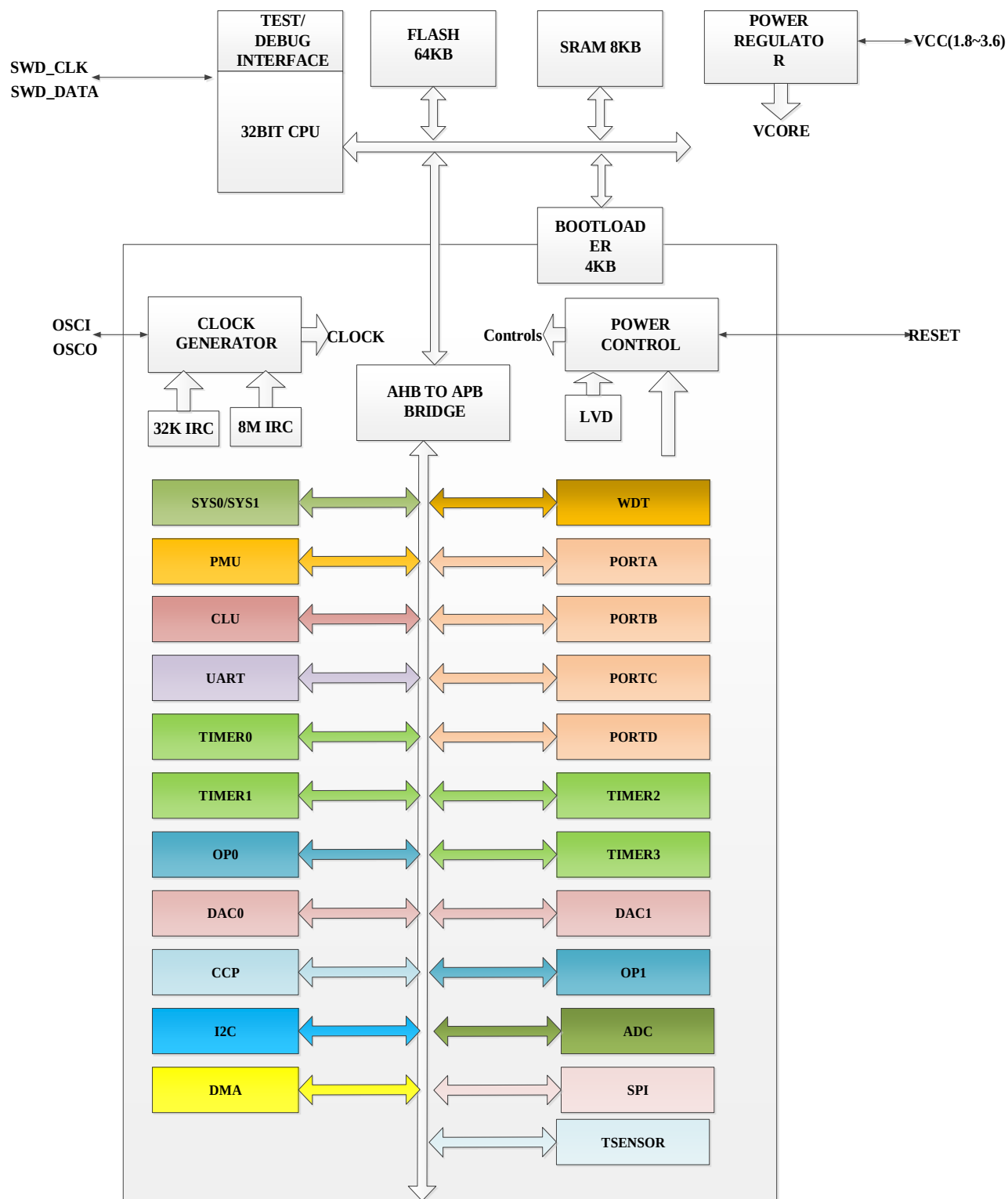
内部低速时钟：RC 模式，256/512kHz（可配置）

PLL 无需高频率的晶振就可使系统达到最大运行速率，最高支持 64MHz

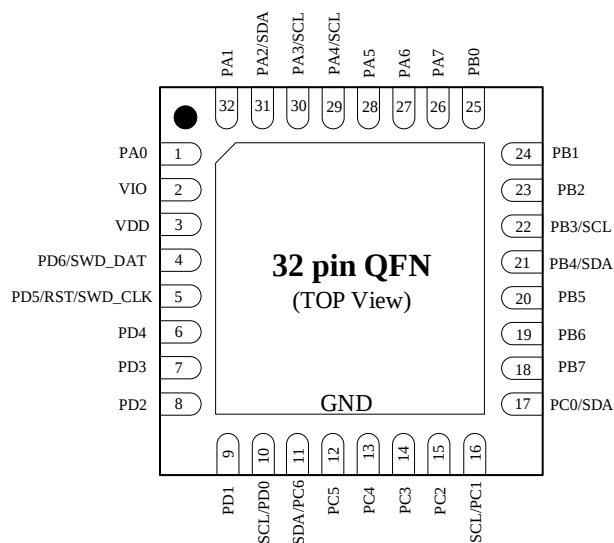
可选时钟输出功能（CKOE），使能输出系统时钟到 MCU 的 I/O 引脚

- ✧ 支持串行调试（SWD）
- ✧ 支持外部晶振时钟监测和恢复
- ✧ 支持在线烧录（ICP）
- ✧ 支持 I2C BootLoader，单双线 Uart BootLoader（IIC 和 Uart 二选一）
- ✧ 工作电压：2.0V~3.6V
- ✧ 封装形式：QFN32/ QFN24/ QFN20

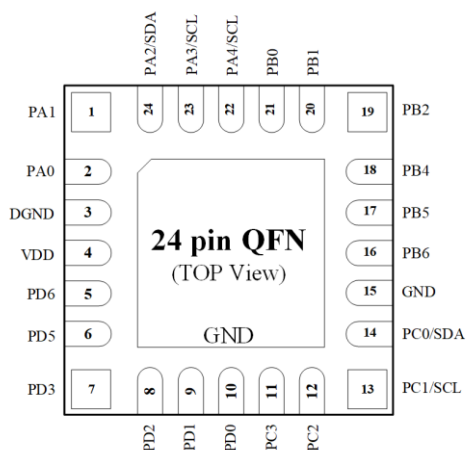
1.2 系统框图



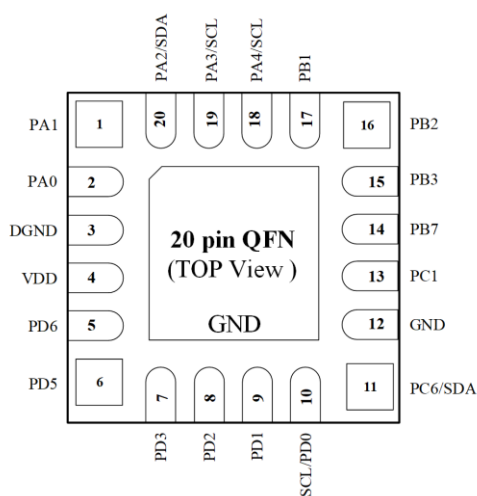
1.3 引脚定义



QFN32 Pinout



QFN24 Pinout



QFN20 Pinout

1.4 引脚对照表

Num	Pin name	QFN32	QFN24	QFN20	Description
1	VDD	3	4	4	Supply Power Input
2	VIO	2	–	–	I/O Supply Power Input
3	DGND	–	3	3	Ground
4	GND	–	15	12	Ground
5	PA0	1	2	2	IO/SCL/PWM00/TOCKI/CCP0OUT/CAPCH0/ VREF2P4/AN0
6	PA1	32	1	1	IO/S_SDI/M_SDI/PWM10/PWM01/T1CKI/CCP1OUT/ CAPCH1/ADC0.0/OP0P.0/OP0N.0/VREF2P4/AN1
7	PA2	31	24	20	IO/SDA/PWM20/PWM10/T10/CCP2OUT/ CLU0OUT/CAPCH2/ADC0.1/OP0P.1/OP0N.1/AN2
8	PA3	30	23	19	IO/SCL/CKOE/AN3/OSCO
9	PA4	29	22	18	IO/SCL/TX/TX_RX/PWM40/PWM20/CK/CCP4OUT/ CAPCH4/ADC0.2/OP0P.2/OP0N.2/OP1OUT0/AN4/OSCI
10	PA5	28	–	–	IO/T2CKI/RX/PWM50/PWM21/DT/CCP5OUT/ CAPCH5/ADC0.3/OP0P.3/OP0N.3/AN5
11	PA6	27	–	–	IO/T3CKI/BCLK/PWM21/PWM30/T10/ CAPCH6/ADC0.4/OP0P.4/OP0N.4/AN6
12	PA7	26	–	–	IO/S_SDI/M_SDI/PWM31/T10/CAPCH7/ ADC0.5/OP0P.5/OP0N.5/OP1P.0/OP1N.0/FVR0UT/AN7
13	PB0	25	21	–	IO/TX/RX/PWM00/PWM40/TOCKI/CCP0OUT/CLU1OUT/ CAPCH8/ADC0.6/OP0P.6/OP0N.6/OP1P.1/OP1N.1/AN8
14	PB1	24	20	17	IO/RX/TX/PWM10/PWM41/T1CKI/CCP1OUT/ CAPCH9/ADC0.7/OP0P.7/OP0N.7/AN9
15	PB2	23	19	16	IO/T2CKI/CKOE/PWM20/PWM50/CCP2OUT/ CAPCH10/ADC0.8/OP0P.8/OP0N.8/OP1OUT1/AN10
16	PB3	22	–	15	IO/SCL/TX/PWM30/PWM51/CK/CCP3OUT/ TX_RX/CAPCH11/ADC0.9/AN11
17	PB4	21	18	–	IO/SDA/RX/PWM40/PWM00/DT/CCP4OUT/ CAPCH12/ADC0.10/OP0OUT3/AN12
18	PB5	20	17	–	IO/T3CKI/BCLK/PWM50/PWM01/CCP5OUT/ CLU2OUT/CAPCH13/ADC0.11/OP1OUT2/AN13
19	PB6	19	16	–	IO/SSN/PWM41/PWM10/CLU3OUT/CAPCH14/ ADC0.12/OP1OUT3/AN14
20	PB7	18	–	14	IO/PWM51/PWM11/CAPCH15/ADC0.13/OP0P.9/AN15
21	PC0	17	14	–	IO/SDA/PWM00/PWM20/CCP0OUT/ CAPCH16/OP1P.2/OP1N.2/AN16

22	PC1	16	13	13	IO/SCL/PWM10/PWM21/CCP1OUT/CAPCH17/ ADC0. 14/OP0OUT0/OP1P. 3/OP1N. 3/AN17
23	PC2	15	12	–	IO/T0CKI/CKOE/PWM20/PWM30/CCP2OUT/ CLU2OUT/CAPCH18/ADC0. 15/OP1P. 4/OP1N. 4/AN18
24	PC3	14	11	–	IO/T1CKI/TX/PWM30/PWM31/CK/CCP3OUT/TX_RX/ CAPCH19/ADC0. 16/OP0OUT1/OP1P. 5/OP1N. 5/AN19
25	PC4	13	–	–	IO/T2CKI/RX/PWM11/PWM40/DT/CCP4OUT/ CAPCH20/ADC0. 17/OP1P. 6/OP1N. 6/AN20
26	PC5	12	–	–	IO/T3CKI/BCLK/PWM01/PWM41/CCP5OUT/CLU3OUT/ CAPCH21/ADC0. 18/OP0OUT2/OP1P. 7/OP1N. 7/AN21
27	PC6	11	–	11	IO/SDA/CAPCH22/ADC0. 19/OP1P. 8/OP1N. 8/AN22
28	PD0	10	10	10	IO/SCL/T0CKI/PWM01/PWM50/CAPCH23/FVR. 4/DAC0/AN23
29	PD1	9	9	9	IO/T1CKI/PWM11/PWM51/CAPCH24/FVR. 5/DAC1/AN24
30	PD2	8	8	8	IO/T2CKI/PWM21/CAPCH25/FVR. 6/AN25
31	PD3	7	7	7	IO/T3CKI/CKOE/PWM31/CAPCH26/FVR. 7/AN26
32	PD4	6	–	–	IO/TX/CK/TX_RX/CAPCH27/AN27
33	PD5	5	6	6	IO/RX/DT/CAPCH28/AN28/RST/SWD_CLK
34	PD6	4	5	5	IO/BCLK/CAPCH29/AN29/SWD_DAT

1.5 引脚功能

Pin Number			PinName	DIG		ANA	
QFN				function	Description	function	Description
32PIN	24PIN	20PIN					
1	2	2	PA0	IO	IO	VREF2.4	2.4V基准电压
				SCL	I2C的时钟	AN0	模拟通道
				SPI_SCK	SPI的时钟		
				PWM00	PWM的输出		
				T0CKI	TIMER0的外部时钟输入		
				CCP0OUT	CCP0的输出		
			CAPCH0	CAP的捕捉通道			
2	-	-	VIO		I/O Supply Power Input		
3	4	4	VDD		Supply Power Input		
4	5	5	PD6	IO	IO	AN29	模拟通道
				BCLK	UART的外部时钟	SWD_DAT	SWD的数据脚
				CAPCH29	CAP的捕捉通道		
5	6	6	PD5	IO	IO	AN28	模拟通道
				RX	UART接收数据引脚	RST	复位引脚
				DT	SUART的数据引脚	SWD_CLK	SWD的时钟引脚
				CAPCH28	CAP的捕捉通道		
6	-	-	PD4	IO	IO	AN27	模拟通道
				TX	UART发送数据引脚		
				CK	SUART的时钟引脚		
				TX_RX	UART的半双工通讯引脚		
				CAPCH27	CAP的捕捉通道		
7	7	7	PD3	IO	IO	FVR.7	FVR的电压选择通道
				T3CKI	TIMER3的外部时钟输入	AN26	模拟通道
				CKOE	系统时钟的输出引脚		
				PWM31	PWM的输出		
				CAPCH26	CAP的捕捉通道		
8	8	8	PD2	IO	IO	FVR.6	FVR的电压选择通道
				T2CKI	TIMER2的外部时钟输入	AN25	模拟通道
				PWM21	PWM的输出		
				CAPCH25	CAP的捕捉通道		
9	9	9	PD1	IO	IO	FVR.5	FVR的电压选择通道
				T1CKI	TIMER1的外部时钟输入	DAC1	DAC的输出
				PWM11	PWM的输出	AN24	模拟通道
				PWM51	PWM的输出		
				CAPCH24	CAP的捕捉通道		

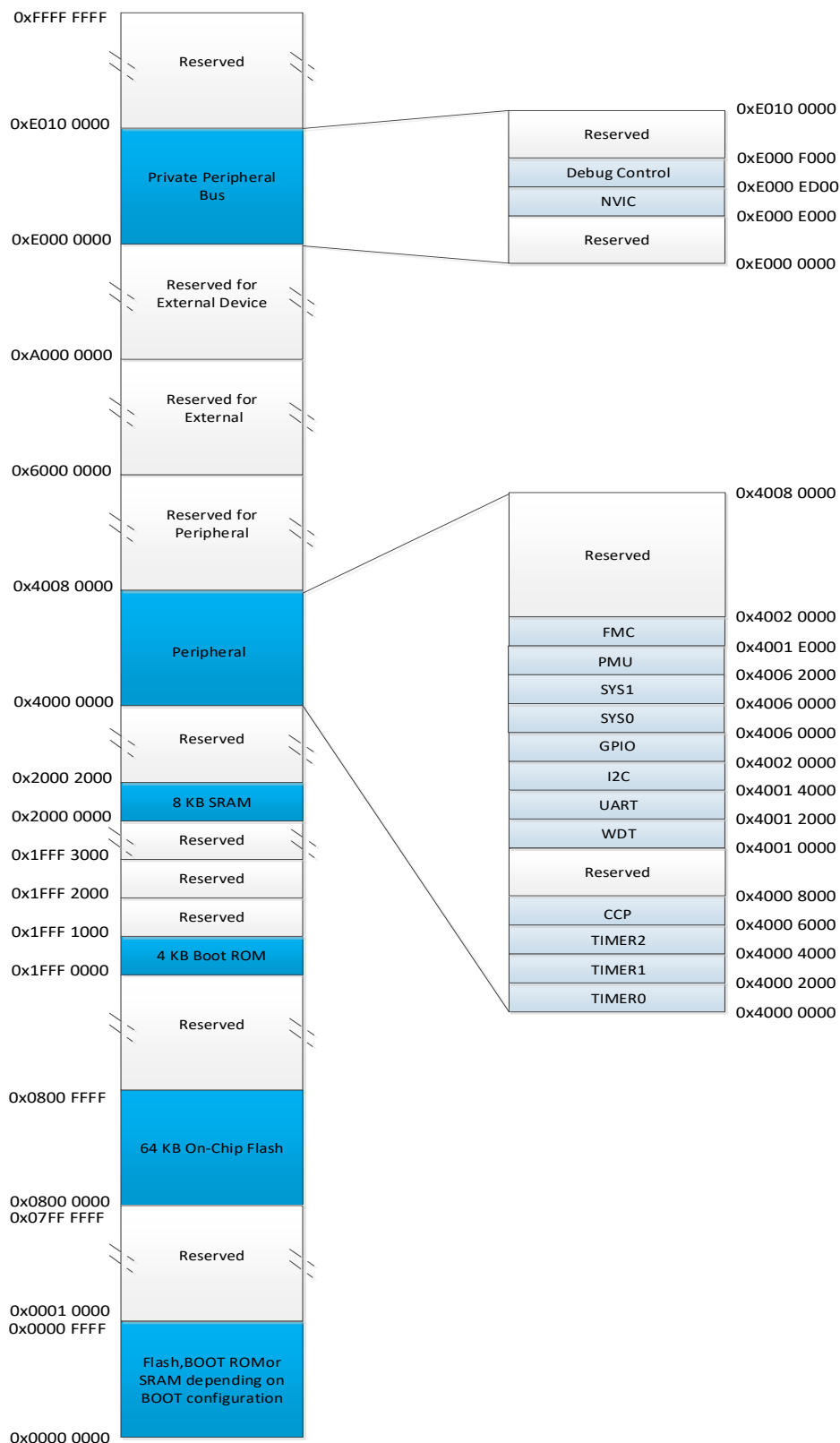
10	10	10	PD0	IO	IO	FVR.4	FVR的电压选择通道
				SCL	I2C的时钟	DAC0	DAC的输出
				T0CKI	TIMER0的外部时钟输入	AN23	模拟通道
				PWM01	PWM的输出		
				PWM50	PWM的输出		
11	-	11	PC6	CAPCH23	CAP的捕捉通道		
				IO	IO	OP1P.8	OP1的正端输入
				SDA	I2C的数据引脚	OP1N.8	OP1的负端输入
				CAPCH22	CAP的捕捉通道	AN22	模拟通道
12	-	-	PC5			ADC0.19	ADC的模拟通道
				IO	IO		
				T3CKI	TIMER3的外部时钟输入	ADC0.18	ADC的模拟通道
				BCLK	UART的外部时钟	OP0OUT2	OP0的输出
				PWM01	PWM的输出	OP1P.7	OP1的正端输入
				PWM41	PWM的输出	OP1N.7	OP1的负端输入
				CCP5OUT	CCP5的输出	AN21	模拟通道
				CLU3OUT	CLU3的输出		
				CAPCH21	CAP的捕捉通道		
13	-	-	PC4	IO	IO	ADC0.17	ADC的模拟通道
				T2CKI	TIMER2的外部时钟输入	OP1P.6	OP1的正端输入
				RX	UART的数据引脚	OP1N.6	OP1的负端输入
				PWM11	PWM的输出	AN20	模拟通道
				PWM40	PWM的输出		
				DT	SUART的数据引脚		
				CCP4OUT	CCP4的输出		
14	11	-	PC3	CAPCH20	CAP的捕捉通道		
				IO	IO	OP1P.5	OP1的正端输入
				T1CKI	TIMER1的外部时钟输入	OP1N.5	OP1的负端输入
				TX	UART的发送数据引脚	AN19	模拟通道
				PWM30	PWM的输出	ADC0.16	ADC的模拟通道
				PWM31	PWM的输出	OP0OUT1	OP0的输出
				CK	SUART的时钟引脚		
				CCP3OUT	CCP的输出		
				TX_RX	UART的半双工通讯引脚		
15	12	-	PC2	CAPCH19	CAP的捕捉通道		
				IO	IO	ADC0.15	ADC的模拟通道
				T0CKI	TIMER0的外部时钟输入	OP1P.4	OP1的正端输入
				CKOE	系统时钟的输出引脚	OP1N.4	OP1的负端输入
				PWM20	PWM的输出	AN18	模拟通道
				PWM30	PWM的输出		
				CCP2OUT	CCP2的输出		
				CLU2OUT	CLU2的输出		
16	13	13	PC1	CAPCH18	CAP的捕捉通道		
				IO	IO	OP1P.3	OP1的正端输入
				SCL	I2C的时钟输入引脚	OP1N.3	OP1的负端输入
				PWM10	PWM的输出	AN17	模拟通道
				PWM21	PWM的输出	ADC0.14	ADC的模拟通道
				CCP1OUT	CCP1的输出	OP0OUT0	OP0的输出
17	14	-	PC0	CAPCH17	CAP的捕捉通道		
				IO	IO	OP1P.2	OP1的正端输入
				SDA	I2C的数据引脚	OP1N.2	OP1的负端输入
				PWM00	PWM的输出	AN16	模拟通道
				PWM20	PWM的输出		
				CCP0OUT	CCP0的输出		
18	-	14	PB7	CAPCH16	CAP的捕捉通道		
				IO	IO	OP0P.9	OP0的正端输入
				PWM51	PWM的输出	AN15	模拟通道
				PWM11	PWM的输出	ADC0.13	ADC的模拟通道
19	16	-	PB6	CAPCH15	CAP的捕捉通道		
				IO	IO	ADC0.12	ADC的模拟通道
				SPI_SSN	SPI的片选	OP1OUT3	OP1的输出
				PWM41	PWM的输出	AN14	模拟通道
				PWM10	PWM的输出		
				CLU3OUT	CLU3的输出		
20	17	-	PB5	CAPCH14	CAP的捕捉通道		
				IO	IO	ADC0.11	ADC的模拟通道
				T3CKI	TIMER3的外部时钟输入	OP1OUT2	OP1的输出
				BCLK	UART的外部时钟	AN13	模拟通道
				PWM50	PWM的输出		

20	17	-	PB5	PWM01	PWM的输出	ADC0.10		ADC的模拟通道
				CCP5OUT	CCP5的输出			
				CLU2OUT	CLU2的输出			
				CAPCH13	CAP的捕捉通道			
21	18	-	PB4	IO	IO	OP0OUT3	OP0的输出	
				SDA	I2C的数据引脚	AN12	模拟通道	
				RX	UART接收数据引脚			
				PWM40	PWM的输出			
				PWM00	PWM的输出			
				DT	SUART的数据引脚			
				CCP4OUT	CCP4的输出			
				CAPCH12	CAP的捕捉通道			
22	-	15	PB3	IO	IO	ADC0.9	ADC的模拟通道	
				SCL	I2C的时钟引脚	AN11	模拟通道	
				TX	UART的发送数据引脚			
				PWM30	PWM的输出			
				PWM51	PWM的输出			
				CK	SUART的时钟引脚			
				CCP3OUT	CCP3的输出			
				TX_RX	UART的半双工通讯引脚			
CAPCH11	CAP的捕捉通道							
23	19	16	PB2	IO	IO	ADC0.8	ADC的模拟通道	
				T2CKI	TIMER2的外部时钟输入	OP0P.8	OP0的正端输入	
				CKOE	系统时钟的输出引脚	OP0N.8	OP0的负端输入	
				PWM20	PWM的输出	OP1OUT1	OP1的输出	
				PWM50	PWM的输出	AN10	模拟通道	
				CCP2OUT	CCP2的输出			
CAPCH10	CAP的捕捉通道							
24	20	17	PB1	IO	IO	ADC0.7	ADC的模拟通道	
				RX	UART的接收数据引脚	OP0P.7	OP0的正端输入	
				TX	UART的发送数据引脚	OP0N.7	OP0的负端输入	
				PWM10	PWM的输出	AN9	模拟通道	
				PWM41	PWM的输出			
				T1CKI	TIMER1的外部时钟输入			
				CCP1OUT	CCP1的输出			
CAPCH9	CAP的捕捉通道							
25	21	-	PB0	IO	IO	ADC0.6	ADC的模拟通道	
				TX	UART的发送数据引脚	OP0P.6	OP0的正端输入	
				RX	UART的接收数据引脚	OP0N.6	OP0的负端输入	
				PWM00	PWM的输出	OP1P.1	OP1的正端输入	
				PWM40	PWM的输出	OP1N.1	OP1的负端输入	
				T0CKI	TIMER0的外部时钟输入	AN8	模拟通道	
				CCP0OUT	CCP0的输出			
				CLU1OUT	CLU1的输出			
CAPCH8	CAP的捕捉通道							
26	-	-	PA7	IO	IO	ADC0.5	ADC的模拟通道	
				SPIS_SDO	SPI从机的数据输出引脚	OP0P.5	OP0的正端输入	
				SPIM_SDI	SPI主机的数据输入引脚	OP0N.5	OP0的负端输入	
				PWM31	PWM的输出	OP1P.0	OP1的正端输入	
				T1O	TIMER1的输出信号	OP1N.0	OP1的负端输入	
				CAPCH7	CAP的捕捉通道	FVROUT	FVR的输出	
		AN7	模拟通道					
27	-	-	PA6	IO	IO	ADC0.4	ADC的模拟通道	
				T3CKI	TIMER3的外部时钟输入	OP0P.4	OP0的正端输入	
				BCLK	UART的外部时钟	OP0N.4	OP0的负端输入	
				PWM21	PWM的输出	AN6	模拟通道	
				PWM30	PWM的输出			
				T1O	TIMER1的输出信号			
CAPCH6	CAP的捕捉通道							
28	-	-	PA5	IO	IO	ADC0.3	ADC的模拟通道	
				T2CKI	TIMER2的外部时钟输入	OP0P.3	OP0的正端输入	
				RX	UART的接收数据引脚	OP0N.3	OP0的负端输入	
				PWM50	PWM的输出	AN5	模拟通道	
				PWM21	PWM的输出			
				DT	SUART的数据引脚			
				CCP5OUT	CCP5的输出			
CAPCH5	CAP的捕捉通道							

29	22	18	PA4	IO	IO	ADC0.2	ADC的模拟通道
				SCL	I2C的时钟引脚	OP0P.2	OP0的正端输入
				TX	UART的发送数据引脚	OP0N.2	OP0的负端输入
				PWM40	PWM的输出	OP1OUT0	OP1的输出
				PWM20	PWM的输出	AN4	模拟通道
				CK	SUART的时钟引脚	OSCI	外部晶振的输入
				CCP4OUT	CCP4的输出		
				TX_RX	UART的半双工通讯引脚		
30	23	19	PA3	CAPCH4	CAP的捕捉通道		
				IO	IO	AN3	模拟通道
				SCL	I2C的时钟引脚	OSCO	外部晶振的输出
31	24	20	PA2	CKOE	系统时钟的输出引脚		
				IO	IO	ADC0.1	ADC的模拟通道
				SDA	I2C的数据引脚	OP0P.1	OP0的正端输入
				PWM20	PWM的输出	OP0N.1	OP0的负端输入
				PWM10	PWM的输出	AN2	模拟通道
				T1O	TIMER1反转模式输出信号		
				CCP2OUT	CCP2的输出		
				CLU0OUT	CLU0的输出		
32	1	1	PA1	CAPCH2	CAP的捕捉通道		
				IO	IO	ADC0.0	ADC的模拟通道
				SPIS_SDI	SPI从机的数据输入引脚	OP0P.0	OP0的正端输入
				SPIM_SDO	SPI主机的数据输出引脚	OP0N.0	OP0的负端输入
				PWM10	PWM的输出	VREF2P4	2.4V基准电压
				PWM01	PWM的输出	AN1	模拟通道
				T1CKI	TIMER1的外部时钟输入		
				CCP1OUT	CCP1的输出		
				CAPCH1	CAP的捕捉通道		

2 CPU 中央处理器

2.1 存储器

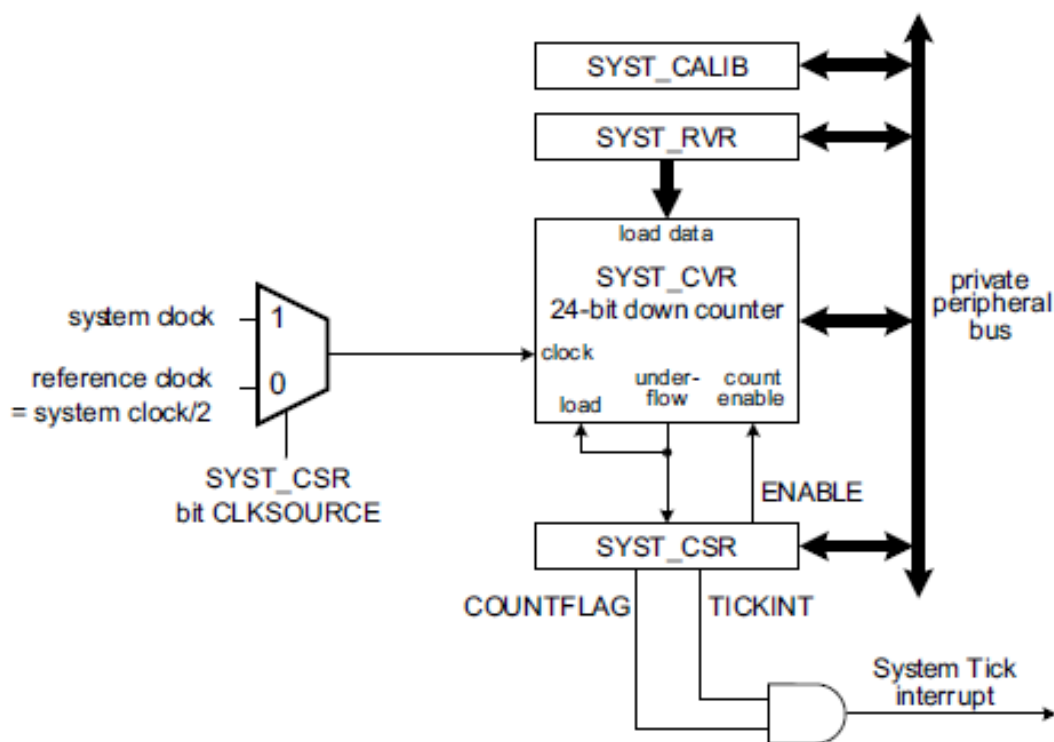


2.2 系统 SysTick 计数器

SysTick 是一个 24 位的系统节拍定时器，是嵌入在 Cortex-M0 内核 NVIC 中，具有自动重载和溢出中断功能的模块，用于操作系统或者其他系统软件管理，方便进行 Cortex-M0 设备的应用开发。

详情请参考 Cortex-M0 用户指南。

2.2.1. 结构框图



SysTick 结构图

2.2.2. 使用方法

SysTick 定时器是一个 24 位递减计数定时器，递减计数至 0 时产生系统中断。

SysTick 定时器由控制寄存器控制使能，时钟频率固定为系统时钟，可以提供固定的 10ms 中断。

SysTick 定时器通过中断控制器时钟更新。

当通过一些操作进入低功耗模式时 SysTick 时钟信号停止，此时 SysTick 定时器停止计数。

当使能 SysTick 定时器，定时器从当前数值 (SYST_CVR) 开始递减计数至 0，在下一个

时钟沿将重新载入 SYST_RVR 的值，然后在随后的时钟里递减计数。当计数器计数至 0 时，COUNTFLAG 状态位置 1，读取后该位清零。

注：当处理器因调试而停止时，计数器不会进行递减计数。

软件务必使用整字（word）方式操作 SysTick 寄存器。

SysTick 计数器重载数据和当前数据不会由硬件来初始化。SysTick 计数器正确的初始化流程如下：

1. 编程重载值到 SYSTICK_LOAD 寄存器中；
2. 可写入任意值到 SYSTICK_VAL 寄存器；
3. 设置控制和状态（SYSTICK_CTRL）寄存器。

2.2.3. SYSTICK 寄存器

2.2.3.1. 系统定时器控制和状态寄存器（SYSTICK_CTRL）

基地址：0xE000 E010

Bit	Name	Description	Attribute	Reset
31:17	Reserved		R	0
16	COUNTFLAG	SysTick 计数器递减计数到 0 时，该位置 1，被读取后清零。	R/W	0
15:3	Reserved		R	0
2	CLKSOURCE	选择 SysTick 定时器时钟源 1：固定为系统时钟；	R	1
1	TICKINT	SysTick 中断使能位 0：禁止 SysTick 中断； 1：使能 SysTick 中断； 注：当 SysTick 计数器递减到 0 时产生中断。	R/W	0
0	ENABLE	SysTick 中断使能位 0：禁止； 1：使能；	R/W	0

2.2.3.2. 系统定时器重载值寄存器（SYSTICK_LOAD）

重载寄存器需设置当 SysTick 递减计数到 0 时，重新装载的数据。此寄存器由软件来初始化。如果系统或外部时钟运行在预期的频率，SYSTICK_CALIB 寄存器的值可以使用作为重载寄存器的值。

$$RELOAD = (\text{SysTick 时钟频率} \times 10\text{ms}) - 1$$

下面的示例说明选择 SysTick 定时器重装值，如获得 10ms 的间隔时间，系统时钟设置为 48MHz。

$$RELOAD = (\text{SysTick时钟频率} \times 10\text{ms}) - 1 = (48\text{MHz} \times 10\text{ms}) - 1 = 0x000752FF\text{bit}$$

基地址：0xE000 E014

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	RELOAD	使能定时器。当计数到 0 时，此值重新装入 SYSTICK_VAL 寄存器。	R/W	0x5F7F9B

2.2.3.3. 系统定时器当前值寄存器 (SYSTICK_VAL)

基地址：0xE000 E018

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	CURRENT	读取该寄存器时返回 SysTick 计数器的当前值； 写入任何数据清除 SysTick 计数器和 SYSTICK_CTRL 寄存器中的 CUNFLAG 位。	R/W	0

2.3 嵌套向量中断控制器 (NVIC)

包括内核异常在内的所有中断，都由 NVIC 来管理。NVIC 特性如下：

1. NVIC 支持 32 个向量中断；
2. 带有硬件掩膜优先级的 4 级可编程中断优先级；
3. 快速响应异常情形和中断处理；
4. 对延迟中断的有效处理；
5. 对系统控制寄存器的操作；
6. 产生软件中断。

2.3.1. 中断和异常向量

Execution No.	Priority	Function	Description	Address Offset
0	-	-	Reserved	0x0000 0000
1	-3	Reset	Reset	0x0000 0004
2	-2	NMI_Handler	非屏蔽中断	0x0000 0008
3	-1	HardFault_Handler	故障的所有种类	0x0000 000C

4~10	Reserved	Reserved	Reserved	–
11	Settable	SVCCall1	–	0x0000 002C
12~13	Reserved	Reserved	Reserved	–
14	Settable	PendSV	–	0x0000 0038
15	Settable	SysTick	–	0x0000 003C
16	Settable	IRQ0	PA_ALL	0x0000 0040
17	Settable	IRQ1	PB_ALL	0x0000 0044
18	Settable	IRQ2	PC_ALL	0x0000 0048
19	Settable	IRQ3	PD_ALL	0x0000 004C
20	Settable	IRQ4	PA0/CCP5	0x0000 0050
21	Settable	IRQ5	PA7	0x0000 0054
22	Settable	IRQ6	PA6	0x0000 0058
23	Settable	IRQ7	PA5	0x0000 005C
24	Settable	IRQ8	PA4	0x0000 0060
25	Settable	IRQ9	PA3	0x0000 0064
26	Settable	IRQ10	PA2	0x0000 0068
27	Settable	IRQ11	PA1	0x0000 006C
28	Settable	IRQ12	CCP2	0x0000 0070
29	Settable	IRQ13	CCP3	0x0000 0074
30	Settable	IRQ14	CCP4	0x0000 0078
31	Settable	IRQ15	CLU	0x0000 007C
32	Settable	IRQ16	SPI	0x0000 0080
33	Settable	IRQ17	DMA	0x0000 0084
34	Settable	IRQ18	TIMER3	0x0000 0088
35	Settable	IRQ19	ADC	0x0000 008C
36	Settable	IRQ20/CCPIRQ	CCP0	0x0000 0090
37	Settable	IRQ21/T2IRQ	TIMER2	0x0000 0094
38	Settable	IRQ22/T1IRQ	TIMER1	0x0000 0098
39	Settable	IRQ23/T0IRQ	TIMER0	0x0000 009C
40	Settable	IRQ24/UARTIRQ	UART	0x0000 00A0

41	Settable	IRQ25/SSPIRQ	I2C	0x0000 00A4
42	Settable	IRQ26/OPIRQ	OP0	0x0000 00A8
43	Settable	IRQ27/OPIRQ	OP1	0x0000 00AC
44	Settable	IRQ28/WDTIRQ	WDT	0x0000 00B0
45	Settable	IRQ29/LVDIRQ	LVD/FSCM	0x0000 00B4
46	Settable	IRQ30/PBIRQ	CCP1	0x0000 00B8
47	Settable	IRQ31/PAIRQ	TS	0x0000 00BC

2.3.2. NVIC 寄存器

2.3.2.1. IRQ0~31 中断设置使能寄存器 (NVIC_ISER)

寄存器功能：ISER 使能中断，并显示使能的中断。

基地址：0xE000 E100

Bit	Name	Description	Attribute	Reset
31:0	SETENA[31:0]	中断设置使能位 0：无影响； 1：使能中断；	R/W	0

2.3.2.2. IRQ0~31 中断除能寄存器 (NVIC_ICER)

寄存器功能：ICER 禁止中断，并显示使能的中断。

基地址：0xE000 E180

Bit	Name	Description	Attribute	Reset
31:0	CLRENA[31:0]	中断清零使能位 0：无影响； 1：禁止中断；	R/W	0

2.3.2.3. IRQ0~31 中断挂起设置寄存器 (NVIC_ISPR)

寄存器功能：ISPR 迫使中断进入挂起状态，并显示正在挂起的中断。

注：写 1 到 ISPR 位需符合：

1. 对正处于挂起状态的中断请求无影响；
2. 禁止的中断设置其状态为挂起状态。

基地址：0xE000 E200

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:0	SETPEND[31:0]	中断挂起设置位 0: 无影响; 1: 中断挂起;	R/W	0
------	---------------	--------------------------------	-----	---

2.3.2.4. IRQ0~31 中断挂起清零寄存器 (NVIC_ICPR)

寄存器功能: ICPR 可以取消中断的挂起状态, 并显示正在挂起的中断。

注: 写入 1 到 ICPR, 并不影响相应中断的有效状态。

基地址: 0xE000E280

Bit	Name	Description	Attribute	Reset
31:0	SETPEND[31:0]	中断挂起清除位 0: 无影响; 1: 中断挂起;	R/W	0

2.3.2.5. IRQ0~31 中断优先级寄存器 (NVIC_IPRn) (n=0~7)

寄存器说明: 中断优先级寄存器为每个中断提供 8-bit 优先权选择级别, 每个寄存器管理 4 个中断源的优先权。这意味着寄存器的数量是明确定义的, 且与执行的中断的数量相对应。

基地址: 0xE000 E400 + 0x4 * n

Bit	Name	Description	Attribute	Reset
31:24	PRI_(4*n+3)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 Bits[31:30], 而 Bits [29:24] 读数为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192;	R/W	0
23:16	PRI_(4*n+2)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 Bits[23:22], 而 Bits [21:16] 读数为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192;	R/W	0
15:8	PRI_(4*n+1)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 Bits[15:14], 而 Bits[13:8] 读数为 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存 192;	R/W	0
7:0	PRI_4*n	每个优先级字段的优先级设定值为 0-192。	R/W	0

		值越小，对应中断的优先级越高。处理器只处理每个优先级字段的 Bits[7:6]，而 Bits[5:0]读出于 0 且写入无效，因此如果写 255 到优先级寄存器，寄存器只保存为 192；		
--	--	---	--	--

2.4 应用中断和复位控制（AIRC）

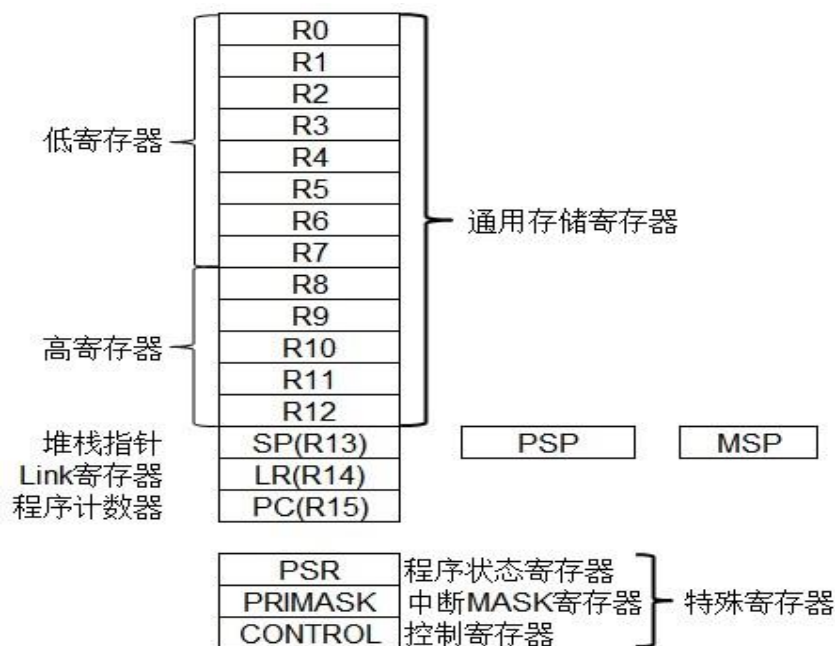
寄存器说明：整个 MCU，包括内核，都可以通过软件设置 AIRC 寄存器的 SYSRESREQ 位来复位。

注：在写入数据到该寄存器时，用户必须同时写入 0x05FA 到 VECTKEY，否则处理器会忽略此次写动作。

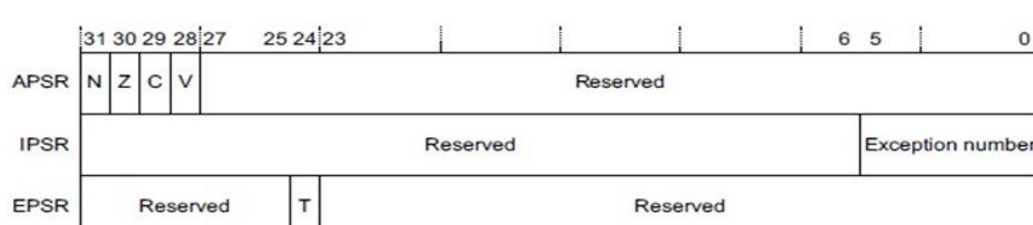
基地址：0xE000 ED0C

Bit	Name	Description	Attribute	Reset
31:16	VECTKEY	寄存器关键字，读取为未知数； 写入 0x05FA 到此位（VECTKEY），否则写寄存器动作无效。	R/W	0
15	ENDIANESS	数据字节顺序应用位 0：小端排列； 1：大端排列；	R/W	0
14:3	Reserved		R	0
2	SYSRESETREQ	系统复位请求位，该位读取为 0 0：没有影响； 1：请求系统复位；	W	0
1	VECTCLRACTIVE	保留以用来调试，该位读取为 0 写入数据到寄存器时，用户必须写入 0 到该位，否则会发生未知情形。	W	0
0	Reserved		R	0

2.5 核心寄存器简介



核心寄存器结构图

Register	Description (参考 Cortex-M0 规格)
R0~R12	通用存储寄存器，用于存储操作数据。
SP (R13)	堆栈指针 SP。在 Thread 模式下，CONTROL 寄存器指示堆栈指针的使用，主堆栈指针 MSP 或者处理堆栈指针 PSP。在复位以后，处理器装载地址 0x00000000 处的值到 MSP。
LR (R14)	Link 寄存器 LR。存储子程序，CALL 功能及其它异常的返回信息。
PSR	程序状态寄存器 PSR 包含了下面几种： 应用程序状态寄存器（APSR）； 中断程序状态寄存器（IPSR）； 执行程序状态寄存器（EPSR）； 注：这些寄存器在 32 位 PSR 中是相互独立的。  PSR 寄存器结构图显示了 32 位寄存器的位分布。APSR 包含 N (31), Z (30), C (29), V (28) 位，其余位为 Reserved。IPSR 包含 Exception number (6-0) 位，其余位为 Reserved。EPSR 包含 T (23) 位，其余位为 Reserved。
PRIMASK	PRIMASK 寄存器可阻止所有已配置优先权的异常事件的发生。
CONTROL	处理器在 Thread 模式下运行时，控制寄存器控制所使用的堆栈。

3 系统控制

3.1 复位

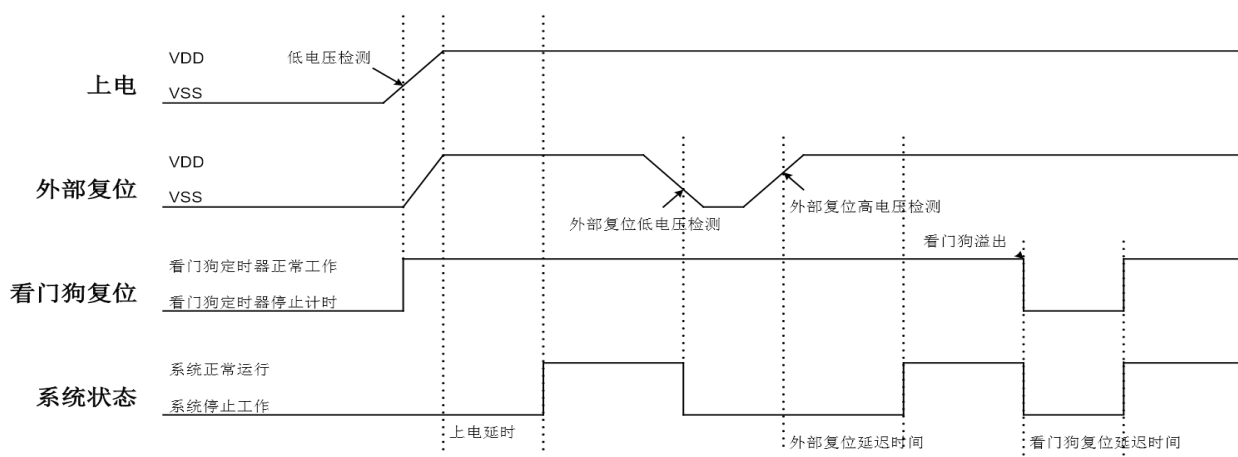
AD3005 MCU 的复位方式有以下 6 种：

1. POR 复位（上电复位）；
2. RST 引脚低电平触发复位（外部复位）；
3. LVD 复位（低电压复位/掉电复位）；
4. WDT 复位（看门狗定时器复位）；
5. SW 复位（软件复位）；
6. I/O 中断复位（设置为 I/O 口中断唤醒，唤醒后触发 CPU 复位）。

系统复位源可通过系统复位状态寄存器（SYS0_RSTST）的不同标志位来识别。复位触发后，复位源作用于 RST 引脚，且 RST 引脚在延迟阶段保持低电平。复位程序的复位向量指向储存器的 0x00000004 地址处。详情请参考中断和 Exception 向量。

系统复位需要一定的时间，并且需要完成完整的上电复位过程。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的启动时间非常短，晶体振荡器的启动时间则比较长。

在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。下面给出了复位时序图。



复位时序图

3.1.1. POR 复位（上电复位）

上电复位与 LVD 的操作密切相关。系统上电过程中，电压趋势为逐渐上升的曲线形式，

需要一定延迟时间才能达到系统所需的正常电压值。上电复位的时序如下：

1. 电源供电：系统检测到电源电压上升并等待其稳定；
2. 外部复位（使能外部引脚时有效）：系统检测外部复位引脚状态，如果不为高电平。则系统保持复位状态直到外部复位引脚复位结束；
3. 系统初始化：所有的系统寄存器被置为默认状态；
4. 振荡器开始工作：振荡器开始提供系统时钟；
5. 执行程序：上电结束后，程序从引导装载程序（Bootloader）处开始运行。

3.1.2. RST 复位（外部复位）

外部复位功能由外部复位引脚控制寄存器 `SYS0_EXRSTCTRL` 控制，初始值为 1，即使能外部复位功能。

外部复位引脚为施密特触发结构，低电平有效。外部复位引脚输入高电平时，系统正常工作。当外部复位引脚输入低电平时，系统开始复位。外部复位操作在上电过程和正常工作模式时有效。需要注意的是，在系统上电完成后，外部复位引脚必须输入高电平，否则系统将一直保持在复位状态（MCU 中 RST 脚和 IO 口复用）。

外部复位的时序如下：

1. 外部复位（当且仅当外部复位引脚为使能状态）：系统检测复位引脚的状态，如果复位引脚不为高电平，则系统会一直保持在复位状态，直到外部复位结束；
2. 系统初始化：所有的系统寄存器被置为初始状态；
3. 振荡器开始工作：振荡器开始提供系统时钟；
4. 执行程序：上电结束后，程序从引导装载程序处（Bootloader）开始运行。

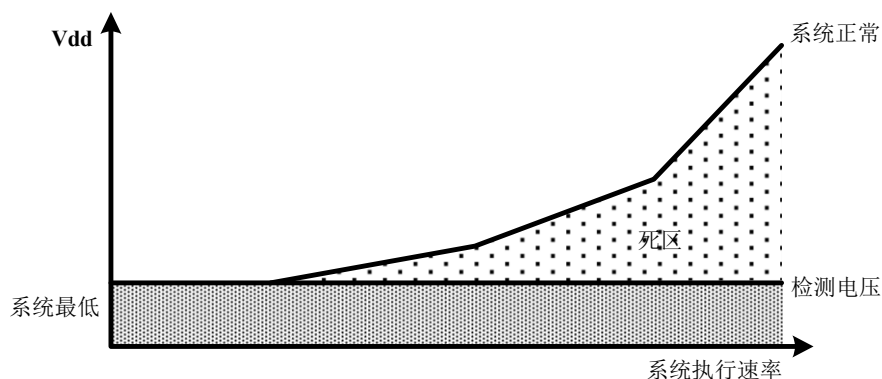
3.1.3. LVD 复位（低电压复位/掉电复位）

3.1.3.1. 系统工作电压

为了改善系统掉电复位的性能，首先必须明确系统具有的最低工作电压值。系统最低工作电压与系统执行速度有关，不同的执行速度下最低工作电压值不同。

如下图所示，系统正常工作电压区域一般高于系统复位电压，同时复位电压由低电压检测（LVD）电平决定。当系统执行速度提高时，系统最低工作电压也相应提高，但由于系统复位电压固定，因此在系统最低工作电压与系统复位电压之间就会出现一个电压区域，当系统

供电电压处在这个电压区域时，系统不能正常工作，也不会产生复位，这个区域即为死区。



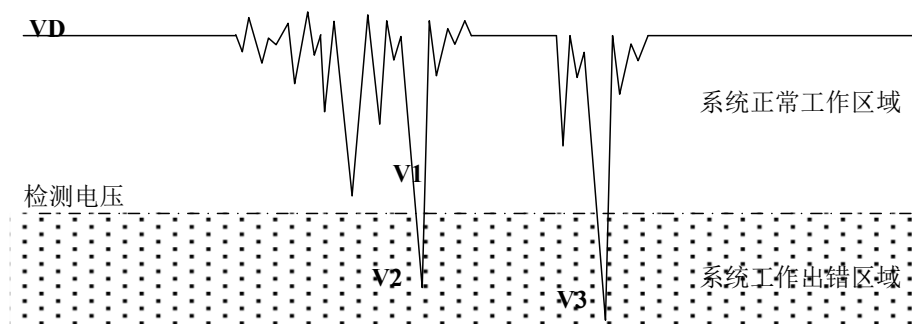
系统工作电压与执行速度关系图

3.1.3.2. 掉电复位

掉电复位针对外部因素引起的系统电压跌落情形（例如：干扰或外部负载的变化），掉电复位可能会引起系统工作状态不正常或程序执行错误。这是由于电压跌落可能会进入系统工作电压的死区。系统死区意味着电源不能满足系统的最小工作电压要求，下图是一个典型的掉电复位示意图。如图，当 VDD 受到严重干扰，电压值被拉低。虚线以上区域系统正常工作，在虚线以下的区域内，系统进入未知的工作状态。当 VDD 跌至 V1 时，系统仍处于正常状态；当 VDD 跌至 V2 和 V3 时，系统进入死区，此时系统工作极易出错。以下情况系统可能进入死区，应尽量避免：

DC 供电： 此时系统多采用电池供电，当电池电压过低或 MCU 驱动负载时，系统电压可能跌落并进入死区。此时电源不会进一步下降到 LVD 检测电压，因此系统维持在死区。

AC 供电： 系统采用 AC 供电时，DC 的电压值受 AC 电源中噪声的影响。当外部负载过高，如 AC 驱动马达时，负载动作产生的干扰也影响到 DC 的稳定。此时 VDD 若由于受到干扰而下降到最低工作电压以下时，系统将有可能进入不稳定工作状态。在 AC 运用中，系统上、下电时间都较长。其中，上电时序保护使得系统正常上电，但下电过程却和 DC 运用中情形类似，AC 电源关断后，VDD 电压在缓慢下降的过程中易进入死区。



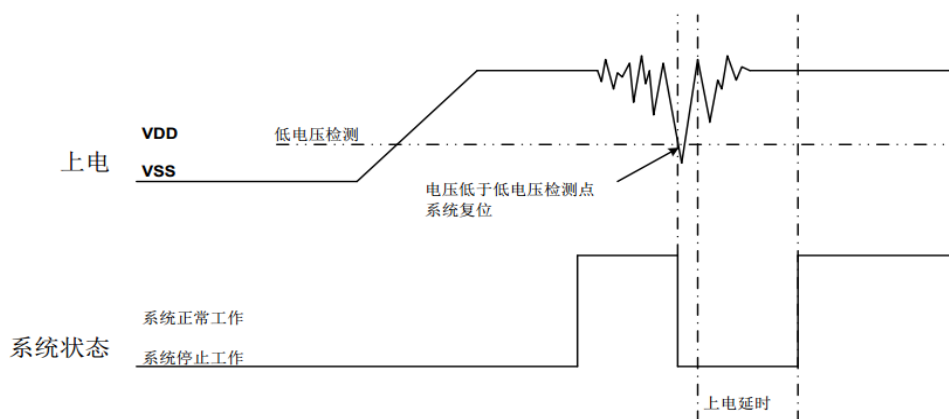
掉电复位示意图

对于改善系统掉电复位的性能，列出以下几点建议：

1. 使用 LVD 复位功能；
2. 使用 WDT 复位功能；
3. 降低系统工作频率；
4. 采用 RST 复位电路（例如：稳压二极管复位电路，电压偏移复位电路，外部 IC 复位等）。

3.1.3.3. LVD 复位

低电压检测(LVD)是 AD3005 MCU 内置的掉电复位保护装置，当 VDD 跌落并低于 LVD 检测电压值时，LVD 会给出一个中断信号到 NVIC，以便触发 CPU 中断。若不能触发中断，则软件通过读取寄存器的显示状态来监控该信号。可以选择一个附加的极限值强制系统复位。但 LVD 检测电平值仅为一个电压点，并不能覆盖所有死区范围，因此采用 LVD 依赖于系统要求和环境状况。电源变化较大时，LVD 能够起到保护作用，如果电源变化触发 LVD，系统工作仍发生错误，那么 LVD 就无法起到保护作用，此时需要采用其它复位方法。下图为 LVD 复位电路工作原理，并给出在 LVD 复位无法正常工作时的参考建议。更多 LVD 信息请参考电气特性。



LVD 复位示意图

如果 LVD 无法正常工作，列出以下几点建议：

1. 采用看门狗复位方式

看门狗定时器用于保证系统正常工作。通常，会在主程序中将看门狗定时器清零，但不要在多个分支程序中清看门狗。若程序正常运行，看门狗不会复位。当系统进入

死区或程序运行出错的时候，看门狗定时器继续计数直至溢出，系统复位。如果看门狗复位后电源仍处于死区，则系统复位失败，保持复位状态，直到系统工作电压恢复到正常值。

2. 降低系统工作频率

系统工作速度越快最低工作电压值越高，从而增大死区电压的范围，因此降低系统工作速度是避免系统进入死区的有效措施。在系统正常工作的情况下，可选择合适的工作速度以避免系统陷入死区，这个方法需要调整整个程序使其满足系统要求。

3. 附加外部复位电路

外部复位方式通过外部复位信号控制 MCU 可靠复位，可以较好改善复位性能。主要采用以下三种外部复位方式：稳压二极管复位电路，电压偏移复位电路和外部 IC 复位。

3.1.4. WDT 复位（看门狗复位）

看门狗复位是系统的一种保护设置。在系统正常运行的状态下，由软件将看门狗定时器清零（喂狗）。若系统运行出错，系统处于未知状态，软件无法正确将看门狗定时器清零，因此看门狗定时器发生溢出，此时系统复位发生。触发看门狗复位后，系统重新启动并运行到正常状态。

看门狗复位的时序如下：

1. 看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；
2. 系统初始化：初始化所有的系统寄存器；
3. 振荡器起振：振荡器正常工作并提供系统时钟；
4. 执行程序：上电复位结束，程序从地址 0x00000000 开始执行。

使用看门狗定时器时应注意以下 3 点：

1. 将看门狗定时器清零之前，先检查 I/O 口的状态和 RAM 数据，可降低系统错误率；
2. 不能在中断程序里清零看门狗定时器，否则无法起到监测系统是否正常运行（指正是否跑飞）的目的；
3. 程序中应有且仅有一个清零看门狗的语句。以此最大限度的发挥看门狗对程序的保护功能。

注：有关看门狗定时器的详细内容，请参阅看门狗定时器章节。

3.1.5. SW 复位（软件复位）

整个 MCU，包括内核部分，都可以通过软件设置 SHYSRESREQ 位来复位。

软件初始化系统复位的流程如下：

1. 通过设置 SYSRESREQ 位准备复位系统；
2. 开始内部复位；
3. 内部复位完成后，MCU 从存储器加载初始的堆栈指针，初始的程序计数器，并由程序计数指定第一条指令，然后开始执行程序。

注：SHYSRESREQ 在 Cortex-M0 规格中位于 AIRC-应用中断和复位控制寄存器中。

3.1.6. I/O 中断复位（中断唤醒）

主要用作低功耗模式的唤醒，详见 3.4.2 章节。

3.2 系统时钟

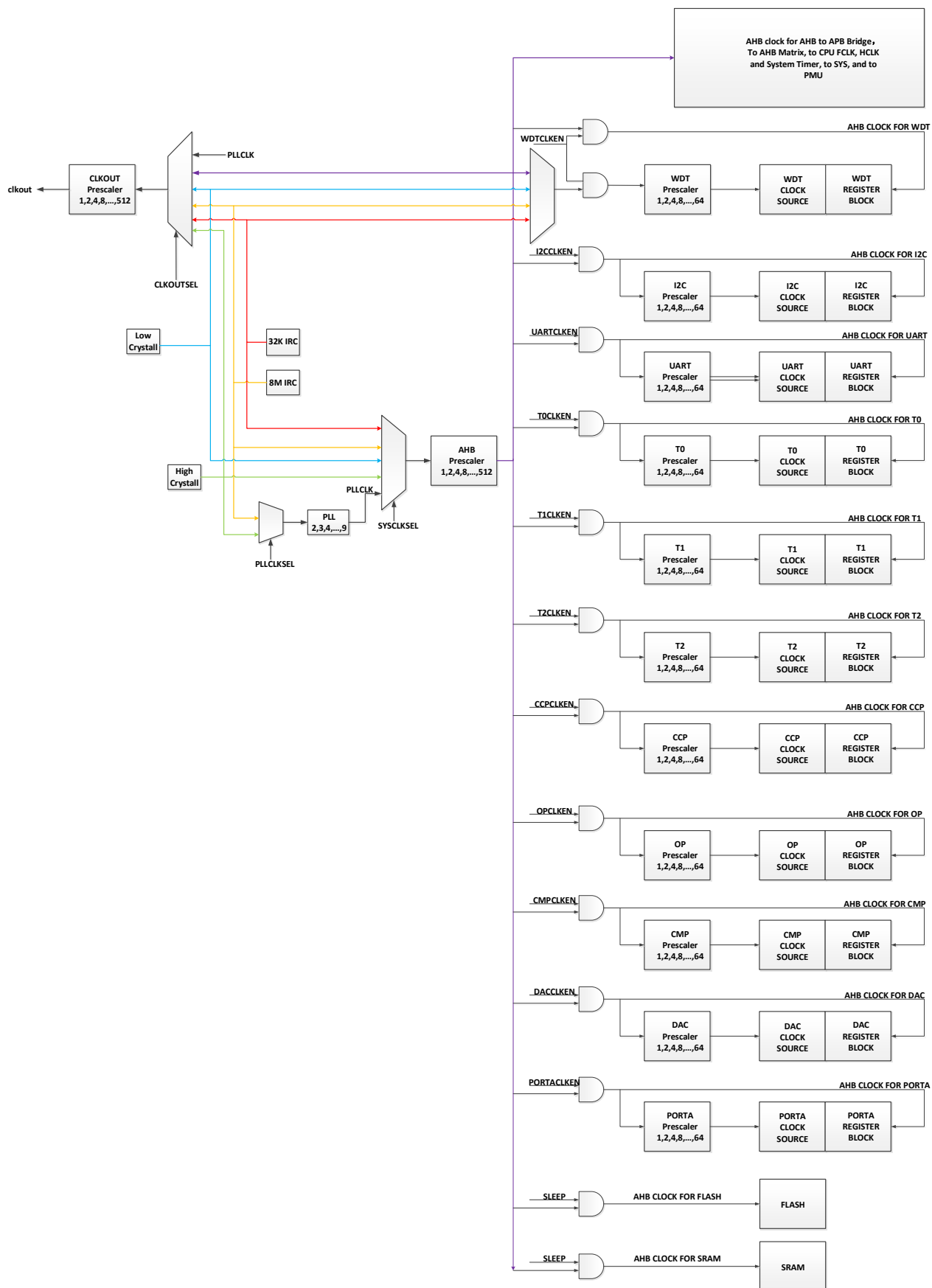
系统时钟（SYSCLK）可以通过不同的时钟源来驱动，列出以下几种：

1. 32MHz 内部高速 RC 时钟（IHRC）；
2. 256KHz 内部低速 RC 时钟（ILRC）；
3. 外部高速晶体振荡器（EHS）；
4. 外部低速晶体振荡器（ELS）；
5. PLL 时钟。

每个时钟源在不使用时都可以独立的打开或关闭，以使功耗达到最优状态。

AD3005 为双时钟 MCU，包括高速时钟和低速时钟。高速时钟由外部振荡电路和内部 PLL 电路提供；低速时钟则由内部低速 RC 振荡电路提供。

3.2.1. 时钟产生框图



3.2.2. 内部时钟源

3.2.2.1. 内部高速 RC 振荡器 (IHRC)

内部高速 RC 振荡器的震荡频率为 32MHz，在普通模式下，其精确度为 $\pm 1\%$ 。通过模拟模块控制寄存器 SYS0_ANBCTRL 中的 IHRCEN 位使能该内部高速 RC 振荡器。

3.2.2.2. 内部低速 RC 振荡器 (ILRC)

内部低速 RC 振荡器的震荡频率为 256KHz。内部低速 RC 振荡器的输出频率容易受到系统电压和环境温度的影响，典型值为 256KHz。

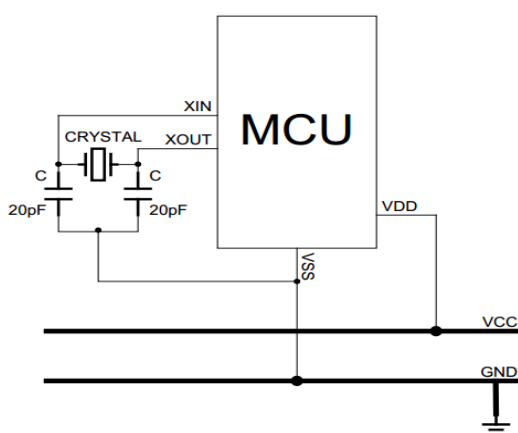
注：只能通过硬件来切换 ILRC 的状态。

3.2.3. 外部时钟源

3.2.3.1. 石英/陶瓷振荡器

石英/陶瓷振荡器通过 MCU 的 XIN 和 XOUT 引脚驱动，针对不同的模式（高/普通/低频模式），其驱动电流也不相同。通过模拟模块控制寄存器 SYS0_ANBCTRL 的 EHSEN 位可以控制 EHS 晶体的状态。

需要注意的是，振荡器和负载电容必须尽可能的靠近 MCU 的振荡器驱动引脚，以便输出最稳定的输出信号。负载电容的容值必须与所选择的振荡器相匹配，否则可能会产生不稳定的震荡频率或无法正常起振。下图为石英/陶瓷振荡器的典型接法。



石英/陶瓷振荡器典型接法示意图

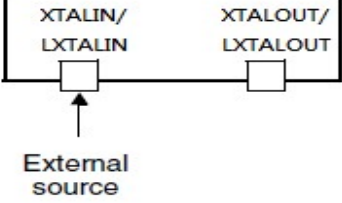
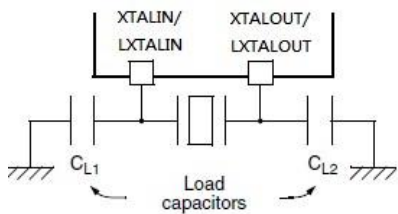
3.2.3.2. 外部高速晶体振荡器 (EHS)

高速振荡电路采用 4MHz~16MHz 晶体振荡器。

3.2.3.3. 外部低速晶体振荡器（ELS）

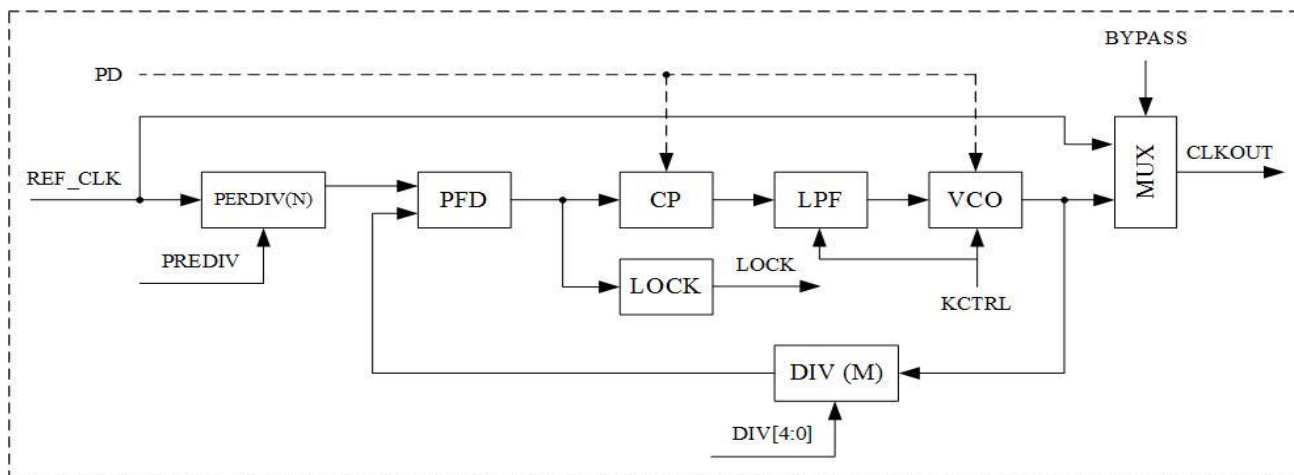
低速振荡电路采用 32768Hz 晶体振荡器。

3.2.3.4. 旁路模式

Clock Source	H/W Configuration	Description
外部时钟源（旁路）		旁路模式下，外部时钟信号以 0~50% 占空比的周期提供给 XTALIN/LXTALIN，而 XTALOUT/LXTALOUT 脚应该是输入信号的反向信号。外部高速晶体的震荡频率可以达到 16MHz。外部低速晶体振荡器使用 32.768KHz。通过寄存器 SYS0_ANBCTRL 的 ELSEN 位选择外部高速/低速模式。
外部 XTAL		系统时钟以主时钟为基础，使用 10~16MHz 的 EHS XTAL 可以使系统产生一个非常精确的时钟频率。而 ELS XTAL 的频率必须为 32.768KHz

3.2.4. PLL 时钟

AD3005 MCU 支持使用 PLL 电路产生系统时钟，用来驱动内核和外设部分的运行。PLL 时钟输入频率为 6MHz~16MHz，时钟输入后通再分频，然后提供给相位频率检测器（PFD）。PFD 比较输入的相位和频率，若它们不匹配，则产生一个控制信号。环路滤波器滤除控制信号，并驱动电压控制振荡器（VCO）产生一个主时钟和两个任意相位。VCO 频率为 40MHz~160MHz。主时钟由 PLL 通过可编程的 Post 分频器进行分频，产生输出时钟（S）。VCO 输出时钟由 M 通过可编程的 Feedback 分频器分频，产生 feedback 时钟。相位频率检测器的输出信号由 Lock 检测器进行监控，当 PLL 锁住时，发出信号到输入时钟。PLL 的稳定时间为 100us。



PLL 时钟结构图

PLL 频率计算和选择:

PLL 频率方程式:

$$F_{PLL} = F_{CLKIN} * M / N$$

PLL 的频率由下列参数决定:

1. FCLKIN: PLLCLKSEL 多路器频率;
2. FVCO: 电压控制振荡器 (VCO) 频率, 40~160MHz;
3. FCLKOUT: PLL 输出频率;
4. N: 系统 PLL Post 分频器比例, 由 PLL 控制寄存器 (SYS0_PLLCTRL) 的 PSEL 位控制;
5. M: 系统 PLL Feedback 分频器比例, 由 PLL 控制寄存器 (SYS0_PLLCTRL) 的 MSEL 位控制。

为给 M、P 和 F 选择合适的值, 建议按照下列条件选择:

1. $6\text{MHz} \leq F_{CLKIN} \leq 16\text{MHz}$;
2. $40\text{MHz} \leq F_{VCO} \leq 96\text{MHz}$;
3. $M = 2, 3, 4, 5, 6, 7, 8, 9 \dots 33$;
4. $P = 2 \text{ or } 4$ (duty 50% +/- 2.5%);
5. $F_{CLKOUT} = 20\text{MHz}, 30\text{MHz}, 40\text{MHz}, 50\text{MHz}, 24\text{MHz}, 36\text{MHz}, 48\text{MHz}$ with jitter < $\pm 500 \text{ ps}$ 。

3.2.5. 系统时钟 (SYSCLK) 选择

系统复位后, IHRC 作为系统时钟。当直接使用时钟源或者使用 PLL 作为系统时钟时,

IHRC 仍然正常工作。若从一个时钟源转换到另一个时钟源时，需要等待目标时钟源准备好（启动延迟后时钟稳定或者锁定 PLL）。若选择一个未准备好的时钟源，则需等待其准备好之后才会开始转换。

SYS0_CSST 寄存器的 Ready 位显示已经准备好的时钟源，SYS0_CLKCFG 寄存器的 SYSCLKST 位则显示的是当前作为系统时钟的时钟源。

3.2.6. CLOCK-OUT 功能

MCU 时钟输出（CLKOUT）性能允许时钟输出到外部 CLKOUT 引脚。此对应 GPIO 的配置寄存器需设定成时钟输出功能。

6 种时钟信号可以作为时钟输出，由 SYS1_AHBCLKEN 寄存器的 CLKOUTSEL 位选择控制。

1. HCLK;
2. IHRC;
3. ILRC;
4. PLL 时钟输出;
5. ELS;
6. EHS。

3.3 系统控制

3.3.1. 系统控制寄存器 0

基地址：0x4008 0000

3.3.1.1. 模拟模块控制寄存器（SYS0_ANBCTRL）

地址偏移：0x000

注：若选择 EHS XTAL/ELS XTAL/IHRC 作为系统时钟或者准备作为系统时钟，则 EHSEN/ELSEN/IHRCEN 位不能被清零。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	EHSEN	外部高速时钟使能位 0：禁止 EHS XTAL;	R/W	0

		1: 使能 EHS XTAL;		
2	ELSEN	外部低速振荡器使能位 0: 禁止外部 32.768KHz 振荡器; 1: 使能外部 32.768KHz 振荡器;	R/W	0
1	ILRCEN	内部低速时钟使能位 0: 禁止内部 256KHz RC 振荡器; 1: 使能内部 256KHz RC 振荡器;	R/W	1
0	IHRCEN	内部高速时钟使能位 0: 禁止内部 32MHz RC 振荡器; 1: 使能内部 32MHz RC 振荡器;	R/W	1

3.3.1.2. PLL 控制寄存器 (SYS0_PLLCTRL)

地址偏移: 0x004

注: 若选择 PLL 作为系统时钟或者准备作为系统时钟, 则 PLEN 位不能被清零。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	PLEN	PLL 使能位 0: 禁止; 1: 使能;	R/W	0
14	KCTRL	VCO 增益控制 0: PLL 输出 40~100MHz; 1: PLL 输出 100~160MHz;	R/W	0
13:12	PLLCLKSEL[1:0]	系统 PLL 时钟源 00: IHRC 8MHz 振荡器; 01: EHS XTAL 4 MHz ~ 16 MHz; XX: 保留;	R/W	0
11	BYPASS	直通位 1: 旁路模式 (clk_out=clk_ref); 0: PLL 模式;	R/W	0
10	LOCK	锁定指示 0: PLL 未上锁或上锁失败; 1: PLL 上锁成功;	R/W	0
9	PIDS	0: Post 分频使能; 1: Post 分频禁止;	R/W	0
8	FSEL	0: PLL 不分频输入; 1: PLL 2 分频输入;	R/W	0
7:5	PSEL[2:0]	Post 分频值 000: P = 2; 001: P = 4; 010: P = 6;	R/W	0

		011: P = 8; 100: P = 10; 101: P = 12; 110: P = 14; 111: P = 16;		
4:0	MSEL[4:0]	Feedback 分频值 00000: M = 2; 00001: M = 3; 00010: M = 4; 00011: M = 5; 00100: M = 6; 00101: M = 7; 00110: M = 8; 00111: M = 9; 01000: M = 10; 01001: M = 11; 01010: M = 12; 01011: M = 13; 01100: M = 14; 01101: M = 15; 01110: M = 16; 01111: M = 17; 10000: M = 18; 10001: M = 19; 10010: M = 20; 10011: M = 21; 10100: M = 22; 10101: M = 23; 10110: M = 24; 10111: M = 25; 11000: M = 26; 11001: M = 27; 11010: M = 28; 11011: M = 29; 11100: M = 30; 11101: M = 31; 11110: M = 32; 11111: M = 33;	R/W	0

注：PLL 输出范围是 40~160 MHz，如果 PLL 输出 40MHz 以下，可通过后分频来实现。
 MCU 在稳定状态下最高支持 64MHz 的时钟频率。

3.3.1.3. 时钟源状态寄存器 (SYS0_CSST)

地址偏移：0x008

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	PLLRDY	PLL 时钟准备标志位 0: PLL 未锁定; 1: PLL 被锁定;	R	0
3	EHSRDY	外部高速时钟准备标志位 0: EHS 振荡器没有准备好; 1: EHS 振荡器已经准备好;	R	0
2	ELSRDY	外部低速时钟准备标志位 0: ELS 振荡器没有准备好; 1: ELS 振荡器已经准备好;	R	0
1	ILRCRDY	ILRC 时钟准备标志位 0: ILRC 没有准备好; 1: ILRC 已经准备好;	R	0
0	IHRCRDY	IHRC 时钟准备标志位 0: IHRC 没有准备好; 1: IHRC 已经准备好;	R	1

3.3.1.4. 系统时钟配置寄存器 (SYS0_CLKCFG)

地址偏移: 0x00C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:4	SYSCLKST[2:0]	系统时钟切换状态 000: IHRC 作为系统时钟; 001: ILRC 作为系统时钟; 010: EHS XTAL 作为系统时钟; 011: ELS XTAL 作为系统时钟; 100: PLL 作为系统时钟; XXX: 保留; 通过硬件设置和清零, 以显示当前作为系统时钟的时钟源。	R	0
3	Reserved		R	0
2:0	SYSCLKSEL[2:0]	系统时钟切换。通过软件设置和清零。 000: IHRC; 001: ILRC; 010: EHS XTAL; 011: ELS XTAL; 100: PLL 输出; XXX: 保留;	R/W	0

3.3.1.5. AHB 时钟预分频寄存器 (SYS0_AHBCP)

地址偏移: 0x010

Bit	Name	Description	Attribute	Reset
7:4	Reserved		R	0
3:0	AHBPRES[3:0]	AHB 时钟源预分频值 000: SYSCLK / 1; 001: SYSCLK / 2; 010: SYSCLK / 4; 011: SYSCLK / 8; 100: SYSCLK / 16; 101: SYSCLK / 32; 110: SYSCLK / 64; 111: SYSCLK / 128;	R/W	0

3.3.1.6. 系统复位状态寄存器 (SYS0_RSTST)

注: 该寄存器包含了除 **DPDWAKEUP** 复位以外的所有的复位源, 这是由于 **PMU_CTRL** 寄存器的 **MODE** 位已经显示了这种复位情况。

地址偏移: 0x014

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	PORRSTF	POR 复位标志位, 发生 POR 复位时, 由硬件设置。 1: 发生 POR 复位; 0: 未发生 POR 复位; 写 1 时将该位清零。	R/W	1
3	EXTRSTF	外部复位标志位。发生外部复位时, 由硬件设置。 1: 发生外部复位; 0: 未发生外部复位; 写 1 时将该位清零。	R/W	0
2	LVDRSTF	LVD 复位标志位。发生 LVD 复位时, 由硬件设置。 1: 发生 LVD 复位; 0: 未发生 LVD 复位; 写 1 时将该位清零。	R/W	0
1	WDTRSTF	WDT 复位标志位。发生 WDT 复位时, 由硬件设置。 1: 发生 WDT 复位; 0: 未发生 WDT 复位; 写 1 时将该位清零。	R/W	0
0	SWRSTF	软件复位标志位。发生软件复位时, 由硬件设置。 1: 发生软件复位; 0: 未发生软件复位;	R/W	0

		写 1 时将该位清零。		
--	--	-------------	--	--

3.3.1.7. LVD 控制寄存器 (SYS0_LVDCTRL)

地址偏移: 0x018

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	LV DEN	LVD 使能位 0: 禁止; 1: 使能;	R/W	0
14	LVDRSTEN	LVD 复位使能位。 0: 禁止; 1: 使能;	R/W	0
13	LV DIE	LVD 中断使能位, 当 LV DEN=1 且 LV DIE=1 时 LVR 模块开始工作 0: 禁止; 1: 使能;	R/W	0
12	LV DIF	LVD 中断标志位 1: 产生中断标志并清除; 0: 没有产生标志;	R/W	0
11:7	Reserved		R	0
6:4	LV DINTLVL[2:0]	LVD 中断电平 000: 1.60V; 001: 1.80V; 010: 2.00V; 011: 2.20V; 100: 2.40V; 101: 2.60V; 110: 3.00V; 111: 3.20V;	R/W	0
3	Reserved		R	0
2:0	LVDRSTLVL[2:0]	LVD 复位电平 000: 1.60V; 001: 1.80V; 010: 2.00V; 011: 2.20V; 100: 2.40V; 101: 2.60V; 110: 3.00V; 111: 3.20V;	R/W	0

3.3.1.8. 外部引脚复位控制寄存器 (SYS0_EXRSTCTRL)

地址偏移: 0x01C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	RESETDIS	1: 使能 PD5 作为 RST 0: 禁止 PD5 作为 RST	R/W	0

3.3.2. 系统控制寄存器 1

基地址: 0x4008 2000

3.3.2.1. 外设时钟使能寄存器 (SYS1_AHBCLKEN)

SYS_AHBCLKEN 寄存器使能 AHB 时钟提供给独立系统和外设模块。

注:

1. 禁止时钟时, 不能通过软件读取外设模块的值, 该值返回为 0;
2. 若 CLKOUTSEL 不为 0, 则硬件直接将 GPIO 设置为 CLKOUT 功能;
3. 为了省电, 用户应该合理关闭 AHB 时钟。

地址偏移: 0x000

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTSEL[2:0]	时钟输出源 000: 禁止; 001: ILRC 时钟; 010: ELS 时钟; 100: HCLK; 101: IHRC 时钟; 110: EHS 时钟; 111: PLL 时钟输出;	R/W	0
27:22	Reserved		R	0
21	SPICLKEN	SPI 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
20	I2C_SLV_CLKEN	I2C_SLV 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
19	CLUCLKEN	CLU 时钟使能位。 0: 禁止;	R/W	0

		1: 使能;		
18	T3CLKEN	T3 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
17	TSCLKEN	TSensor 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
16	ADCCLKEN	ADC 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
15	PDCLKEN	PD 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
14	PCCLKEN	PC 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
13	CKOECLKEN	CKOE 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
12	LVTCLKEN	LVT 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
11	DAC0CLKEN	DAC0/ DAC1 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
10	OP1CLKEN	OP1 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
9	OP0CLKEN	OP0 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
8	WDTCLKEN	WDT 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
7	UARTCLKEN	UART 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
6	Reserved		R	0
5	CCPCLKEN	CCP 时钟使能位。 0: 禁止; 1: 使能;	R/W	0

4	T2CLKEN	T2 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
3	T1CLKEN	T1 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
2	TOCLKEN	T0 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
1	PBCLKEN	PB 时钟使能位。 0: 禁止; 1: 使能;	R/W	0
0	PACLKEN	PA 时钟使能位。 0: 禁止; 1: 使能;	R/W	0

3.3.2.2. APB 时钟分频寄存器 0 (SYS1_APB0)

注: SYS1_APB0 寄存器将使能后的 APB 时钟分频后提供给独立系统和外设模块。

地址偏移: 0x004

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	UARTPRE[2:0]	UART 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
27	Reserved		R	0
26:24	I2CPRE[2:0]	I2C 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0

23	Reserved		R	0
22:20	CCPPRE[2:0]	CCP 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
19	Reserved		R	0
18:16	T2PRE[2:0]	T2 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
15	Reserved		R	0
14:12	T1PRE[2:0]	T1 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
11	Reserved		R	0
10:8	TOPRE[2:0]	T0 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
7	Reserved		R	0

6:4	PBPRES[2:0]	PB 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
3	Reserved		R	0
2:0	PAPRES[2:0]	PA 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0

3.3.2.3. APB 时钟分频寄存器 1 (SYS1_APBPCP1)

注: SYS1_APBPCP1 寄存器将使能后的 APB 时钟分频后提供给独立系统和外设模块。

地址偏移: 0x008

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTPRE[2:0]	CKOE 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
27	Reserved		R	0
26:24	ADCPRE[2:0]	ADC 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8;	R/W	0

		100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;		
23	Reserved		R	0
22:20	SYSTICPRE[2:0]	SYSTIC 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
19	Reserved		R	0
18:16	LVTPRE[2:0]	LVT 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
15	Reserved		R	0
14:12	DACOPRE[2:0]	DACO/DAC1 分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
11	Reserved		R	0
10:8	OP1PRE[2:0]	OP1 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16;	R/W	0

		101: HCLK/32; 110: HCLK/64; 111: HCLK/128;		
7	Reserved		R	0
6:4	OPOP[2:0]	OP0 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
3	Reserved		R	0
2:0	WDTPRE[2:0]	WDT 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0

3.3.2.4. APB 时钟分频寄存器 2 (SYS1_APBPCP2)

注: SYS1_APBPCP2 寄存器将使能后的 APB 时钟分频后提供给独立系统和外设模块。

地址偏移: 0x018

Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
26:24	CLUPRE[2:0]	CLU 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
23:15	Reserved			

14:12	TSPRE[2:0]	TESNSOR 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
11	Reserved		R	0
10:8	T3PRE[2:0]	T3 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
7	Reserved		R	0
6:4	PDPRE[2:0]	PD 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0
3	Reserved		R	0
2:0	PCPRE[2:0]	PC 时钟预分频值 000: HCLK/1; 001: HCLK/2; 010: HCLK/4; 011: HCLK/8; 100: HCLK/16; 101: HCLK/32; 110: HCLK/64; 111: HCLK/128;	R/W	0

3.3.2.5. 接口复位寄存器 (SYS1_PRST)

注：硬件自动清零。

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20	CLURST	CLU 复位使能位。 0：不使能； 1：使能；	R/W	0
19	T3RST	T3 复位使能位。 0：不使能； 1：使能；	R/W	0
18	PDRST	PD 口复位使能位。 0：不使能； 1：使能；	R/W	0
17	PCRST	PC 口复位使能位。 0：不使能； 1：使能；	R/W	0
16	CRYPTRSTEN	CRYPT 复位使能位。 0：不使能； 1：使能；	R/W	0
15	TSRST	TESNSOR 复位使能位。 0：不使能； 1：使能；	R/W	0
14	ADCRST	ADC 复位使能位。 0：不使能； 1：使能；	R/W	0
13:12	Reserved		R	0
11	DACORST	DAC0 复位使能位。 0：不使能； 1：使能；	R/W	0
10	CMPRST	CMP 复位使能位。 0：不使能； 1：使能；	R/W	0
9	OPRST	OP 复位使能位。 0：不使能； 1：使能；	R/W	0
8	WDTRST	WDT 复位使能位。 0：不使能； 1：使能；	R/W	0
7	UARTRST	UART 复位使能位。 0：不使能；	R/W	0

		1: 使能;		
6	I2CRST	I2C 复位使能位。 0: 不使能; 1: 使能;	R/W	0
5	CCPRST	CCP 复位使能位。 0: 不使能; 1: 使能;	R/W	0
4	T2RST	T2 复位使能位。 0: 不使能; 1: 使能;	R/W	0
3	T1RST	T1 复位使能位。 0: 不使能; 1: 使能;	R/W	0
2	T0RST	T0 复位使能位。 0: 不使能; 1: 使能;	R/W	0
1	PBRST	PB 口复位使能位。 0: 不使能; 1: 使能;	R/W	0
0	PARST	PA 口复位使能位。 0: 不使能; 1: 使能;	R/W	0

3.3.2.6. BOOT 选择器 (SYS1_CONFIG)

注: 硬件自动清零。

地址偏移: 0x010

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	BOOT	0: FLASH 映射到 32' h0000_0000 - 32' h0001_0000 地址; 1: SRAM 映射到 32' h0000_0000 - 32' h0000_2000 地址;	R/W	0

3.4 系统操作模式

针对不同的时钟速率以及功率消耗要求, AD3005 MCU 内置 4 种不同的操作模式, 用来控制和显示振荡器和模拟外设的运行状态。列出以下四种。

- ACTIVE 模式
- IDLE 模式
- POWER SAVE 模式

- POWER OFF 模式

3.4.1. 正常模式（ACTIVE）

在正常模式下，ARM Cortex-M0 内核、存储器、外设都由系统时钟来计时。SYS1_AHBCLKEN 寄存器控制外设的运行状况。不同的外设有各自独立的外设时钟，一般由系统时钟接入，但可以通过各自时钟分频器进行分频操作，同时这些外设时钟还可以单独禁止或使能。通过模块的使能位，也可以单独的控制各种不同的模拟模块（IHRC, EHS XTAL, ELS XTAL, PLL, FLASH, LVD, OP, OP1, DAC）。

3.4.2. 低功耗模式（IDLE/POWER SAVE/POWER OFF）

可以通过配置寄存器选择 3 种不同的低功耗模式：IDLE 模式，POWER SAVE 模式，POWER OFF 模式。PMU_CTRL 寄存器控制使得 MCU 进入不同的低功耗模式。在不同的运用场景下，通过改变时钟源，重新配置 PLL 值，或者改变系统时钟的分频值来改变系统时钟速率，从而在设计需求的系统运行速度和系统功耗之间找到一个平衡点。

程序运行期间，为减少系统功耗可选择关闭芯片上各独立外设的时钟，来达到降低功耗的目的。

注：

1. Power save 模式和 Power off 模式下，不支持在线调试；
2. 低功耗模式下，将没有输出的引脚设置为合适的模式以减小功耗，强烈建议将这些引脚设置为输入上拉模式。

3.4.2.1. IDLE 模式

IDLE 模式下，ARM Cortex-M0 内核的系统时钟停止工作，同时指令也暂停执行。

外设功能：若在 SYS1_AHBCLKEN 寄存器中使能时钟，则 IDLE 模式下继续工作，并且任何中断可使处理器恢复运行。IDLE 模式减少了处理器，内存系统和相关的控制器以及内部总线使用的动态功耗。模拟模块（IHRC, EHS XTAL, ELS XTAL, PLL, FLASH, LVD, OP, OP1, DAC）的运行状态由模拟模块的使能位决定。处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。

中断发生时可以将系统从 IDLE 模式下唤醒。IDLE 模式下，RESET 引脚保持有效状态。通过以下步骤，可以进入 IDLE 模式：

1. 写入 0x01 到 PMU_CTRL 寄存器；
2. 执行 ARM Cortex-M0 WFI 指令。

3.4.2.2. POWER SAVE 模式

在 POWER SAVE 模式下，ARM Cortex-M0 内核的系统时钟停止工作，指令也暂停执行。

振荡器的电源状态处于掉电模式时，外设功能的时钟停止工作，高速时钟源也停止工作，此时仅有低速时钟源（ELS XTAL 和 ILRC）继续工作。处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。所有的 GPIO 引脚都有唤醒功能，用户必须通过 GPIO 寄存器设置引脚唤醒的边沿极性，使其可以唤醒 MCU（只有边沿触发才有唤醒功能）。当 GPIO 产生中断事件到 ARM 内核时，系统退 PWSAVE 模式，此外必须在 NVIC 中使能中断对应的输入引脚。

POWER SAVE 模式下，RESET 引脚保持有效状态。通过下面的步骤，可以进入 POWER SAVE 模式：

1. 写入 0x02 到 PMU_CTRL 寄存器中；
2. 执行 ARM WFI 指令。

POWER SAVE 模式的优点在于，它可以使振荡器和 PLL 等时钟发生模块处于掉电状态，从而更好地降低系统的动态功耗。此外，在 POWER SAVE 模式下可使 FLASH 处于掉电状态以降低静态的漏电功耗，但这也会导致唤醒 FLASH 存储器需要更长的时间。

3.4.2.3. POWER OFF 模式

POWER OFF 模式下，整个芯片的电源（关闭内置电压调整器）和时钟都处于关闭状态，包括需要关闭的 GPIO 引脚。此时，处理器状态寄存器，外设寄存器和内部 SRAM 的值不再保持。但 MCU 提供 2 个 BACKUP 寄存器来保存数据，GPIO 引脚的状态能被锁定。所有的 GPIO 引脚都可以作为 DPDWAKEUP 引脚使用，但 DPDWAKEUP 引脚用于唤醒 MCU 时须设置为输入上拉模式并在进入 PWOFF 模式之前保持高电平。用户必须通过 GPIO 寄存器设置引脚状态（输入上拉，输入下拉，输入悬浮，输出高，输出低，开漏），并通过 PMU_LATCHCTRL1 寄存器锁存 GPIO 引脚的状态。

MCU 被唤醒后，用户可以读取 PMU_CTRL 寄存器的内容，并核实复位状态为唤醒事件将 MCU 从 POWER OFF 模式中唤醒而不是冷复位。若 MCU 从 POWER OFF 模式中被唤醒，用户必须在 GPIO 寄存器中进行同样的设置（输入上拉，输入下拉，输入悬浮，输出高，输出低，

开漏)，并通过 PME_LATCHCTRL2 寄存器释放 GPIO 引脚的状态。

3.4.2.3.1. 进入 POWER OFF 模式

通过下面的步骤可以使系统从普通模式进入 PWOFF 模式：

1. 设置 GPIO 引脚
2. 在 PWOFF 模式下保存数据到 Backup 寄存器的 DATA 位（可选）
3. 写入 0x5A5A0001 到 PMU_IO0 寄存器中锁存所有 GPIO 引脚的状态
4. 写入 0x03 到 PMU_CTRL 寄存器
5. 步骤 1 到步骤 5 所花费的时间应该超过 20us
6. 执行 ARM Cortex-M0 WFI/WFE 指令

这 6 个步骤执行完成后，PMU 关闭内置电压调整器，并等待 DPDWAKEUP 引脚的唤醒信号。

3.4.2.3.2. 退出 POWER OFF 模式

通过下面的步骤可以将系统从 PWOFF 模式中唤醒：

1. 将某个 GPIO 引脚产生电平变化；
 - 1) PMU 打开内部电压调整器，当内核电压到达上电复位（POR）的电压值时，触发系统复位，芯片重启；
 - 2) 除 BACKUP 和 PMU_CTRL 外的所有寄存器都复位。
2. 读取 PMU_CTRL 寄存器的内容，确认复位是由 PWOFF 触发而不是冷复位；
3. 将 PMU_CTRL 寄存器清零；
4. 读取存储在 BACKUP 寄存器中的数据；（可选）
5. 写入 0xA5A50001 到 PMU_IO1 寄存器中释放所有 GPIO 引脚的状态。

设置 PMU，等待进入下一次 POWER OFF 模式。

3.5 系统唤醒

系统进入睡眠模式后，由 F/W 使能或禁止高速时钟。若高速时钟停止运行，MCU 需要从睡眠模式下唤醒，则 MCU 需要等待 2048 个外部高速振荡器时钟周期和 32 个内部高速振荡器时钟周期，以使振荡电路进入稳定工作状态，等待的这段时间就称为唤醒时间。唤醒时间结束后，系统才进入到普通模式。

系统在睡眠模式时不执行程序。唤醒触发信号可以将系统唤醒进入正常模式。唤醒功能内置中断操作，系统被唤醒时触发系统执行中断服务程序。

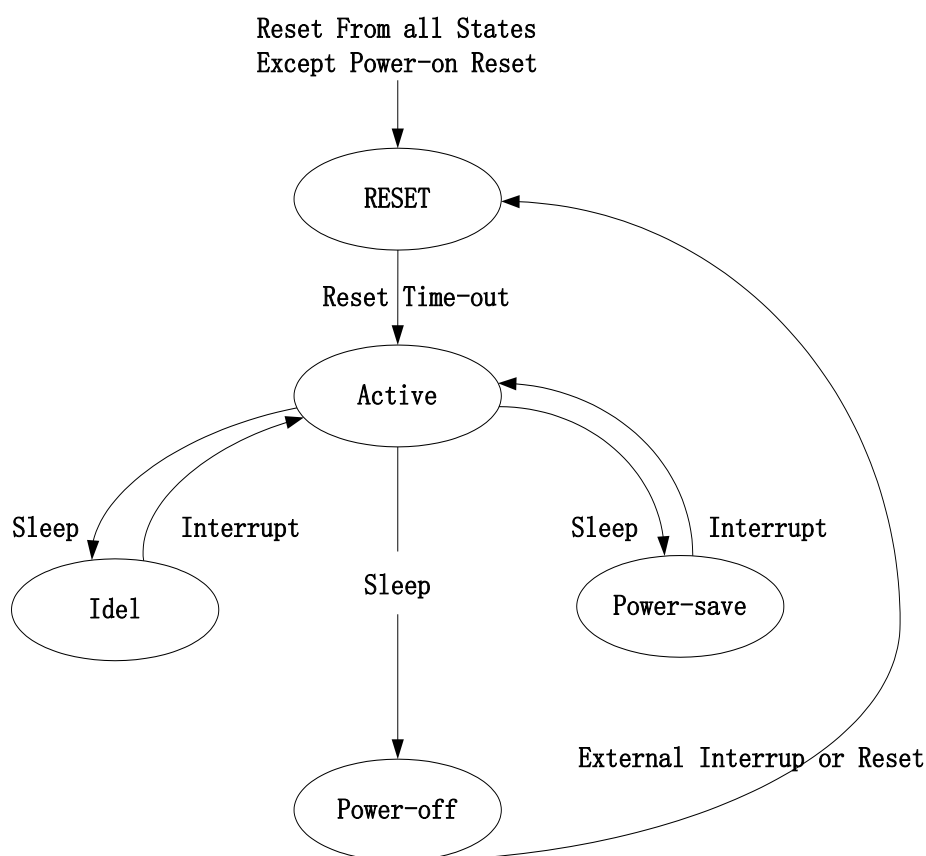
- IDLE 模式的唤醒触发源为所有中断和 RESET 引脚。
- POWER SAVE 的唤醒触发源为 GPIO 中断，TIMER1、TIMER2、TIMER3 中断（此时 T1/2/3 时钟源为低速时钟），OP 中断，WDT 中断和 RESET 引脚。
- POWER OFF 模式的唤醒触发源为 GPIO 引脚，该引脚在进入 POWER OFF 模式前为输入上拉模式并被锁存。

注：若时钟没有停止运行，则系统从睡眠模式唤醒就不需要唤醒时间。

3.5.1. PMU

电源管理单元（Power Management Unit, PMU）是控制 MCU 电源功能的模块。

3.5.1.1. PMU 状态结构



PMU 状态转换图

3.5.1.2. PMU 操作模式和接口工作状态

低功耗模式工作情况：

Module	ACTIVE	IDLE	POWER SAVE	POWER OFF
HFOSC	√	√		
PLL	√	√		
CPU	√			
FLASH	√	√	√	
SRAM	√	√	√	
TIMER0	√	√		
TIMER1	√	√	√	
TIMER2	√	√	√	
TIMER3	√	√	√	
I2C	√	√		
UART	√	√		
CCP	√	√		
LFOSC	√	√	√	
WDT	√	√		
OP	√	√		
LDO	√	√	√	
BGR	√	√	√	
LVT	√	√	√	
POR	√	√	√	
IO	√	√	√	√
RESET	√	√	√	

低功耗模式下的唤醒方式：

Module	ACTIVE	IDLE	POWER SAVE	POWER OFF
TIMER0	√	√		
TIMER1	√	√	√	
TIMER2	√	√	√	
TIMER3	√	√	√	

I2C	√	√		
UART	√	√		
CCP	√	√		
WDT	√	√		
OP	√	√	√	
LVT	√	√	√	
IO	√	√	√	√
RESET	√	√		

3.5.1.3. PMU 寄存器

基地址：0x4008 4000

3.5.1.3.1. BANKUP 寄存器 0

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13	BACKUPBUSY	BACKUP0 寄存器, BACKUP1 状态寄存器 1: 正在写入; 0: 写入完成;	R	0
12:8	BACKUPADDR	进入 PWOFF 模式前保留数据或 PWOFF 复位后读取数的地址; 共有 16 个 BYTE, 用来存储数据, 这个选地址; 注: 这个寄存器要按 char 来访问, 先写 addr, 再写/读数据。	R/W	0
7:0	BACKUPWRDATA	在 PWOFF 模式前写入需保留的 BACKUP DATA 数据;	R/W	0

3.5.1.3.2. BANKUP 寄存器 1

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:13	Reserved		R	0
12:8	Reserved		R	0
7:0	BACKUPRDATA	PWOFF 复位后通过 BACKUP 寄存器 0 <BACKUPADDR> 中对应的地址可读取进入 PWOFF 之前保存的对应地址的 BACKUPDATA 数据;	R/W	0

注：PWOFF 唤醒需要通过 BACKUP0 寄存器写入地址恢复相应地址的数据。

3.5.1.3.3. 电源控制寄存器 (PMU_CTRL)

电源控制寄存器可选择控制 ARM Cortex-M0 进入不同的掉电模式（IDLE 模式，PWSAVE 模式，PWOFF 模式），并且提供独立的标志位来指示是否进入 IDLE 模式、PWSAVE 模式或 PWOFF 模式。

注：在 PWOFF 模式下，只要 VDD 引脚仍然接有电源，PMU_CTRL 寄存器就会保留数据；只有在芯片的所有电源都被断开后，此寄存器才会复位。

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:0	SMCR[3:0]	低功耗模式选择控制位 000：禁止； 001：WFI 指令使 MCU 进入 IDLE 模式； 010：WFI 指令使 MCU 进入 POWER SAVE 模式； 011：WFI 指令使 MCU 进入 POWER OFF 模式； 其它：禁止。	R/W	0

3.5.1.3.4. I/O 锁存使能寄存器 0 (PMU_IO0)

进入 POWER OFF 状态后，I/O 状态由 LATCHEN 锁存的值决定，写入 0x5A5A0001，进行锁存；写入 0xA5A50001 清除 GPIO 锁定功能，由内部寄存器决定 I/O 状态。

注：锁存的是输出状态、上下拉、哪个 IO 唤醒、中断，输入可以读到实际值，输出无法改变。

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:16	Key[15:0]	写入 0X5A5A，使能操作此寄存器	W	0
15:2	Reserved		R	0
1	LATCHEN_BUSY	GPIO 寄存器写入状态 Read：1：表示写入忙； 0：表示更新完成	R	0

0	LATCHDIS	锁存使能位 Read: 0: 无; 1: 锁存数据 GPIO。	R/W	0
---	----------	---------------------------------------	-----	---

3.5.1.3.5. I/O 锁存解除寄存器 1 (PMU_IO1)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:16	Key[15:0]	写入 0XA5A5, 使能操作此寄存器	W	0
15:2	Reserved		R	0
1	LATCHEN _BUSY	GPIO 切换到锁存使能位 1: 表示切换忙; 0: 表示更新完成;	R	0
0	LATCHDIS	GPIO 锁存状态清除使能位 0: 不起作用; 1: 使能 GPIO 锁存功能;	R/W	0

3.5.1.3.6. 模拟寄存器 (PMU_ANA0)

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:16	Key[15:0]	写入 0X3AF0, 使能操作此寄存器;	W	0
15:9	Reserved		R	0
8	IWDTEN	IWDT 的使能 1: 允许使能; 0: 禁止使能;	R/W	0
7:5	IWDTDIV	IWDT 分频选择 000: 1 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;	R/W	0
4	SMTEN	0: 0.2VDD/0.4VDD (IO 低电平变为高电平时为 0.4VDD, 高电平变为低电平时为 0.2VDD); 1: 0.3VDD/0.7VDD (IO 低电平变为高电平时为 0.7VDD, 高电平变为低电平时为 0.3VDD)	R/W	0
3	SUTEN	CPU 上电延时使能位	R	0

		1: 表示切换忙; 0: 表示更新完成;		
2:0	SUT	CPU 上电延时时间设置 000: 0.5us; 001: 1us; 010: 2us; 011: 4us; 100: 8us; 101: 16us; 110: 32us; 111: 64us;	R/W	0

3.5.1.3.7. 模拟寄存器 (PMU_ANA1)

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:16	Key[15:0]	写入 0X3AF1, 使能操作此寄存器;	W	0
15:10	Reserved		R	0
9	ALGEN	使能 ALG 位 1: 使能; 0: 禁止;	R/W	0
8	Reserved			
7	OSC8MEN	使能 OSC8M 位 1: 使能; 0: 禁止;	R/W	1
6	OSC256KEN	使能 OSC256K 位 1: 使能; 0: 禁止;	R/W	1
5	OSC256K_SEL	OSC256K 时钟分频位 1: 允许分频; 0: 禁止分频;	R/W	0
4	BGR_BUF_EN	使能 BGR_BUF 位 1: 使能; 0: 禁止;	R/W	1
3	BGR_EN	使能 BGR 位 1: 使能; 0: 禁止;	R/W	1
2	LD01P5_EN	使能 LD01.5V 位 1: 使能; 0: 禁止;	R/W	0
1	LDO_LOADEN	使能 LDO_LOAD 位	R/W	0

		1: 使能; 0: 禁止;		
0	LDO_OSC_EN	使能 LDO_OSC 位 1: 使能; 0: 禁止;	R/W	0

3. 5. 1. 3. 8. BGR_TRIM (PMU_ANA2)

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:16	Key[15:0]	写入 0X3AF2, 使能操作此寄存器;	W	0
15:13	Reserved		R	0
12:9	BGR_TEMP_TRIM[3:0]	BGR_TEMP 的 TRIM 值;	R/W	0
8:4	BGR_TRIM	BGRTRIM (1V0 电压相关) 值;	R/W	0
3	Reserved		R	0
2:0	LD01P5_TRIM	LD01.5VTRIM (内核工作电压相关) 值;	R/W	111

3. 5. 1. 3. 9. OSC8M_TRIM (PMU_ANA3)

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:16	Key	写入 0X3AF3, 使能操作此寄存器;	W	0
15:13	Reserved		R	0
12	FSEL	IHRC 倍频设置;	R/W	0
11:0	OSC8M_TRIM	OSC8M 的 TRIM 值;	R/W	0

3. 5. 1. 3. 10. I/O 锁存控制寄存器 0 (PMU_ANA4)

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:16	Key	写入 0X3AF4, 使能操作此寄存器;	W	0
15:0	Reserved		R	0

3. 5. 1. 3. 11. I/O 锁存控制寄存器 1 (PMU_ANA5)

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:16	Key	写入 0X3AF5，使能操作此寄存器；	W	0
15:8	Reserved		R	0
7:0	OSC256K_TRIM	OSC256K 的 TRIM 值；	R/W	0

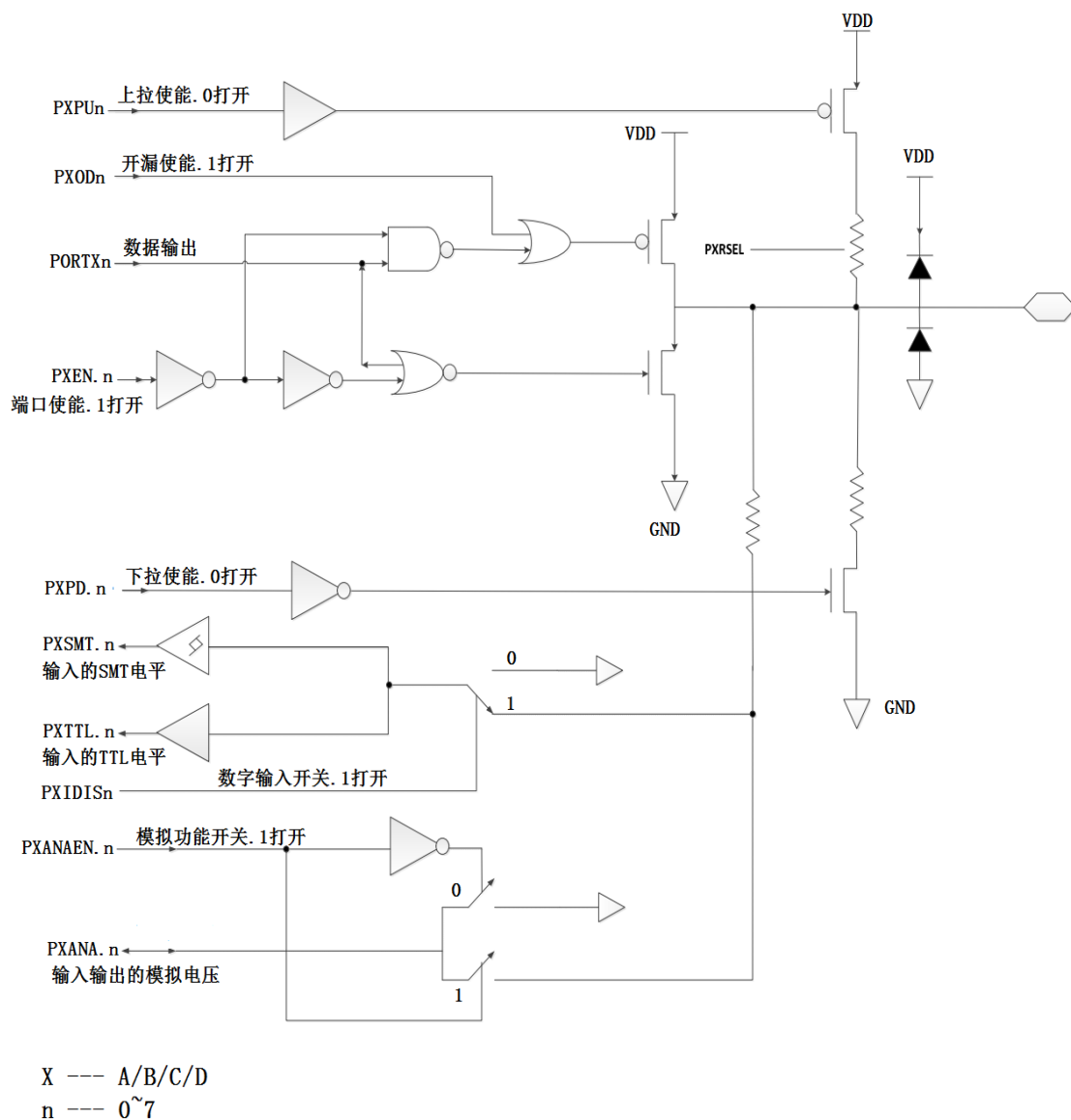
4 GPIO 端口

4.1 GPIO 概述

GPIO 端口可以通过软件来设置为输入或输出模式。

- 每个独立的引脚可以作为外部中断输入引脚使用；
- 可以通过单一的上升沿，下降沿，或者双边沿触发来触发中断；
- I/O 配置寄存器控制对应 I/O PORT 的电气特性；
- I/O 引脚可通过内部上拉/下拉电阻寄存器选择不同的上拉/下拉电阻；
- I/O 引脚可按照特殊功能寄存器表配置为对应的复用引脚。

4.2 GPIO 引脚电路结构图



引脚电路结构图

4.3 GPIO 寄存器

注：在默认状态下，所有的 GPIO 引脚都为输入模式和悬浮状态，可通过 GPIO 寄存器配置为输入上拉，输入下拉，输入悬浮，输出高，输出低和开漏模式。也可通过 GPIO 寄存器选择不同的上拉/下拉电阻。

4.3.1. PA 寄存器

基地址：0x4006 0000

4.3.1.1. TRISA I/O 方向寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	TRISA[7:0]	1: IO 为输入模式; 0: IO 为输出模式;	R/W	1111_1111

4.3.1.2. PORTA 数据输出寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PORTA[7:0]	输出数据（写为 IO 状态）;	R/W	0

4.3.1.3. PINA 数据输入寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PINA[7:0]	输入数据（读取 IO 状态）;	R	0

4.3.1.4. PAPU 上拉使能寄存器

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAPU[7:0]	1: 关闭上拉; 0: 使能上拉;	R/W	1111_1111

4.3.1.5. PAPUR 上拉电阻选择寄存器

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAPUR[7:0]	1: 100K; 0: 20K;	R/W	0

4.3.1.6. PAPD 下拉使能寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAPD[7:0]	1: 关闭下拉; 0: 使能下拉;	R/W	1111_1111

4.3.1.7. PAOD 开漏使能寄存器

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAOD[7:0]	1: 打开开漏; 0: 关闭开漏;	R/W	0

4.3.1.8. PAIS 中断设置寄存器

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAIS[7:0]	选择 PA. x 是由电平触发还是边沿触发 (x=0~7) 0: 边沿触发; 1: 电平触发;	R/W	0

4.3.1.9. PAIBS 中断双边沿设置寄存器

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAIBS[7:0]	选择是否由双边沿触发 PA. x 的中断 (x = 0~7) 0: 由寄存器 PAIEV 控制 PA. x 的中断; 1: 由双边沿触发 PA. x 的中断;	R/W	0

4.3.1.10. PAIEV 中断事件寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAIEV[7:0]	选择 PA. x 在上升沿或下降沿触发中断 (x=0~7) 0: 取决于寄存器 PAIS 的设置, 上升沿或者 PA. x 为高电平触发中断;	R/W	0

		1: 取决于寄存器 PAIS 的设置, 下降沿或者 PA. x 为低电平触发中断;		
--	--	---	--	--

4.3.1.11. PAIE 中断使能寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IE[7:0]	PA. x 中断使能位 (x = 0~7) 0: 禁止; 1: 使能;	R/W	0

4.3.1.12. PAIF 中断源状态寄存器

该寄存器显示 PA 控制中断源的状态, 若 PAIE 寄存器的位置 1, 则发送 GPIO 中断到中断控制器中。

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IF[7:0]	PA 中断源标志位 (x = 0~7) 0: 无中断; 1: 有中断发生;	R	0

4.3.1.13. PAIC 中断清零寄存器

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IC[7:0]	PA. x 中断标志清零使能位 (x = 0~7) 0: 没有影响; 1: 清除 PA. x 中断标志位;	R/W	0

4.3.1.14. PAANA 模拟输入/输出使能寄存器

偏移地址: 0x034

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	ANA[7:0]	PA. x 模拟输入输出使能位 (x = 0~7) 0: 关闭引脚 PA. xANA _x 的模拟输入输出功能; 1: 打开引脚 PA. xANA _x 的模拟输入输出功能;	R/W	0

4.3.1.15. PAIDIS 数字输入/输出使能寄存器

偏移地址：0x038

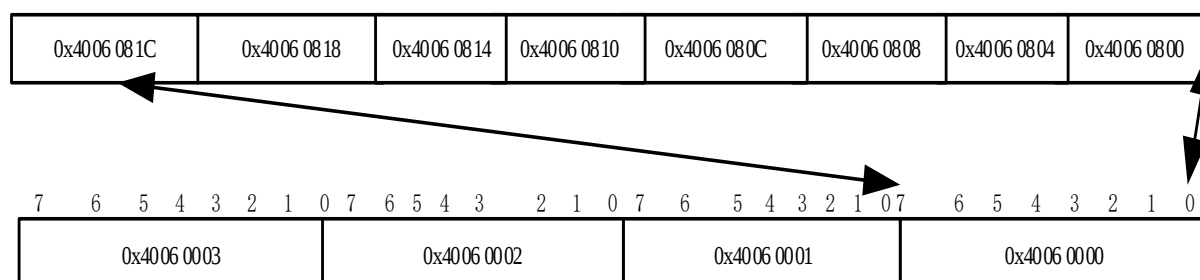
Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IDIS[7:0]	PA. x 数字输入/输出使能寄存器 (x = 0~7) 0: 数字输入打开; 1: 数字输入关闭;	R/W	0

4.3.1.16. PAMASK 位带寄存器

偏移地址：0x0800~0x0FFC

PAMASK 寄存器的地址范围为 0x40060800~0x40060ffc，其每一个地址依次分别控制 TRISA、PORTA、PINA、PAPU、PAPUR、PAPD、PAOD、PAIS、PAIBS、PAIEV、PAIE、PAIF、PAIC、PAANA、PAIDIS、PASMTV 寄存器的从低到高的每一位。如图以 TRISA 为例。

例如：TRISA.0: 40060800, TRISA.1: 40060804, TRISA.2: 40060808;



4.3.1.17. PASMTV 施密特寄存器

偏移地址：0x03C

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
7:0	SMTA[7:0]	0: 0.2VDD/0.4VDD (IO 由低变高时为 0.4VDD, 由高变低时为 0.2VDD); 1: 0.3VDD/0.7VDD (IO 由低变高时为 0.7VDD, 由高变低时为 0.3VDD);	R/W	1111_1111

4.3.1.18. PAALTF 复用功能寄存器

偏移地址：0x040

Bit	Name	Description	Attribute	Reset
31:25	Reserved		R	0

24:22	PA7ALTF[2:0]	PA7 功能寄存器:	R/W	0
21:19	PA6ALTF[2:0]	PA6 功能寄存器:	R/W	0
18:16	PA5ALTF[2:0]	PA5 功能寄存器:	R/W	0
14:12	PA4ALTF[2:0]	PA4 功能寄存器:	R/W	0
11:9	PA3ALTF[2:0]	PA3 功能寄存器:	R/W	0
8:6	PA2ALTF[2:0]	PA2 功能寄存器:	R/W	0
5:3	PA1ALTF[2:0]	PA1 功能寄存器:	R/W	0
2:0	PA0ALTF[2:0]	PA0 功能寄存器:	R/W	0

具体功能定如下（ALTF 选择 n 则代表当前功能为 FUNCn 对应的功能）:

序号	Pin name	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7
1	PA0	IO	SCK (SPI)		PWM00	PWM00	TOCKI	CCP00OUT	
2	PA1	IO	S_SDI	M_SDO	PWM10	PWM01/T1CKI		CCP10UT	
3	PA2	IO	SDA		PWM20	PWM10	T10	CCP20UT	CLU00UT
4	PA3	IO	SCL	CKOE					
5	PA4	IO	SCL	TX	PWM40	PWM20	CK	CCP40UT	TX_RX
6	PA5	IO	T2CKI	RX	PWM50	PWM21	DT	CCP50UT	
7	PA6	IO	T3CKI	BCLK	PWM21	PWM30	T10		
8	PA7	IO	S_SDO	M_SDI	PWM31	PWM31	T10		

IO 功能映射:

数字功能由 PIN FUNC 寄存器控制选择，模拟功能有 ANASEL 控制，FUSE 由配置字确认功能映射设置步骤一般如下:

1. 将 IO 设置为合适的方向（一般模拟功能需设置为输入并打开模拟功能）;
2. 在 ALTF 中将相应 IO 设置为合适的功能;
3. 配置相应的外设。

4.3.2. PB 寄存器

基地址: 0x4006 2000

4.3.2.1. TRISB I/O 方向寄存器

偏移地址: 0x000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	TRISB[7:0]	1: IO 为输入模式; 0: IO 为输出模式;	R/W	1111_1111

4.3.2.2. PORTB 数据输出寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PORTB[7:0]	输出数据 (写为 IO 状态);	R/W	0

4.3.2.3. PINB 数据输入寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PINA[7:0]	输入数据 (读取 IO 状态);	R	0

4.3.2.4. PBPU 上拉使能寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBPU[7:0]	1: 关闭上拉; 0: 使能上拉;	R/W	1111_1111

4.3.2.5. PBPUR 上拉电阻选择寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBPUR[7:0]	1: 100K; 0: 20K;	R/W	0

4.3.2.6. PBPD 下拉使能寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:8	Reserved		R	0
7:0	PBPD[7:0]	1: 关闭下拉; 0: 使能下拉;	R/W	1111_1111

4.3.2.7. PBOD 开漏使能寄存器

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBOD[7:0]	1: 打开开漏; 0: 关闭开漏;	R/W	0

4.3.2.8. PBIS 中断设置寄存器

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBIS[7:0]	选择 PB. x 是由电平触发还是边沿触发 (x=0~7) 0: 边沿触发; 1: 电平触发;	R/W	0

4.3.2.9. PBIBS 中断双边沿设置寄存器

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBIBS[7:0]	选择是否由双边沿触发 PB. x 的中断 (x=0~7) 0: 由寄存器 PAIEV 控制 PB. x 的中断; 1: 由双边沿触发 PB. x 的中断;	R/W	0

4.3.2.10. PBIEV 中断事件寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PBIEV[7:0]	选择 PB. x 在上升沿或下降沿触发中断 (x=0~7) 0: 取决于寄存器 PBIS 的设置, 上升沿或者 PB. x 为高电平触发中断; 1: 取决于寄存器 PBIS 的设置, 下降沿或者 PB. x 为低电平触发中断;	R/W	0

4.3.2.11. PBIE 中断使能寄存器

偏移地址：0x028

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IE[7:0]	PB. x 中断使能位 (x = 0~7) 0: 禁止; 1: 使能;	R/W	0

4.3.2.12. PBIF 中断源状态寄存器

该寄存器显示 PB 控制中断源的状态，若 PBIE 寄存器的位置 1，则发送 GPIO 中断到中断控制器中。

偏移地址：0x02C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IF[7:0]	PB 中断源标志位 (x = 0~7) 0: 无中断; 1: 有中断发生;	R	0

4.3.2.13. PBIC 中断清零寄存器

偏移地址：0x030

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IC[7:0]	PB. x 中断标志清零使能位 (x = 0~7) 0: 没有影响; 1: 清除 PB. x 中断标志位;	R/W	0

4.3.2.14. PBANA 模拟输入/输出使能寄存器

偏移地址：0x034

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	ANA[7:0]	PB. x 模拟输入输出使能位 (x = 0~7) 0: 关闭引脚 PB. xANAx 的模拟输入输出功能; 1: 打开引脚 PB. xANAx 的模拟输入输出功能;	R/W	0

4.3.2.15. PBIDIS 数字输入/输出使能寄存器

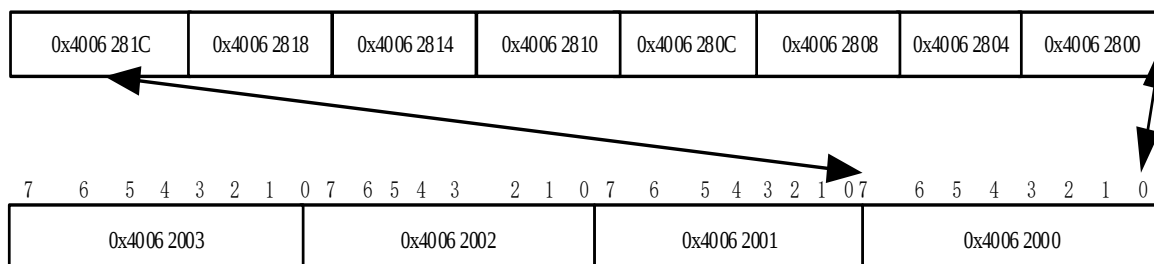
偏移地址：0x038

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	IDIS[7:0]	PB. x 数字输入/输出使能寄存器 (x = 0~7) 0: 数字输入打开; 1: 数字输入关闭;	R/W	0

4.3.2.16. PBMASK 位带寄存器

偏移地址: 0x2800~0x2FFC

PBMASK 寄存器的地址范围为 0x40062800~0x40062ffc, 其每一个地址依次分别控制 TRISB、PORTB、PINB、PBPUR、PBPUR、PBPUR、PBOD、PBIS、PBIBS、PBIEV、PBIE、PBIF、PBIC、PBANA、PBIDIS、PBSMTV 寄存器的从低到高的每一位。如图举例控制 TRISB 寄存器。



4.3.2.17. PBSMTV 施密特寄存器

偏移地址: 0x03C

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
7:0	SMTB[7:0]	0: 0.2VDD/0.4VDD (IO 由低变高时为 0.4VDD, 由高变低时为 0.2VDD); 1: 0.3VDD/0.7VDD (IO 由低变高时为 0.7VDD, 由高变低时为 0.3VDD);	R/W	1111_1111

4.3.2.18. PBALTF 复用功能寄存器

偏移地址: 0x040

Bit	Name	Description	Attribute	Reset
31:25	Reserved		R	0
24:22	PB7ALTF[2:0]	PB7 功能寄存器;	R/W	0
21:19	PB6ALTF[2:0]	PB6 功能寄存器;	R/W	0
18:16	PB5ALTF[2:0]	PB5 功能寄存器;	R/W	0

14:12	PB4ALTF[2:0]	PB4 功能寄存器;	R/W	0
11:9	PB3ALTF[2:0]	PB3 功能寄存器;	R/W	0
8:6	PB2ALTF[2:0]	PB2 功能寄存器;	R/W	0
5:3	PB1ALTF[2:0]	PB1 功能寄存器;	R/W	0
2:0	PB0ALTF[2:0]	PB0 功能寄存器;	R/W	0

具体功能定如下（ALTF 选择 n 则代表当前功能为 FUNCn 对应的功能）:

序号	Pin name	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7
1	PB0	IO	TX	RX	PWM00	PWM40	TOCKI	CCP0OUT	CLU1OUT
2	PB1	IO	RX	TX	PWM10	PWM41	T1CKI	CCP1OUT	
3	PB2	IO	T2CKI	CKOE	PWM20	PWM50		CCP2OUT	
4	PB3	IO	SCL	TX	PWM30	PWM51	CK	CCP3OUT	TX_RX
5	PB4	IO	SDA	RX	PWM40	PWM00	DT	CCP4OUT	
6	PB5	IO	T3CKI	BCLK	PWM50	PWM01		CCP5OUT	CLU2OUT
7	PB6	IO	SSN		PWM41	PWM10			CLU3OUT
8	PB7	IO			PWM51	PWM11			

4.3.3. PC 寄存器

基地址: 0x4006 4000

4.3.3.1. TRISC I/O 方向寄存器

偏移地址: 0x000

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	TRISC[6:0]	1: IO 为输入模式; 0: IO 为输出模式;	R/W	1111_111

4.3.3.2. PORTC 数据输出寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PORTC[6:0]	输出数据（写为 IO 状态）;	R/W	0

4.3.3.3. PINC 数据输入寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PINC[6:0]	输入数据读（读取 IO 状态）；	R	0

4.3.3.4. PCPU 上拉使能寄存器

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCPU[6:0]	1: 关闭上拉； 0: 使能上拉；	R/W	1111_111

4.3.3.5. PCPUR 上拉电阻选择寄存器

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCPUR[6:0]	1: 100K； 0: 20K；	R/W	0

4.3.3.6. PCPD 下拉使能寄存器

偏移地址：0x014

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCPD[6:0]	1: 关闭下拉； 0: 使能下拉；	R/W	1111_111

4.3.3.7. PCOD 开漏使能寄存器

偏移地址：0x018

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCOD[6:0]	1: 打开开漏； 0: 关闭开漏；	R/W	0

4.3.3.8. PCIS 中断设置寄存器

偏移地址：0x01C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCIS[6:0]	选择 PC. x 是由电平触发还是边沿触发 (x=0~6) 0: 边沿触发; 1: 电平触发;	R/W	0

4.3.3.9. PCIBS 中断双边沿设置寄存器

偏移地址：0x020

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCIBS[6:0]	选择是否由双边沿触发 PC. x 的中断 (x=0~6) 0: 由寄存器 PCIEV 控制 PC. x 的中断; 1: 由双边沿触发 PC. x 的中断;	R/W	0

4.3.3.10. PCIEV 中断事件寄存器

偏移地址：0x024

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PCIEV[6:0]	选择 PC. x 在上升沿或下降沿触发中断 (x=0~6) 0: 取决于寄存器 PCIS 的设置, 上升沿或者 PC. x 为高电平触发中断; 1: 取决于寄存器 PCIS 的设置, 下降沿或者 PC. x 为低电平触发中断;	R/W	0

4.3.3.11. PCIE 中断使能寄存器

偏移地址：0x028

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IE[6:0]	PC. x 中断使能位 (x = 0~6) 0: 禁止; 1: 使能;	R/W	0

4.3.3.12. PCIF 中断源状态寄存器

该寄存器显示 PC 控制中断源的状态, 若 PCIE 寄存器的位置 1, 则发送 GPIO 中断到中断控制器中。

偏移地址：0x02C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IF[6:0]	PC 中断源标志位 (x = 0~6) 0: 无中断; 1: 有中断发生;	R	0

4.3.3.13. PCIC 中断清零寄存器

偏移地址：0x030

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IC[6:0]	PC. x 中断标志清零使能位 (x = 0~6) 0: 没有影响; 1: 清除 PC. x 中断标志位;	R/W	0

4.3.3.14. PCANA 模拟输入/输出使能寄存器

偏移地址：0x034

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	ANA[6:0]	PC. x 模拟输入输出使能位 (x = 0~6) 0: 关闭引脚 PC. xANAx 的模拟输入输出功能; 1: 打开引脚 PC. xANAx 的模拟输入输出功能;	R/W	0

4.3.3.15. PCIDIS 数字输入/输出使能寄存器

偏移地址：0x038

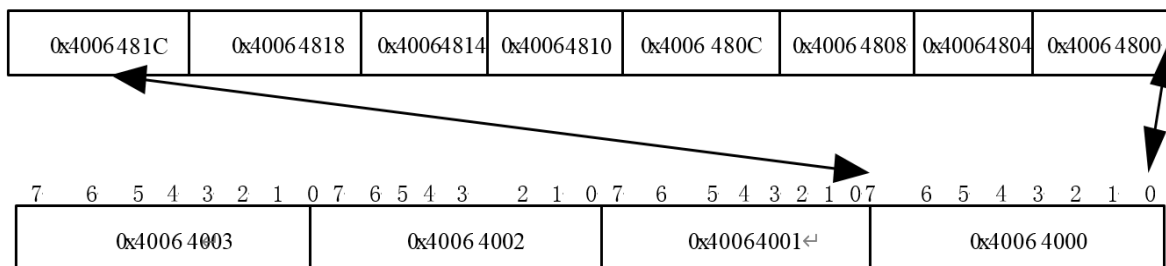
Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IDIS[6:0]	PC. x 数字输入/输出使能寄存器 (x = 0~6) 0: 数字输入打开; 1: 数字输入关闭;	R/W	0

4.3.3.16. PCMASK 位带寄存器

偏移地址：0x4800~0x4FFC

PCMASK 寄存器的地址范围为 0x40064800~0x40064ffc，其每一个地址依次分别控制 TRISC、PORTC、PINC、PCPU、PCPUR、PCPD、PCOD、PCIS、PCIBS、PCIEV、PCIE、PCIF、PCIC、PCANA、PCIDIS、PCSMTV 寄存器的从低到高的每一位。如图举例控制 TRISC

寄存器。



4.3.3.17. PCSMTV 施密特寄存器

偏移地址：0x03C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	SMT6[6:0]	0: 0.2VDD/0.4VDD (IO 由低变高时为 0.4VDD, 由高变低时为 0.2VDD); 1: 0.3VDD/0.7VDD (IO 由低变高时为 0.7VDD, 由高变低时为 0.3VDD);	R/W	1111_111

4.3.3.18. PCALTF 复用功能寄存器

偏移地址：0x040

Bit	Name	Description	Attribute	Reset
31:22	Reserved		R	0
21:19	PC6ALTF[2:0]	PC6 功能寄存器;	R/W	0
18:16	PC5ALTF[2:0]	PC5 功能寄存器;	R/W	0
14:12	PC4ALTF[2:0]	PC4 功能寄存器;	R/W	0
11:9	PC3ALTF[2:0]	PC3 功能寄存器;	R/W	0
8:6	PC2ALTF[2:0]	PC2 功能寄存器;	R/W	0
5:3	PC1ALTF[2:0]	PC1 功能寄存器;	R/W	0
2:0	PC0ALTF[2:0]	PC0 功能寄存器;	R/W	0

具体功能定如下 (ALTF 选择 n 则代表当前功能为 FUNCn 对应的功能):

序号	Pin name	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7
1	PC0	IO	SDA		PWM00	PWM20		CCP0OUT	
2	PC1	IO	SCL		PWM10	PWM21		CCP1OUT	
3	PC2	IO	TOCKI	CKOE	PWM20	PWM30		CCP2OUT	CLU2OUT

4	PC3	IO	T1CKI	TX	PWM30	PWM31	CK	CCP3OUT	TX_RX
5	PC4	IO	T2CKI	RX	PWM11	PWM40	DT	CCP4OUT	
6	PC5	IO	T3CKI	BCLK	PWM01	PWM41		CCP5OUT	CLU3OUT
7	PC6	IO	SDA						

4.3.4. PD 寄存器

基地址：0x40066 000

4.3.4.1. TRISD I/O 方向寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	TRISD[6:0]	1: IO 为输入模式; 0: IO 为输出模式;	R/W	1111_111

4.3.4.2. PORTD 数据输出寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PORTD[6:0]	输出数据（写为 IO 状态）;	R/W	0

4.3.4.3. PIND 数据输入寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PIND[6:0]	输入数据读（读取 IO 状态）;	R	0

4.3.4.4. PDPU 上拉使能寄存器

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDPU[6:0]	1: 关闭上拉; 0: 使能上拉;	R/W	1111_111

4.3.4.5. PDPUR 上拉电阻选择寄存器

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDPUR[6:0]	1: 100K; 0: 20K;	R/W	0

4.3.4.6. PDPD 下拉使能寄存器

偏移地址：0x014

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDPD[6:0]	1: 关闭下拉; 0: 使能下拉;	R/W	1111_111

4.3.4.7. PDOD 开漏使能寄存器

偏移地址：0x018

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDOD[6:0]	1: 打开开漏; 0: 关闭开漏;	R/W	0

4.3.4.8. PDIS 中断设置寄存器

偏移地址：0x01C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDIS[6:0]	选择 PD. x 是由电平触发还是边沿触发 (x=0~6) 0: 边沿触发; 1: 电平触发;	R/W	0

4.3.4.9. PAIBS 中断双边沿设置寄存器

偏移地址：0x020

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDIBS[6:0]	选择是否由双边沿触发 PD. x 的中断 (x = 0~6) 0: 由寄存器 PDIEV 控制 PD. x 的中断;	R/W	0

		1: 由双边沿触发 PD. x 的中断;		
--	--	----------------------	--	--

4.3.4.10. PDIEV 中断事件寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	PDIEV[6:0]	选择 PD. x 在上升沿或下降沿触发中断 (x=0~6) 0: 取决于寄存器 PDIS 的设置, 上升沿或者 PD. x 为高电平触发中断; 1: 取决于寄存器 PDIS 的设置, 下降沿或者 PD. x 为低电平触发中断;	R/W	0

4.3.4.11. PDIE 中断使能寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IE[6:0]	PD. x 中断使能位 (x = 0~6) 0: 禁止; 1: 使能;	R/W	0

4.3.4.12. PDIF 中断源状态寄存器

该寄存器显示 PD 控制中断源的状态, 若 PDIE 寄存器的位置 1, 则发送 GPIO 中断到中断控制器中。

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IF[6:0]	PD 中断源标志位 (x = 0~6) 0: 无中断; 1: 有中断发生;	R	0

4.3.4.13. PDIC 中断清零寄存器

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IC[6:0]	PD. x 中断标志清零使能位 (x = 0~6) 0: 没有影响;	R/W	0

		1: 清除 PD. x 中断标志位;		
--	--	--------------------	--	--

4.3.4.14. PDANA 模拟输入/输出使能寄存器

偏移地址: 0x034

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	ANA[6:0]	PD. x 模拟输入输出使能位 ($x = 0 \sim 6$) 0: 关闭引脚 PD. xANAx 的模拟输入输出功能; 1: 打开引脚 PD. xANAx 的模拟输入输出功能;	R/W	0

4.3.4.15. PDIDIS 数字输入/输出使能寄存器

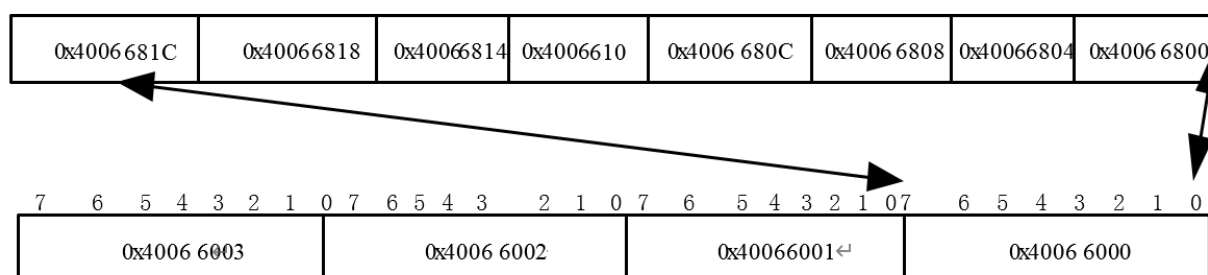
偏移地址: 0x038

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	IDIS[6:0]	PD. x 数字输入/输出使能寄存器 ($x = 0 \sim 6$) 0: 数字输入打开; 1: 数字输入关闭;	R/W	0

4.3.4.16. PDMASK 位带寄存器

偏移地址: 0x6800~0x8FFC

PDMASK 寄存器的地址范围为 0x40066800~0x400668ffc, 其每一个地址依次分别控制 TRISC、PORTC、PINC、PDPU、PDPUR、PDPD、PDOD、PDIS、PDIBS、PDIEV、PDIE、PDIF、PDIC、PDANA、PDIDIS、PDSMTV 寄存器的从低到高的每一位。如图举例控制 TRISD 寄存器。



4.3.4.17. PDSMTV 施密特寄存器

偏移地址: 0x03C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0

6:0	SMTD[6:0]	0: 0.2VDD/0.4VDD (IO 由低变高时为 0.4VDD, 由高变低时为 0.2VDD); 1: 0.3VDD/0.7VDD (IO 由低变高时为 0.7VDD, 由高变低时为 0.3VDD);	R/W	1111_111
-----	-----------	--	-----	----------

4.3.4.18. PDALTF 复用功能寄存器

偏移地址: 0x040

Bit	Name	Description	Attribute	Reset
31:22	Reserved		R	0
21:19	PD6ALTF[2:0]	PD6 功能寄存器;	R/W	0
18:16	PD5ALTF[2:0]	PD5 功能寄存器;	R/W	0
14:12	PD4ALTF[2:0]	PD4 功能寄存器;	R/W	0
11:9	PD3ALTF[2:0]	PD3 功能寄存器;	R/W	0
8:6	PD2ALTF[2:0]	PD2 功能寄存器;	R/W	0
5:3	PD1ALTF[2:0]	PD1 功能寄存器;	R/W	0
2:0	PD0ALTF[2:0]	PD0 功能寄存器;	R/W	0

具体功能定如下 (ALTF 选择 n 则代表当前功能为 FUNCn 对应的功能):

序号	Pin name	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7
1	PD0	IO	SCL	TOCKI	PWM01	PWM50			
2	PD1	IO	T1CKI		PWM11	PWM51			
3	PD2	IO	T2CKI		PWM21				
4	PD3	IO	T3CKI	CKOE	PWM31				
5	PD4	IO		TX			CK		TX_RX
6	PD5	IO		RX			DT		
7	PD6	IO		BCLK					

5 通用定时器

AD3005 的通用定时器是一个通过可编程预分频/后分频器驱动的 32/24/16 位自动装载定时/计数器。共有 4 个通用定时器，每个定时器都是完全独立的，没有互相共享任何资源。它们可以一起同步操作。

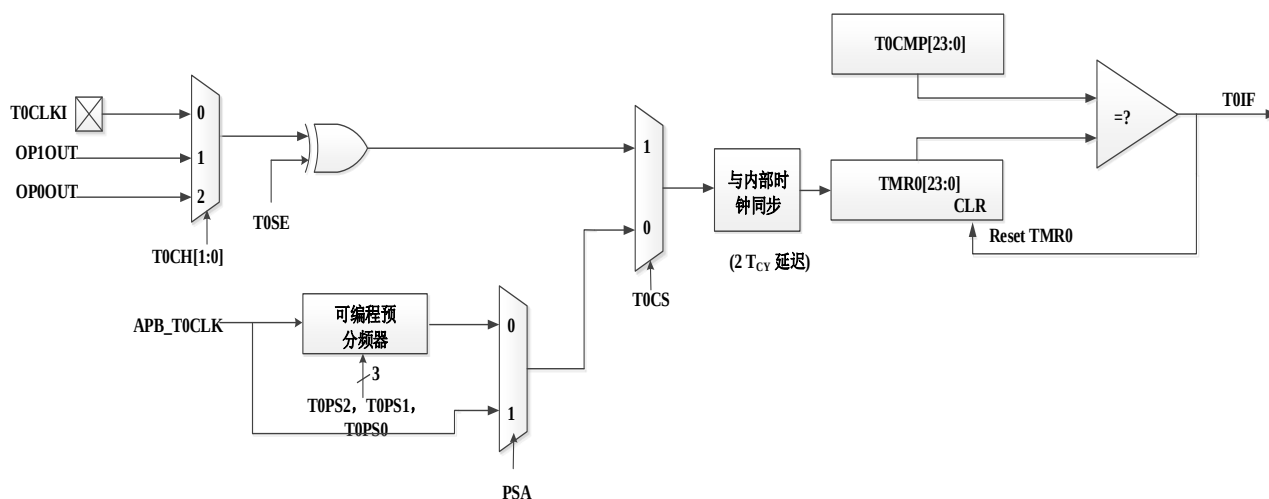
5.1 定时器 T0

5.1.1. 定时器 T0 特性

T0 模块定时/计数器具有如下特性：

- 24 位定时器/计数器，当 TMR0 计数与 T0CMP 值相等时，自动重新装载 0 值
- TMR0, T0CMP 可读写
- 3 位软件可编程的专用预分频器
- 可选择内部或外部时钟源
- 当计数器 TMR0 到 T0CMP 相等时时，产生中断信号 T0IF
- 时钟源可以选择 T0CKI, OP0OUT, OP1OUT, APB T0CLK
- 外部时钟的边沿选择

5.1.2. 定时器 T0 电路结构



定时器 T0 电路结构图

5.1.3. 定时器 T0 寄存器

基地址：0x4000 0000

5.1.3.1. T0CON 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	TODMAEN	Timer0 的 DMA 的使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
9	TOON	Timer0 定时器使能位 0: 禁止; 1: 使能;	R/W	0
8	TOIE	Timer0 中断使能位 0: 禁止; 1: 使能;	R/W	0
7:6	TOCH[1:0]	Timer0 外部时钟选择 00: TOCKI (注: TOCKI 的输入频率必须小于系统时钟的 1/8); 01: OP0OUT; 10: OP1OUT; 11: Reserved;	R/W	0
5	TOCS	Timer0 时钟源选择位 1: TOCH 输入信号作为时钟信号; 0: APB_T0_CLK 作为时钟信号;	R/W	0
4	TOSE	Timer0 时钟源边沿选择位 1: TOCKI 引脚下降沿传输时递增; 0: TOCKI 引脚上升沿传输时递增;	R/W	0
3	TOPSA	Timer0 预分频器分配位 1: Timer0 时钟输入避开预分频器输出; 0: Timer0 时钟输入来自预分频器输出;	R/W	0
2:0	TOPS[2:0]	Timer0 预分频比选择 111: 1:256 预分频值; 110: 1:128 预分频值; 101: 1:64 预分频值; 100: 1:32 预分频值; 011: 1:16 预分频值; 010: 1:8 预分频值; 001: 1:4 预分频值; 000: 1:2 预分频值;	R/W	0

5.1.3.2. TMR0 寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	TMR0	Timer0 计数器值;	R/W	0

5.1.3.3. T0CMP 寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	T0CMP	Timer0 比较值;	R/W	24' hffff_ff

5.1.3.4. T0I 寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	T0IF	Timer0 中断标志位, 写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R/W	0

5.2 定时器 T1

5.2.1. 定时器 T1 特性

T1 模块定时/计数器具有如下特性:

- 32 位定时器/计数器
- 可读写
- 可选择内部或外部时钟源
- 当计数器 TMR1 与 T1CMP 值相等时产生溢出中断
- CCP 模块特殊事件触发器复位 TMR1 计数器
- 可自动装载模式
- 时钟源可以选择 T1CKI, OP0OUT, OP1OUT, T1CLK
- 外部时钟的边沿选择
- 支持单脉冲模式、周期模式、反转输出模式和连续计数模式

5.2.2. 定时器 T1 操作模式（通过 T1CON 寄存器选择）

定时器控制器提供 4 种工作模式，单脉冲（One-Shot）模式、周期（Periodic）模式、反转（Toggle）和连续计数（Continuous Counting）模式。每种操作模式如下所述：

1. 单脉冲模式

如果定时器工作在单脉冲模式（T1M[1:0] = 00）且 T1ON 置 1，定时器的计数器开始上数。当 TMR1 的值等于定时器比较寄存器 T1CMP 的值，TMR1 的值被硬件清 0，然后定时器停止计数，并产生中断标志位 T1IF。如果 T1IE（中断使能位）置 1，则中断发生。

2. 周期模式

如果定时器工作在周期模式（T1M[1:0]=01）且 T1ON（定时器使能位）置 1，定时器计数器开始上数。一旦 TMR1 的值达到定时器比较寄存器 T1CMP 的值，TMR1 的值将被硬件清 0，计数器再次开始计数，并产生中断标志位 T1IF。同时如果 IE 使能，将发生定时器中断。该模式下定时器一直计数，并和 T1CMP 比较，直到 T1ON 被清 0。

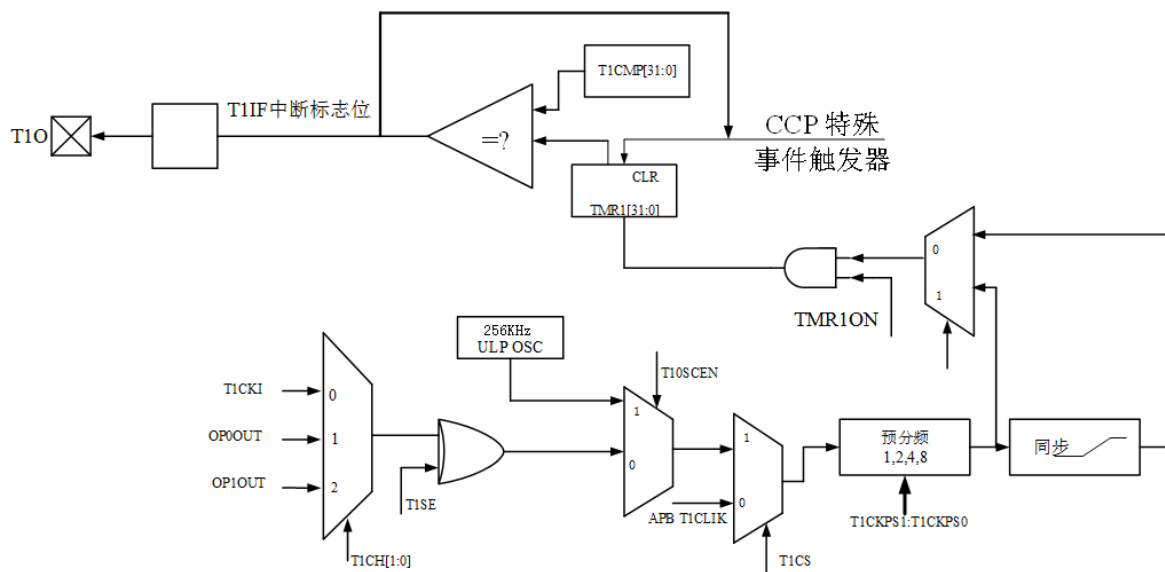
3. 反转输出模式

如果定时器工作在反转输出模式（T1M[1:0]=10）且 T1ON（定时器使能位）置 1，定时器/计数器开始向上计数。反转输出模式和周期模式几乎相同，除了当 TIF 位被置为 1 时反转输出将从 T1O 输出信号，就是说引脚上的信号将发生反转，所以反转输出模式输出的信号占空比是 50%。

4. 连续计数模式

如果定时器工作在连续计数模式（T1M[1:0]=11）且 T1ON（定时器使能位）置 1，定时器开始向上计数。一旦 TMR1 的值等于 T1CMP 的值，TIF 被置为 1 并且 TMR1 继续向上计数。同时，如果 IE 使能，定时器中断将发生，用户可以立即改变 T1CMP 的值而无需关闭计数器并重新计数。

5.2.3. 定时器 T1 电路结构



定时器 T1 电路结构图

5.2.4. 定时器 T1 寄存器

基地址：0x4000 1000

5.2.4.1. T1CON 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:13	Reserved		R	0
12	T1DMAEN	Timer1 的 DMA 的使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
11:10	T1M[1:0]	Timer1 计数器的工作模式选择 00: 单脉冲模式; 01: 周期模式; 10: 反转输出模式; 11: 连续计数模式;	R/W	0
9	T1ON	Timer1 定时器使能位 0: 禁止; 1: 使能;	R/W	0
8	T1IE	Timer1 中断使能位 0: 禁止; 1: 使能;	R/W	0
7:6	T1CH[1:0]	Timer1 外部时钟选择	R/W	0

		00: T1CKI (注: T0CKI 的输入频率必须小于系统时钟的 1/8); 01: OP0out; 10: OP1out; 11: Reserved;		
5	T1SE	Timer1 时钟源边沿选择位 1: T1CKI 引脚下降沿传输时递增; 0: T1CKI 引脚上升沿传输时递增;	R/W	0
4:3	T1PS[1:0]	Timer1 输入时钟预分频比选择 00: 1:1 预分频值; 01: 1:2 预分频值; 10: 1:4 预分频值; 11: 1:8 预分频值;	R/W	0
2	T1OSCEN	Timer1 256K ULP OSC 使能位 1: 使能; 0: 禁止;	R/W	0
1	T1SYNC	Timer1 外部时钟输入同步选择位 如果 TMR1CS = 1: 1: 不同步外部时钟输入 0: 同步外部时钟输入 如果 TMR1CS = 0: 此位被忽略, TMR1CS=0 时 Timer1 使用内部时钟;	R/W	0
0	T1CS	Timer1 时钟源选择位 1: 来自 T1CKI 引脚的外部时钟 (上升沿计数) 或者 256K OSC 时钟, 根据 T1OSCEN 值确认; 0: 内部时钟 (APB T1CLK);	R/W	0

5.2.4.2. TMR1 寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:0	TMR1	Timer1 计数器值;	R/W	0

5.2.4.3. T1CMP 寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:0	T1CMP	Timer1 计数器比较值;	R/W	0

5.2.4.4. T1I 寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
0	T1IF	Timer1 中断标志，写 1 清中断标志 0：未产生中断； 1：产生中断；	R/W	0

5.3 定时器 T2

5.3.1. 定时器 T2 特性

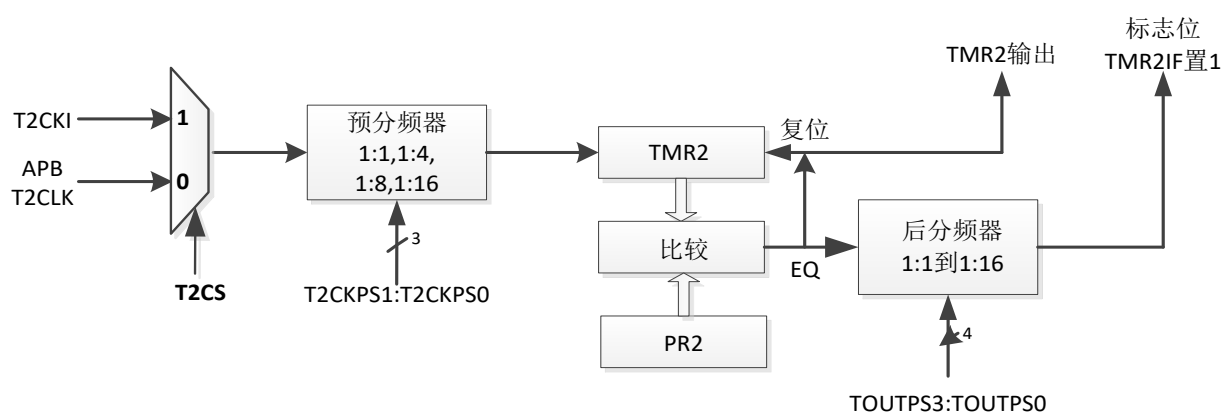
T2 模块定时/计数器具有如下特性：

- 16 位定时器（TMR2 寄存器）
- 16 位周期寄存器（PR2）
- 可读写（TMR2 和 PR2 寄存器均可）
- 可软件编程的预分频器（1:1、1:4、1:8 和 1:16）
- 可软件编程的后分频器（1:1 到 1:16）

注：如“定时器 T2 电路结构”所示，后分频影响中断触发情况。当后分频为 2 的情况下，计数器 TMR2 完成两次从 0 计数到 PR2 时，触发中断标志。以此类推

- TMR2 与 PR2 匹配时中断
- CCP 模块可选用 TMR2 输出来产生时钟位移

5.3.2. 定时器 T2 电路结构



定时器 T2 电路结构图

5.3.3. 定时器 T2 寄存器

基地址：0x4000 2000

5.3.3.1. T2CON 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	T2DMAEN	Timer2 的 DMA 的使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
9	T2TRI	Timer2 自减模式使能位 1: 使能; 0: 禁止;	R/W	0
8	T2CS	Timer2 时钟源选择位 0: 内部 APB T2CLK; 1: 外部引脚 T2CKI 输入; 注: T2CKI 的输入频率必须小于系统时钟的 1/8;	R/W	0
7	T2IE	Timer2 中断使能位 0: 禁止; 1: 使能;	R/W	0
6:3	T2OUTPS[3:0]	Timer2 输出后分频比选择 0000: 1:1 后分频值; 0001: 1:2 后分频值; 0010: 1:3 后分频值; 0011: 1:4 后分频值; 0100: 1:5 后分频值; 0101: 1:6 后分频值; 0110: 1:7 后分频值; 0111: 1:8 后分频值; 1000: 1:9 后分频值; 1001: 1:10 后分频值; 1010: 1:11 后分频值; 1011: 1:12 后分频值; 1100: 1:13 后分频值; 1101: 1:14 后分频值; 1110: 1:15 后分频值; 1111: 1:16 后分频值;	R/W	0
2	T2ON	Timer2 使能位 1: 使能; 0: 禁止;	R/W	0
1:0	T2PS[1:0]	Timer2 时钟预分频比选择 00: 1:1 预分频值; 01: 1:4 预分频值; 10: 1:8 预分频值;	R/W	0

		11: 1:16 预分频值;		
--	--	----------------	--	--

5.3.3.2. TMR2 寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:0	TMR2	Timer2 计数器比较值;	R/W	0

5.3.3.3. PR2 寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	PR2	PR2 设定值;	R/W	0xffff

5.3.3.4. T2I 寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
0	T2IF	Timer2 中断标志, 写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R/W	0

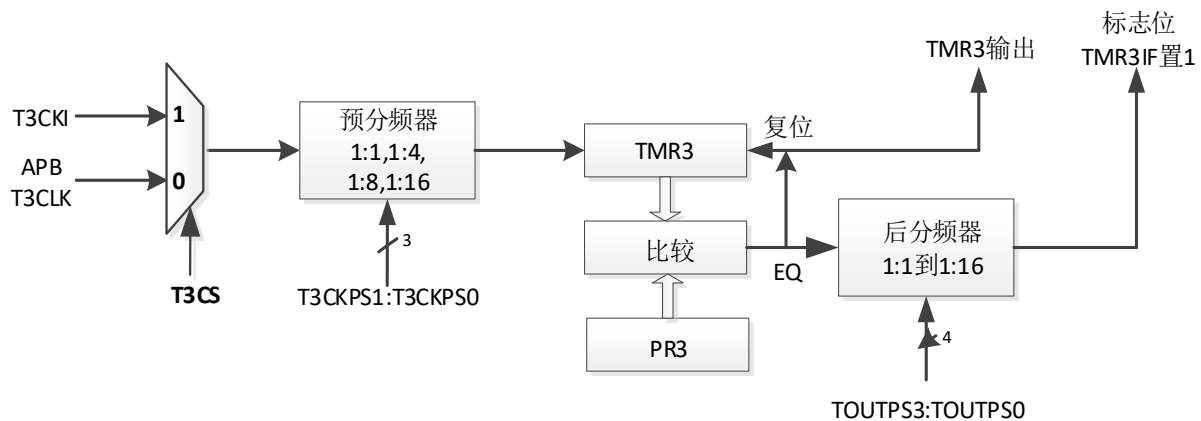
5.4 定时器 T3

5.4.1. 定时器 T3 特性

T3 模块定时/计数器具有如下特性:

- 16 位定时器 (TMR3 寄存器)
- 16 位周期寄存器 (PR3)
- 可读写 (TMR3 和 PR3 寄存器均可)
- 可软件编程的预分频器 (1:1、1:4、1:8 和 1:16)
- 可软件编程的后分频器 (1:1 到 1:16);【注: 后分频的详细说明见: T2 特性说明小节】
- TMR3 与 PR3 匹配时中断

5.4.2. 定时器 T3 电路结构



5.4.3. 定时器 T3 寄存器

基地址：0x4000 B000

5.4.3.1. T3CON 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	T3DMAEN	Timer3 的 DMA 的使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
9	T3TRI	Timer3 自减模式使能位 1: 使能; 0: 禁止;	R/W	0
8	T3CS	Timer3 时钟源选择位 0: 内部 APB T3CLK; 1: 外部引脚 T3CKI 输入; 注: T3CKI 的输入频率必须小于系统时钟的 1/8;	R/W	0
7	T3IE	Timer3 中断使能位 0: 禁止; 1: 使能;	R/W	0
6:3	T3OUTPS[3:0]	Timer3 输出后分频比选择 0000: 1:1 后分频值; 0001: 1:2 后分频值; 0010: 1:3 后分频值; 0011: 1:4 后分频值; 0100: 1:5 后分频值; 0101: 1:6 后分频值;	R/W	0

		0110: 1:7 后分频值; 0111: 1:8 后分频值; 1000: 1:9 后分频值; 1001: 1:10 后分频值; 1010: 1:11 后分频值; 1011: 1:12 后分频值; 1100: 1:13 后分频值; 1101: 1:14 后分频值; 1110: 1:15 后分频值; 1111: 1:16 后分频值;		
2	T3ON	Timer3 使能位 1: 使能; 0: 禁止;	R/W	0
1:0	T3PS[1:0]	Timer3 时钟预分频比选择 00: 1:1 预分频值; 01: 1:4 预分频值; 10: 1:8 预分频值; 11: 1:16 预分频值;	R/W	0

5.4.3.2. TMR3 寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TMR3	Timer3 计数器值;	R/W	0

5.4.3.3. PR3 寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	PR3	PR3 设定值;	R/W	0xffff

5.4.3.4. T3I 寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
0	T3IF	Timer3 中断标志, 写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R/W	0

6 看门狗定时器

6.1 看门狗定时器 (WDT)

6.1.1. 看门狗定时器概述

若系统进入了错误的工作状态，看门狗定时器可以在合理的时间范围内使 MCU 复位。使能看门狗后，若用户程序清除看门狗定时器失败，则在预订的时间范围内，看门狗会使系统复位或者产生一个中断。

看门狗模块由 128 倍预分频的分频器和 8 位计数器组成，时钟经预分频后提供给定时器。定时器递增计数。因此，看门狗的最小间隔为 $TWDT_PCLK \times 128 \times 1$ ，最大间隔为 $TWDT_PCLK \times 128 \times 256$ 。

看门狗按照下列方法进行操作：

1. 通过 WDTCLKSEL 寄存器为看门狗选择时钟源；
2. 通过 APB 时钟预分频寄存器 1 (SYS1_APBCLKP1) 的 WDTPRE 位为看门狗设置预分频值；
3. 在 WDT_TC 寄存器中设置看门狗定时器的恒定重装值；
4. 使能看门狗，通过 WDT_CFG 寄存器设置看门狗定时器的操作模式；
5. 写入 0X55AA 到 WDT_FEED 寄存器清零看门狗，防止看门狗计数器溢出引起复位或产生中断。

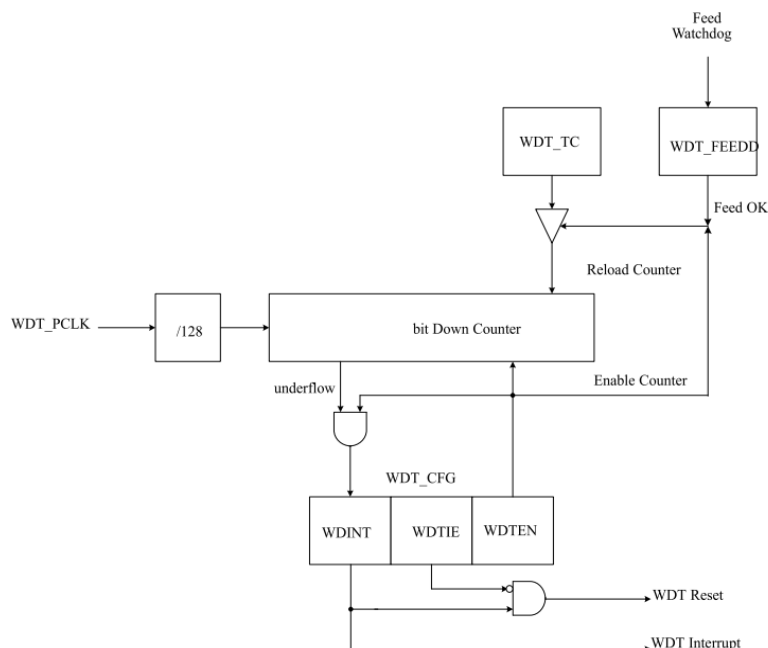
设置 WDT_CFG 寄存器的 WDTEN 位时，看门狗开始运行，时间恒定值装入看门狗计数器，计数器开始递增计数。看门狗在复位模式下工作时，计数器溢出后，CPU 复位，随着外部复位，系统从向量表中将数据导入堆栈指针和程序计数器中。不管何时写入 0x55AA 到 WDT_FEED 寄存器中，WDT_TC 的值装入看门狗计数器，并阻止看门狗复位或者中断。

看门狗定时器的时钟有两种：HCLK 和 WDT_PCLK。HCLK 用于 AHB 访问看门狗寄存器，并由系统时钟衍生；WDT_PCLK 用于看门狗定时器计数。下列几种时钟都可用作 WDT_PCLK 的时钟源：PCLKPRE，IHRC，ILRC，High XTAL，Low XTAL。

注：

1. 可在 AHB 时钟使能寄存器 (SYS1_AHBCLKEN) 中禁止看门狗寄存器模块以省电。
2. 当看门狗正在运行时，任何时候都可能发生看门狗中断或复位。

6.1.2. 看门狗定时器电路结构



看门狗定时器电路结构图

6.1.3. 看门狗定时器寄存器

基地址：0x4000 4000

6.1.3.1. WDTCON（看门狗配置寄存器）

WDTCON 寄存器通过 WDTEN 和 WDTIE 位控制看门狗的操作。该寄存器显示看门狗定时器中断的 Raw 状态。若 WDTINT 和 WDTIE 位都置 1，则 WDT 中断发送到中断控制器中。

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	WDTINT	看门狗中断标志位，写 1 将该位清零，清零前软件重置看门狗； 0：未产生中断； 1：产生中断（当且仅当 WDTIE = 1）；	R/W	0
1	WDTIE	看门狗中断使能位 0：看门复位后，将 WDTINT 位清零； 1：看门复位后，产生中断（看门狗中断模式）；	R/W	0
0	WDTEN	看门狗使能位 0：禁止；	R/W	0

		1: 使能（同时将 WDT_TC 的值装入看门狗计数器）		
--	--	------------------------------	--	--

6.1.3.2. 看门狗时钟源寄存器（WDTCLK）

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2:0	CLKSEL[2:0]	选择看门狗时钟源 000: PCLKPRE; 001: IHRC; 010: ILRC; 011: EHS X' TAL; 100: ELS X' TAL;	R/W	0

6.1.3.3. WDTTC（看门狗定时器常量寄存器）

WDT_TC 寄存器决定超时的时间，每次清看门狗，都会重装 WDT_TC 的数值到看门狗定时器中。它是一个 8 位计数器，定时器的超时时间为： $TWDT_PCLK \times 128 \times 1 \sim TWDT_PCLK \times 128 \times 256$ 。

$$\begin{aligned} \text{看门狗的溢出时间} &= (30.5176\mu s \times 1) \times 128 \times 1 \sim (0.0625ms \times 32) \times 128 \times 256 \\ &= 3.906ms \sim 65536ms \end{aligned}$$

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	TC[7:0]	看门狗定时器恒定重装载值 = $TC[7:0] + 1$ 0000 0000: 定时器恒定值 = 1; 0000 0001: 定时器恒定值 = 2; 1111 1110: 定时器恒定值 = 255; 1111 1111: 定时器恒定值 = 256;	R/W	0

6.1.3.4. WDTFEED（看门狗 Feed 寄存器）

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	FV[15:0]	写看门狗 Feed 值（读取为 0）; 写 0x55AA: 清看门狗，WDT_TC 的值重载到看门狗计数	W	0

		器；		
--	--	----	--	--

6.2 独立看门狗定时器（IWDT）

6.2.1. 独立看门狗定时器概述

若系统进入了错误的工作状态，独立看门狗可以在合理的时间范围内使 MCU 复位。使能独立看门狗时，若用户程序清除独立看门狗定时器失败，则在预订的时间范围内，独立看门狗会使系统复位。

独立看门狗由预分频器和 16 位计数器组成，时钟经过预分频后提供给定时器，定时器开始递增计数。

通过向 PMU_ANA0 寄存器的 IWDTEN 位写 1，同时向 IWDT_FEED 寄存器写入 0xAA55 来使能 IWDT 开始运行，通过配置 PMU_ANA0 寄存器的 IWDTDIV 位选择计数时间，计数器开始递增计数。独立看门狗在复位模式下工作时，计数器溢出后，CPU 复位，随着外部复位，系统从向量表中将数据导入堆栈指针和程序计数器中。

不管何时写入 0xAA55 到 IWDT_FEED 寄存器中，WDT_TC 的值装入独立看门狗计数器，并阻止看门狗复位。当独立看门狗正在运行时，任何时候都可能发生独立看门狗复位。

6.2.2. 独立看门狗定时器寄存器

6.2.2.1. IWDTFEED（独立看门狗定时器配置寄存器）

基地址：0x4000 4000

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	FV	写内部看门狗 Feed 值（读取为 0）； 写 0xAA55：清内部看门狗，WDT_TC 的值重载到看门狗计数器；	R/W	0

注：使用 SWD 接口后 IWDT 无效。IWDT 只有在 Bootloader 下载模式下有效。

7 CCP 捕捉/比较/PWM

7.1 CCP 概述

捕捉/比较/PWM (CCP) 模块包含 6 组 32 位寄存器，它们可被用作：6 个 32 位捕捉寄存器、6 个 32 位比较寄存器、6 个 PWM 主/从占空比寄存器。捕捉/比较/PWM 寄存器 CCPR_n 由两个 16 位寄存器组成：CCPR_nL (低 16bit) 和 CCPR_nH (高 16bit)。CCPCON_n 寄存器控制 CCP_n 的操作。比较匹配将产生特殊事件触发信号，该信号会使 TMR1 寄存器清零。

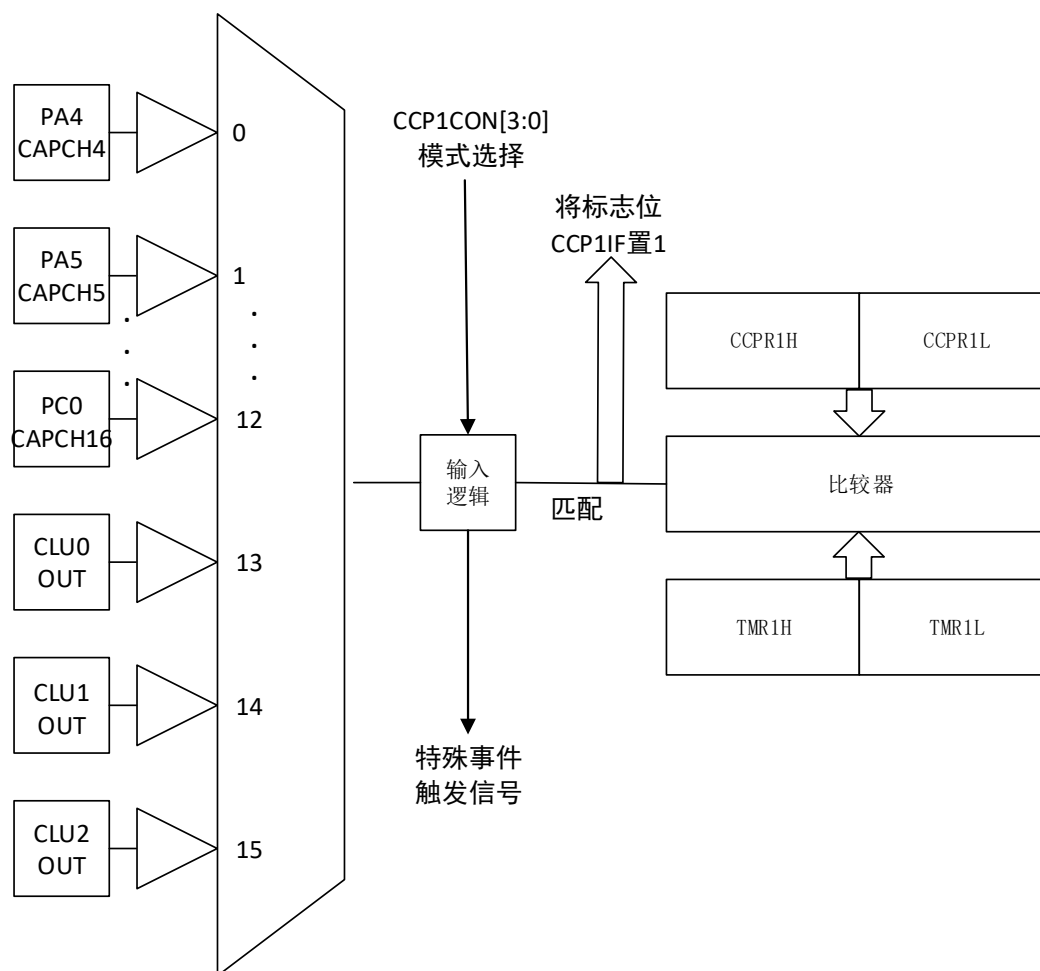
7.2 捕捉/比较模式

在捕捉模式下，当引脚 CCP_n 发生事件时，CCPR_nH:CCPR_nL 将捕捉 TMR1 寄存器的 32 位值。

在比较模式下，CCPR_n 寄存器的 16 位值不断与一对 TMR1 寄存器的值进行比较。

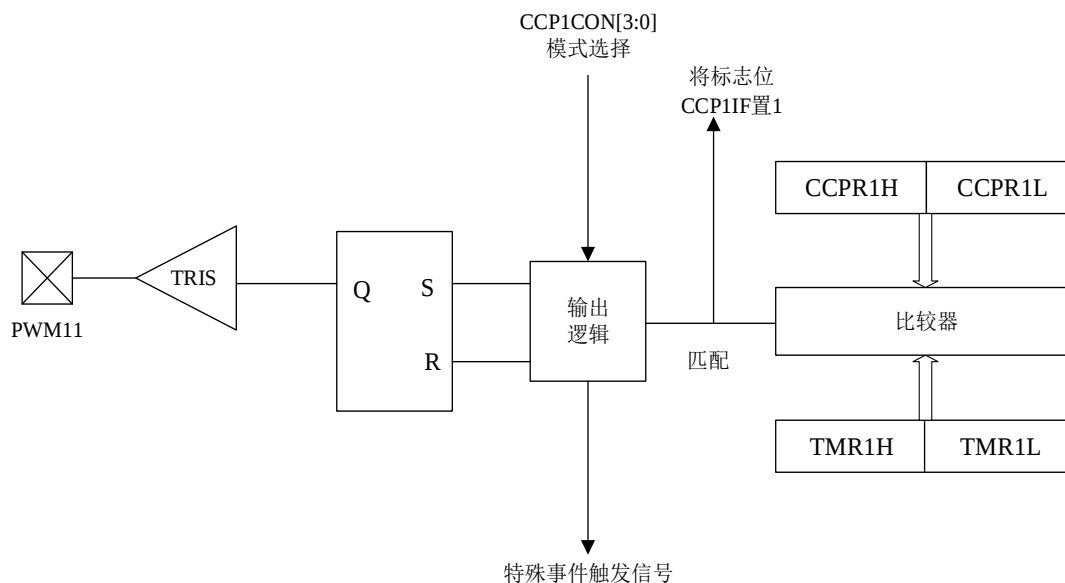
捕捉/比较事件由 CCPCON_n[3:0] 进行配置。

7.2.1. 捕捉模式电路结构



捕捉模式电路结构图（以 CCP1 为例）

7.2.2. 比较模式电路结构



比较模式电路结构图（以 CCP1 为例）

7.2.3. CCP 寄存器

基地址：0x4000 3000

7.2.3.1. CCPR0 寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:16	CCPR0H	CCPR0 计数器值高位；	R/W	0
15:0	CCPR0L	CCPR0 计数器值低位；	R/W	0

7.2.3.2. CCPR1 寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:16	CCPR1H	CCPR1 计数器值高位；	R/W	0
15:0	CCPR1L	CCPR1 计数器值低位；	R/W	0

7.2.3.3. CCPR2 寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:16	CCPR2H	CCPR2 计数器值高位;	R/W	0
15:0	CCPR2L	CCPR2 计数器值低位;	R/W	0

7.2.3.4. CCPR3 寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:16	CCPR3H	CCPR3 计数器值高位;	R/W	0
15:0	CCPR3L	CCPR3 计数器值低位;	R/W	0

7.2.3.5. CCPR4 寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:16	CCPR4H	CCPR4 计数器值高位;	R/W	0
15:0	CCPR4L	CCPR4 计数器值低位;	R/W	0

7.2.3.6. CCPR5 寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:16	CCPR5H	CCPR5 计数器值高位;	R/W	0
15:0	CCPR5L	CCPR5 计数器值低位;	R/W	0

7.2.3.7. CCPCON0 寄存器 (CCP 控制寄存器 0)

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCPODMA	CCP0 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
8	CCPOIE	CCP0 中断使能位 1: 使能; 0: 禁止;	R/W	0
7:4	CCPOCH	CCP0 捕获触发信号选择;	R/W	0
3:0	CCPOM	CCP0 模式选择;	R/W	0

CCP0CH 配置表:

BIT[7:4]	CCP0CH[3:0]: CCP0 捕获触发信号选择
0000	外部 CAPCH0 (PA0) 引脚输入信号;
0001	外部 CAPCH1 (PA1) 引脚输入信号;
0010	外部 CAPCH2 (PA2) 引脚输入信号;
0011	外部 CAPCH3 (PA3) 引脚输入信号;
0100	外部 CAPCH4 (PA4) 引脚输入信号;
0101	外部 CAPCH5 (PA5) 引脚输入信号;
0110	外部 CAPCH6 (PA6) 引脚输入信号;
0111	外部 CAPCH7 (PA7) 引脚输入信号;
1000	外部 CAPCH8 (PB0) 引脚输入信号;
1001	外部 CAPCH9 (PB1) 引脚输入信号;
1010	外部 CAPCH10 (PB2) 引脚输入信号;
1011	外部 CAPCH11 (PB3) 引脚输入信号;
1100	外部 CAPCH12 (PB4) 引脚输入信号;
1101	CRYH 时钟输出信号;
1110	CRYL 时钟输出信号;
1111	ILRC 时钟输出信号;

CCP0M 配置表:

BIT[3:0]	工作模式	CCP0M[3:0]: CCP0 模式选择
0000	禁止模式	禁止捕捉/比较/PWM (复位 CCP0 模块);
0100	比较模式	选择 CCPR0 匹配时将输出置为高电平 (CCP0IF 位置 1), 输出引脚为 PA0/PB0/PC0, TIMER1 溢出时, PA0/PB0/PC0 为低电平;
0101	比较模式	选择 CCPR0 匹配时将输出置为低电平 (CCP0IF 位置 1), 输出引脚为 PA0/PB0/PC0, TIMER1 溢出时, PA0/PB0/PC0 为高电平;
0110	比较模式	选择 CCPR0 匹配时将产生软件中断 (CCP0IF 位置 1, 但 PA0/PB0/PC0 引脚不受影响);
0111	比较模式	选择 CCPR0 触发特殊事件 (CCP0IF 位置 1, PA0/PB0/PC0 引脚不受影响); 清零 Timer1, 并启动 ADC 采集 (如果 ADCON=1);
1000	捕捉模式	每个下降沿发生, 捕捉值存入 CCPR0 寄存器, 并产生中断标志 CCP0IF;
1001	捕捉模式	每个上升沿发生, 捕捉值存入 CCPR0 寄存器, 并产生中断标志 CCP0IF;
1010	捕捉模式	每 4 个上升沿发生一次, 捕捉值存入 CCPR0 寄存器, 并产生中断标志 CCP0IF;

1011	捕捉模式	每 16 个上升沿发生一次，捕捉值存入 CCP0 寄存器，并产生中断标志 CCP0IF；
1100	捕捉模式	下降沿复位 TIMER1，捕捉值存入 CCP0 寄存器，并产生中断标志 CCP0IF；
1101	捕捉模式	上升沿复位 TIMER1，捕捉值存入 CCP0 寄存器，并产生中断标志 CCP0IF；
1110	捕捉模式	下降沿捕捉值存入 CCP0 寄存器，不产生中断标志 CCP0IF；
1111	捕捉模式	上升沿捕捉值存入 CCP0 寄存器，不产生中断标志 CCP0IF；
001X	PWM 模式	启动 CCP0 PWM 模式；

7.2.3.8. CCPCON1 寄存器（CCP 控制寄存器 1）

偏移地址：0x01C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCP1DMA	CCP1 的 DMA 使能位 1: DMA 使能； 0: DMA 禁止；	R/W	0
8	CCP1IE	CCP1 中断使能位 1: 使能； 0: 禁止；	R/W	0
7:4	CCP1CH	CCP1 捕获触发信号选择；	R/W	0
3:0	CCP1M	CCP1 模式选择；	R/W	0

CCP1CH 配置表：

BIT[7:4]	CCP1CH[3:0]：CCP1 捕获触发信号选择
0000	外部 CAPCH4 (PA4) 引脚输入信号；
0001	外部 CAPCH5 (PA5) 引脚输入信号；
0010	外部 CAPCH6 (PA6) 引脚输入信号；
0011	外部 CAPCH7 (PA7) 引脚输入信号；
0100	外部 CAPCH8 (PB0) 引脚输入信号；
0101	外部 CAPCH9 (PB1) 引脚输入信号；
0110	外部 CAPCH10 (PB2) 引脚输入信号；
0111	外部 CAPCH11 (PB3) 引脚输入信号；
1000	外部 CAPCH12 (PB4) 引脚输入信号；
1001	外部 CAPCH13 (PB5) 引脚输入信号；

1010	外部 CAPCH14 (PB6) 引脚输入信号;
1011	外部 CAPCH15 (PB7) 引脚输入信号;
1100	外部 CAPCH16 (PC0) 引脚输入信号;
1101	CLU00OUT 的输出信号;
1110	CLU10OUT 的输出信号;
1111	CLU20OUT 的输出信号;

CCP1M 配置表:

BIT[3:0]	工作模式	CCP1M[3:0]: CCP1 模式选择
0000	禁止模式	禁止捕捉/比较/PWM (复位 CCP1 模块);
0100	比较模式	选择 CCPR1 匹配时将输出置为高电平 (CCP1IF 位置 1), 输出引脚为 PA1/PB1/PC1, TIMER1 溢出时, PA1/PB1/PC1 为低电平;
0101	比较模式	选择 CCPR1 匹配时将输出置为低电平 (CCP1IF 位置 1), 输出引脚为 PA1/PB1/PC1, TIMER1 溢出时, PA1/PB1/PC1 为高电平;
0110	比较模式	选择 CCPR1 匹配时将产生软件中断 (CCP1IF 位置 1, 但 PA1/PB1/PC1 引脚不受影响);
0111	比较模式	选择 CCPR1 触发特殊事件 (CCP1IF 位置 1, PA1/PB1/PC1 引脚不受影响); 清零 Timer1, 并启动 ADC 采集 (如果 ADCON=1);
1000	捕捉模式	每个下降沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1001	捕捉模式	每个上升沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1010	捕捉模式	每 4 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1011	捕捉模式	每 16 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1100	捕捉模式	下降沿复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1101	捕捉模式	上升沿复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCP1IF;
1110	捕捉模式	下降沿捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF;
1111	捕捉模式	上升沿捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCP1IF;
001X	PWM 模式	启动 CCP1 PWM 模式;

7.2.3.9. CCPCON2 寄存器 (CCP 控制寄存器 2)

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCP2DMA	CCP2 的 DMA 使能位	R/W	0

		1: DMA 使能; 0: DMA 禁止;		
8	CCP2IE	CCP2 中断使能位 1: 使能; 0: 禁止;	R/W	0
7:4	CCP2CH	CCP2 捕获触发信号选择;	R/W	0
3:0	CCP2M	CCP2 模式选择;	R/W	0

CCP2CH 配置表:

BIT[7:4]	CCP2CH[3:0]: CCP2 捕获触发信号选择
0000	外部 CAPCH8 (PB0) 引脚输入信号;
0001	外部 CAPCH9 (PB1) 引脚输入信号;
0010	外部 CAPCH10 (PB2) 引脚输入信号;
0011	外部 CAPCH11 (PB3) 引脚输入信号;
0100	外部 CAPCH12 (PB4) 引脚输入信号;
0101	外部 CAPCH13 (PB5) 引脚输入信号;
0110	外部 CAPCH14 (PB6) 引脚输入信号;
0111	外部 CAPCH15 (PB7) 引脚输入信号;
1000	外部 CAPCH16 (PC0) 引脚输入信号;
1001	外部 CAPCH17 (PC1) 引脚输入信号;
1010	外部 CAPCH18 (PC2) 引脚输入信号;
1011	外部 CAPCH19 (PC3) 引脚输入信号;
1100	外部 CAPCH20 (PC4) 引脚输入信号;
1101	CLU3OUT 的输出信号;
1110	OP0OUT 的输出信号;
1111	OP1OUT 的输出信号;

CCP2M 配置表:

BIT[3:0]	工作模式	CCP2M[3:0]: CCP2 模式选择
0000	禁止模式	禁止捕捉/比较/PWM (复位 CCP2 模块);
0100	比较模式	选择 CCPR2 匹配时将输出置为高电平 (CCP2IF 位置 1), 输出引脚为 PA2/PB2/PC2, TIMER1 溢出时, PA2/PB2/PC2 为低电平;
0101	比较模式	选择 CCPR2 匹配时将输出置为低电平 (CCP2IF 位置 1), 输出引脚为 PA2/PB2/PC2, TIMER1 溢出时, PA2/PB2/PC2 为高电平;

0110	比较模式	选择 CCPR2 匹配时将产生软件中断 (CCP2IF 位置 1, 但 PA2/PB2/PC2 引脚不受影响);
0111	比较模式	选择 CCPR2 触发特殊事件 (CCP2IF 位置 1, PA2/PB2/PC2 引脚不受影响); 清零 Timer1, 并启动 ADC 采集 (如果 ADCON=1);
1000	捕捉模式	每个下降沿发生, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1001	捕捉模式	每个上升沿发生, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1010	捕捉模式	每 4 个上升沿发生一次, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1011	捕捉模式	每 16 个上升沿发生一次, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1100	捕捉模式	下降沿复位 TIMER1, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1101	捕捉模式	上升沿复位 TIMER1, 捕捉值存入 CCPR2 寄存器, 并产生中断标志 CCP2IF;
1110	捕捉模式	下降沿捕捉值存入 CCPR2 寄存器, 不产生中断标志 CCP2IF;
1111	捕捉模式	上升沿捕捉值存入 CCPR2 寄存器, 不产生中断标志 CCP2IF;
001X	PWM 模式	启动 CCP2 PWM 模式;

7.2.3.10. CCPCON3 寄存器 (CCP 控制寄存器 3)

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCP3DMA	CCP3 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
8	CCP3IE	CCP3 中断使能位 1: 使能; 0: 禁止;	R/W	0
7:4	CCP3CH	CCP3 捕获触发信号选择;	R/W	0
3:0	CCP3M	CCP3 模式选择;	R/W	0

CCP3CH 配置表:

BIT[7:4]	CCP3CH[3:0]: CCP3 捕获触发信号选择
0000	外部 CAPCH12 (PB4) 引脚输入信号;
0001	外部 CAPCH13 (PB5) 引脚输入信号;
0010	外部 CAPCH14 (PB6) 引脚输入信号;
0011	外部 CAPCH15 (PB7) 引脚输入信号;

0100	外部 CAPCH16 (PC0) 引脚输入信号;
0101	外部 CAPCH17 (PC1) 引脚输入信号;
0110	外部 CAPCH18 (PC2) 引脚输入信号;
0111	外部 CAPCH19 (PC3) 引脚输入信号;
1000	外部 CAPCH20 (PC4) 引脚输入信号;
1001	外部 CAPCH21 (PC5) 引脚输入信号;
1010	外部 CAPCH22 (PC6) 引脚输入信号;
1011	外部 CAPCH23 (PD0) 引脚输入信号;
1100	外部 CAPCH24 (PD1) 引脚输入信号;
1101	外部 CAPCH25 (PD2) 引脚输入信号;
1110	外部 CAPCH26 (PD3) 引脚输入信号;
1111	外部 CAPCH27 (PD4) 引脚输入信号;

CCP3M 配置表:

BIT[3:0]	工作模式	CCP3M[3:0]: CCP3 模式选择
0000	禁止模式	禁止捕捉/比较/PWM (复位 CCP3 模块);
0100	比较模式	选择 CCPR3 匹配时将输出置为高电平 (CCP3IF 位置 1), 输出引脚为 PA3/PB3/PC3, TIMER1 溢出时, PA3/PB3/PC3 为低电平;
0101	比较模式	选择 CCPR3 匹配时将输出置为低电平 (CCP3IF 位置 1), 输出引脚为 PA3/PB3/PC3, TIMER1 溢出时, PA3/PB3/PC3 为高电平;
0110	比较模式	选择 CCPR3 匹配时将产生软件中断 (CCP3IF 位置 1, 但 PA3/PB3/PC3 引脚不受影响);
0111	比较模式	选择 CCPR3 触发特殊事件 (CCP3IF 位置 1, PA3/PB3/PC3 引脚不受影响); 清零 Timer1, 并启动 ADC 采集 (如果 ADCON=1);
1000	捕捉模式	每个下降沿发生, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1001	捕捉模式	每个上升沿发生, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1010	捕捉模式	每 4 个上升沿发生一次, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1011	捕捉模式	每 16 个上升沿发生一次, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1100	捕捉模式	下降沿复位 TIMER1, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1101	捕捉模式	上升沿复位 TIMER1, 捕捉值存入 CCPR3 寄存器, 并产生中断标志 CCP3IF;
1110	捕捉模式	下降沿捕捉值存入 CCPR3 寄存器, 不产生中断标志 CCP3IF;
1111	捕捉模式	上升沿捕捉值存入 CCPR3 寄存器, 不产生中断标志 CCP3IF;

001X	PWM 模式	启动 CCP3 PWM 模式;
------	--------	-----------------

7.2.3.11. CCPCON4 寄存器 (CCP 控制寄存器 4)

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCP4DMA	CCP4 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
8	CCP4IE	CCP4 中断使能位 1: 使能; 0: 禁止;	R/W	0
7:4	CCP4CH	CCP4 捕获触发信号选择;	R/W	0
3:0	CCP4M	CCP4 模式选择;	R/W	0

CCP4CH 配置表:

BITS[7:4]	CCP4CH[3:0]: CCP4 捕获触发信号选择
0000	外部 CAPCH0 (PA0) 引脚输入信号;
0001	外部 CAPCH1 (PA1) 引脚输入信号;
0010	外部 CAPCH2 (PA2) 引脚输入信号;
0011	外部 CAPCH3 (PA3) 引脚输入信号;
0100	外部 CAPCH4 (PA4) 引脚输入信号;
0101	外部 CAPCH5 (PA5) 引脚输入信号;
0110	外部 CAPCH6 (PA6) 引脚输入信号;
0111	外部 CAPCH7 (PA7) 引脚输入信号;
1000	外部 CAPCH8 (PB0) 引脚输入信号;
1001	外部 CAPCH9 (PB1) 引脚输入信号;
1010	外部 CAPCH10 (PB2) 引脚输入信号;
1011	外部 CAPCH11 (PB3) 引脚输入信号;
1100	外部 CAPCH12 (PB4) 引脚输入信号;
1101	外部 CAPCH13 (PB5) 引脚输入信号;
1110	外部 CAPCH14 (PB6) 引脚输入信号;
1111	外部 CAPCH15 (PB7) 引脚输入信号;

CCP4M 配置表:

BIT[3:0]	工作模式	CCP4M[3:0]: CCP4 模式选择
0000	禁止模式	禁止捕捉/比较/PWM（复位 CCP4 模块）；
0100	比较模式	选择 CCPR4 匹配时将输出置为高电平（CCP4IF 位置 1），输出引脚为 PA4/PB4/PC4，TIMER1 溢出时，PA4/PB4/PC4 为低电平；
0101	比较模式	选择 CCPR4 匹配时将输出置为低电平（CCP4IF 位置 1），输出引脚为 PA4/PB4/PC4，TIMER1 溢出时，PA4/PB4/PC4 为高电平；
0110	比较模式	选择 CCPR4 匹配时将产生软件中断（CCP4IF 位置 1，但 PA4/PB4/PC4 引脚不受影响）；
0111	比较模式	选择 CCPR4 触发特殊事件（CCP4IF 位置 1，PA4/PB4/PC4 引脚不受影响）；清零 Timer1，并启动 ADC 采集（如果 ADCON=1）；
1000	捕捉模式	每个下降沿发生，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1001	捕捉模式	每个上升沿发生，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1010	捕捉模式	每 4 个上升沿发生一次，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1011	捕捉模式	每 16 个上升沿发生一次，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1100	捕捉模式	下降沿复位 TIMER1，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1101	捕捉模式	上升沿复位 TIMER1，捕捉值存入 CCPR4 寄存器，并产生中断标志 CCP4IF；
1110	捕捉模式	下降沿捕捉值存入 CCPR4 寄存器，不产生中断标志 CCP4IF；
1111	捕捉模式	上升沿捕捉值存入 CCPR4 寄存器，不产生中断标志 CCP4IF；
001X	PWM 模式	启动 CCP4 PWM 模式；

7.2.3.12. CCPCON5 寄存器（CCP 控制寄存器 5）

偏移地址：0x02C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9	CCP5DMA	CCP5 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
8	CCP5IE	CCP5 中断使能位 1: 使能; 0: 禁止;	R/W	0
7:4	CCP5CH	CCP5 捕获触发信号选择;	R/W	0
3:0	CCP5M	CCP5 模式选择;	R/W	0

CCP5CH 配置表:

BIT[7:4]	CCP5CH[3:0]: CCP5 捕获触发信号选择
0000	外部 CAPCH14 (PB6) 引脚输入信号;
0001	外部 CAPCH15 (PB7) 引脚输入信号;
0010	外部 CAPCH16 (PC0) 引脚输入信号;
0011	外部 CAPCH17 (PC1) 引脚输入信号;
0100	外部 CAPCH18 (PC2) 引脚输入信号;
0101	外部 CAPCH19 (PC3) 引脚输入信号;
0110	外部 CAPCH20 (PC4) 引脚输入信号;
0111	外部 CAPCH21 (PC5) 引脚输入信号;
1000	外部 CAPCH22 (PC6) 引脚输入信号;
1001	外部 CAPCH23 (PD0) 引脚输入信号;
1010	外部 CAPCH24 (PD1) 引脚输入信号;
1011	外部 CAPCH25 (PD2) 引脚输入信号;
1100	外部 CAPCH26 (PD3) 引脚输入信号;
1101	外部 CAPCH27 (PD4) 引脚输入信号;
1110	外部 CAPCH28 (PD5) 引脚输入信号;
1111	外部 CAPCH29 (PD6) 引脚输入信号;

CCP5M 配置表:

BIT[3:0]	工作模式	CCP5M[3:0]: CCP5 模式选择
0000	禁止模式	禁止捕捉/比较/PWM (复位 CCP5 模块);
0100	比较模式	选择 CCPR5 匹配时将输出置为高电平 (CCP5IF 位置 1), 输出引脚为 PA5/PB5/PC5, TIMER1 溢出时, PA5/PB5/PC5 为低电平;
0101	比较模式	选择 CCPR5 匹配时将输出置为低电平 (CCP5IF 位置 1), 输出引脚为 PA5/PB5/PC5, TIMER1 溢出时, PA5/PB5/PC5 为高电平;
0110	比较模式	选择 CCPR5 匹配时将产生软件中断 (CCP5IF 位置 1, 但 PA5/PB5/PC5 引脚不受影响);
0111	比较模式	选择 CCPR5 触发特殊事件 (CCP5IF 位置 1, PA5/PB5/PC5 引脚不受影响); 清零 Timer1, 并启动 ADC 采集 (如果 ADCON=1);
1000	捕捉模式	每个下降沿发生, 捕捉值存入 CCPR5 寄存器, 并产生中断标志 CCP5IF;
1001	捕捉模式	每个上升沿发生, 捕捉值存入 CCPR5 寄存器, 并产生中断标志 CCP5IF;
1010	捕捉模式	每 4 个上升沿发生一次, 捕捉值存入 CCPR5 寄存器, 并产生中断标志

		CCP5IF;
1011	捕捉模式	每 16 个上升沿发生一次，捕捉值存入 CCPR5 寄存器，并产生中断标志 CCP5IF;
1100	捕捉模式	下降沿复位 TIMER1，捕捉值存入 CCPR5 寄存器，并产生中断标志 CCP5IF;
1101	捕捉模式	上升沿复位 TIMER1，捕捉值存入 CCPR5 寄存器，并产生中断标志 CCP5IF;
1110	捕捉模式	下降沿捕捉值存入 CCPR5 寄存器，不产生中断标志 CCP5IF;
1111	捕捉模式	上升沿捕捉值存入 CCPR5 寄存器，不产生中断标志 CCP5IF;
001X	PWM 模式	启动 CCP5PWM 模式;

7.3 PWM 模式

7.3.1. PWM 模式概述

PWM 模式可以在相应引脚上产生脉宽调制信号。占空比、周期和分辨率由以下寄存器决定:

1. PR2;
2. T2CON;
3. CCPR0L, CCPR1L, CCPR2L, CCPR3L, CCPR4L, CCPR5L;、
4. CCP0CON, CCP1CON, CCP2CON, CCP3CON, CCP4CON, CCP5CON;
5. PWMCON0, PWMCON1, PWMCON2, PWMCON3, PWMCON4。

在脉宽调制（PWM）模式下，CCP 模块会在 PWMn0 和 PWMn1 引脚上产生最大 16 位分辨率的 PWM 输出信号。由于 PWMn0 和 PWMn1 引脚与端口数据锁存器复用，该引脚必须设置为输出以使能 PWMn0 和 PWMn1 引脚输出驱动器。

可通过写 PR2 寄存器来指定 PWM 周期。可以用以下公式计算 PWM 周期:

$$PWM周期 = \frac{[PR2[15:0] + 1]}{T2_CLK}$$

当 TMR2 等于 PR2 时，在下一个递增计数周期中将发生以下三件事件:

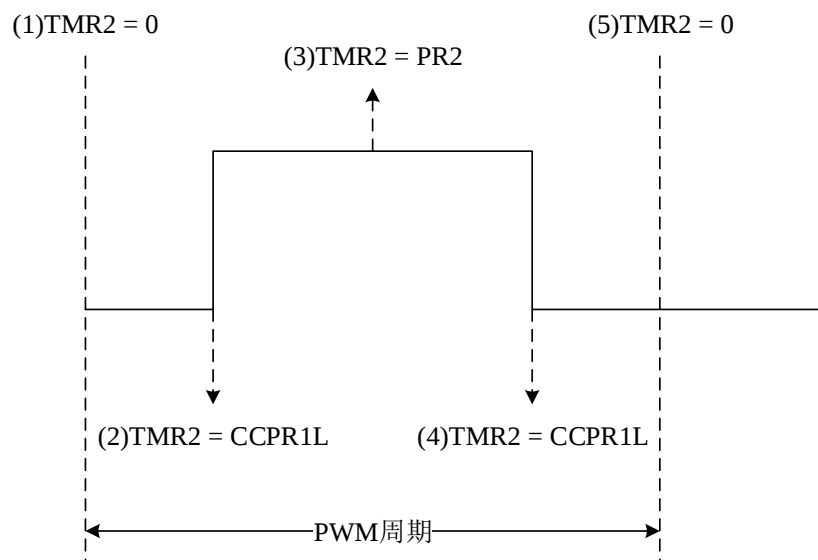
1. TMR2 被清零;
2. PWM20 和 PWM21 引脚被置 1; (例外情况: 如果 PWM 占空比=0%, 引脚将不会被置 1。)
3. PWM 占空比从 CCPR1L 锁存到 CCPR1H。

通过写入 CCPRnL 寄存器来指定 PWM 的占空比。最高分辨率可达 16 位。计算 PWM 占

空比的公式如下：

$$PWM \text{ 占空比} = CCPRnL / PR2$$

7.3.1.1. PWM 中心对齐模式



PWM 中心对齐模式示意图

当 TMR2TRI (T2CON<9>) 开启时，PWM 使能中心对齐模式，此时 PWM 工作过程如下：

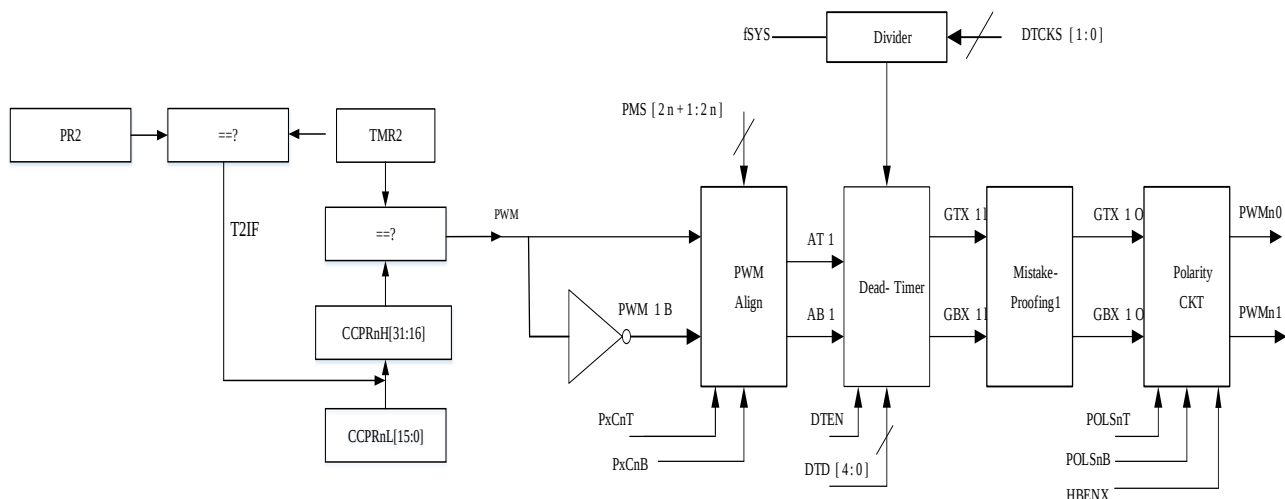
1. PWM 周期开始，TMR2 开始从 0 递增；
2. 当 TMR2 与 CCPR1L 相等时，PWM 开始第一次高低电平变化，TMR2 继续递增；
3. 当 TMR2 与 PR2 相等时，TMR2 开始自减；
4. 当 TMR2 再次与 CCPR1L 相等时，PWM 再次变化高低电平；
5. 当 TMR2 自减为 0 时，此时 PWM 周期结束，开始下一个 PWM 周期。

此时，实际上：

$$PWM \text{ 周期} = 2 \cdot \frac{(PR2 + 1)}{T2_CLK}$$

注意：普通 PWM 模式切换中心对齐模式需要将 PWM 关闭后重新打开。

7.3.1.2. PWM 互补模式输出



PWM 互补输出模式示意图

本章节的内容中，上臂指：PWM00，PWM10，PWM20，PWM30，PWM40，PWM50；

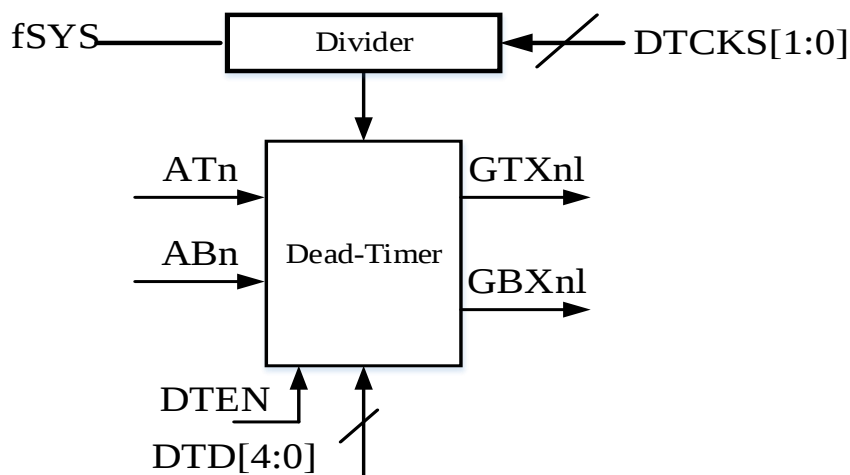
下臂指：PWM01，PWM11，PWM21，PWM31，PWM41，PWM51；

注：PWMn0 与 PWMn1 为一组互补信号 n 的取值 0~5。

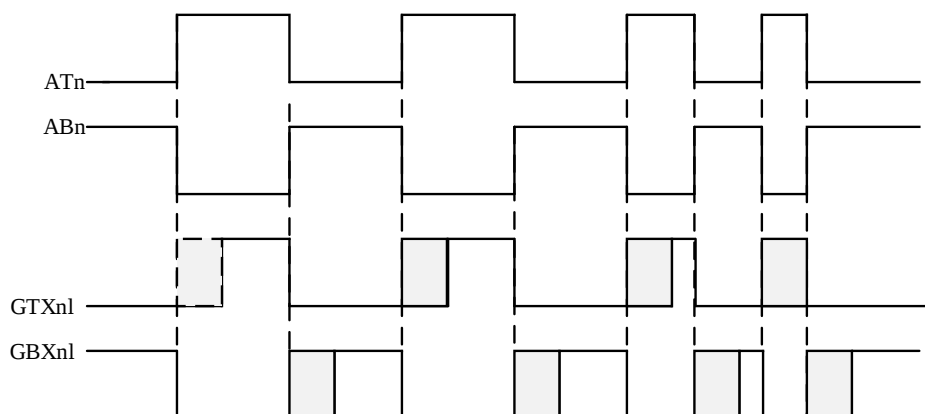
7.3.1.3. PWM 死区时间

死区时间电路设计的目的为，插入死区时间可确保外部驱动电路晶体管对的上下臂在转态时不会瞬间导通（上下臂 MOS 皆开启）而产生短路电流。为了消除这种危险，设计了一段死区时间，确保输出转态的过程中，两个晶体管处于不会同时导通的状态。死区时间插入使能或除能由 DTC 寄存器的 DTEN 位控制。死区时间要控制在 $0.3\mu\text{s}\sim 5\mu\text{s}$ 左右，可通过 DTCKS1~DTCKS0 位选择死区时钟源，并通过 DTD4~DTD0 位对插入的死区时间进行调整。

下图为死区时间方框图和插入死区时间时序图。需注意的是，若开启死区时间功能，只有在上沿时插入死区时间，下降沿不变化。



死区时间框图



插入死区时间时序图（深色为死区时间）

7.3.2. PWM 寄存器

基地址：0x4000 3000

7.3.2.1. PWMCON0 (PWM 控制寄存器 0)

偏移地址：0x030

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27:26	PXC5	PWMS5=11 时，上/下臂输出选择 00：上/下臂输出都关闭； 01：上臂输出关闭/下臂输出导通：输出低； 10：上臂输出导通/下臂输出关闭：输出高； 11：上/下臂输出都关闭（防止上/下臂同时导通）；	R/W	0
25:24	PXC4	PWMS4=11 时，上/下臂输出选择 00：上/下臂输出都关闭； 01：上臂输出关闭/下臂输出导通：输出低； 10：上臂输出导通/下臂输出关闭：输出高； 11：上/下臂输出都关闭（防止上/下臂同时导通）；	R/W	0
23:22	PXC3	PWMS3=11，上/下臂输出选择 00：上/下臂输出都关闭； 01：上臂输出关闭/下臂输出导通：输出低； 10：上臂输出导通/下臂输出关闭：输出高； 11：上/下臂输出都关闭（防止上/下臂同时导通）；	R/W	0
21:20	PXC2	PWMS2=11 时，上/下臂输出选择 00：上/下臂输出都关闭； 01：上臂输出关闭/下臂输出导通：输出低； 10：上臂输出导通/下臂输出关闭：输出高；	R/W	0

		11: 上/下臂输出都关闭（防止上/下臂同时导通）;		
19:18	PXC1	PWMS1=11 时，上/下臂输出选择 00: 上/下臂输出都关闭; 01: 上臂输出关闭/下臂输出导通: 输出低; 10: 上臂输出导通/下臂输出关闭: 输出高; 11: 上/下臂输出都关闭（防止上/下臂同时导通）;	R/W	0
17:16	PXC0	PWMS0=11 时，上/下臂输出选择 00: 上/下臂输出都关闭; 01: 上臂输出关闭/下臂输出导通: 输出低; 10: 上臂输出导通/下臂输出关闭: 输出高; 11: 上/下臂输出都关闭（防止上/下臂同时导通）;	R/W	0
15:12	Reserved		R	0
11:10	PWMS5	PWM5 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM5 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0
9:8	PWMS4	PWM4 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM4 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0
7:6	PWMS3	PWM3 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM3 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0
5:4	PWMS2	PWM2 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM2 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0
3:2	PWMS1	PWM1 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM1 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0
1:0	PWMS0	PWM0 调制模式选择 00: 互补式控制; 01: 非互补式上臂调制; 10: 非互补式下臂调制; 11: PWM0 控制 (PxClT 和 PxClB 位分别控制上/下臂输出)	R/W	0

7.3.2.2. PWMCON1 (PWM 控制寄存器 1)

偏移地址: 0x034

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27:26	POLS5	00: PWM50 的同相输出, PWM51 同相输出; 01: PWM50 的反相输出, PWM51 同相输出; 10: PWM50 的同相输出, PWM51 反相输出; 11: PWM50 的反相输出, PWM51 反相输出;	R/W	0
23:24	POLS4	00: PWM40 的同相输出, PWM41 同相输出; 01: PWM40 的反相输出, PWM41 同相输出; 10: PWM40 的同相输出, PWM41 反相输出; 11: PWM40 的反相输出, PWM41 反相输出;	R/W	0
23:22	POLS3	00: PWM30 的同相输出, PWM31 同相输出; 01: PWM30 的反相输出, PWM31 同相输出; 10: PWM30 的同相输出, PWM31 反相输出; 11: PWM30 的反相输出, PWM31 反相输出;	R/W	0
21:20	POLS2	00: PWM20 的同相输出, PWM21 同相输出; 01: PWM20 的反相输出, PWM21 同相输出; 10: PWM20 的同相输出, PWM21 反相输出; 11: PWM20 的反相输出, PWM21 反相输出;	R/W	0
19:18	POLS1	00: PWM10 的同相输出, PWM11 同相输出; 01: PWM10 的反相输出, PWM11 同相输出; 10: PWM10 的同相输出, PWM11 反相输出; 11: PWM10 的反相输出, PWM11 反相输出;	R/W	0
17:16	POLS0	00: PWM00 的同相输出, PWM01 同相输出; 01: PWM00 的反相输出, PWM01 同相输出; 10: PWM00 的同相输出, PWM01 反相输出; 11: PWM00 的反相输出, PWM01 反相输出;	R/W	0
15:11	Reserved		R	0
10	PWM50POE	PWM50 和 PWM51 输出使能位 1: 使能; 0: 禁止;	R/W	0
9	Reserved		R	0
8	PWM40POE	PWM40 和 PWM41 输出使能位 1: 使能; 0: 禁止;	R/W	0
7	Reserved		R	0
6	PWM30POE	PWM30 和 PWM31 输出使能位 1: 使能; 0: 禁止;	R/W	0

5	Reserved		R	0
4	PWM20POE	PWM20 和 PWM21 输出使能位 1: 使能; 0: 禁止;	R/W	0
3	Reserved		R	0
2	PWM10POE	PWM10 和 PWM11 输出使能位 1: 使能; 0: 禁止;	R/W	0
1	Reserved		R	0
0	PWM00POE	PWM00 和 PWM01 输出使能位 1: 使能; 0: 禁止;	R/W	0

7.3.2.3. PWMCON2 (PWM 控制寄存器 2)

注：死区时钟源 f_{DT} 为 timer2 分频后的时钟。

偏移地址：0x038

Bit	Name	Description	Attribute	Reset
31:30	DTCKS3	选择死区时钟 f_{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;	R/W	0
29	DTEN3	死区时间使能位 0: 禁止; 1: 使能;	R/W	0
28:24	DTD3	死区时间计数器, 死区时间= $(1 + DTD[4:0])/f_{DT}$;	R/W	0
23:22	DTCKS2	选择死区时钟 f_{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;	R/W	0
21	DTEN2	死区时间使能/除能控制位 0: 禁止; 1: 使能;	R/W	0
20:16	DTD2	死区时间计数器, 死区时间= $(1 + DTD[4:0])/f_{DT}$;	R/W	0
15:14	DTCKS1	选择死区时钟 f_{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频;	R/W	0

		11: 8 分频;		
13	DTEN1	死区时间使能位 0: 禁止; 1: 使能;	R/W	0
12:8	DTD1	死区时间计数器, 死区时间=(1 + DTD[4:0])/f _{DT} ;	R/W	0
7:6	DTCKS0	选择死区时钟 f _{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;	R/W	0
5	DTEN0	死区时间使能位 0: 禁止; 1: 使能;	R/W	0
4:0	DTD0	死区时间计数器, 死区时间=(1 + DTD[4:0])/f _{DT} ;	R/W	0

7.3.2.4. PWMCON3 (PWM 控制寄存器 3)

偏移地址: 0x03C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:14	DTCKS5	选择死区时钟 f _{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;	R/W	0
13	DTEN5	死区时间使能位 0: 禁止; 1: 使能;	R/W	0
12:8	DTD5	死区时间计数器, 死区时间=(1 + DTD[4:0])/f _{DT} ;	R	0
7:6	DTCKS4	选择死区时钟 f _{DT} 分频 00: 不分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;	R/W	0
5	DTEN4	死区时间使能位 0: 禁止; 1: 使能;	R/W	0
4:0	DTD4	死区时间计数器, 死区时=(1 + DTD[4:0])/f _{DT} ;	R/W	0

7.3.2.5. PWMCON4 (PWM 控制寄存器 4)

偏移地址: 0x05C

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29	HBEN5	PWM5 模块使能位 1: 使能; 0: 禁止;	R/W	0
28	HBEN4	PWM4 模块使能位 1: 使能; 0: 禁止;	R/W	0
27	HBEN3	PWM3 模块使能位 1: 使能; 0: 禁止;	R/W	0
26	HBEN2	PWM2 模块使能位 1: 使能; 0: 禁止;	R/W	0
25	HBEN1	PWM1 模块使能位 1: 使能; 0: 禁止;	R/W	0
24	HBEN0	PWM0 模块使能位 1: 使能; 0: 禁止;	R/W	0
23:22	Reserved		R	0
21	PTMODE5	保护模式, 强制保护 PWM5 的输出值 (上桥默认输出 0, 这里只是下桥) 1: 使能保护值为 1; 0: 使能保护值为 0;	R/W	0
20	PTMODE4	保护模式, 强制保护 PWM4 的输出值 (上桥默认输出 0, 这里只是下桥) 1: 使能保护值为 1; 0: 使能保护值为 0;	R/W	0
19	PTMODE3	保护模式, 强制保护 PWM3 的输出值 (上桥默认输出 0, 这里只是下桥) 1: 使能保护值为 1; 0: 使能保护值为 0;	R/W	0
18	PTMODE2	保护模式, 强制保护 PWM2 的输出值 (上桥默认输出 0, 这里只是下桥) 1: 使能保护值为 1; 0: 使能保护值为 0;	R/W	0
17	PTMODE1	保护模式, 强制保护 PWM1 的输出值 (上桥默认输出 0, 这里只是下桥) 1: 使能保护值为 1;	R/W	0

		0: 使能保护值为 0;		
16	PTMODE0	保护模式，强制保护 PWM01 的输出值（上桥默认输出 0，这里只是下桥） 1: 使能保护值为 1; 0: 使能保护值为 0;	R/W	0
15:8	Reserved		R	0
7	PWMAF	PWM 发生过自动关闭位 1: 使能; 0: 禁止;	R	0
6	PWMAF	PWM 发生过自动启动位 1: 使能; 0: 禁止;	R	0
5	Reserved		R	0
4	OPOFLTEN	OP0OUT 为故障输入，从 0 变为 1 后，禁止 PWM 输出使能位 1: 使能; 0: 禁止;	R/W	0
3	OP1FLTEN	OP1OUT 为故障输入，从 0 变为 1 后，禁止 PWM 输出使能位 1: 使能; 0: 禁止;	R/W	0
2	IOFLTEN	PC0 映射为故障输入，从 0 变为 1 后，禁止 PWM 输出使能位 1: 使能; 0: 禁止;	R/W	0
1	ASTART	发生故障后 (ACLOSE = 1)，当故障都解除后，自动启动 PWM 输出使能位 1: 使能; 0: 禁止;	R/W	0
0	ACLOSE	发生故障后，自动关闭 PWM 输出使能位 1: 使能; 0: 禁止;	R/W	0

7.3.2.6. PWMADCON0 (PWM 控制 AD 采集寄存器 0)

偏移地址: 0x040

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWMOIE	PWM 的中断屏蔽位;	R/W	0
22:18	PWMOADCH	ADC 采集通道的选择;	R/W	0

17	PWM0ADPOS	1: 使能 PWM00 由 1 变为 0 采集; 0: 使能 PWM00 由 0 变为 1 采集;	R/W	0
16	PWM0ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM0ADRES	PWM0AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.7. PWMADCON1 (PWM 控制 AD 采集寄存器 1)

偏移地址: 0x044

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM1IE	PWM 的中断屏蔽位;	R/W	0
22:18	PWM1ADCH	ADC 采集通道的选择;	R/W	0
17	PWM1ADPOS	1: 使能 PWM10 由 1 变为 0 采集; 0: 使能 PWM10 由 0 变为 1 采集;	R/W	0
16	PWM1ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM1ADRES	PWM1AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.8. PWMADCON2 (PWM 控制 AD 采集寄存器 2)

偏移地址: 0x048

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM2IE	PWM 的中断屏蔽位;	R/W	0
22:18	PWM2ADCH	ADC 采集通道的选择;	R/W	0
17	PWM2ADPOS	1: 使能 PWM20 由 1 变为 0 采集; 0: 使能 PWM20 由 0 变为 1 采集;	R/W	0
16	PWM2ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM2ADRES	PWM2AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.9. PWMADCON3 (PWM 控制 AD 采集寄存器 3)

偏移地址: 0x04C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM3IE	PWM 的中断屏蔽位;	R/W	0
22:18	PWM3ADCH	ADC 采集通道的选择;	R/W	0
17	PWM3ADPOS	1: 使能 PWM30 由 1 变为 0 采集; 0: 使能 PWM30 由 0 变为 1 采集;	R/W	0
16	PWM3ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM3ADRES	PWM3AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.10. PWMADCON4 (PWM 控制 AD 采集寄存器 4)

偏移地址: 0x050

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM4IE	PWM 的中断屏蔽位;	R/W	0
22:18	PWM4ADCH	ADC 采集通道的选择;	R/W	0
17	PWM4ADPOS	1: 使能 PWM40 由 1 变为 0 采集; 0: 使能 PWM40 由 0 变为 1 采集;	R/W	0
16	PWM4ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM4ADRES	PWM4AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.11. PWMADCON5 (PWM 控制 AD 采集寄存器 5)

偏移地址: 0x054

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM5IE	PWM 的中断屏蔽位;	R/W	0
22:18	PWM5ADCH	ADC 采集通道的选择;	R/W	0

17	PWM5ADPOS	1: 使能 PWM50 由 1 变为 0 采集; 0: 使能 PWM50 由 0 变为 1 采集;	R/W	0
16	PWM5ADEN	ADC 采集使能位 1: 使能; 0: 禁止;	R/W	0
15:12	Reserved		R	0
11:0	PWM5ADRES	PWM5AD 采集结果 (是将 ADRES 自动拷贝到此处);	R/W	0

7.3.2.12. PWMADCON6 (PWM 控制 AD 采集寄存器 6)

偏移地址: 0x058

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PWMADDLY	PWM 采集 AD 的延时时间 (时间单位为 APB_PWM_CLK);	R/W	0

7.3.2.13. CCPIF (CCP 中断标志寄存器)

偏移地址: 0x060

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	PWM5IF	PWM5 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
10	PWM4IF	PWM4 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
9	PWM3IF	PWM3 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
8	PWM2IF	PWM2 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
7	PWM1IF	PWM1 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
6	PWM0IF	PWM0 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
5	CCP5IF	CCP5 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0

4	CCP4IF	CCP4 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
3	CCP3IF	CCP3 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
2	CCP2IF	CCP2 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
1	CCP1IF	CCP1 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0
0	CCP0IF	CCP0 中断产生标志位 1: 产生中断; 0: 未产生中断;	R	0

7.3.2.14. CCPIC (CCP 中断清除寄存器)

偏移地址: 0x064

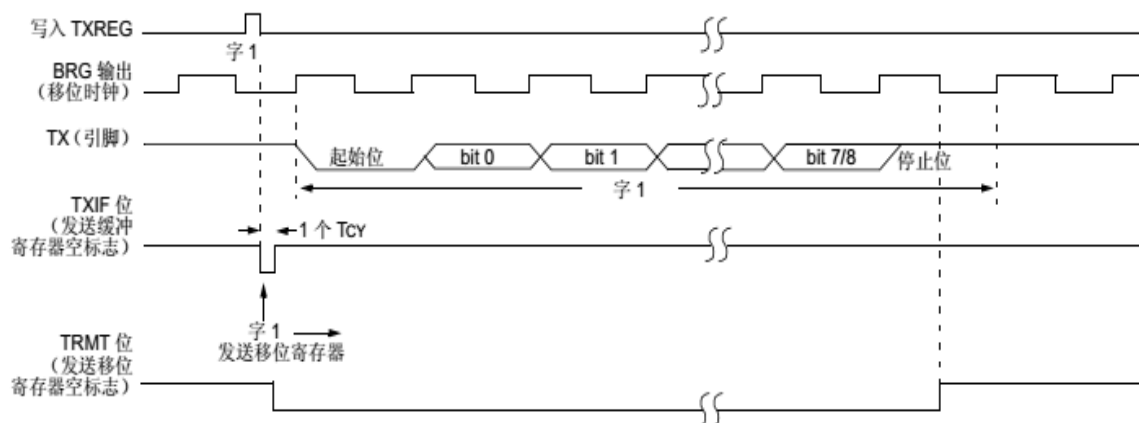
Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	PWM5IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
10	PWM4IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
9	PWM3IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
8	PWM2IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
7	PWM1IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
6	PWM0IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
5	CCP5IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0

4	CCP4IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
3	CCP3IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
2	CCP2IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
1	CCP1IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0
0	CCP0IC	中断标志清除位 1: 清中断; 0: 没有作用;	R/W	0

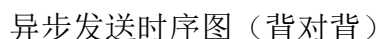
断。不管 TXIE 的状态如何，只要中断发生，TXIF 就会置 1 并且不能用软件清零。TXIF 不会在 TXREG 装入新数据时立即被清零，而是在装入指令后的第二个指令周期复位。因此在 TXREG 装入新数据后立即查询 TXIF，会返回无效结果。标志位 TXIF 指示的是 TXREG 寄存器的状态，而另一个位 TRMT (TXSTA<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位，它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行轮询。

设置异步发送的操作步骤如下：

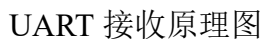
1. 初始化 SPBRG，设置合适的波特率。按需要将 BRGMOD 设定，以获得目标波特率；
2. 如果需要中断，将使能位 TXIE 置位；
3. 若需要发送 9 位数据，将发送位 TX9EN 置 1。发送的第 9 位可以是地址位也可以是数据位；
4. 通过将 TXEN 位置 1 使能发送，此操作同时也会将 TXIF 位置 1；
5. 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位和 TXREG 一起装入寄存器；
6. 将数据装入 TXREG 寄存器（启动发送）；
7. 若想使用中断，请确保将打开 UART 对应中断。



异步发送时序图



8.2.1.2. 异步接收

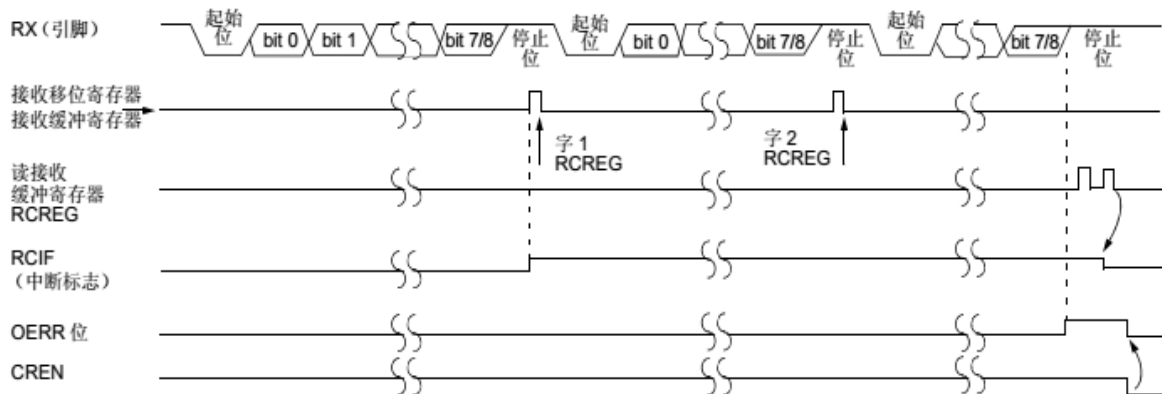


如“UART 接收原理图”所示，在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而主接收串行移位器的工作频率等于比特率或 F_{osc} 。

设置异步发送的操作步骤如下:

1. 初始化 SPBRG, 设置合适的波特率。按需要将 BGRMOD 设定, 以获得目标波特率;
2. 如果需要中断, 将使能位 RCIE 置位;
3. 若需要接收 9 位数据, 将接收位 RX9EN 置 1;

4. 通过将 CREN 位置 1，使能接收；
5. 当接收完成时标志位 RCIF 将被置 1，此时如果使能位 RCIE 已值 1，还将产生一个中断；
6. 通过读 RCREG 寄存器来读取接收到的 8 位数据，读 RCREG 寄存器同时可以获取第 9 位数据（如果已使能），并判断是否在接收过程中是否发生了错误；
7. 如果发生错误，通过将使能位 CREN 清零来清除错误；
8. 若想使用中断，请确保打开对应中断。



UART 接收时序图

8.2.2. 同步模式

将 CSRC 位 (TXSTA<7>) 置 1 可以进入同步主控模式。在此模式中，数据以半双工方式（即发送和接收不同时进行）发送。发送数据时，禁止接收，反之亦然。将 SYNC 位 (TXSTA<4>) 置 1 可进入同步模式。此外，应将使能位 SPEN (RCSTA<7>) 置 1，分别把 TX 和 RX 引脚配置为 CK（时钟）和 DT（数据）线。主控模式意味着处理器在 CK 时钟线上发送主控时钟信号。

8.2.2.1. 同步主动

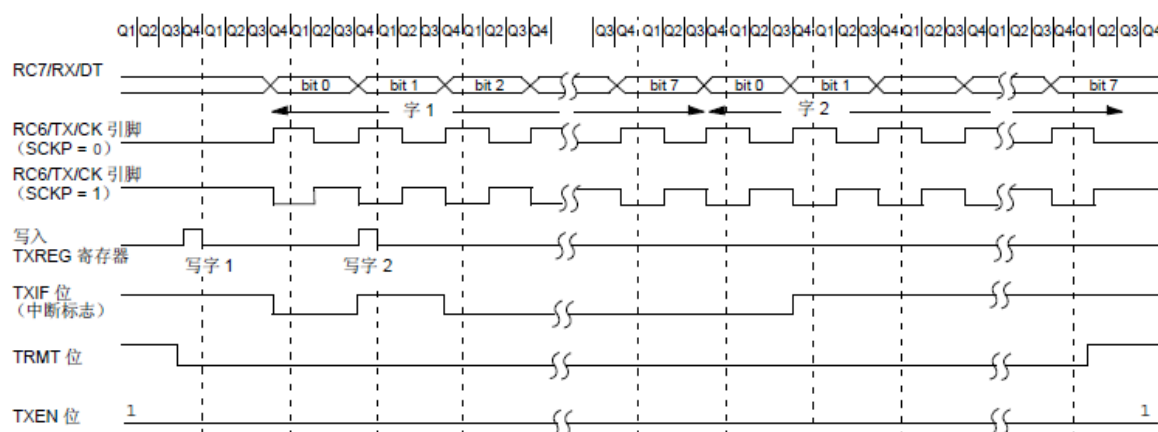
8.2.2.1.1. 同步主控发送

发送器的核心是发送（串行）移位寄存器（TSR）。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据，而 TXREG 寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后，才会向 TSR 寄存器装入新数据。当最后一位发送完成时，就会将 TXREG 寄存器中的新数据（如果有的话）装入 TSR。

一旦 TXREG 寄存器向 TSR 寄存器传输了数据（在 1 个 TCY 内发生），TXREG 寄存器就为空，同时标志位 TXIF 被置 1。可以通过将中断使能位 TXIE 置 1 或清零来使能/禁止该中断。TXIF 的设置与使能位 TXIE 的状态无关，且不能用软件清零。仅在新数据写入 TXREG 寄存器时，TXIF 才会复位。TXIF 表示的是 TXREG 寄存器的状态，而另一个标志位 TRMT（TXSTA<1>）则表示 TSR 寄存器的状态。TRMT 是只读位，它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断均无关联，因此要判断 TSR 寄存器是否为空，用户只能对此位进行轮询。TSR 寄存器并未映射到数据存储器中，因此用户不能直接访问它。

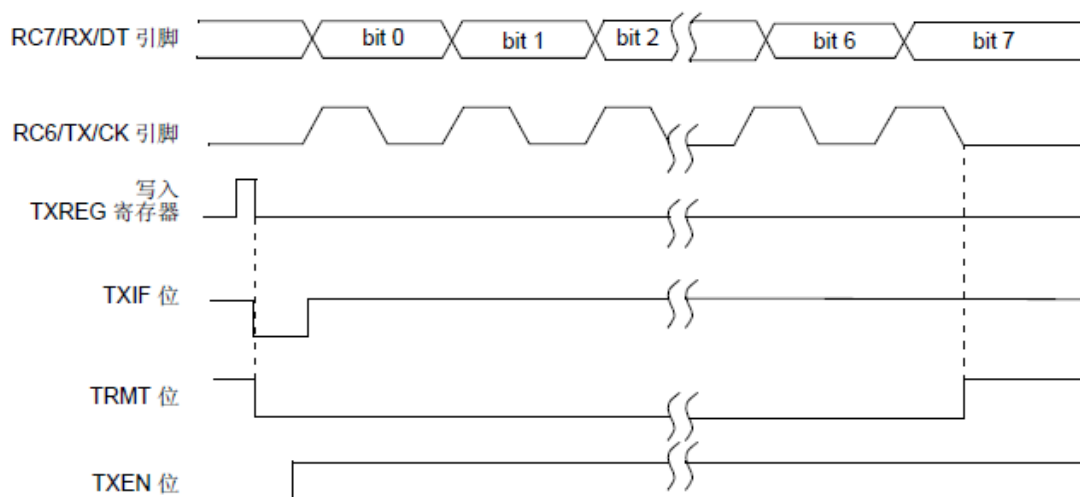
设置同步主控发送操作的步骤如下：

1. 初始化 SPBRG 寄存器，设置合适的波特率。按需要设置 BGRM，以获得目标波特率；
2. 通过将 SYNC，SPEN，CSRC 位置 1，使能同步主控串口；
3. 若需要中断，将中断使能位 TXIE 置 1；
4. 若需要发送 9 位数据，将 TX9EN 位置 1；
5. 将 TXEN 位置 1，使能发送；
6. 如果选择发送 9 位数据，将第 9 位数据装入同时装入 TXREG；
7. 将数据装入 TXREG 寄存器，启动发送；
8. 若想使用中断，请确保打开对应中断。



注： 同步主控模式，SPBRG = 0，连续发送两个 8 位字节。

同步发送时序



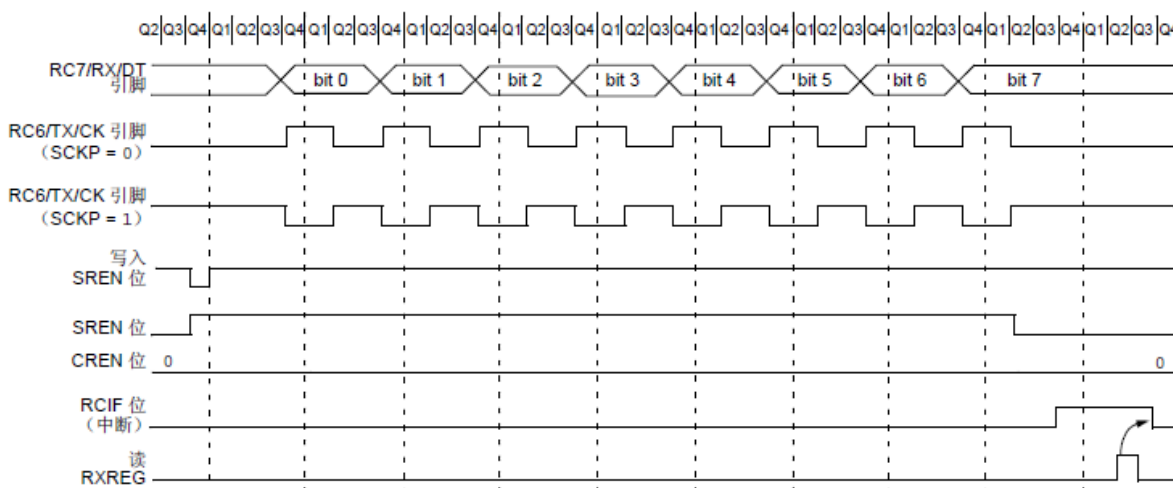
同步发送时序（由 TXEN 位控制）

8.2.2.1.2. 同步主控接收

一旦选择了同步模式，只要将单字节接收使能位 SREN (RCSTA<5>) 或连续接收使能位 CREN (RCSTA<4>) 置 1，即可使能接收。在时钟的下降沿采样 RX 引脚上的数据。如果将使能位 SREN 位置 1，则只接收单个字。如果将使能位 CREN 位置 1，则会连续接收数据，直到将 CREN 位清零。如果两个位均被置 1，则 CREN 具有更高优先级。

设置同步主控接收操作的步骤如下：

1. 初始化 SPBRG 寄存器，设置合适的波特率。按需要设置 BGRM，以获得目标波特率；
2. 通过将 SYNC，SPEN，CSRC 位置 1，使能同步主控串口；
3. 确保将 CREN 和 SREN 位清零；
4. 若需要中断，将中断使能位 RCIE 置 1；
5. 若需要接收 9 位数据，将 RX9EN 位置 1；
6. 若需要单字节接收，将 SREN 位置 1；若需要连续接收，将 CREN 位置 1；
7. 当接收完成时中断标志位 RCIF 将置 1，此时如果中断使能位 RCIE 为 1，将产生一个中断；
8. 通过读 RCREG 寄存器来读取接收到的 8 位数据，读 RCREG 寄存器获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误；
9. 如果发生错误，将 CREN 位清零以清除错误；
10. 若想使用中断，请确保打开对应中断。



同步接受时序（由 SREN 位控制）

8.2.2.2. 同步从动

将 CSRC (TXSTA<7>) 位清零可进入同步从动模式。此模式与同步主控模式的区别在于移位时钟由 CK 引脚上的外部时钟提供（主控模式中由内部时钟提供）。这使得器件能在任何低功耗模式下发送或接收数据。

8.2.2.2.1. 同步从动发送

除了休眠模式以外，同步主控、从动模式的工作方式是完全相同的。如果向 TXREG 写入两个字，然后执行睡眠指令，则将发生以下事件：

1. 第一个字立即传送到 TSR 寄存器进行发送；
2. 第二个字仍保留在 TXREG 寄存器中；
3. 不会将标志位 TXIF 置 1；
4. 当第一个字移出 TSR 后，TXREG 寄存器将把第二个字送入 TSR，同时将标志位 TXIF 置 1；
5. 如果中断使能位 TXIE 已置 1，中断将把器件从休眠状态唤醒。如果使能了全局中断，程序则会跳转到中断向量处执行。

设置同步从动发送操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步从动串口；
2. 将 CREN 和 SREN 位清零；
3. 若需要中断，将中断使能位 TXIE 置 1；

4. 若需要发送 9 位数据，将 TX9EN 位置 1；
5. 将使能位 TXEN 置 1 使能发送；
6. 将数据装入 TXREG 寄存器，如果选择发送 9 位数据，将第 9 位数据装入 TX9D 位，启动发送；
7. 若想使用中断，请确保打开对应中断。

8.2.2.2.2. 同步从动接收

除了休眠模式、空闲模式以及在从动模式下忽略 SREN 位以外，同步主控和同步从动模式的工作方式完全相同。如果在进入休眠或空闲模式前将 CREN 位置 1，使能接收，那么在该低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到 RCREG 寄存器，如果中断使能位 RCIE 已置 1，产生的中断将把器件从低功耗模式唤醒。如果使能了全局中断，程序则会跳转到中断向量处执行。

设置同步从动接收操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步从动串口；
2. 若需要中断，将中断使能位 RCIE 置 1；
3. 若需要接收 9 位数据，将 RX9 位置 1；
4. 将使能位 CREN 置 1，使能接收；
5. 当接收完成时，RCIF 位将被置 1。如果使能位 RCIE 已置 1，还将产生一个中断；
6. 通过读 RCREG 寄存器来读取接收到的 8 位数据，并可以获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误；
7. 如果发生错误，将 CREN 位清零以清除错误；
8. 若想使用中断，请确保打开对应中断。

8.3 USART 寄存器定义

基地址：0x4000 5000

8.3.1. RCSTA 接收控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0

8	RCIE	UART 接收中断使能位 1: 使能; 0: 禁止;	R/W	0
7	SPEN	UART 使能位 1: 使能 (将 RX/DT 和 TX/CK 引脚配置为串口引脚); 0: 禁止 (保持在复位状态); 注: 串口使能后, 再配置管脚复用。	R/W	0
6	RX9EN	UART9 位接收使能位 1: 使能; 0: 禁止;	R/W	0
5	SREN	UART 单字节接收使能位 异步模式: 忽略; 同步主控模式: 1: 使能; 0: 禁止; 此位在接收完成后清零; 同步从动模式: 忽略;	R/W	0
4	CREN	UART 连续接收使能位 1: 使能; 0: 禁止;	R/W	0
3	RXPARTY	UART 接收校验检测使能位 1: 自动校验第 9 位校验位, 校验位填充到 UARTCON 中的 parity 位; 0: 禁止;	R/W	0
2	FERR	UART 帧错误检测标志位 1: 帧错误 (可以通过读 RCREG 寄存器刷新并接收下一个有效字节); 0: 无帧错误;	R	0
1	OERR	UART 溢出错误标志位 1: 溢出错误 (可以通过清零 CREN 位清除); 0: 无溢出错误;	R	0
0	Reserved		R	0

8.3.2. TXSTA 发送控制寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	TXIE	UART 发送中断使能位 1: 使能; 0: 禁止;	R/W	0

7	CSRC	UART 时钟源选择位 异步模式：忽略； 同步模式： 1：主控模式（时钟来自内部 BRG）； 0：从动模式（时钟来自 BCLK 管脚）；	R/W	0
6	TX9EN	UART9 位发送使能位 1：使能； 0：禁止；	R/W	0
5	TXEN	UART 发送使能位 1：使能； 0：禁止；	R/W	0
4	SYNC	USART 模式选择位 1：同步模式； 0：异步模式；	R/W	0
3	TXPARITY	使能 UART 硬件自动奇偶校验位 1：自动填充发送第 9 位校验位； 0：第 9 位由 TXD9 填充；	R/W	0
2	Reserved		R	0
1	TRMT	UART 发送移位寄存器状态位 1：TSR 空； 0：TSR 满；	R	1
0	Reserved		R	0

8.3.3. UARTCON 控制寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	UARTRCDMAEN	UARTRC 的 DMA 使能位 1：DMA 使能； 0：DMA 禁止；	R/W	0
10	UARTTXDMAEN	UARTTX 的 DMA 使能位 1：DMA 使能； 0：DMA 禁止；	R/W	0
9	CARDEN	UART 7816 协议使能位 1：使能； 0：禁止；	R/W	0
8	TCTRL	UART 7816 协议控制位 1：使能； 0：禁止；	R/W	0
7	INVMODE	UART 数据电平反向接受发送使能位（波形反相）	R/W	0

		异步：1：使能； 0：禁止； 同步：忽略；		
6	PARITY	UART 接收奇偶校验位 1：表示有奇数个 1（奇校验）； 0：表示有偶数个 1（偶校验）；	R/W	0
5	ODDEVEN	UART 设定发送、接收奇偶校验位功能位 1：奇校验； 0：偶校验；	R/W	0
4	BOUDCLK	UART 波特率发生器时钟源选择位 1：外部时钟； 0：内部系统时钟；	R/W	0
3	HALFDUPLEX	UART 单线半双工使能位 1：使能； 0：禁止；	R/W	0
2	DLSB	UART 数据位高低位选择位 1：高位在前（MSB）； 0：低位在前（LSB）；	R/W	0
1:0	STOPBIT[1:0]	UART 停止位个数设置位 00：1 位停止位； 01：2 位停止位； 10：3 位停止位；	R/W	0

8.3.4. SPBRG 波特控制寄存器

偏移地址：0x014

Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
14:13	BRGM	UART 波特率时钟分频控制 00：波特率时钟 64 分频， $UART\ APB\ CLK / (64[SPBRG])$ ； 01：波特率时钟 16 分频， $UART\ APB\ CLK / (16[SPBRG])$ ； 10：波特率时钟 4 分频， $UART\ APB\ CLK / (4[SPBRG])$ ； 11：波特率时钟 2 分频， $UART\ APB\ CLK / (2[SPBRG])$ ；	R/W	0
12:0	BRG[12:0]	UART 波特率发生器；	R/W	0

注：BGR[12:0]，UART 波特率发生器数值计算详见<7.3.8 BGR 寄存器>章节。

8.3.5. TXREG 发送寄存器

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	TX9D	发送第九位；	W	0
7:0	TXREG[8:0]	发送缓冲器，写此位硬件自动清 TXIF 标志；	W	0

8.3.6. RCREG 接收寄存器

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	RC9D	接收第九位；	R	0
7:0	RCREG[7:0]	接收缓冲器，读此位硬件自动清 RCIF 标志；	R	0

8.3.7. UARTI 中断标志寄存器

偏移地址：0x018

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	RCIF	RXREG 不为空中断 0：未产生中断； 1：产生中断；	R	0
0	TXIF	TXREG 为空中断，写 TXREG 寄存器，在第二个指令周期硬件自动清该标志位，此位写无效。 0：未产生中断； 1：产生中断；	R	0

8.3.8. BRG 寄存器（波特率发生器）

BRG 是一个 13 位的波特率发生器，波特率计算公式如下：

BRGM[1:0]		BOUD_CLK	波特率公式
0	0	0	$APB\ CLK / [64 * (SPBRG)]$
0	1	0	$APB\ CLK / [16 * (SPBRG)]$
1	0	0	$APB\ CLK / [4 * (SPBRG)]$

1	1	0	$APB\ CLK / [2 * (SPBRG)]$
0	0	1	$BCLK / [64 * (SPBGR)]$
0	1	1	$BCLK / [16 * (SPBGR)]$
1	0	1	$BCLK / [4 * (SPBGR)]$
1	1	1	$BCLK / [2 * (SPBGR)]$

针对工作在异步模式下,工作频率 APB CLK 为 16MHz,采用 13 位 BRG,目标波特率为 9600bps 的器件:

低速模式:

$$\text{目标波特率} = APB\ CLK / (64([SPBRG]))$$

求解 SPBRG:

$$\begin{aligned} X &= ((APB\ CLK / \text{目标波特率}) / 64) \\ &= ((16000000 / 9600) / 64) \\ &= 26.04 \\ &\approx 26 \end{aligned}$$

$$\text{计算得到的波特率} = \frac{16000000}{64 \times 26} = 9615$$

$$\begin{aligned} \text{误差} &= \frac{\text{波特率计算结果} - \text{目标波特率}}{\text{目标波特率}} \\ &= (9615 - 9600) \\ &= 0.16 \text{ 误差} \end{aligned}$$

高速模式:

$$\text{目标波特率} = APB\ CLK / (16([SPBRG]))$$

求解 SPBRG:

$$\begin{aligned} X &= ((APB\ CLK / \text{目标波特率}) / 16) \\ &= ((16000000) / 9600 / 16) + 1 \\ &= 104.17 \\ &\approx 104 \end{aligned}$$

$$\text{计算得到的波特率} = \frac{16000000}{16 \times 104} = 9615$$

$$\text{误差} = \frac{\text{波特率计算结果} - \text{目标波特率}}{\text{目标波特率}}$$

$$= (9615 - 9600)$$

$$= 0.16 \text{ 误差}$$

9 I2C 接口

9.1 I2C 概述

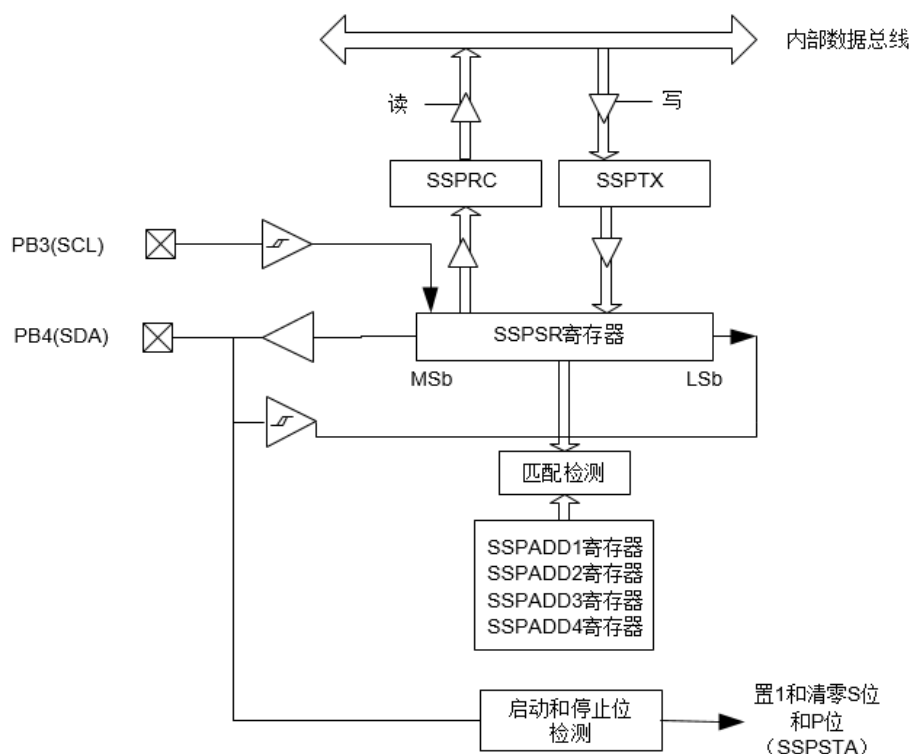
9.1.1. I2C 简介

I2C 接口仅用于实现 I2C 硬件从机功能，通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。支持广播地址呼叫和不同地址长度，支持从机的硬件发送和接收，允许连接到标准（100kHz）或高速（400kHz）的 I2C 总线。此外可以根据特定设备的需要，选择使用 DMA 以减轻 CPU 的负担。

数据引脚（SDA）和时钟引脚（SCL）可以任意搭配，但必须在 GPIO 的特殊功能寄存器中使能 I2C 接口：

- 串行时钟（SCL）：PA3、PA4、PB3、PC1、PD0；
- 串行数据（SDA）：PA2、PB4、PC0、PC6；

注：引脚复用为 I2C 接口时，必须打开开漏功能。



I2C 功能框图

9.1.2. I2C 特性

- 并行总线/I2C 总线协议转换器；
- 双级 Buffer 缓冲的发送/接收功能；
- I2C 从设备功能；
 - 7/10 位地址长度模式
 - 可编程的 I2C 地址检测
 - 可响应 4 个从地址
 - 可屏蔽 1 个从地址
 - 启动位检测
 - 停止位检测
- 产生和检测 7 位/10 位地址和广播呼叫；
- 支持不同的通讯速度；
 - 标准速度（100kHz）
 - 高速速度（400kHz）
- 状态标志；
 - 接收/发送数据类型为数据/地址标志
 - 主机读/写信息标志
 - 接收/发送缓冲器状态标志
- 5 种中断触发方式；
 - 总线悬空触发中断
 - 停止中断
 - 启动中断
 - 发送中断
 - 接收中断
- 发送/接收中断标志硬件自动清除；
- 可选的拉长 ACK/NACK 应答功能；

9.2 I2C 从机模式

在从机模式下，一旦通过配置 SSPEN 和 SSPM 使能主同步串行端口（MSSP）模块，I2C

总线接口就会持续等待启动条件，当启动条件出现，移位寄存器 SSPSR 会在时钟信号（SCL）的高电平采样所有的输入位，在第 8 个时钟信号（SCL）的下降沿，移位寄存器 SSPSR<7:1> 的值会和 SSPADD、SSPADD1、SSPADD2、SSPADD3 地址寄存器的值进行比较，如果地址匹配，会发生事件 9.3.1（主机写时）或 9.3.2（主机读时）。

注：当从机地址为 10 位时在第 11 个时钟信号（SCL）的下降沿，移位寄存器 SSPSR<10:1> 的值会和 SSPADD、SSPADD1、SSPADD2、SSPADD3 地址寄存器的值进行比较。

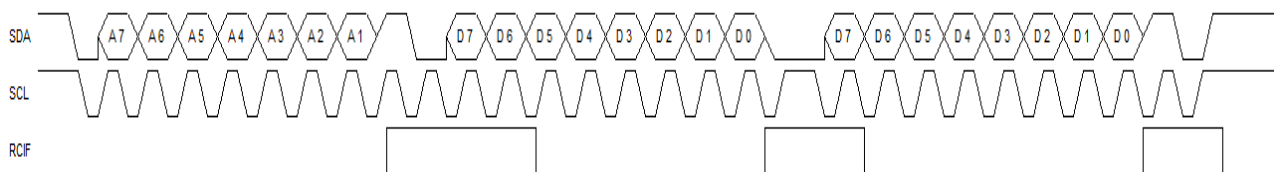
9.2.1. I2C 主机写时的从机模式

主机写时的从机执行步骤：

1. 移位寄存器 SSPSR 的值被载入 SSPRC 寄存器；
2. 接收缓冲器状态位 BFRC 置 1（满）；
3. 产生 ACK 应答信号；
4. 在第 8 个时钟信号 SCL 下降沿，接收中断标志位 RCIF 被置 1。

当主机写时，接收的地址字节的最后一位（R/W 位）为 0，若地址匹配，SSPSTAT 寄存器的 RW 位置 0。移位寄存器 SSPSR 接收到的地址被装入 SSPRC 寄存器，且 SDA 信号保持低电平（ACK）。

同时，在主机写模式下，从机每接收一个数据都会产生一个 RCIF 中断标志位。此时，RCIF 在时钟信号 SCL 的第 8 个脉冲的上升沿置位。当读出 SSPRC 的数据时，RCIF 中断标志位由硬件自动清零。



I2C 从动模式接收时序图

9.2.2. I2C 主机读时的从机模式

主机读时的从机执行步骤：

1. 移位寄存器 SSPSR 的值被载入 SSPRC 寄存器；
2. 接收缓冲器状态位 BFRC 置 1（满）；
3. 产生 ACK 应答信号；

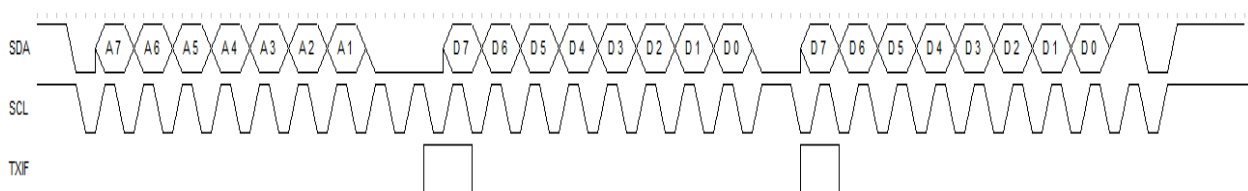
4. 在第 9 个时钟信号 SCL 下降沿（ACK 信号发送完成后），发送中断标志位 TXIF 被置 1。

当主机读时，接收的地址字节的最后一位（R/W 位）为 1，若地址匹配，SSPSTAT 寄存器的 RW 位置 1。SSPTX 寄存器中的数据被装入 SSPSR 移位寄存器。ACK 应答信号在时钟信号 SCL 的第 9 位发送。

在主机读模式下，来自主机的 ACK/NACK 将在时钟信号 SCL 的第 9 个高电平被锁存。如果数据信号 SDA 为高电平（NACK），表示数据传输完成，从机锁存 NACK 信号，复位从动逻辑（复位 SSPSTAT 寄存器），同时从机重新等待下一个启动位的出现；如果 SDA 线为低电平（ACK），从机锁存 ACK 信号，此时从机必须已经在 SSPTX 中写入了待发送的数据。

同时，主机读模式下每发送一个数据都会产生一个 TXIF 中断标志位。此时，TXIF 在时钟信号 SCL 的第 9 个脉冲（ACK/NACK）的下降沿置位；当向 SSPTX 写入数据时，TXIF 中断标志位由硬件自动清零。

注：从机发送的数据必须提前被装入 SSPTX 寄存器（可以在 I2C 配置时，写一个数据到 SSPTX 寄存器）。SSPSR 移位寄存器将收到的 8 个数据依次在时钟信号 SCL 的下降沿移出到 I2C 内部总线，确保在时钟信号 SCL 为高电平期间数据信号 SDA 是有效的。



I2C 从动模式发送时序图

9.2.3. 支持广播呼叫地址

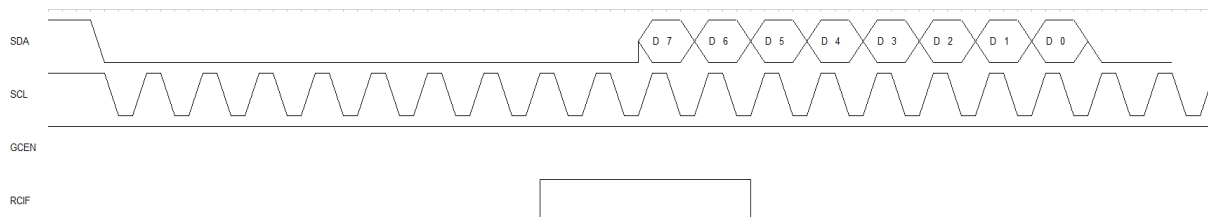
在 I2C 总线的寻址过程中，通常由启动条件后的第一个字节决定主机将寻址哪一个从机。但广播呼叫地址例外，它能寻址所有从机。广播呼叫地址是由 I2C 协议为特定目的保留的 8 个地址之一，它由全 0 组成，且 R/W=0。当使用广播呼叫地址时，理论上所有的从机都应该发送一个应答信号进行响应。

当使能广播呼叫位（GCEN）（SSPCON1<7>置 1）和主同步串行端口（MSSP）模块时，I2C 总线接口即可识别广播呼叫地址。此时 I2C 总线接口持续等待启动条件，当启动条件出现，移位寄存器 SSPSR 会在时钟信号（SCL）的高电平采样所有的输入位，在第 8 个时钟信号（SCL）的下降沿，移位寄存器 SSPSR<7:1>的值会和 SSPADD、SSPADD1、SSPADD2、

SSPADDR3 地址寄存器的值进行比较，如果没有发生地址匹配，最后会与广播呼叫地址进行比较。

如果与广播呼叫地址匹配，SSPSR 的值将被传输到 SSPRC 寄存器，BFRC 标志位置 1，并且在第 8 位的上升沿 RCIF 中断标志位置 1。

当响应中断时，可以通过读取 SSPRC 的内容来检查中断源，该值可以用于判断是特定器件的地址还是一个广播地址。



从动模式广播呼叫地址时序图

9.3 I2C 寄存器

9.3.1. I2C 寄存器概述

通过对控制寄存器 SSPCON0、SSPCON1 和状态寄存器 SSPSTAT 的读写可以分别设置 I2C 接口的工作模式，查看 I2C 的工作状态。SSPCON0 和 SSPCON1 寄存器是可读写的，SSPSTAT 的低六位是只读的，最高位是可读写的。

SSPSR 是一个移位寄存器，负责在时钟信号（SCL）的高电平将总线上的位数据进行读入或在时钟信号（SCL）的下降沿将 SSPTX 的数据按位发送到总线上，它与 SSPRC 和 SSPTX 构成 I2C 接口发送和接收的双级缓冲 Buffer，共同实现 I2C 寄存器数据的移入和移出。

SSPSR 与 SSPRC 构成双级接收缓冲 Buffer，用于对内部总线上 I2C 数据的读入，当移位寄存器 SSPSR 接收到一个完整的字节后，该字节被送入 SSPRC 寄存器，同时将中断标志位 RCIF 置 1；

SSPSR 与 SSPTX 构成双级发送缓冲 Buffer，用于将 SSPTX 寄存器数据写出到 I2C 内部总线上。当向 SSPTX 寄存器写入一个完整的字节后，该字节被送入移位寄存器 SSPSR，同时在发送完数据后的下一个时钟下降沿将中断标志位 TXIF 置 1。

SSPADDR、SSPADDR1、SSPADDR2、SSPADDR3 寄存器用来保存从机的地址，主机通过呼叫该地址实现与从机的通信。SSPADDMASK，用来保存需要屏蔽的从机地址。

9.3.2. I2C 寄存器定义

基地址：0x4000 D000

9.3.2.1. SSPCON0 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	WCOL	写冲突检测位 1: 正在发送前一个数据时，又有数据写入 SSPTX 寄存器，而 SSPTX 已经保存了一个数据，并且 SSPSR 寄存器也存储了一个数据（必须用软件清零）； 0: 未发生冲突；	R/W	0
6	SSPOV	接收溢出指示位 在接收模式下： 1: SSPRC 寄存器仍在保存前一个数据，同时 SSPSR 寄存器又接收到一个新的数据（必须用软件清零） 0: 无溢出；	R/W	0
5	SSPEN	I2C 接口使能位 1: 使能 I2C 接口并将 SDA 和 SCL 引脚配置为串口引脚； 0: 禁止 I2C 接口并将上述引脚配置为 I/O 端口引脚； 注：当使能该位时，必须将 SDA 和 SCL 引脚正确地配置为输入引脚或输出引脚；	R/W	0
4	Reserved	保留	R	0
3:0	SSPM [3:0]	I2C 模式选择 1111: I2C 从动模式，10 位地址； 1110: I2C 从动模式，7 位地址； 0111: I2C 从动模式，10 位地址； 0110: I2C 从动模式，7 位地址； 注：1111 与 0111，1110 与 0110 功能完全相同。	R/W	0

9.3.2.2. SSPCON1 控制寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0

9	I2CRCDMAEN	I2CRC 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
8	I2CTXDMAEN	I2CTX 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
7	GCEN	I2C 广播呼叫使能位 1: 当 SSPSR 接收到广播呼叫地址 (0000h) 时允许中断; 0: 禁止;	R/W	0
6	ACKSTAT	I2C 应答状态位 1: 未收到来自主器件的应答; 0: 收到来自主器件的应答;	R/W	0
5	Reserved		R	0
4	ACKEN	I2C 应答位 1: 无应答 (为 1 时, 本次 SSPSR 接收完, 发送 NACK 给主机); 0: 应答 (为 0 时, 本次 SSPSR 接收完, 发送 ACK 给主机); 注: I2C 应答位只控制数据应答, 收到地址后硬件自动应答。	R/W	0
3: 0	Reserved		R	0

9.3.2.3. SSPSTAT 状态寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	CKE	I2C 广播广播特定输入使能位 1: 使能; 0: 禁止;	R/W	0
5	DA	I2C 数据/地址标志位 1: 上一个接收的字节是数据; 0: 上一个接收的字节是地址;	R	0
4	STOP	I2C 停止标志位 1: 检测到停止位; 0: 未检测到停止位; 注: 当复位或 SSPEN 被清零时, 该位被清零;	R	0
3	START	I2C 启动标志位 1: 检测到启动位; 0: 未检测到启动位;	R	0

2	RW	I2C 读/写信息标志位 1: 读 (表示主控读); 0: 写 (表示主控写); 注: 该位用来保存在最近一次地址匹配后的 R/W 位信息, 仅在从地址匹配开始到下一个启动位、停止位或非 ACK 位之间有效;	R	0
1	BFRC	I2C 接收缓冲器状态位 1: SSPSR 已满 (不包括 ACK 位和停止位); 0: SSPSR 为空 (不包括 ACK 位和停止位);	R	0
0	BFTX	I2C 发送缓冲器状态位 1: SSPSR 已满; 0: SSPSR 为空;	R	0

9.3.2.4. SSPADDR 地址寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	SSPADD	I2C 器件地址 0 (7 位或者 10 位, 根据 SSPM 定义);	R/W	0

9.3.2.5. SSPADDR1 控制寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	SSPADD1	I2C 器件地址 1 (7 位或者 10 位, 根据 SSPM 定义);	R/W	0

9.3.2.6. SSPADDR2 地址寄存器

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	SSPADD2	I2C 器件地址 2 (7 位或者 10 位, 根据 SSPM 定义);	R/W	0

9.3.2.7. SSPADDR3 地址寄存器

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	SSPADD3	I2C 器件地址 3 (7 位或者 10 位, 根据 SSPM 定义);	R/W	0

9.3.2.8. SSPADDRMASK 地址屏蔽寄存器

偏移地址：0x014

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	SSPADD	I2C 器件地址屏蔽位（7 位或者 10 位，根据 SSPM 定义）；	R/W	0

9.3.2.9. SSPRC 接收寄存器

偏移地址：0x018

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SSPRC	与 SSPSR 构成接收双接收 Buffer；	R	0

9.3.2.10. SSPTX 发送寄存器

偏移地址：0x01C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SSPTX	与 SSPSR 构成接收双发送 Buffer；	R	0

9.3.2.11. SSPI 中断寄存器

偏移地址：0x020

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20	FLOATIE	总线悬空使能位 1：使能； 0：禁止； 注：SDA 或 SCL 持续低电平超过 65536 个时钟分频时钟发生中断	R/W	0
19	STOPIE	I2C 停止中断使能位 1：使能； 0：禁止；	R/W	0
18	STARTIE	I2C 启动中断使能位 1：使能； 0：禁止；	R/W	0
17	TXIE	I2C 发送中断使能位 1：使能； 0：禁止；	R/W	0

16	RCIE	I2C 接收中断使能位 1: 使能; 0: 禁止;	R/W	0
15:5	Reserved		R	0
4	FLOATIF	总线悬空标志位 1: 产生中断; 0: 未产生中断;	R	0
3	STOPIF	I2C 停止标志位 1: 产生中断; 0: 未产生中断;	R	0
2	STARTIF	I2C 启动中断标志位 1: 产生中断; 0: 未产生中断;	R	0
1	TXIF	I2C 发送中断标志位 1: 产生中断; 0: 未产生中断;	R	0
0	RCIF	I2C 接收中断标志位 1: 产生中断; 0: 未产生中断;	R	0

9.3.2.12. SSPIC 中断清除寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	FLOATIC	I2C 总线悬空标志清除位 1: 清除中断; 0: 没有作用;	R/W	0
3	STOPIC	I2C 停止中断标志清除位 1: 清除中断; 0: 没有作用;	R/W	0
2	STARTIC	I2C 启动中断标志清除位 1: 清除中断; 0: 没有作用;	R/W	0
1	TXIF	此位通过写 SSPTX 寄存器, 硬件自动清除该标志位; (详见从机模式章节)	R	0
0	RCIF	此位通过读 SSPRC 寄存器, 硬件自动清除该标志位; (详见从机模式章节)	R	0

9.3.2.13. SSPACKDLY 延时寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:13	Reserved		R	0
12	ACKDLYON	I2CACK 时钟延时使能位 从机发送完 8 位数据后 SCL 被保持为低电平，持续 T_{ACKDLY} 时间后释放 SCL； 1：使能时钟延时； 0：禁止时钟延时，需软件将 ACKDLY 位清零；	R/W	0
11:0	ACKDLY	I2CACK 延时时间 $T_{ACKDLY} = ACKDLY * T_{I2C}$ (T_{I2C} : I2C 周期)；	R/W	0

10 SPI 接口

10.1 SPI 接口概述

SPI 串行接口可以与其他外围设备或微控制器设备进行通信，AD3005 SPI 接口支持主从模式，4~16 位数据的收发，接收和发送数据 FIFO 深度为 4。

10.2 SPI 接口寄存器定义

基地址：0x4000 E000

10.2.1. CTRL（SPI 使能相关寄存器）

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
8	SPIDMAEN	SPI DMA 的使能位 0: DMA 禁止; 1: DMA 使能;	R/W	0
7:6	Reserved		R	0
5	DMARXFULL	SPI DMA 接收缓冲器状态位 1: 接收缓冲已满; 0: 接收缓冲未滿;	R/W	0
4	DMATXEMPTY	SPI DMA 发送缓冲器状态位 1: 发送缓冲为空; 0: 发送缓冲未空;	R/W	0
3	RXFIFOEN	SPI RX FIFO 的使能位 0: 禁止; 1: 使能;	R/W	0
2	TXFIFOEN	TX FIFO 的使能位 0: 禁止; 1: 使能;	R/W	0
1	FMEN	SPI 收发功能使能位 0: 禁止; 1: 使能;	R/W	0
0	SPIEN	SPI 使能位 0: 禁止; 1: 使能;	R/W	0

10.2.2. TX_FIFO (SPI 发送数据寄存器)

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TXDATA	发送数据;	W	0

10.2.3. RX_FIFO (SPI 接收数据寄存器)

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	RXDATA	接收数据;	R	0

10.2.4. FIFO_STA (SPI FIFO 状态寄存器)

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:19	Reserved		R	0
19:16	RXNUM	计数器指示在 RX FIFO 中存储了多少个数据字;	R	0
15:11	Reserved		R	0
10:8	TXNUM	计数器指示在 TX FIFO 中存储了多少个数据字;	R	0
7:4	Reserved		R	0
3	RXFULL	SPI RX FIFO 接收满标志位 0: RX FIFO 未滿; 1: RX FIFO 已滿;	R	0
2	RXEMPTY	SPI RX FIFO 接收空标志位 0: RX FIFO 未空; 1: RX FIFO 为空;	R	1
1	TXFULL	SPI TX FIFO 发送满标志位 0: TX FIFO 未滿; 1: TX FIFO 已滿;	R	0
0	TXEMPTY	SPI TX FIFO 发送空标志位 0: TX FIFO 未空; 1: TX FIFO 为空;	R	1

10.2.5. CLK_DIV0 (SPI 时钟分频寄存器)

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	CLKDIV	系统时钟分频寄存器, SPI: SCK 时钟分频器; FSCK = FPCLK / CLK_DIV, CLK_DIV 为偶数值;	R/W	0x20

10.2.6. SPI_MODE (SPI 模式选择寄存器)

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SPI MODE	SPI 模式选择位 0: SPI 从动模式; 1: SPI 主动模式;	R/W	0

10.2.7. SPI_CTRL (SPI 控制寄存器)

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	NSSTOGGLE	SPI 片选控制位 0: 每个数据项末尾的 NSS 切换; 1: 直到 SPI 主设备停止, NSS 才会更改;	R/W	0
7	CPOL	SPI 时钟极性控制位 0: 串行时钟的无效状态为低电平; 1: 串行时钟的无效状态为高;	R/W	0
6	CPHA	SPI 时钟相位控制位 0: 串行时钟在第一个数据位的中间切换; 1: 串行时钟在第一个数据位开始时切换;	R/W	0
5:4	TMOD	SPI 模式控制 00: 发送和接收 01: 仅发送; 10: 仅接收; 11: TMOD 保持最后设置的值不变;	R/W	0
3:0	DTATSIZE	SPI 通讯数据的帧大小控制 注: 0、1、2 帧大小为 8 位, 可实现 4-16 位大小;	R/W	7

10.2.8. SPI_STA（SPI 工作状态寄存器）

偏移地址：0x01C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SPI_WORKING	SPI 的工作状态 1: 工作中; 0: 空闲;	R	0

10.2.9. INTR_EN（SPI FIFO 中断使能寄存器）

偏移地址：0x024

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	STOPIE	SPI 停止中断使能位 0: 禁止; 1: 使能;	R/W	0
9	RXWERRIE	RX FIFO 写错误中断使能位 0: 禁止; 1: 使能;	R/W	0
8	RXRERRIE	RX FIFO 读错误中断使能位 0: 禁止; 1: 使能;	R/W	0
7	RXFULLIE	RX FIFO 满中断使能位 0: 禁止; 1: 使能;	R/W	0
6	RXEMPTYIE	RX FIFO 空中断使能位 0: 禁止; 1: 使能;	R/W	0
5	Reserved		R	0
4	TXWERRIE	TX FIFO 写错误中断使能位 0: 禁止; 1: 使能;	R/W	0
3	TXRERRIE	TX FIFO 读错误中断使能位 0: 禁止; 1: 使能;	R/W	0
2	TXFULLIE	TX FIFO 满中断使能位 0: 禁止; 1: 使能;	R/W	0
1	TXEMPTYIE	TX FIFO 空中断使能位	R/W	0

		0: 禁止; 1: 使能;		
0	Reserved		R	0

10. 2. 10. INTR_STA (SPI FIFO 中断标志寄存器)

注: INTR_MASK 未屏蔽, 标志位才有效, 产生 SPI 外设中断。有关这些位的详细说明, 见 “RAW_INTR_STA”。

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	STOPIF	SPI 停止中断标志位 1: 产生中断; 0: 未产生中断;	R	0
9	RXWERRIF	RX FIFO 写错误中断标志位 1: 产生中断; 0: 未产生中断;	R	0
8	RXRERRIF	RX FIFO 读错误中断标志位 1: 产生中断; 0: 未产生中断;	R	0
7	RXFULLIF	RX FIFO 满中断标志位 1: 产生中断; 0: 未产生中断;	R	0
6	RXEMPTYIF	RX FIFO 空中断标志位 1: 产生中断; 0: 未产生中断;	R	0
5	Reserved		R	0
4	TXWERRIF	TX FIFO 写错误中断标志位 1: 产生中断; 0: 未产生中断;	R	0
3	TXRERRIF	TX FIFO 读错误中断标志位 1: 产生中断; 0: 未产生中断;	R	0
2	TXFULLIF	TX FIFO 满中断标志位 1: 产生中断; 0: 未产生中断;	R	0
1	TXEMPTYIF	TX FIFO 空中断标志位 1: 产生中断; 0: 未产生中断;	R	0
0	Reserved		R	0

10.2.11. RAW_INTR_STA (SPI FIFO 状态寄存器)

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	RAW_SPISTOPIF	SPI 主机: 如果 TX FIFO 为空, 或 RX FIFO 已满, 则在 FSM 进入 SPIM_IDLE 状态前产生中断; SPI 从机: 一次数据传输后将产生中断;	R	0
9	RAW_RXWERRIF	读溢出标志: 在 RX FIFO 已满后继续写入, 产生中断;	R	0
8	RAW_RXRERRIF	读为空标志: 在 RX FIFO 为空后继续读取, 产生中断;	R	0
7	RAW_RXFULLIF	RX FIFO 已满;	R	0
6	RAW_RXEMPTYIF	RX FIFO 为空;	R	0
5	Reserved		R	0
4	RAW_TXWERRIF	写溢出标志: 在 TX FIFO 已满后继续写入, 产生中断;	R	0
3	RAW_TXRERRIF	写为空标志: 在 TX FIFO 为空后继续发送, 产生中断;	R	0
2	RAW_TXFULLIF	TX FIFO 已满;	R	0
1	RAW_TXEMPTYIF	TX FIFO 为空;	R	0
0	Reserved		R	0

10.2.12. INTR_MASK (SPI 中断使能寄存器)

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	MASK_STOPEN	SPI 停止中断使能位 0: 禁止; 1: 使能;	R/W	0
9	MASK_RXWERREN	RX FIFO 写错误中断使能位 0: 禁止; 1: 使能;	R/W	0
8	MASK_RXRERREN	RX FIFO 读错误中断使能位 0: 禁止; 1: 使能;	R/W	0

7	MASK_RXFULLEN	RX FIFO 满中断使能位 0: 禁止; 1: 使能;	R/W	0
6	MASK_RXEMPTYEN	RX FIFO 空中断使能位 0: 禁止; 1: 使能;	R/W	0
5	Reserved		R	0
4	MASK_TXWERREN	TX FIFO 写错误中断使能位 0: 禁止; 1: 使能;	R/W	0
3	MASK_TXRERREN	TX FIFO 读错误中断使能位 0: 禁止; 1: 使能;	R/W	0
2	MASK_TXFULLEN	TX FIFO 满中断使能位 0: 禁止; 1: 使能;	R/W	0
1	MASK_TXEMPTYEN	TX FIFO 空中断使能位 0: 禁止; 1: 使能;	R/W	0
0	Reserved		R	0

10.2.13. INTR_CLR (SPI 中断标志清除寄存器)

偏移地址: 0x034

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	STOPIC	SPI 停止中断标志清除位 0: 没有作用; 1: 清中断;	W	0
9	RXWERRIC	RX FIFO 写错误中断标志清除位 0: 没有作用; 1: 清中断;	W	0
8	RXRERRIC	RX FIFO 读错误中断标志清除位 0: 没有作用; 1: 清中断;	W	0
7	RXFULLIC	RX FIFO 满中断标志清除位 0: 没有作用; 1: 清中断;	W	0
6	RXEMPTYIC	RX FIFO 空中断标志清除位 0: 没有作用; 1: 清中断;	W	0

5	Reserved		R	0
4	TXWERRIC	TX FIFO 写错误中断标志清除位 0: 没有作用; 1: 清中断;	W	0
3	TXRERRIC	TX FIFO 读错误中断标志清除位 0: 没有作用; 1: 清中断;	W	0
2	TXFULLIC	TX FIFO 满中断标志清除位 0: 没有作用; 1: 清中断;	W	0
1	TXEMPTYIC	TX FIFO 空中断标志清除位 0: 没有作用; 1: 清中断;	W	0
0	Reserved		R	0

11 VREF 基准电压源

11.1 VREF 概述

11.1.1. VREF 简介

AD3005 MCU 包含基准电压源 FVR。基准电压源（FVR）可用于为 ADC、DAC 提供基准电压，或由连接到 VREF 引脚的其他电路使用。

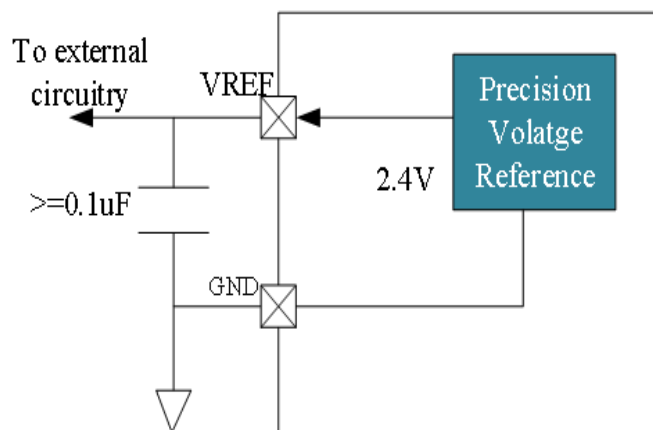
11.1.2. VREF 特性

基准电压源包括以下特性：

1. 输出至 VREF 引脚至源片外模拟电路；
2. 电压为固定的 2.4V。

11.1.3. VREF 使用

当使用基准电压源时，VREF 引脚应配置为模拟模式，并且需要至少 0.1 μF 的外部旁路电容接地。如下图所示。



VREF 基准电压源连接示意图

11.2 VREF 寄存器

基地址：0x4000 A000

11.2.1. FVRCON（FVR 控制寄存器）

偏移地址：0x014

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:19	Reserved		R	0
18:17	VREF2P4OUTSEL	电压 2.4V 的输出 IO 口 00: PA0; 01: PA1; 10: PA2; 11: PA3;	R/W	0
16	VREF2P4EN	VREF_2.4V 使能位 (VREF_2.4V 与 FVR 无关) 1: 使能; 0: 禁止;	R/W	0
15:8	Reserved		R	0
7	OUT2IOEN	FVR 输出到 PA7 使能 (应使能模拟输入输出) 1: 使能输出; 0: 禁止输出;	R/W	0
6	PGAEN	FVR 电压放大 1 倍使能位 1: 使能; 0: 禁止;	R/W	0
5	FVREN	FVR 使能位 1: 使能; 0: 禁止;	R/W	0
4	Reserved		R	0
3	Reserved		R	0
2:0	VOLTSEL	FVR 电压选择 000: 1.0V; 001: NTC; 010: 1.0V; 011: 1/4AVCC; 100: PD0; 101: PD1; 110: PD2; 111: PD3;	R/W	0

12 RAIL TO RAIL 模拟运放

12.1 模拟运放概述

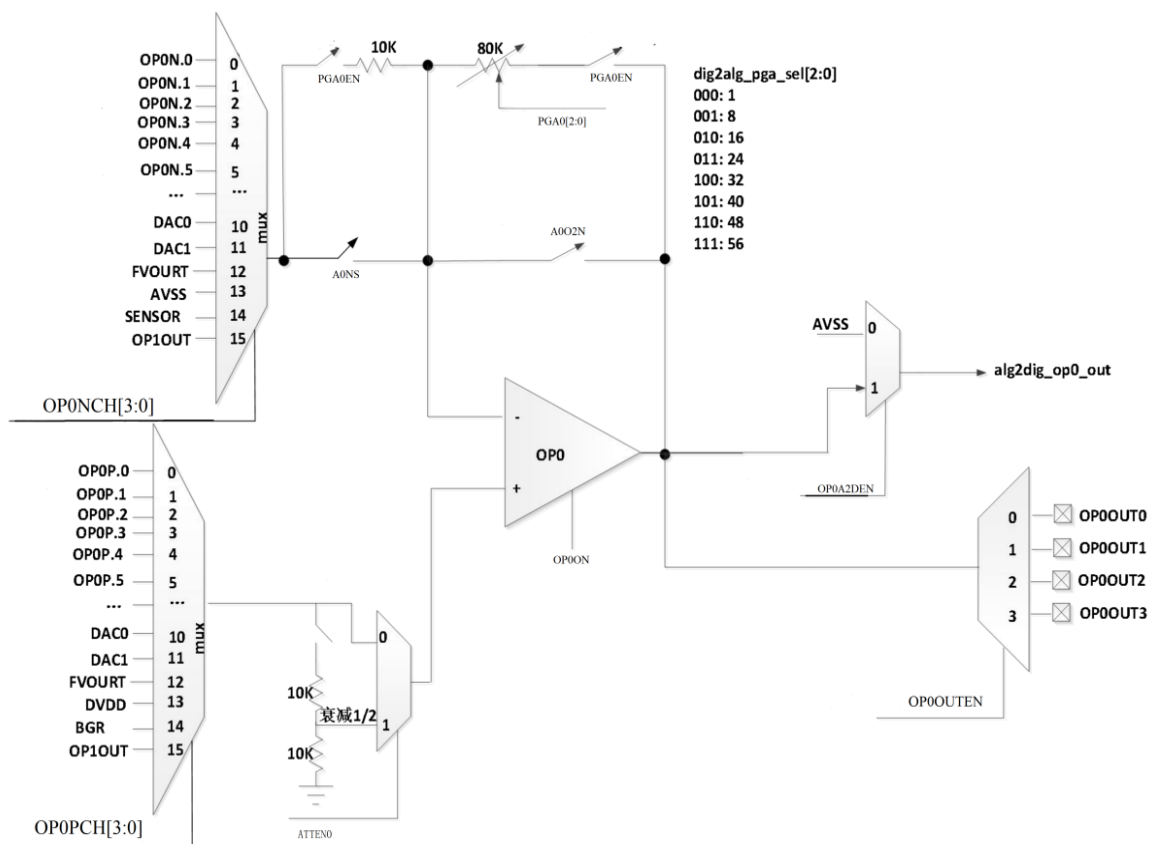
模拟运放是一种线性集成放大电路，不仅可以对微弱的电信号进行放大，还可以进行电压的比较和跟随。

注：OP 的输出引脚，除了配置功能复用外还需要配置 I/O 口状态为输入。

12.2 运放 OP0

12.2.1. OP0 运放模式

运放引脚与 GPIO 引脚共用，由 OP0ON 位控制，OP0ON = 1 时，由 OP0NCH 选择的引脚为运放的负极输入引脚，由 OP0PCH 选择的引脚作为运放正极输入引脚。有 OP0OCH3~0 控制运放输出引脚是否与 GPIO 连接，OP0OCH.n = 1 时，运放的输出端连接至 GPIO 口，并禁止 GPIO 功能。当 OP0A2DEN = 1 时，可以通过 OP0OUT 标志位读取运放的输出状态。



OP0 运放示意图

12.2.2. OP0 运放寄存器

基地址：0x4000 7000

12.2.2.1. OP0CON0 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:27	Reserved		R	0
26	POL	OP0 输入数字波形极性控制位 0: 同向; 1: 反向;	R/W	0
25	COMPEN	OP0 比较器模式使能位 0: 运放模式; 1: 比较器模式;	R/W	0
24:23	OUT2IOCH	OP0 运放输出结果到 GPIO 00: 输出到 PC1; 01: 输出到 PC3; 10: 输出到 PC5; 11: 输出到 PB4;	R/W	0
22	OUTFEN	OP0 输出滤波使能位 0: 禁止; 1: 使能;	R/W	0
21:18	OUTFOTS	OP0 输出滤波时间设置 滤波时间 = (OUTFOTS [3:0] * 2 + 1) * OP0CLK;	R/W	0
17	OUT2IOEN	OP0 运放输出结果到 GPIO 使能位 0: 禁止; 1: 使能;	R/W	0
16	OUT2DEN	OP0 运放输出到数字使能位 0: 禁止; 1: 使能;	R/W	0
15	AONS	将运放输入端与 OP0 NCH 选择的信号短接使能位 0: 断开; 1: 短接;	R/W	0
14	A002N	将运放输出端与输入负端短接, 形成 Buffer 使能位 0: 断开; 1: 短接;	R/W	0
13	PGAEN	OP0 PGA 使能位 0: 禁止; 1: 使能;	R/W	0
12: 10	PGA[2:0]	OP0 运放比例控制 000: 1 倍;	R/W	0

		001: 8 倍; 010: 16 倍; 011: 24 倍; 100: 32 倍; 101: 40 倍; 110: 48 倍; 111: 56 倍;		
9	INTPE	OP0 中断触发沿上升沿控制位 1: 上升沿触发 (OP0OUT 由低到高); 0: 关闭;	R/W	0
8	INTNE	OP0 中断触发沿下降沿控制位 1: 下降沿触发 (OP0OUT 由高到低); 0: 关闭;	R/W	0
7:4	PCH[3:0]	OP0 的正端 0000: OP0P. 0 (PA1); 0001: OP0P. 1 (PA2); 0010: OP0P. 2 (PA4); 0011: OP0P. 3 (PA5); 0100: OP0P. 4 (PA6); 0101: OP0P. 5 (PA7); 0110: OP0P. 6 (PB0); 0111: OP0P. 7 (PB1); 1000: OP0P. 8 (PB2); 1001: OP0P. 9 (PB7); 1010: DAC0; 1011: DAC1; 1100: FVROUT; 1101: DVDD; 1110: BGR; 1111: OP1OUT;	R/W	0
3:0	NCH[3:0]	OP0 的负端 0000: OP0N. 0 (PA1); 0001: OP0N. 1 (PA2); 0010: OP0N. 2 (PA4); 0011: OP0N. 3 (PA5); 0100: OP0N. 4 (PA6); 0101: OP0N. 5 (PA7); 0110: OP0N. 6 (PB0); 0111: OP0N. 7 (PB1); 1000: OP0N. 8 (PB2); 1001: VREF2P4; 1010: DAC0; 1011: DAC1;	R/W	0

		1100: FVR0UT; 1101: AVSS; 1110: SENSOR; 1111: OP10UT;		
--	--	--	--	--

12.2.2.2. OP0CON1 控制寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	DMAEN	OP0 的 DMA 使能位 0: DMA 禁止; 1: DMA 使能;	R/W	
3	ATTEN	OP0 正向端输入衰减 50%使能位 0: 不衰减; 1: 衰减 50%;	R/W	0
2	OUTR	OP0 输出实时结果控制位 0: 输出 0; 1: 输出 1;	R/W	0
1	IE	OP0 中断使能位 0: 禁止; 1: 使能;	R/W	0
0	OPOEN	OP0 使能位 0: 禁止; 1: 使能;	R/W	0

12.2.2.3. OP0CON2 控制寄存器

偏移地址: 0x008

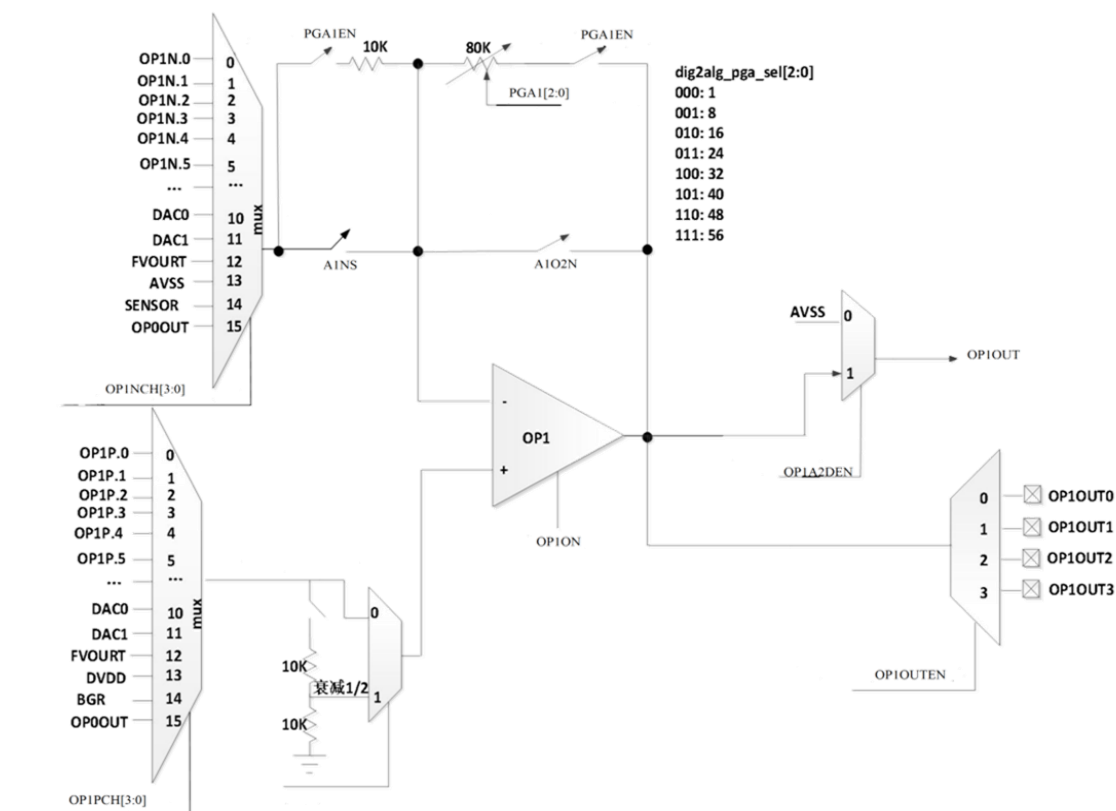
Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	IF	OP0 中断标志, 写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R/W	0

12.3 运放 OP1

12.3.1. OP1 运放模式

运放引与 GPIO 引脚共用, 由 OP1ON 位控制, OP1ON=1 时。有 OP1OCH3~0 控制运放输出引脚是否与 GPIO 连接, OP1OCH.n=1 时, 运放的输出端连接至 GPIO 口, 并禁止 GPIO

功能。可以通过 OP1OUT 标志位读取比较器的输出状态。



OP1 运放示意图

12.3.2. OP1 运放寄存器

基地址：0x4000 8000

12.3.2.1. OP1CON0 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:27	Reserved		R	0
26	POL	OP1 输入数字波形极性控制位 0: 同向; 1: 反向;	R/W	0
25	CMPEN	OP1 比较器模式使能位 0: 运放模式; 1: 比较器模式;	R/W	0
24:23	OUT2IOCH	OP1 运放输出结果到 GPIO 00: 输出到 PA4; 01: 输出到 PB2; 10: 输出到 PB5; 11: 输出到 PB6;	R/W	0

22	OUTFEN	OP1 输出滤波使能位 0: 禁止; 1: 使能;	R/W	0
21:18	OUTF1TS	OP1 输出滤波时间设置 滤波时间 = (OUTF1TS [3:0] * 2 + 1) * OP1CLK;	R/W	0
17	OUT2IOEN	OP1 运放输出结果到 GPIO 使能位 0: 禁止; 1: 使能;	R/W	0
16	OUT2DEN	OP1 运放输出到数字使能位 0: 禁止; 1: 使能;	R/W	0
15	A1NS	将运放输入端与 OP1 NCH 选择的信号短接使能位 0: 断开; 1: 短接;	R/W	0
14	A102N	将运放输出端与输入负端短接, 形成 Buffer 使能位 0: 断开; 1: 短接;	R/W	0
13	PGAEN	OP1 PGA 使能位 0: 禁止; 1: 使能;	R/W	0
12: 10	PGA[2:0]	OP1 运放比例控制 000: 1 倍; 001: 8 倍; 010: 16 倍; 011: 24 倍; 100: 32 倍; 101: 40 倍; 110: 48 倍; 111: 56 倍;	R/W	0
9	INTPE	OP1 中断触发沿上升沿控制位 1: 上升沿触发 (OP1OUT 由低到高); 0: 关闭;	R/W	0
8	INTNE	OP1 中断触发沿下降沿控制位 1: 下降沿触发 (OP1OUT 由高到低); 0: 关闭;	R/W	0
7:4	PCH[3:0]	OP1 的正端 0000: OP1P.0 (PA7); 0001: OP1P.1 (PB0); 0010: OP1P.2 (PC0); 0011: OP1P.3 (PC1); 0100: OP1P.4 (PC2); 0101: OP1P.5 (PC3);	R/W	0

		0110: OP1P. 6 (PC4); 0111: OP1P. 7 (PC5); 1000: OP1P. 8 (PC6); 1001: VREF2P4; 1010: DAC0; 1011: DAC1; 1100: FVR0UT; 1101: DVDD; 1110: BGR; 1111: OP00UT;		
3:0	NCH[3:0]	OP1 的负端 0000: OP1N. 0 (PA7); 0001: OP1N. 1 (PB0); 0010: OP1N. 2 (PC0); 0011: OP1N. 3 (PC1); 0100: OP1N. 4 (PC2); 0101: OP1N. 5 (PC3); 0110: OP1N. 6 (PC4); 0111: OP1N. 7 (PC5); 1000: OP1N. 8 (PC6); 1001: VREF2P4; 1010: DAC0; 1011: DAC1; 1100: FVR0UT; 1101: AVSS; 1110: SENSOR; 1111: OP00UT;	R/W	0
31:27	Reserved		R	0
26	POL	OP1 输入数字波形极性控制位 0: 同向; 1: 反向;	R/W	0
25	CMPEN	OP1 比较器模式使能位 0: 运放模式; 1: 比较器模式;	R/W	0

12.3.2.2. OP1CON1 控制寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:6	Reserve			
4	DMAEN	OP1 的 DMA 使能 0: DMA 禁止;	R/W	0

		1: DMA 使能;		
3	ATTEN	OP1 正向端输入衰减 50%使能 0: 不衰减; 1: 衰减 50%;	R/W	0
2	OUTR	OP1 输出实时结果 0: 输出 0; 1: 输出 1;	R/W	0
1	IE	OP1 中断使能位 0: 禁止; 1: 使能;	R/W	0
0	OP1EN	OP1 使能位 0: 禁止; 1: 使能;	R/W	0

12.3.2.3. OP1CON2 控制寄存器

偏移地址: 0x008

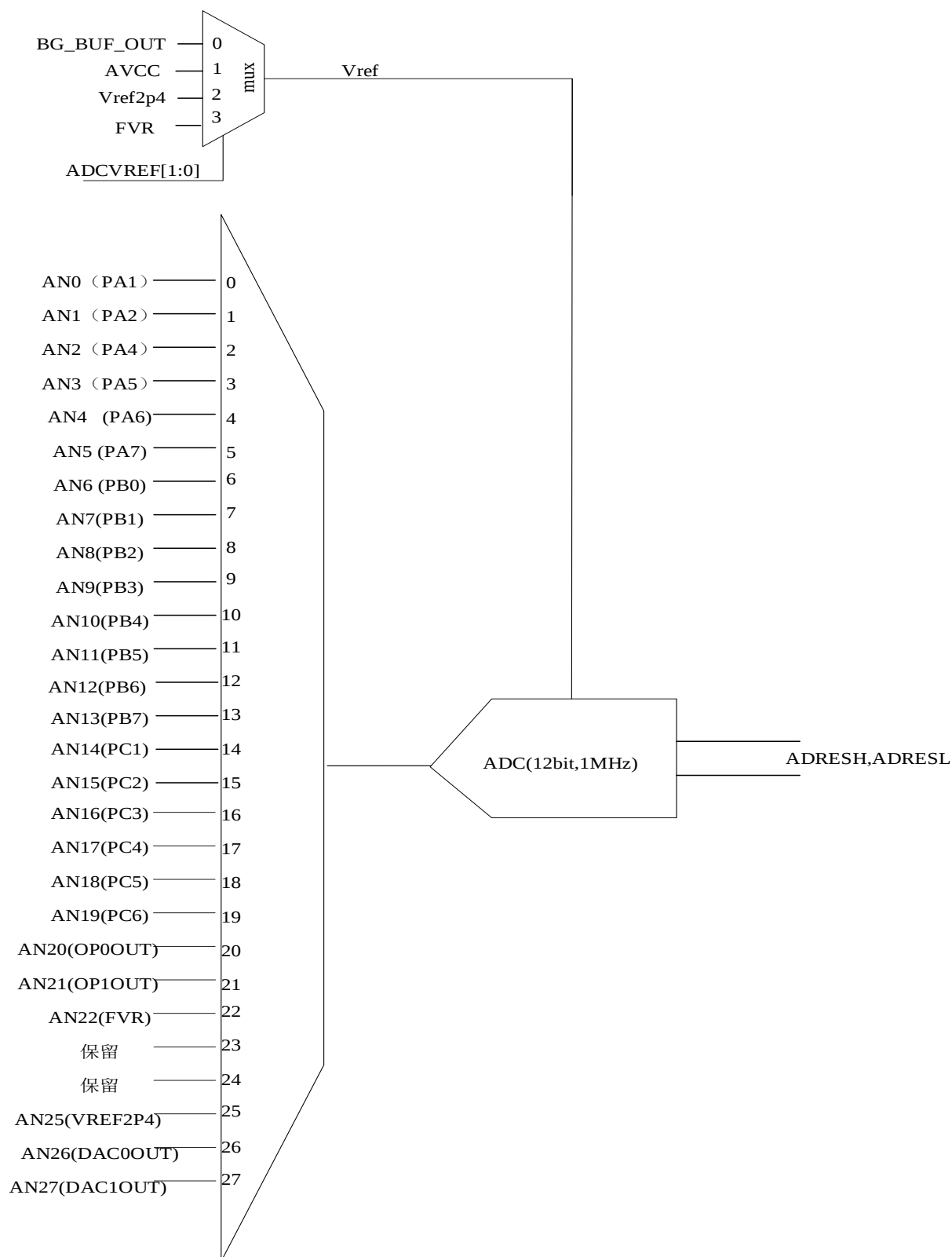
Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	IF	OP1 中断标志, 写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R/W	0

13 ADC 模数转换

13.1 ADC 模数转换概述

AD3005 MCU 包含一个 30 通道输入的 12 位 ADC，能够将一个模拟输入转换成 12 位数字信号。在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采样。采集时间一结束，即可启动 A/D 转换。当 A/D 转换完成后，转换结果被装入 ADRES 寄存器，GO/DONE 位被清零且 A/D 中断标志位 ADIF 被置 1。

13.1.1. ADC 模数转换结构示意图

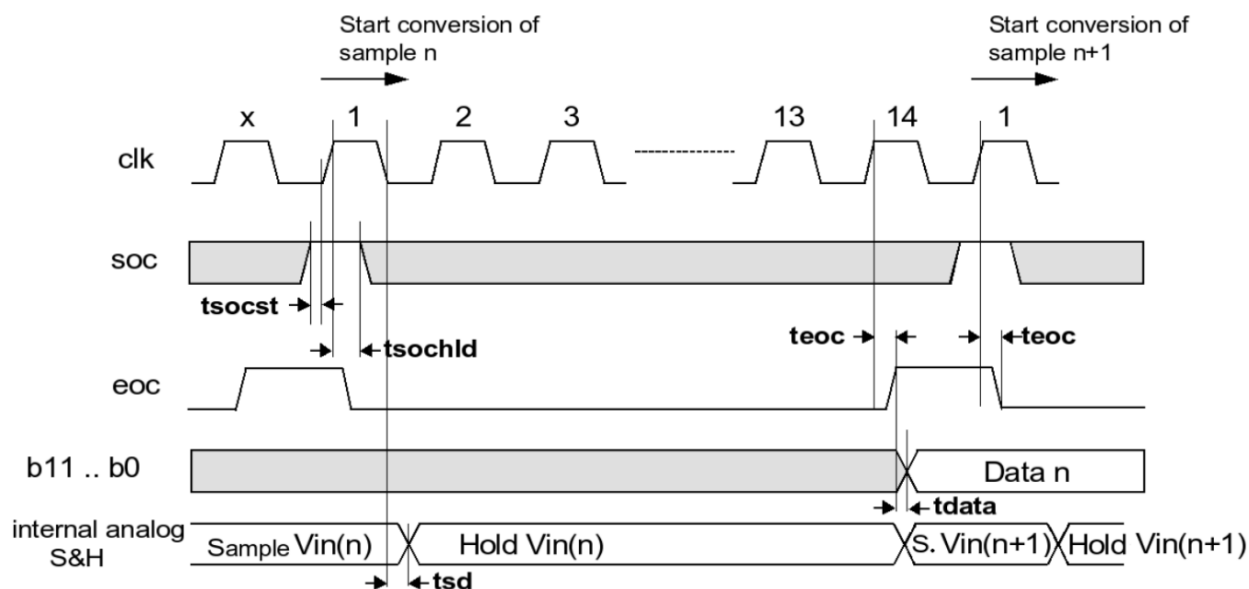


ADC 模数转换结构示意图

13.1.2. A/D 转换步骤

A/D 转换主要经过以下 7 个步骤：

1. 配置 A/D 模块
 - 选择参考电压（通过 ADCCON0[11:10]寄存器）
 - 选择 A/D 输入通道（通过 ADCON1[4:0]寄存器）
 - 选择 A/D 采集时间（通过 ADCON0[9:7]寄存器）
 - 选择 A/D 转换时钟（通过 ADCON0[6:4]寄存器）
 - 使能 A/D 模块（通过 ADCON0[0]寄存器）
2. 需要时，配置 A/D 中断
 - 清零 ADCIF 位
 - 将 ADCIE 位置 1
 - 使能总中断
3. 如果需要，需等待所需的采集时间。
4. 启动转换：（注：使能 ADC 后，10 个时钟周期可以启动转换，ADC 转换周期为 15 个时钟周期）
 - 将 GO/DONG 位置 1（ADCON0[1]）
5. 等待 A/D 转换完成，通过以下两种方法之一判断转换是否完成：
 - 1) 查询 GO/DONE 位是否被清零；
 - OR
 - 2) 等待 A/D 中断；
6. 读取 A/D 结果寄存器（ADRES），需要时将 ADIF 位清零。
7. 如需再次进行 A/D 转换，返回步骤 1 或步骤 2。



A/D 转换时序示意图

13.2 ADC 模数转换寄存器

基地址：0x4000 A000

13.2.1. ADCCON0 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:13	Reserved		R/W	0
12	DMAEN	ADC 的 DMA 的使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
11:10	VREF[1:0]	ADC 参考电压选择 00: BGR (1V0); 01: AVCC; 10: VREF2P4; 11: FVR;	R/W	0
9:7	ACQT	ADC 采集时间选择 111: 15 T _{ADCLK} ; 110: 13 T _{ADCLK} ; 101: 11 T _{ADCLK} ; 100: 9 T _{ADCLK} ; 011: 7 T _{ADCLK} ; 010: 5 T _{ADCLK} ; 001: 3 T _{ADCLK} ;	R/W	0

		000: 0 TADCLK;		
6:4	TRATS	ADC 转换时钟选择（设置为高速的情况下） 110: $T_{ADCLK}/128$; 101: $T_{ADCLK}/64$; 100: $T_{ADCLK}/32$; 011: $T_{ADCLK}/16$; 010: $T_{ADCLK}/8$; 001: $T_{ADCLK}/4$; 000: $T_{ADCLK}/2$; 注：设置低速时，时钟分频数为高速下的分频数乘以 8，如低速情况下设置为 000 时，ADC 转换时钟为： $T_{ADCLK}/(2*8)$ ；	R/W	0
3	ADRST	ADC 模拟模块复位，用户可以不用设置，置为 0；	R/W	0
2	SPEED	ADC 速度控制位 1: 高速； 0: 低速；	R/W	0
1	IE	ADC 中断使能位 1: 使能； 0: 禁止；	R/W	0
0	ADEN	ADC 转换使能位 1: 使能； 0: 禁止；	R/W	0

13.2.2. ADCCON1 通道选择寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4:0	ADCH[4:0]	ADC 模拟通道选择 00000: 通道 0 (PA1); 00001: 通道 1 (PA2); 00010: 通道 2 (PA4); 00011: 通道 3 (PA5); 00100: 通道 4 (PA6); 00101: 通道 5 (PA7); 00110: 通道 6 (PB0); 00111: 通道 7 (PB1); 01000: 通道 8 (PB2); 01001: 通道 9 (PB3); 01010: 通道 10 (PB4); 01011: 通道 11 (PB5);	R/W	0

		01100: 通道 12 (PB6); 01101: 通道 13 (PB7); 01110: 通道 14 (PC1); 01111: 通道 15 (PC2); 10000: 通道 16 (PC3); 10001: 通道 17 (PC4); 10010: 通道 18 (PC5); 10011: 通道 19 (PC6); 10100: OP0OUT; 10101: OP1OUT; 10110: FVR; 10111: 保留; 11000: 保留; 11001: VREF2P4; 11010: DAC0; 11011: DAC1;		
--	--	--	--	--

13.2.3. ADCCON2 使能寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	GO/DONE	ADC 转换状态位 当 ADEN = 1 时: Write: 1: 启动转换; 0: 无意义; Read: 1: A/D 转换正在进行; 0: ADC 空闲;	R/W	0

13.2.4. ADCCON3 控制寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	IF	ADC 中断标志位, 写 1 清中断标志 1: 产生中断; 0: 未产生中断;	R/W	0

13.2.5. ADCRES 转换结果寄存器

偏移地址: 0x004

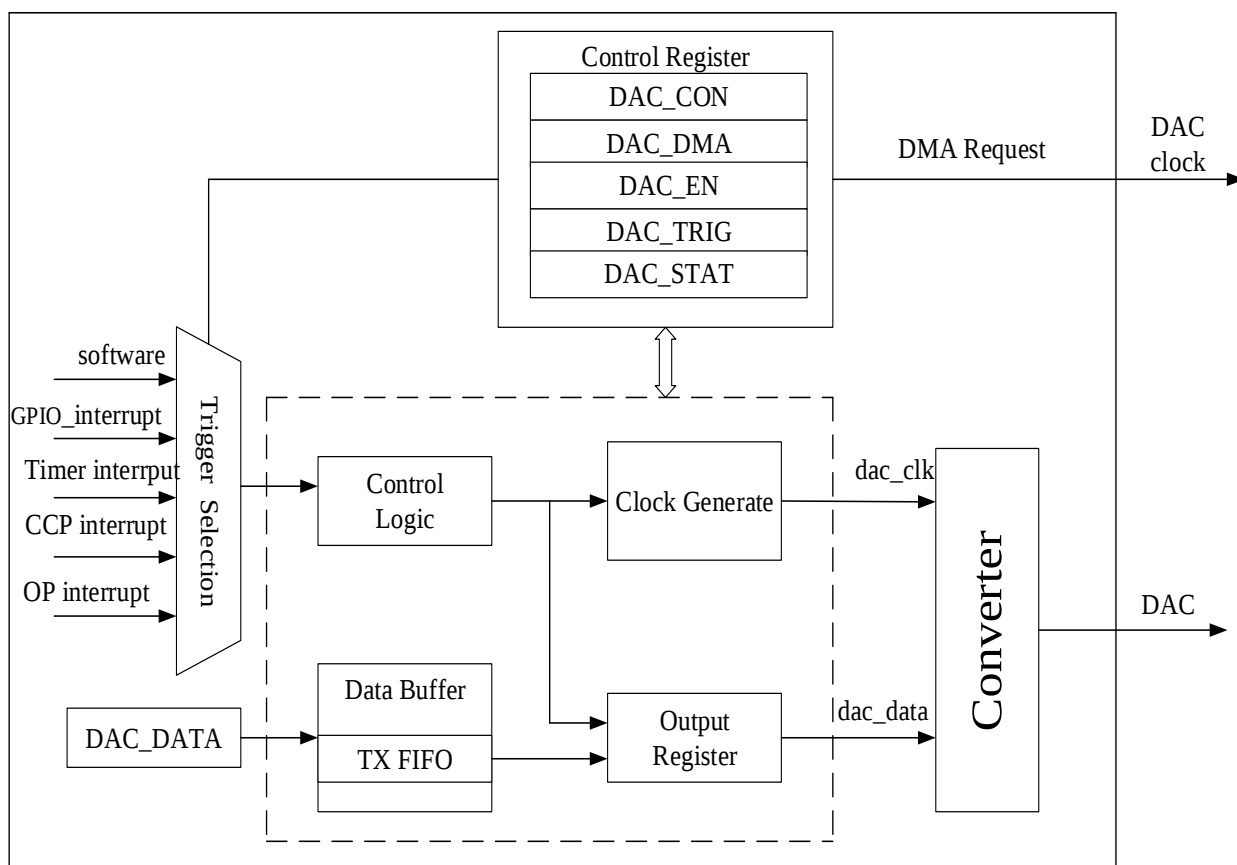
Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11::0	RESULTS	ADC 的转换结果;	R/W	0

14 DAC 数模转换器

14.1 DAC 数模转换器概述

数字/模拟转换模块（DAC）是 12 位电压输出的数字/模拟转换器，可以与 DMA 控制寄存器配合使用。DAC 有两个输出通道，每个 DAC 通道都有各自的转换器。

14.1.1. DAC 数模转换器结构示意图



DAC 结构示意图

14.1.2. DAC 数模转换器特性

DAC 数模转换器具有以下特性：

- 12 位电压输出的数字/模拟转换器；
- 支持外部触发，CCP 触发，定时器软件触发转换；
- 支持高达 1MHz 的转换；
- 有 DMA 功能；
- 提供锯齿波和三角波的产生；

- 提供数据输出反转功能。

14.2 DAC 数模转换器功能介绍

14.2.1. 输出电压

经过 DAC 线性转换后，数字寄存器的输入数值会转换为相应的输出电压值，每个 DAC 通道引脚的模拟输出电压由以下方程计算：

$$V_{OUT} = V_{REF} \times \left(\frac{Data}{4096} \times 83.95\% + 6.62\% \right)$$

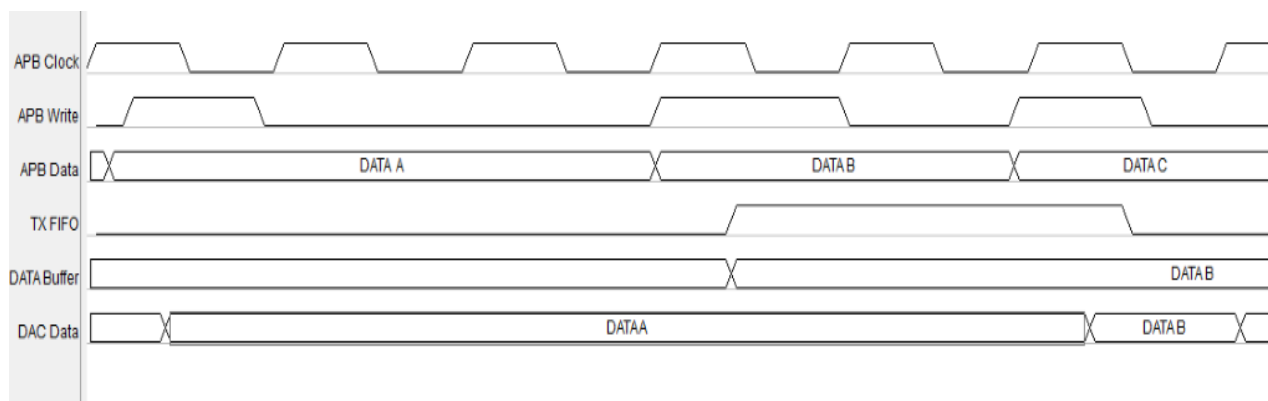
14.2.2. 触发选择

如果设置了 DAC_CON.TRIEN 控制位，则转换可以由外部事件（定时计数器）触发。DAC_CON.TRISSEL<4:0>控制位确定 17 个可能事件中的哪一个将触发转换。每次 DAC 接口检测到所选触发器上的上升沿，就传输存储在 DAC_DATA 寄存器中的数据。在触发发生后，接口 DAC_DATA 数据被更新一个 APB 周期。

注：DAC_CON 寄存器在使能 DAC 后不能被更改。

14.2.3. 数据转换

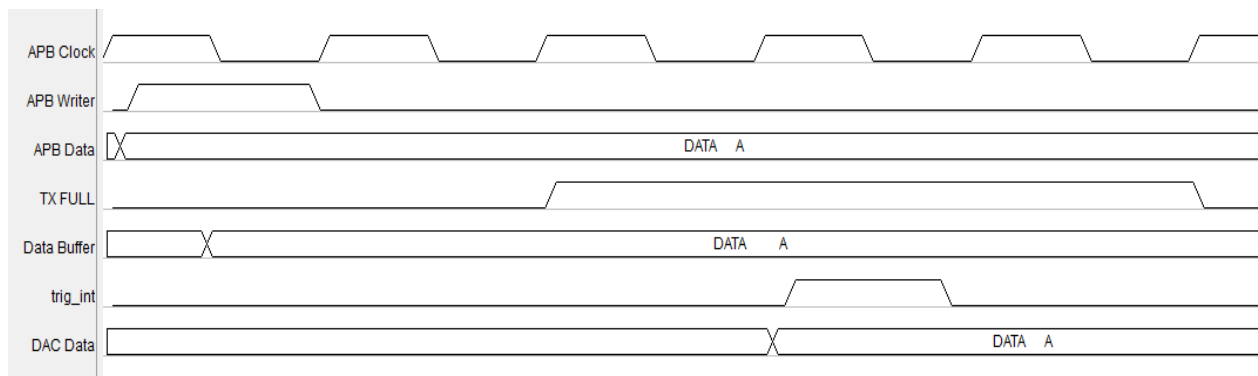
DAC 在数据传输中集成了数据缓冲器。当没有设置 DAC_CON.TRIEN 时，存储在 DAC_DATA 寄存器中的数据在 1 个 PCLK 周期后自动传输到数据输出逻辑。此时也可以写入寄存器 DAC_DATA，新数据将存储在数据缓冲区中，并在完成传输了先前的数据之后传输到数据输出逻辑。



数据传输的时序图（DAC_CTRL.TRIEN = 0）

如果设置了 DAC_CON.TRIEN 位，则当数据写入 DAC_DATA 寄存器时，数据将保存在

数据缓冲区中并等待触发器发生，数据传输将在 1 个 PCLK 周期之后开始。此时，对 DAC_DATA 数据寄存器的任何写入操作都是无用的。



数据传输时序图 (DAC_CTRL.TRIEN = 1)

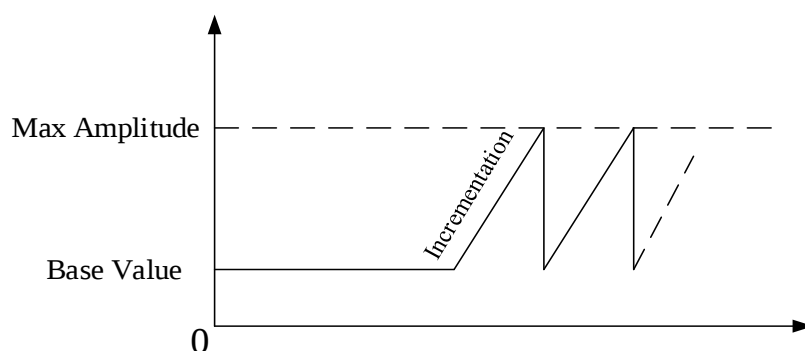
14.2.4. 波形生成

DAC 由 DAC_CON.WAVE <1:0> 配置，可以产生三种波形。

- 递增锯齿波；
- 递减锯齿波；
- 三角波。

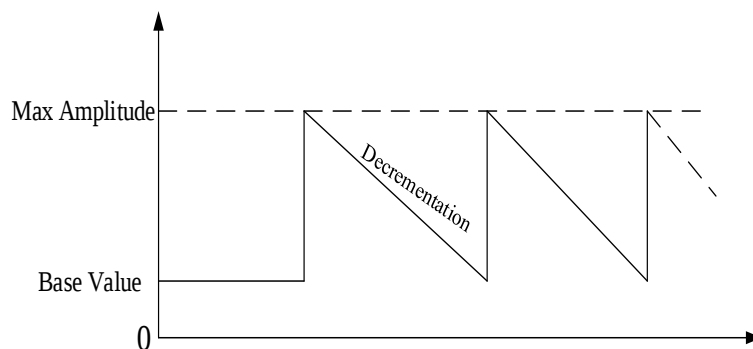
可以在直流或慢变信号上增加小振幅三角波，振幅通过 DAC_CON 寄存器中的 MAMP<3:0> 位来配置。内部计数器递增、递减或先递增，取决于 DAC_CON.WAVE<1:0> 位的设置。然后将该计数器的值加到 DAC_DATA 中而不溢出，并将其存储到数据缓冲器中。

在递增锯齿波模式下，只要计数器小于 DAC_CON.MAMP<3:0> 位所定义的最大幅度，计数器就递增。一旦达到配置的幅度，计数器返回到 0，然后再次递增，以此类推。



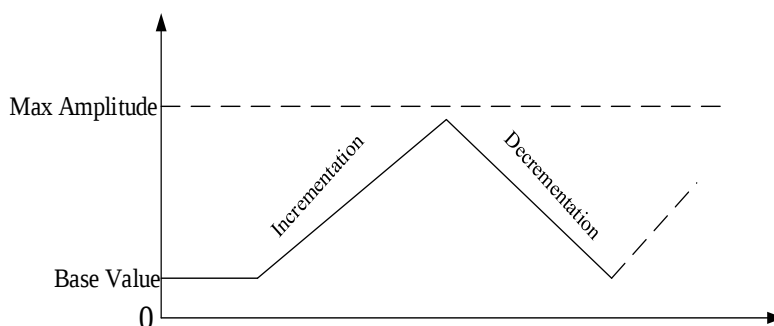
DAC 产生锯齿波示意图 (向上计数模式)

在递减锯齿波模式下，计数器递减到零。一旦到零，计数器被重新加载到由 DAC_CON.MAMP<3:0> 位定义的最大幅度，然后再次减量等等。



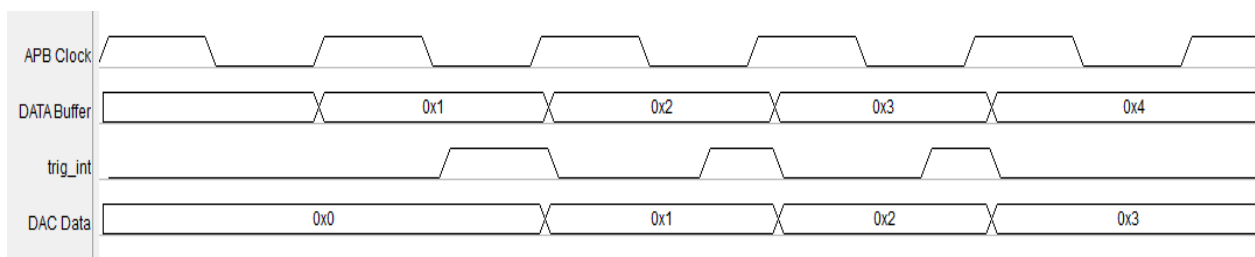
DAC 产生锯齿波示意图（向下计数模式）

在三角波模式下，只要计数器小于 DAC_CON.MAMP<3:0>定义的最大振幅，计数器就递增。一旦达到配置的幅度，计数器递减到 0，然后再次递增，以此类推。



DAC 产生三角波示意图（上/下计数）

可以通过清除 DAC_EN 寄存器来重置三角波的产生。



具有波形生成的 DAC 转换图（触发器启用）

14. 2. 5. 数据输出求反

启用 DAC_CON.INVREN 位，则输出数据将被反转。例如，将 DAC_DATA 数据写入 0xA、0xB 和 0xFF3，然后输出数据将为 0xFF5、0xFF4 和 0xC。

注：当在波生成模式中，此位没有效果。

14.2.6. DMA 请求

使用 DMA 向 DAC 写入数据，在 DAC_DMACH 寄存器中设置 TXDMAEN 位。当数据缓冲器为空时，DAC 将请求发送到 DMA，等待 DMA 写入单个数据。

14.3 DAC 数模转换器寄存器

14.3.1. DA0 寄存器

基地址：0x4000 9000

14.3.1.1. DA0CON 控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:9	TRIGSEL	DAC0 转换触发选择（每触发一次 DAC0 转换一次） 00000: DA0_STRIG; 00001: T0IF; 00010: T1IF; 00011: T2IF; 00100: T3IF; 00101: CCP0IF; 00110: CCP1IF; 00111: CCP2IF; 01000: CCP3IF; 01001: CCP4IF; 01010: CCP5IF; 01011: PAIF; 01100: PBIF; 01101: PCIF; 01110: PDIF; 01111: OP0IF; 10000: OP1IF;	R/W	0
8	TRIGEN	DAC0 触发使能位 0: 禁止; 1: 使能;	R/W	0
7	Reserved		R	0
6:3	MAMP	DAC0 屏蔽振幅选择 0000: 不屏蔽 bit0/振幅等于 1; 0001: 不屏蔽 bits<1:0>/振幅等于 3;	R/W	0

		0010: 不屏蔽 bits<2:0>/振幅等于 7; 0011: 不屏蔽 bits<3:0>/振幅等于 15; 0100: 不屏蔽 bits<4:0>/振幅等于 31; 0101: 不屏蔽 bits<5:0>/振幅等于 63; 0110: 不屏蔽 bits<6:0>/振幅等于 127; 0111: 不屏蔽 bits<7:0>/振幅等于 255; 1000: 不屏蔽 bits<8:0>/振幅等于 511; 1001: 不屏蔽 bits<9:0>/振幅等于 1023; 1010: 不屏蔽 bits<10:0>/振幅等于 2047; >=1011: 不屏蔽 bits<11:0>/振幅等于 4095;		
2:1	WAVE	DAC0 波形生成使能 00: 普通 DAC 输出; 01: 生成递增波形; 10: 生成递减波形; 11: 生成三角波;	R/W	0
0	INVEN	DAC0 输出数据反转控制位 1: 反转; 0: 未反转;	R/W	0

14.3.1.2. DA0ST 状态寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	TXFULL	DAC0 TX FIFO 存储器满标志位 0: 未滿; 1: 已滿;	R	0
0	BUSY	DAC0 忙碌标志位 0: 空闲; 1: 忙碌;	R	0

14.3.1.3. DA0EN 使能寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	DA0EN	DAC0 使能位, 默认从 PDO 输出 0: 禁止; 1: 使能;	R/W	0
1:0	VREF[1:0]	DAC0 参考电压选择位	R/W	0

		00: VREF2P4; 01: AVCC;		
--	--	---------------------------	--	--

14.3.1.4. DA0DMA 使能寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DMAEN	DAC0 DMA 使能位 0: DMA 禁止; 1: DMA 使能;	R/W	0

14.3.1.5. DA0TRIG 触发寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	STRIGEN	DAC0 软件触发使能位 0: 禁止; 1: 使能;	R/W	0

14.3.1.6. DA0_DATA 数据寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11: 0	TXDATA	数据寄存器;	R/W	0

14.3.2. DA1 寄存器

基地址: 0x4000 9000

14.3.2.1. DA1CON 控制寄存器

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:9	TRIGSEL	DAC1 转换触发选择 00000: DA0_STRIG; 00001: T0IF; 00010: T1IF;	R/W	0

		00011: T2IF; 00100: T3IF; 00101: CCP0IF; 00110: CCP1IF; 00111: CCP2IF; 01000: CCP3IF; 01001: CCP4IF; 01010: CCP5IF; 01011: PAIF; 01100: PBIF; 01101: PCIF; 01110: PDIF; 01111: OP0IF; 10000: OP1IF;		
8	TRIGEN	DAC1 触发使能位 0: 禁止; 1: 使能;	R/W	0
7	Reserved			0
6:3	MAMP	DAC1 屏蔽振幅选择 0000: 不屏蔽 bit0/振幅等于 1; 0001: 不屏蔽 bits<1:0>/振幅等于 3; 0010: 不屏蔽 bits<2:0>/振幅等于 7; 0011: 不屏蔽 bits<3:0>/振幅等于 15; 0100: 不屏蔽 bits<4:0>/振幅等于 31; 0101: 不屏蔽 bits<5:0>/振幅等于 63; 0110: 不屏蔽 bits<6:0>/振幅等于 127; 0111: 不屏蔽 bits<7:0>/振幅等于 255; 1000: 不屏蔽 bits<8:0>/振幅等于 511; 1001: 不屏蔽 bits<9:0>/振幅等于 1023; 1010: 不屏蔽 bits<10:0>/振幅等于 2047; >=1011: 不屏蔽 bits<11:0>/振幅等于 4095;	R/W	0
2:1	WAVE	DAC1 波形生成使能 00: 普通 DAC 输出; 01: 生成递增波形;	R/W	0
0	INVEN	DAC1 输出数据反转使能位 1: 反转; 0: 未反转;	R/W	0

14.3.2.2. DA1ST 状态寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:2	Reserved		R	0
1	TXFULL	DAC1 TX FIFO 存储器满标志位 0: 未滿; 1: 已滿;	R	0
0	BUSY	DAC1 忙碌标志位 0: 空闲; 1: 忙碌;	R	0

14.3.2.3. DA1EN 使能寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	DA1EN	DAC1 使能位, 默认从 PD1 输出 0: 禁止; 1: 使能;	R/W	0
1:0	VREF[1:0]	DAC1 参考电压选择 00: VREF2P4; 01: AVCC;	R/W	0

14.3.2.4. DA1DMA 使能寄存器

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DMAEN	DAC1 DMA 使能 0: DMA 禁止; 1: DMA 使能;	R/W	0

14.3.2.5. DA1TRIG 触发寄存器

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	STRIGEN	DAC1 软件触发位 0: 禁止; 1: 使能;	R/W	0

14.3.2.6. DA1DATA 数据寄存器

偏移地址: 0x034

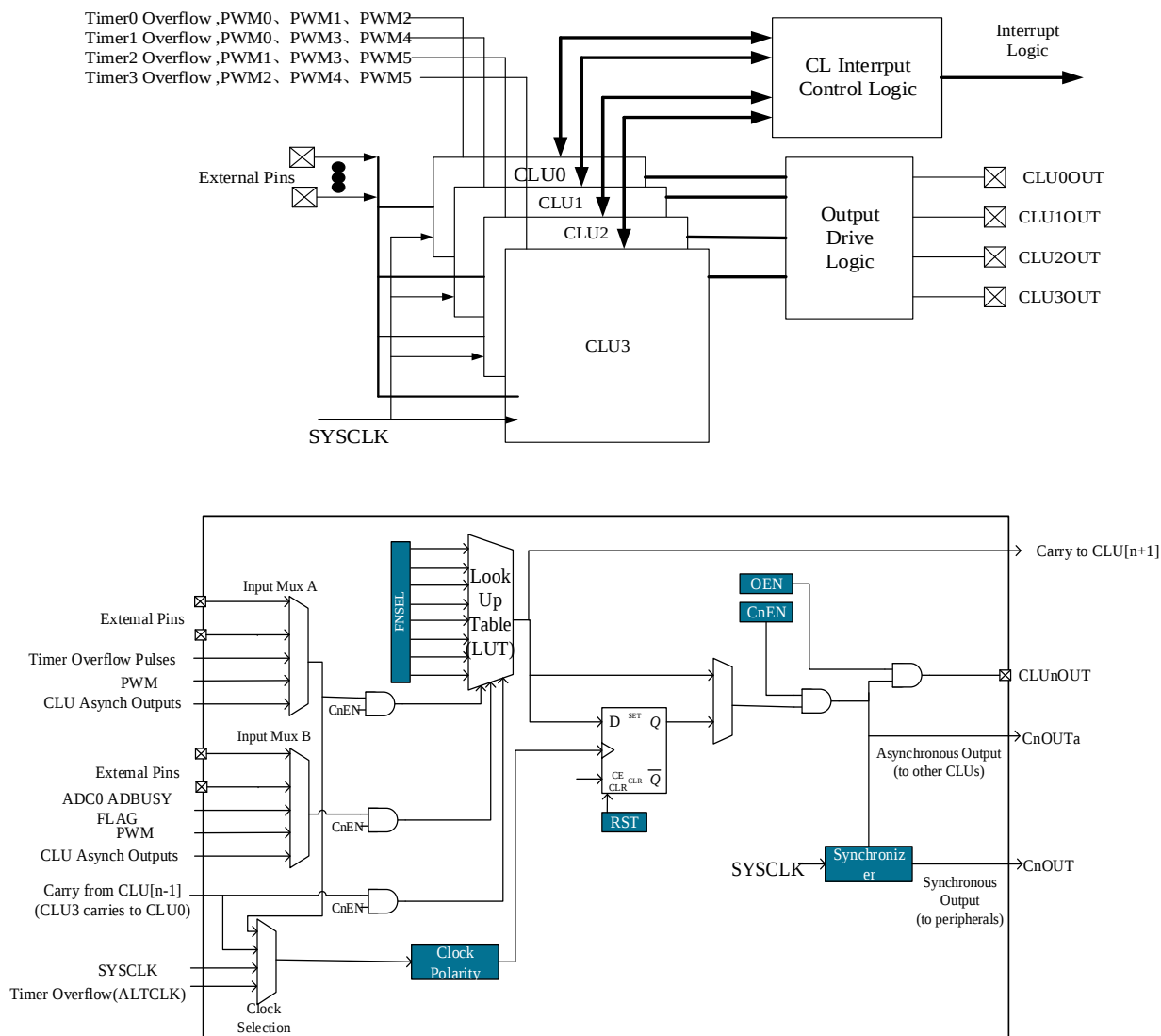
Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11:0	TXDATA	数据寄存器;	R/W	0

15 CLU 逻辑配置单元

15.1 CLU 逻辑配置单元概述

可配置逻辑（Configurable Logic Units）模块提供多个用户编程数字逻辑块，无需 CPU 干预即可运行。它由四个专用的独立可配置逻辑单元（CLU）组成，支持用户可编程异步和同步布尔逻辑操作。许多内部和外部信号可以用作每个 CLU 的输入，输出可以是路由到端口 I/O 或直接选择外设输入。

15.1.1. CLU 逻辑配置单元结构示意图



CLU 逻辑配置单元结构示意图

15.1.2. CLU 逻辑配置单元特性

可配置逻辑块的主要功能如下：

- 四个可配置逻辑单元（CLU），具有直接引脚和内部逻辑连接；
- 每个单元支持 256 种不同的组合逻辑功能（AND，OR，XOR，多路复用等），并包括用于同步操作的时钟触发器；
- 单元可以同步或异步操作；
- 可以级联在一起以执行更复杂的逻辑功能；
- 可与 UART 和 SPI 等串行外设或定时器和 PWM 通道等定时外设配合使用；
- 可用于同步和触发多个片上资源（ADC，DAC，定时器等）；
- 异步输出可用于从低功耗状态唤醒。

15.2 CLU 逻辑配置单元功能介绍

15.2.1. 配置顺序

在启用单个 CLU 之前，固件应配置功能选择，多路复用器输入和输出功能。CLU 初始化

包括以下一般步骤：

1. 在 CLUnMX 中选择 LUT 的 A 和 B 输入；
2. 使用 CLUnFN 选择 LUT 功能；
3. 通过 CLUnCF 配置 CLU；
4. 如果为 CLU 选择 D 触发器输出（OUTSEL=1），建议设置 RST=1 将触发器输出复位为 0；
5. 设置 CLIE0 中所需的任何中断。每个模块的下降/上升沿中断使用 CnFIE/CnRIE 位使能；
6. 将 CLEN0 中的 CnEN 位置 1，使能 CLU。通过在 CLEN0 中设置多个位，固件可以同时启用多个 CLU；
7. 如果需要直接引脚输出，固件可以通过将 CLUnCF 中的 OEN 位置 1 来使能输出。

15.2.2. 输入多路复用器输入选择

每个 CLU 有两个主逻辑输入（A 和 B）和一个进位输入（C）。A 和 B 输入由 CLUnMX 寄存器中的 MXA 和 MXB 字段选择，可以是许多不同的内部和外部信号之一。当选择另一个 CLU 输出作为输入时，使用来自该 CLU 的异步输出，从而实现更复杂的布尔逻辑功能。

CLU 输入的引脚输入不是 SYSCLK 同步的。CLU 输入的其他内部外设（如定时器）是 SYSCLK 同步的，因为这些外设是 SYSCLK 同步的。对于捕获模式中的定时器，脉冲需要至少为 1 SYSCLK 宽才能保证捕获边沿。但是，仍然可以捕获更窄的脉冲。因此，包含 CLU 的固件不能依赖于定时器捕获小于 1 SYSCLK 周期宽度的脉冲。

注：当使用定时器溢出事件作为输入时，定时器溢出事件是一个脉冲，在一个 SYSCLK 周期内为逻辑高电平，在剩余的定时器周期内为逻辑低电平。进位输入 C 是前一个 CLU 的 LUT 输出。例如，CLU1 上的进位输入是 CLU0 的 LUT 输出。CLU0 的进位输入是 CLU3 的 LUT 输出。

CLU 输入选择表 1:

CLUnMX. MXA	CLU0A	CLU1A	CLU2A	CLU3A
0000	C0OUTa	C0OUTa	C0OUTa	C0OUTa
0001	C1OUTa	C1OUTa	C1OUTa	C1OUTa
0010	C2OUTa	C2OUTa	C2OUTa	C2OUTa
0011	C3OUTa	C3OUTa	C3OUTa	C3OUTa
0100	Timer0 Overflow	Timer1 Overflow	Timer2 Overflow	Timer3 Overflow
0101	PWM0	PWM0	PWM1	PWM2
0110	PWM1	PWM3	PWM3	PWM4
0111	PWM2	PWM4	PWM5	PWM5
1000	PA0	PA4	PA0	PA2
1001	PA2	PA5	PA1	PA3
1010	PA4	PB0	PB0	PA6
1011	PA6	PB2	PB1	PA7
1100	PB0	PB4	PB6	PB2
1101	PB2	PB5	PB7	PB3
1110	PB4	PC0	PC0	PC2

1111	PB6	PC2	PC1	PC3
------	-----	-----	-----	-----

CLU 输入选择表 2:

CLUnMX. MXB	CLU0B	CLU1B	CLU2B	CLU3B
0000	C0OUTa	C0OUTa	C0OUTa	C0OUTa
0001	C1OUTa	C1OUTa	C1OUTa	C1OUTa
0010	C2OUTa	C2OUTa	C2OUTa	C2OUTa
0011	C3OUTa	C3OUTa	C3OUTa	C3OUTa
0100	ADBUSH	ADBUSH	ADBUSH	ADBUSH
0101	PWM3	PWM1	PWM0	PWM0
0110	PWM4	PWM2	PWM2	PWM1
0111	PWM5	PWM5	PWM4	PWM3
1000	PA1	PA6	PA2	PA0
1001	PA3	PA7	PA3	PA1
1010	PA5	PB1	PB2	PA4
1011	PA7	PB3	PB3	PA5
1100	PB1	PB6	PB4	PB0
1101	PB3	PB7	PB5	PB1
1110	PB5	PC1	PC2	PC0
1111	PB7	PC3	PC3	PC1

15.2.3. 输出配置

每个 CLU 向系统提供异步和同步（同步到 **SYSCCLK**）输出。通过读取 **CLOUT0** 寄存器，固件可以随时读取同步输出。CLU 输出可以直接从 LUT 或从 a 导出锁存 D 型触发器输出，由 **CLUnCF** 中的 **OUTSEL** 位控制。当 CLU 被禁用时（**CLEN0** 中的 **CnEN** 为 0），它们都是输出将保持在逻辑 0。D 触发器时钟可以由四个源中的一个配置，由 **CLUnCF** 中的 **CLKSEL** 字段选择。触发器时钟可以

可选择使用 **CLKINV** 位进行反转。每个 CLU 都有以下选项来为其触发器提供时钟：

1. **CARRY_IN**: 来自先前 CLU 的进位（C）输入。第一个 CLU 使用最后一个 CLU 的进位。

2. MXA_INPUT: CLU 的 A 输入, 由 MXA 寄存器字段定义。
3. SYSCLK: 系统时钟。
4. ALTCLK: CLU0 的定时器 0 溢出, CLU1 的定时器 1 溢出, CLU2 的定时器 2 溢出, CLU3 的定时器 3 溢出。

当使用 D 触发器输出时, 可以通过将 1 写入 CLUnCF 中的 RST 位来随时将触发器复位为逻辑 0。输出不会保持在此复位状态 (复位后 RST 返回 0)。

CLU 输出也可以存在于所选引脚上。

CLU 输出信号到内部外设 (另一个 CLU 输入除外) 是 SYSCLK 同步的。CLU 输出信号到任何 CLU 输入都不是 SYSCLK 同步的。

15.2.4. LUT 配置

每个 CLU 中的布尔逻辑功能由 LUT 确定, 并且可以通过编程寄存器 CLUnFN 中的 FNSEL 字段来改变。LUT 实现为 8 输入多路复用器。FNSEL 的位映射到 8 个多路复用器输入, LUT 的输出通过 A, B 和 C 输入的组合来选择。

A Input	B Input	C Input	LUT Output
0	0	0	FNSEL. 0
0	0	1	FNSEL. 1
0	1	0	FNSEL. 2
0	1	1	FNSEL. 3
1	0	0	FNSEL. 4
1	0	1	FNSEL. 5
1	1	0	FNSEL. 6
1	1	1	FNSEL. 7

LUT 真值表

使用 LUT 可以实现任何 3 输入布尔逻辑功能。确定要编入 FNSEL 的值给定逻辑函数, 可以使用 LUT 真值表。例如, 要实现布尔函数 (A AND B), 对于 A 和 B 为 1 的任何组合, LUT 输出应为 1, 对于所有其他组合, LUT 输出应为 0。表中的最后两行 (对应于 FNSEL.7 和 FNSEL.6) 符合此条件, 因此 FNSEL 应编程为 11000000b 或 0xC0。

第二个例子, 如果需要函数 (A XOR B), 则对应于 FNSEL.2, FNSEL.3, FNSEL.4 和

FNSEL.5 的行将是逻辑 1，而对于 FNSEL.0，FNSEL1， FNSEL.6 和 FNSEL.7 则是逻辑 0。
因此，应将 FNSEL 编程为 00111100b 或 0x3C 以实现此功能。

15.3 CLU 逻辑配置单元寄存器

基地址：0x4000 C000

15.3.1. CLEN0 使能寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	C3DMAEN	CLU3 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
6	C2DMAEN	CLU2 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
5	C3DMAEN	CLU1 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
4	C0DMAEN	CLU0 的 DMA 使能位 1: DMA 使能; 0: DMA 禁止;	R/W	0
3	C3EN	CLU3 使能位 0: CLU3 禁止, 输出低电平; 1: CLU3 使能;	R/W	0
2	C2EN	CLU2 使能位 0: CLU2 禁止, 输出低电平; 1: CLU2 使能;	R/W	0
1	C1EN	CLU1 使能位 0: CLU1 禁止, 输出低电平; 1: CLU1 使能;	R/W	0
0	C0EN	CLU0 使能位 0: CLU0 禁止, 输出低电平; 1: CLU0 使能;	R/W	0

15.3.2. CLIE0 中断使能寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	C3RIE	CLU3 的上升沿产生中断使能位 0: 禁止; 1: 使能;	R/W	0
6	C3FIE	CLU3 的下降沿中断使能位 0: 禁止; 1: 使能;	R/W	0
5	C2RIE	CLU2 的上升沿产生中断使能位 0: 禁止; 1: 使能;	R/W	0
4	C2FIE	CLU2 的下降沿中断使能位 0: 禁止; 1: 使能;		
3	C1RIE	CLU1 的上升沿产生中断使能位 0: 禁止; 1: 使能;	R/W	0
2	C1FIE	CLU1 的下降沿中断使能位 0: 禁止; 1: 使能;	R/W	0
1	C0RIE	CLU0 的上升沿产生中断使能位 0: 禁止; 1: 使能;	R/W	0
0	C0FIE	CLU0 的下降沿中断使能位 0: 禁止; 1: 使能;	R/W	0

15.3.3. CLIF0 中断标志寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	C3RIF	CLU3 的上升沿（与 SYSCLK 同步）产生中断标志位 （必须由固件清除） 0: 未产生中断; 1: 产生中断;	R	0
6	C3FIF	CLU3 的下降沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0
5	C2RIF	CLU2 的上升沿产生中断标志位 0: 未产生中断;	R	0

		1: 产生中断;		
4	C2FIF	CLU2 的下降沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0
3	C1RIF	CLU1 的上升沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0
2	C1FIF	CLU1 的下降沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0
1	CORIF	CLU0 的上升沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0
0	COFIF	CLU0 的下降沿产生中断标志位 0: 未产生中断; 1: 产生中断;	R	0

15.3.4. CLUT0 输出寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	C3OUT	CLU3 输出;	R/W	0
2	C2OUT	CLU2 输出;	R/W	0
1	C1OUT	CLU1 输出;	R/W	0
0	C0OUT	CLU0 输出;	R/W	0

15.3.5. CLU0MX 多路复用寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:4	MXA	CLU0 A 输入多路复用器选择;	R/W	0
3:0	MXB	CLU0 B 输入多路复用器选择;	R/W	0

15.3.6. CLU0FN 功能选择寄存器

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	FNSEL	功能选择 CLU0 LUT。LUT 是一个 8 输入多路复用器，其输入是 FNSEL 的位。多路复用器选择信号为 (MS bit first): MXA, MXB, Carry-in;	R/W	0

例子:

FNSEL = 0xC0 实现: MXA 和 MXB

FNSEL = 0xE4 实现: (Carry & MXA) | ((~Carry) & MXB)

第二个例子是多路复用器，其中 Carry 用于选择 MXA 或 MXB。

15.3.7. CLU0CF 配置寄存器

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	OUTSEL	CLU 的输出选择位 0: 选择 D 触发器输出; 1: 选择 LUT 输出;	R/W	0
6	OUTEN	CLU 端口异步输出到选定的 GPIO 引脚使能位 0: 禁止; 1: 使能;	R/W	0
5:4	Reserved		R	0
3	RST	CLU 的 D 触发器的复位控制位 0: 未复位; 1: 复位;	R/W	0
2	CLKINV	CLU D 触发器时钟信号反转控制位 0: 未反转; 1: 反转;	R/W	0
1:0	CLKSEL	CLKSEL 的时钟选择 00 : MXA_INPUT MXA 输入; 01 : CARRY_IN 进位信号; 10 : ALTCLK 备用时钟信号 CLU0ALTCLK; 11 : SYSClk;	R/W	0

15.3.8. CLU1MX 多路复用寄存器

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:8	Reserved		R	0
7:4	MXA	CLU1 A 输入多路复用器选择;	R/W	0
3:0	MXB	CLU1 B 输入多路复用器选择;	R/W	0

15.3.9. CLU1FN 功能选择寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	FNSEL	功能选择 CLU1 LUT。LUT 是一个 8 输入多路复用器，其输入是 FNSEL 的位。多路复用器选择信号为 (MS bit first): MXA, MXB, Carry-in;	R/W	0

例子:

FNSEL = 0xC0 实现: MXA 和 MXB

FNSEL = 0xE4 实现: (Carry & MXA) | ((~Carry) & MXB)

第二个例子是多路复用器，其中 Carry 用于选择 MXA 或 MXB。

15.3.10. CLU1CF 配置寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	OUTSEL	CLU 的输出选择 0: 选择 D 触发器输出; 1: 选择 LUT 输出;	R/W	0
6	OUTEN	CLU 端口异步输出到选定的 GPIO 引脚使能位 0: 禁止; 1: 使能;	R/W	0
5:4	Reserved		R	0
3	RST	CLU 的 D 触发器的复位控制位 0: 未复位; 1: 复位;	R/W	0
2	CLKINV	CLU D 触发器时钟信号反转控制位 0: 未反转; 1: 反转;	R/W	0
1:0	CLKSEL	CLKSEL 的时钟选择 00 : MXA_INPUT MXA 输入;	R/W	0

		01 : CARRY_IN 进位信号; 10 : ALTCLK 备用时钟信号 CLU0ALTCLK; 11 : SYSCLK;		
--	--	---	--	--

15.3.11. CLU2MX 多路复用寄存器

偏移地址: 0x02C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:4	MXA	CLU2 A 输入多路复用器选择;	R/W	0
3:0	MXB	CLU2 B 输入多路复用器选择;	R/W	0

15.3.12. CLU2FN 功能选择寄存器

偏移地址: 0x030

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	FNSEL	功能选择 CLU2 LUT。LUT 是一个 8 输入多路复用器, 其输入是 FNSEL 的位。多路复用器选择信号为 (MS bit first): MXA, MXB, Carry-in;	R/W	0

例子:

FNSEL = 0xC0 实现: MXA 和 MXB

FNSEL = 0xE4 实现: (Carry & MXA) | ((~Carry) & MXB)

第二个例子是多路复用器, 其中 Carry 用于选择 MXA 或 MXB。

15.3.13. CLU2CF 配置寄存器

偏移地址: 0x034

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	OUTSEL	CLU 的输出选择位 0: 选择 D 触发器输出; 1: 选择 LUT 输出;	R/W	0
6	OUTEN	CLU 端口异步输出到选定的 GPIO 引脚使能位 0: 禁止; 1: 使能;	R/W	0
5:4	Reserved		R	0

3	RST	CLU 的 D 触发器的复位控制位 0: 未复位; 1: 复位;	R/W	0
2	CLKINV	CLU D 触发器时钟信号反转控制位 0: 未反转; 1: 反转;	R/W	0
1:0	CLKSEL	CLKSEL 的时钟选择 00 : MXA_INPUT MXA 输入; 01 : CARRY_IN 进位信号; 10 : ALTCLK 备用时钟信号 CLU0ALTCLK; 11 : SYSCLK;	R/W	0

15.3.14. CLU3MX 多路复用寄存器

偏移地址: 0x038

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:4	MXA	CLU3 A 输入多路复用器选择;	R/W	0
3:0	MXB	CLU3 B 输入多路复用器选择;	R/W	0

15.3.15. CLU3FN 功能选择寄存器

偏移地址: 0x03C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	FNSEL	功能选择 CLU3 LUT。LUT 是一个 8 输入多路复用器, 其输入是 FNSEL 的位。多路复用器选择信号为 (MS bit first): MXA, MXB, Carry-in;	R/W	0

例子:

FNSEL = 0xC0 实现: MXA 和 MXB

FNSEL = 0xE4 实现: (Carry & MXA) | ((~Carry) & MXB)

第二个例子是多路复用器, 其中 Carry 用于选择 MXA 或 MXB。

15.3.16. CLU3CF 配置寄存器

偏移地址: 0x040

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:8	Reserved		R	0
7	OUTSEL	CLU 的输出选择位 0: 选择 D 触发器输出; 1: 选择 LUT 输出;	R/W	0
6	OUTEN	CLU 端口异步输出到选定的 GPIO 引脚使能位 0: 禁止; 1: 使能;	R/W	0
5:4	Reserved		R	0
3	RST	CLU 的 D 触发器的复位控制位 0: 未复位; 1: 复位;	R/W	0
2	CLKINV	CLU D 触发器时钟信号反转控制位 0: 未反转; 1: 反转;	R/W	0
1:0	CLKSEL	CLKSEL 的时钟选择 00 : MXA_INPUT MXA 输入; 01 : CARRY_IN 进位信号; 10 : ALTCLK 备用时钟信号 CLUOALTCLK; 11 : SYSCLK;	R/W	0

15.3.17. CLUIC0 中断清除寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	C3RIC	CLU3 上升沿中断标志清除位 0: 没有作用; 1: 清除中断;	R/W	0
6	C3FIC	CLU3 下降沿中断标志清除位 0: 未清除中断标志; 1: 清除中断标志;	R/W	0
5	C2RIC	CLU2 上升沿中断标志清除位 0: 没有作用; 1: 清除中断;	R/W	0
4	C2FIC	CLU2 下降沿中断标志清除位 0: 未清除中断标志; 1: 清除中断标志;	R/W	0
3	C1RIC	CLU1 上升沿中断标志清除位 0: 没有作用; 1: 清除中断;	R/W	0
2	C1FIC	CLU1 下降沿中断标志清除位	R/W	0

		0: 未清除中断标志; 1: 清除中断标志;		
1	CORIC	CLU0 上升沿中断标志清除位 0: 没有作用; 1: 清除中断;	R/W	0
0	COFIC	CLU0 下降沿中断标志清除位 0: 未清除中断标志; 1: 清除中断标志;	R/W	0

16 DMA 控制器

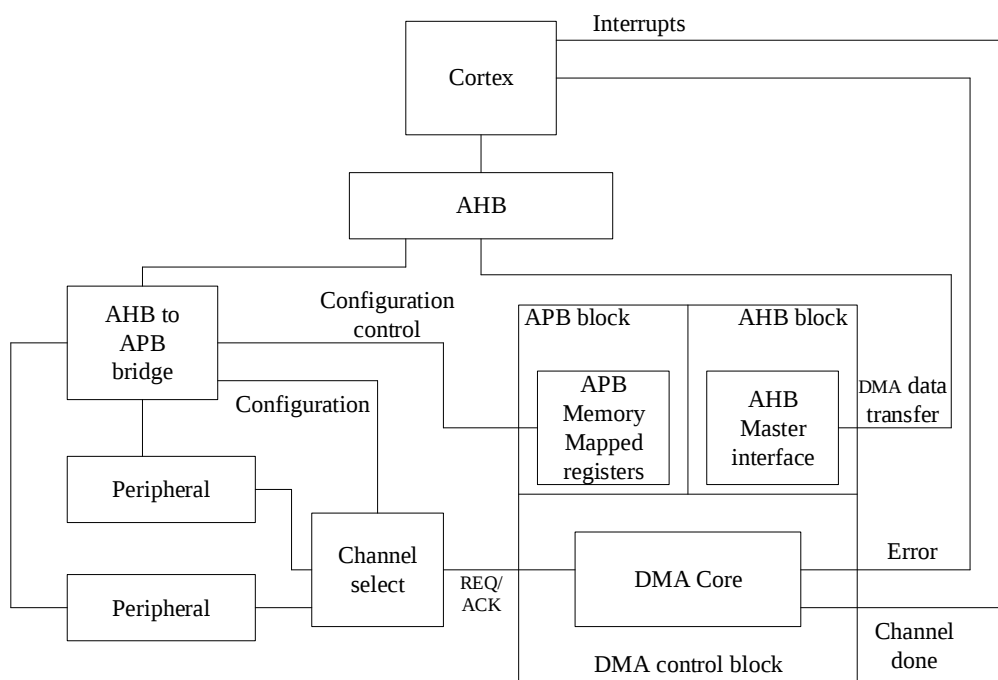
16.1 DMA 概述

直接内存访问（DMA）控制器独立于 CPU 执行内存操作，这具有减少能耗和 CPU 工作量的优势，并使系统能够保持低能耗模式，例如在将数据从 UART 移至 RAM。

DMA 控制器包括四个主要部分：

- APB 模块，允许软件配置 DMA 控制器；
- AHB 块，允许 DMA 读写 DMA 描述符的源和目标的数据；
- DMA 控制块，用于控制 DMA 的操作；
- 通道选择块，将外设 DMA 请求送入相应的 DMA 通道。

16.1.1. DMA 架构示意图



DMA 架构图

16.1.2. DMA 特性

1. DMA 控制器可作为存储器映射的外设访问；
2. 可能的数据传输包括：

- RAM /FLASH 到外设;
 - RAM 到 FLASH;
 - 外设到 RAM;
 - RAM /FLASH 到 RAM。
3. DMA 控制器具有 4 个独立通道;
 4. 每个通道都有主要和备用两个描述符;
 5. 每个通道的配置包括:
 - 传输模式配置;
 - 优先级配置;
 - 字计数配置;
 - 字大小配置。
 6. DMA 通道可以由以下几种源之一触发:
 - 通信模块 (UART, SPI, I2C);
 - 计时器 (TIMER);
 - 模拟模块 (DAC, ADC, OP);
 - 数字模块 (CCP, CLU);
 7. 通道号和外设之间的可编程映射-所有 DMA 通道均可由现有可用源触发;
 8. 转移完成后中断 DMA 支持数据传输, 在执行 UART 通信时, 能耗极低。

16.2 DMA 功能介绍

DMA 控制器具有高度的灵活性。它无需处理器内核的参与就能够在外设和内存之间传输数据。这减轻了处理器的工作量, 避免了复制大量数据或避免频繁中断服务外围设备。通过使 DMA 与 UART 自主工作进行数据传输, 它还可以用于减少系统功耗, 而不必将处理器内核从睡眠中唤醒。

DMA 控制器包含 4 个独立的通道。通过写入配置寄存器, 这些通道中都可以连接到可用的外设触发源, 此外, 每个通道都可以由软件触发。在启用通道之前, 软件必须注意将此配置写入内存。触发通道后, DMA 控制器将首先从系统内存中读取通道描述符, 然后它将继续执行描述符指定的存储器传输。描述符包含要读取的存储器地址, 要写入的存储器地址, 要传输的字节数等。

除了基本的传输模式外，DMA 控制器还支持两种高级传输模式。乒乓操作和分散/聚集操作。乒乓传输适用于高速外围通信的流数据传输，因为在处理器内核仍在处理前一个字节时，DMA 可以立即检索下一个传入的数据字节。分散/聚集操作可以从内存执行一系列任务，并允许通过软件实现复杂的方案。

通过对通道使用不同的优先级级别并设置 DMA 控制器重新仲裁的字节数，可以确保传输严格的按时间进行。

16.2.1. 通道选择

每个 DMA 通道都有一个请求源配置寄存器。SOURCESEL0~SOURCESEL3 分别为 4 个通道选择监听外设的 DMA 请求。

注：当所选外设的时钟关闭时，DMA 通道不应处于活动状态。

16.2.2. DMA 控制

16.2.2.1. 仲裁率

配置控制器在 DMA 传输期间进行仲裁的时间，能够减少为更高优先级的频道提供服务的等待时间。控制器提供 4 bit 的 R_{power} 位用于配置重新仲裁之前发生的 AHB 总线传输次数。比如，输入的值 R 为 4，那么仲裁率则为 2 的 4 次幂，那么，微控制器每 16 个 DMA 传输进行一次仲裁。

R _{power}	DMA 传输后进行仲裁
B0000	X=1
B0001	X=2
B0010	X=4
B0011	X=8
B0100	X=16
B0101	X=32
B0110	X=64
B0111	X=128
B1000	X=256
B1001	X=512

B1010-b1111	X=1024
-------------	--------

AHB 总线中转仲裁间隔表

注：必须注意不要分配具有过大 R_power 的低优先级通道，因为这会阻止控制器在重新仲裁之前为高优先级请求提供服务。

用户指定的 DMA 传输次数 N 。当 $N > 2R$ 且不是 $2R$ 的整数倍时，控制器将始终执行 $2R$ 传输序列，直到剩余 $N < 2R$ 待传输。控制器在 DMA 周期结束时执行剩余的 N 次传输。 R_power 位的值存储在通道控制数据结构中。有关数据结构中 R_power 位的位置的更多信息。

16.2.2.2. 优先级

控制器进行仲裁时，它将使用以下信息确定要服务的下一个频道：

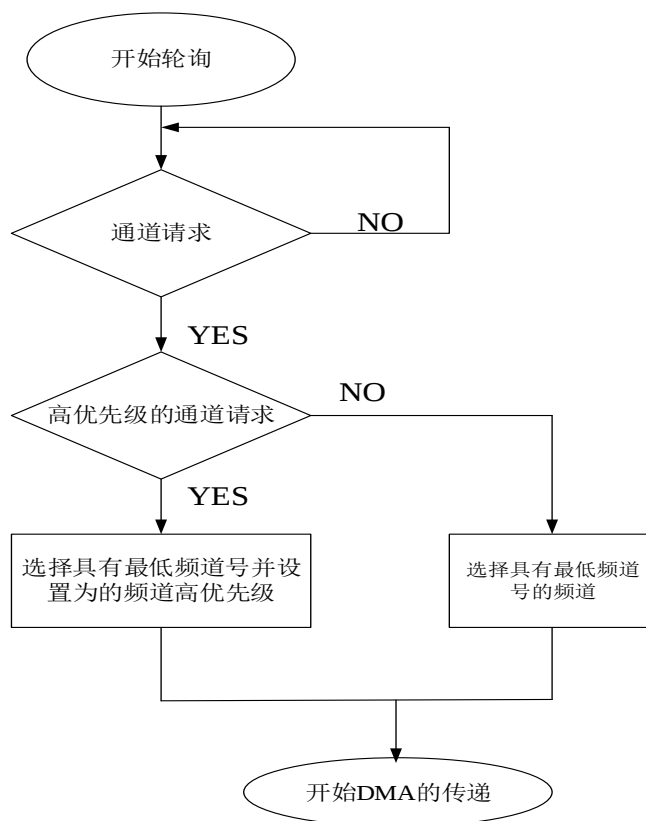
- 通道号
- 分配给通道的优先级，默认或高。

通过配置寄存器将每个通道配置为使用默认优先级或高优先级。0 通道具有最高优先级，并且随着通道号的增加，通道的优先级降低。以优先级降序列出 DMA 通道优先级。

通道顺序	优先级的设置	通道优先级降序
0	High	DMA 最高优先级通道
1	High	—
2	High	—
3	High	—
0	Default	—
1	Default	—
2	Default	—
3	Default	DMA 最低优先级通道

DMA 通道优先级表

DMA 传输完成后，控制器将轮询所有可用的 DMA 通道。显示了用于确定接下来要执行的 DMA 传输的过程。



轮询流程示意图

16.2.3. 周期类型

通过设置 Cycle_ctrl 位控制控制器执行 DMA 模式。

控制周期	描述
000	通道控制数据结构无效
001	基本 DMA 传输
010	自动请求
011	乒乓操作
100	使用主要数据结构进行内存分散/聚集
101	使用备用数据结构进行内存分散/聚集
110	使用主要数据结构的外围分散/聚集
111	使用备用数据结构进行外围分散/聚集

注：cycle_ctrl 位于 channel_cfg 存储器位置

对于所有周期类型，控制器在 2R 次 DMA 传输后进行仲裁。如果将 2R 值设置为低优先级通道，则它将阻止所有其他通道执行 DMA 传输，直到低优先级 DMA 传输完成为止。因

此，设置 R_power 时需要注意，不要显著增加高优先级通道的延迟。

16.2.3.1. 周期无效

控制器完成 DMA 周期后，会将周期类型设置为无效，以防止其重复相同的 DMA 周期。

16.2.3.2. 基本操作

在这种模式下，将控制器配置为使用主要或备用数据结构。在启用通道 C 且控制器收到对此通道的请求后，此 DMA 周期的流程如下：

1. 控制器执行 2R 传输。如果剩余的传输次数变为零，则流程继续步骤 3；
2. 控制器仲裁：
 - 1) 如果更高优先级的通道正在请求服务，则控制器为该通道服务；
 - 2) 如果外围设备或软件向控制器发出请求信号，则它会继续执行步骤 1。
3. 控制器在一个 HCLK 周期内将 dma_done 设置为高电平。这意味着 DMA 周期已完成。

16.2.3.3. 自动请求

当控制器以这种模式运行时，需要接收单个请求即可使其完成整个 DMA 周期。这使得能够进行大量的数据传输，而不会显著增加为服务更高优先级的请求提供服务的等待时间，或者不需要来自处理器或外围设备的多个请求。将控制器配置为使用主要或备用数据结构。在启用通道 C 且控制器收到对此通道的请求后，此 DMA 周期的流程如下：

1. 控制器对通道 C 执行 2R 传输。如果剩余传输次数为零，则流程继续执行步骤 3；
2. 控制器进行仲裁。当通道 C 具有最高优先级时，则 DMA 周期在步骤 1 继续；
3. 控制器在一个 HCLK 周期内将 dma_done 设置为高电平，DMA 周期完成。

16.2.3.4. 乒乓操作

在乒乓模式下，控制器使用一种数据结构（主数据结构或备用数据结构）执行 DMA 周期，然后使用另一种数据结构执行 DMA 周期。控制器继续从主节点切换到备用节点，直到读取无效的数据结构或主机处理器禁用通道为止。

16.2.3.5. 内存分散聚集

在存储器分散收集模式下，控制器接收初始请求，然后使用主要数据结构执行四个 DMA 传输。传输完成后，它将使用备用数据结构启动 DMA 周期。在此循环完成之后，控制器将使用主要数据结构执行另外四个 DMA 传输。控制器继续从主要切换到备用，直到其中一个：

1. 主处理器为基本周期配置备用数据结构;
2. 读取无效的数据结构;

注: 控制器完成 N 次主要传输后, 通过将 `cycle_ctrl` 字段设置为 0 来使主要数据结构无效。

当分散收集事务使用自动请求周期完成时, 控制器仅声明 `dma_done [C]`。

在分散收集模式下, 控制器使用主要数据结构对备用数据结构进行编程。

16.2.4. 错误信息

如果控制器在 AHB 主界面上检测到错误响应, 则:

1. 禁用与错误对应的通道;
2. 将 `dma_err` 信号拉高;

16.2.5. 通道控制数据结构配置

提供一个系统存储器区域来包含通道控制数据结构。该系统内存必须:

1. 提供控制器和主机处理器可以访问的系统内存的连续区域;
2. 基址是通道控制数据结构总大小的整数倍。

数据控制配置寄存器如下:

Bit	Name	Description	Attribute	Reset
31:30	DST_INC	目的地址自增设置 (依据源地址位宽) 源数据位宽为 byte: 00: byte; 01: halfword; 10: word; 11: 不自增; 源数据位宽为 halfword: 00: 无效; 01: halfword; 10: word; 11: 不自增; 源数据位宽为 word: 00: 无效; 01: 无效; 10: word; 11: 不自增;	R/W	0
29:28	DST_SIZE	目的数据大小, 需和源数据位宽一致;	R/W	0
27:26	SRC_INC	源地址自增设置 (依据源地址位宽) 源数据位宽为 byte: 00: byte; 01: halfword; 10: word;	R/W	0

		11: 不自增; 源数据位宽为 halfword: 00: 无效; 01: halfword; 10: word; 11: 不自增; 源数据位宽为 word: 00: 无效; 01: 无效; 10: word; 11: 不自增;		
25:24	SRC_SIZE	源数据位宽设置: 00: byte; 01: halfword; 10: word; 11: 不自增;	R/W	0
23:22	Reserved		R	0
21	DST_PROT_CTRL	写目的数据特权使能位 0: 禁止; 1: 使能;	R/W	0
20:19	Reserved			
18	SRC_PROT_CTRL	读源数据特权使能位 0: 禁止; 1: 使能;	R/W	0
17:14	R_power	仲裁率设置位, 即 DMA 传输 N 次后再次仲裁 0000: N=1; 0001: N=2; 0010: N=4; 0011: N=8; 0100: N=16; 0101: N=32; 0110: N=64; 0111: N=128; 1000: N=256; 1001: N=512; 1010~1111: N=1024;	R/W	0
13:4	N_minus_1	DMA 传输次数控制位: 10 位值表示 DMA 的传输次数减 1 0000000000: 1 次 DMA 传输; 0000000001: 2 次 DMA 传输; 0000000010: 3 次 DMA 传输; 0000000011: 4 次 DMA 传输; 0000000100: 5 次 DMA 传输; 1111111111: 1024 次 DMA 传输;	R/W	0

3	Next_useburst	控制当控制器执行外部 scatter/gather 并完成使用备用数据结构的 DMA 周期时， chnl_useburst_set [C]位的设置。 0: 设置为 0; 1: 设置为 1;	R/W	0
2:0	Cycle_ctrl	DMA 的模式设置: 000: 无效模式, 使指针数据结构无效; 001: 基本模式, DMA 控制器在进入仲裁前, 必须接收一个新的请求; 010: 自动请求模式, DMA 控制器在仲裁过程中会自动插入对相应通道的请求, 这意味着初始请求足以使 DMA 周期完成; 011: 乒乓模式, DMA 控制器使用两种数据结构之一执行 DMA 传输, 完成后将使用另外的数据结构继续 DMA 传输。再次完成后, 当处理器已更新主数据结构, 将继续使用主数据结构执行 DMA 周期, 直到读取无效的数据结构或者处理器将 cycle_ctrl 位更改为基本模式或自动请求模式等; 100: 内存分散-聚集模式, 仅使用主数据结构; 101: 内存分散-聚集模式, 仅使用备用数据结构; 110: 内存分散-聚集模式, 仅使用主数据结构; 111: 内存分散-聚集模式, 仅使用备用数据结构;	R/W	0

16.3 DMA 寄存器

基地址: 0x4000 6000

16.3.1. DMA_STATUS 状态寄存器

偏移地址: 0x000

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20:16	CHNUM	通道数递减;	R/W	0
15:8	Reserved		R	0
7:4	STATE[3:0]	控制当前状态 0000: 空状态; 0001: 读取通道控制器数据; 0010: 读取数据结束指针; 0011: 读取目标数据结束指针; 0100: 读取源数据;	R/W	0

		0101: 写入目标数据; 0110: 等待 DMA 请求清除; 0111: 写入通道控制器数据; 1000: 停止; 1001: 完成; 1010: 分散-聚集传递;		
3:1	Reserved		R	0
0	EN	DMA 的使能 1: 使能; 0: 禁止;	R/W	0

16.3.2. DMA_CONFIG 配置寄存器

偏移地址: 0x004

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	CHRPOT	通道保护控制, 对 DMA 控制器进行访问特权 1: 有访问特权; 0: 没有访问特权;	W	0
4:1	Reserved		R	0
0	EN	启用 DMA 控制器 1: DMA 使能; 0: DMA 禁止;	W	0

16.3.3. DMA_CTRLBASE 通道控制数据库指针寄存器

偏移地址: 0x008

Bit	Name	Description	Attribute	Reset
31:0	CTRLBASE	通道控制数据库指针;	R/W	0x0000_0000

16.3.4. DMA_ALTCTRLBASE 通道备用控制数据库指针寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:0	ALTCTRLBASE	通道备用控制数据库指针;	R/W	0x0000_0040

16.3.5. DMA_CHWAITSTATUS 通道等待请求状态寄存器

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3WAITSTATUS	通道 3 等待请求状态;	R	1
2	CH2WAITSTATUS	通道 2 等待请求状态;	R	1
1	CH1WAITSTATUS	通道 1 等待请求状态;	R	1
0	CH0WAITSTATUS	通道 0 等待请求状态;	R	1

16.3.6. DMA_CHSWREQ 通道软件请求寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3SWREQ	通道 3 软件请求;	W	0
2	CH2SWREQ	通道 2 软件请求;	W	0
1	CH1SWREQ	通道 1 软件请求;	W	0
0	CH0SWREQ	通道 0 软件请求;	W	0

16.3.7. DMA_CHUSEBURSTS 设置通道突发寄存器

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3USEBURSTS	通道 3 突发设置;	R/W	0
2	CH2USEBURSTS	通道 2 突发设置;	R/W	0
1	CH1USEBURSTS	通道 1 突发设置;	R/W	0
0	CH0USEBURSTS	通道 0 突发设置;	R/W	0

16.3.8. DMA_CHUSEBURSTC 清除通道突发寄存器

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3USEBURSTC	通道 3 突发清除;	W	0
2	CH2USEBURSTC	通道 2 突发清除;	W	0

1	CH1USEBURSTC	通道 1 突发清除;	W	0
0	CH0USEBURSTC	通道 0 突发清除;	W	0

16.3.9. DMA_CHREQMASKS 设置通道掩码请求寄存器

偏移地址: 0x020

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3REQMASKS	设置通道 3 掩码请求;	R/W	0
2	CH2REQMASKS	设置通道 2 掩码请求;	R/W	0
1	CH1REQMASKS	设置通道 1 掩码请求;	R/W	0
0	CH0REQMASKS	设置通道 0 掩码请求;	R/W	0

16.3.10. DMA_CHREQMASKC 清除通道掩码请求寄存器

偏移地址: 0x024

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3REQMASKC	清除通道 3 掩码请求;	W	0
2	CH2REQMASKC	清除通道 2 掩码请求;	W	0
1	CH1REQMASKC	清除通道 1 掩码请求;	W	0
0	CH0REQMASKC	清除通道 0 掩码请求;	W	0

16.3.11. DMA_CHENS 设置通道使能寄存器

偏移地址: 0x028

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3ENS	设置通道 3 使能;	R/W	0
2	CH2ENS	设置通道 2 使能;	R/W	0
1	CH1ENS	设置通道 1 使能;	R/W	0
0	CH0ENS	设置通道 0 使能;	R/W	0

16.3.12. DMA_CHENC 清除通道使能寄存器

偏移地址：0x02C

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3ENC	清除通道 3 使能;	W	0
2	CH2ENC	清除通道 2 使能;	W	0
1	CH1ENC	清除通道 1 使能;	W	0
0	CH0ENC	清除通道 0 使能;	W	0

16.3.13. DMA_CHALTS 设置通道备用寄存器

偏移地址：0x030

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3ALTS	设置通道 3 备用结构;	R/W	0
2	CH2ALTS	设置通道 2 备用结构;	R/W	0
1	CH1ALTS	设置通道 1 备用结构;	R/W	0
0	CH0ALTS	设置通道 0 备用结构;	R/W	0

16.3.14. DMA_CHALTC 清除通道备用寄存器

偏移地址：0x034

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3ALTC	清除通道 3 备用结构;	W	0
2	CH2ALTC	清除通道 2 备用结构;	W	0
1	CH1ALTC	清除通道 1 备用结构;	W	0
0	CH0ALTC	清除通道 0 备用结构;	W	0

16.3.15. DMA_CHPRIS 设置通道优先级寄存器

偏移地址：0x038

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:4	Reserved		R	0
3	CH3PRIS	设置通道 3 优先级;	R/W	0
2	CH2PRIS	设置通道 2 优先级;	R/W	0
1	CH1PRIS	设置通道 1 优先级;	R/W	0
0	CH0PRIS	设置通道 0 优先级;	R/W	0

16. 3. 16. DMA_CHPRIC 清除通道优先级寄存器

偏移地址: 0x03C

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3PRIC	清除通道 3 优先级;	W	0
2	CH2PRIC	清除通道 2 优先级;	W	0
1	CH1PRIC	清除通道 1 优先级;	W	0
0	CH0PRIC	清除通道 0 优先级;	W	0

16. 3. 17. DMA_ERRORC 清除总线错误寄存器

偏移地址: 0x040

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	ERRORC	清除总线;	R/W	0

16. 3. 18. DMA_CHREQSTATUS 通道请求状态寄存器

偏移地址: 0xE10

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3REQSTATUS	通道 3 请求状态;	R	0
2	CH2REQSTATUS	通道 2 请求状态;	R	0
1	CH1REQSTATUS	通道 1 请求状态;	R	0
0	CH0REQSTATUS	通道 0 请求状态;	R	0

16. 3. 19. DMA_CHSREQSTATUS 单通道请求状态寄存器

偏移地址：0xE18

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CH3SREQSTATUS	单通道 3 请求状态；	R	0
2	CH2SREQSTATUS	单通道 2 请求状态；	R	0
1	CH1SREQSTATUS	单通道 1 请求状态；	R	0
0	CH0SREQSTATUS	单通道 0 请求状态；	R	0

16. 3. 20. DMA_IF 中断标志寄存器

偏移地址：0x1000

Bit	Name	Description	Attribute	Reset
31	ERR	DMA 的错误中断标志；	W	0
30:4	Reserved		R	0
3	CH3DONE	通道 3 中断标志；	W	0
2	CH2DONE	通道 2 中断标志；	W	0
1	CH1DONE	通道 1 中断标志；	W	0
0	CH0DONE	通道 0 中断标志；	W	0

16. 3. 21. DMA_IFS 中断标志配置寄存器

偏移地址：0x1004

Bit	Name	Description	Attribute	Reset
31	ERR	DMA 的错误中断标志设置；	W	0
30:4	Reserved		R	0
3	CH3DONE	通道 3 中断标志设置；	W	0
2	CH2DONE	通道 2 中断标志设置；	W	0
1	CH1DONE	通道 1 中断标志设置；	W	0
0	CH0DONE	通道 0 中断标志设置；	W	0

16. 3. 22. DMA_IFC 清除中断标志寄存器

偏移地址：0x1008

Bit	Name	Description	Attribute	Reset
31	ERR	DMA 的清除错误中断标志；	W	0
30:4	Reserved		R	0
3	CH3DONE	通道 3 清除中断标志；	W	0
2	CH2DONE	通道 2 清除中断标志；	W	0
1	CH1DONE	通道 1 清除中断标志；	W	0
0	CH0DONE	通道 0 清除中断标志；	W	0

16. 3. 23. DMA_IEN 中断使能寄存器

偏移地址：0x100C

Bit	Name	Description	Attribute	Reset
31	ERR	DMA 的错误中断使能；	R/W	0
30:4	Reserved		R	0
3	CH3DONE	通道 3 中断使能；	R/W	0
2	CH2DONE	通道 2 中断使能；	R/W	0
1	CH1DONE	通道 1 中断使能；	R/W	0
0	CH0DONE	通道 0 中断使能；	R/W	0

16. 3. 24. DMA_CH0_CTRL 通道控制寄存器

偏移地址：0x1100

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5:0	SOURCESEL0	通道 0 选择 DMA 触发的源；	R/W	0

16. 3. 25. DMA_CH1_CTRL 通道控制寄存器

偏移地址：0x1104

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0

5:0	SOURCESEL1	通道 1 选择 DMA 触发的源;	R/W	0
-----	------------	-------------------	-----	---

16. 3. 26. DMA_CH2_CTRL 通道控制寄存器

偏移地址: 0x1108

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5:0	SOURCESEL2	通道 2 选择 DMA 触发的源;	R/W	0

16. 3. 27. DMA_CH3_CTRL 通道控制寄存器

偏移地址: 0x110C

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5:0	SOURCESEL3	通道 3 选择 DMA 触发的源;	R/W	0

16. 3. 28. SOURCESEL 触发源选择寄存器

Bit	Name	Description	Attribute	Reset
000_000	0		000_000	0
000_001	CCP5IIF	CCP5 DMA 请求;	000_001	CCP5IIF
000_010	CCP4IIF	CCP4 DMA 请求;	000_010	CCP4IIF
000_011	CCP3IIF	CCP3 DMA 请求;	000_011	CCP3IIF
000_100	CCP2IIF	CCP2 DMA 请求;	000_100	CCP2IIF
000_101	CCP1IIF	CCP1 DMA 请求;	000_101	CCP1IIF
000_110	CCP0IIF	CCP0 DMA 请求;	000_110	CCP0IIF
000_111	CLU3IF	CLU3 DMA 请求;	000_111	CLU3IF
001_000	CLU2IF	CLU2 DMA 请求;	001_000	CLU2IF
001_001	CLU1IF	CLU1 DMA 请求;	001_001	CLU1IF
001_010	CLU0IF	CLU0 DMA 请求;	001_010	CLU0IF
001_011	SPIIF	SPI DMA 请求;	001_011	SPIIF
001_100	T3IF	T3 DMA 请求;	001_100	T3IF
001_101	T2IF	T2 DMA 请求;	001_101	T2IF
001_110	T1IF	T1 DMA 请求;	001_110	T1IF

001_111	TOIF	TO DMA 请求;	001_111	TOIF
010_000	ADCIF	ADC DMA 请求;	010_000	ADCIF
010_001	DAC0IF	DAC0 DMA 请求;	010_001	DAC0IF
010_010	DAC1IF	DAC1 DMA 请求;	010_010	DAC1IF
010_011	I2C_TX_IF	I2C_TX DMA 请求;	010_011	I2C_TX_IF
010_100	I2C_RX_IF	I2C_RX DMA 请求;	010_100	I2C_RX_IF
010_101	UART_TX_IF	UART_TX DMA 请求;	010_101	UART_TX_IF
010_110	UART_RX_IF	UART_RX DMA 请求;	010_110	UART_RX_IF
010_111	OP0IF	OP1 DMA 请求;	010_111	OP0IF
100_000	OP1IF	OP0 DMA 请求;	100_000	OP1IF

17 HDIV 硬件除法器

17.1 HDIV 概述

硬件除法器（HDIV）对高效能应用很有帮助。硬件除法器为一有符号整数除法器同时含商数与余数输出。

17.1.1. HDIV 特性

HDIV 主要特性：

- 无符号整数的计算与有符号（2 的补码）整数的计算；
- 32 位被除数含 32 位除法器计算容量；
- 32 位商数与 32 位余数输出；
- 除零警示标志；
- 9 HCLK 完成除法计算。

17.1.2. HDIV 功能描述

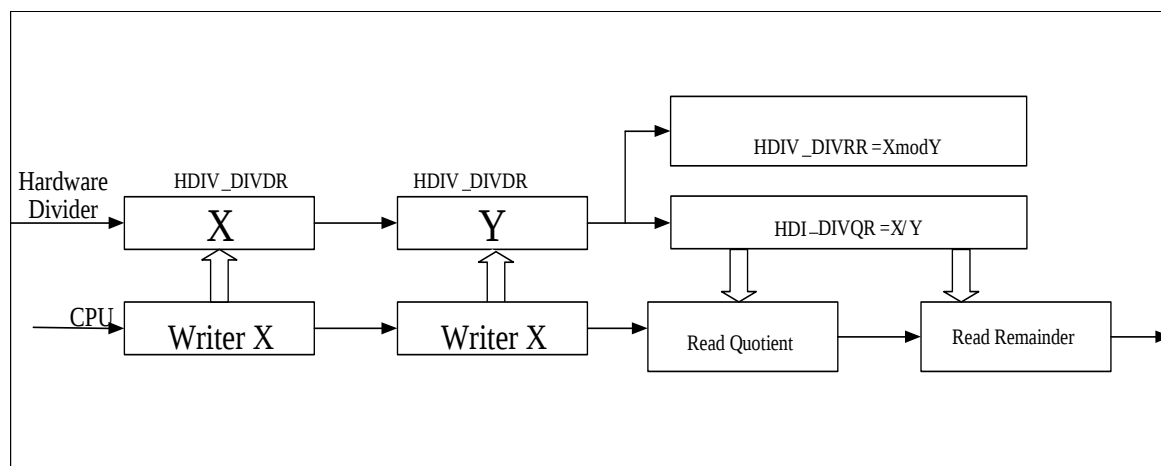
要使用硬件除法器，首先须设定被除数、除数和符号设定，接着输入 START 信号后自动触发计算。通过读 HDIV_DIVQR 与 HDIV_DIVRR 寄存器可得到计算结果，包含商数与余数。

若除数为 0，HDIV_DIVSTAT 之 DIV0 标志为 1。

被除数为 32 位有符号整数且除数为 32 位有符号整数。商数为 32 位有符号整数而余数为 32 位有符号整数。

<硬件除法器运作流程示意图>标示硬件除法器之运算流程。为了计算 X / Y ，CPU 必须写 X 至 HDIV_DIVDR 寄存器，接着写 Y 至 HDIV_DIVSR。CPU 可读 HDIV_DIVQR 与 HDIV_DIVRR 寄存器以取得 HDIV_DIVSR 被写之后的计算结果。

17.1.3. HDIV 运行流程示意图



硬件除法器运作流程示意图

17.2 HDIV 寄存器

基地址：0x400A 0000

17.2.1. HDIV_DIVDR 被除数寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:0	DIVD	计算之前的被除数；	R/W	0

17.2.2. HDIV_DIVDSR 除数寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:0	DIVD	计算之前的除数；	R/W	0

17.2.3. HDIV_DIVQR 商寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:0	DIVD	计算完成之后保持除法的商结果；	R/W	0

17.2.4. HDIV_DIVRR 余数寄存器

偏移地址：0x00C

Bit	Name	Description	Attribute	Reset
31:0	DIVD	计算完成之后保存除法器的余数结果；	R/W	0

17.2.5. HDIV_DIVSTAT 除法器状态寄存器

偏移地址：0x010

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	START	开始计算使能位 1: 使能； 0: 禁止；	R/W	0
2	SIGN	符号的选择位 1: 有符号； 0: 无符号；	R/W	0
1	DIV0	除数零警告位 1: 除数是 0； 0: 除数不是 0； 注意：每当写入除数寄存器时更新；	R/W	0
0	BUSY	除法器状态位 1: 除法器运行； 0: 除法器空闲；	R	0

18 FLASH 可编程存储器（FMC）

18.1 FLASH 概述

AD3005 MCU 集成了在线编程 FLASH 存储器的特性，以方便存储升级的代码。可以由 AD3005 MCU 编程接口或者应用代码灵活地对 FLASH 存储器进行编程控制。AD3005 MCU 为用户提供了安全选项以防止未经授权的不安全信息存储到 FLASH 存储器中。

- 在编程或者擦除 FLASH 过程中，MCU 停止工作，但外设（定时器、WDT、I/O、PWM 等）仍在正常工作。
- 在编写 FLASH 或者擦除 FLASH 之前，须禁止看门狗。
- 擦除动作会将 FLASH 页中的所有位都置为逻辑 1。
- 硬件会控制时钟，编程完成后，硬件释放时钟，MCU 会执行下一条指令。
- 当 BOOTROM 配置字使能后，FLASH 的最后 8K 空间将不能读写，作为 BOOTROM 使用，此时 FLASH 空间大小为 56K。

18.1.1. 嵌入式 FLASH 存储器

FLASH 存储器为 32 位宽度的存储器，可用于存储代码和数据常量，位于芯片内存的特定地址。

内置的高性能 FLASH 存储器模块有以下主要特性：存储器构架，FLASH 存储器包括用户 ROM 和引导 ROM。

用户 ROM	16K × 32 位，分为 64 页，每页 1024 字节
--------	-------------------------------

基于 AHB 协议，FLASH 接口实现指令和数据的访问，实现了对 FLASH 存储器的操作（编程/擦除），编程/擦除的操作可在产品的工作电压范围内执行。

18.1.2. FLASH 特性

可编程存储器 FLASH 具有以下特性：

1. 读取接口（32 位）；
2. FLASH 编程/擦除动作；
3. 编译选项（Code Option）包括代码加密选项（CRYPTEN），需要通过配置字更新下载。

嵌入式 FLASH 存储控制器（FMC）管理主存储块和编译选项的执行。编程/擦除操作需

要的高压由内部产生。主要的 FLASH 存储器可以通过设定不同的代码加密级别（CS）进行读/写保护。

在对 FLASH 存储器进行写选择时，任何试图对 FLASH 存储器进行读取的操作都会中断总线，写操作完成后，才会开始正确的读操作。这就意味着在进行写/擦除操作时，不能安排读取代码或者数据。在 FLASH 存储器进行写和擦除操作时，如果 TBIT 关闭，IHRC 应该由 FMC 设为 ON 状态，否则 TBIT 打开，IHRC 可以关闭。可以通过 ICP 和 ISP 对 FLASH 存储器进行编程和擦除操作。

18.2 FLASH 功能描述

18.2.1. 扇区

Block	Page	Logic Address	Write Address	Size/Byte
User ROM	0	0x08000000~0x080003FF	0x08000000~0x080003FF	1024
	1	0x08000400~0x080007FF	0x08000400~0x080007FF	1024
	·			·
	·			·
	62	0x0800F800~0x0000FBFF	0x0800F800~0x0800FBFF	1024
	63	0x0800FC00~0x0000FFFF	0x0800FC00~0x0800FFFF	1024

18.2.2. 读操作

作为一个通用的存储空间，嵌入式 FLASH 模块可以直接进行访问。读操作是访问 FLASH 模块内容并提供需要的数据。读取界面由一个读取控制器（一面访问 FLASH 存储器），和一个 AHB 接口（另一面与 CPU 联系）组成。读取界面的主要任务是产生控制信号，从 FLASH 存储器中读取 CPU 需要的内容。

18.2.3. 编程/擦除

FLASH 存储器的擦除操作按页执行。为了保证不产生过度的编程，IHRC 提供 FLASH 编程和擦除时钟。

18.2.4. 嵌入式引导加载程序

嵌入式引导加载程序位于引导 ROM 区域，已经固化。

18.2.5. FLASH 存储控制器 (FMC)

FMC 控制 FLASH 存储器的编程和擦除。

18.2.6. 编程 FLASH 存储器

FLASH 存储器可以一次编程 4 字节(2us), 写入数据存储, 通过执行标准的字写入操作, CPU 可以编程主要的 FLASH 存储器。FLASH_CTRL 寄存器的 PG 位必须设置为 1, FMC 读取指定的 FLASH 存储器地址的数据, 检查是否已经擦除, 若没有擦除, 跳过编程操作, 通过 FLASH_STATUS 寄存器的 PGERR 位发出一个警告。

开始擦除/编程, 查找超过页边界的地址。

开始擦除/编程, 查找不合规则的地址 (>ROM size)。

填充数据到超过页边界的地址。

标准模式下的主要 FLASH 存储器编程流程如下:

1. 选择写入 4 个字节(FLASH_DATA[31: 0])还是写入 2 个字节 (FLASH_DATA[15: 0])
2. 设置 FLASH_CTRL 寄存器的 PG 位为 1;
3. 填充目标地址到 FLASH_ADDR 寄存器;
4. 写入数据填入 FLASH_DATA 寄存器;
5. 设置 START 位开始进行编程;
6. 等待 BUSY 位复位;
7. 读取编程的值或者 CKSUM, 并进行校验。(可选)

18.2.7. 擦除操作

FLASH 存储器可以按页擦除, 也可全部擦除 (批量擦除)。当加密使能, Cache CFM 区域禁止擦除和写入。

可以利用 FMC 的页擦除特性将 FLASH 存储器的整页进行擦除。要擦除整页, 其流程如下:

1. 设置 FLASH_CTRL 寄存器的 PER 位为 1;
2. 对 FLASH_ADDR 寄存器进行编程来选择一页进行擦除;
3. 设置 FLASH_CTRL 寄存器的 START 位为 1;
4. 等待 BUSY 位复位;

5. 读取擦除页的内容或者 CKSUM，并进行校验。（可选）

18.2.8. 读保护

在编译选项中设置代码安全字节可以激活读保护选项。

当 FLASH 存储器读保护选项由受保护变为不受保护时，可以由硬件在重新烧录读保护选项前将用户 ROM 进行批量擦除。

18.2.9. 硬件 CHECKSUM

硬件 CHECKSUM 采用 CRC16-XMODEM 算法，对 ROM 区以页为单位进行校验。若使能读保护，用户可以通过烧录器或者 ISP AP 读出硬件的 CHECKSUM 值。

18.3 FLASH 寄存器

基地址：0x4008 6000

18.3.1. CFG 配置寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:16	FMCKEY	FMC 验证位 读取为 0，需要写入数据到该寄存器时，必须写入 0x5AFA 到 FMCKEY，否则会忽略对该寄存器的写动作；	W	0
15:8	Reserved		R	0
7:4	Reserved		R	0
3	Reserved		R	0
2:0	WAIT[2:0]	FLASH 读取等待时间 000b: HCLK ≤ 20MHz; 001b: 20MHz < HCLK ≤ 40MHz; 010b: 40MHz < HCLK ≤ 60MHz; 011b: 60MHz < HCLK ≤ 80MHz; 100b: 80MHz < HCLK ≤ 100MHz; 101b: 100MHz < HCLK ≤ 120MHz; 110b: 120MHz < HCLK ≤ 140MHz; 111b: 140MHz < HCLK ≤ 160MHz;	R/W	0

18.3.2. STATUS 状态寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	ICACHE_ERR	当加密使能时，FLASH 编程地址为 ICACHE CFM 映射地址范围内，则产生错误标志 0：读 0 表示无错误，写 0 将该位清零； 1：有错误；	R/W	0
2	ERR	编程错误标志位 0：读 0 表示无错误，写 0 将该位清零； 1：下列情形是由硬件进行设置 —开始擦除/编程，查找超过页边界的地址； —开始擦除/编程，查找不合规则的地址（>ROM size）； —填充数据到超过页边界的地址；	R/W	0
1	Reserved		R	0
0	BUSY	繁忙指示标志位 0：FLASH 操作不繁忙； 1：正在处理 FLASH 操作，在 FLASH 操作开始时设置该位（同时 EOP 位清零），操作完成后或者硬件出错时复位；	R/W	0

18.3.3. CTRL 控制寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	CHK	CRC16 CHKSUM 计算选择。自动计算 FAR 指定页的 1024 字节的 FLASH 校验和。不能与 PER 或者 PG 同时置为 1。该位只能由软件置 1，BUSY 位复位时，该位也复位。 1：启动； 0：停止；	R/W	0
6	START	开始页擦除或者页编程，该位只能由软件置 1，BUSY 位复位时，该位也复位。 1：启动； 0：停止；	R/W	0
5:2	Reserved		R	0
1	PER	页擦除操作选择位 该位只能由软件置 1，BUSY 位复位时，该位也复位。	R/W	0

		1: 启动; 0: 停止;		
0	PG	FLASH 编程操作选择位。 该位只能由软件置 1, BUSY 位复位时, 该位也复位。 1: 启动; 0: 停止;	R/W	0

18.3.4. DATA 数据寄存器

偏移地址: 0x00C

Bit	Name	Description	Attribute	Reset
31:0	DATA[31:0]	需要编程的数据;	R/W	0

18.3.5. ADDR 地址寄存器

软件升级需要擦除或者编程的 FLASH 地址, 填满 FLASH 地址之前, PG 位或者 PER 位需设置为 1。

注: 当 FLASH_STATUS 寄存器的 BUSY 位置 1 时, 会阻止对该寄存器的写访问。

偏移地址: 0x010

Bit	Name	Description	Attribute	Reset
31:0	FAR[31:0]	FLASH 地址 当加密使能时, 不支持对 ICACHE 定义 CFM 区域做擦除和写入动作; 选择按页擦除时, 则选择一个页面进行擦除; 或者选择按页编程时, 则选择一个页面进行编程。	R/W	0

18.3.6. CHKSUM 校验寄存器

偏移地址: 0x014

Bit	Name	Description	Attribute	Reset
31:16	CHKSUM_BUFF[15:0]	BUFFER 写入 1024 字节读 Checksum 值, 采用 $CRC16 - XMODEM (x^{16} + x^{12} + x^5 + 1)$	R	0
15:0	CHKSUM_FLASH[15:0]	FLASH 写入 1024 字节, 读 Checksum 值, 采用 $CRC16 - XMODEM (x^{16} + x^{12} + x^5 + 1)$	R	0

18.3.7. CRC0 寄存器

偏移地址: 0x018

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:10	Reserved		R	0
9	CRC_INIT	CRC16 初始化成 0xFFFF;	R/W	0
8	CRC_CALC	CRC16 计算一个字节;	R/W	0
7:0	CRC_IN[7:0]	输入的准备校验的 CRC 值;	R/W	0

18.3.8. CRC1 寄存器

偏移地址: 0x01C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CRC_OUT[15:0]	用CRC16 – XMODEM ($x^{16} + x^{12} + x^5 + 1$) 计算的 CKSUM 值。可以写入, 作为 CRC 的初始值;	R/W	0

19 ICACHE 指令高速缓冲存储器（CMC）

19.1 ICACHE 概述

AD3005 内置 1K 字节的指令 Cache，四路组相联结构，采用 LFU 替换算法实现。

19.2 ICACHE 寄存器定义

基地址：0x4008 8000

19.2.1. CFG 配置控制寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	CFLUSH	Cache 内容清空使能位 1: 清空; 0: 未清空;	R/W	0

20 CONFIG FUSE 加密功能

20.1 CONFIG FUSE 概述

AD3005 CONFIG FUSE 通过配置字进行定义，CPU 不能读取或者写入 FUSE 寄存器，只能通过 BOOTLOADER 写入。

20.2 CONFIG FUSE 寄存器定义

20.2.1. FUSE0 配置寄存器

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:7	Reserved		R	0
6	SWDDIS	SWD 引脚禁止位。通过配置字使能打开，寄存器只能读，不能写。 0: 使能 SWD 引脚 (PD5 作为 SWDIO 引脚，PD6 作为 SWCLK 引脚)； 1: 禁止 (PD5 和 PD6 作为 GPIO 引脚)；	R	1
5	CRYPTEN	使能 CFM 定义区域的程序加密，通过 BOOTLOADER 设定控制字。如果使能加密，则自动将 SWD 功能禁止。 1: 开启； 0: 关闭；	R	1
4	CACHE_ON	Cache 开启，系统默认打开，如要设定关闭，需要通过配置字更新，并擦除整个 FLASH。 1: 开启 (默认开启，用户无法配置)；	R	1
3:0	CFM[3:0]	Cache FLASH 加密映射地址范围，系统默认加密范围为 0~64K；如需要修改，需要通过配置字更新，并擦除整个 FLASH。当加密使能时，该区域内不支持 CPU 的 FLASH 擦除和写入处理。 0000b: 0000-0FFFH (0~4K)； 0001b: 0000-1FFFH (0~8K)； 0010b: 0000-2FFFH (0~12K)； 0011b: 0000-3FFFH (0~16K)； 0100b: 0000-4FFFH (0~20K)； 0101b: 0000-5FFFH (0~24K)； 0110b: 0000-6FFFH (0~28K)； 0111b: 0000-7FFFH (0~32K)； 1000b: 0000-0FFFH (0~36K)； 1001b: 0000-1FFFH (0~40K)；	R	0111

		1010b: 0000-2FFFFH (0~44K); 1011b: 0000-3FFFFH (0~48K); 1100b: 0000-4FFFFH (0~52K); 1101b: 0000-5FFFFH (0~56K); 1110b: 0000-6FFFFH (0~60K); 1111b: 0000-7FFFFH (0~64K);		
--	--	--	--	--

21 SWD 调试

21.1 SWD 概述

SWD 功能集成到 ARM Cortex-M0 中，配置好的 ARM Cortex-M0 可支持 4 个断点和 2 个监控点。

21.1.1. SWD 特性

SWD 具有以下特性：

1. 支持 ARM Serial Wire 调试（SWD）模式；
2. 直接调试访问所有存储器、寄存器和外设；
3. 调试期不需要目标源；
4. 最高支持 4 个断点；
5. 支持 2 个可作为触发器的数据观察点。

21.2 SWD 引脚说明

Pin Name	Type	Description	GPIO Configuration
SWCLK	I/O	SWD 模式下的 Serial Wire 时钟引脚；	PD6
SWDIO	I/O	SWD 模式下的 Serial Wire 数据输入/输出引脚；	PD6

21.3 SWD 调试注意事项

21.3.1. 局限性

调试模式改变了 ARM Cortex-M0 CPU 用于降低功耗的工作模式，而这一影响还会波及到整个系统。这意味着不应该在调试阶段进行功耗测量，否则测量结果会比正常应用操作时偏高。

在调试阶段，每当 CPU 停止工作时，SysTick 定时器会自动停止工作，其它的外设则不受影响。

21.3.2. 恢复调试功能

用户代码可以禁止 SWD 功能以便作为 GPIO 使用，此时不能通过 SWD 功能来调试或

者下载 FW。

在引导程序期间，AD3005 提供引导加载程序来检查 RST（引导引脚）的状态，若 PA1 为低电平，MCU 会用引导加载程序代替用户程序。

21.3.3. SWD 引脚上的内部上拉/下拉电阻

为了避免不受控制的 I/O 电平，在 SWD 输入引脚上内置了上拉电阻和下拉电阻。

- SWDIO/JTMS：内部上拉；
- SWCLK/JTCK：内部下拉。

软件一旦禁止了 SWD 功能，GPIO 控制器将再次控制相关功能。

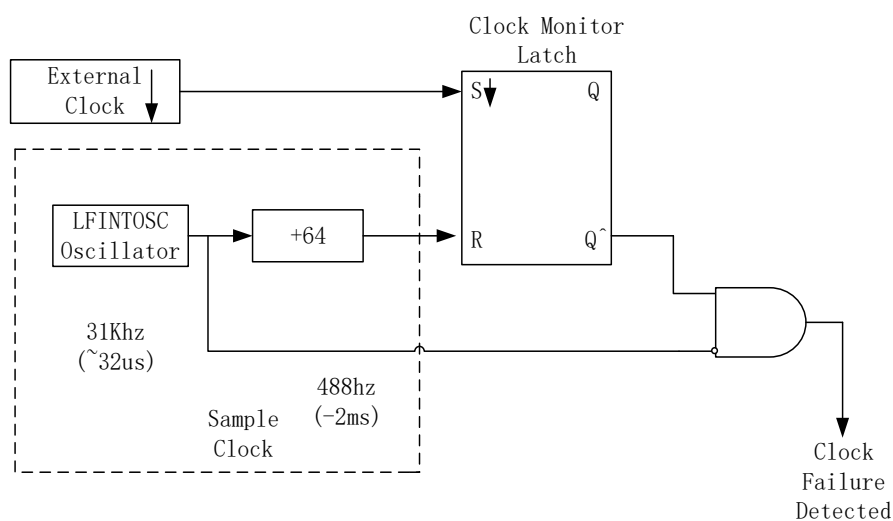
22 FSCM 时钟监测

22.1 自动防故障装置时钟监控

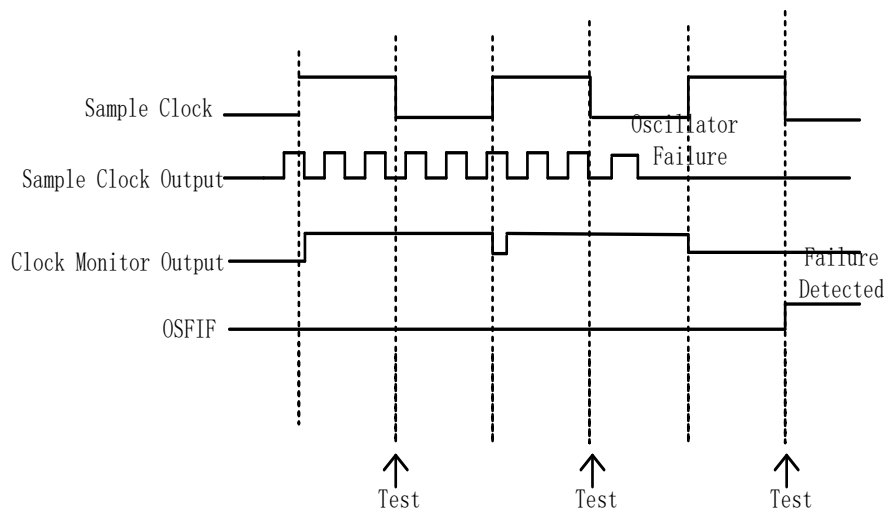
故障安全时钟监视器（FSCM）允许设备在外部振荡器或外部时钟发生故障时继续运行。如果选择振荡器模式，FSCM 可以在振荡器启动定时器（OST）过期后的任何时间检测振荡器故障。当选择外部时钟模式时，FSCM 可以在设备从复位释放后立即检测到故障。FSCM 是通过设置 FCMEN 位来启用的。FSCM 适用于外部振荡器（EHS）模式。

22.2 自动防故障装置检测

FSCM 模块通过监控时钟下降沿并使用 LFINTOSTC 作为时间基准来检测故障的振荡器，见<FSCM 结构示意图>，一个故障的振荡器的检测将花费 LFINTOSTC 的 32 到 96 个周期。见图 21-2 显示了 FSCM 模块的时序图。



FSCM 结构图



FSCM 时序图

22.3 故障安全操作

当检测到外部时钟出现故障时，FSCM 将 CPU 时钟切换到内部时钟源 IHRC8M 时钟，并设置中断标志 FSCMIF。当设置中断使能 FSCMIE 位时，会产生一个中断，如果启用了 SYS0_CSST 寄存器中的 FSCMIE 位，然后中断服务例程（ISR）中的用户固件可以采取步骤来减轻故障时钟可能出现的问题。在故障安全状态清除之前，系统时钟将继续从内部时钟源获取。

22.4 故障安全条件清除

当存在故障安全条件时，用户必须采取以下操作来清除该条件，然后才能使用外部源恢复正常运行。故障发生后，自动切换为内部时钟 IHRC8M 时钟继续工作。

22.5 重置或从睡眠中唤醒

FSCM 用于检测外部振荡器故障。当 FSCM 与外部振荡器一起使用时，振荡器启动计时器（OST）计数必须在 FSCM 激活之前过期。OST 是在从睡眠中唤醒和任何类型的重置之后使用的。当 FSCM 启用时，俩速启动是同时启用。因此当 OST 运行时，设备总是在执行代码。

22.6 FSCM 寄存器

基地址：

偏移地址：

Bit	Name	Description	Attribute	Reset
-----	------	-------------	-----------	-------

31:5	Reserved		R	0
7	FSCMIF	FSCM 中断标志位，写 1 清中断标志 0: 未产生中断; 1: 产生中断;	R	0
6	FSCMIE	FSCM 中断使能标志位 0: 关闭; 1: 开启;	R	0
5	FSCMEN	FSCM 使能位 0: 关闭; 1: 开启;	R	0
4	PLLRDY	PLL 时钟准备标志位 0: PLL 未锁定; 1: PLL 已锁定;	R	0
3	EHSRDY	外部高速时钟准备标志位 0: EHS 振荡器没有准备好; 1: EHS 振荡器已经准备好;	R	0
2	ELSRDY	外部低速时钟准备标志位 0: ELS 振荡器没有准备好; 1: ELS 振荡器已经准备好;	R	0
1	ILRCRDY	ILRC 时钟准备标志位 0: ILRC 没有准备好; 1: ILRC 已经准备好;	R	0
0	IHRCRDY	IHRC 时钟准备标志位 0: IHRC 没有准备好; 1: IHRC 已经准备好;	R	1

23 TSENSOR 温度传感器

注：TSENSOR 时钟一般不高于 16MHz。

23.1 TSENSOR 寄存器定义

基地址：0x4000 F000

23.1.1. TSCFG1 配置寄存器

偏移地址：0x000

Bit	Name	Description	Attribute	Reset
31:24	MASK_TSIE	中断使能；	R/W	0
23:16	TSIC	中断清除；	R/W	0
15	TSIE	总中断使能位；	R/W	0
14: 6	Reserved		R	0
5	TSEN	温度传感器的使能位 1：使能； 0：禁止；	R/W	0
4	Reserved		R	0
3	DEMEN	抽取滤波电流的使能位 1：使能； 0：禁止；	R/W	0
2:0	CHSEL	通道温度校准；	R/W	0

23.1.2. TSCFG2 配置寄存器

偏移地址：0x004

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TEMPAH	高温报警；	R/W	0

23.1.3. TSCFG3 配置寄存器

偏移地址：0x008

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0

29:21	MDLY		R/W	0
20:18	SYSPRE	Tsensor 时钟分频 000: 128 001: 64 010: 32 011: 16 100: 8 101: 4 110: 2 111: 1	R/W	0
17:16	CLKSEL	时钟选择;	R/W	0
15:0	DOWNRATE		R/W	0

24 电气参数

24.1 直流电气特性

VDD = 3.3V, T = 25°C

附表 1: 直流参数

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD		1.25MHz~72MHz	2.0		3.6	V
输入漏电	Vleak	所有输入脚	VIN=VDD, 0			±1	μA
输入高电平	Vih	所有输入脚		0.4/0.7VDD		VDD	V
输入低电平	Vil	所有输入脚		0		0.2/0.3VDD	V
上拉电阻 RSEL=1	Rpu	PA0-7 PB0-7	VIN=0		100		Kohm
上拉电阻 RSEL=0	Rpu	PA0-7 PB0-7	VIN=0		30		Kohm
下拉电阻 RSEL=1	Rpd	PA0-7 PB0-7	VIN=VDD		100		Kohm
下拉电阻 RSEL=0	Rpd	PA0-7 PB0-7	VIN=VDD		30		Kohm
输出高电平驱动电流	Ioh	所有输出脚	Voh=VDD-0.5V		9.6		mA
输出低电平驱动电流	Iol	所有输出脚	Vol=0.5V		13.7		mA
静态功耗	Ids	VDD	LVR 关 WDT		0.5	1	μA
			开 WDT		3	10	
			开 LVR		2	5	
			开 RTC		10	20	
动态功耗	Idd	VDD	VDD=3.3V Fosc=20MHz 无负载			2	mA

附表 2: 运放参数

符号	条件	最小	典型	最大	单位
GBWP	VDD=2.0V~3.7V, T=27°C		3.69		MHz
SR	VDD=2.0V~3.7V, T=27°C		13.26		V/μs
Offset	VDD=2.0V~3.7V, T=27°C		10		mV

CMRR	VDD=2.0V-3.7V, T=27℃		74.17		dB
AOL	VDD=2.0V-3.7V, T=27℃		115.08		dB

24.2 交流电气特性

VDD = 3.3V, T = 25℃

特性	符号	条件	最小	典型	最大	单位
外部晶振频率	Fhosc	T=25℃ VDD=2.0-3.6V, Cp=12.5pF	4	16	16	MHz
	Flosc	T=25℃ VDD=2.0-3.6V, Cp=20pF		32.768		KHz
内部高频 RC 振荡频率	Fhrc	T=25℃ VDD=2.0-3.6V		32		MHz
WDT 振荡器频率	Fwdt	T=25℃ VDD=3.3V		256		KHz
振荡器起振时间	Toxov	T=25℃ VDD=3.3V		120		ms

24.3 LVR 电气特性

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
VLVR	低电压复位电压	3.6V	LVR 使能	1.6		3.2	V

24.4 LDO 电气特性（内核电压）

参数	条件	最小	典型	最大	单位
Output voltage			1.5		V
Power Supply Current	Normal mode		49		uA
Load regulation	LDO_OSC(for hfosc & pll)		2		mA
	LDO_OUT(for digital)		18		
Line regulation	2.0V-3.6V, Iload=1mA			0.3	%

24.5 BGR 电气特性

参数	条件	最小	典型	最大	单位
Reference voltage	VDD=3.3V, 25°C		1.0		V
Ref. voltage accuracy	After calibration, 25°C		±0.5		%
Temperature Stability	-40~85°C		13		ppm/°C
Power Supply Current	Normal mode		12		uA
Voltage Drift	VDD=2.0V~3.6V			0.1	%

24.6 Recommended Operating Conditions

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Operating Supply Voltage on VDD	VDD		2.2	—	3.6	V
Operating Supply Voltage on VIO	VIO		2.2	—	VDD	V
			1.71	—	VDD	V
System Clock Frequency	fSYSCLK		—	—	73.5	MHz
Operating Ambient Temperature	TA		-400	—	105	°C

24.7 Power Consumption

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Digital Core Supply Current						
Normal Mode-Full speed with code executing from flash	IDD	FSYSCLK = 72 MHz (HFOSC1) ²	—	12.9	15	mA
		FSYSCLK = 24.5 MHz (HFOSC0) ²	—	4.2	5	mA
		FSYSCLK = 1.53 MHz (HFOSC0) ²	—	625	1050	μ A
		FSYSCLK = 80 kHz ³	—	155	575	μ A

Idle Mode—Core halted with peripherals running	I _{DD}	FSYSCLK = 72 MHz (HFOSC1) ²	—	9.6	11.1	mA
		FSYSCLK = 24.5 MHz (HFOSC0) ²	—	3.14	3.8	mA
		FSYSCLK = 1.53 MHz (HFOSC0) ²	—	520	950	μ A
		FSYSCLK = 80 kHz ³	—	135	550	μ A
Suspend Mode—Core halted and high frequency clocks stopped, Supply monitor off.	I _{DD}	LFO Running	—	125	545	μ A
		LFO Stopped	—	120	535	μ A
Snooze Mode—Core halted and high frequency clocks stopped. Regulator in low-power state, Supply monitor off.	I _{DD}	LFO Running	—	23	430	μ A
		LFO Stopped	—	19	425	μ A
Stop Mode—Core halted and all clocks stopped, Internal LDO On, Supply monitor off.	I _{DD}		—	120	535	μ A
Shutdown Mode—Core halted and all clocks stopped, Internal LDO Off, Supply monitor off.	I _{DD}		—	0.2	2.1	μ A
Analog Peripheral Supply Currents						
High-Frequency Oscillator	I _{HFOSC}	Operating at 34.354 MHz, T _A = 25 ° C	—	137.6	—	μ A
Low-Frequency Oscillator	I _{LFOSC}	Operating at 256 kHz, T _A = 25 ° C	—	2.59	—	μ A
ADC ⁴	I _{ADC}	1 Msps, 12-bit conversions Normal bias settings	—	3.083	—	mW

		$V_{CC} = 3.3\text{ V}, V_{DD} = 1.5\text{ V}$				
Internal ADC0 Reference ⁵	IVREFFS	High Speed Mode	—	285.2	—	μA
Temperature Sensor	ITSENSE		—	262		μA
Digital-to-Analog Converters (DAC0, DAC1) ⁶	IDAC		—	2.24	—	mW
Voltage Supply Monitor (VMON0)	IVMON		—	15	20	μA

24.8 Reset and Supply Monitor

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
VDD Supply Monitor Threshold	V _{VDDM}		1.95	2.0	2.05	V
Power-On Reset (POR) Threshold	V _{POR}	Rising Voltage on VDD	—	1.2	—	V
		Falling Voltage on VDD	0.91	—	1.49	V
VDD Ramp Time	t _{RMP}	Time to V _{DD} > 2.2 V	10	—	—	μs
Reset Delay from POR	t _{POR}	Relative to V _{DD} > V _{POR}	3	10	31	ms
Reset Delay from non-POR source	t _{RST}	Time between release of reset source and code execution	—	50	—	μs
RST Low Time to Generate Reset	t _{RSTL}		15	—	—	μs
Missing Clock Detector Response Time (final rising edge to reset)	t _{MCD}	FSYSCLK > 1 MHz	—	0.625	1.2	ms
Missing Clock Detector Trigger Frequency	F _{MCD}		—	7.5	13.5	kHz
VDD Supply Monitor Turn-On Time	t _{MON}		—	2	—	μs

24.9 Flash Memory

Parameter	Symbol	Test Condition	Min	Typ	Max	Units
Write Time ^{1,2}	t _{WRITE}	One Byte, FSYSCLK = 24.5 MHz	19	20	21	μs
Erase Time ^{1,2}	t _{ERASE}	One Page, FSYSCLK = 24.5 MHz	5.2	5.35	5.5	ms
VDD Voltage During Programming ³	V _{PROG}		2.2	—	3.6	V
Endurance (Write/Erase Cycles)	N _{WE}		20k	100k	—	Cycles
CRC Calculation Time	t _{CRC}	One 256-Byte Block SYSCLK = 48 MHz	—	5.5	—	μs

24.10 Power Management Timing

Parameter	Symbol	Test Condition	Min	Typ	Max	Units
Idle Mode Wake-up Time	t _{IDLEWK}		2	—	3	SYSCLKs
Suspend Mode Wake-up Time	t _{SUS- PENDWK}	SYSCLK = HFOSCO CLKDIV = 0x00	—	170	—	ns
Snooze Mode Wake-up Time	t _{SLEEPWK}	SYSCLK = HFOSCO CLKDIV = 0x00	—	12	—	μs

24.11 Internal oscillators

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
High Frequency Oscillator (32 MHz)						
Oscillator Frequency	f _{HFOSCO1}	Full Temperature and Supply Range	—	34.354	—	MHz
Power Supply Sensitivity	PSS _{HFOSC 1}	T _A = 25 °C	—	32.747	—	ppm/V

Temperature Sensitivity	TS _{HFOSC1}	V _{DD} = 3.3 V	—	19.328	—	ppm/° C
Low Frequency Oscillator (256 kHz)						
Oscillator Frequency	f _{LFOSC}	Full Temperature and Supply Range	—	256	—	kHz
Power Supply Sensitivity	PSS _{LFOSC}	T _A = 25 ° C	—	0.05	—	%/V
Temperature Sensitivity	TS _{LFOSC}	V _{DD} = 3.3 V	—	84.08	—	ppm/° C

24.12 External Clock Input

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
External Input CMOS Clock Frequency (at EXTCLK pin)	f _{CMOS}		0	—	50	MHz
External Input CMOS Clock High Time	t _{CMOSH}		9	—	—	ns
External Input CMOS Clock Low Time	t _{CMOSL}		9	—	—	ns

24.13 ADC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Resolution	N _{bits}	12 Bit Mode		10.45		Bits
Throughput Rate	f _S	12 Bit Mode	—	—	1	Msp/s
Power-On Time	t _{PWR}		—	10	—	μ s
SAR Clock Frequency	f _{SAR}		—	—	14	MHz
Voltage Reference Range	V _{REF}		1	—	V _{IO}	V
Input Voltage Range ²	V _{IN}		0	—	V _{REF} /G _{ain}	V
DC Performance						
Integral Nonlinearity	INL	12 Bit Mode	—	-4/+3	—	LSB
Differential Nonlinearity	DNL	12 Bit Mode	—	±1	—	LSB

(Guaranteed Monotonic)						
Offset Error ⁴	E _{OFF}	12 Bit Mode	—	7	—	LSB
Dynamic Performance 11.71875 kHz Sine Wave Input 1 dB below full scale, Max throughput, using AGND pin						
Signal-to-Noise	SNR	12 Bit Mode	—	66.5	—	dB
Signal-to-Noise Plus Distortion	SNDR	12 Bit Mode	—	64.6	—	dB
Total Harmonic Distortion (Up to 10 th Harmonic)	THD	12 Bit Mode	—	-69.3	—	dB
Spurious-Free Dynamic Range	SFDR	12 Bit Mode	—	73.8	—	dB

24. 14V Voltage Reference

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Internal Fast Settling Reference						
Output Voltage (Full Temperature and Supply Range)	V _{REFFS}			1.0		V
Temperature Coefficient	TC _{REFFS}	-40-85°C	—	13	—	ppm/°C
Turn-on Time	t _{REFFS}	-40-85°C, 精度 0.5%	—	0.04	—	ms
Power Supply Rejection	PSRR _{REFFS}		—	32.747	—	ppm/V
On-chip Precision Reference						
Valid Supply Range	V _{DD}	2.4 V Output	2.7	—	3.6	V
Output Voltage	V _{REFP}	2.4 V Output, V _{DD} = 3.3 V, T = 25 °C		2.4		V
		2.4 V Output		2.4		V
Turn-on Time, settling to 0.5 LSB	t _{VREFP}		—	3	—	ms
Short-circuit current	ISC _{VREFP}		—	—	8	mA
Power Supply Rejection	PSRR _{VREFP}		—	69.17	—	dB

External Reference						
Input Current	I_{EXTREF}	ADC Sample Rate = 1 Msps; V_{REF} = 3.0 V	—	5	—	μA

24.15 Temperature Sensor

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Uncalibrated Offset	V_{OFF}	$T_A = 0^\circ C$	—	751	—	mV
Uncalibrated Offset Error ¹	E_{OFF}	$T_A = 0^\circ C$	—	19	—	mV
Slope	M		—	2.82	—	mV/ $^\circ C$
Slope Error ¹	E_M		—	29	—	$\mu V/^\circ C$
Linearity	LIN	$T = 0^\circ C$ to $70^\circ C$	—	-0.1 to 0.15	—	$^\circ C$
		$T = -20^\circ C$ to $85^\circ C$	—	-0.2 to 0.35	—	$^\circ C$
		$T = -40^\circ C$ to $105^\circ C$	—	-0.4 to 0.8	—	$^\circ C$
Turn-on Time	t_{ON}		—	3.5	—	μs
Temp Sensor Error Using Typical Slope and Factory- Calibrated Off- set ^{2, 3}	E_{TOT}	$T = 0^\circ C$ to $70^\circ C$	-2.6	—	1.8	$^\circ C$
		$T = -20^\circ C$ to $85^\circ C$	-2.9	—	2.7	$^\circ C$
		$T = -40^\circ C$ to $105^\circ C$	-3.2	—	4.2	$^\circ C$

24.16 DACS

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Output Voltage	V_{OUT}		0	—	V_{DD}	V
Resolution	N _{bits}			11.92		Bits
Throughput Rate	f _S		—	—	1	Msps
Integral Nonlinearity	INL	DAC0 and DAC1	-0.15		0.15	LSB

Differential Nonlinearity	DNL		-0.15	—	1.2	LSB
Output Settling Time to 1% Full-scale	t _{SETTLE}	V _{OUT} change between 25% and 75% Full Scale	—	0.1	—	μs
Power-on Time	t _{PWR}		—	10	—	μs
Voltage Reference Range	V _{REF}		2	—	V _{DD}	V
Total Harmonic Distortion	THD	V _{OUT} = 11.71875 KHz sine wave, 10% to 90%	—	79.49	—	dB
Offset Error	E _{OFF}	V _{REF} = 2.4 V	—	208	—	mV
Full-Scale Error	E _{FS}	V _{REF} = 2.4 V	—	7.3	—	%

24.17 Configurable Logic

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Propagation Delay	t _{DLY}	Through single CLU Using an external pin	—	—	35.3	ns
		Through single CLU Using an internal connection	—	3	—	ns
Clocking Frequency	F _{CLK}	1 or 2 CLUs Cascaded	—	—	73.5	MHz
		3 or 4 CLUs Cascaded	—	—	36.75	MHz

24.18 Port I/O

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Output High Voltage (High Drive)	V _{OH}	I _{OH} = -7 mA, V _{IO} ≥ 3.0 V	V _{IO} - 0.7	—	—	V
		I _{OH} = -3.3 mA, 2.2 V ≤ V _{IO} < 3.0 V I _{OH} = -1.8 mA, 1.71 V ≤ V _{IO} < 2.2 V	V _{IO} × 0.8	—	—	V
Output Low Voltage (High Drive)	V _{OL}	I _{OL} = 13.5 mA, V _{IO} ≥ 3.0 V	—	—	0.6	V
		I _{OL} = 7 mA, 2.2 V ≤ V _{IO} < 3.0 V	—	—	V _{IO} × 0.2	V

		$I_{OL} = 3.6 \text{ mA}, 1.71 \text{ V} \leq V_{IO} < 2.2 \text{ V}$				
Output High Voltage (Low Drive)	V_{OH}	$I_{OH} = -4.75 \text{ mA}, V_{IO} \geq 3.0 \text{ V}$	$V_{IO} - 0.7$	—	—	V
		$I_{OH} = -2.25 \text{ mA}, 2.2 \text{ V} \leq V_{IO} < 3.0 \text{ V}$ $I_{OH} = -1.2 \text{ mA}, 1.71 \text{ V} \leq V_{IO} < 2.2 \text{ V}$	$V_{IO} \times 0.8$	—	—	V
Output Low Voltage (Low Drive)	V_{OL}	$I_{OL} = 6.5 \text{ mA}, V_{IO} \geq 3.0 \text{ V}$	—	—	0.6	V
		$I_{OL} = 3.5 \text{ mA}, 2.2 \text{ V} \leq V_{IO} < 3.0 \text{ V}$ $I_{OL} = 1.8 \text{ mA}, 1.71 \text{ V} \leq V_{IO} < 2.2 \text{ V}$	—	—	$V_{IO} \times 0.2$	V
Input High Voltage	V_{IH}		$0.7 \times V_{IO}$	—	—	V
Input Low Voltage	V_{IL}		—	—	$0.3 \times V_{IO}$	V
Pin Capacitance	C_{IO}		—	7	—	pF
Weak Pull-Up Current ($V_{IN} = 0 \text{ V}$)	I_{PU}	$V_{DD} = 3.6$	-30	-20	-10	μA
Input Leakage (Pullups off or Analog)	I_{LK}	$GND < V_{IN} < V_{IO}$	-1.1	—	4	μA
Input Leakage Current with V_{IN} above V_{IO}	I_{LK}	$V_{IO} < V_{IN} < V_{IO} + 2.5 \text{ V}$ Any pin except P3.0, P3.1, P3.2, or P3.3	0	5	150	μA

24.19 Thermal Conditions

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Thermal Resistance (Junction to Ambient)	θ_{JA}	QFN24 Packages	—	30	—	$^{\circ}\text{C/W}$
		QFN32 Packages	—	26	—	$^{\circ}\text{C/W}$
Thermal Resistance (Junction to Case)	θ_{JC}	QFN24 Packages	—	32.8	—	$^{\circ}\text{C/W}$
		QFN32 Packages	—	20	—	$^{\circ}\text{C/W}$

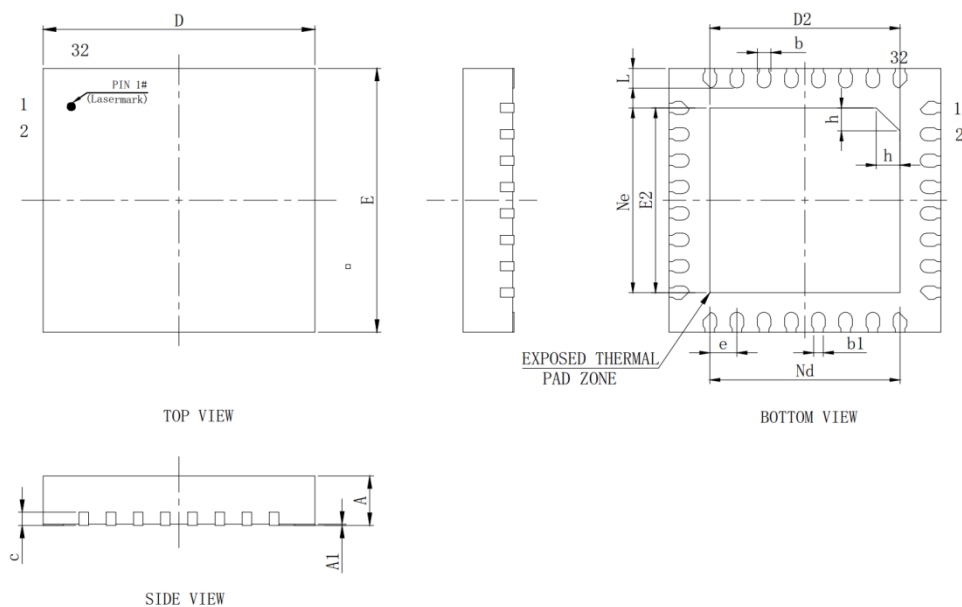
24. 20 Absolute Maximum Ratings

Parameter	Symbol	Test Condition	Min	Max	Unit
Ambient Temperature Under Bias	T _{BIAS}		-55	125	° C
Storage Temperature	T _{STG}		-65	150	° C
Voltage on VDD	V _{DD}		GND-0.3	3.6	V
Voltage on VIO ²	V _{IO}		GND-0.3	V _{DD} +0.3	V
Total Current Sunk into Supply Pin	I _{VDD}		—	400	mA
Total Current Sourced out of Ground Pin	I _{GND}		400	—	mA
Current Sourced or Sunk by any I/O Pin or RSTb	I _{IO}		-100	100	mA

25 封装信息

25.1 QFN32 封装图

大小: 4mm×4mm×0.75mm

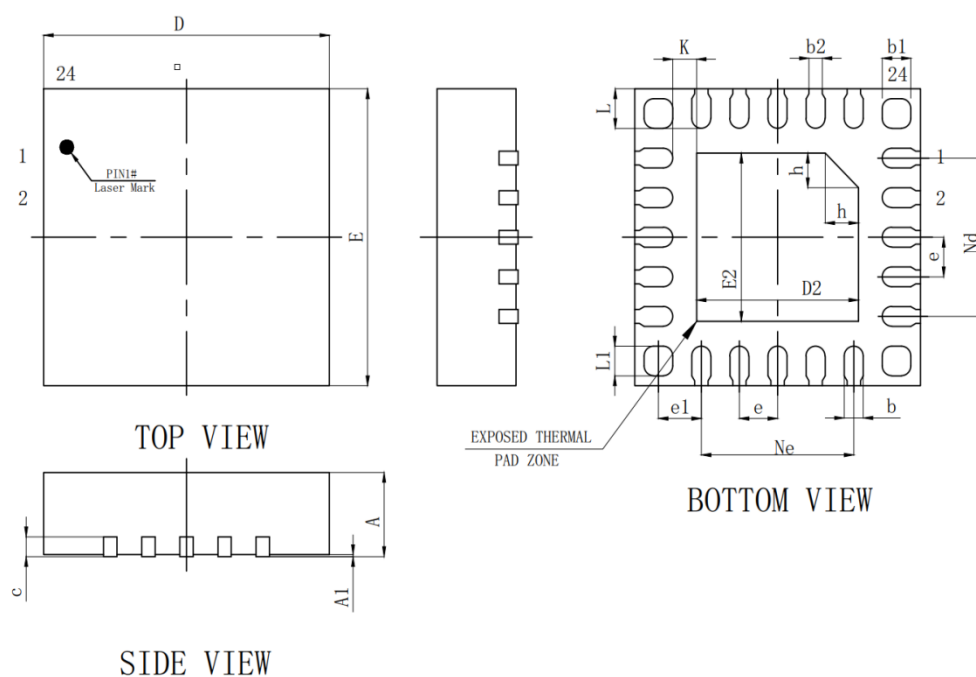


Dimension	Min	Typ	Max
A	0.70	0.75	0.80
	0.80	0.85	0.90
	0.85	0.90	0.95
A1	0	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.60	2.65	2.70
e	0.40 BSC		
Nd	2.80 BSC		
E	3.90	4.00	4.10
E2	2.60	2.65	2.70
Ne	2.80 BSC		
K	0.20	—	—

L	0.35	0.40	0.45
L1	0.30	0.35	0.40
L2	0.15	0.20	0.25
h	0.30	0.35	0.40

25.2 QFN24 封装图

大小: 3mm×3mm×0.85mm

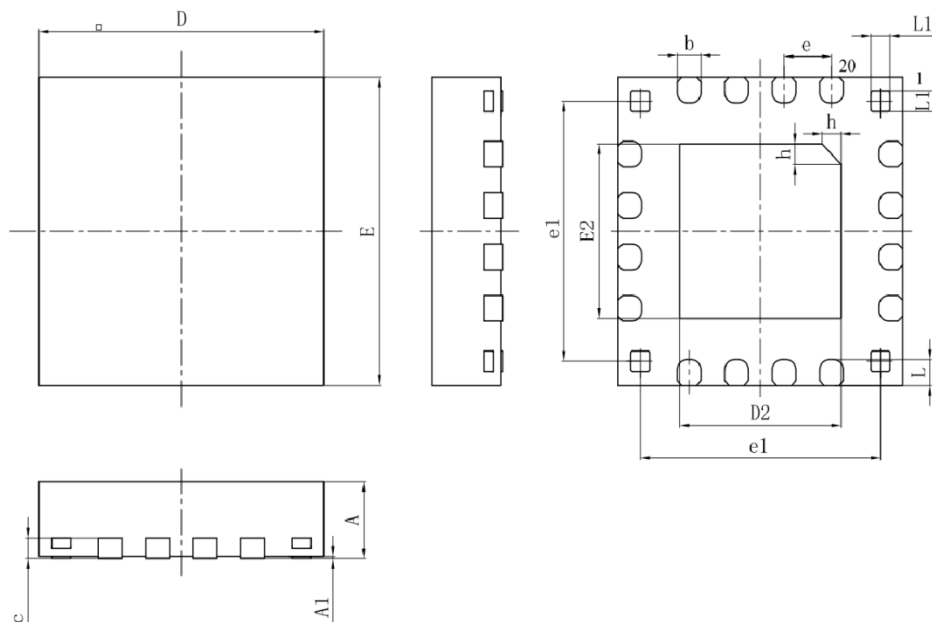


Dimension	Min	Typ	Max
A	0.80	0.85	0.90
	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
b1	0.25	0.30	0.35
b2	0.14 REF		
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.60	1.70	1.80
Ne	1.60 BSC		
Nd	1.60 BSC		
e	0.40 BSC		
e1	0.45 BSC		
E	2.90	3.00	3.10
E2	1.60	1.70	1.80

L	0.35	0.40	0.45
L1	0.25	0.30	0.35
h	0.30	0.35	0.40
k	0.20	0.25	0.30

25.3 QFN20 封装图

大小: 3mm×3mm×0.75mm



Dimension	Min	Typ	Max
A	0.70	0.75	0.80
A1	—	0.02	0.05
b	0.15	0.25	0.35
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.60	1.70	1.80
e	0.50 BSC		
e1	2.53 BSC		
E	2.90	3.00	3.10
E2	1.60	1.70	1.80
L	0.15	0.25	0.35
L1	0.15	0.20	0.25
h	0.10	0.20	0.30

26 订购信息

丝印信息

现行ADUC的单片机表面印有一栏信息：产品代码和日期码。

		Marking	B305	1	19	x
			Device code	Year	Week	Internal Usage
			Year: 0: 2020; 1: 2021			
			Week: 01: 第01周; 52:第52周			

标签信息

货品内外包装上粘贴的标签上包含：产品名称，封装信息，芯片批号，丝印信息，出货日期及包装数量。

		产品名称	Part No: AD3005Q32-6XTR Package:QFN32 Lot No: NN4SL49 Marking : B305119 Date: 2021-07-09 QTY: 40000pcs			
		封装信息				
		芯片批号				
		丝印信息				
		出货日期				
		包装数量				

空片

采购信息

AD3005				
产品名称	封装信息	FLASH	工作温度	包装方式
AD3005Q32-6XTR	QFN32, 绿色封装	32K	-40-85°C	Tape & Reel
AD3005Q32-8XTR	QFN32, 绿色封装	64K	-40-85°C	Tape & Reel
AD3005Q24-6XTR	QFN24, 绿色封装	32K	-40-85°C	Tape & Reel
AD3005Q20-6XTR	QFN20, 绿色封装	32K	-40-85°C	Tape & Reel